

Short-Term load forecasting using a neuro-fuzzy model based on entropy maximisation

Georges Kariniotakis

▶ **To cite this version:**

Georges Kariniotakis. Short-Term load forecasting using a neuro-fuzzy model based on entropy maximisation. Med Power 2002, Nov 2002, Athènes, Greece. hal-00534191

HAL Id: hal-00534191

https://minesparis-psl.hal.science/hal-00534191

Submitted on 4 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Short-Term Load Forecasting Using a NeuroFuzzy Model Based on Entropy Maximisation

George Kariniotakis
Ecole des Mines de Paris,
Centre d'Energétique,
B.P. No 207, 06904 Sophia-Antipolis, France.
Tel: +33-493957599, Fax: +33-493957535, kariniotakis@cenerg.cma.fr

ABSTRACT: The paper presents a new short-term load forecasting approach based on dynamic fuzzy logic modelling. The developed model produces forecasts for the next 48 hours, which are updated every hour. Such a sliding window scheme is different than conventional models that operate usually once a day. The paper emphasizes on developing appropriate learning and on-line adaptation schemes based on the maximal entropy principle. In contrast to the traditional approach, such schemes permit to avoid overfitting of the model to the data. Thus, the ability of the model to predict new data (generalisation) is maximized. The architecture of the model is selected using non-linear optimisation techniques such the non-linear Simplex. The model has been developed in the frame of the EU research project *More-Care* and implemented for on-line use at the islands of Crete and Madeira. Results from the case studies are presented showing the efficiency of the approach.

Keywords: Short-term load forecasting, adaptive fuzzy-neural networks, entropy, on-line software.

I. INTRODUCTION.

Several approaches have been developed for load forecasting, which has become nowadays one of the major areas of research in electrical engineering. Forecasts, ranging from some minutes up to several days ahead are required by various functions of power system management and operation. Especially in the last years, following the liberalization of power supply industry in several countries, the accuracy of load forecasts is seen as a major criterion for the good performance of the various players of the market.

The load process as such, depends on the hour of the day and the day of the week, while several external factors like temperature and other weather-related variables play an important role. By using a simple ('naïve') predictor such as the load of the same hour of previous day or same day last week, then a MAPE (mean absolute percentage error) performance up to10 % can be obtained.

Paper accepted for presentation at the 3rd Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion MED POWER 2002, jointly organized by National Technical University of Athens, IEE Hellas, Israel and Cyprus, Athens, Greece, November 4-6, 2002

The contribution of an advanced method in improving this error is translated in important economic gains [1]. A great variety of forecasting methods has been applied on the problem. They may be classified as time series models, in which the load is modelled as a function of its past-observed values, and casual models, in which the load is modelled as a function of exogenous variables, especially weather or social ones [2]. In the last decade, artificial intelligence methods, and mainly neural networks, have been extensively used for load forecasting. An extensive review is provided in [2] with an analysis of the modelling problems related to artificial intelligence based time-series forecasting.

The present paper presents a global methodology for developing an artificial intelligence based model for on-line use. This methodology is characterised by the following steps:

1. Selection of the architecture and the types of input of the model using a non-linear optimisation algorithm.
2. Development of an appropriate generalisation criterion for taking decisions on the model architecture.
3. Development of an efficient learning procedure based on cross-validation and early stopping of learning process in order to avoid overfitting.
4. Employment of simulated annealing for the learning rates to control learning.
5. Implementation of a regularisation process; for this, an appropriate learning algorithm is developed for estimating the parameters of the model based on error and information maximisation. Regularisation permits to obtain models with optimal generalisation.

Given that the above procedure of model development is automatised, it is characterised by "objectivity" and leads to parsimonious model configurations with optimal performance in out-of sample data.

Once, the optimal model is selected, it is then implemented as an on-line module able to operate in a self-adapting mode. Here, the on-line module has been integrated in the "More-Care" Energy Management System (EMS) developed within the frame of a European Community project. More-Care aims to optimise the management of autonomous power systems with high integration from renewable energy sources. The power system scheduling is performed by economic dispatch and unit commitment modules. A primary input to these functions is forecasts of load and renewables units production. Given the higher variability of load in such power systems and the intermittent production of renewables, frequent updates of the forecasts are required.

Moreover, in autonomous systems the short-term load-forecasting task becomes more difficult due to the fact that the quality of input load data is often affected by unpredictable events such as power-cuts, or even blackouts, or interruptions of communication with the SCADA system [4,6].

