



Multinodal load forecasting for distribution systems using a fuzzy-artmap neural network

Thays Abreu^{a,*}, Aline J. Amorim^a, Carlos R. Santos-Junior^b, Anna D.P. Lotufo^a, Carlos R. Minussi^a

^a Electrical Engineering Department, UNESP – São Paulo State University, Av. Brasil 56–P.O. Box 31, CEP: 15385-000, Ilha Solteira, SP, Brazil

^b IFSP - Federal Institute of Education, Science and Technology of São Paulo, Campus Hortolândia, Hortolândia, São Paulo, Brazil

ARTICLE INFO

Article history:

Received 22 December 2017

Received in revised form 16 May 2018

Accepted 25 June 2018

Available online 3 July 2018

Keywords:

Load forecasting

Electrical system distribution

Artificial neural networks

Adaptive resonance theory

ABSTRACT

This work proposes a predictor system (multinodal forecasting) considering several points of an electrical network, such as substations, transformers, and feeders, based on an adaptive resonance theory (ART) neural network family. It is a problem similar to global forecasting, with the main difference being the strategy to align the input and output of the data with several parallel neural modules. Considering that multinodal prediction is more complex compared to global prediction, the multinodal prediction will use a fuzzy-ARTMAP neural network and a global load participation factor. The advantages of this approach are as follows: (1) the processing time is equivalent to the processing required for global forecasting (i.e., the additional time processing is quite low); and (2) Fuzzy-ARTMAP neural networks converge significantly faster than backpropagation neural networks (improved benchmark in precision). The preference for neural networks of the ART family is due to the characteristic stability and plasticity that these architectures have to provide results in a fast and precise way. To test the proposed forecast system, the results are presented for nine substations from the database of an electrical company.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Precisely knowing the electrical load is a prior condition for implementing strategies to provide consumers with quality, cost-effective electrical services (e.g., voltage, frequency, and waveform shape). Most of the proposals found in the literature are for global load forecasting, which is the sum of all demanded loads. In this case, the predicted global load is divided considering the several busses of the system using some heuristic method. This division generally leads to certain errors that can compromise the quality of the required studies for a power system's operation (load flow analysis, voltage, and angle stability, among other important studies). Thus, this work proposes to develop a multinodal load forecasting method considering the load at several points of the electrical network, such as substations, transformers, and feeders, based on an adaptive resonance theory (ART) [1] family neural network. It is a similar problem to global load forecasting, with the main difference corresponding to the use of a strategy to prepare the input and output data with several parallel neural modules

aggregated to a global predictor. Therefore, historical consumption data provided by the electrical distribution systems are used. In this system, the load behavior is expected to repeat with some uncertainty. This uncertainty can be mitigated by utilizing exogenous information from historical sources in the prediction system. Generally, this was implemented as in the classical example known as the Box–Jenkins approach [2,3]. Today, researchers prefer the use of intelligent techniques, e.g., artificial neural networks (ANNs) [4], fuzzy logic [5], etc. However, multinodal prediction is complex in comparison to global prediction. The multinodal prediction method developed in this work employs two important resources: the use of the fuzzy-ARTMAP neural network (FANN) proposed by Grossberg [1] and the use of a global load participation factor (GLPF) [6,7]. The advantages of this approach are: (1) the low processing time in comparison to global forecasting and (2) using a FANN results in solutions orders of magnitude faster than backpropagation (BP) [8], which is a major benchmark of precision for ANNs. Full convergence is not even assured with BP neural networks when using diversified data. The preference for the ART family of ANNs [1] is due to the characteristic stability and plasticity that these architectures have to provide results quickly and precisely.

* Corresponding author.

E-mail address: thays7abreu@gmail.com (T. Abreu).

It is emphasized that the stability analysis of neural networks is an important matter considering the need to assure convergence in the training phase. Reference [9] provides a study in this sense.

To test the proposed prediction system, the results are presented considering substations of a power system centralized dataset (CDS) [10].

The load forecasting addressed in this paper is for 24 h in advance with 1 h (or ½ h) discretization. However, other resolutions can be achieved with little modification.

