SHORT-TERM LOAD FORECASTING FOR INDUSTRIAL CUSTOMERS USING FASART AND FASBACK NEURO-FUZZY SYSTEMS

Domingo A. Gundín
Dept. of Signal Theory, Comm. and Telem. Eng.
University of Valladolid, Spain
gundin@pireo.tel.uva.es

Celiano García
Iberdrola, S.A.
Valladolid, Spain
celiano.garcia@iberdrola.es

Eduardo Gómez, Yannis A. Dimitriadis, Guillermo Vega
Dept. of Signal Theory, Comm. and Telem. Eng.
University of Valladolid, Spain
edugom@tel.uva.es, yannis@tel.uva.es, vega@pireo.tel.uva.es

Abstract - This paper studies the problem of Short-Term Load Forecasting (STLF) for industrial customers. Since they have a large impact on power consumption and a particular load demand, an accurate forecast is specially important. For this task we study the application of two neuro-fuzzy systems, FasArt and FasBack, in addition to other techniques such as Multilayer Perceptron (MLP) with the backpropagation (BP) learning algorithm, as well as standard statistical Autoregressive Integrated Moving Average (ARIMA) processes. The experimental study is performed using real data provided by a major Spanish company. While the most accurate predictions are achieved similarly with FasBack and MLP, the former features easy knowledge extraction and on-line learning capabilities that make FasBack a better choice.

Keywords - Short-term load forecasting, industrial customers, neuro-fuzzy systems, FasArt, FasBack, rule extraction

1 INTRODUCTION

THE recent deregulation of the electrical power market for production and distribution in Spain, and other EU countries, has changed the scenario of the sector. Now electricity producers and suppliers are different companies, the latter buying power to producers in order to distribute it to final customers. An almost real-time power spot market has been established, so electricity has turned into a commodity to be traded at market prices [2]. A power supplier needs to have an estimation of its load demand in order to buy energy in the spot market. Errors in prediction will imply paying higher prices. In addition, the prediction has to be done one day ahead, and for each hour, every day. Therefore, the supply industry has an urgent need to forecast power consumption in order to be competitive.

Although most emphasis in literature has been found for aggregated power consumption [10], there is a special interest for short-term (STLF) with respect to specific large and medium-size industrial customers. The factors that affect the demand placed by a sole industrial customer are significantly different. Among them, the most relevant are production level, price scheme and vacations or days off.

Artificial Neural Networks (ANN) have been applied to several power systems problems, short-term forecasting being one of the most typical. Most proposed models are based on MLP networks [10]. Statistical models, like Box & Jenkins stochastic time series models [1], are also widely used in the literature (see [11] for example). Moreover, neuro-fuzzy systems are specially interesting for this problem, since they can provide interpretable rules from available data, while achieving satisfactory prediction results. Among these, FasArt and FasBack [4] are two Adaptive Resonance Theory (ART) [8] based neuro-fuzzy systems that have been applied to many engineering tasks, including system identification [3] and control of non-linear systems [7]. Their design allows them to perform well for non-linear, noisy, function identification [4]. Therefore, these models may be used for forecasting future values of load time series.

In addition to the use of these models for the actual prediction, the forecasting method deals with several other problems. First, available input variables must be detected. For industrial customers this is a tough issue, not only because the demand of different customers is affected by different variables, but specially because those related to their production process will not be delivered due to industrial privacy. We also perform a classification of customers, based on available information. Depending on this classification, recommendations can be made concerning the prediction technique, and the actual input variables to be used.

The rest of this paper is organized as follows: section 2 describe the proposed method for STLF, paying special attention to characteristics that make FasArt and FasBack systems suitable for this task, as well as how to handle industrial customers load properties; section 3 discusses experimental results obtained using real data provided by Iberdrola S.A., a major Spanish power company. Finally, section 4 presents the main conclusions and future research.

2 THE PROPOSED FORECASTING METHOD

In this section, our proposed prediction scheme is explained. First, we outline the differences between aggregated and industrial customers load. Next, properties of these customers are described, as well as the methodology followed to deal with their load. Finally, the properties of FasArt and FasBack are addressed, specially those that make them fit in the problem.

