Standardization of Short-Term Load Forecasting Models

M. López [#1], S. Valero[#2], *Member IEEE*, C. Senabre[#], J. Aparicio[*], A. Gabaldon[+], *Member IEEE*

Universidad Miguel Hernández de Elche (UMH)

[#]*Dpto. Ingeniería de Sistemas Industriales. Área de Ing. Eléctrica*

[*]*Centro de Investigación Operativa*

Avd. de la Universidad s/n, 03202, Elche, Spain.

[+]*Universidad Politécnica de Cartagena (UPCT)*

[1] lopezgarcia.m.81@gmail.com

[2] svalero@umh.es

Abstract— **There has been a significant production of load forecasting models over the last 5 years. These models present a wide variety of techniques, most of them using novel artificial intelligence approaches. Load forecasting is a complex matter and it is the result of several processes that, depending on the database, may be of more or less importance. However, most models focus their attention only on one process like the "forecasting engine", neglecting other processes like variable selection or pre-processing. This paper proposes a standard scheme for load forecasting models that includes all sub-processes within load forecasting. The analysis of load forecasting models through this scheme allows identifying the effect of each process on the overall performance of the model. Also, proposing load forecasting models following this scheme will enhance benchmarking possibilities and hybridization of models. Finally, this paper presents such analysis of an actual load forecasting model.**

I. Introduction

Electric load forecasting has received considerable attention over the last 5 years. The number of published articles in scientific journals regarding this topic has increased each year, and the liberalization process of the energy markets of countries all around the world makes it probable for this number to keep increasing or at least to remain constant.

There has been a significant evolution of the complexity of the models proposed over time. Early models were basically a statistical analysis of the auto-correlation of the load series; some of them also included correlation with other external variables (mostly meteorological) [1]. The introduction of artificial intelligence techniques has mostly wiped out these "classical" statistical models and a wide variety of neural network based models have emerged [2]. In addition, models based on fuzzy logic, application of wavelet transforms and other novel and hybrid approaches appear on current publications.

The attention drawn by this matter should have caused a significant improvement of the average performance of the proposed methods or, at least, provide with numerous options to select from for a specific application. However, this conclusion is not as trivial as it should be due to the lack of standardization in data bases, internal procedures and performance reports.

Most of the published models pay deep attention to the "forecasting engine", meaning that the main focus of their work is the forecasting process rather than other processes that may reveal to be also very relevant to the overall performance of the model like data pre-processing, variable selection or training data selection. The adoption of a standard model in which all relevant stages of forecasting are present would provide authors with a structure in which it is easier to address each of the processes and, more importantly, it is possible to isolate the effect that each process has in performance, thus enabling the users of the models to perform actual evaluations, hybridization and benchmarking.

In addition, the expression of the performance with a single precision indicator (mostly Mean Absolute Percentage Error), may be misleading and can be improved by using other indicators regarding different aspects of forecasting performance like variability, maximum error, etc. The use of a more complete performance report may be used as a selection guide to help choosing the most suitable model for a specific application.

Finally, as it was mentioned before, there are many published models; nevertheless, their use of different data bases makes it really difficult to actually compare the result of two different models. While some data bases may be highly

This research was financed by the Research Project with reference GV/2010/080-Valencian Government, Department of Education, and directed by professor Ph.D Sergio Valero Verdú. Also, this work is part of the research proposal by professor S. Valero and the Ph.D student Miguel López García awarded the "I Ayudas a la Investigación" 2011 granted by the Iberdrola Foundation.

S. Valero, M. López and C. Senabre are with Universidad Miguel Hernández de Elche, 03202 Elche, Spain (e-mail: svalero@umh.es, granmicky@hotmail.com).

predictable (low variability on consumer behavior, large influence of other known variables…), other may be more difficult to forecast, voiding the comparative value of the forecasting performance of two models applied to different data bases.

