

# Short-term electricity demand and gas price forecasts using wavelet transforms and adaptive models

Hang T. Nguyen<sup>a,\*</sup>, Ian T. Nabney<sup>a</sup>

<sup>a</sup>*Non-linearity and Complexity Research Group, School of Engineering and Applied Science,  
Aston University, Aston Triangle, Birmingham, B4 7ET, UK*

---

## Abstract

This paper presents some forecasting techniques for energy demand and price prediction, one day ahead. These techniques combine wavelet transform (WT) with fixed and adaptive machine learning/time series models (multi-layer perceptron (MLP), radial basis functions, linear regression, or GARCH). To create an adaptive model, we use an extended Kalman filter or particle filter to update the parameters continuously on the test set. The adaptive GARCH model is a new contribution, broadening the applicability of GARCH methods. We empirically compared two approaches of combining the WT with prediction models: multicomponent forecasts and direct forecasts. These techniques are applied to large sets of real data (both stationary and non-stationary) from the UK energy markets, so as to provide comparative results that are statistically stronger than those previously reported. The results showed that the forecasting accuracy is significantly improved by using the WT and adaptive models. The best models on the electricity demand/gas price forecast are the adaptive MLP/GARCH with the multicomponent forecast; their MSEs are 0.02314 and 0.15384 respectively.

*Key words:* multi-layer perceptron, radial basis function, GARCH, linear regression, adaptive models, and wavelet transform.

---

---

\*Corresponding author, Tel: +44 121 257 7718, Fax: +44 121 204 3685, email address: thihangn@aston.ac.uk

## 1. Introduction

### 1.1. Context

An important characteristic of electricity is that it cannot be stored. Therefore, an electricity demand forecast is valuable for power generators who can use such forecasts to schedule operations of their power stations to match generation capacity with demand. Electricity demand forecasting is also considered one of the fundamental pieces of information for trading in the energy market because the power price depends on demand.

Accurate electricity/gas price forecasting is very important for traders in the energy market, especially those associated with energy generators. If an energy generator makes an accurate forecast of the market price, it can develop a strategy to maximise its own profits and minimise risk due to price spikes by appropriate trading in forward contracts. It can also plan its actions to maximise benefits or utilities by reducing/increasing its generation. Energy suppliers can use short-term price forecasts to adjust their bidding strategies to achieve the maximum benefit. In addition, understanding the process of forward price development can help the generators make profits through trading on the forward market.

A number of statistical methodologies have been proposed for energy price

---

#### <sup>0</sup>Abbreviations:

ACF	autocorrelation function	MAPE	mean absolute percent error
ANN	artificial neural network	MLP	multi-layer perceptron
AR	autoregressive	MSE	mean squared error
ARMA	autoregressive moving average	NMSE	normalised mean squared error
ARIMA	autoregressive integrated moving average	NN	neural networks
ARD	automatic relevance determination	NSE	normalised squared errors
AVCM	absolute values of correlation matrix	PACF	partial autocorrelation function
BM	benchmark model	PF	particle filter
CM	correlation matrix	RBF	radial basis function
EKF	extended Kalman filter	RHWT	redundant Haar wavelet transform
GARCH	generalised autoregressive conditional heteroschedastic	RW	random walk model
IR	improvement ratio	SP	sum of P-values
LR	linear regression	SSM	state space model
MAE	mean absolute error	WT	wavelet transform

and demand forecasting. Many approaches based on time series models have been used for price forecasting, such as AR models [1], autoregressive integrated moving average (ARIMA) models [2] [3] [4], and generalised autoregressive conditional heteroschedastic models (GARCH) [5]. Moreover, neural networks (NNs) are used widely for electricity price/demand forecasting in the literature [6], [7]. Due to the complexity of the environment, the functional relationships we are looking for might be non-linear. Several researchers have proposed additional procedures to improve accuracy, such as pre-processing procedures [8] and regularisation methods [6]. Another approach for improving forecasting performance is multiple NNs. The use of a committee of NNs for forecasting is suggested in [9]. Similarly, cascaded neural networks are proposed in [10]. Pao [11] proposed new hybrid non-linear models that combine a linear model with an artificial neural network (ANN). A novel configuration combining an AR(1) with a high pass filter was presented in [12].

This paper uses several standard forecasting models such as the multi-layer perceptron (MLP), radial basis function (RBF), linear regression (LR), and generalized autoregressive conditional heteroschedastic (GARCH) model. We propose two techniques to improve performance of these prediction models. The first technique is to use a wavelet transform (WT) as a pre-processing procedure. The WT can produce a good local representation of the signal in both the time and frequency domains. This technique has previously been used [2], [8], [13], and [14]. However, this paper makes some contributions as explained in Section 1.2.

The second technique is a hybrid framework to create an adaptive forecasting model, which is a combination of a filter (such as Kalman filter (KF), or Extended Kalman Filter (EKF) or particle filter (PF)) and a standard prediction model (such as RBF, MLP, LR or GARCH). The forecast model is used to forecast the next value of a time series, and the filter is used to update parameters for the forecast model online whenever a new value of the time series is observed. There is a large range of previous papers applying this hybrid framework to different applications, such as recursively re-estimating parameters of the Black-Scholes model from observations (the Black-Scholes model is a well-known financial model for options pricing) [15], online learning parameters

of RBF and linear regression models [16] [17], predicting exchange rates [18], estimating wind turbine power generation [19], and predicting New England electricity prices [20]. The results on these previous papers show that this hybrid framework is good in the speed of learning and the accuracy of predictions.

### 1.2. Paper contributions

This paper provides an empirical comparison of a set of machine learning/time series forecasting models in order to explore the following issues:

- The choice of the type of prediction model from MLP, RBF, LR, and GARCH.
- The value of a transformation of the target variable prior to modelling, using the redundant Haar wavelet transform (RHWT). We compare the prediction performance of prediction models without RHWT and two combination methods:
  - Multicomponent forecast: a RHWT decomposes the target value  $y$  into wavelet components, and then each component is forecast with a separate model.
  - Direct forecast: using the components of RHWT as input variables to a single forecast model to directly predict the target.
- Model parameters are either estimated just once or continuously updated in the testing period. We evaluated the performance of the standard forecast methods (i.e. MLP/RBF/LR/GARCH) with two variations:
  - Fixed forecast models, i.e. models whose parameters are fixed after training on a training set.
  - Adaptive forecast models, i.e. hybrid of filters (EKF/PF) and machine learning/time series forecast models, where parameters are estimated on a training set and then adapted continuously on the test set using the filter.

We tested these models for forecasting one-day ahead electricity demand and one-day ahead gas forward price in the UK market.

This paper is an extension of our previous conference paper [21]. In this new paper, we combine the WT with a wider range of models, including not only fixed models as in [21] but also adaptive models. In [21], we tested the forecast models on a single small dataset (6 sub-datasets of gas forward prices). In this new paper we carry out experiments on two large datasets: (1) electricity daily demand with 821 observations, and (2) 72 sub-datasets of gas forward prices; each sub-dataset includes approximately 200 observations. Therefore, the results in this paper compare more models and are more statistically robust than the previous one.

Compared with earlier work, our paper has the following contributions. First, we propose new forecasting methods which are the combination of WT, a range of machine learning/time series models, and filters. Second, although combining WT with a time series or neural network models has already appeared, previous papers only used either multicomponent forecast or direct forecast. In this paper, we use both types of forecast and compare their prediction accuracy, which provides an answer to the question of which is better for energy datasets. The experimental results on the UK data showed that multicomponent forecasts outperform models without WT and direct forecasts. Third, we combine filters (EKF/PF) with machine learning/time series models to create adaptive models, whose parameters are updated online during forecasts. Among these adaptive models, the adaptive GARCH model is proposed for the first time in this paper. Moreover, we use not only the EKF for adaptive models as earlier authors but also the PF. The benefits of using the PF are that it makes no *a priori* assumption of Gaussian noise and also that it is not necessary to linearise the prediction model. Fourth, besides historical data of target variable (e.g. electricity demand or gas forward price) and its WT components, a number of exogenous variables (e.g. temperature, wind speed, day pattern, electricity supply and electricity price etc.), are also considered as input variables. Some pre-processing procedures (presented in Section 5.2) are used to choose the relevant input variables for each forecasting model. Finally, beside the standard error measures, we also provided two additional measures for comparing and ranking models performance, i.e. improvement ratio (presented in Section 5.3.2) and hypothesis testing for multiple models (presented in Section 5.3.3).

### 1.3. Paper structure

This paper is organised as follows. In Section 2, the RHWT for decomposing data into components is presented. Section 3 defines the detailed forecasting framework. The fixed forecasting models and the method for creating their corresponding adaptive models are described in Section 4. Numerical results and evaluation on data from the UK energy markets are given in Section 5. Section 6 provides some conclusions.

## 2. Redundant Haar wavelet transform

### 2.1. Why RHWT?

Most research in the literature uses symmetric WTs, such as Daubechies, Morlet, or Symlet, for forecasting applications [2], [8], [13], and [14]. However, using this type of WT for prediction is not appropriate because in the symmetric wavelet, the wavelet coefficients take into account not only previous information but also future information (see Figure 1), but in a forecasting problem, we can only use data obtained earlier in time. Some earlier papers have mentioned this problem and use an à trous wavelet transform or a redundant Haar wavelet transform in financial time series forecasting [22] and electricity load forecasting [23], [24], [25].

In addition, a discrete WT normally has two stages: (1) computing detail and approximation coefficients with high- and low-pass filters, and (2) decimation, i.e. retaining one data point out of every two. The main advantage of decimation is reducing the storage requirement. However, decimation leads to the loss of phase information. To overcome this, we can use a redundant or non-decimated wavelet transform [23]. In a redundant WT, only stage (1) is completed. All components of a redundant WT have the same length as the original time series. Therefore, there is a one-to-one correspondence between the original data and decomposition coefficients at a given time step.

