# VERY SHORT-TERM LOAD FORECASTING USING EXPONENTIAL SMOOTHING AND ARIMA MODELS

Alexandra KOTILLOVÁ

University of Žilina, Faculty of Management Science and Informatics, Slovak Republic

e-mail: kotillova@gmail.com

**Abstract**

*Autoregressive integrated moving average (ARIMA) model and exponential smoothing are one of the most popular linear models in time series forecasting during the past three decades. This paper uses 30-minutes Australian electricity demand observations to evaluate methods for prediction 30 minutes ahead. For comparison was designed "industry" model.*

***Keywords***: *very short-term electricity load forecasting, prediction, autocorrelation analysis, ARIMA model, Box-Jenkins methodology, exponential smoothing, back propagation neural network*

## 1 INTRODUCTION

To forecast electricity load data have significance for two main reasons. First: the market operator uses load forecasting to determine how much electricity is required in the upcoming 30 minutes and accepts the best bids to meet the demand while minimizing the cost. Second: with a good forecast government can save money. Hobbs in his paper say, that 1 % of MAPE (Mean Absolute Percentage Error) for a 10 000 MW generator saves up to US$ 1.6million a year [1] .

## 2 DATA AND FEATURE SET

I work with public 30 minutes electricity load data for the state New South Wales [2] , what is an example of very short-term load forecasting. Data contains 30 minutes records about the demand and it is 48 values per day, 336 per week, 1344 per every four weeks, etc… For one year it is 17 520 values. I abstract only three months from the dataset. I choose June, July and August 2010. In next three pictures are yearly data for year 2010, monthly data for August 2010 and daily data for 1-$^{th}$ of August. In this work I am using monthly data for June and July 2010 for training and monthly data for August 2010 for testing. In Figure 1 is yearly electricity data for year 2010.

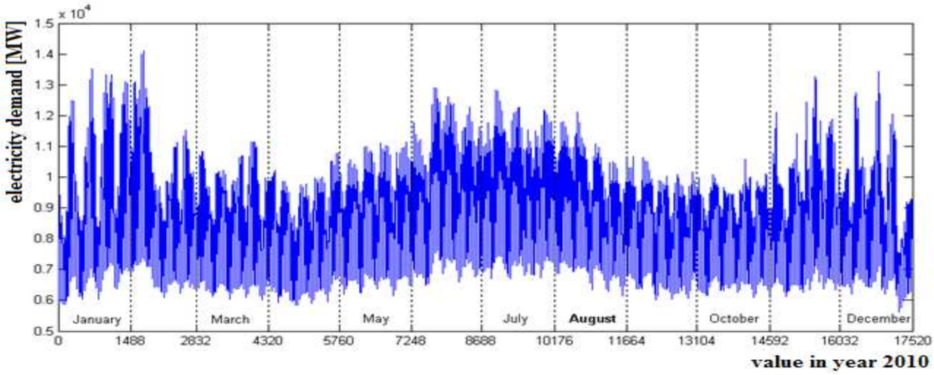


**Figure 1** Yearly data for Year 2010

Figure 2 shows a typical 7 days load pattern. During the works days (from Monday to Friday) it is higher daily pattern as during the weekend.

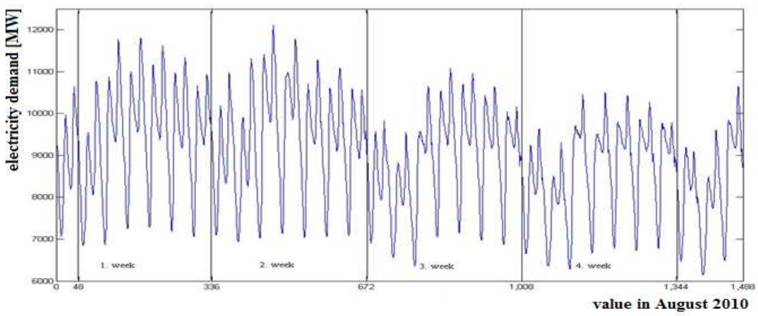


**Figure 2** Monthly data for August 2010

On the third picture reader can see how it looks like in one day. Local minimum is at 4:30 a.m., and then at 9:30 a.m. is the first maximum and the second maximum is in the evening at 6:30 p.m.