This paper presents a standard forecasting model structure to address the issues exposed above. As a result of a thorough revision of the literature, section 2 shows the most significant and frequent processes in load forecasting. Section 3 proposes a standard scheme that includes all these processes. Finally, in section 4, an actual load forecasting model is analyzed through the standard scheme showing the effect that each process has on the performance of the model

II. Approaches to Load Forecasting

As it was aforementioned, there have been proposed many load forecasting models in the last 10 years. In this section we provide an overview of the components of a load forecasting model, providing examples from actual published works. The inclusion of a specific model in this review does not rely on the performance of the model but on the fact that one or more of the processes included in the model are either innovative or representative of a popular technique. The review spans from the early data mining processes to the final performance evaluation.

The process of data gathering is seldom explained in the literature, maybe because it is considered that it only applies to one particular data base. However, it is important to describe the way raw data is treated before is considered a database because the raw data may include missing or inconsistent values that may be interesting to either discard [3] [4] or reconstruct [5], [4] them. If so, these treatments need to be described and reported.

Other treatments are also common like normalization or decomposition. Normalization of the data may have two goals: provide the forecasting engines with variables within the same range [6], [7] and to make the series stationary [4]. Series decomposition consists in the application of some sort of filter or transform that splits the series into 2 or more sub-series, each of them having specific features. Then, different models are used to forecast each sub-series. The most common decomposition technique is wavelet transform [5]-[9].

Once the database is complete, it is common to label each data set, usually by assigning a tag to each day in the database. The model described in [6] classifies days into weekdays and weekends, and then uses two separate models for each class. The classification process may include more categories, like the day of the week [4] or even a more complex system including weather variables or actual load [8], [10]. Also, the use of this classification may not be used to separate data into different models but to be used as an input of a single forecasting model [8] [10]. Finally, even if a classification process *per se* is not present in the model, the analysis of the results is usually categorized to express performance under different conditions [9].

Input selection is generally well documented. It describes the process followed to determine which variables should be fed as input to the forecasting engine. Some models simply use a specific pattern of data (both meteorological and load) known at the time at the forecast as input for the forecast engine [4], [6], [8], [10]. This type of input selection is based on trial an error and the authors report it as the optimum pattern. Other models proposed a more complex system in which the optimum input is obtained through an algorithm [7], [9]. These systems are based on a preliminary analysis of the load series that determines which variables are more closely correlated to the desired output. The advantage of the latter is that it is better suited to other databases than the former.

The input defines which data are used as a single data point. This probably includes values of loads from previous time (last 24-hours, last week's load profile…). However, despite the chronological nature of input selection, it should not be confused with the selection of the training period. The training period defines which data points (known data sets from the past) are used when training or adjusting the forecasting engine.

The information obtained from classifying the data is usually put to use in the training period selection. Many models use only days of the same type as training data, independently of how these days were classified [8], [9]. Depending of the complexity of the classification process this selection may be adequate. However, other models include methods to exclude days from different seasons of the year [4], [6], [7], [10] and / or that limit the amount of years in the past to get historical data from [4].

The term "forecasting engine" has been previously referred to in this document as the process in charge of actually producing a forecast. This is the stage of load forecasting that most models focus on the most and the one that makes the most significant difference on how the model is structured. However, despite the difference that may appear in the surface, the various techniques described in the literature are just different solutions of the same problem.

Modern models describe forecasting engines based on neural networks or other type of artificial intelligence. There are two basic design decisions to assemble a functioning neural network: topology and training algorithms. The former depends basically on the input and output vectors and the result of several trial and error tests, while the latter allows for a wide diversity of possible implementations: back propagation [10], evolutionary algorithms [7], particle swarm organization [6]… More classical models applying regression approximations use least square schemes to obtain the model's coefficients [9].

As a new trend, some models propose hybrid techniques in which the forecasting engine is tuned using two or more steps combining different methods like the Levenberg-Marquad algorithm and evolutionary algorithm [7].

A common problem to all artificial intelligence based methods is over-fitting. Load forecasting models should implement training stopping techniques to solve this problem. The technique used may be as simple as setting a maximum of training iterations based on experience [4] or more complex like using separate data sets for some sort of cross-validation [7].