### 2.2. Computing RHWT

Assuming that there is a time series  $y_t$ ,  $t = 1, 2, \dots, T$ , Figure 2 shows how to compute its RHWT coefficients to the  $n$ -th decomposition level. At level  $i$ , the detail coefficients  $D_i$  are retained, while the approximation coefficients  $A_i$  are

decomposed into a further level of detail  $D_{i+1}$  and approximation coefficients  $A_{i+1}$ . It can be shown that the original time series can now be reconstructed as  $y_t = A_{n,t} + D_{n,t} + \dots + D_{1,t}$ .

Note that to calculate a coefficient at level  $i + 1$  at time  $t$  ( $A_{i+1,t}$  or  $D_{i+1,t}$ ), we need to use the value of time series  $A_i$  at time step  $t - 2^i$ . Therefore, at level  $i + 1$ , it is impossible to exactly define the value of these coefficients before time step  $2^{i+1} - 1$ . After applying the RHWT, this paper will consider only those coefficients after time step  $2^n - 1$ . Each component represents the data in a frequency range that is less volatile and easier to forecast than the original time series  $y$ .

### 3. Forecasting frameworks with WT

#### 3.1. Multicomponent forecast

The multicomponent forecasting framework is shown in Figure 3. This is a combination of forecasting models and the RHWT. A dataset is divided into two sub-datasets: (1) a training set to estimate the model parameters and (2) a test set to evaluate performance of these models by calculating appropriate error functions. The forecasting framework for a time series  $y_t$  consists of four steps:

*Step 1:* Use the RHWT to decompose  $y_t$  of the training set and the test set separately:  $A, D_n, D_{n-1}, \dots, D_1$ .

*Step 2:* Determine the input vectors (including any exogenous variables) for each model for predicting each component. Details of how we determined the input variables will be presented in Section 5.2.

*Step 3:* In the training phase, the training sets are used to develop forecasting models (i.e. estimate parameters), one forecasting model for each component.

*Step 4:* In the test phase, the developed models are used to predict the future value of the components from the current observable data. The outputs of these models at time  $t$  are the values of  $A, D_n, D_{n-1}, \dots, D_1$  at time step  $t + 1$ . In this work, the models used for forecasting are fixed or adaptive MLP/RBF/GARCH/LR. The inverse WT is used to compute the forecast value of  $y_t$  from the predictions of the components.

### 3.2. Direct forecast

Like the multicomponent forecast method, the target time series  $y_t$  in this method (shown in Figure 4) is also decomposed into WT components. These components and exogenous variables are also used as candidates for input variables. However, the main difference between the two methods is that the direct forecast method directly predicts the time series  $y_t$  while the multicomponent forecast method separately forecasts wavelet transform components and the forecast value of  $y_t$  is derived from these forecast components by using the inverse wavelet transform. Therefore, the direct forecast method has only one forecast model while the multicomponent forecast method has several forecast models, one model for each WT component.

## 4. Fixed and adaptive forecasting models

This section summarises several forecasting models previously used in energy markets: MLP, RBF, GARCH, and LR. We also describe how to produce the corresponding adaptive models using filters. The parameters of a fixed prediction model are estimated using the training set only, and the test set is not used to adjust parameters. These fixed forecasting models work well on stationary datasets. However, their performance degrades in predicting non-stationary datasets. The characteristics of a non-stationary time series change over time; thus the trend and volatility of training set might be different from these quantities of the corresponding test set. Therefore, the parameters of the prediction model, which are inferred from the training set, become “out of date” after some time. This means that these parameters might no longer capture the correct characteristics of the test set and this might lead to poor prediction performance. One approach to reduce the effect of the above issue is to use adaptive models.

### 4.1. Fixed models

This paper uses four standard prediction models: MLP, RBF, LR, and GARCH.

#### Neural networks



An MLP consists of a number of perceptrons organised in layers. Each perceptron has several inputs and one output, which is a non-linear function of the inputs. It has been shown that networks with one hidden layer are capable of approximating any continuous functional mapping if the number of hidden units is large enough [26]. Therefore, only two-layer networks will be considered in this paper.

The RBF network is the main alternative to MLP for non-linear modeling by neural network. The hidden unit in the RBF model computes a non-linear function of the distance between the input vector and a weight vector.

### **Linear regression**

Linear regression (LR) is a simple model where the output is a linear combination of inputs. Unlike AR, ARMA, or ARIMA models, the input vector of a LR can include both historical values of target variables and exogenous variables (e.g. temperature, other components of RHWT, etc.).

Our LR is equivalent to AR with exogenous variables. We computed the autocorrelation function (ACF) of the residuals of LR models and found that the autocorrelation was very small for all non-zero lags, thus there is no need for moving average terms. We also built forecasting models for log returns (i.e. time series  $r_t = \log(p_t/p_{t-1})$ , where  $p_t$  is gas price), but they did not performed as well. Therefore in our paper using LR is enough, neither ARMA nor ARIMA is necessary.

### **GARCH**

In MLP, RBF or LR, the errors are assumed to be homoschedastic (i.e. the variance of the residual is assumed to be independent of time). The GARCH [27] can be used to model changes in the variance of the errors as a function of time.

More detailed descriptions of these four prediction models are presented in Appendix A.

#### *4.2. Adaptive model framework*

In an adaptive model, a filter (extended Kalman filter (EKF) or particle filter (PF)) will be used to update parameters of a model by treating the weights as the states of a non-linear dynamic system. This can be considered as an estimation

problem where the weight values are unknown. The general framework for adaptive models is shown in Figure 5.

In the training phase, the training sets are used to estimate parameters of an MLP (or RBF/GARCH/LR) model in the usual way. In the test phase, two steps are repeated at each time step:

*Step 1:* When a new observation is available, the filter updates parameters of the predictive model.

*Step 2:* Use the predictive model with the latest estimated parameters to predict the next value.

In an adaptive model, we can choose to update all the parameters or only a subset. The experimental results showed that results on updating a subset are a little better and hence we restrict our attention to this case. The following sections describe the EKF and PF, and how to use them in adaptive models.

#### 4.3. Filters

This paper uses the EKF and the PF. They are based on a state space model (SSM); we assume that the observed time series  $y_t$  is a function of random variables  $z_t$  (the hidden state vector) which are not observed. In an adaptive model,  $z_t$  are the model parameters. It is also assumed that we do not know the dynamics of the observation, but do know the dynamics of hidden state space:

$$z_{t+1} = f_t(z_t) + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, Q_t) \quad (1)$$

$$y_t = h_t(z_t) + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, R_t), \quad (2)$$

where  $f_t$  and  $h_t$  are state transition function and output function respectively, and  $\varepsilon_t$  and  $\epsilon_t$  are zero-mean Gaussian noise;  $z_0$  is the system initial condition, modelled as a Gaussian random vector  $z_0 \sim \mathcal{N}(\pi_0, P_0)$ . For our task,  $f_t$  is the identity.

The EKF and PF algorithms track the posterior probabilities of hidden variables  $z_t$  given a sequence of observed variables up to time  $t$ :  $p(z_t | \{y\}_1^t)$ , where  $\{y\}_1^t = \{y_1, \dots, y_t\}$ . The EKF is an extension of the Kalman filter (KF) [28]. The difference is that the KF is designed for linear SSMs (i.e.  $h_t$  and  $f_t$  are linear functions) only while the EKF can be applied to either linear or non-linear models. The EKF does not solve the original problem, but approximates

it by locally linearising the non-linear functions  $h_t$  and  $f_t$  around previous state estimates. The linearisations are made with a first order Taylor expansion.

When the strength of the non-linearity of the functions  $h_t$  and  $f_t$  is great, the linearisations are poor approximations; and the EKF does not work well. The particle filter (PF) is an alternative method which avoids the bad effects of linearisation. The PF is a sampling-based method. In addition, EKF is limited to Gaussian noise for  $\epsilon_t$  and  $\varepsilon_t$  while there is no assumption of noise distributions on the PF. Details of the EKF and the PF are given in Appendix B.

#### 4.4. Adaptive MLP/RBF/LR models

In the adaptive MLP model, we update the bias of the second layer  $\omega_{k0}^{(2)}$  only; in the adaptive RBF model, we update the second layer weight  $\omega_{kj}$  only; and in the adaptive LR model, only the bias  $b$  is updated (see definition of these parameters in Appendix A). This implies that the models adapt only to changes in the mean of the time series. Adapting all the parameters gave worse results. Denote these updated parameters  $\theta$ , and the remaining parameters of a model  $\varpi$ . From equations (A.1), (A.2), (A.3), (A.4), and (A.5), we can summarise the input/output relationship of the models as follows

$$y_t = h(x_t, \theta, \varpi) \quad (3)$$

- **On the training set:** we used the same training algorithm in fixed model to estimate parameters, denoted  $\theta_0, \varpi_0$ .
- **On the test set:** two steps are recursively repeated.

*Step 1:* Update parameters of the model using the EKF/PF. The non-linear SSM is given by

$$\theta_t = \theta_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, Q) \quad (4)$$

$$y_t = h(x_t, \theta_t, \varpi_0) + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, R). \quad (5)$$

$\theta_t$  is the hidden state vector. Parameters  $Q, R$ , and  $P_0$  of the non-linear SSM can be estimated by using maximum log likelihood [29] or just set to reasonable values. Other parameters of non-linear SSM are given by

$$\pi_0 = \theta_0, F_t = I \text{ (i.e. identity matrix)}$$

$$H_t = \nabla h|_{\theta_t^{t-1}},$$

(see definition of these parameters in Appendix B).

*Step 2:* Predict one time step ahead:

$$y_t = h(x_t, \theta_t^{t-1}, \varpi_0).$$

#### 4.5. Adaptive GARCH model

In an adaptive GARCH, all the parameters are fixed at the estimates derived from the training set, with the exception of the bias  $\tilde{\beta}$  (see definition of this parameter in Appendix A.4) which is adapted on-line on the test set.