2.1 Why STLF for industrial customers?

STLF solutions proposed in literature are focused on aggregated load, i.e. the total load from all customers, including residential, commercial and medium and big-size industrial customers [5]. Meteorological conditions (temperature, humidity, etc.), hour of day, day of week and past load values are the factors most considered to affect the aggregated load. Piras et al. [13] identified 32 different types of input variables reported in the literature. However, a significant percentage of load demand is derived from medium and big-size customers, and factors affecting the load are significantly different. Production level, energy price and days off are the most relevant. On the contrary, temperature is rarely important, since most of the industrial energy consumption is due to production machinery, and heating is marginal. Because of its peculiarities and the large impact in total demand, some power suppliers would like to deal with industrial customers demand separately.

2.2 Properties of industrial customers load

In general, load curves presented by industrial customers can be modeled as a time series: current load demand depends on previous load values. Difficulties to apply this modeling arise from the fact that different customers may present totally different behaviors. In addition, the load curve is often noisy, non-linear and with strong seasonal behavior. Furthermore, the load is affected by exogenous variables such as days off and energy price.

The load curve, $x(t)$, $1 \leq t \leq N$, consists of hourly spaced consumption measures, being N the length of the data set.

Among the factors affecting the load, we can emphasize the presence of noise that could be due to diverse sources, such as system perturbations or uncertainties in the measurement instrumentation. Atypical values, normally called *outliers* [1], are also observed in the data. Moreover, the load curve may present several levels of seasonality, such as daily, weekly or longer (as it is commonly addressed in aggregated STLF [10]). In addition, the customer can perform a cyclical production scheme, and therefore the load may present other seasonal patterns.

The autocorrelation function of $x(t)$ is calculated to find *periodicities* in the load. Peaks found on this function give the period. If there are several periods, $x(t)$ can be differentiated s times, $z(t) = \nabla^s x(t) = x(t) - x(t-s)$, according to [1] to eliminate the first periodicity s, and the autocorrelation of $z(t)$ can be calculated to find out other periods. Normally a load curve presents at least one periodicity, for example weekly.

The *price of energy* is different depending of period of the day, week day, season, etc. Some customers increase their production (and consequently their power consumption) when the energy price decreases. However, other customers do not care about this, and there is no relationship between energy price and production for them. If $p(t)$, $1 \leq t \leq N$, denotes the energy price, correlation between $x(t)$ and $p(t)$ at lag 0 can be calculated as a measure of the dependence of load with its price.

In *days off*, the energy demand decreases close to zero. Knowledge in advance about these days is important for an accurate prediction, as we shall see later, but it often requires information from the customer, since days off do not restrict to national or regional vacations. Days off can be coded by $d(t)$, $1 \leq t \leq N$, using a binary codification.

2.3 Customer classification

As mentioned above, industrial customers heterogeneity is one of the most important problems of making an overall forecasting scheme. It is impossible to have a single criterion for selecting input variables for all customers. On the other hand, to determine the input variables most adequate for each customer can be an important burden on the operation in charge of the forecasting models, and we would like to simplify this work as much as possible. Thus, we would like to classify customers that are similar, so that the load prediction scheme for all of them uses the same input variables. In addition it is interesting to have as much knowledge as possible about customer behavior, relationships between different customers, and discriminate customers following their characteristics.

A customer classification is performed from their main features: sector (i.e. chemical, textile, etc.); size (related with their mean annual consumption in GWh); seasonality (found in load curve); load dependence on energy price, and days off (percentage of total). As explained above, temperature is nor relevant for industrial customers.

These features have been obtained by the operator experience and some heuristics. In addition, labels *high*, *medium* or *low* in periodicity or energy price dependence are suggested after experimental work from a large set of customers. Thresholds for these fuzzy sets were experimentally found using data from several customers of Iberdrola S.A. These included, but were not restricted to, the three customers studied in Section 3.1.

The factors that are considered for the classification are summarized in Table 1. According to them, four prototypes were selected from the customers studied so far. Then, given a new customer we should find to which prototype is closer, and start from the same input variables used by the prototype. Though these need not be the definitive input variables, having them simplifies the variable selection procedure. In section 3.1 we will show three different customers, each of a different type.