The output of the model defines the topology of the neural network. Some models propose forecasting engines that forecast one hour at a time and that need of 24 consecutive iterations to obtain a daily curve. The output of each iteration may be considered as an input for the next one [] although it is also possible that each hour is forecasted by independent models. Nevertheless, it is also possible to find models that produce daily load profiles directly.

Finally the performance analysis is usually based on the mean average percentage error (MAPE). This value is a good estimator of the model's accuracy but it is only valid if the forecasted period is long enough to make the result significant and if such period of time includes all the different conditions in which the model can be applied (weekdays, weekends, summer, winter, holidays…)

This brief overview of the stages of load forecasting based on some actual examples set the foundations of the standard model proposed in the next section.

III. Standard Model

Even though, the published works in load forecasting do not follow a specific standard, it is possible to infer from them a certain common series of processes. As it was referred in section 2, some models are more detailed in specific processes while others focus more on others. However, it is possible to outline a standard workflow that all forecasting models follow. Fig. 1 shows this standard model as a flowchart depicting all five stages of load forecasting: data pre-processing, variable selection, training data selection, forecasting and data post-processing.

These five stages are present in all forecasting models, at least to some extent; but within them it is possible to find specific process that may or may not be used by each model. In fact, each model is characterized by a different implementation and combination of these processes.

Each and every stage is responsible for the failure or success of the forecasting model and none can be overlooked. In the following sub-sections each stage will be described in terms of the desired input and output for each one and the different process to be found within them.

A. Data Pre-processing

The input for this stage is a database of raw data including historical load and any other exogenous variables. The format for this database will depend on the actual application. It is important to understand that the data presented as input may contain errors or missing values occurred from mistakes in the data mining process. Also, planned events like daylight saving time may provoke singularities on the data that need to be addressed.

The processes to be found in this stage may be described as follows:

- Filtering: The aim of filtering data is to eliminate abnormalities. Ideally, any inconsistency found in the data (missing values, sudden abnormal changes,…) will be reconstructed by extrapolation or other techniques. However, if the damaged data is irretrievable then the effect on other stages can be contained by identifying these corrupted data.
- Normalizing: Training algorithms may become unreliable if absolute values of the different variables are not comparable. Normalization of variables within a given range is a mandatory procedure to avoid this issue. Also, the raw data may contain long-term trends that disable the possibility of comparing newer to older data. Normalization using moving averages is a successful technique to solve this problem.