- **On the training set:** we use maximum likelihood to compute GARCH parameters as in fixed model (see Appendix A.4), and let  $\delta_0 = \{\alpha_{0,0}, \dots, \alpha_{m,0}, \gamma_{1,0}, \dots, \gamma_{r,0}\}$ ,  $\hat{\beta}_0$ , and  $\tilde{\beta}_0$  be these parameters.
- **On the test set:** We use a filter to update the value of  $\tilde{\beta}_t$ . For this purpose, a SSM is constructed as follows:

$$\tilde{\beta}_t = \tilde{\beta}_{t-1} + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, Q) \quad (6)$$

$$y_t = h_t(\tilde{\beta}_t) + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, R_t) \quad (7)$$

where the bias  $\tilde{\beta}$  is the hidden state vector, the output function  $h_t(\tilde{\beta}_t) = \tilde{\beta}_t + \hat{\beta}_0 x_t$ , and parameters  $Q$  and  $P_0$  of the SSM can be estimated by using maximum log likelihood (using the Kalman smoother) [29], or just set to reasonable values. The state transition function  $f_t$  is chosen to be identity (see Equation 6) because parameter  $\tilde{\beta}$  does a random walk: no pre-determined dynamics (i.e. entirely data dependant).  $F_t = 1$ ,  $H_t = 1$  (see definition of these parameters in Appendix B). Other parameters of the SSM are given by

$$\begin{aligned} \pi_0 &= \tilde{\beta}_0, \\ R_t &= \alpha_{0,0} + \sum_{i=1}^m \alpha_{i,0} \varepsilon_{t-i}^2 + \sum_{j=1}^r \gamma_{j,0} R_{t-j}. \end{aligned}$$

On the test set, two steps are recursively repeated through the observations in time order:

*Step 1:* Estimate parameter  $\tilde{\beta}_t$  of the GARCH model:  $\tilde{\beta}_t^{t-1} = E \left[ \tilde{\beta}_t | \{y\}_1^{t-1} \right]$ .

In order to obtain this parameter, we use the EKF/PF to estimate the mean of

the hidden state at time step  $t - 1$  given observations up to time  $t - 1$ :  $\tilde{\beta}_{t-1}^{t-1}$ . From equation (6), the estimate of  $\tilde{\beta}$  at time step  $t$  given  $\{y\}_1^{t-1}$  is  $\tilde{\beta}_t^{t-1} = \tilde{\beta}_{t-1}^{t-1}$ .

*Step 2:* Use the GARCH model with the latest estimated parameters to predict the time series at time  $t$ :

$$y_t = \tilde{\beta}_t^{t-1} + \hat{\beta}_0 x_t.$$

## 5. Experimental results

### 5.1. Data

We evaluated the performance of the algorithms on two problems: forecasting the daily electricity demand and forecasting the price of a monthly gas forward product (i.e. forward contract). Both datasets were taken from the UK energy market, and were provided by E.ON. The first dataset (see Figure 6) contains 821 observations of the daily total electricity demand ( $d$ ) of all users in Great Britain, from 7<sup>th</sup> October 2004 to 3<sup>rd</sup> May 2007. The data has two seasonality effects: (1) a weekly pattern with lower consumption at weekends and (2) an annual pattern where the consumption is higher during the colder part of the year. There are about 15 public holidays (Christmas, Easter and Bank holiday) per year. The demand values on public holidays are significantly smaller than on other days. There are two approaches for dealing with this issue. The first is to exclude these observations from the dataset. To forecast demand of public holidays, we should consider the corresponding day of previous years, and build a separate model for public holiday data only. The second is to include public holidays in the dataset, but introduce a dummy variable (equal to 1 for public holidays and 0 otherwise). Since the main objective of this paper is to evaluate the effects of WT and EKF/PF on the standard prediction models, we chose the first approach and have removed the observations on public holidays from the dataset. The first 525 observations were used as training set and the last 296 observations were used as the test set.

The second dataset (see Figure 7) consists of prices of monthly gas forward products. The monthly gas product is a forward contract for supplying gas in a single month in the future. This data is price of gas forward products Jun-2006 to May-2008 and is sampled daily from 1<sup>st</sup> December 2005 to 30<sup>th</sup> April 2008. In

the UK energy market, it is possible to trade gas from one to six months (before 1<sup>st</sup> May 2007) or five months (after 1<sup>st</sup> May 2007) ahead. Thus, there are five or six months of daily price data (approximately 110-130 data points) for each monthly gas product. For example, the Jul-2006 product can be traded from 3<sup>rd</sup> January 2006 to 30<sup>th</sup> June 2006. We created 72 sub-datasets; we divided each forward product into three parts: each part was a test set of a sub-dataset. Each test set was associated to one training set which includes the multiple forward products. Of course, observations of a training set occurred before the associated test set. Figure 8 shows how training and test sets were allocated for these sub-datasets. Data for the May-08 product was divided into three parts: each part is a test set of a sub-dataset. In sub-dataset 1, the test set is the first 1/3 of the observations of May-08 product, the training set includes data from Apr-08, Mar-08, Feb-08, and Jan-08 products.

## 5.2. Variable selection and pre-processing

In addition to electricity demand and monthly forward gas prices, there are a large number of exogenous variables which may be potential candidates for inputs. However, only some of them are relevant for prediction. Using irrelevant variables as input will reduce the performance of the forecasting models. Therefore, selecting correct inputs for each kind of model is very important.

The potential inputs include historical data of electricity demand ( $d$ ), electricity supply for the UK ( $s$ ), historical data of real average temperature in Celsius degrees ( $\tau$ ), wind speed, sunset time, system marginal prices (SMP) sell of gas ( $m$ ), gas demand ( $g$ ), price of monthly gas forward product ( $p$ ), forward price of monthly/seasonal/annual baseload/peak load electricity products, forward price of monthly/seasonal gas product, weekday ahead/weekend ahead gas price (denote the price of 1-winter ahead gas forward product by  $p^w$ ), system average price (SAP) of gas/electricity, exchange rate GBP:USD, oil spot price, and day pattern (i.e. day of the week). Weather forecast data was not available.

The potential inputs also include the WT components of the target variable. This paper uses 2-level WT. Denote WT components of electricity demand by  $A, D_2, D_1$ , and denote WT components of price of monthly gas forward product by  $A', D'_2, D'_1$ .

In the training phase, various measures were used to select the relevant input variables, including the correlation matrix (CM), autocorrelation function (ACF), partial autocorrelation function (PACF) and automatic relevance determination (ARD). The first three methods were used to select input variables for linear models (i.e. GARCH and LR). We computed the CM of the target and exogenous variables; and the exogenous variables which were highly correlated to the targets were chosen. We also computed the ACF and PACF of target time series. Lags with high correlations were selected as input variables as well.

Note that, because the day of the week is a periodic variable, we represented it by two dummy variables:  $swd = \sin(2\pi i/7)$  and  $cwd = \cos(2\pi i/7)$ , where  $i = 1$  to  $7$  corresponding to Monday to Sunday. There is another approach to deal with seasonality: using multiple-equation models with different equations for the different days. We have implemented both approaches, however, the results of first approach were better and are presented in the paper.

The temperature is known to have a non-linear relationship with electricity demand [30] [31]. Therefore, in linear models, instead of using historical real average temperatures ( $\tau$ ), we used a scaled value ( $\hat{\tau}$ ), which is linear with electricity demand plus noise. The methodology for computing  $\hat{\tau}$  has been mentioned in several previous papers [32], [33], [34]. We plotted a scatter plot of the electricity demand versus the average temperature in the training set; and we realised that if we divide data into four groups: working day with  $\tau < 14^0C$  or  $\tau \geq 14^0C$ , and weekend with  $\tau < 14^0C$  or  $\tau \geq 14^0C$ , the demand in each group is well modelled by a linear function of the average temperature plus noise. The scaled temperature  $\hat{\tau}$  is defined by these relationships and given by:

$$\hat{\tau} = \begin{cases} -18.89\tau + 1171.40 & \text{if working day and } \tau < 14^0C \\ 2.09\tau + 862.69 & \text{if working day and } \tau \geq 14^0C \\ -16.06\tau + 1009.86 & \text{if weekend and } \tau < 14^0C \\ -1.09\tau + 781.45 & \text{if weekend and } \tau \geq 14^0C \end{cases}$$

Figure 9 shows the ACF and PACF of the daily electricity demand time series  $d$  and its component  $A$ . From this figure, lags 1, 7 and 8 were chosen as inputs for LR and GARCH models for forecasting  $d$ . Lags 1, 2, 3, 4, 7, 8 and 9 were chosen as inputs for LR and GARCH models for predicting component  $A$ . Figure 10 shows absolute values of correlation matrix (AVCM) of target

variables and potential candidates for inputs. Because we are concerned only the correlations of first four attributes (i.e. targets) with the others, we did not plot the full AVCM, but only first four rows of the AVCM. A summary of the input variables for LR and GARCH models is shown in Table 1. Note the significant differences in lags selected for the different components of the multicomponent forecast model.

Input variables for non-linear models (i.e. MLP and RBF) were chosen using ARD [35], which is a Bayesian technique to evaluate the importance of each input variable. Each potential input variable is assumed to be sampled from a zero-mean Gaussian distribution: the inverse of the variance is called hyperparameter. The evidence procedure is used to optimise these hyperparameters during model training. If a hyperparameter is small, it is likely that its associated variable has large weight. This means that the associated variable is important and should be chosen as an input. Details of ARD and an implementation can be found in [37], chapter 9. In the electricity demand forecast using models without WT, hyperparameters range from 7.72 to 198.95. The variables associated with the smallest hyperparameters were chosen as inputs for MLP/RBF models: they are shown in Table 2.

The same procedures were implemented for the gas forward price dataset, see Table 3.

### 5.3. Model evaluation

There are 36 different models by combining the standard forecasting models with the RHWT and the filters. These models were tested on two large datasets (i.e. electricity demand and gas price); they are compared systematically on the same data.

#### 5.3.1. Benchmark models

Because electricity demand is strongly seasonal with a period of one week, the benchmark model for this dataset is a model in which demand of a day is assumed to be the same as the demand of the same day in the previous week.

A random walk (RW) model is used as a benchmark to evaluate the performance of forecasting monthly gas forward price. In many heavily traded markets for financial products (e.g. currency, stock market indices), the RW is



a very strong benchmark (of the efficient market hypothesis). A RW is given by:  $y_{t+1} = y_t + \varepsilon$ , where  $\varepsilon$  is zero-mean noise. The model predicts that tomorrow's price will be equal to today's price.

