A Hybrid Ensemble Model Applied to the Short-Term Load Forecasting Problem

Conference Paper · July 2006

DOI: 10.1109/IJCNN.2006.247141 · Source: IEEE Xplore

CITATIONS
13

READS
101

6 authors, including:

Ricardo Menezes Salgado
Universidade Federal de Alfenas
35 PUBLICATIONS **275** CITATIONS
SEE PROFILE

Takaaki Ohishi
University of Campinas
53 PUBLICATIONS **681** CITATIONS
SEE PROFILE

Rosangela Ballini
University of Campinas
153 PUBLICATIONS **1,399** CITATIONS
SEE PROFILE

Clodoaldo A M Lima
University of São Paulo
84 PUBLICATIONS **932** CITATIONS
SEE PROFILE

2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A Hybrid Ensemble Model Applied to the Short-Term Load Forecasting Problem

R. M. Salgado, J. J. F. Pereira, T. Ohishi, R. Ballini, C. A. M. Lima and F. J. Von Zuben

Abstract — **In this paper we present a methodology based on a combination of many distinct predictors in an ensemble, named hybrid ensemble model, to obtain a more accurate output using the results of single predictors. As basic components, we have used Artificial Neural Networks and Support Vector Machines models. In order to evaluate the performance, the hybrid model was required to predict a 24h daily series energy consumption of a Brazilian electrical operation unit located in the northeast of Brazil. The proposed ensemble model has reached an error 25% smaller than that achieved by the best single predictor. The model was initialized several times to confirm that ensembles of predictors also tend to produce low variance profiles.**

I. Introduction

The forecast of the load demand is a fundamental task in the operation of electric energy systems, because several decision making processes, such as system operation planning, security analysis and market decisions, are strongly influenced by the predicted load. In this context, a significant error in the load forecast may result in economic losses, security constraint violations, and degradation in system operation. So, accurate and reliable load forecasting models are essential for a suitable system operation.

The problem of load forecasting can be classified in long, medium and short-term, depending on the application area. Long-term forecasting is important for the expansion of the system's capacity. Medium term is important to organize the fuel supply, supporting operation and interchange scheduling. Short-term forecasting is generally used for daily programming and operation of the electrical system, energy transfer, and demand management [3].

Particularly, in the context of short-term hydrothermal scheduling, load forecasting is important to elaborate the next day's operation scheduling, because errors in load forecasting can guide to serious consequences, affecting the efficiency and the safety of the system (cost increasing, insufficient supply of electrical energy, given the current demand).

The load forecasting is a traditional research and development area. A wide variety of models for short-term load forecasting have been reported in the literature, based on statistical methods, such as exponential smoothing [6], Box and Jenkins models [7], Kalman filters [8] and regression [9].

Artificial Neural Networks (ANNs) have been widely applied to short-term load forecasting problems [10]. One of the main reasons for their success is the universal approximation capability presented by ANNs; such models are able to approximate, with arbitrary levels of accuracy, any continuous mapping defined on a compact (closed and bounded) domain [14]. However, ANNs have also shown some drawbacks due to their generic structure. Two commonly cited disadvantages are that the neural models usually require the estimation of a large number of parameters to achieve good results and that it is usually hard to design a priori an appropriate network topology and/or to choose the types of nonlinearities (activation functions) [15]. Further, when considering the supervised mode of learning, it is necessary to cope with the bias/variance tradeoff [16].

Nowadays, there is much interest in the study of Support Vector Machines (SVM), both for regression [1] and classification [22] problems. The SVM approach is based on the minimization of the structural risk [23], which asserts that the generalization error is delimited by the sum of the training error and a parcel that depends on the Vapnik-Chervonenkis dimension. By minimizing this summation, high generalization performance may be obtained. Besides, the number of free parameters in SVM does not explicitly depends upon the input dimensionality of the problem at hand. Another important feature of the support vector learning approach is that the underlying optimization problems are inherently convex and have no local minima, which comes as the result of applying Mercer's conditions on the characterization of kernels [24].

However, the SVM applicability has been hampered by the necessity of choosing a priori (i) the kernel function (responsible for the mapping); (ii) the parameter(s) of the kernel function ; (iii) the loss function (penalty function); and (iv) the trade-off parameter C (which controls the trade-off between closeness to the data and smoothness). Sometimes, this turns to be a non-effective process as each kernel function has it pros and cons. Lima et al. [25] have then conceived ensembles of SVMs in order to alleviate performance bottlenecks incurred with the "kernel function choice" problem.

By other means, ensembles of neural networks [28], [29] involve the generation, selection, and linear/nonlinear combination of a set of individual ANN designed to simultaneously cope with the same task. This is typically done through the variation of some configuration parameters and/or employment of different training procedures, such as bagging and boosting [31]. Such ensembles, a.k.a.

R. M. Salgado, J. J. F. Pereira and T. Ohishi are: DENSIS – FEEC – UNICAMP, 13082-852 Campinas/SP BRAZIL (e-mail: {ricardo, joaquim, taka}@densis.fee.unicamp.br).
R. Ballini: DTE – IE – UNICAMP, 13083-857 Campinas/SP BRAZIL (e-mail: ballini@eco.unicamp.br).
C. Lima and F. J. Von Zuben: DCA – FEEC – UNICAMP , 13083-852, Campinas/SP BRAZIL (e-mail: {moraes,vonzuben}@dca.fee.unicamp.br}

committees, should properly integrate the knowledge embedded in the component networks (devised to provide redundancy), and have frequently produced accurate and robust models [30].

