

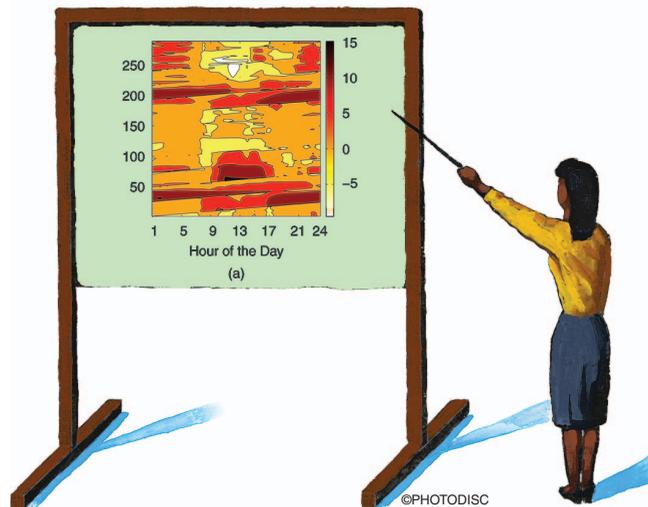
Matteo De Felice
ENEA, ITALY and
Xin Yao
University of Birmingham, UK

Short-Term Load Forecasting with Neural Network Ensembles: A Comparative Study

I. Introduction

Load Forecasting plays a critical role in the management, scheduling and dispatching operations in power systems, and it concerns the prediction of energy demand in different time spans. In future electric grids, to achieve a greater control and flexibility than in actual electric grids, a reliable forecasting of load demand could help to avoid dispatch problems given by unexpected loads, and give vital information to make decisions on energy generation and purchase, especially market-based dynamic pricing strategies. Furthermore, accurate prediction would have a significant impact on operation management, e.g. preventing overloading and allowing an efficient energy storage. In fact, in an environment where the fluctuations may depend from a big number of factors (and where some of them are unknown or hardly predictable), forecasting the demand using all the information provided by metering and sensing technologies is vital in order to have an effective management of peak demands (load shifting).

The ability to predict future behavior and energy demand is part of the intelligence needed by Smart Grids, where information technology is



strongly applied, and, more in general, future distribution networks (see EU ADDRESS project [1]). An intensive use of Distributed Generation raises new challenges, such as the need of a ‘distributed intelligence’ in order to deal with data originated in diverse places and to perform effective choices in a dynamic environment. An example of this new scenario is presented in Vale et al. [2] where various optimization heuristics are applied to economic dispatch problem in Smart Grids.

An investigation in such a complex scenario might start in small-scale, focusing on buildings, studying the performances of well-established forecast-

ing techniques and allowing an extension of the results to larger and more complex scenarios. In this work, we present an application of linear and non-linear models to the problem of Short-Term Load Forecasting (STLF) using real data from seven office buildings.

Most forecasting methods use statistical approaches or artificial intelligence algorithms. The most applied methods are Box-Jenkins approaches, exponential and Holt-Winters methods, Neural Networks (NN) based methods and, more recently, Support Vector Machines [3]. An introduction to time series applications can be found in Brockwell and Davis [4] and a good survey on various methodologies for load forecasting is in Feinberg and Genethliou [5].

Neural Networks have been applied successfully to a wide variety of real-world applications, as well to forecasting in several engineering fields. Neural networks demonstrated their ability to extract and learn the relationships between observed variables, leading to excellent results also in STLF [6].

This paper analyses their applications to this particular short-term load forecasting problem, showing the effectiveness of combining a high number of networks together in an ensemble. The term ‘ensemble’ describes a group of learning machines that work together on the same task, in the case of neural networks they

A preliminary version of this paper appeared in: M. De Felice, X. Yao, “Neural Networks Ensembles for Short-Term Load Forecasting,” presented at 2011 IEEE Symposium on Computational Intelligence Applications in Smart Grid, Paris, France, 11-14 April, 2011.

are trained on the same data, run together and their outputs are combined as a single one [7]. In this work, we apply this methodology to the forecasting with the aim of obtaining a reliable prediction.

This paper is organized as following: in Section II we define the STLF problem discussing its relationship with Smart Grids and then we describe the real-data used in this paper. A description of forecasting model is given in Sections III and IV. Section V briefly introduces neural network ensembles. Experimentations are described in Section VI and the results presented are discussed in Section VII. Finally, Section VIII concludes this paper.

II. Short-Term Load Forecasting

Prediction of the system load over an interval usually from one hour to one week is known as short-term load forecasting (STLF), an important procedure in real-time energy management.

Predicting the energy demand allows an optimal allocation of power, trying to keep the ratio between overall costs and system efficiency as low as possible. This process is also critical for peak shaping and demand response (DR). In fact, an accurate forecasting is an important factor in order to have a fast response predicting demand fluctuations and managing energy storages in an optimal way. Demand Response is a mechanism used to control the peak demand, giving the customer feedback

about the status of the electricity grid and market prices in order to reduce (or postpone) his electricity consumption. An accurate load forecasting allows the customer to plan his activities to consume electricity when the electricity costs are lower, and it also allows the operators to manage the grid to achieve better efficiency and lower costs.

Load forecasting may be included into all the operations performed by Energy Management System (EMS) in order to manage electricity supplies more effectively with the distribution company. Furthermore, a load forecasting is still more critical where the dependence from weather-depending renewable generation is particularly pronounced (micro-grids).

For this kind of problem, various factors should be considered, such as weather data or, more in general, all the factors influencing the load/consumption pattern. This means that for an accurate load forecasting, exogenous variables may be considered and they differ according to customer type: residential, commercial and industrial.

Load data usually exhibits seasonality, sometimes showing more than one periodicity: in fact the load at a given moment may be dependent on the load on the previous hour and also on the previous day and so on.