### 5.3.2. Errors

Three types of prediction errors on the test sets were computed. They are the mean absolute percent error (MAPE), normalised mean squared error (NMSE), and mean absolute error (MAE) which are defined by

$$\begin{aligned} e_{MAPE} &= \frac{1}{T} \sum_{t=1}^T \left| \frac{y_t - \hat{y}_t}{y_t} \right| \\ e_{NMSE} &= \frac{\sum_{t=1}^T (y_t - \hat{y}_t)^2}{\sum_{t=1}^T (y_t - E[y])^2} \\ e_{MAE} &= \frac{1}{T} \sum_{t=1}^T |y_t - \hat{y}_t|, \end{aligned}$$

where  $y$  is the real demand/price,  $\hat{y}$  is the forecast demand/price,  $E[y]$  is the mean of  $y$ , and  $T$  is the number of observations in the test set.

We also computed the improvement ratio (IR) of the errors compared with the corresponding errors of the benchmark model (BM). For example, the IR of NMSE of a model M comparing with NMSE of the BM is given by:

$$IR_{NMSE}(M) = \frac{e_{NMSE}(BM) - e_{NMSE}(M)}{e_{NMSE}(BM)}.$$

Variables were normalised to zero mean and unit variance. Forecast time series were converted to the original domain, before computing the errors.

### 5.3.3. Hypothesis testing

Assume that there are  $m$  forecasting models. We calculate  $m$  series  $e^1, \dots, e^m$ , each of them associated with one forecast model, where  $e^i$  represents the error of the  $i$ th model. In order to assess the relative performance of a large number of models on (potentially) a large number of test sets, the sum of P-values for each model  $i$  ( $SP^i$ ) was evaluated as follows:

$$SP^i = \frac{1}{m} \sum_{j=1}^m P_{ij},$$

where

$$P_{ij} = \begin{cases} \text{P-value of paired t-test } (e^i, e^j) \text{ with hypothesis } e^i > e^j & \text{when } i \neq j \\ 0.5 & \text{when } i = j \end{cases}$$

The smaller  $e^i$  is, the larger  $SP^i$ . Because  $e^i$  represents the error of the  $i$ th model, the smaller  $e^i$ , the better the  $i$ th model is. Therefore, we can rank performance of models based on  $SP^i$ .

In the electricity demand experiment, because there is only one test set (containing 296 observations), the series  $e^i$  associated with the  $i$ th model is the time series of normalised squared errors (NSE) of the observations:  $NSE^i = [NSE_1^i, \dots, NSE_{296}^i]$ , where

$$NSE_t^i = \frac{(y_t - \hat{y}_t^i)^2}{Cov(y)}, \quad t = 1, \dots, 296,$$

where  $\hat{y}_t^i$  is the forecast value of  $y$  at time  $t$  by the  $i$ th model.

In the gas price forecast, there are 72 sub-datasets. Therefore, instead of using  $NSE$  as in the electricity demand dataset, we used  $NMSE$ ; the series  $e^i$  associate with the  $i$ th model is:  $NMSE^i = [NMSE_1^i, \dots, NMSE_{72}^i]$ , where  $NMSE_k^i$  is the normalised mean squared error of the  $i$ th model for sub-dataset  $k$ ,  $k = 1, \dots, 72$ ,  $i = 1, \dots, m$ .

#### 5.4. Results on daily electricity demand forecast

The program was written in Matlab. The number of hidden units in MLP models for forecasting  $d$  (in raw MLP model),  $A$ ,  $D_2$ ,  $D_1$  (in multiforecast models), and  $d$  (in direct forecast model) were 9, 9, 8, 12, and 11 respectively. The numbers of hidden units in all RBF models (for  $d$ ,  $A$ ,  $D_2$ , and  $D_1$ ) are 50. Appendices A.1 and A.2 presented how to determine these parameters. We used MLPs with `tanh` activation functions.

Table 4 contains the  $IR_{NMSE}$  and errors of the prediction methods for daily electricity demand forecasting. The table shows that the proposed methods are significantly better than the benchmark. Prediction performance of the non-linear models (MLP/RBF) is much better than linear models (GARCH/LR): The  $IR_{MSE}$  of fixed MLP and RBF are 81% and 68% while that of fixed LR and GARCH are 56% and 59% only. Therefore, we will focus on the effects of the WT and the EKF/PF on MLP and RBF.

Table 4 shows that models with WT outperform the models without wavelet transform, which proves the usefulness of the WT. For example, the MSE of the fixed MLP (RBF) model is 0.02827 (0.04714) while that of the fixed MLP

(RBF) model combined with multicomponent forecast is 0.02293 (0.02855). The multicomponent forecast achieves better results than the direct forecast. The multicomponent forecast combined with fixed/adaptive MLP are the best with MSE of 0.02293 and 0.02314, their MSE improves more than 84% compared to the MSE of the benchmark model.

In this dataset, the performance of fixed and adaptive models was almost the same. The adaptive models did slightly better than fixed models when the WT was not used, and vice versa on MLP/RBF with multicomponent forecasting. The adaptive models on this dataset did not work as well as on the gas forward price (to be presented in the next section). The MLP models provided better prediction accuracy than the RBF, LR and GARCH models.

#### 5.5. Results on price of monthly gas forward products

Number of hidden units in MLP models for forecasting  $p$  (in raw MLP model),  $A'$ ,  $D'_2$ ,  $D'_1$  (in multiforecast models), and  $p$  (in direct forecast model) are 6, 6, 6, 5, and 6 respectively. The numbers of hidden units in all RBF models (for  $d$ ,  $A$ ,  $D_2$ , and  $D_1$ ) are 40. Appendices A.1 and A.2 presented how to determine these parameters. We used MLPs with  $\tanh$  activation functions.

The gas forward price dataset consists of 72 sub-datasets. The  $IR_{NMSE}$ , NMSE, MAPE, MAE, and SP were computed for each sub-dataset and for each fixed or adaptive prediction methods. Their averaged values are shown in Table 5. Because LR and GARCH are much better than MLP and RBF models ( $IR_{MSE}$  of fixed MLP model is only 0.94% and the results of the RBF model are worse than the benchmark model), in this paper, we will focus on the effects of WT and adaptive models on GARCH and LR models only.

Similar to results on the daily electricity demand dataset, multicomponent forecasts outperform the models without wavelet transform. However results of direct forecasts are worst than that of models without wavelet transform. The adaptive models are better than the fixed models. For example, the MSE improvement ratio for the fixed GARCH (LR) model was 2.26% (3.85%) while that of the adaptive GARCH (LR) model was 5.92% (5.33%). The adaptive GARCH models with multicomponent forecast achieved the best results with an MSE of 0.15384, which improves 15.10% compared to the MSE of the benchmark

model. The GARCH models generally provided better prediction accuracy than the LR, MLP, and RBF models.

In general, the adaptive models with PF are expected to provide better performance than the adaptive models with EKF because the PF does not require as many assumptions as in EKF. However, the results of these adaptive models were almost the same in these tests. This could be explained by the linearity of the SSMS. The state transition functions in equations (4) and (6) are linear. The outputs of GARCH and LR models were linear in parameters, thus so were their state space models. MLP and RBF models are non-linear functions, but we only adjusted on-line the bias of the second layer in MLP and the second layer weights in RBF which have a linear relationship with the outputs. Therefore the SSMS for adaptive MLP and RBF models were also linear. The local linearisation of output functions and space transition functions on EKF were perfect and the EKF could provide good updates.

## 6. Conclusions

The paper presents two frameworks for using WT as a pre-processing procedure for prediction applications. A range of machine learning and time series models are presented, such as GARCH, MLP, RBF, and LR. The results of electricity demand forecasting and gas price forecast on several large datasets show that the use of a WT improves the prediction performance. Multicomponent forecasting outperforms the methods without wavelet transform and the direct forecast.

This paper also showed how to combine each type of prediction model with filters (EKF/PF). It was shown experimentally that the adaptive models did improve prediction performance, especially on the gas forward price data (which is a non-stationary time series). However, this improvement is not as great as the improvement induced by using the WT. In these prediction models, noise is assumed to be Gaussian, therefore we can use both extended Kalman filter (EKF) and particle filter (PF) as filters. The adaptive models with PF and the adaptive models with EKF achieved similar results.

There are 36 different models by combining the standard forecasting models with the WT and the filters. These models were tested on two large datasets.

Beside the popular errors, we proposed a hypothesis testing and improvement ratio for comparison of different models.

In electricity demand forecasting, MLP and RBF models are generally better than GARCH and LR models while GARCH and LR are better than MLP and RBF in gas price forecast. In the electricity demand forecast, the multicomponent forecast combined with adaptive MLPs are the best with MSE 0.02314; its MSE improves more than 84% compared to the MSE of the benchmark model. In the gas price forecast, the adaptive GARCH models with multicomponent forecast achieves best results with MSE of 0.15384, which improve 15.1% comparing to MSE of the random walk model.

## 7. Acknowledgment

Hang T. Nguyen would like to thank to E.ON for their financial support and their permission to submit the paper for publication. The authors are grateful to Greg Payne, Matthew Cullen, David Turner, Stuart Griffiths, Nick Sillito, Daniel Crispin and David Jones from E.ON for providing the datasets, and their valuable advice. Some part of program have been written using source code from the NETLAB toolbox by Ian Nabney (available at <http://www.ncrg.aston.ac.uk/netlab/>) and Kalman filter code by Kevin Murphy (available at <http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html>). The authors also thank the anonymous referees for their insightful comments which have significantly improved the paper.