In interconnected systems, load forecasts are updated a limited number of times per day, i.e. once or twice a day and at specific time origins. Here, the developed model operates with a sliding window scheme, that is, it produces forecasts every hour covering a horizon of the next 48 hours.

The case-studies of the Islands of Crete in Greece and Madeira in Portugal are presented. The load-forecasting module has been integrated within the More-Care EMS and installed for on-line operation at these islands. The input to the model is only past load values. However, due to the self-adaptive operating mode, the model is able to account for the influence of temperature or other external factors.

II. DESCRIPTION OF THE PREDICTION MODEL.

The forecasting model receives past hourly values of load as input, as well as explanatory data, such as the type of the day and the type of the hour. The general form of the model is:

$$\hat{P}(t+1/t) = f(P(t), P(t-1), \ldots, P(t-m), D(t+1), H(t+1)) \qquad (1)$$

where :

$[P(t), P(t-1), \ldots, P(t-m)]^T$ is the vector of past load values,

$D(t+1)$ is the type of day to which the predicted load corresponds. It takes values between 1-7. Given the properties of the fuzzy model for local modelling, this input variable permits to account for weekend days. If one associates values >7 to special days, then the same single model may handle also this type of days.

$H(t+1)$ is the type of hour to which the predicted load corresponds. It takes values between 1-24.

$f(.)$ is the generic fuzzy-neural function described below.

Multi-step ahead forecasts are generated using (1) in an iterative way, i.e., in order to produce a forecast for $t+2$, the forecast for $t+1$ is fed back as input to the model.

An alternative approach is to develop multi-output models, or to tune a different model for each time-step. The implementation of these approaches is more complex and requires high development effort. The inclusion of H and D variables in (1) compensates for the particularities of multi-step ahead forecasting.

A. The fuzzy-neural network model.

The fuzzy model can be expressed in the form of rules of the type:

"IF $\underline{x}$ is A THEN y is B"

where $\underline{x}$, y are linguistic variables and A, B are fuzzy sets. In the case of time-series prediction, rules may have the form:

R: IF x_1 is A_1, and …, and x_n is A_n THEN $y = g(x_1, \ldots, x_n)$

where:

$x_1,\ldots,x_n$ are real-valued variables representing input variables of the system defined in the universes of discourse $X_1,\ldots,X_n$ respectively.

$A_1,\ldots,A_n$ are fuzzy sets.

y is variable of the consequence whose value is inferred. In the specific problem it represents future load $\hat{P}(t+1), \hat{P}(t+2), \ldots$

$g(.)$ is a function that implies the value of y when $x_1,\ldots,x_n$ satisfy the premise. The function $g(.)$ in the consequent part of the rules may be a linear or a non-linear one or even a constant. In the case of a linear function the fuzzy rule-base takes the form:

R^1: IF x_1 is A_1^1,…, and x_n is A_n^1 THEN $y^1 = p_0^1 + p_1^1 x_1 + \ldots + p_n^1 x_n$

$\vdots$ $\vdots$

R^m: IF x_1 is A_1^m,…, and x_n is A_n^m THEN $y^m = p_0^m + p_1^m x_1 + \ldots + p_n^m x_n$

Each rule gives an estimation of the output y^i according to the conditions defined by the fuzzy sets in the premises. In the context of timeseries prediction, each variable x_i in the premise corresponds to a past value of the process (i.e. power: $P(t)$, $P(t-1)$…), or past values of explanatory input (i.e. type of day $D(t+1)$, type of hour $H(t+1)$, etc.

A linear function in the consequence is indeed an ARX (autoregressive with exogenous variables) model. It is clear that with the above definitions, the rule-base consists of an ensemble of "local" models. Local modelling is a desired property of theprediction model since it permits to account for special situations like holidays or weekends.