Forecasting effectiveness can be measured by its accuracy (the difference

between predicted and real value) but also the maximum error achieved (error peak) is a critical factor. In fact, the effectiveness of the energy management of a Smart Grid may be strongly affected by the error peaks and a predictor with low variance might be preferred to a predictor with a better average error but with higher error peaks. Underestimating the energy demand may have a negative impact on the Demand Response and it makes the control of overload conditions harder, especially where energy storage is absent or under-sized. On the other hand, an overestimation may create an unexpected surplus of production. In both cases, the higher the estimation error, the higher the managing costs involved, e.g. an energy district could be forced to buy energy from the grid at higher costs than it would have in case of a better prediction.

A. Real Data

Data used in this work has been collected from seven office buildings, all located in a new industrial area in Rome, Italy. Although there were many various available signals, our work is focused on the following data:

- 1) Electricity hourly overall load data: it takes into account lighting, air conditioning/heating and appliances.
- 2) Water consumption: hot water consumed by people working into the buildings
- 3) Luminosity: data collected from a solarimeter

All the data has been collected for the entire 2010 but for this work, we decided to focus on the period starting from 1/1/2010 to 1/4/2010, for a total of 2160 hourly samples.

In Table 1(a), building characteristics are shown, all the buildings considered have a gas heating system and a water-cooled air conditioning system. Moreover, in Table 1(b), we provide statistical properties of involved data sets.

In Figure 1, for each building, data for the entire available data signal and the partial autocorrelation function (PACF) is plotted. As expected, load data presents an evident daily and weekly periodicity as it is shown by PACF plots in Figure 1.

TABLE 1 Problem information.

(A) BUILDING CHARACTERISTICS					
BUILDING	SURFACE (M^2)	VOLUME (M^3)	NUMBER OF PEOPLE		
A	4968	13414	380		
B	5024	13565	290		
C	4960	13392	210		
D	4968	13414	360		
E	6688	18000	410		
F	4960	13392	440		
G	3360	9072	300		
(B) STATISTICAL INFORMATION ABOUT DATA SETS (MEASURE UNIT IS kW)					
BUILDING DATA	MIN.	1ST QUARTILE	MEAN	3RD QUARTILE	MAX
A	24.2	29	41.8	57	83
B	22.6	25	38.96	58.05	73.8
C	26.8	31.8	41.52	53.8	71.6
D	23.04	29.28	45.15	64.68	91.68
E	33.4	40.4	52.61	66.25	93.4
F	29.8	41.2	46.63	50.80	72.4
G	94.2	109.6	121.7	137.4	184.8

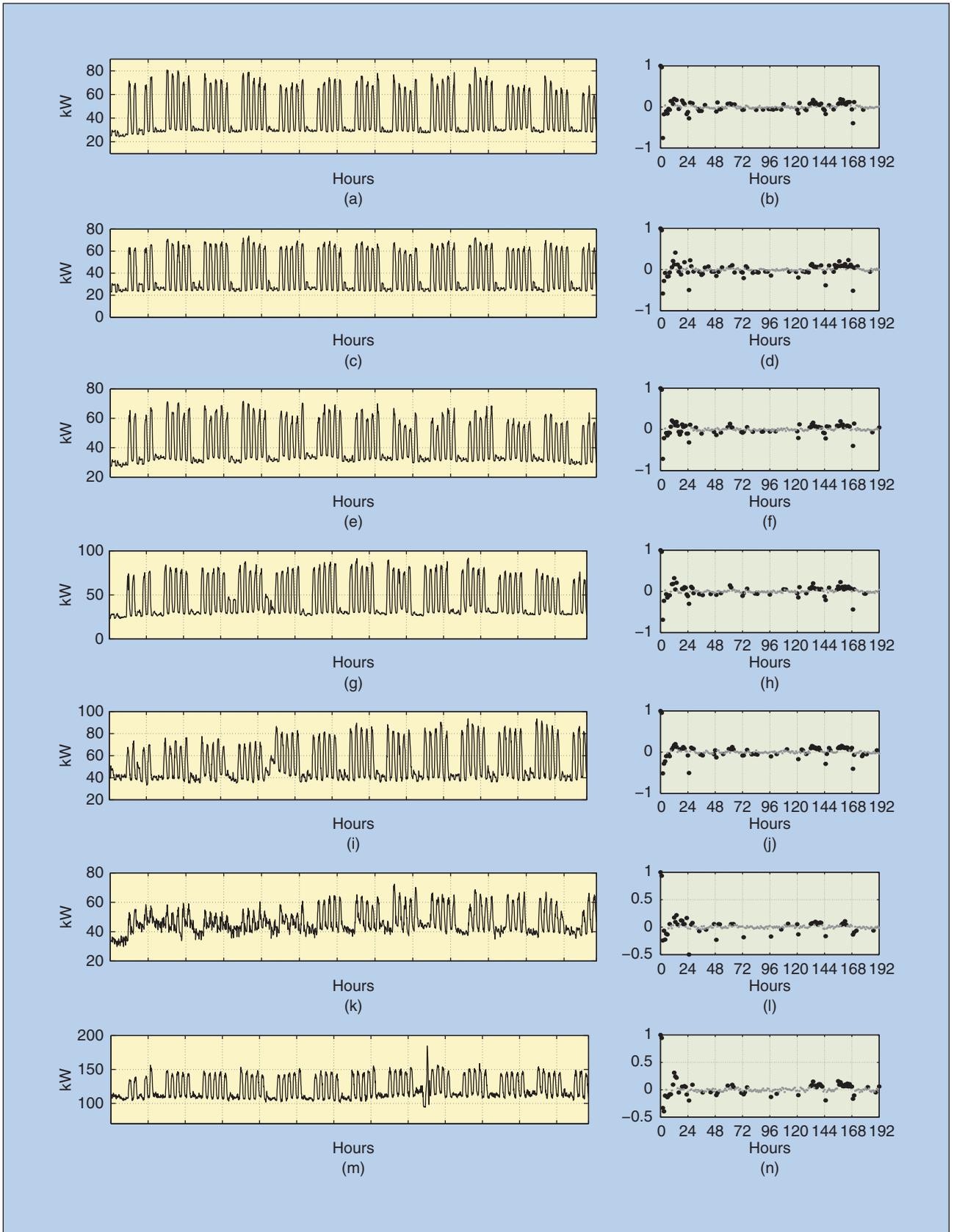


FIGURE 1 Building loads and Partial Autocorrelation Functions (PACF). In PACF plots, light gray points are the points within the 95% confidence bounds. (a) Building A, (b) PACF A, (c) Building B, (d) PACF B, (e) Building C, (f) PACF C, (g) Building D, (h) PACF D, (i) Building E, (j) PACF E, (k) Building F, (l) PACF F, (m) Building G, and (n) PACF G.