Besides being useful for grouping customers with similar behavior, and serving for input variable selection, the actual prediction model can be suggested by classification results, though this issue is not considered here.

Features	**Possible values**
Sector	Given by an expert
Size	$\{H, M, L\}$
Periodicity	Period and $\{H, M, L, I\}$
Dependence on EP	$\{H, M, L, I\}$
Days off	$\%$ of total

Table 1: Features considered for Customer Classification. EP: Energy Price. H: high. M: medium. L: low. I: insignificant

2.4 Input variable selection

A set of input variables must be made up to train the forecasting model. This is customer dependent, as we discussed in previous sections. Not only past values of load, $x(t - 2 \cdot 24)$ or earlier, energy price, $p(t)$, or days off, $d(t)$, can be considered, but also other related variables, such as *day-of-week code*, *hour-of-day code*, etc. could be included as *candidate* variables in order to improve the accuracy of the prediction. Thus, a set of candidate variables can be made made up, given the customer classification, as well as operator experience. Next, the most relevant ones could be identified by cross-validation techniques.

A periodicity is included in input variables provided that its classification rate is *medium* or *high*, and it is coded as usual [6]:

$$
\begin{aligned}
c_1^T(t) &= \sin(t \cdot 2\pi/T) \\
c_2^T(t) &= \cos(t \cdot 2\pi/T)
\end{aligned}
\qquad (1)
$$

where t is the hour from the beginning of the measurements, and T is the period in hours (i.e. 168 for a weekly periodicity).

In addition, past load values, $x(t-T), x(t-T-1)\ldots$, could also be included due to their high correlation with $x(t)$, but only if $T \geq 48$. This is because the prediction should be made one day ahead, but at that time only one day ago measurements are available. This could be seen as having to make the prediction *two* days ahead. If energy price dependence is rated with *medium* or *high*, $p(t)$ should also be included as an input variable. Similarly, if there are days off, $d(t)$ should be included. This is imperative if these days are a significant percentage of total, otherwise this can avoided.

2.5 Forecasting models

Neural networks have been widely applied to forecast problems, including STLF [10]. This is mostly because their learning capabilities allows them to build prediction models without *a priori* knowledge from the problem. Therefore, they fit as *black box* predictors in many forecasting schemes.

Among them, Multilayer Perceptrons (MLPs) [14, 9] are very popular because of their conceptual simplicity. They consist of several layers of processing units called neurons. Usually three layers are used: in the output layer one neuron outputs the forecasted variable; in the input layer, neurons are fed with input variables; an intermediate, or hidden layer links the other two. Each hidden neuron weights the signals coming from the input layer, produces a non-linear function of this weighted sum, and feeds it to the output layer. Output neurons generally produce a linear function of the weighted sum of signals from the hidden layer. Finding adequate weights is the task of training, and is usually carried out through a gradient descent algorithm, with backpropagation (BP) of the error [15] from the output to the hidden layers.

Though MLP+BP nets can theoretically approximate any function [9], they have several inconveniences. First, they pose a stability-plasticity dilemma [8], i.e. once they have been trained if they learn new patterns (plasticity) they forget previous knowledge (stability). Thus, they are not adequate to offer an adaptive solution. In addition, the knowledge contained in their weights cannot be expressed in human understandable terms. Adaptive Resonance Theory neural networks [8] solve this inconveniences. Among them, FasArt and FasBack [4] were proposed for function approximation.

FasArt is both a neural network and a fuzzy system, in both cases responding to an outer structure as that depicted in Figure 1. As a fuzzy system, it has a number of fuzzy rules with fuzzy antecedents (`IF` part), stored in the ARTa module, and fuzzy consequents (`THEN` part), stored in the ARTb module. The interART module links them in a many-to-one mapping, i.e. different antecedents may have the same consequent. An example of such fuzzy rule is shown in Figure 2. The rule is true if *both* antecedent components belong to their respective fuzzy set, as shown for input pair (x_1, x_2) in the figure. However, a fuzzy rule is not *either* true *or* not true; rather, it is true to some degree. In the case of Figure 2 this degree is given by $\eta_1 \cdot \eta_2$, where η is computed as shown in Figure 3. It is noteworthy that several rules can be true to some extent at the same time, and therefore participate to produce the output, which is computed as a weighted average of the `THEN` part of all active rules.