Fuzzy sets in the premises are modelled here using Gaussian functions:

$$\mu_{A_j}(x_j) = \exp\left(-\left(\frac{x_j - a_j^i}{b_j^i}\right)^2\right)$$

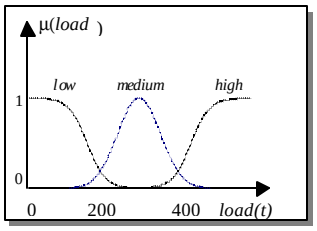


Figure 1: *Representation of fuzzy load. "Load" is a linguistic variable with three terms "low", "medium", and "high" represented as fuzzy sets with the membership functions shown in the Figure.*

In the case of a linear autoregressive function with exogenous variables in the consequence, the model may be written analytically as following:

$$\hat{y} = \frac{\sum_{i=1}^{r}\left(p_0^i + \sum_{j=1}^{m} p_j^i x_j\right)\prod_{j=1}^{m}\mu_{A_j^i}(x_j)}{\sum_{i=1}^{r}\prod_{j=1}^{m}\mu_{A_j^i}(x_j)} = \frac{\sum_{i=1}^{r}\hat{y}^i w^i}{\sum_{i=1}^{r} w^i} \qquad (2)$$

In (2), the unknown parameters to be estimated through a learning procedure are: p_j^i, a_j^i, b_j^i. It is however necessary to define the number of rules in the fuzzy rule-base depending

on the input variables selected and the number of fuzzy sets associated to each variable.

III. MODEL DEVELOPMENT AND GENERALIZATION.

Model building is characterized by two phases: optimisation of the model architecture and tuning of the model internal parameters (learning).

These two phases are driven by the requirement for good "generalization". Generalization is the capacity of the model to perform well when it predicts new data (data not used during the two phases of model development). It is a primary requirement for the efficient on-line use of a model.

In order to optimise generalisation, the tuning of the model parameters is performed taking into account [5]:

- *Learning rules* based on stochastic gradient for tuning the parameters a, b, p of the model.
- Learning rules are appropriately developed to minimize simultaneously prediction error and the Information content of the model (max entropy). This acts as a *self-regularization* process that permits to avoid overfitting of the data.
- *Simulated annealing* is performed for controlling the evolution of the learning process through appropriate adaptation of the learning rate.
- *Early-stopping* is applied to the learning process is early-to avoid overfitting.
- *Cross-validation* is applied to terminate learning. For this purpose, a subset of the data (validation set S_V) is reserved.

The cross-validation criterion (*prediction risk*) is expressed as a weighted function of the performance of the model over the whole prediction horizon. By this way, generalization is optimised for multi-step ahead prediction, which is the ultimate purpose of the model.

The above process permits to tune optimally a model with a specific architecture. The architecture of a model is defined by the types of input variables and the number of fuzzy sets associated to each one. For each type of input data it is needed to decide the number of past values to be used as input.

This selection process, which is also similar to other types of models like neural networks, is a time consuming one due to the infinite number of combinations that can be tested. Often it is performed by trial-and-error, where several candidate configurations are tested. It is noted that the evaluation of each candidate model requires carrying out complete learning for estimating the model parameters.

In this work, the trial-and-error has been replaced by a fully automated process for model architecture optimisation. The constrained non-linear simplex ("Complex") optimisation algorithm is used for this purpose. The algorithm has been modified for handling both discreet and continuous decision variables [5]. The optimisation process is based on the evaluation of the surface of the generalization function (defined as the performance of a model on the validation set) using a complex of points. Each point corresponds to a candidate model. The computational cost is high due to the necessity of the algorithm to tune each candidate model. However, in global, the automatic nature of the process permits to save considerable engineering time compared to the trial-and-error.

An alternative genetic algorithm approach did not present any advantages with respect to the simpler "Complex" algorithm. Genetic algorithms appeared to be less parsimonious w.r.t the number of models they need to test in order to converge compared to the Complex algorithm.

Each decision variable q_i in Complex represents the number of fuzzy sets associated to each type of input data. In the special case, when the algorithm converges to zero-number of fuzzy sets for a specific type of data, then this input is excluded from the model as non-significant. By this way the algorithm performs input selection. When the number of fuzzy sets is converging to one, then the variable does not participate in the premises, but appears only in the function of the consequent part. Parsimony in the selection of input is critical to avoid overfitting from overparametrized models.

It is worthwhile to notice that, when the number of fuzzy sets for all variables is either one or zero then a single "rule" is obtained. The premise has no significance and the model corresponds to a simple linear function of the input variables. This limit case corresponds to the ARX class of models. Consequently, the optimisation process can indeed exclude the use of a non-linear fuzzy model and lead to a classical linear one. By this way, a parallel selection between linear and non-linear models is performed.