Aiming at combining complimentary properties manifested by the aforementioned approaches, in this work we concentrate our efforts on the descriptive characterization and empirical evaluation of a novel hybrid methodology denoted Hybrid ensemble. By this means, SVMs and ANNs models can be readily incorporated as component into ensembles, making it possible, for instance, the allocation of ANNs with distinct topologies and/or SVMs with distinct kernel functions. The main purpose is to augment the generalization capability of ensemble models.

The proposed model was applied to predict the active load (MW/h) 24h ahead. The effectiveness of the proposed approach is illustrated taking as case study a Brazilian electrical operation unit located in the northeast of Brazil.

II. Ensemble Forecasting: Methodology

A. Short-Term Load Forecasting

Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. Load forecasting is used by energy management systems to establish operational plans for power stations and their generation units. They are also necessary for energy suppliers, financial institutions, and other participants in energy generation, transmission, distribution, and markets.

In this paper, the objective of short-term load forecasting is high performance in the 24 hours ahead load prediction. This approach aims to help the planning in real time of electric energy systems. Fig. 1 shows the classic load forecasting model. In evaluating the forecast model, a database consisting of hourly load is used to adjust the parameters of the model and to produce a load forecasting. For a good performance of the model, it is essential to select appropriate input variables.

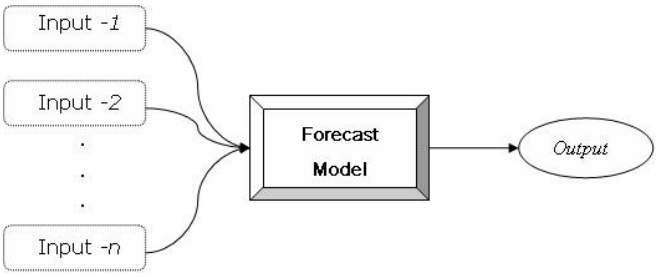

Fig. 1. General Forecast Model.

B. Ensembles

The term "ensemble" is the one commonly used for the combination of a set of learning machines (hereafter referred to as components, models or predictors) that provide isolated solutions to the same task, generally obtained by different means (e.g., by employing different machine learning paradigms such as ANNs, decision trees, etc.) [17], [18] and [19]. As pointed out by Sharkey [26], "combining a set of imperfect predictors can be thought of as a way of managing the recognized limitations of the individual predictors; each component is known to make errors, but they are combined in such a way as to minimize the effect of these errors".

Many works have been done in investigating why and how an ensemble of learning machines works. The basic motivation is to find proper mechanisms for exploiting, instead of ignoring, the information constructed by each component throughout its learning process, in such a way as to produce a final model containing the best of each individual capability generated. Irrespective of the method employed to train the components and then to create the ensemble, the idea *per se* of combining estimators is justifiable, having in mind the bias-variance (BV) analysis [20]. The bias of an estimator can be characterized as a measure of its ability to generalize correctly to a test dataset once trained, whereas its variance can be regarded as a measure of how sensitive are the estimator's outcomes to the data on which it was trained.

Krogh and Vedelsby [21] has formally proved that the generalization error of an ensemble can be shown to consist of two distinct portions, viz. a term measuring the average generalization error of each member and a term, called ambiguity, which accounts for the disagreement amongst the generalization patterns presented by the components. Consequently, promoting less ambiguity should incur higher overall generalization performance. Towards this end, there are a number of training issues that can be manipulated for this purpose (see subsection B.2). Fig. 2 presents the general form of an ensemble dedicated to regression problems, such as forecasting.

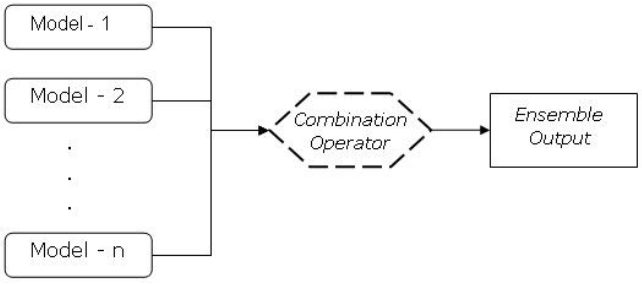

Fig. 2. General Ensemble Model.

In the construction process of an ensemble, there are three basic steps: generation of many different candidates to components, selection of components, and combination of their results. To implement this methodology, usually three data sets are needed: a data set for generating the candidates to components, another for selecting them, and yet another one to test the ensemble's performance. After generating several candidates to components, the selection of the best subset of those components is fulfilled. Generally, this selection is made aiming at decreasing the generalization error. The remaining step to construct an ensemble is to combine the components. In other words, the outputs of the selected components must be someway combined to generate a unique output for the ensemble. In what follows, additional details are provided concerning the design procedure adopted in this paper for ensembles.