The ability to predict future behaviors and energy demand is part of the intelligence needed by Smart Grids.

III. Forecasting Models

In this section, we provide a brief description of the models involved in this work.

A. Naive Model

In order to perform a meaningful comparison for the forecasting, a naive model should be introduced in order to quantify the improvement given by more intelligent and complex forecasting techniques. For seasonal data, a naive model might be defined as:

$$x_t = x_{t-S} \quad (1)$$

with S as the appropriate seasonality period. This model gives a prediction at time t presenting the value observed exactly a period of S steps before. For this work, after the considerations of the previous section, we put the value of $S = 168$ which corresponds to a week given that the data considered is hourly data.

B. Box-Jenkins Models

A time series model is certainly considered the first choice in approaching a forecasting problem, especially the autoregressive integrated moving average model (ARIMA), which considers the nonstationarity of the data, presented in the landmark work of Box and Jenkins [8]. In this model, future value of a signal is assumed to be a linear function of past observations with the addition of an error term (assumed with zero mean and independently and identically distributed).

Given the seasonality of the used data (as described in section II.A) the seasonal variant of the ARIMA models, called SARIMA, has been chosen. SARIMA models has been normally used for forecasting in various application fields (e.g. [9,10]).

Once introduced the backshift and the first difference operators as:

$$B^k x_t = x_{t-k} \quad (2)$$

$$\nabla^n x_t = (1 - B)^n x_t \quad (3)$$

$$\nabla_s^N x_t = (1 - B_s)^N x_t \quad (4)$$

a seasonal ARIMA model denoted as $ARIMA(p, d, q) \times (P, D, Q)_s$ has the following form:

$$\Phi_p(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \alpha + \Theta_Q(B^s)\theta(B)e_t, \quad (5)$$

where x_t is the value of the signal at time t and e_t the error term (supposed to be a white noise process). The terms d and D represent the degree of differencing and the operators $\Phi_p(B^s)$ and $\Theta_Q(B^s)$ are respectively the seasonal autoregressive and the seasonal moving average operators of orders P and Q as:

$$\Phi_p(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_p B^{ps} \quad (6)$$

$$\Theta_Q(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \dots - \Theta_Q B^{Qs}. \quad (7)$$

The non-seasonal operators $\phi(B)$ and $\theta(B)$ are similar to the seasonal ones (eqs. 6 and 7) assuming $s = 1$.

The ARIMA model can be extended adding exogenous inputs I , such a model is called ARIMAX, the same the SARIMA model becomes a SARI-MAX model. Thus, the equation 5 becomes:

$$\Phi_p(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \Gamma I_t + \alpha + \Theta_Q(B^s)\theta(B)e_t. \quad (8)$$

In this paper, the selection process for the values of p, d, q, P, D and Q has been performed with the method proposed by Hyndman and Khandakar [11], implemented with R software [12], which explores the model space selecting the best model via AIC (Akaike Information Criterion) measure.

The model obtained and used in this work has the following orders: $p = 2, d = 1, q = 1, P = 1, D = 1, Q = 1$.

Various implementations of ARIMA/ARIMAX models for STLF has been described in literature, see Cho et al. [13], Fan and McDonald [14], and the review by Hagan and Behr [15].

IV. STLF and Neural Networks

The first applications of NNs to STLF were in the 1990s [16,17] and since then many different applications of NNs to load forecasting were presented.

From the previous models, which linearity can represent a strong limit for complex real-world problems, Neural Networks are non-linear modelling tools designed to find the best input-output mapping of data observed during the so-called 'training' phase. Most widely used are feed-forward neural networks such as multilayer perceptrons (MLPs) and Radial Basis Function Networks (RBFNs), with the weights training usually performed with gradient descent algorithms.

The simplest NN-based prediction structure we can consider for the forecasting is the one which takes in input the lagged samples of the output as:

$$x_t = f(x_{t-1}, \dots, x_{t-N}, w) + e_t. \quad (9)$$

This kind of model is sometimes called Nonlinear Autoregressive Model (NAR) [18] and, differently from ARIMA models presented in Section III.B, it depends also from neural network parameters like weights (w).

Similarly we can use external (exogenous) information I available at time t to make the prediction of x_t and thus the previous model becomes:

$$x_t = f(x_{t-1}, \dots, x_{t-N}, I_t, w) + e_t. \quad (10)$$

The number of inputs nodes, i.e. the number of lags, is a critical variable for the forecasting application. In fact, a low number may not provide enough information for an accurate forecasting and too high number could make the training less effective, due to a larger and more complex solution space. After a set of preliminary tests where we tested several inputs lags combinations, we

selected the lags L corresponding to the previous 24 hours plus one-week and two-weeks lags, thus having $L = [1, 2, \dots, 24, 168, 336]$ (this choice is also based on the analysis of PACF function presented in Figure 1). The neural network model is the following:

$$x_t = f(x_{t-1}, \dots, x_{t-24}, x_{t-168}, x_{t-336}, \mathbf{w}) + e_t. \quad (11)$$

As in Eq. 10, the neural network model with external data is obtained adding to the previous the information I at time t .