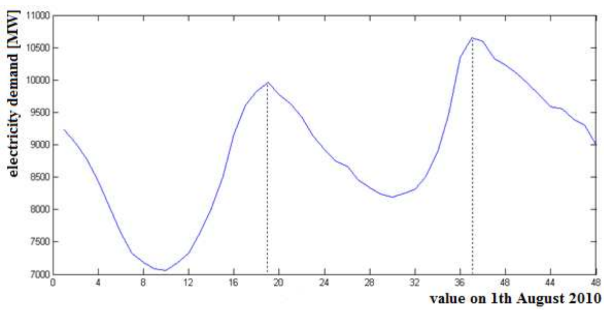


**Figure 3** Daily data for 1th of August 2010

As a model was used autocorrelation analysis. First was analyzed the cyclic nature of the dataset then was abstracted 34 helpful features. Second was tested all features on correlation.

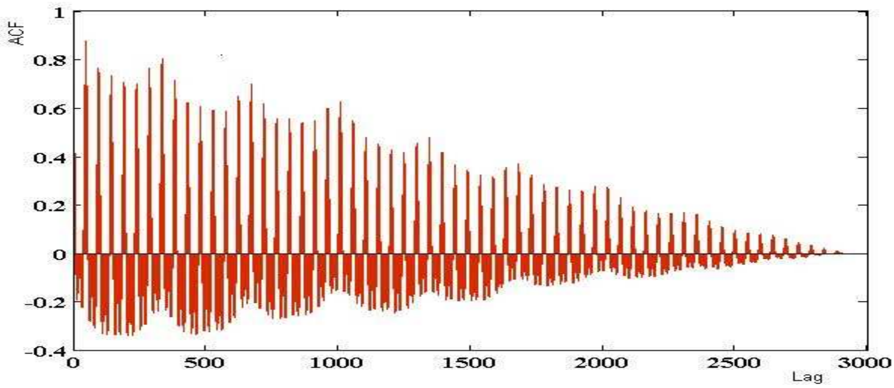


**Figure 4** ACF function for training set

I extract the load variables at the highest 5 points and the 3 lags around them: 3 before and 3 after them. The model is in simple regression form:

$$\hat{y}_{t+1} = b_0 + b_1 X_t + b_2 X_{t-1} + \ldots + b_6 X_{t-5} + b_7 XD_{t-3} + \ldots + b_{13} XD_{t+3} + b_{14} XDD_{t-3} + \ldots + b_{20} XDD_{t-3} + $$
$$+ b_{21} XD6_{t-3} + \ldots + b_{27} XD6_{t+3} + b_{28} XW_{t-3} + \ldots + b_{34} XW_{t-3}$$

Used variables are described in table 1.

**Table 1** cluster of high spikes and selected variables

| | Lag peak | Variables | Description |
|---|---|---|---|
| 1 | 1<br>same day | 6 variables<br>$X_t, \ldots, X_{t-5}$ | forecast day at times:<br>t,…, t-5 |
| 2 | 48<br>1 day | 7 variables:<br>$XD_{t-3}, \ldots, XD_{t+3}$ | 1 day before at times:<br>t-3,…, t+3 |
| 3 | 96<br>2 days | 7 variables:<br>$XDD_{t-3}, \ldots, XDD_{t+3}$ | 2 days before at times:<br>t-3,…, t+3 |
| 4 | 288<br>6 days | 7 variables:<br>$XD6_{t-3}, \ldots, XD6_{t+3}$ | 6 days before at times:<br>t-3,…, t+3 |
| 5 | 336<br>1 week | 7 variables:<br>$XW_{t-3}, \ldots, XW_{t+3}$ | 1 week before at times:<br>t-3,…, t+3 |

## 3 METHODOLOGY

In work was applied Exponential smoothing and seasonal ARIMA model. Both of them are standard linear regression algorithms. For computation was used R[3] , Weka library[4] and Matlab[5] . As accuracy metric was used MAPE.