## Appendix A Fixed prediction models

### A.1 MLP

For an MLP with two layers,  $d$  input variables  $x = \{x(1), \dots, x(d)\}$ ,  $M$  hidden units, and  $c$  output units  $y = \{y(1), \dots, y(c)\}$ , the output can be calculated as follows

$$a_j^{(1)} = \sum_{i=1}^d \omega_{ij}^{(1)} x(i) + \omega_{j0}^{(1)}, \quad j = 1, \dots, M \quad (\text{A.1})$$

$$y(k) = \sum_{j=1}^M \omega_{kj}^{(2)} g(a_j^{(1)}) + \omega_{k0}^{(2)}, \quad k = 1, \dots, c, \quad (\text{A.2})$$

where  $\omega_{ji}^{(1)}$  and  $\omega_{kj}^{(2)}$  are the weights of the first and second layers respectively, and the activation function  $g(\cdot)$  is usually logistic sigmoidal or  $\tanh$ .

We optimise the model parameters by maximising likelihood and evidence procedure. More details on the theory and source code for training an MLP can be found in [37]. When training an MLP, we often encounter overfitting. Overfitting is a problem where the model fits the noise in the training data rather than the underlying generator and may lead to large errors on unseen data. There are several approaches to overcome this problem, such as *early stopping* [6] or using a committee to combine different networks. In this paper, we use *weight decay* to regularise the model by penalising large weights and imposing smoothness. The Bayesian evidence procedure is used to compute the optimal hyperparameters [38]. The number of hidden units is determined from the *number of well-determined* parameters found by the *evidence procedure*; see [39] section 10.4.

## A.2 RBF

The outputs  $y = \{y(1), \dots, y(c)\}$  of an RBF model for input  $x$  are given by:

$$r_j = \|x - \mu_j\| \quad j = 0, \dots, M \quad (\text{A.3})$$

$$y(k) = \sum_{j=0}^M \omega_{kj} \phi_j(r_j), \quad k = 1, \dots, c, \quad (\text{A.4})$$

where  $x$  represents the input of the RBF model,  $\mu_j$  are the cluster centres or first layer weights, and  $r_j$  is the distance between the input and the cluster centre  $\mu_j$ . Here  $\phi_j$  represents the basis functions (usually Gaussian) and  $\omega_{kj}$  is the second layer weight corresponding to the  $k^{th}$  output unit and the  $j^{th}$  basis function. More information about the theory and source code for training RBF models can be found in [37].

We used 10-fold cross-validation to select the number of basis functions of RBF. In a  $k$ -fold cross validation, the training set is divided into  $k$  nearly equally sized segments (or folds). We perform  $k$  iterations of training and validation. In each iteration, a single segment is used for validation and the remaining  $k - 1$  segments are used for training the model, so for each model there are  $k$  error values (we used NMSE here). The average of the errors is the cross-validation error of the model. This procedure is performed for the different RBF models

with different numbers of basis functions. Since the cross-validated error of a model on the training set maybe taken as an estimate for the error of the model on unseen data, the network structure corresponding to the smallest cross-validation error is chosen.

### A.3 LR

This model is given by:

$$y = \sum_{i=1}^d w(i)x(i) + b = w^T x + b, \quad (\text{A.5})$$

where  $y$  represents the output of target data,  $w = \{w(1), \dots, w(d)\}$  is the weight vector,  $b$  is bias and  $x = \{x(1), \dots, x(d)\}$  represents the input vector. We estimate parameters of LR by maximum likelihood. Details about theory and source code for training LR model can be found in [37].

### A.4 GARCH

The GARCH( $r, m$ ) model is given by:

$$\begin{aligned} y_t &= \tilde{\beta} + \hat{\beta}x_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, n_t) \\ n_t &= \alpha_0 + \sum_{i=1}^m \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^r \gamma_j n_{t-j}, \end{aligned}$$

with constraints

$$\alpha_i, \gamma_j > 0 \quad (\text{A.6})$$

$$\sum_{i=1}^m \alpha_i + \sum_{j=1}^r \gamma_j < 1, \quad (\text{A.7})$$

where  $x_t$ ,  $y_t$ , and  $\varepsilon_t$  represent the input vector, output vector, and error of the model respectively,  $n_t$  is variance of error  $\varepsilon_t$ ,  $\beta = \{\tilde{\beta}, \hat{\beta}\}$  is the parameter vector output function.

We can fit a GARCH model using maximum likelihood. The constraints in Equation (A.6) can be removed by substituting  $\alpha_i = \exp(\hat{\alpha}_i)$ ,  $\gamma_j = \exp(\hat{\gamma}_j)$  and we optimise with respect to  $\hat{\alpha}_i$ ,  $\hat{\gamma}_j$  instead of  $\alpha_i, \gamma_j$ . To satisfy the constraints in Equation (A.7), we used the penalty function method [37] to optimise the model.

## Appendix B Filters

### B.1 Extended Kalman filter

EKF is a recursive algorithm; one iteration of the EKF is composed of the following consecutive steps [40]:

- Prediction:

$$\begin{aligned} z_t^{t-1} &= f_t(z_{t-1}^{t-1}) \\ P_t^{t-1} &= F_t P_{t-1}^{t-1} F_t' + Q_t. \end{aligned}$$

- Update

$$\begin{aligned} K_t &= P_t^{t-1} H_t' [H_t P_t^{t-1} H_t' + R_t]^{-1} \\ z_t^t &= z_t^{t-1} + K_t [y_t - h_t(z_t^{t-1})] \\ P_t^t &= [I - K_t H_t] P_t^{t-1}, \end{aligned}$$

where  $F_t$  and  $H_t$  are the Jacobian matrices of the functions  $f_t(\cdot)$  and  $h_t(\cdot)$ ,  $F_t = \nabla f_t|z_t^t$  and  $H_t = \nabla h_t|z_t^{t-1}$ , and

$$\begin{aligned} z_t^r &= E[z_t|\{y\}_1^r] \\ P_t^r &= E[(z_t - z_t^r)(z_t - z_t^r)'|\{y\}_1^r]. \end{aligned}$$

### B.2 Particle filter

The PF is a sampling-based method. Firstly, we sample  $N_p$  times from initial distribution  $z_{0,i} \sim p(z_0) = \mathcal{N}(\pi_0, P_0)$ , allocating equal weights  $w_{0,i} = 1/N_p$  to each sample (or ‘‘particle’’). Next the state mean at  $t+1$  given  $y_{t+1}$  is estimated as follows:

- Evolve particles using transition function  $f$ :  $z_{t+1,i} = f_{t+1}(z_{t,i}) + \eta_i$ , where  $\eta_i$  is sampled from the appropriate noise distribution (here  $\eta_i \sim \mathcal{N}(0, Q_{t+1})$ ),  $i = 1, \dots, N_p$ .
- Re-weight particles when a new observation is available:  $w'_{t+1,i} \propto w_{t,i} \cdot p(y_{t+1}|z_{t+1,i})$ , where  $p(y_{t+1}|z_{t+1,i}) = \mathcal{N}(y_{t+1}|h_{t+1}(z_{t+1,i}), R_{t+1})$ ,  $i = 1, \dots, N_p$ .
- Normalise the weights:  $w_{t+1,i} = w'_{t+1,i} / \sum_{i=1}^{N_p} w'_{t+1,i}$ .



- The mean of state  $z$  at time step  $t + 1$  is the weighted average of particles:

$$E[z_{t+1}|y_{t+1}] = \sum_{i=1}^{N_p} w_{t+1,i} z_{t+1,i}.$$

In practice, after a large number of time steps, all but a small number of particles may have negligible weight. The problem with this degeneracy is that most of the particles contribute insignificantly to  $E[z_{t+1}|y_{t+1}]$ , but they still consume computational effort. We can reduce the effect of this problem by resampling [36].

## References

- [1] Nogales FJ, Contreras J, Conejo AJ, Espinola R. Forecasting next-day electricity prices by time series models. *IEEE Transactions on Power Systems* 2002; 17(2): 342-348.
- [2] Conejo AJ, Plazas MA, Espinola R, Molina AB. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *IEEE Transactions on Power Systems* 2005; 20(2): 1035-1042.
- [3] Abdel-Aal RE, Al-Garni AZ. Forecasting monthly electric energy consumption in eastern Saudi Arabia using univariate time-series analysis. *Energy* 1997; 22: 1059-1069.
- [4] Chavez SG, Bernat JX, Coalla HL. Forecasting of energy production and consumption in Asturias (northern Spain). *Energy* 1999; 24: 183-198.
- [5] Zheng H, Xie L, Zhang L. Electricity price forecasting based on GARCH model in deregulated market. In: *Power Engineering Conference*. 2005. p. 410-416.
- [6] Gao F, Guan ZH, Cao XR, Papalexopoulos A. Forecasting power market clearing price and quantity using a neural network method. In: *Proceedings of the Power Engineering Summer Meet*, vol. 4. Seattle, 2000. p.2183-2188.
- [7] Ekonomou L. Greek long-term energy consumption prediction using artificial neural networks. *Energy* 2010; 35(2): 512-517.

- [8] Amjady N, Keynia F. Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm. *Energy* 2009; 34(1): 46-57.
- [9] Guo JJ, Luh PB. Improving market clearing price prediction by using a committee machine of neural networks. *IEEE Transactions on Power Systems* 2004; 19(4): 1867-1876.
- [10] Zhang L, Luh PB, Kasiviswanathan K. Energy clearing price prediction and confidence interval estimation with cascaded neural networks. *IEEE Transactions on Power System* 2003; 1: 99-105.
- [11] Pao HT. Forecasting energy consumption in Taiwan using hybrid nonlinear models. *Energy* 2009; 34:1438-1446.
- [12] Saab S, Badr E, Nasr G. Univariate modeling and forecasting of energy consumption: the case of electricity in Lebanon. *Energy* 2001; 26: 1-14.
- [13] Tai N, Stenzel J, Wu H. Techniques of applying wavelet transform into combined model for short-term load forecasting. *Electric Power Systems Research* 2005; 76: 525-533.
- [14] Xu H, Niimura T. Short-term electricity price modeling and forecasting using wavelets and multivariable time series. In: *Power Systems Conference and Exposition*, vol. 1. 2004. p. 208- 212.
- [15] Niranjana M. On data driven of options prices using neural networks. In: *Forecasting Financial Markets: Advances for Exchange Rate, Interest Rate and Asset Management*. London, 1999. p. 1-13.
- [16] Nabney IT, McLachlan A, Lowe D. Practical method of tracking of non-stationary time series applied to real world data. In: *Aero Sense 96: Applications and Science of Artificial Neural Networks II*, SPIE Proceedings, vol. 2760. 1996. p. 152-163.
- [17] Patil SA, Irwin R, Srinivasan S, Prasad S, Lazarou G, Picone J. Sequential state-space filters for speech enhancement. In: *Proceedings of the IEEE SoutheastCon* 2006. p. 240-243.