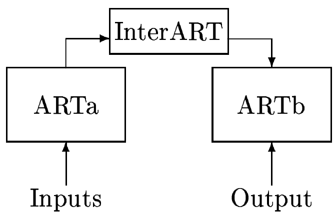

Figure 1: FasArt architecture, consisting of an ARTa module, where the antecedents of the fuzzy rules are stored, the ARTb module that holds the consequents, and the interART module.

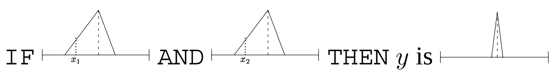

Figure 2: Example of fuzzy rule, as those managed by FasArt or FasBack.

As a neural network, FasArt constructs the fuzzy rules in the following way: each of the antecedents corresponds to a *fuzzy template*, stored in the network weights w_i, c_i and w_i^c, as shown in Figure 3. When a training vector is presented it is checked if it is similar to any of the existing templates, i.e. if the pattern belongs to any of the existing fuzzy sets. If so, the template is updated to reflect this input vector, by modifying weights w_i, c_i and w_i^c, which in turn changes the shape of the fuzzy antecedent. If the training pattern does not match any existing template, a new one is created. In addition, the correctness of the prediction is checked, i.e. the THEN part predicted by the active rule must match the real output presented during training, or otherwise a new rule is created. Through this process, FasArt decides both the number of fuzzy rules, and the shape of their antecedents, using only the available training data, and no *a priori* knowledge.

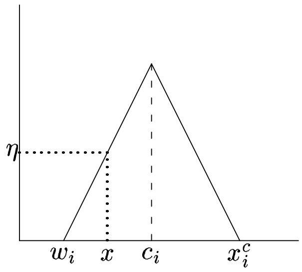

Figure 3: FasArt membership function for a fuzzy set i, and the associated weights, w_i, c_i and w_i^c. For an input pattern x, the membership degree to the fuzzy set is given by η.

FasBack is a modification of FasArt, that uses backpropagation of the errors to adapt the shape of the fuzzy antecedents once the rules have been created (i.e. adapt weights w_i, c_i and w_i^c), and thus enhance prediction accuracy. Both in FasArt and FasBack, the number of rules created can be somewhat controlled by user parameters ρ and γ, which balance the compromise between complexity and accuracy. However, because FasBack algorithm will itself improve accuracy, its ρ and γ parameters can be tuned so that it achieves similar accuracy to that of FasArt, but with smaller complexity, as we shall see in section 3.2.

More details, as well as step-by-step algorithmic description of FasArt and FasBack, can be found in [4].

Finally, it is worth mentioning that other classic approaches are also present in STLF literature. In particular, ARIMA models have been used to build forecasting schemes. Though this is a well known, reliable statistical technique, it has several disadvantages. First, since ARIMA models are univariate, they can only use past load values. To account for the type of day, $d(t)$, and the energy price, $p(t)$, a different approach has been used: since these variables are discrete, a different time series was built for each of their possible values, and then appropriate ARIMA models identified. Then, the prediction can be made with the adequate model at each moment. Alternatively, we could perform an intervention analysis [1] with similar purposes. However, this process is tedious and error prone. In addition, these type of techniques could be severely affected by the presence of outliers [1], that may have to be detected and removed in advance.

3 EXPERIMENTAL RESULTS

In this section, we validate and discuss the proposed forecasting method. A comparison among FasArt, FasBack, MLP+BP and statistical ARIMA models is drawn. Experimental data were provided by a major Spanish power supplier, Iberdrola S.A. Besides, an example of easy knowledge extraction by interpretable rules from proposed models is shown.