The choice of the number of hidden layers used and the nodes included into each of them is usually made following some rule of thumb [19, 20]. Another factor that may affect performance of neural network for time series forecasting is the number of output neurons. If we want to forecast at time $t + 1$ starting from time t (forecasting horizon of one period), then the number of output neurons is obviously one. When the forecasting horizon is greater than one, the number of output neurons varies according to the approach being used. The ‘direct’ forecast method has the number of output neurons equals to the forecast horizon, i.e. the network outputs are $x_{t+1}, x_{t+2}, \dots, x_{t+k}$. On the other hand, if the ‘iterative’ forecast method is adopted, the number of output neurons is equal to one: the predicted value x_{t+1} is used as an input for the successive period prediction, until the end of the forecast horizon. This way of forecasting is the same approach that is used in Box–Jenkins models.

In our work, we chose RBFNs because of their easiness of implementation and training, and their well-established use for classification and forecasting purposes (e.g. see [21, 22]).

In the RBFN design phase, after a set of preliminary tests, we have chosen the value of 64 Gaussian basis functions into the hidden layer. Given that RBFNs radial-basis functions are located into the input space, in our implementation their coordinates are initialized randomly selecting input vectors. Then a scaled-conjugate gradi-

Prediction of the system load over an interval usually from one hour to one week is known as short-term load forecasting (STLF), an important procedure in realtime energy management.

ent (SCG) algorithm is applied for the optimisation of weights and function coordinates. The value of RBF functions used usually represents a trade-off between neural network performances and computational time needed for the training algorithm. In fact, in our case the algorithm involved the calculation of Hessian matrix and its eigendecomposition, both having an exponential scaling runtime with respect to the number of weights (see [23] for more details).

V. Neural Network Ensembles

A neural network ensemble is a combination of a set of NNs which tries to cope with a problem in a robust and efficient way [24]. The simplest way we have to combine M neural networks for a regression problem is an arithmetic mean of their outputs (y_i):

$$\hat{y}_{\text{ens}}(\mathbf{x}_k) = \frac{1}{M} \sum_{i=1}^M \hat{y}_i(\mathbf{x}_k). \quad (12)$$

Negative Correlation Learning (NCL) has been introduced by Liu and Yao [25] with the aim of negatively correlate the error of each network within the ensemble. In this method, instead of training each network separately, a penalty term is introduced to minimize the correlation between the error of the network with the errors of the rest of the ensemble.

The ensembling method we considered in this work is called Regularized Negative Correlation Learning (RNCL) and has been proposed by Chen and Yao in [23]. This method improves NCL adding a regularization term with the objective of reduce the overfitting problem. Regularization helps the network to improve its generalization capability, penalizing large weights which may lead to rough outputs.

In RNCL ensembles, each network i has the following error function:

$$e_i = \frac{1}{M} \sum_{k=1}^N (\hat{y}_i(\mathbf{x}_k) - y(k))^2 - \frac{1}{M} \sum_{k=1}^N (\hat{y}_i(\mathbf{x}_k) - \hat{y}_{\text{ens}}(\mathbf{x}_k))^2 + \alpha_i \mathbf{w}_i^T \mathbf{w}_i. \quad (13)$$

The first term is clearly the error of the i -th neural network, the second the correlation between each network and the output of the ensemble (see Eq. 12) and finally the last term is the regularization term with its parameter $\alpha \in [0, 1]$.

VI. Experimental Studies

In this section, we evaluate the RNCL neural networks ensembles on buildings data, finally comparing it with the SARIMA model. In the first part, we examine the performance of the ensemble of 20 RBFNs with respect the single performance of each network, then we compare both with a SARIMA model. Finally, we would like to show how the methodologies proposed are able to exploit information contained into additional data.

In our experiments, we considered the following iterative forecasting methodology:

- 1) We train a model (RNCL ensemble or SARIMA model) on a part of the dataset (*training set*)
- 2) We test the model on the remaining part of the data set (testing part), starting from time $t + k$, with $k \in [1, N]$, and predicting up to 24 hours ahead ($t + 1, \dots, t + 24$)
- 3) We repeat step 2. for each value of k

It’s important to note that the predicted load is used as ‘past’ data from the model (see sections III and IV), i.e.

TABLE 2 RBFN errors on testing part with standard deviations. All the values are rounded to two decimals.

BUILDING	MAE			MSE			MAPE		
	AVG.	STD.	RNCL	AVG.	STD.	RNCL	AVG.	STD.	RNCL
A	6.84	1.75	5.57	81.73	35.63	46.63	19.02	5.83	16.21
B	5.79	2.60	3.83	65.09	52.59	25.98	18.27	9.49	12.84
C	5.31	1.38	4.04	49.25	21.80	25.84	14.82	4.09	11.86
D	7.82	2.31	6.03	106.24	54.93	50.92	21.88	7.67	17.49
E	2.99	0.69	2.33	16.21	8.07	9.79	5.69	1.60	4.26
F	4.11	0.11	3.98	39.86	1.95	38.05	8.76	0.26	8.48
G	3.85	0.44	3.12	28.51	8.86	16.56	3.16	0.36	2.55

TABLE 3 Ratio of the testing error of the best/worst neural network to its respective ensemble (ratio_{BEST}/ratio_{WORST}).

	A	B	C	D	E	F	G
ratio _{BEST} (MAE)	0.67	0.78	0.74	0.64	0.97	0.99	1.03
ratio _{BEST} (MAPE)	0.59	0.65	0.62	0.51	0.99	0.99	1.04
ratio _{WORST} (MAE)	1.94	3.36	2.06	2.13	2.24	1.08	1.50
ratio _{WORST} (MAPE)	2.02	3.48	1.85	2.25	2.61	1.09	1.52

the error propagates into the future. The proposed methodology can be easily turned into a continuous learning methodology. In fact at step 2, instead of using the model created during the previous step (i.e. on the training part) we may update the model with a selected strategy.

The training part considered is the data between the 1/1/2010 and 19/3/2010 (1848 hourly samples), the remaining part (until 1/4/2010) has been assigned to the testing set (312 hourly samples).