2. Related works

Global load forecasting accounts for the majority of the publications available in the literature based on techniques such as ANN [11], fuzzy logic [12–14], genetic algorithms [13], classical procedures [15–17], fuzzy-ART&ARTMAP ANN [18,19], ARIMA [2,3,16], ANN based on Levenberg–Marquardt [20] training method, ANN by gradient descent learning [21], and load forecasting based on multiregression–decomposition model [22]. There are few publications dealing with multinodal load forecasting, which include the following. In 1996, Chen et al. [23] realized a load forecasting method for three substations serving different types of consumers (residential, commercial, and industrial). The proposal was to improve the prediction with ANN, considering the effect of temperature on load demand. Hsu and Chen [24] propose a calculation method for regional load forecasting (Taiwan region) using ANN with a backpropagation algorithm [8], and comparing this with a regression model. They conclude that the ANN performance is superior to the regression models. Singh et al. [25] realize load forecasting with a feedforward ANN for a half-hour period of three days in a multiregion in Australia. Altran [6] proposes a multinodal load forecasting using a multilayer perceptron ANN, with training based on backpropagation, but substituting the usual activation function with a radial basis one. Salgado et al. [26] realize a load demand forecasting by bus using an ANN. In 2011, Nose-Filho et al. [7] have proposed a generic short-term load forecasting model that can be applied to multinodal load forecasting using a general regression neural network (GRNN) [27]. Two other methodologies were used: one predicts loads individually and another uses the participation factors and global load forecasting. For example, in [28], the authors propose a methodology for predicting the electrical current in several transmission lines based on a statistical method. Reference [29] presents a system called MOSAIC that uses a bottom-up simulation tool to determine the current and future consumption and production load curve in an area of the electrical system. Reference [30] introduces the basic principles of an ANN with its training-phase executed by a backpropagation method [4] for predicting the short-term load of a distribution network.

Considering the importance of the mode of load forecasting in several points of the system, such as substations, transformer, and feeders, the objective of this work is to develop a methodology to solve load forecasting problems on the nodes of the system, from the point of generation to the consumer. Therefore, it is a proposal with differentiated characteristics in relation to the specialized literature, i.e., the total load prediction is realized, as well as the evaluation of the load distribution at several interest points according to the user (electrical distribution companies, etc.) based on FANN. The justification was presented previously, where the prediction was executed very fast, always converging and with resources that can be modified by changing the training strategy of the neural network (including or removing components in input and output data), among other tasks that improve the solution quality without needing to retrain the neural network. It is a procedure based on incremental training (continuous learning) [1].

3. Proposed methodology

The multinodal load forecasting method proposed in this work is based on ANN concepts [4] and the GLPF [6,7]. The ANN used is the Fuzzy-ARTMAP [1], the implementation of which is presented in Appendix A.

The prediction system is composed of two important parts: (1) global load forecasting (GLF) and (2) multinodal load forecasting (MLF) (composed by n local prediction modules). Executing the GLF, the MLF is realized using several parallel modules, each predicting the load of a considered bus (transformer, feeder, substation, etc.). Utilizing these models, the proposal of load forecasting is formulated.

3.1. Global load forecasting

The GLF by the ANN is formulated based on window movement [31], a strategy for selecting input variables usually constituted by load values associated with several times (present hour and past hours) and various weather factors. The output corresponds to the next hour's load. The input and output vectors form a window, which is delayed with a velocity of 1 h (or ½ h) until completing the dataset available for the training phase. Once the training is completed (i.e., a successful routine), the ANN is able to realize the prediction. The successful prediction is evaluated on some traditional inference metric, for example, the mean absolute percent error (MAPE) [31].

The formulation presented as follows, i.e., the vector representation (input and output of the ANN), is by line and not by column, as is usually adopted in the literature. This is a recommendation of authors [1] of the ART descendent ANN, where the algebraic operations are easier and more intelligible.

This paper considers the vectors \mathbf{a} and \mathbf{b} containing the input and output data of the ANN, respectively. Taking the reference hour “ h ”, the input vector \mathbf{a} is defined by:

$$\mathbf{a} = [\mathbf{L}\mathbf{T}\mathbf{E}] \quad (1)$$

where:

\mathbf{L} : vector with the load information of hour h and past loads (arbitrated according to strategies adopted by the user);

\mathbf{T} : temperature vector;

\mathbf{E} : vector containing exogenous information (time, weather factors, etc.).