3.1 Case study

Pursuing our goal of validating the proposed forecasting method, three customers of Iberdrola S.A. in Spain were chosen among a large set of them. The load values were measured from April, 1 1999 to August, 31 2001 for all of them. Some information of these customers is hidden due to industrial privacy. They had a significantly different behavior, showing the heterogeneity of customers. In particular, the three of them were classified differently using the method described in section 2.3. However, some similarities can also be found among customers. *Customer A* is a concrete manufacturer with a high dependence on energy price, as well as high weekly and medium daily periodicities. Furthermore, the 7.92% of the days were days off. Thus, its load curve would be difficult to forecast. *Customer B* is a car tire manufacturer with a bigger mean annual consumption also rated as high. The influence of energy price in load curve was rated as low and the only significant factor affecting the load was a medium weekly periodicity. Finally, *Customer C* is a wood products manufacturer. Mean annual consumption was rated as medium and the load curve showed a medium daily periodicity. The 13.35% of the days corresponded to days off. Significant characteristics of their classification are summarized in Table 2.

C#	DO	DP	WP	DEP	Size
A	7.92%	M	H	H	H
B	4.86%	L	M	L	H
C	13.35%	M	L	L	M

Table 2: Classification of three industrial customers addressed in the experimental work. C#: Customer id, DO: Percentage of days off, DP: Daily periodicity, WP: Weekly periodicity, DEP: Dependence on Energy Price

All available data from each customer ($21,216$ hourly load measures) were divided into two equal parts. Each set was used to train the ANN models, then they were interchanged to test the performance of models (i.e. data set 1 was used to test the model that had been trained with data set 2, and viceversa). Furthermore, training pairs were randomly ordered before presenting them to the ANN. The final forecast accuracy was obtained as a simple average of the results from those two test sets. This approach, *cross validation*, ensure certain data set independence [12] and can avoid overtraining [10]. Due to the difficulty to derive ARIMA models, they were only identified with the first part, and they were tested using the second part.

Input variables were identified following strictly the proposed customer classification. Since each customer had a different classification, different sets of input variables were used for training each ANN model. Other input variables, such as *hour-of-day code*, *day-of-week code*, etc. or lagged values of $x(t-2\cdot 24)$ or earlier could have also been taken into account, but our goal is to validate the whole proposed method instead of having a little more accuracy.

Besides selecting input variables, we have to tune the user parameters of the ANN models. This selection can be done for each customer in order to improve accuracy, though a comparison among results may be somewhat imprecise. On the other hand, if the same parameters were used for all of them, the results could be compared for each model and customer more easily. In this paper, we followed the first approach, since it corresponds to a more realistic prediction environment.

With respect to ARIMA models, we followed the standard procedure for their identification. First the model must be identified by a statistical analysis of data, and its parameters estimated. Next, this model has to be validated by residual analysis [1]. Finally, the actual prediction is obtained from this identified model. Since these models can only deal with univariate data, we need to follow more complex approaches to account for variables such as dependence on energy price or days off, as explained in section 2.5.

In order to evaluate the accuracy of forecasting models, two widely accepted quantitative measures, such as RMSE (Root Mean Square Error) and RRMSE (Relative RMSE), have been used. The former penalize large individual errors with respect to many small errors. Large errors may have worse consequences for the power supplier, making RMSE an important figure of merit. RRMSE gives a relative measure of accuracy. Furthermore, the number of rules generated by FasArt and FasBack is also shown, as it expresses model complexity.

All simulations were run under MATLAB 6.0 for Windows on a AMD K7 1.4 GHz computer with 512MB of RAM. The MLP+BP model was developed using Matlab Neural Network Toolbox 4.0, and FasArt and FasBack were programmed in C by the authors. On average, training and testing took 2.5s for FasArt, 38.2s for FasBack, and 118.2s for the MLP, for 50 training iterations.

3.2 Results

As already mentioned in section 3.1, *customer A* load demand seems *a priori* difficult to forecast. This is mostly due to a marked dependence on the energy price and type of day, as well as the combination of daily and weekly periodicities. In fact, the least accurate predictions are obtained for this customer, as shown in Table 4. In addition, because the customer behavior is more complex, involving more variables, many rules are needed to describe such behavior with FasArt and FasBack. This number is extremely high in FasArt, spoiling by itself the explainability of the knowledge acquired. However, the number of rules is far more reasonable in FasBack, constituting a rule set that is human understandable, and that explains the customer behavior. We will illustrate this in section 3.3. The difference in the number of rules produced by the networks stems from the backpropagation learning used by FasBack. It refines the rules, making them more accurate. Therefore, user parameters balancing complexity and accuracy can be set to wards *less complexity*. Moreover, because rules created in the first training iteration will be accurate, no other rules will be created afterwords even if training has proceeded for 50 iterations.