1) Types of Components

In order to solve the load forecasting problem, we adopt

as candidates to compose an ensemble neural networks models, more specifically multilayer perceptrons, and support vector machines.

a) Multilayer Perceptrons (MLPs)

MLP architectures are known as the most frequently adopted neural network models for regression tasks. An MLP consists of *n* inputs nodes, *h* hidden layer nodes, and *m* output nodes connected in a feed-forward fashion via multiplicative weights that can be arranged in a weight matriz **W**. The MLP must be trained with historical data to find the appropriate values for the elements in matrix **W**, given the number of neurons in the hidden layer. In this paper, the learning algorithm employed is the well-known error back-propagation and their variations, such as Fletcher & Reeves and Pollack-Ribière conjugate gradient.

b) Support Vector Machines (SVMs)

Given a training data set of *N* points $\{(\mathbf{x}_t, \mathbf{y}_t)\}$, $t = 1,..., N$, with input data $\mathbf{x}_t \in \Re^n$ and output data $y_t \in \Re$, the function *f*, which is expected to correctly approximate the input-output mapping, can be written as

$$f(\mathbf{x}_t) = \mathbf{w}^T \varphi(\mathbf{x}_t) + b\,, \tag{1}$$

where $\varphi(.): \Re^n \to \Re^{n_h}$ is a function mapping the input space into a so-called higher dimensional feature space. The weight vector $\mathbf{w} \in \Re^{n_h}$ is considered to be in a primal weight space [23] [24]. Parameters **w** and *b* can be obtained by solving the following optimization problem in such a primal weight space:

$$\min_{\mathbf{w},b} \Phi(\mathbf{w},\xi,\xi^*) \equiv \min_{\mathbf{w},b} \frac{1}{2}(\mathbf{w}^T\mathbf{w}) + \frac{C}{k}\left(\sum_{t=1}^{N}(\xi_t)^k + \sum_{t=1}^{N}(\xi_t^*)^k\right), \tag{2}$$

subject to constraints

$$\begin{cases} y_t - \mathbf{w}^T\varphi(\mathbf{x}_t) - b \le \varepsilon + \xi_t\,, & t = 1,\cdots,N \\ \mathbf{w}^T\varphi(\mathbf{x}_t) + b - y_t \le \varepsilon + \xi_t^*\,, & t = 1,\cdots,N \\ \xi_t \ge 0\,,\ \xi_t^* \ge 0\,, & t = 1,\cdots,N \end{cases} \tag{3}$$

where ξ_t, ξ_t^*, t=1,...,*N*, are slack variables measuring the difference between y_t and the actual SVM output, and *C* is a tradeoff parameter set in advance to control the contribution of each term in the cost function of problem (2) to the complexity of the resulting SVM model. The value of the parameter *k* is set in accordance with the employed loss function, which is an auxiliary cost function that measures the approximation quality of the generated mapping (i.e., it is a function of ε in (3)) [24]. For the ε-insensitive loss function, $k = 1$ and for the ε-quadratic loss function, $k = 2$.

For convenience, problem (2) is usually converted into an equivalent problem defined in a dual space. Considering $k = 2$, then the parameters of function *f* can be determined by solving a new quadratic optimization problem:

$$\begin{aligned} \min_{\alpha,\alpha^*} L(\alpha,\alpha^*) \equiv \min_{\alpha,\alpha^*} & -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{2C}\sum_{i=1}^{N}(\alpha_i^*)^2 \\ & + \frac{1}{2C}\sum_{i=1}^{N}(\alpha_i)^2 - \sum_{i=1}^{N}\varepsilon(\alpha_i + \alpha_i^*) - \sum_{i=1}^{N} y_i(\alpha_i^* + \alpha_i) \end{aligned} \tag{4}$$

subject to: $\sum_{i=1}^{N}\alpha_i^* = \sum_{i=1}^{N}\alpha_i\,, \quad \alpha_i^* \ge 0, \quad \alpha_i \ge 0, \quad i = 1,\cdots,N\,,$

where $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ is a particular kind of function, known as kernel, that strictly follows the constraints imposed by Mercer's Theorem and that provides a one-step, implicit calculation of the product between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ [27]. Depending upon how this inner-product kernel is generated, it is possible to construct different learning machines characterized by nonlinear decision surfaces of their own. In Table I, we enumerate the various kernels generally adopted during the simulation experiments. A more detailed discussion on these kernels may be found elsewhere [22].

TABLE I
KERNELS - SVM

Type Kernel	Expression	Parameter
i.Linear	$K(x_i,x_j) = x_i \cdot x_j$	-----
ii.Polynomial	$K(x_i,x_j) = (x_i \cdot x_j + 1)^d$	d
iii.Gaussian Radial Basis Function	$K(x_i,x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$	σ
iv.Exponential Radial Basis Function	$K(x_i,x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right)$	σ
v.Hyperbolic Tangent	$K(x_i,x_j) = \tanh(b(x_i \cdot x_j) + c)$	b, c
vi.Fourier Series	$K(x_i,x_j) = \frac{sin(\delta + \frac{1}{2})(x_i - x_j)}{sin(\frac{1}{2}(x_i - x_j))}$,	δ
vii.Linear Splines	$K(x,y) = 1 + xy + xy\min(x,y) - \frac{(x+y)}{2}(\min(x,y))^2 + \frac{1}{3}(\max(x,y))^3$	-----

2) Generation of the components

The main purpose of an ensemble is to provide performance improvement, which is obtained when a few requirements are satisfied by the candidates to components. However, the generation of components in still the most relevant and the most demanding phase of the whole design process.