For this work, we considered three different error measures. Two of them refers to absolute error, the first one is the Mean Absolute Error (MAE) which the following formula:

$$MAE(y) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (14)$$

and the second one is the common Mean Squared Error (MSE) defined as:

$$MSE(y) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (15)$$

Then the other measure is the Mean Absolute Percentage Error (MAPE) defined as:

$$MAPE(y) = 100 \cdot \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (16)$$

Both absolute and percentage error measures are important for this work: the former is fundamental to evaluate the practical utilisation of presented techniques, the latter allows to compare the methodologies on buildings with different scale of values.

At the end, we have the forecasting testing error matrix (E_T) which contains all the 24-hours forecasting errors $e(k) = y(k) - \hat{y}(k)$:

$$E_T = \begin{bmatrix} e(k) & e(k+1) & \dots & e(k+24) \\ e(k+1) & e(k+2) & \dots & e(k+25) \\ \vdots & \vdots & \ddots & \vdots \\ e(k+i) & e(k+i+1) & \dots & e(k+i+24) \end{bmatrix}. \quad (17)$$

All the error measures shown in this section are computed on the entire matrix E_T .

As already stated in section IV, NNs weights are usually initialized with uniform random numbers and training algorithms are normally strongly affected by starting conditions. For this reason, NNs trained with the same algorithm tend to exhibit a large variety of errors, this variability is summarized in Table 2 where average errors with

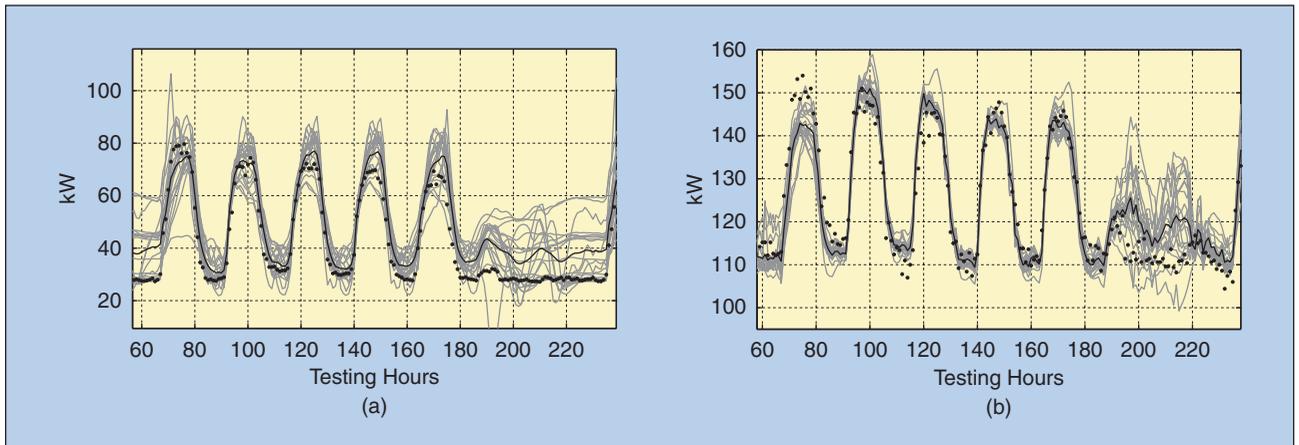


FIGURE 2 Portion of the 12-hours ahead forecasting obtained by each single RBFN (light gray) and the whole ensemble (black line) compared with the target signal (black dots). (a) Building D and (b) Building G.

standard deviations of the testing errors of RBFNs are provided. For each error measure, we added a column showing the value of the RNCL ensemble.

We can see how the performances of neural networks are evidently problem-dependent, in the buildings considered in this work, we can observe a percentage error very low (2.55%) for building G and on the other hand, particularly high for building D (17.49%). It's worth nothing that in the first case we observe a low standard deviation, in fact the RBFNs show a MAPE between 2.66% and 3.89%, so the RNCL ensemble gives a performance better than the best network within the ensemble, on the other hand, in building D the error is between 8.96% and 39.27%, with the ensemble showing an error twice higher than the best network. We summarize this ratio between the error made by the best network and its ensemble in Table 3, where we can evaluate the importance of neural network ensembles on dealing with the variability of neural networks performance: it seems obvious that using an ensemble you can improve the results obtained by the worst NN among a large number (in this case 20) but it may be surprising to see that it allows (like in building G) to improve the results of the best NN.

To give a visual example of the estimation provided by the ensemble with respect to each RBFN composing it, see Figure 2 where an example of 12-hours ahead forecasting is provided. It is clearly evident how during the weekends (right part of the shown signal), neural networks have not enough information to give an accurate forecast and in fact their output becomes extremely 'noisy' and inaccurate.

In Table 4 the comparison, focused on MSE and MAPE error, with naive and SARIMA model is provided. We can observe that in the first four buildings (A-B-C-D) the RNCL ensemble shows a drastic higher error than SARIMA and naive model, while in the last three buildings (E-F-G) network ensembles performs better than naive model with results nearer to SARIMA.

A reliable forecasting of load demand could help to avoid dispatch problems given by unexpected loads, and to give vital information to make appropriate decisions on energy generation and purchase, especially market-based dynamic pricing strategies.

A. Introduction of External Data

Given the ability to find relationships between input and output data of neural networks, here we investigate the effects on neural network results with the introduction of additional information as inputs, in particular with the following four inputs:

- 1) Hour of the day (1-24)
- 2) Working-day flag (0/1)
- 3) Hot water hourly consumption (l)
- 4) Luminosity (lux)

The first two gives important information about the expected load pattern, in fact in office buildings is typical to have the demand concentrated during the working hours (9 AM - 17 PM in our case) and working days (Monday-Friday in our case). Differently, other two measures can give information about a part of the demand (luminosity is related to lighting consumption) and about the activity inside the building.