C#	Input variables
A	$x(t-2\cdot 24), x(t-7\cdot 24), d(t), p(t),$ $c_1^{168}(t), c_2^{168}(t), c_1^{24}(t), c_2^{24}(t)$
B	$x(t-7\cdot 24), d(t), c_1^{168}(t), c_2^{168}(t)$
C	$x(t-2\cdot 24), x(t-7\cdot 24), d(t), c_1^{24}(t), c_2^{24}(t)$

Table 3: Input variables selected for each customer of Table 2, in order to train ANN models. C#: Customer id

With respect to the other forecast approaches, it can be seen that MLPs can achieve similar accuracy. However, the knowledge it acquires cannot be explained, as opposed to the neuro-fuzzy systems. Furthermore, the time to train the MLP for 50 iterations was much longer that theirs (more than three times FasBack's). On the contrary, to build the ARIMA model we did not need much computation, but rather human effort and heuristic decisions to determine the model parameters. In addition, its best performance is worse than that achieved by any of the neural models. This can be due to the importance of days off, that introduces irregularities in the time series, and the presence of *outliers* in the data.

Customer B presents a more uniform behavior. Therefore, fewer input variables are needed to achieve a satisfactory prediction of load demand, as shown in Table 3. Because of this, all models achieve better performance than with customer A. This is specially noticeable for ARIMA models, because customer B is much less affected by days off. Also because of its relative simplicity, FasBack can offer a compact description of the customer behavior, with 48 fuzzy rules, that in addition have less antecedents. Though its performance is comparable to that of the MLP and FasArt, none of the latter can be interpreted, the MLP because of its construction, and FasArt because of the excessive number of rules.

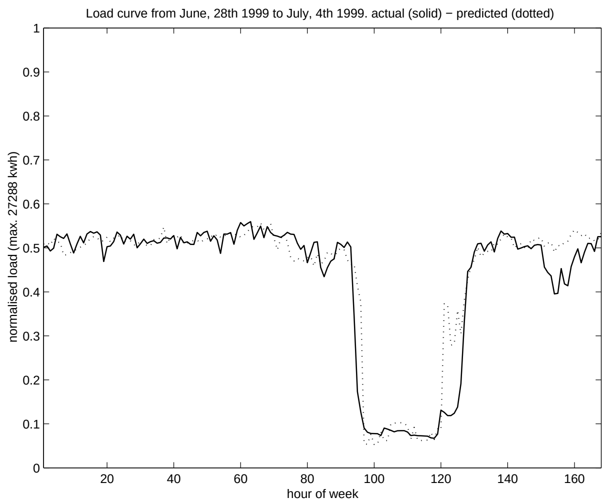

Figure 4: Actual load curve and FasBack forecast for *customer C*. Normalized to maximum from Apr. 1999 to Aug. 2001

Finally, *customer C* is an intermediate case. It has daily periodicity rated as *medium*, but most important, it has a large amount of days off. Therefore, the relation to variable $d(t)$ should be critical. This fact strongly penalizes the ARIMA model, that cannot account for this dependence satisfactorily. For the other systems, the above comments also hold for this customer. It is worth adding that the error achieved by all systems is smaller than for customer A, because the mixed periodicities present in that customer, that poses a difficulty for the networks to find the relations between input and output variables.

C#	Results	FasArt	FasBack	MLP	ARIMA
A	RMSE	0.1515	0.1362	0.1356	0.1474
	RRMSE	0.2025	0.1865	0.1858	0.2143
	Rules	4308	99	–	–
B	RMSE	0.0584	0.0583	0.0587	0.0927
	RRMSE	0.1142	0.1140	0.1148	0.1280
	Rules	1176	48	–	–
C	RMSE	0.0923	0.0871	0.0863	0.1350
	RRMSE	0.1290	0.1224	0.1212	0.1983
	Rules	1192	108	–	–

Table 4: Results for the customers under study with ANN and ARIMA models

Figure 4 shows a piece of the actual and predicted load curve for customer B, using FasBack as a forecasting model. It shows that the trend of the demand is well tracked, though in the low level the prediction is not very accurate. This is reasonable to expect, since minor variations on the energy demand are unpredictable. In fact, the predicted curve is much smoother than the actual. This is convenient, since tracking small variations would require a more complex network, with many more rules, and besides it would not actually contribute to the accuracy of the prediction, as already discussed.