Here, the objective of the generation of components is to synthesize ANN's and SVM's with good individual performance and outputs as dissimilar as possible. The generation methods will follow one of two approaches: (*i*) preprocessing training data; (*ii*) preprocessing the parameters and structural features of ANN's and/or SVM's.

a) Preprocessing Data

In the case of the availability of distinct training sets, we can train components with the same architecture and obtain different individual results. Preprocessing of training data is necessary to obtain distinct training sets, and this procedure is often adopted to create ensembles with components capable of expressing distinct behavior [2], [26], [4].

The most straightforward way to generate distinct sets is

to adopt sampling without repetition, taking the whole training set as input [26]. The great drawback of this method can be noted when the number of samples in the training set is small; in this case, the subsets will be also small and possibly with a low representative power, guiding to low performance predictors [4].

An alternative proposal to alleviate the effect of the limited number of samples is to adopt sampling with repetition. In other words, the same sample can be chosen many times for the same subset. As the subsets will have the cardinality of the original set, repetition of arbitrary samples imply absence of others in that subset. This method is called bagging (*Bootstrap Aggregating*) [31]. In this work, we adopted this method to generate the subsets that will represent training datasets for the candidates to compose the ensemble.

b) Preprocessing: Structural Aspects

The structural aspects of preprocessing are devoted to creating a pool of components that generalize in different forms, even when using the same training dataset.

In the case of ANN's, the main methods that have been employed for the creation of neural network-based ensemble members can be classified as [26]: (*i*) varying the set of initial random weights, (*ii*) varying the topology, (*iii*) varying the training algorithm. The purpose is to create predictors with good performance and with decorrelated outputs.

In the case of SVM's, the adjustable parameters are: kernel type, parameter tradeoff *C*, loss function type. Two distinct SVM's trained with the same dataset can generate two candidates which distinct behavior. Essentially, this variability in performance is not desired when synthesizing a machine learning tool, but can be properly explored in the context of ensembles.

3) Selection of Components

In this phase, the candidates already implemented will be considered of interest when they present an individual good performance and when they are diverse at the output. However, there is no theoretical foundation that can guarantee that those components will contribute positively; it is just an empirical conclusion. After generating the components, it is possible to select the best by assigning an individual performance criteria to each component, based on decorrelation indices and error rates, choosing only the n best or until some stopping condition is satisfied.

In this work, we considered the same technique proposed in [5], where the components are ordered according to their mean square error (MSE). The selected components will be those that present a performance that admits at most 20% of degradation when compared with the performance of the best candidate.

4) The combined forecasting model

The idea of using a combined forecasting model is not new. Since the pioneering works of Reid [11] and of Bates and Granger [12], several attempts have been conducted to indicate that a combination of forecasts often outperforms the forecasts obtained from a single source.

The main problem when combining forecasts can be described as follows. Supposed there are *r* forecasts such as $\hat{y}_1(t), \hat{y}_2(t), \ldots, \hat{y}_r(t)$. The question is how to combine these different forecasts into a single forecaster $\hat{y}_{ag}(t)$, which is intended to be a more accurate one. The general form of the model for such a combination can be defined as:

$$\hat{y}_{ag} = \sum_{i=1}^{r} w_i \hat{y}_i \tag{5}$$

where w_i denotes the assigned weight of $\hat{y}_i(t)$. There are a variety of methods available to determine the weights used in the combined forecasts. First of all, the equal weights method, which uses an arithmetic average of the individual forecasts, is a relatively easy and robust method. However, since the components are diverse in terms of behavior and average performance, in practice a simple average may not be the best technique to use [13].

In this work, we use a simple model for the linear combination of predictions, with a neural structure composed of a single layer and just one linear neuron (see Fig. 3). The weights are adjusted using gradient descent.

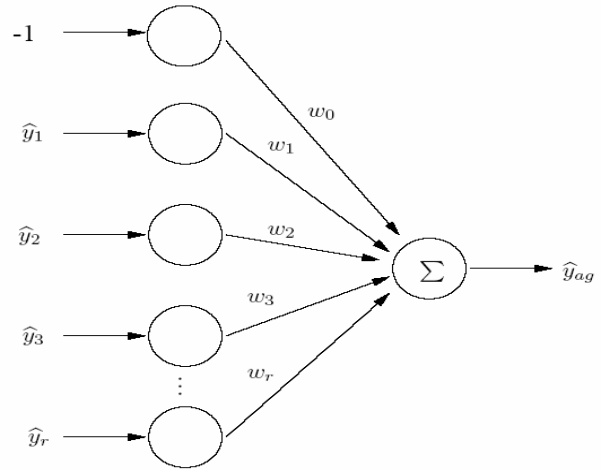

Fig. 3. Combination Operator.