As stated in Eq. 10 we use the external information at time t to predict $y(t)$, this is possible because a part of external information (I^k) are known in the future, like the hour of the day and the working day flag. The other two measures (I^u) are forecasted as well, so the Eq. 10 becomes:

$$x_t = f(x_{t-1}, \dots, x_{t-N}, I_t^k, \hat{I}_t^u), \quad (18)$$

where \hat{I} is the information predicted with a neural network.

Table 5 shows the variation of MAPE error due to the introduction of the four additional inputs in both neural network and SARIMA models. Neural networks are evidently able to exploit the useful information for forecasting contained into additional data (-40% average variation) more than SARIMA model (-1.12% average variation).

In order to analyze the improvement given by the introduction of external

TABLE 4 Comparison between neural networks (best testing one and RNCL ensemble), naive model and SARIMA model. In bold the lowest error for each building.

BUILDING	MSE				MAPE			
	BEST RBF	RNCL	NAIVE	SARIMA	BEST RBFN	RNCL	NAIVE	SARIMA
A	25.55	46.63	13.89	10.48	9.62	16.21	5.06	4.27
B	14.39	25.98	3.98	2.10	8.35	12.84	3.60	2.90
C	17.21	25.84	8.1	3.54	7.43	11.86	4.96	3.19
D	27	50.92	14.87	11.85	8.96	17.49	6.03	5.91
E	8.63	9.79	15.96	8.68	4.22	4.26	5.14	3.93
F	37.29	38.05	50.11	35.95	8.41	8.48	9.2	8.58
G	18.48	16.56	18.06	14.24	2.66	2.55	2.76	2.42

TABLE 5 Effects of the introduction of additional data on MAPE error with SARIMA and RNCL ensemble. In bold the lowest error for each building.

BUILDING	RNCL ENSEMBLE			SARIMA		
	NO EXT.	EXT	±	NO EXT.	EXT.	±
A	16.21	4.60	-71.6%	4.27	4.23	-0.94%
B	12.84	3.60	-80%	2.90	2.91	+0.34%
C	11.86	4.1	-65.4%	3.19	3.02	-5.33%
D	17.49	6.84	-60.9%	5.91	5.95	+0.67%
E	4.26	4.16	-2.3%	3.93	3.88	-1.27%
F	8.48	8.51	+0.35%	8.58	8.57	-0.12%
G	2.55	2.6	+1.96%	2.42	2.39	-1.24%

Forecasting effectiveness can be measured by its accuracy (the difference between predicted and real value) but the maximum error achieved (error peak) is a critical factor.

data we can examine the difference between errors made by RNCL ensemble before and after the introduction of additional inputs. This kind of analysis is

visible in Figure 3 where we show the difference of absolute error ($E_T^- - E_T^+$) between the testing matrix (see Eq. 17) of the RNCL without external data

(E_T^-) and with additional data (E_T^+), positive errors mean that external data have improved the forecasting, vice versa for negative errors. Results shown in Figure 3 reflect the values presented in Table 5, in some buildings the introduction of external data drastically improves the forecasting accuracy. We can observe different error patterns for each building but however we can draw some general conclusions: in all the buildings the improvement is located more frequently