### 3.1 Exponential smoothing

Exponential smoothing is probably the most used forrecast method. We know a simple, double, or triple exponencial smoothing. For seasonal time series is usefull seasonal Holt-Winters model [6] . It is an exponencial smoothing for data with a trend and seasonal behaviour. The additive seasonality version of the method is presented in expressions (1)-(4).

$$L_s = \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{s-1} + T_t) \quad (1)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (2)$$

$$S_t = \gamma(y_t - L_t) + (1-\gamma)S_{t-s} \quad (3)$$

$$\hat{y}_{t+k} = L_t + kT_t + S_{t+k-s} \quad (4)$$

Where $L_s$ is smoothed level of series, $T_s$ is smoothed additive trend at the end of period $t$. $S_i$ is smoothed seasonal index at the end of period t. $\hat{y}_{t+k}$ is forecast for k steps ahead. The multiplicative seasonality version of the method is presented in expressions (5)-(8).

$$L_s = \alpha(y_t / S_{t-s}) + (1-\alpha)(L_{s-1} + T_t) \quad (5)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (6)$$

$$S_t = \gamma(y_t / L_t) + (1-\gamma)S_{t-s} \quad (7)$$

$$\hat{y}_{t+k} = (L_t + kT_t)S_{t+k-s} \quad (8)$$

Here all variables are the same, but $T_s$ is in this case smoothed multiplicative trend at the end of period *t*.

$\alpha, \beta, \gamma$ are the parameters. $\alpha$ is the smoothing parameter for the level of the series. $\beta$ is smoothing parameter for the trend and $\gamma$ is smoothing parameter for seasonal index. Each of them is from interval <0,1>. Optimal values of parameters alpha, beta and gamma were computed with softvare R. The table 2 shows optimal values of parameters for every model. Intraday seasonality means that seasonal index is 48 and intraweek seasonal index is 336.

**Table 2** looking for optimal model

| Model | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|
| Intraday seasonality | | | |
| HW - additive | 0.9853545 | 0.1091507 | 1 |
| HW – mult. | 0.9865522 | 0.1261081 | 1 |
| Intraweek seasonality | | | |
| HW - additive | 0.9249508 | 0.1110802 | 1 |
| HW – mult. | 0.9280155 | 0.1338507 | 1 |

In next table 3 we can see the comparison of results for one-step-ahead forecast. The best results give additive model with seasonal index 48, i.e. one day.

**Table 3** accuracy of models

| Model | MAE | MAPE |
|---|---|---|
| Intraday seasonality | | |
| HW - additive | 32.3229 | 0.3663 |
| HW – mult. | 32.8437 | 0.3710 |
| Intraweek seasonality | | |
| HW - additive | 34.1124 | 0.3851 |
| HW – mult. | 46.0109 | 0.5224 |

### 3.2 Seasonal ARIMA

Autoregressive integrated moving average models was described by Box and Jenkins in year 1970 [7] The Box-Jenkins approach to modeling time series consists of four phases: identification, estimation and testing, diagnostic checking and forecasting. The multiplicative seasonal ARIMA model with just one seasonal pattern can be written as equation (9).

$$\Phi_p(B)\Lambda_P\left(B^s\right)\Delta^d\nabla_s{}^D y_t = \Theta_q(B)\Gamma_Q\left(B^s\right)\varepsilon_t \tag{9}$$

where B is the lag operator, *s* is the number of periods in a seasonal cycle, $\Delta$ is the difference operator , $\nabla_s$ is the seasonal difference operator, d and D are the orders of differencing, $\varepsilon_t$ is a white noise error term and $\Phi_p$, $\Lambda_p$, $\Theta_q$ and $\Gamma_q$ are polynomial functions of order *p, P, q, Q*.