III. Application of the Ensemble Model in Short-Term Load Forecasting

The proposed ensemble model was configured using various ANN's and SVM's predictors, calibrated with different configurations. The ensemble is divided in three stages: generating and training the components, validating and selecting the components, and combining and predicting. The application involves short-term load forecasting.

To determine the load demand curve of the next day (24 hours ahead) an adjustment of the models is accomplished to deal with every hour of the day. This way, for each time interval an individual forecasting model is adjusted. So, 24 forecasts will be accomplished for each component of the ensemble, aiming at obtaining the prediction of the whole demand for each hour of the day.

Table II shows the training algorithm, number of components and neurons, and learning rate used on the MLP model. Table III shows the number of SVM models, and their respective configuration parameters.

TABLE II
ARGUMENTS USED: MLP

Comp.	Training Algorithm	Nº Neuro	Learning Rate	Momentum Rate
1	Backpropagation (BPM)	10	0.01	0.9
2	Backpropagation (BPM)	12	0.01	0.9
3	Davidon-Flecther-Powell (DFP)	10	-	-
4	Davidon-Flecther-Powell (DFP)	12	-	-
5	Fletcher & Reeves (FR)	10	-	-
6	Fletcher & Reeves (FR)	12	-	-
7	Gradient (GRAD)	10	-	0.9
8	Gradient (GRAD)	12	-	0.9
9	Pollack-Ribière (PR)	10	-	-
10	Pollack-Ribière (PR)	12	-	-

TABLE III
ARGUMENTS USED: SVM

Comp.	Kernel function	|Grid| |Sigma| |Scale|	Constant *C*	Loss Function
11	Linear	-	5	ε – quadratic
12	Polynomial	2	5	ε - insensitive
13	Rbf	3	5	ε – quadratic
14	Tangent	2	5	ε - insensitive
15	Spline	-	5	ε – quadratic
16	Fourier	3	5	ε – quadratic
17	eRbf	3	5	ε - insensitive

All SVM components having the loss function ε-insensitive were configured with $\varepsilon = 0$.

A. Data Segmentation

In this work we have used a daily series energy consuming, measured in an electrical system at northeast of Brazil. The measures were made in the period from July, 6th 2001 to September, 3rd 2001.

Table IV shows, the maximum, the minimum, the average, and standard deviation of load series. Comparing the standard deviation value with the average load value, in Table IV, we may note that the behavior of this series has low variability. Under these circumstances, high-quality individual predictors where achieved.

TABLE IV
LOAD SERIES CHARACTERISTICS

Max Load	*Min Load*	*Average Load*	*Std Deviation*
64.5 Mw	21,2 Mw	35,2 Mw	2,36 Mw

In Fig. 4, we can note that the consuming profile shows great regularity with the peak load occurring in the period from 5:00pm to 10:00pm. Moreover, we can note that the consumption curves have distinct profiles depending on the day of the week, i.e. working days or weekends, as expected. Instead of treating the weekdays separately, we will let the ensemble detect such a well-known variability.

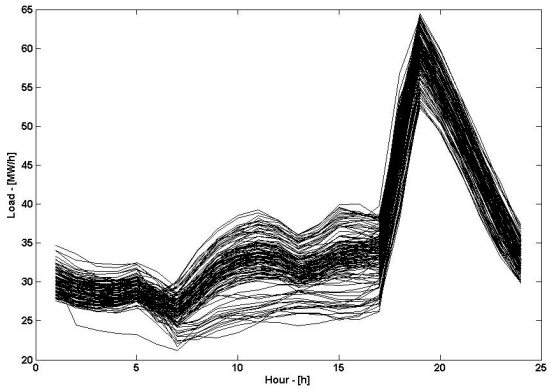

Fig. 4. Curves of energy consumption.

The training set is composed of 88% of the whole data, thus trying to improve the most the performance of the candidates to components of the ensemble. Validation and test sets correspond to 12% of the data (6% each). In this way, we hope that low errors in the validation process will indicate low errors in the overall prediction.

IV. RESULTS

The proposed ensemble model is composed of component generation, validation and selection, and test of the resulting ensemble. The validation phase is used to select the best candidates to compose the ensemble. In the test phase, the selected components are combined to generate the ensemble.

For the training and validation phases, we have used recorded data from 6/1/2001 to 9/25/2001, and for the overall evaluation, it was allocated specifically the day 10/3/2001. In order to evaluate and to compare the performance of the models, we adopted the mean absolute percentage error (MAPE) (Eq. 6), and the mean square error (MSE) (Eq. 7) between the observed and estimated loads in the prediction.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{\left|x_i - \hat{x}_i\right|}{x_i} \tag{6}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2 \tag{7}$$

where x_i denotes the actual load and $\hat{x}_i$ denotes the forecasted load.

To analyze the stability of the ensemble model, we initialized the parameters of the MLP and the SVM models 26 times and we considerer the performance of the ensemble in the validation set. In what follows, the obtained results are outlined.

A. Validation Phase

Fig. 5 shows the percentage of the components selected in the validation phase, with 26 initializations of the model. Note that the three components most often selected were: 15-[21%], 12-[15%] and 5-[11%].