The model architecture optimisation problem is formulated as following:

$$\min_{q} \quad J_V(q) = \frac{1}{2}\sum_{k=1}^{n} w_k^J \sum_{P(t+k)\in S_V} \left(P(t+k) - \hat{P}(t+k/t)\right)^2$$

under the constraints:

1. $q_{\min} \le q \le q_{\max}$
2. $q_{FM\,\min} \le \prod_{q_i \in q_{FM}} q_i \le q_{FM\,\max}$
3. q_i=integer, $q_i \in q_{FM}$
4. $\hat{P} = f_M\left(z^* / q\right)$, z^*: $\min_{z} \quad J_L(z/q) = \frac{1}{2}\left(P(t+k) - \hat{P}(t+k/t)\right)^2$, $\forall\, P(t+k) \in S_L,\ k = 1,\ldots n$

- The optimization function $J_V(q)$ is the *prediction risk* over the whole look-ahead horizon n estimated using the data of the validation set S_V. The weights w_k^J serve to define the importance of each time step (i.e. to indicate that a short-term or long-term performance is targeted).
- The explicit constraint 1 determines the universe of discourse of the vector $q = [q_{FM} : q_{LRS}]^T$ of decision variables. This vector contains the number of fuzzy sets associated to each input variable (q_{FM}) and other optimized variables like the learning rates (q_{LRS}).

- The implicit constraint 2 defines upper and lower limits for the number of rules. The number of rules is given by the product of the number of fuzzy sets associated to each input variable. The lower limit q_{FMmin} should be greater or equal than 2 if one would like to obtain a fuzzy model. A value of one for the lower limit gives the possibility to the algorithm to propose an autoregressive model when such models appear in the consequence of a rule.
- The constraint 3 denotes that the number of fuzzy sets takes integer values.
- The implicit constraint 4 means simply that forecasts on the validation set are made using a fuzzy model $f_M(.)$ having an architecture defined by q and with parameters z^* that have been obtained after an appropriate training process.

IV. LEARNING AND INFORMATION MINIMIZATION.

Too complex model architectures have the capacity to store not only the principal (and predictable) characteristics and regularities of the data but also the (unpredictable) noise. The incorporation within the model of non-predictable information deteriorates generalization. However, by forcing the internal representation of a model, in an unsupervised way, to minimize the information transferred from the data, one can ameliorate generalization.

In this Section, a methodology is proposed according to which, the generalization of a fuzzy model can be improved if the *Information* content stored to the model is minimal, or equivalently, if *entropy* is maximal. In the literature, the principle of *maximum entropy* has been applied to problems that necessitate unsupervised learning for self-organization.

Here, the supervised learning is modified through an extra penalty term introduced into the cost function in order to control the complexity and the internal representation of the membership functions in an unsupervised way. This term is the Information and is introduced in the inverse sense of the maximum entropy principle, that is, it aims to minimize Information.

Information is minimized under the condition that the model can produce targets with appropriate accuracy, task guaranteed by the supervised part of the learning algorithm. This means that if the information concerning the training patterns is not stored at all, then learning is impossible.

The developed technique can be viewed as a "regularization" technique for improving generalization of a fuzzy model. Techniques having the same target have been applied to neural networks, i.e. weight decay, weight elimination etc.

Below, appropriate learning rules are derived for the FARX class of fuzzy models with Gaussian membership functions in the premises and autoregressive functions in the consequences.

Let's present initially what is defined as *entropy of a fuzzy set*. A membership function $m_j(x)$ represents the strength (certitude) that a given input x belongs to a certain class j. The most uncertain state is a state in which the hidden unit produces an activity close to 0.5. In this case it is impossible to tell whether the input pattern belongs to the class or not. In contrast, a membership value of 1 or 0 indicates with certainty whether the input belongs or not to the class. If H_i represents this uncertainty, it can be formulated as:

$$H_j = -m_j \log(m_j) - (1-m_j)\log(1-m_j) \tag{3}$$

The above relation is the definition of the entropy of a fuzzy set [3] and is directly inspired by the entropy of Shannon. As shown in Figure 2, the fuzzy entropy H_j is maximised for the most uncertain membership value of 0.5. Given the fuzzy entropy we can define information I_j as:

$$I_j = H_j^{\max} - H_j \tag{4}$$

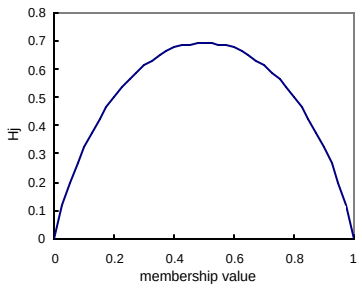


Figure 2: Entropy H_j of a fuzzy set as a function of the membership value $m_j(x)$.