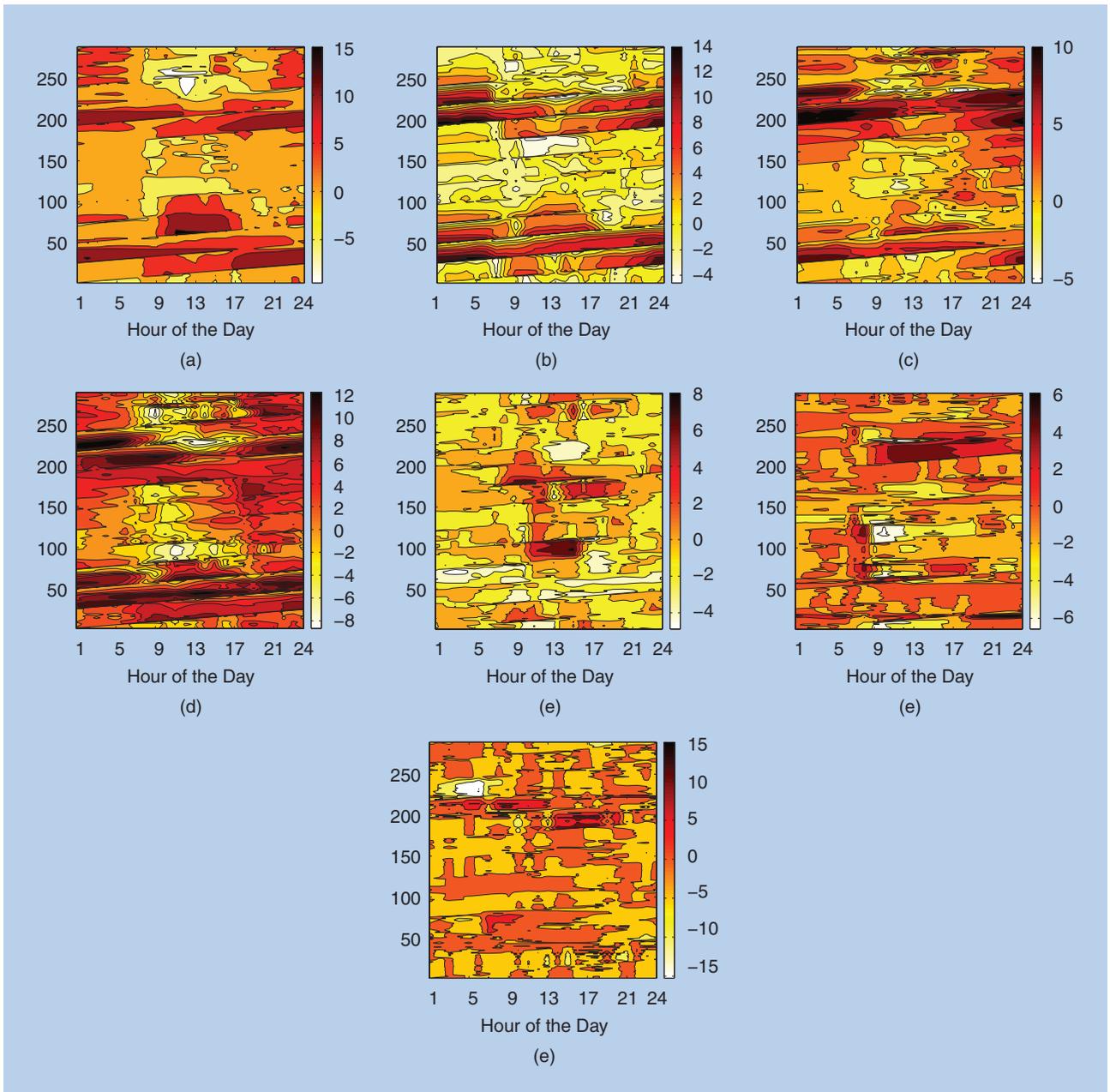


FIGURE 3 Comparison between error matrix ($E_T = E_T^- - E_T^+$) of RNCL ensemble before (E_T^-) and after (E_T^+) the introduction of external data. Errors shown are absolute (kW). (a) Building A, (b) Building B, (c) Building C, (d) Building D, (e) Building E, (f) Building F, and (g) Building G.

on non-working hours (hours 1 – 8 an 17 – 24) than in central hours and there are some particular time periods (in range 30 – 70 and 200 – 230), when weekends occur, with clear improvements (visible as darker horizontal bands). As shown in Figure 2, it is evident how additional data help neural networks to predict the behavior outside working hours, as expected from the introduction of information about the working days and weekends.

Finally, an analysis of the maximum absolute error achieved by RNCL and SARIMA models is given in Figure 4. In four on seven buildings (A-B-D-F) RNCL ensemble produces a lower maximum absolute error than SARIMA model and however only in building G the RNCL maximum error is markedly greater (almost twice) than SARIMA.

VII. Discussion

Our comparison is aimed at showing how the various techniques are able to exploit additional information and how the RNCL ensemble method leads to a drastic improvement over the utilisation of neural networks for STLF.

Results shown in Table 4 underlines the best performances of a SARIMA model on all the buildings taken into account in this paper. It may be not surprising to see that in some buildings (e.g. B and F) a naive model gives error values near the ones obtained with the other two models, this is a clear sign that the ‘conservative’ behavior of that model, which proposes as a prediction the output already observed (see Section III.A), is able to cope in some cases with complexity as good as more advanced models in absence of useful and reliable information for the forecasting.

As specified before, the number of radial-basis functions into the hidden layer has been chosen after a set of preliminary experiments. In order to give more details about the behavior of RNCL ensembling, and more in general RBFNs, we gives in Table 6 average errors for four different number of functions into the hidden layer. We can

RNCL ensemble is able to achieve a marked error reduction after the introduction of external data, while a linear model such as ARIMA/SARIMA, in spite of its good results presented in this paper, is not able to exploit the introduction of additional data.

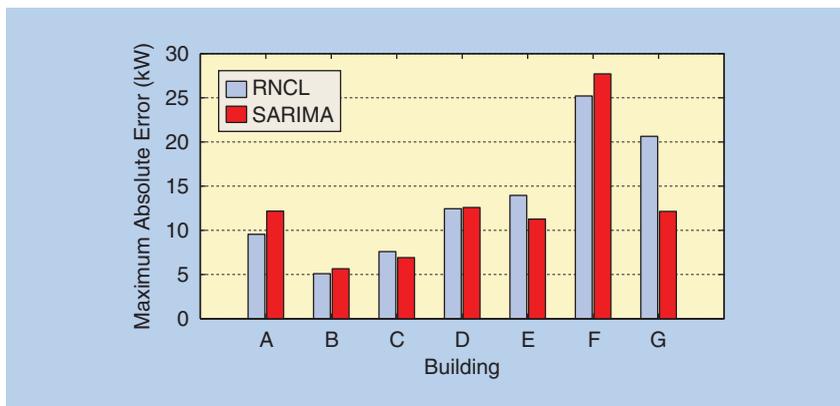


FIGURE 4 Maximum absolute error obtained by RNCL and SARIMA models with additional data.

observe how the performance degrades with the increasing of the number of hidden functions, a clear example of overfitting phenomenon.

Another critical topic raised by experimentations is the feature selection process. As we observed, the introduction of additional data has improved the performances of neural networks more than SARIMA model: the former includes additional data in a nonlinear way (see Eq. 10) trying to model even complex relationships between input and outputs, while the latter incorporates external data as a simple linear factor (see Eq. 8). In real-world cases, we usually have available a large amount of different data typologies and for each of them, we have multiple options about the choice of sampling rate, normalization, smoothing factors, etc. Thus, the problem of selecting the optimal subset of available data becomes critical for two

reasons: at first, a selection allows to reduce the data dimensionality with positive consequences (e.g. faster computation or reduction of memory requirements) and secondly, the process of feature selection may gives useful information about the system we are observing.

VIII. Conclusion

This paper analyzes the application of Regularized Negative Correlation Learning (RNCL) ensemble methodology [23] to the problem of energy load hourly prediction, with an investigation on the effects of the introduction of external data, i.e. information related to the energy load. Various models of different complexity have been tested on real load data coming from seven office buildings located in Rome, Italy. Although selected buildings are not part of a Smart Grid, we think that the

TABLE 6 Average errors on all the buildings obtained with RNCL ensemble at the change of the number of functions into the hidden layer.

NUMBER OF FUNCTIONS	NUMBER EXTERNAL DATA			WITH EXTERNAL DATA		
	MAE	MSE	MAPE	MAE	MSE	MAPE
16	3.33	21.84	7.52	2.50	14.16	5.10
32	3.89	28.57	9.65	2.42	13.32	4.97
64	4.13	30.54	10.53	2.39	12.61	4.92
128	4.30	32.32	11.04	2.56	13.72	5.45

An ensemble shows the possibility to combine the output of several neural networks achieving a better trade-off between variance of output and error performance.

questions raised by this paper can be extended to any more general scenario.