- [18] Andreou AS, Georgopoulos EF, Likothanassis SD. Exchange-rates forecasting: a hybrid algorithm based on genetically optimized adaptive neural networks. *Computational Economics* 2002; 20: 191-210.
- [19] Li S, Wunsch DC, O'Hair E, Giesselmann MG. Wind turbine power estimation by neural network with Kalman filter training on SIMD parallel machine. In: *International Joint Conference on Neural Networks*, vol. 5. 1999. p. 3430-3434.
- [20] Zhang L, Luh PB. Power market clearing price prediction and confidence interval estimation with fast neural network learning. In: *Proceeding of the IEEE 2002 Power Engineering Society Winter Meeting*, vol. 1. 2002. p. 268-273.
- [21] Nguyen HT, Nabney IT. Combining the wavelet transform and forecasting models to predict gas forward prices. In: *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*. 2008. p. 311-317.
- [22] Zhang BL, Coggins R, Jabri A, Dersch D, Flower B. Multiresolution forecasting for futures trading using wavelet decompositions. *IEEE Transactions on Neural Networks* 2001; 12(4): 765-775.
- [23] Starck JL, Murtagh F. MR/Finance multiresolution analysis of time series. 2001. See also: <http://thames.cs.rhul.ac.uk/~multires/doc/mrfin.pdf>
- [24] Benaouda D, Murtagh F, Starck JL, Renaud O. Wavelet-based non linear multiscale decomposition model for electricity load forecasting. *Neurocomputing* 2006; 70: 139-154.
- [25] Dong ZY, Zhang BL, Huang Q. Adaptive neural network short-term load forecasting with wavelet decomposition. In: *IEEE Porto Power Tech Conference*, vol. 2. 2001.
- [26] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks* 1989; 2(5): 359-366.
- [27] Bollerslev T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 1986; 31:307-327.

- [28] Kalman RE. A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering 1960; 82 (Series D): 35-45.
- [29] Ghahramani Z, Hinton GE. Parameter estimation for linear dynamical system, report no. CRG-TR-96-2. Canada: University of Toronto, 1996.
- [30] Bessec M, Fouquau J. The non-linear link between electricity consumption and temperature in Europe: A threshold panel approach. Energy Economics 2008; 30(5):2705-2721.
- [31] Henley A, Peirson, J. Non-linearities in electricity demand and temperature: parametric versus non-parametric methods. Oxford Bulletin of Economics and Statistics 1997; 59(1): 149-62.
- [32] Engle RF, Granger CWJ, Rice J, Weiss A. Semiparametric estimates of the relation between weather and electricity sales. Journal of the American Statistical Association 1986; 81(394): 310-320.
- [33] Cancelo JR, Espasa A, Grafe R. Forecasting the electricity load from one day to one week ahead for the Spanish system operator. International Journal of Forecasting 2008; 24:588-602.
- [34] Moral-Carcedo J, Vic´ens-Otero J. Modelling the non-linear response of Spanish electricity demand to temperature variations. Energy Economics 2005; 27: 477-494.
- [35] MacKay D. Bayesian method for back prop network. In: Domany E, Hemmen JL, and Schulten K, editors. Models of Neural Networks III 1994. Springer. p. 211-254.
- [36] Arulampalam SM, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing 2002; 50(2): 174-188.
- [37] Nabney IT. NETLAB: Algorithms for Pattern Recognition. Great Britain: Springer; 2002.
- [38] Mackay D. Bayesian interpolation. Neural Computation 1992; 4: 415-447.

- [39] Bishop CM. Neural Networks for Pattern Recognition. Oxford University Press; 1995.
- [40] Ribeiro M. Kalman and extended Kalman filter: Concept, derivation and properties. Portugal: Institute for Systems and Robotics; 2004.

### Figure Captions

Figure 1: Values used to compute wavelet coefficients at time  $t$ . (a) Symmetric WTs. (b) asymmetric WT.

Figure 2: Computation of wavelet coefficients of different scales in the RHWT.

Figure 3: The multicomponent forecast method. (a) Training phase, (b) Test phase.

Figure 4: Direct forecast model.

Figure 5: The adaptive MLP/RBF/GARCH/LR. (a) Training phase. (b) Test phase.

Figure 6: Dataset 1: daily electricity demand. The training set is the earlier section and the test set is the later section.

Figure 7: Dataset 2: price of monthly forward gas products Jun-2006 to May-2008. Data is sampled from 1<sup>st</sup> December 2005 to 30<sup>th</sup> April 2008.

Figure 8: Training and test set allocation for gas price forecasting.

Figure 9: ACF and PACF of electricity demand  $d$  and component  $A$ . (a) PACF of  $d$ . (b) ACF of  $d$ . (c) PACF of component  $A$ . (d) ACF of component  $A$ .

Figure 10: AVCM for determining input variables for daily electricity demand forecast: darker colours represent higher values. The 51 attributes in the CM are target variables and potential candidates for inputs (i.e. lags 1 or 2 of the targets and exogenous variables). A list of these exogenous variables are shown in Section 5.2. The first four attributes, numbered by 1, 2, 3, 4, are the targets  $d_t$ ,  $A_t$ ,  $D_{2,t}$ , and  $D_{1,t}$  respectively. The variables which are highly correlated to the targets are  $d_{t-1}$ ,  $A_{t-1}$ ,  $D_{1,t-1}$ ,  $d_{t-2}$ ,  $A_{t-2}$ ,  $s_{t-1}$ ,  $s_{t-2}$ ,  $\hat{\tau}_{t-1}$ ,  $g_{t-1}$ ,  $swd_t$ , and  $cwd_t$  (numbered by 5, 6, 8, 9, 10, 13, 14, 16, 22, 23 and 24 respectively).

### **Table Captions**

Table 1: Input variables of GLM and LR models for daily electricity demand.

Table 2: Input variables of MLP and RBF models for daily electricity demand.

Table 3: Input variables for forecasting price of monthly forward gas products.

Table 4: Errors and NMSE improvement ratio of forecasting methods for the electricity demand dataset. “MLP+EKF” and “MLP+PF” referred to adaptive MLP models with EKF and PF respectively. Similar notation was used for RBF models. Notation “mf” and “df” refers to multicomponent forecast and direct forecast methodologies respectively.

Table 5: Average errors and NMSE improvement ratio of forecasting methods for the gas forward price dataset.

Methodologies	Target	Input variables
Without RHWT	$d_t$	$d_{t-1}, d_{t-7}, d_{t-8}, s_{t-1},$ $\hat{\tau}_{t-1}, cwd_t, g_{t-1}$
Multicomponent forecast	$A_t$	$A_{t-1}, A_{t-2}, A_{t-3}, A_{t-4},$ $A_{t-7}, A_{t-8}, A_{t-9}, d_{t-1}, d_{t-7},$ $s_{t-1}, \hat{\tau}_{t-1}, cwd_t, g_{t-1}$
	$D_{2,t}$	$D_{2,t-1}, D_{2,t-2}, D_{2,t-4}, D_{2,t-5},$ $D_{2,t-13}, D_{2,t-14}, D_{2,t-15}, D_{1,t-1}, cwd_t$
	$D_{1,t}$	$D_{1,t-5}, D_{1,t-7}, swd_t$
Direct forecast	$d_t$	$d_{t-1}, d_{t-7}, d_{t-8}, A_{t-1}, s_{t-1},$ $\hat{\tau}_{t-1}, cwd_t, g_{t-1}$

Table 1: Input variables of GLM and LR models for daily electricity demand.

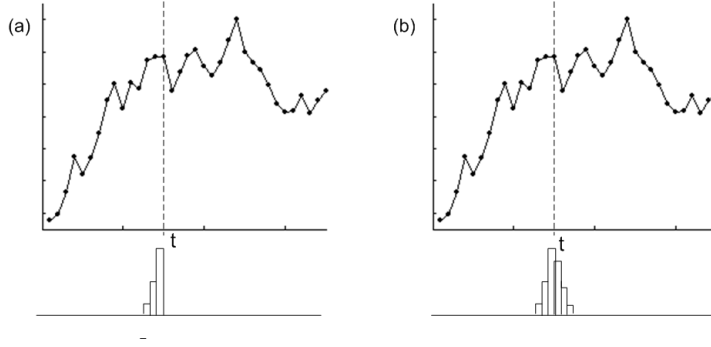


Figure 1: Values used to compute wavelet coefficients at time  $t$ . (a) Symmetric WTs. (b) asymmetric WT.



Methodologies	Target	Input variables
Without RHWT	$d_t$	$d_{t-1}, d_{t-6}, d_{t-7}, d_{t-8}, s_{t-1},$ $s_{t-2}, \tau_{t-1}, m_{t-2}, swd_t, cwd_t$
Multicomponent forecast	$A_t$	$A_{t-1}, A_{t-2}, A_{t-3}, A_{t-7}, A_{t-9},$ $d_{t-1}, d_{t-2}, s_{t-1}, s_{t-2},$ $\tau_{t-1}, m_{t-2}, swd_t, cwd_t$
	$D_{2,t}$	$D_{2,t-1}, D_{2,t-2}, D_{2,t-4},$ $D_{2,t-5}, D_{2,t-7}, d_{t-1}, d_{t-2},$ $A_{t-1}, s_{t-1}, swd_t, cwd_t$
	$D_{1,t}$	$D_{1,t-2}, D_{1,t-4}, D_{1,t-5},$ $D_{1,t-7}, d_{t-1}, swd_t$
Direct forecast	$d_t$	$d_{t-1}, d_{t-6}, d_{t-7}, d_{t-8}, A_{t-1},$ $A_{t-2}, D_{2,t-1}, s_{t-1}, s_{t-2},$ $\tau_{t-1}, m_{t-2}, swd_t, cwd_t$

Table 2: Input variables of MLP and RBF models for daily electricity demand.