The above definition can be generalised in order to define the Information of a fuzzy model. Let the model have m rules. To each rule, a local linear ARX model is associated composing the consequent part. The weight w^i of a rule represents the uncertainty with which the entire input pattern $\underline{x} = [x_1, x_2, \ldots, x_m]^T$ belongs to the associated membership functions of the premises. Since each membership function in the premise of a rule takes values between 0 and 1, the weight of a rule, which is the product of the membership functions, takes also values between 0 and 1. We can further proceed and define the entropy of a fuzzy model as a function of the normalised weight $\tilde{w}^i$ of a rule. By this way, not only the organisation of the membership functions within a single rule is taken into account, but also the self-organisation of the entire rule-base. In this case, it is:

$$u^i \triangleq \tilde{w}^i = \frac{w^i}{\sum_{i=1}^{m} w^i} \tag{5}$$

The observed fuzzy entropy E_N over the whole learning set N is defined as:

$$H_N \triangleq -\sum_{t=1}^{N}\sum_{i=1}^{m}\left[u_t^i \log u_t^i + (1-u_t^i)\log(1-u_t^i)\right] \tag{6}$$

It is noted that in the definition of u^i it is not included the output of a rule. This would be interesting in order to define mutual information between input and output. However, it is not possible since the values y^i are not strictly positive.

The maximum entropy is found easily to be: $H_N^{\max} = Nm\log 2$. Following the definition of the information of a fuzzy set, we can define the normalised total information content of a model as:

$$I_N^{pu} \hat{=} 1 - H_N / H_N^{\max} = 1 + \frac{1}{Nm\log 2}\sum_{t=1}^{N}\sum_{i=1}^{m}\left[u_t^i \log u_t^i + (1-u_t^i)\log(1-u_t^i)\right] \quad (7)$$

The objective of the learning process is to minimise the above normalised information together with the sum of prediction errors E_N. The parallel minimisation of the errors is necessary since it guarantees that the model learns the targets with accuracy. The following combined objective function is thus defined:

$$J_N \hat{=} E_N + a \cdot I_N^{pu} \quad (8)$$

where a is an appropriate weighting constant. In order to derive learning rules based on the stochastic gradient, the instantaneous value of the above function is minimised:

$$J(t) \hat{=} E(t) + a \cdot I^{pu}(t) \quad (9)$$

where :

$$I^{pu}(t) \hat{=} 1 - H(t)/H^{\max} = 1 + \frac{1}{m\log 2}\sum_{i=1}^{m}\left[u_t^i \log u_t^i + (1-u_t^i)\log(1-u_t^i)\right]$$

The following learning rules are derived according to the above cost function:

$$a_j^i(t+1) = a_j^i(t) + h_a \frac{2\left(x_j - a_j^i(t)\right)}{b_j^i(t)^2} \cdot \left[\tilde{w}^i\left(y^i - \hat{y}\right)e(t) - \frac{a}{m\log 2}u_t^i\left(1-u_t^i\right)\log\left(\frac{u_t^i}{1-u_t^i}\right)\right]$$

$$b_j^i(t+1) = b_j^i(t) + h_b \frac{2\left(x_j - a_j^i(t)\right)^2}{b_j^i(t)^3} \cdot \left[\tilde{w}^i\left(y^i - \hat{y}\right)e(t) - \frac{a}{m\log 2}u_t^i\left(1-u_t^i\right)\log\left(\frac{u_t^i}{1-u_t^i}\right)\right]$$

$$p_j^i(t+1) = p_j^i(t) + h_p \cdot \tilde{w}^i \cdot c_j \cdot e(t) \qquad c_j = \begin{cases} 1 & \text{if } j = 0 \\ x_j & \text{if } j > 0 \end{cases}$$

IV. ON-LINE MODULE IMPLEMENTATION.

The fuzzy-neural network forecasting model has been implemented into an on-line module within the More-Care EMS in order to produce predictions for a horizon of 48 hours ahead. The module is called by the EMS scheduler to update forecasts in an hourly (or 20 minutes) basis. The data management is performed by a relational database, which handles input, output, history and also the fuzzy rule base. ODBC functionality is used for the communication of the module with that database. Figure 3 shows the More-Care MMI and the display of the load forecasts.