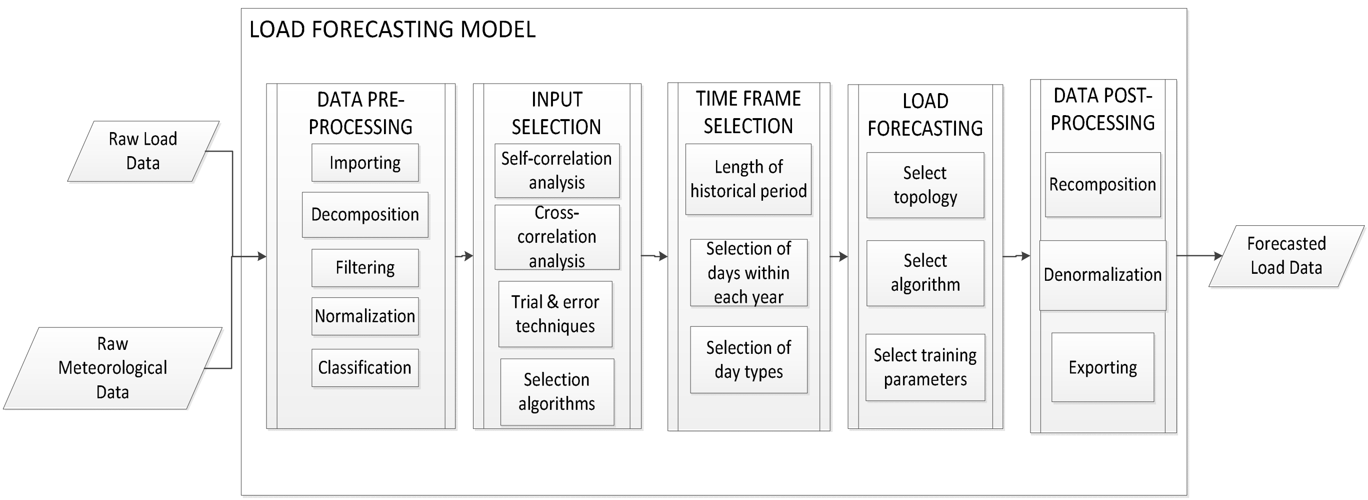

Fig.1. Structure of the proposed forecasting model.

- Classifying: Databases should not only contain the actual raw data, but also some sort of labeling system that adds more information. Whether that load profile belongs to a holiday, a regular Monday or any other made up category is information that becomes useful at later stages.
- Decomposing: Transform techniques like Fourier or Wavelets allow decomposing the series into separate series each of them containing different information. These separate series may be forecasted separately, allowing the use of differently configured forecasting models, each of them optimized to forecast a specific sub-series.

The output for the next stage should comprise the enhanced data from the input, re-arranged in the desired format and expanded with new data added.

B. *Variable Selection*

There are many factors that affect future load, however, the effect they have varies from one region to another and it may even vary with time. The input for the variable selection stage is a series of vectors properly formatted in the data pre-processing stage. The aim of variable selection is to eliminate the components of these vectors that are not relevant while keeping the ones that actually improve the forecast.

It is not possible to *a priori* know which variables are better for a forecast. However, there are some processes that may result useful finding the optimum combination:

- Self-correlation analysis: Load series have high self-correlation coefficients. This means that the next value of the series is highly dependent on previous values. However, it is important to know which previous values are more relevant in this dependency. Self-correlation analysis provides a way to determine such previous values which are a good estimator for the input variables from the own load series.
- Cross-correlation analysis: As in self-correlation, cross-correlation analysis identifies which previous values from a different time series the next load value depends on. Again, higher cross-correlation coefficients point out the best candidates as input variables.
- Trial and error: Even though analytical techniques may yield a good result, it is advisable to develop processes that can improve the selection of variables by trial and error methods.

The output of this stage is a database that includes only the relevant information for the desired forecast.

C. *Training Period Selection*

In order to obtain an accurate forecast, not only it is necessary to know the right variables, but also to learn from the proper examples. The training period selection stage selects the data points that will teach the forecasting engine the desired behavior. Therefore, it is important to select as training data the data points that show the behavior the forecasted data is likely to reproduce.

This stage may appear similar to variable selection but, as it was aforementioned, training period selection refers not to which variables are more relevant but to which past time periods are more similar to the period to be forecasted.

The process is linked with the classification made in the pre-processing stage as, usually, only days of the same category will be used in training. Nevertheless, classification is not the only valid approach:

- Selection of days from similar time of the year, both current year and previous ones.
- Selection of days based on similarities: even if classification was not used in pre-processing it may appear in this stage as a process of finding which past days are similar to the known information of the forecasted day.

The output of this stage is a group of data points that are valid to training the forecasting engine.

D. *Forecasting*

Forecasting stage comprises the processes needed to actually produce a forecast. The input is a training database containing enough datasets of all relevant variables needed to perform the training of the forecasting engine. This training will allow the engine to learn the expected behavior.

There are numerous forms that a forecasting engine can take. The two main groups are probably ARIMA models and artificial intelligence models. Even though these two types seem different, in terms of input and output they are equivalent. The difference, of course, lies in the nature of the processes needed to obtain such output. Some of the following processes are mutually exclusive but it is still worthy to provide a tentative list of tasks needed to set up a forecasting engine:

- Select engine type: Whether a type of ARIMA model, a neural network, a self-organizing map or a fuzzy model will be used.
- Select topology: Sometimes the number of parameters, fuzzy rules neurons or layers is determined by the amount of inputs or other previously defined settings. Nevertheless, there are cases in which is necessary to select the rest of parameters.
- Select training algorithm: Both regression models and artificial intelligence ones need a technique to estimate parameters or neuron weights.
- Select training duration: It is necessary to define at what point the training of the engine will stop to avoid over-fitting.