Given the increasing criticality of load forecasting for electric grids, especially when hardly predictable factors are introduced (e.g. generation from renewable sources) and where complex factors like dynamic pricing and peak/load shifting become common and desirable.

A common drawback about the application of neural networks is their variability due their high sensitivity to initial conditions. A high variance of performances measures may be a big limit for the applicability of such techniques to an engineering field where the reliability is commonly preferred to the overall average accuracy. RNCL ensemble shows the possibility to combine the output of several neural networks achieving an interesting trade-off between roughness of output and error performance.

The comparison with a common seasonal time-series model such as SARIMA and a naive model gives useful information about the utilisation of neural network ensembles. Furthermore, this work underlines the importance of a comparison with trivial models, especially in forecasting, in order to understand the intrinsic characteristics of the data object of forecasting (an interesting discussion about the creation of error measures including this factor is in [26]).

RNCL ensemble is able to achieve a marked error reduction after the introduction of external data, while a linear model such as ARIMA/SARIMA, in spite of its good results presented in this paper, is not able to exploit the introduction of additional data. In this work, after an investigation about HVAC systems and people activities, and after a statistical analysis of data sets, we selected a subset of additional inputs. However, it would be interest-

ing a further investigation about the selection of optimal subset among all the available inputs to reduce forecasting errors and improve the knowledge about the correlations present between available data. Similar conclusions may be drawn on the selection of optimal input lags (L) in order to reduce the complexity of NN parameters space and speed up the training phase.

An interesting development would be the introduction of a feature selection mechanism into the overall forecasting framework, in order to have an optimal set of inputs for each forecasting problem. Finally, another future step would be the application of RNCL and neural network on a realistic scenario involving electricity generation and energy storage with the analysis of the obtained cost/efficiency.

Acknowledgments

Part of this work was done while the first author visited CERCIA, School of Computer Science, University of Birmingham, UK. We would like to thank H. Chen for providing the RNCL source code and for his support. This work was partially funded by EU IntellicIS (COST Action IC0806) and by CERCIA.

References

- [1] [Online]. Available: <http://www.addressfp7.org/>
- [2] Z. A. Vale, H. Morais, H. Khodr, B. Canizes, and J. Soares, "Technical and economic resources management in smart grids using heuristic optimization methods," in *Proc. Power and Energy Society General Meeting*, New York: IEEE, July 2010, pp. 1–7.
- [3] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, 2009.
- [4] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Berlin: Springer-Verlag, Mar. 2002.
- [5] E. A. Feinberg and D. Genethliou, "Load forecasting," in *Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence*, J. Chow, F. Wu, and J. Momoh, Eds. Berlin: Springer-Verlag, 2005, pp. 269–285.
- [6] H. Hippert, C. Pedreira, and R. Souza, "Neural networks for short-term load forecasting: A review and

evaluation," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, Feb. 2001.

[7] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, 1990.

[8] G. E. P. Box and G. M. Jenkins, *Time Series Analysis; Forecasting and Control*. San Francisco, CA: Holden-Day, 1970.

[9] L. Fillatre, D. Marakov, and S. Vaton, "Forecasting seasonal traffic flows," *Comput. Sci. Dept.*, ENST Bretagne, Brest, Paris, 2003.

[10] P. Pai and C. Lin, "Using support vector machines to forecast the production values of the machinery industry in Taiwan," *Int. J. Adv. Manufact. Technol.*, vol. 27, no. 1, pp. 205–210, 2005.

[11] R. J. Hyndman and Y. Khandakar. (2007, June). *Automatic time series forecasting: The forecast package for R*. Dept. Econometrics Bus. Statist., Monash Univ., Monash Econometrics and Business Statistics Working Papers 6/07. [Online]. Available: <http://ideas.repec.org/p/msh/ebswps/2007-6.html>

[12] R. Development Core Team. (2010). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. [Online]. Available: <http://www.R-project.org/>

[13] M. Cho, J. Hwang, and C. Chen, "Customer short term load forecasting by using ARIMA transfer function model," in *Proc. Int. Conf. Energy Management and Power Delivery (EMPDP'95)*, Nov. 1995, vol. 1, pp. 317–322.

[14] J. Fan and J. McDonald, "A real-time implementation of short-term load forecasting for distribution power systems," *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 988–994, 1994.

[15] M. T. Hagan and S. M. Behr, "The time series approach to short term load forecasting," *IEEE Trans. Power Syst.*, vol. 2, no. 3, pp. 785–791, 1987.

[16] T. Peng, N. Hubele, and G. Karady, "An adaptive neural network approach to one-week ahead load forecasting," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1195–1203, 1993.

[17] M. Djukanovic, B. Babic, D. Sobajic, and Y.-H. Pao, "Unsupervised/supervised learning concept for 24-house load forecasting," *IEE Proc. Gener., Trans. Distrib.*, vol. 140, no. 4, pp. 311–318, 1993.

[18] T. Chow and C. Leung, "Nonlinear autoregressive integrated neural network model for short-term load forecasting," *IEE Proc. Gener., Trans. Distrib.*, vol. 143, no. 5, pp. 500–506, Sept. 1996.

[19] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[20] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[21] D. Wedding, et al., "Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model," *Neurocomputing*, vol. 10, no. 2, pp. 149–168, 1996.

[22] I. Nabney, "Efficient training of RBF networks for classification," in *Proc. Ninth Int. Conf. Artificial Neural Networks (ICANN'99)*, 1999, vol. 1, pp. 210–215.

[23] H. Chen and X. Yao, "Regularized negative correlation learning for neural network ensembles," *IEEE Trans. Neural Networks*, vol. 20, no. 12, pp. 1962–1979, 2009.

[24] X. Yao and Y. Liu, "Making use of population information in evolutionary artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 3, pp. 417–425, 1998.

[25] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Netw.*, vol. 12, no. 10, pp. 1399–1404, 1999.

[26] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.