3.3 *Rule extraction from FasBack*

In previous section we elaborated on the fact that FasBack can produce a reduced number of rules after the training, that can be easily understood. In Figure 5 we show three of these rules for customer C, with their confidence rate (between 0 and 1). Corresponding to the input variables, the rules state their antecedents on the type of day, two days and one week ago load values and time within the day. The type of day can only be *normal* or *day off*. The past load values are numbers, that can have a degree of membership to the triangular fuzzy sets depicted in the figure. For example, in rule #1, $x(t-2\cdot 24)$ activates the rule maximally if it is around 0.8. The hour of day antecedents are depicted in a circle, in which hours start at 0 at the right side, and go anticlockwise to end in hour 24 at the same point. For example, this antecedent for rule #55 can be read as *from 1am to 6am*.

Rule #1 is a normal rule that says that in a normal day the forecast should basically follow the load demanded two days ago. In plain English, rule #1 could be read as *for a normal day, whatever time of the day, if the load two days ago was high, and also a week ago, then predict high*. Note that this rule is also active, but not as much, if the load two days ago is not so high.

Rule #55 a more specific rule, also for normal days. It is active if the time of the day is in the first quarter. Since the period starts with the day, it corresponds to night hours. In words, it says that *for a normal day, but during the night hours, and especially if the load two days ago was medium and a week ago was low, predict low*. As this company has not a dependence on energy price, it is reasonable to expect that it will consume less during the night. Note, however, that this rule confidence is not very high. This is because other nights there is high demand. Therefore, if this rule is active but also some other that predicts high demand, the prediction of this rule will account little to the total.

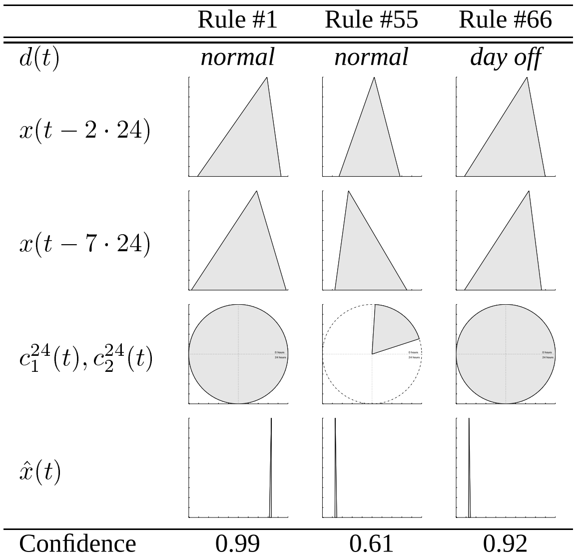

Figure 5: Rules extracted from FasBack, after training to forecast customer C load

Finally, rule #66 is a typical rule for days off. It states that *if the day is a day off, no matter the hour of the day, and though two days or a week ago the demand was high, predict low demand*. This is reasonable to expect, since in days off the work in the company will be reduced, and so the demand.

To see the performance of FasBack in the test stage, let us assume that only these three rules constitute the knowledge base (not true). It we had to forecast the load for a day off, only rule # 66 would be active, and its prediction definitive. If, for a normal day, the hour of day was around midday, only rule #1 would serve to compute the prediction. However, if for a normal day the hour was around 3am, and the demand two days ago and seven days ago was medium, rule #55 would be too active, and also rule #1, though not as much. However, since rule #55 has less confidence, both rules could probably contribute to the prediction similarly. Thus, the predicted demand would be medium, or slightly high.

4 CONCLUSIONS AND FUTURE WORK

This paper has proposed a STLF scheme that uses either neuro-fuzzy, neural or statistical models. The importance of this study stems from the recent deregulation of the power market in Spain, in which forecasting the demand becomes a critical issue for supplier companies. Moreover, the forecast of the demand placed solely by large industrial customers has been pointed out as a need by many of these suppliers.