The output of this stage is the actual forecast. Needless to say, depending on the data pre-processing, the forecast may not be the final desired result and some post-processing is required.

E. Data Post-processing

Data post-processing includes both constructing a forecast in an understandable format from the one produced by the forecasting engine and reporting the performance of the forecast.

The processes that accomplish the first part are the inverse of the ones implemented in the pre-processing stage: renormalizing, recomposing, etc. Regarding performance reporting, these are several measures of performance that provide a good understanding of how good a forecasting model actually is:

- Accuracy: The most common measure for accuracy is MAPE. It is important to report it over significant time period.
- Robustness: Besides a good average performance, it is important to determine if the model ever produces catastrophic errors (APE above a certain threshold) and, if so, how often this happens.
- Real-time application: Assess whether the model is able to produce forecasts in a timely manner and if it is able to learn as new information becomes available.
- Universality: The result of a forecasting model depends highly on the database. The same model may perform well under a certain circumstances but poorly under different ones. Therefore, it is not significant to provide one result. It is advised to provide several test results from different databases or, even better, a study of the characteristics of the database used.

IV. Analysis and Performance Report

In this section, an actual model will be analyzed following the proposed standard scheme. The aim of this analysis is to identify how the actual model implements the different stages in order to highlight improvement opportunities in poorly developed processes.

In addition, the purpose of this paper is to identify the effect that each stage has on the final performance of the model. To this end, slight variations of the model will be tested, each one eliminating or simplifying the processes implemented in the model, creating a "dummy" stage to whom the contribution of the model's stage can be compared.

The model analyzed is described in detail in [4]. It is beyond our reach to validate the design of such model or to include its complete definition in this paper. The processes defined in [4] will be referred to but only superficially described:

A. Data pre-processing.

The database used comprises historical load and meteorological data. The processes implemented in this stage are filtering, normalizing and classifying. There is no decomposition of the series.

In order to analyze the implementation of this stage, four tests have been run. Their results are shown in table 1:

- Elimination of the filtering: All abnormalities of the data are kept and every day is forecasted and evaluated. The filter that reconstructed missing data was not removed to avoid computational problems.
- Increase of filtering level: The amount sudden variation considered an abnormality is reduced by half. The number of valid days is therefore reduced.
- Reduction of types of days: Only workday and weekend is considered instead of all 7 days of the week.
- Elimination of normalization: Data are introduced to the forecasting engine as raw data, without any normalization.

B. Variable selection

The variable selection process uses trial and error to rule out the use of meteorological variables. Also, it ran some tests to determine that the proper input was the load profile of the day before.

The simplification of this stage comes from using as input a common combination of temperature and previous day load. The results are shown in table 2.

C. Training period selection

The selection of the training data points relies on the classification made in the pre-processing stage. The training data is therefore formed by days of the same category of the forecasted day. Also it limits the selected days to those within a certain radius around the forecasted day, to eliminate days from different seasons of the year.

This stage can be analyzed by including 2 changes:

- Extending the period selection to all available data.
- Extending the period selection but maintain the condition of using days only of the same type.

Table 3 shows the result of these two tests.

TABLE I
RESULTS OF PRE-PROCESSING DUMMY STAGE

PERFORMANCE REPORT	NO FILTER	MORE STRICT FILTER	ONLY 2 DAY TYPES	NO NORM.	ORIGINAL METHOD
MAPE	6,24%	2,58%	3,89%	5,48%	2,89%
MAX	25,30%	4,35%	7,78%	8,54%	6,72%
# VALID DAYS	365	275	306	306	306