Methodologies	Target	Input variables
Without RHWT	$p_t$	$p_{t-1}, p_{t-2}, p_{t-1}^w, p_{t-2}^w$
Multicomponent forecast	$A'_t$	$A'_{t-1}, A'_{t-2}, p_{t-1},$ $p_{t-2}, p_{t-1}^w, p_{t-2}^w$
	$D'_{2,t}$	$D'_{2,t-1}, D'_{1,t-1}, D'_{1,t-2}$
	$D'_{1,t}$	$D'_{1,t-1}, D'_{1,t-2}$
Direct forecast	$p_t$	$p_{t-1}, p_{t-2}, A'_{t-1},$ $A'_{t-2}, p_{t-1}^w, p_{t-2}^w$

Table 3: Input variables for forecasting price of monthly forward gas products.

<b>Models</b>	<b>IR(NMSE)</b>	<b>NMSE</b>	<b>MAPE</b>	<b>MAE</b>	<b>SP</b>
Benchmark	0.0%	0.14951	0.02998	29444	0.02381
Fixed GARCH	56.8%	0.06464	0.01922	18574	0.08798
Fixed LR	59.5%	0.06059	0.01896	18303	0.13882
Fixed MLP	81.1%	0.02827	0.01379	13042	0.58127
<b>Fixed MLP + mf</b>	<b>84.7%</b>	<b>0.02293</b>	<b>0.01258</b>	<b>12077</b>	<b>0.90427</b>
Fixed MLP +df	82.7%	0.02585	0.01321	12515	0.73692
MLP + EKF	81.7%	0.02732	0.01361	12899	0.65945
<b>MLP + EKF + mf</b>	<b>84.5%</b>	<b>0.02314</b>	<b>0.01260</b>	<b>12084</b>	<b>0.86689</b>
MLP + EKF + df	82.9%	0.02561	0.01317	12475	0.75151
MLP + PF	81.7%	0.02742	0.01360	12886	0.64803
<b>MLP + PF + mf</b>	<b>84.5%</b>	<b>0.02314</b>	<b>0.01259</b>	<b>12077</b>	<b>0.86458</b>
MLP + PF + df	82.9%	0.02564	0.01316	12463	0.74343
Fixed RBF	68.5%	0.04714	0.01570	15006	0.16913
Fixed RBF + mf	80.9%	0.02855	0.01395	13443	0.59129
Fixed RBF +df	75.1%	0.03726	0.01554	14771	0.33580
RBF + EKF	70.6%	0.04393	0.01575	15035	0.24737
RBF + EKF + mf	80.8%	0.02873	0.01409	13546	0.56889
RBF + EKF + df	75.5%	0.03657	0.01543	14676	0.38720
RBF + PF	70.2%	0.04449	0.01573	15014	0.23807
RBF + PF + mf	80.8%	0.02871	0.01408	13538	0.57966
RBF + PF + df	75.5%	0.03664	0.01544	14681	0.37561

Table 4: Errors and NMSE improvement ratio of forecasting methods for the electricity demand dataset. "MLP+EKF" and "MLP+PF" referred to adaptive MLP models with EKF and PF respectively. Similar notation was used for RBF models. Notation "mf" and "df" refers to multicomponent forecast and direct forecast methodologies respectively.

<b>Models</b>	<b>IR(NMSE)</b>	<b>NMSE</b>	<b>MAPE</b>	<b>MAE</b>	<b>SP</b>
Benchmark	0.00%	0.17566	0.01823	2.066	0.26457
LR	3.85%	0.17247	0.01855	2.061	0.39394
LR+mf	13.32%	0.15758	0.01739	2.029	0.75999
LR+df	2.14%	0.17949	0.01887	2.088	0.06821
LR+EKF	5.07%	0.16907	0.01818	2.049	0.49429
LR+EKF+mf	13.97%	0.15623	0.01710	2.025	0.79884
LR+EKF+df	3.44%	0.17595	0.01849	2.078	0.17573
LR+PF	5.33%	0.16316	0.01784	2.024	0.54686
LR+PF+mf	14.34%	0.15009	0.01672	2.007	0.86170
LR+PF+df	3.80%	0.16954	0.01814	2.055	0.24447
GARCH	2.26%	0.17490	0.01858	2.013	0.27014
GARCH+mf	12.96%	0.15837	0.01734	2.046	0.70791
GARCH+df	1.51%	0.17883	0.01875	2.124	0.09964
GARCH+EKF	5.92%	0.16782	0.01812	2.049	0.64699
<b>GARCH+EKF+mf</b>	<b>15.10%</b>	<b>0.15384</b>	<b>0.01699</b>	<b>2.019</b>	<b>0.93351</b>
GARCH+EKF+df	4.25%	0.17478	0.01845	2.073	0.31210
GARCH+PF	5.91%	0.16783	0.01812	2.049	0.61759
<b>GARCH+PF+mf</b>	<b>15.10%</b>	<b>0.15384</b>	<b>0.01699</b>	<b>2.017</b>	<b>0.95753</b>
GARCH+PF+df	4.25%	0.17476	0.01845	2.073	0.34599

Table 5: Average errors and NMSE improvement ratio of forecasting methods for the gas forward price dataset.

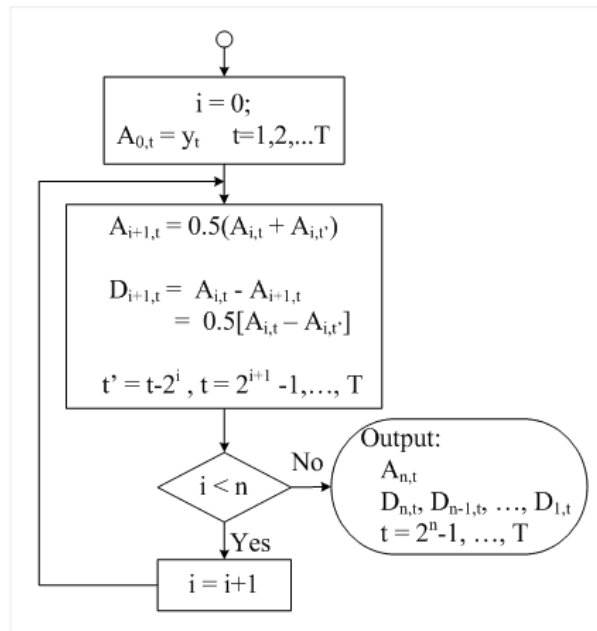


Figure 2: Computation of wavelet coefficients of different scales in the RHWT.

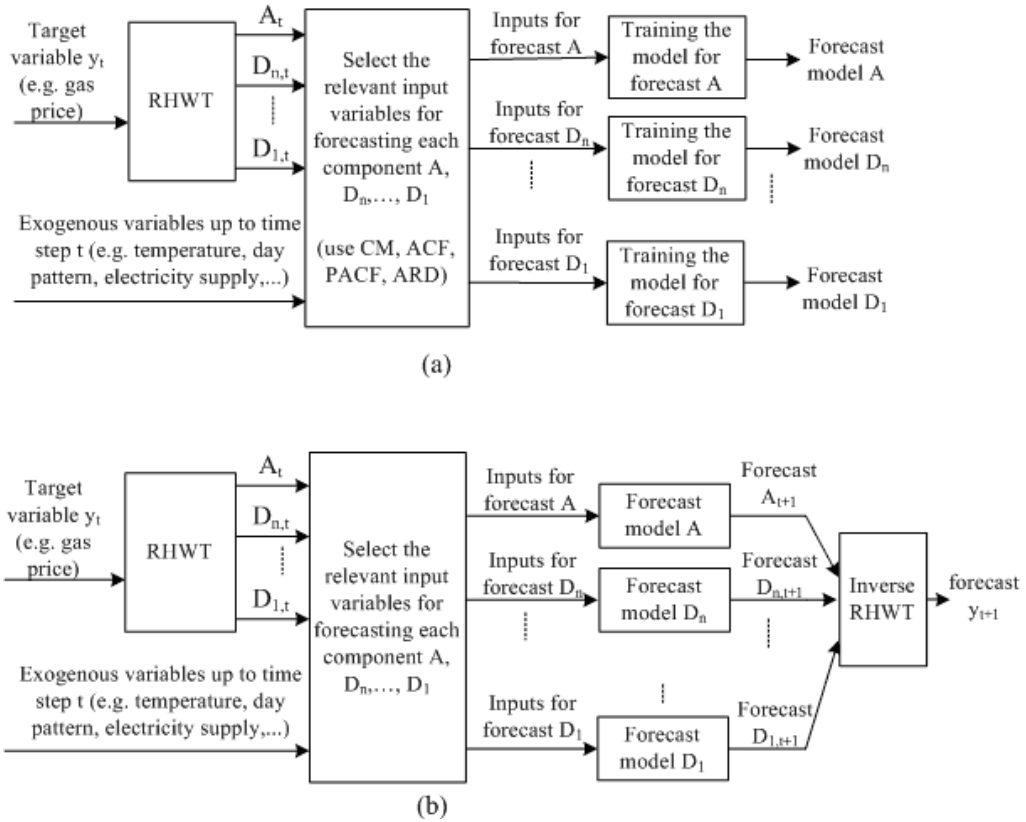


Figure 3: The multicomponent forecast method. (a) Training phase, (b) Test phase.

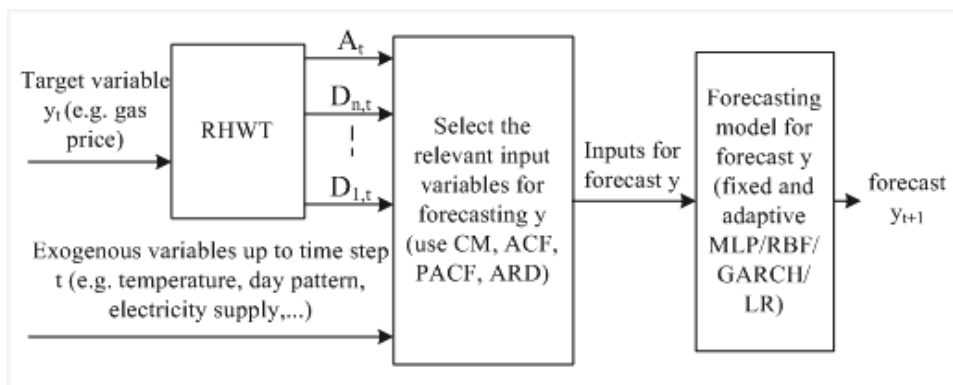


Figure 4: Direct forecast model.