This model is generally referred as an ARIMA(*p*,*d*,*q*)x(P,D,Q) model where *p, d*, and *q* are non-negative integers that refers to the order of the autoregressive, integrated, and moving average parts of the model. Where *p* is the order of the autoregressive part of the model, *d* is the order of differencing done to the data to make it stationary and *q* is the order of the moving average part of the model. *P* is order of seasonal AR proces, *Q* is order of seasonal MA proces, *D* is order of seasonal difference, s is length of seasonal period. In real situation *P, D, Q* is equal or less like 1.

At first it is necessarily to identify the type of the model and the values of the parameters. In the case of solving more models we choose the model where AIC (Akai information criteria), respectively SBC (Schwartz-Bayes criteria) are minimal and Log likelihood is maximal. At the end we verify if the residual component is white noise. From the table 4 can see that the best results of errors gives us ARIMA(6,1,1)$(1,1,1)_{48}$ model (on Table 4). MAPE for this model is 1,468%.

Table 4 **looking for optimal model**

| Model | Log(L) | AIC | MSE |
|---|---|---|---|
| ARIMA(1,1,1)(1,1,1)$_{48}$ | -8375.92 | 16761.83 | 6032 |
| ARIMA(2,1,1)(1,1,1)$_{48}$ | -8370.96 | 16753.92 | 5993 |
| ARIMA(3,1,1)(1,1,1)$_{48}$ | -16640.95 | 33295.89 | 5924 |
| ARIMA(4,1,1)(1,1,1)$_{48}$ | -16634.75 | 33285.49 | 5902 |
| ARIMA(5,1,1)(1,1,1)$_{48}$ | -16634.37 | 33286.75 | 5900 |
| ARIMA(6,1,1)(1,1,1)$_{48}$ | -16634.05 | 33288.11 | 5899 |

### 3.3 Industry model

On the webpage of the market operator we can find a model which company can use for forecasting. They used data from the previous 5 lags on that day and also the load 5 lags at exactly the same time one week ago. The industry features set is make like natural logarithm from differences. On this features set is typically employed back-propagation neural network. We can see the model on table 5. For the purposes of this work we will call this model "industry model".

**Table 5** industry feature sets

| variables | used features | description |
|---|---|---|
| 5 variables $X_t,\ldots,X_{t-4}$ | | forecast the same day: |
| | $ln\,\Delta(X_t)$ | $ln\,\Delta(X_t) = ln(X_t) - ln(X_{t-1})$ |
| | $ln\,\Delta(X_{t-1})$ | $ln\,\Delta(X_{t-1}) = ln(X_{t-1}) - ln(X_{t-2})$ |
| | $ln\,\Delta(X_{t-2})$ | $ln\,\Delta(X_{t-2}) = ln(X_{t-2}) - ln(X_{t-3})$ |
| | $ln\,\Delta(X_{t-3})$ | $ln\,\Delta(X_{t-3}) = ln(X_{t-3}) - ln(X_{t-4})$ |
| 6 variables $XW_{t+1},\ldots,XW_{t-4}$ | | forecast the same day 1 week before: |
| | $ln\,\Delta(XW_{t+1})$ | $ln\,\Delta(XW_{t+1}) = ln(XW_{t+1}) - ln(XW_t)$ |
| | $ln\,\Delta(XW_t)$ | $ln\,\Delta(XW_t) = ln(XW_t) - ln(XW_{t-1})$ |
| | $ln\,\Delta(XW_{t-1})$ | $ln\,\Delta(XW_{t-1}) = ln(XW_{t-1}) - ln(XW_{t-2})$ |
| | $ln\,\Delta(XW_{t-2})$ | $ln\,\Delta(XW_{t-2}) = ln(XW_{t-2}) - ln(XW_{t-3})$ |
| | $ln\,\Delta(XW_{t-3})$ | $ln\,\Delta(XW_{t-3}) = ln(XW_{t-3}) - ln(XW_{t-4})$ |