TABLE II
RESULTS OF VARIABLE SELECTION DUMMY STAGE

PERFORMANCE REPORT	PREV. DAY LOAD + TEMP	ORIGINAL METHOD
MAPE	3,37%	2,89%
MAX	6.89%	6,72%
# VALID DAYS	306	306

TABLE III
RESULTS OF TRAINING PERIOD SELECTION DUMMY STAGE

PERFORMANCE REPORT	ALL AVAILABLE DATA	SAME DAY TYPE	ORIGINAL METHOD
MAPE	4.38%	3.15%	2,89%
MAX	11,30%	5,27%	6,72%
# VALID DAYS	306	306	306

D. Forecasting engine

The model is based on self-organizing maps [11] (SOM). It is, therefore, an artificial intelligence based model. The number of neurons in the input layer is defined by the size of the input vector. The output layer is however a 2-dimensional grid defined by the user. The size of the grid is chosen through trial and error. There are several other parameters selected by this method: sheet topology of the grid, Gaussian neighborhood function, linear initialization of the map and batch training algorithm. All these parameters are default settings in the Matlab Som Toolbox [12]. In addition, the number of training iterations is also set by trial and error.

The output of the forecasting engine is specified by a mask that interprets which information is known at the time of the forecast and which is the desired output. In this model, the output is a 24 values vector corresponding to a daily load profile.

To evaluate the effect of properly selecting the parameters the following tests will be carried out. In this case, table 4 includes the computational cost in terms of seconds to forecast a full year:

- Reducing the size of the map: The output layer is formed by 5 rows of 5 neurons each instead of the 10 x 10 grid.
- Reducing training epochs: The default setting in the Som Toolbox is changed from 'long' to 'short'
- Use sequential training instead of batch training.

E. Post-processing

Since the pre-processing stage used normalization, the first post-processing task is to renormalize. Then all classification information is also added, in order to better analyze the result.

As for performance report, there are many MAPE results corresponding to some the trial and error tests used to justify which input variables or training period should be used.

TABLE IV
RESULTS OF FORECASTING STAGE

PERFORMANCE REPORT	SMALLER MAP	SHORTER TRAINING	SEQUENTIAL TRAINING	ORIGINAL METHOD
MAPE	3,34%	3,15%	2,89%	2,89%
MAX	7,31%	7,35%	6,72%	6,72%
TIME FOR 365 DAYS (s)	161	176	1653	1254

However, the performance of the definite model is only described in terms of MAPE with a brief analysis of its evolution throughout the year.

There is only one database used from which no information is given. Therefore it is highly difficult to foresee if the proposed model is applicable in any other case.

Improvement of the post-processing stage would come from establishing a better performance report including information about the robustness of the result. The histogram shown in fig. 2, may be advisable to include. Also, including the results of another database, preferably from a different region would give a better understanding of the model's performance.

F. Analysis

The contribution of each stage is clearly relevant and all of them deserve the proper attention when designing a forecasting model. Obviously, some contributions are easier to obtain than other, as it may be easier to design a filter for missing values than a complex forecasting engine. Nevertheless, all five stages should be well established when creating and describing a model.

Regarding the improvement possibilities of the model, the pre-processing stage does not implement any decomposition of the load which, in some cases it has been proven successful. In the variable selection, it should consider optimizing the input variables for shorter periods of time because the best input for a summer day is probably not the optimum for winter. Training period selection is quite complete, although the same claim made to variable selection can be made here. Most of the design of the forecasting engine is based on trial

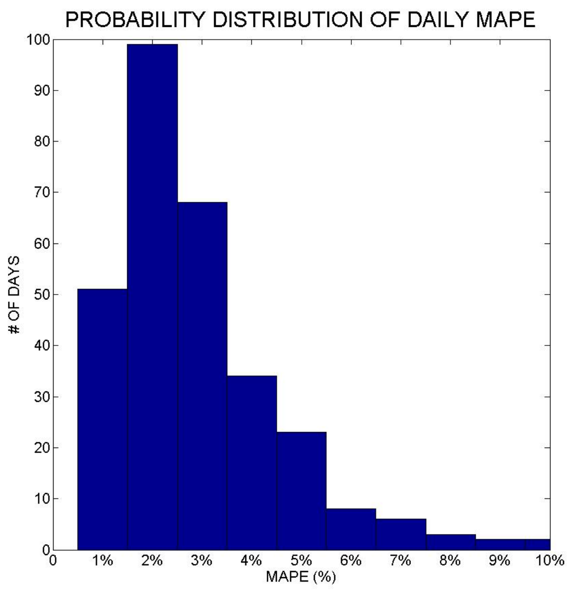

Fig.2. Histogram of the MAPE obtained by the original model.