A. Load preprocessing.

The robustness of a prediction model on corrupted input varies according to the type of model. It is expected that in any case, the quality of the results be affected. A preprocessing or load correction (LC) module has been developed to restore problems in the data and prepare the input for the load forecasting functions. Data restoring covers cases of erroneous and missing data values due to SCADA and communication problems (i.e. interruption of links to power stations) and also major power cuts and blackouts.

Normal variability of the load or minor power cuts are not considered. The LC-module also reconstructs historic input when this is not available (i.e. for several days); this enables the software to be operational after a period of stoppage, or when load data are missing for long periods.

The LC-module integrates several generic rules for on-line error detection. It then restores problematic data based on a number of correction rules. Thresholds for these rules are controllable from the database. This type of preprocessing was found to considerably improve the on-line performance of the load forecasting models.

V. CASE-STUDIES.

The cases of the islands of Crete in Greece and Madeira in Portugal have been studied.

Being isolated, the power system of Crete is a “weak” one. A characteristic of this system is high wind power penetration (67 MW installed) as well as very severe sea pollution during the autumn. The annual peak is at the level of 500 MW. The main transmission voltage is 150KV.

The installed thermal power capacity comprises 6 steam turbines of nominal power approx. 110 MW, 4 diesel units of 50 MW, 9 gas turbines of 260 MW and one combined cycle plant of nominal capacity 135 MW. These units are concentrated in two power stations close to the main load centers.

The learning and architecture optimisation procedures were based on hourly data of years 1998-1999. Data of 2000 were used to evaluate the performance of the model. Figure 4 shows the MAPE performance of the two cases. Due to the sliding window scheme, each time-step in the figure aggregates errors corresponding to different hours of the day and day of the week.

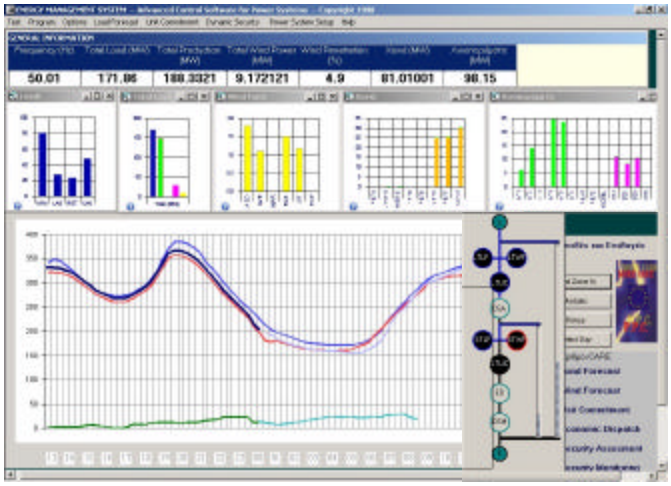


Figure 3: More-Care MMI with display of load and renewables forecasts.

The average performance for 1-24 hours ahead is 3.72% for the case of Crete and 2.98% for the case of Madeira. These values however are not directly comparable with error usually found in the literature for the case of interconnected systems. This is due to the sliding window scheme followed here but also because errors in the standard MAPE criterion are normalised with a "low" actual power (island system). A criterion based on relative variance of errors, taking into account the variance of the timeseries, would give more comparable results.

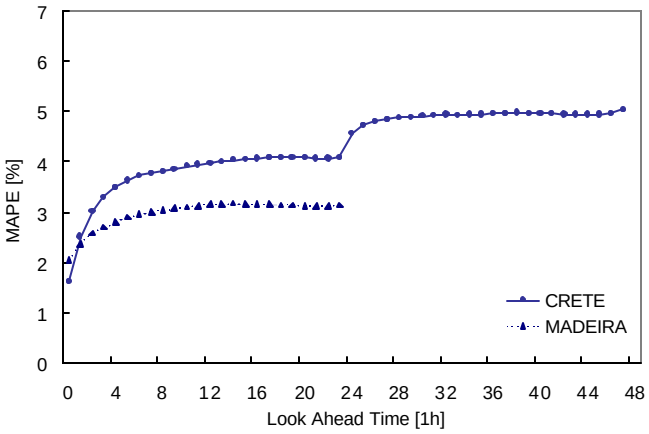


Figure 4: *MAPE performance of the load forecasting model for the cases of Crete and Madeira.*