From 5 and 6 variables we received 4 and 5 features set and new regression model was constructed as:

$$\hat{y}_{t+1} = b_0 + b_1 ln\,\Delta(X_t) + b_2 ln\,\Delta(X_{t-1}) + b_3 ln\,\Delta(X_{t-2}) + b_4 ln\,\Delta(X_{t-3}) + b_5 ln\,\Delta(XW_{t+1}) + \\ + b_6\ ln\,\Delta(XW_t) + b_7\ ln\,\Delta(XW_{t-1}) + b_8\ ln\,\Delta(XW_{t-2}) + b_9\ ln\,\Delta(XW_{t-3})$$

The application of back-propagation neural network is generalized delta rule thus involves two phases: During the first phase is the input presented and propagated forward through the network to compute the output values for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each hidden unit in the network and appropriate weight changes are calculated. It was used a supervised learning in which mentioned weights were calculated using back-propagation algorithm [8] . More information about algorithm and algorithm itself you can find in [9] . The value of MAPE from this model is equal 1,421%.

## 4 CONCLUSION

It was evaluated the predictive power of 2 nested feature sets constructed based on autocorrelation analysis. The last figure 5 presents the MAPEs results of each algorithm.

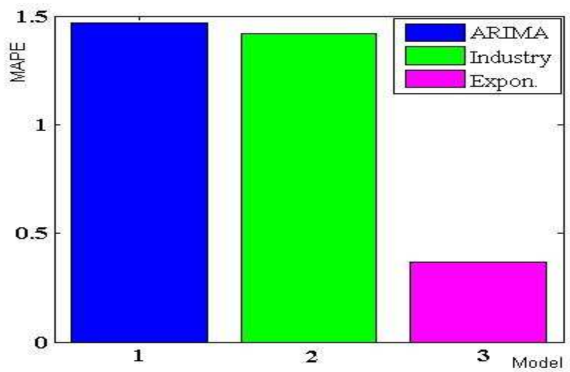


**Figure 5** MAPE for one step ahead comparison between the models

Exponential smoothing gives better result as industry model, but in some paper [10] [11] ARIMA methodology is combined with back-propagation neural network and this hybrid model can reach very good results. Future plans involve implementation of neural network on the data. Then they will be compared and confronted with solutions from industry model. Better approach will be chosen and new algorithm or improvements in existing algorithms will be done.

**Acknowledgement:** This work has been supported by the grant VEGA 1/0667/10.

## REFERENCES

[1] HOBBS, B.F., JITPRAPAIKULSARN, S., et al: Analysis of the Value for Unite Commitment of Improved Load Forecasts, IEEE Transactions on Power Systems, vol.14, no. 4, pp. 1342-1348, 1999.

[2] AEMO (2011), available: http://www.aemo.com.au/

[3] R Development Core Team: R: A language and environment for statistical computing and R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org, 2008.

[4] HALL, M., FRANK, E., HOLMES, G., et al: The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, 2009.

[5] Matlab: software was developed using MATLAB (2010a, The MathWorks), available: http://www.mathworks.com/products

[6] TAYLOR, J. W.: Short-term electricity demand forecasting using double seasonal exponencial smoothing, Journal of the Operational Research Society, vol. 54, pp. 799-805, 2003.

[7] MAKRIDAKIS S., WHEELWRIGHT S., HYNDMAN R.: Forecasting-methods and applications, 3rd edition, pp. 312-373, 1998.

[8] ZHANG, G.P. An investigation of neural networks for linear time-series forecasting. In Computer and Operations Research. Vol. 28. No. 12. Elsevier. October 2001. pp. 1183-1202.

[9] MARČEK, D., MARČEK, M.: Neurónové siete a ich aplikácie. Edis - University of Zilina. 2006. ISBN: 9788080704971.

[10] RINGWOOD, J.V., BOFFELI, D., MURRAY, F.T.: Forecasting electricity demand on short, medium and long time scales using neural networks, Journal of Intelligent and Robotic Systems, vol. 31, pp.129-147, 2001.

[11] ZHANG, G.P.: Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, vol. 51, pp. 159-175, 2003.