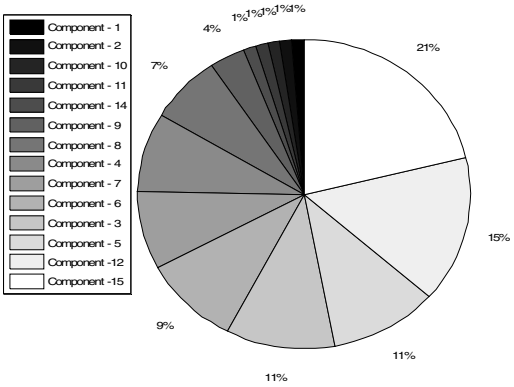

Fig. 5. Components Selection – Validation Phase [%]

Fig. 6 shows MSE calculated in the validation phase. The errors presented by the components during this phase have high variability, and this fact reinforces the idea of choosing the best candidates to take part in the ensemble.

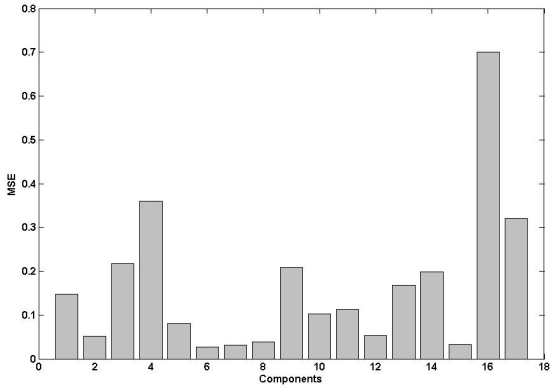

Fig. 6. MSE Components – Validation Phase.

The selection of candidates was made considering the MSE, when chosen the three best candidates in the validation phase. By means of this criterion, the 15th candidate (SVM with the spline kernel function), and the 6th and 7th candidates (ANN adjusted with Fletcher & Reeves and Gradient algorithms, respectively) have been selected.

B. Prediction Phase

Fig. 7 shows the percentage of components selected in prediction phase with 26 initializations of the model. Note that the three candidates most often selected were: 15-[28%], 8-[19%] and 7-[16%]. Comparing Fig. 5 and 7, it can be noted that the 15th candidate presented the better results. Analyzing the others candidates that composed the ensemble, note that the components presented better results in the validation phase also presented better performance in the prediction phase.

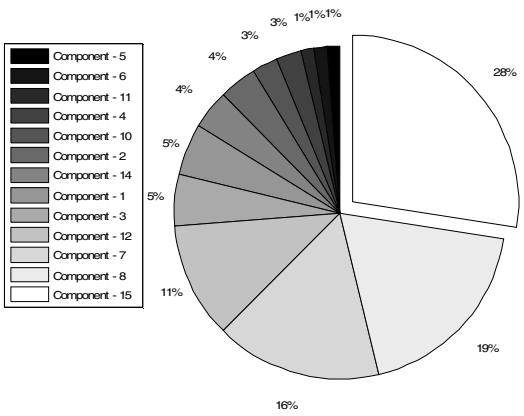

Fig. 7. Best Components in the Test Phase.

Fig. 8 shows the MAPE obtained by the resulting ensemble model and the individual predictors. Note that the proposed ensemble model is able to generate stable solutions when compared with the best components.

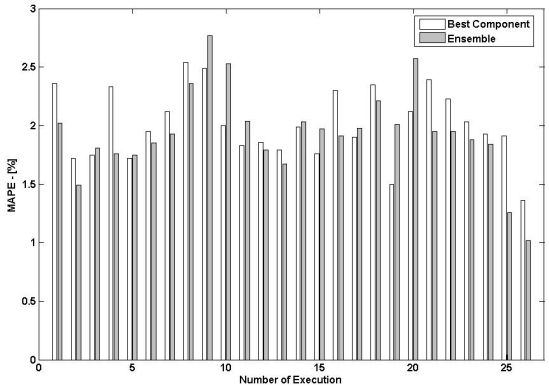

Fig. 8. Ensemble and Best Components After 26 Execution.

C. Ensemble: the best result

Fig. 9 shows load curves resulting from all single components (candidates to take part in the ensemble) in the scenario with the best performance for the ensemble.

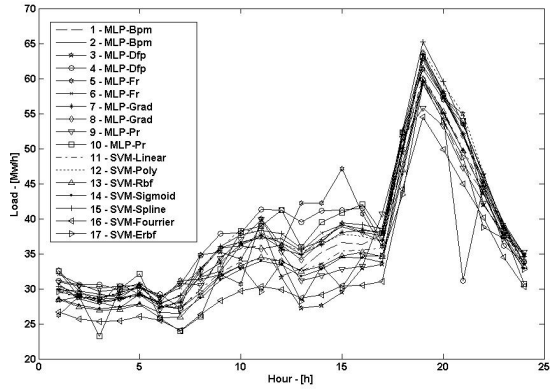

Fig. 9. Forecasted Loads Curves – Individual components of the ensemble.

In Table V, we can note that the obtained errors in the prediction were compatible with the errors in the validation

process. The selected components in the validation process were those that, in the majority of the cases, have obtained the smallest predictions errors.