and error and, therefore, it is assumed that the best parameters have been chosen. However, the question of whether a SOM network performs better than a MLP network or any other type remains unexplored in the model.

V. CONCLUSION

The main purpose of this paper is to provide with a framework for designing and evaluating forecasting models. Many of the recently proposed models have been described, showing that all the processes within those models can be arranged in a standard scheme.

This standard scheme is a useful guide to develop new forecasting models, enhancing the possibilities of combining parts of different models and promoting hybridization.

The application of this scheme to analyze an actual model helps to appreciate that all stages of load forecasting are relevant but also it facilitates finding improvement possibilities for the model analyzed.

In conclusion, it is advised to follow the standard scheme proposed as it is a useful tool for engineers, scientist and researchers and it may set the first stone into standardization of load forecasting models.

REFERENCES

[1] Weron, Rafal, “Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach”, 2007, John Wiley & Sons, Ltd.

[2] Hippert, H. S.; Pedreira, C. E.; Souza, R. C.”; “Neural networks for short-term load forecasting: a review and evaluation“ IEEE Transactions on Power Systems, vol. 16, no. 1. (2001), pp. 44-55.

[3] Kiartzis, S. J.; Bakirtzis A. G.; Petridis, V., “Short-term load forecasting using neural networks”, Electric Power Systems Research, volume: 33, issue: 1, April 1995, pages:1-6.

[4] M. López, S. Valero, C. Senabre, J. Aparicio, A. Gabaldón, “Development of a Model for Short-Term Load Forecasting with Neural Networks and its Application to the Electrical Spanish Market”, 8th International Conference on the European Energy Market, pp. 321 – 326. 2011.

[5] M. Hanmandlu, and B. K. Chauhan, “Load forecasting using hybrid models”, IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 20-29, Feb. 2011.

[6] Z. A. Bashir and M. E. El-Hawary, “Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks”, IEEE Transactions on Power Systems, vol. 24, no. 1, pp. 20-27, Feb. 2009.

[6] Z. A. Bashir and M. E. El-Hawary, “Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks”, IEEE Transactions on Power Systems, vol. 24, no. 1, pp. 20-27, Feb. 2009.

[7] N. Amjady and F. Keynia, “Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm”, Energy The International Journal, n. 34, pp. 46-57, 2009.

[8] Ying Chenm Peter B. Luh, Che Guan, Laurent D. Michel, Matthew A. Coolbeth, Peter B. Friedland and Stephen J. Rourke, “Short-Term Load Forecasting: Similar Day-based Wavelet Neural Networks”, IEEE Transactions on Power Systems, vol. 25, no. 1, pp. 20-27, Feb. 2010.

[9] C. M. Lee, and C. N. Ko, “Short–term load forecasting using lifting scheme and ARIMA models”, Expert Systems with Applications, n. 38, pp. 5902-5911, 2011.

[10] Paras Mandal, Tomonobu Senjyu, Naomitsu Urasaki and Toshihisa Funabashi, “A neural network based several-hour-ahead electric load forecasting using similar days approach”, International Journal of Electrical Power & Energy Systems, volume: 28, December 2006, pages: 367-373.

[11] Kohonen, T., “Self-Organisation and Associative Memory” (3rd edn.), Springer-Verlag, Berlin, 1989

[12] Vesanto, J.; Himberg, J; Alhoniemi, E.; Parhankangas, J., “SOM Toolbox for Matlab 5”, ISBN 951-22-4951-0, www.cis.hut.fi/projects/somtoolbox/.