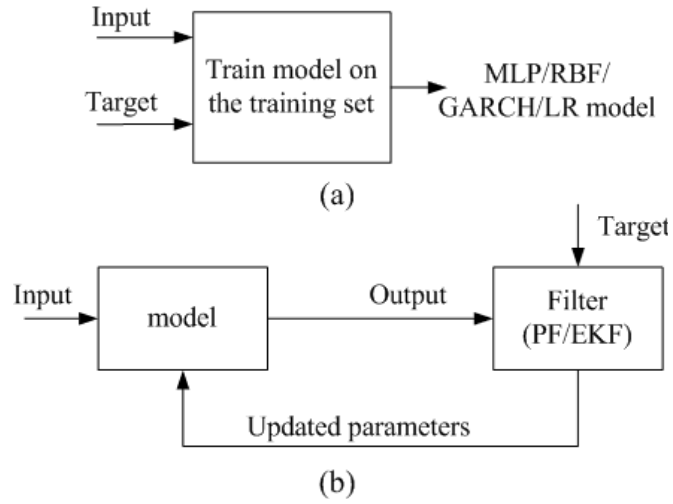


Figure 5: The adaptive MLP/RBF/GARCH/LR. (a) Training phase. (b) Test phase.

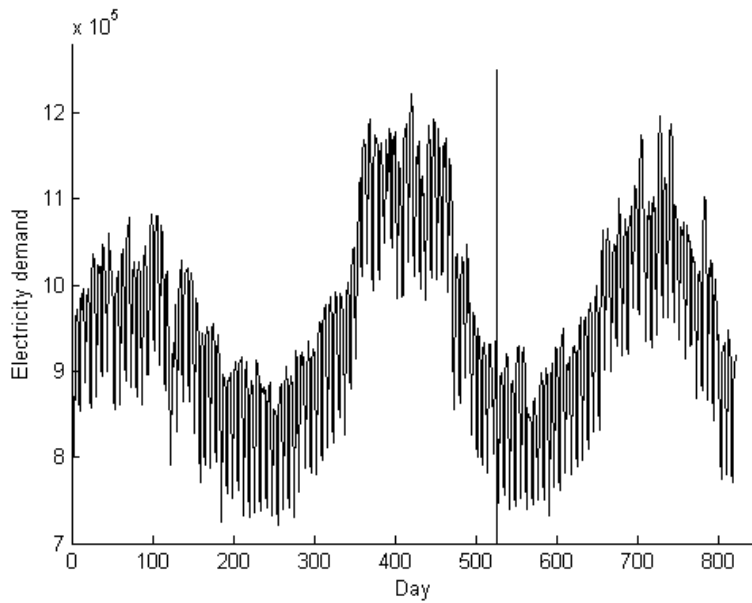


Figure 6: Dataset 1: daily electricity demand. The training set is the earlier section and the test set is the later section.

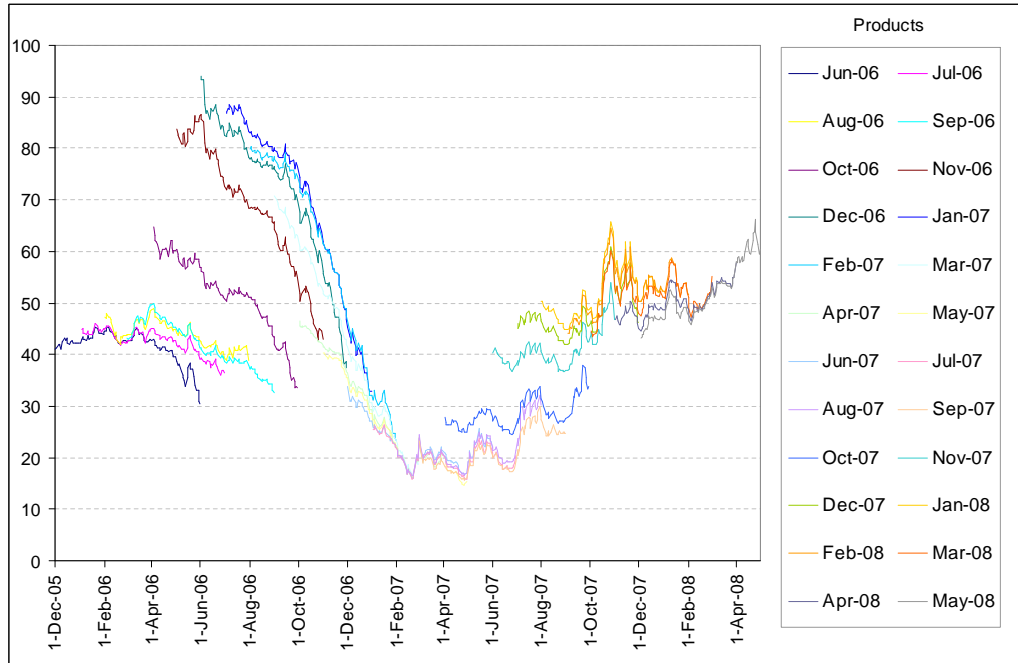


Figure 7: Dataset 2: price of monthly forward gas products Jun-2006 to May-2008. Data is sampled from 1<sup>st</sup> December 2005 to 30<sup>th</sup> April 2008.

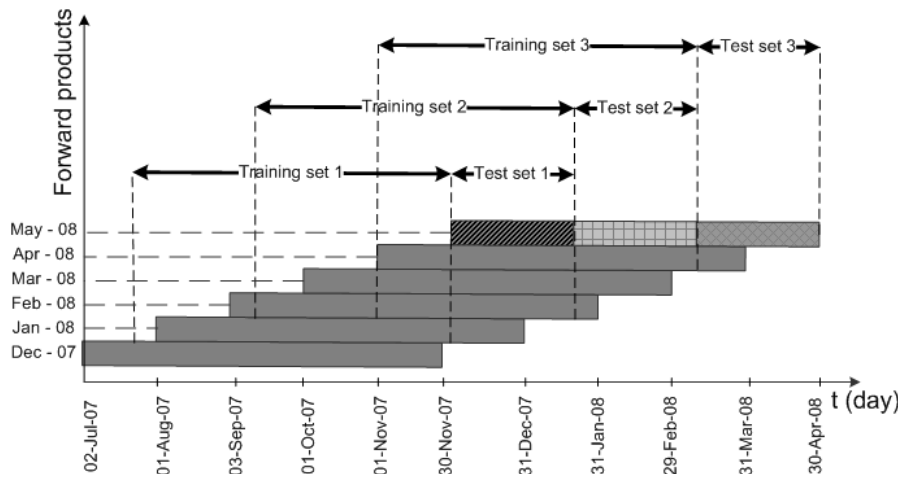


Figure 8: Training and test set allocation for gas price forecasting.

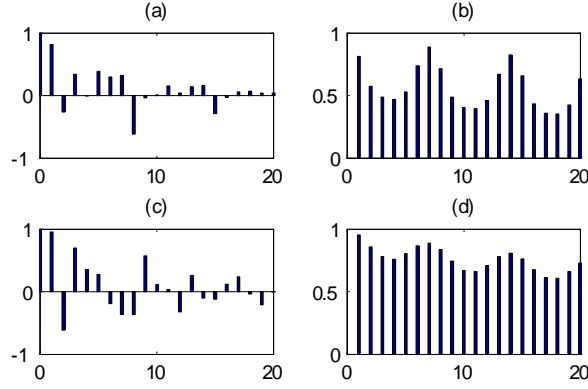


Figure 9: ACF and PACF of electricity demand  $d$  and component  $A$ . (a) PACF of  $d$ . (b) ACF of  $d$ . (c) PACF of component  $A$ . (d) ACF of component  $A$ .

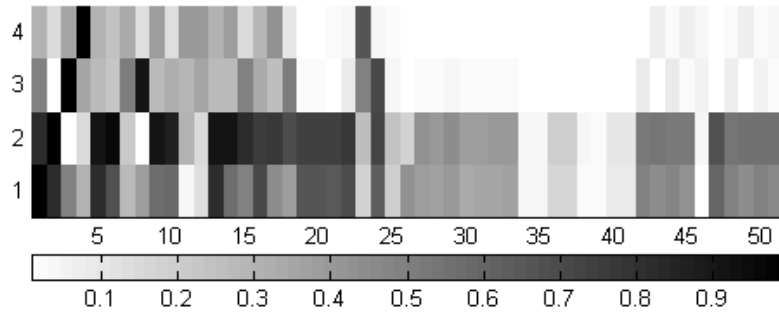


Figure 10: AVCM for determining input variables for daily electricity demand forecast: darker colours represent higher values. The 51 attributes in the CM are target variables and potential candidates for inputs (i.e. lags 1 or 2 of the targets and exogenous variables). A list of these exogenous variables are shown in Section 5.2. The first four attributes, numbered by 1, 2, 3, 4, are the targets  $d_t$ ,  $A_t$ ,  $D_{2,t}$ , and  $D_{1,t}$  respectively. The variables which are highly correlated to the targets are  $d_{t-1}$ ,  $A_{t-1}$ ,  $D_{1,t-1}$ ,  $d_{t-2}$ ,  $A_{t-2}$ ,  $s_{t-1}$ ,  $s_{t-2}$ ,  $\hat{\tau}_{t-1}$ ,  $g_{t-1}$ ,  $swd_t$ , and  $cwd_t$  (numbered by 5, 6, 8, 9, 10, 13, 14, 16, 22, 23 and 24 respectively).