TABLE V
PREDICTIONS ERRORS– CANDIDATES TO COMPONENTS OF THE ENSEMBLE

Candidates	MAPE -%	MSE
1	4.01	4.68
2	2.08	0.86
3	8.00	16.10
4	7.79	25.08
5	6.55	10.14
6	2.34	1.03
7	1.36	0.47
8	2.03	0.88
9	6.49	10.81
10	6.29	7.15
11	5.04	4.61
12	2.86	1.75
13	7.10	7.93
14	7.56	9.76
15	1.93	0.85
16	15.39	38.28
17	8.41	17.95

One can realize, observing the predicted curves, that among the 17 predictors originally created, six have reached MAPE lower than 3%, that is, 36% of the candidates have solved the problem in a proper way. Still observing the load curves, we can realize that 64% of the candidates were not efficient in obtaining the predicted curve. This fact does not imply that this tool is not recommended to solve the load-forecasting problem. As the parameters of the models were randomly determined, there is the probability that some parameters of individual candidates are not those which maximize the performance.

Fig. 10 shows, in the best result for the ensemble, the predicted curves by the selected models to create the overall ensemble forecast.

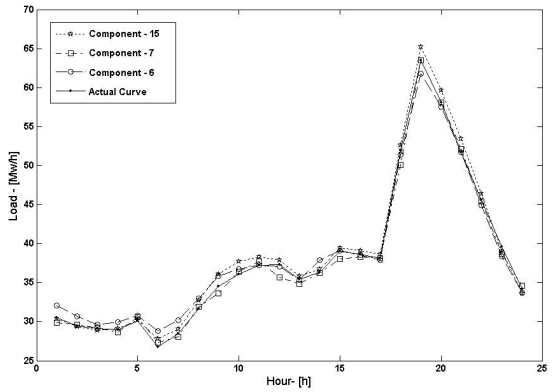

Fig. 10. Forecasted Curve – Selected candidates to components

According to the errors presented in Table V, the selected candidates to components have MAPE lower than 2.34%. It is important to highlight that some candidates are pruned during the selecting process (8th candidate, for instance) which presented inferior errors than those selected; this happens due to the selection applied at the validation phase.

Fig. 11 shows the predicted load curve from the ensemble and also the real load. It is possible to note that the combiner was able to extract the characteristics from the selected components.

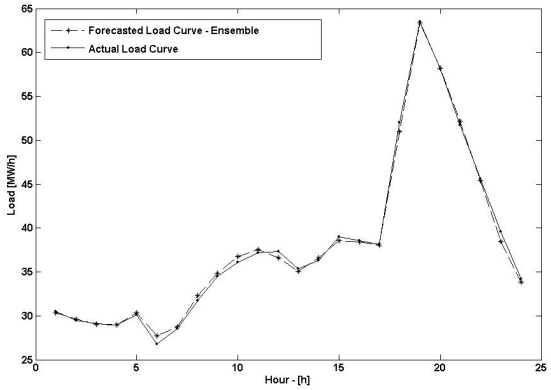

Fig. 11. Actual Curve × Predicted Ensemble Curve.

On Table VI, the errors obtained from the ensemble are depicted. In terms of MAPE, the ensemble showed 99% of hits, corroborating the efficiency of the ensemble model when treating the proposed problem.

TABLE VI
ERROR: ENSEMBLE

MAPE	MAE	MSE
1.02	0.37	0.22

Fig. 12 shows the MAPE of the individual components compared with the ensemble. The ensemble was able to overcome the result of all the components showing a prediction with a high level of hits. The ensemble was able to reach an MAPE index 25% lower than the best individual component.

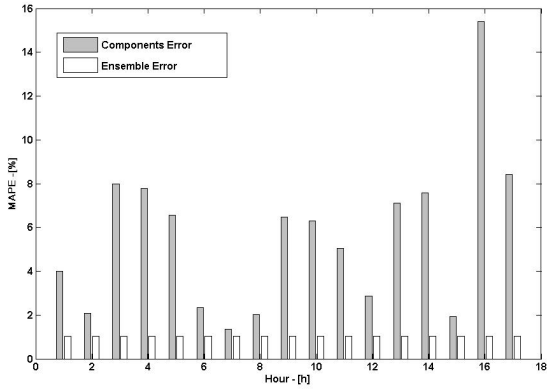

Fig. 12. MAPE: Ensemble × Components (best result).

The combination process proposed in this work has showed a good result, because it was able to identify the best characteristics of each component of the ensemble. If the combination process used was based on the arithmetic mean of the components, the ensemble would reach an MAPE around 5.60%, whereas with the neural combiner the reached MAPE was 1.02% (the best individual component has showed an MAPE of 1.36%). In this sense, one can realize that the process of combining the results has a strong influence on the final results of the ensemble.

V. Conclusion

According to the obtained results employing the ensemble methodology with a hybrid composition, the model has showed robustness, generalization capability and relevant information extraction, from the available candidates to components, in order to obtain a more accurate forecasting.

The process of generating the components was efficient, because it provided components that generalized in a dissimilarity way. This made it possible the acquirement of variability in the obtained solutions. The phase of selecting the best individual components has been successful, because the selected components showed high levels of hits in the test and prediction phase.