Large industrial customers hide some information that could be useful for the forecast, and therefore it must be done with few input variables, most significantly previous load values. To avoid the burden of identifying input variables for each customer, we proposed the customers be classified into customer prototypes, and then select a common set of candidate variables for each prototype. This may result into poorer prediction, but will ease the extension of these techniques to a broader number of customers.

Once the customer was classified, a prediction model could be derived. Though properties of statistical models are very well known, building one such model usually requires much human intervention, specially if input variables do not restrict to past load values. On the contrary, we deal with neural networks as almost *black boxes*. This is useful to extend the forecast to more customers.

Though we found experimentally that two of this neural systems, namely MLP and FasBack, achieved better performance than others, we showed that FasBack can also be seen as a fuzzy system. As a consequence, the knowledge it extracted from the data can be expressed into a rather compact set of fuzzy rules. This rules can be used to understand more thoroughly the behavior of the customer, as well as to explain how the forecasted load has been calculated. This property makes FasBack a suitable technique to embed into the forecasting scheme.

Current research on the application side looks for validating these results on a larger number of customers, as well as considering other possible input variables. On the neural network side, we look for improving FasArt or FasBack to achieve similar performance but with more reduced a set of rules.

ACKNOWLEDGMENTS

This research work was partially supported by the 'Educational Cooperative Program' between Iberdrola, S.A. and the University of Valladolid. The authors would like to thank all researchers of the 'Cooperative & Intelligent Systems Research Group' at the University of Valladolid, and specially R. Sacristán Martínez.

REFERENCES

[1] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA, USA, 1976.

[2] D. W. Bunn. Forecasting loads and prices in competitive power markets. *Proceedings of the IEEE*, 88(2):163–169, 2000.

[3] J.M. Cano, Y.A. Dimitriadis, M. Araúzo, and J. López. FasArt: A new neuro-fuzzy architecture for incremental learning in system identification. In *Proceedings of the IFAC World Congress*, volume F, pages 133–138, San Francisco, USA, June 1996.

[4] J.M. Cano, Y.A. Dimitriadis, E. Gómez, and J. López. Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, 14(4–5):407–425, May 2001.

[5] W. Charyroniuk, E. D. Box, W. J. Lee, M. S. Chen, P. Kotas, and P. Van Olinda. Neural networks based demand forecasting in deregulated environment. In *Proceedings of IEEE Industrial & Commercial Power Systems Technical Conference*, Sparks, NV, USA, May 1999.

[6] S Drezga and S. Rahman. Input variable selection for ANN-based short-term load forecasting. *IEEE Transactions on Power Systems*, 13(4):1238–1244, November 1998.

[7] E. Gómez, J. M. Cano, M. J. Araúzo, Y. A. Dimitriadis, and J. López. Adaptive IMC using fuzzy neural networks for the control of non-linear systems. In N. Mårtensson, R. Mackay, and S. Björgvinsson, editors, *Changing the Ways We Work*, pages 792–801. IOS Press, Amsterdam, 1998.

[8] S. Grossberg. *Studies of mind and brain: neural principles of learning, perception, develpment cognition, and motor control*. Reidel Press, Boston, MA, EEUU, 1982.

[9] S. Haykin. *Neural Networks: A comprensive foundation*. McMillan College Publ. Co., New York, USA, 1994.

[10] H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, February 2001.

[11] G. Juberías, R. Yunta, J. García, and C. Mendivil. A new ARIMA model for hourly load forecasting. In *Proceedings of IEEE Transmission and Distribution Conference*, volume 1, pages 314–319, 1999.

[12] A. Koufakou, M. Georgiopoulus, G. Anagnastopoulus, and T. Kasparis. Cross-validation in Fuzzy ARTMAP for large databases. *Neural Networks*, 14(9):1279–1292, November 2001.

[13] A. Piras, A. Germond, B. Buchenel, K. Imhof, and Y. Jaccard. Heterogeneous artificial neural networks for short term electrical load forecasting. *IEEE Transactions on Power Systems*, 11(1):397–402, February 1996.

[14] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[15] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. *Parallel distributed processing: explorations in microstructure of cognition*, volume 1-2. Bradford Books, Cambridge, MA, EEUU, 1986.