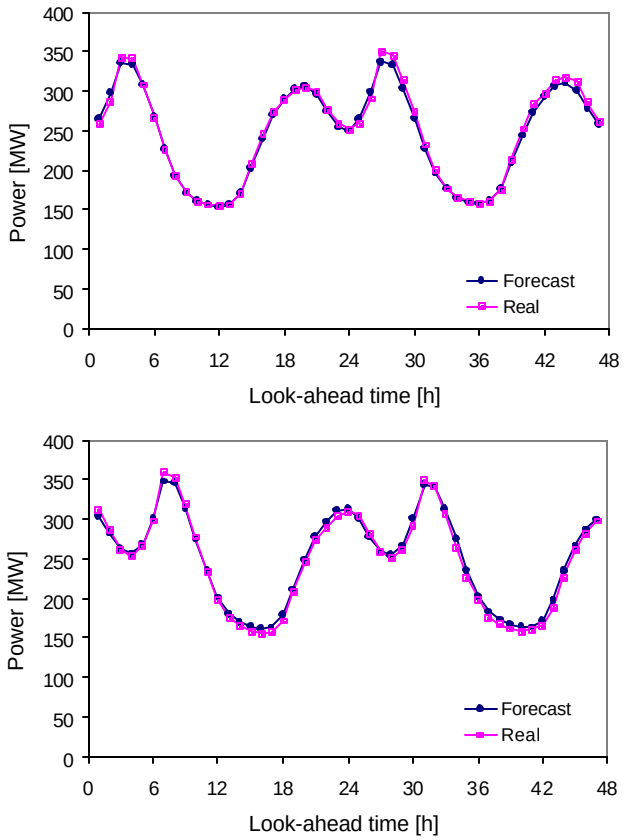


Figure 5: Case-study of Crete: Two sets of 48 hours ahead forecasts performed at two different time origins with a distance of 42 hours.

VI. CONCLUSIONS.

The paper presented an adaptive fuzzy-neural network model for short-term load forecasting. Emphasis was given on developing a procedure that provides model configurations with optimal generalisation capability. This is done through regularisation during learning and also through optimisation of the model architecture based on a non-linear constrained optimisation algorithm. The model gave satisfactory results for the case studies of Crete and Madeira, where it has been implemented for on-line use. Currently, the on-line operation of the load-forecasting module is under evaluation.

REFERENCES

[1] Douglas, A.P., et al, "Risk due to load forecast uncertainty in short-term power system planning", III Trans. Power Systems, Vol. 13, No 4, pp. 1493-1499, 1988.

[2] Hippert, H.S., Perdeira, C.E., Souza, R.C., "Neural Networks for Short-Term Load Forecasting: A Review and Evaluation", IEEE Trans. On Power Systems, Vol.16, No 1, February 2001, pp. 44-55.

[3] Deluca A., Termini S., "A definition of non-probabilistic entropy in the setting of fuzzy set theory", Inform. Control, 20 (1972), pp. 301-312.

[4] "Load and Wind Power Forecasting Methods for the Optimal Management of Isolated Power Systems with High Wind Penetration", A.G. Dutton, G Kariniotakis, J.A. Halliday, E. Nogaret, Wind Engineering, Vol. 23, No 2, 2000.

[5] "Contribution to the development of an advanced control system for the optimal management of autonomous wind-diesel systems." G. N. Kariniotakis, PhD thesis, Ecole Nationale Supérieure des Mines de Paris, Centre d'Energétique, December 1996.

[6] The "CARE" System Overview: Advanced Control Advice for Power Systems with Large-scale Integration of Renewable Energy Sources", N. Hatziargyriou et al, Wind Engineering, Vol. 23, No 2, 2000.

ACKNOWLEDGMENT

This work has been performed in the frame of the MORE-CARE project (ERK5-CT1999-00019) supported by the European Union. PPC and EEM are acknowledged for providing the data for the case study.

BIOGRAPHIES

George Kariniotakis was born in Athens, Greece in 1967. He received his production and management engineering and MSc degrees from the Technical University of Crete, Greece in 1990 and 1992 respectively and his PhD degree from Ecole des Mines de Paris, France in 1996. He is currently with the Center of Energy Studies of Ecole des Mines de Paris as a scientific project manager. He is a member of IEEE. His research interests include renewable energies, distributed generation and artificial intelligence.