Thus, this technique emerges as a promising alternative to deal with short-term load forecasting problems. Perspectives for further research include a deeper comparative analysis with other proposals for short-term load forecasting, and an extension of the proposed methodology to handle other prediction tasks.

Acknowledgment

This work was partially supported by the Research Foundation of the State of São Paulo (FAPESP) and the Brazilian National Research Council (CNPq).

References

[1] Cortes, C.; Vapnik, V. (1995) Support vector networks. Machine Learning, vol. 20, pp. 273-297.

[2] Parmanto, B.; Munro, P. W.; Doyle, H. R. (1996) Reducing variance of committee prediction with resampling technique. Connection Science. Special Issue on Combining Artificial Neural: Ensemble Approaches, vol. 8, no 3 & 4, pp. 405-426.

[3] Rahman, S. and Hazim, O. (1993). A generalized knowledge-based short-term load-forecasting technique, IEEE Transactions on Power Systems 8(2).

[4] Tumer, K.; Ghosh, J. (1996) Error correlation and error reduction in ensemble classifiers. Connection Science. Special Issue on Combining Artificial Neural: Ensemble Approaches, vol. 8, no 3 & 4, pp. 385-404.

[5] Perrone M. P. (1993) Improving regression estimates: Averaging methods for variance reduction with extensions to general convex measure optimization, PhD Thesis, Brown University.

[6] Christiaanse, W. R. (1971). Shor-Term load forecasting usin general exponential smoothing, IEEE Transc. On Power Apparatus and Systems, 90:900-911.

[7] Meslier, F. (1978). New advances in short-term load forecasting using Box and Jenkins approach, IEEE/PES Winter Meeting, pp-51-55.

[8] Irisarri, G. D., Widergren, S. E. and Yehsakul, P. D. (1982). On-line load forecasting for energy control center application, IEEE Transac. On Power Apparatus and Systems 101: 71-78.

[9] Hesterberg, A. D. P. T. C. (1989). A regression based approach to short-term system load forecasting, Proc. Of PICA Conference, pp.414-423.

[10] Maier, H. R. and Dandy, G. D. (2000), Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and application, Environmental Modelling and Software, vol 15, pp. 101-124.

[11] Reid, D. J. (1968) Combining three estimates of gross domestic product. Economics, 35:431-444.

[12] Bates, J. M. Granger, GWJ (1969), The combination of forecasts, Operations Research Quaterly, 20:451-468.

[13] Kang, BH, (1986), Unstable weights in the combination of forecasts. Management Science, 32:683-695.

[14] K. Hornik, Multilayer feedforward networks are universal approximators". Neural Networks, 2(5): 359-366, 1989.

[15] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, 1999.

[16] S. Geman, E. Bienenstock, and R. Doursat, Neural Networks and the Bias/Variance Dilemma, Neural Computation, 4(1): 1-58, 1992.

[17] W.G. Baxt. Improving the accuracy of an artificial neural network using multiple differently trained networks, Neural Computation 4(5), 135-144, 1992.

[18] T. G. Dietterich. Ensemble methods in machine learning, in: Proc. International Workshop on Multiple Classifier Systems (MCS), LNCS 1857, pp. 1-15, Italy, Springer, 2000.

[19] L. Hansen and P. Salamon. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993-1001, 1990.

[20] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. Neural Computation 4, 1-58, 1992.

[21] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning, In: G. Tesauro, D. Touretzky, and T. Leen, (Eds.), Advances in Neural Information Processing Systems 7, 231-238, Cambridge, MA. MIT Press, 1995.

[22] Gunn S. Support Vector Machine for Classification and Regression. *Image Speech & Intelligent Systems Group*, *Technical Report ISIS-1-98*, University of Southampton, Nov. 1998.

[23] Vapnik V.N. The Nature of Statistical Learning Theory, *Springer Verlag*, 1995.

[24] Cristianini N., J. Shawe-Taylor. An Introduction to Support Vector Machines, *Cambridge. Press*, 2000.

[25] C. A. M. Lima, A. L. V. Coelho, and F. J. Von Zuben. Ensembles of Support Vector Machines for Regression Problems, in: Proc. of International Joint Conference on Neural Networks (IJCNN'02), 2381-2386, Hawaii, 2002.

[26] A. Sharkey (Ed.). Combining artificial neural nets*:* Ensemble and modular multi-net systems, Springer-Verlag, London, 1999.

[27] B. Schölkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA, 2002.

[28] Reilly, R. L.; Scofield, C. L. Elbaum, C; Copper, L. N. Learning system architetures composed of multiple learning modules, In *Proc. IEEE First Int. Conf. On Neural Networks*, vol 2, 1987, IEEE.

[29] Scofield, C. Kenton, L; Chang, J. Multiple neural net architetures for character recognition. In *Proc. Compcon, San Francisco,* CA. February 1991, pp 487-491, IEEE Comp, Soc. Press.

[30] Baxt, W. G. Improving the accuracy of an articial neural network using multiple differently trained network. Neural Computation, vol. 4, n 5, pp 135-144, 1992.

[31] Breiman, L. (1996b) Bagging predictors. Machine Learning, vol. 24, no 2, pp. 123-140.