Vector \mathbf{a} is formed by analogic data (vectors \mathbf{L} and \mathbf{T}) and binary data (vector \mathbf{E}). Vector \mathbf{E} includes chronologic data such as time, date, and season among other parameters required by the load predictor problem. As usual, the dimension of the analogic part is inferior to that of the binary part. FANN does not have problems with the dimensions of the input vector, nor with the dimensions of the samples for the training phase. However, if necessary, the dimension of vector \mathbf{E} and, consequently, vector \mathbf{a} , can be reduced using a Fuzzy-ART module (unsupervised training) to form clusters, which are also introduced in the binary codification. This procedure can increase the efficiency of the load prediction calculation. It is also emphasized that the FANN presents the advantage of using a combination of analog and binary data.

In each window, vector \mathbf{L} is composed of the hour load h , as well as three or five correlated components corresponding to past load values. Section 3 (Results and Discussion) presents the strategy used in this paper. For example, the load values corresponding to hours h , $(h - 1)$, $(h - 2)$, and $(h - 3)$ are influenced by measurements on the same day of the previous week and/or previous month. Vector \mathbf{T} usually contains the load temperature, i.e., $\mathbf{T} \in \mathbb{R}^1$. However, this vector can contain the maximum and minimum temperatures of the day, as adopted in some publications, or another strategy of interest to the user.

The ANN output corresponding vector is defined as:

$$\mathbf{b} = [L(h + 1)] \quad (2)$$

where:

$L(h + 1)$ is the total electrical load referred to 1 h (or ½ h) in advance.

It is observed that vector \mathbf{b} has only one component, i.e., $\mathbf{b} \in \mathbb{R}^1$.

3.2. Multinodal load forecasting formulation

The MLF is realized similarly to the global predictor, i.e., for each window, the weight adjustment must be realized as in GLF in the other modules that compose the MLF. In this model the local modules in every bus of the electrical system are parallel, i.e., they can be processed independently.

The following strategy is proposed to improve the training velocity of the local module. The input vector of each module is formed by the load value corresponding to the hour ($h + 1$), i.e., the output value of GLF plus three components corresponding to the global load participation factor ($GFPF$) of hours (h), ($h - 1$), and ($h - 2$). In this case, the dimension is 4. Therefore, the following are definitions of such ANN vectors:

$$\mathbf{a}_p = [L_p(h + 1)GLPF_p(h)GLPF_p(h-1)GLPF_p(h-2)] \quad (3)$$

$$\mathbf{b}_p = [GLPF_p(h + 1)] \quad (4)$$

for $p = 1, 2, \dots, n$.

where:

n : number of interested busses of the power system;

p : identification index of the interested bus;

$GLPF_p$: global load participation factor associated with the p -th bus.

The variable $GLPF$ is defined according to [6] and [7]:

$$GLPF_p(h) = L_p(h) / TL(h) \quad (5)$$

where:

$L_p(h)$: load value of bus p related to hour h ;

$TL(h)$: total load of the system (sum of all electrical loads demanded of the power system) referred to hour h ;

$$= \sum_{p=1}^n \{L_p(h)\}, \quad k = 1, 2, \dots, n. \quad (6)$$

Considering the parameter $GFPF_p$, as defined on Eq. (5), its behavior is less variant (well behaved), in comparison to the behavior of L_p (load corresponding to the p -th bus). Therefore, it is possible to use in each local module (in each considered bus) a lesser number of inputs than the global input vector module (aggregated load). This reduces the processing time in local modules. For example, the experiment presented in Section 4 follows:

$$T_{GLF} = 6.80 \text{ s}$$

$$T_{MLF} = 39.72 \text{ s} (7\text{ANN, that is, } 5.67 \text{ s per substation});$$

where:

T_{GLF} : total execution time of the global ANN training;

T_{MLF} : total execution time of the local ANN corresponding to the mean execution of the several local modules.

3.3. Adjustment and selection of interest nodes

Considering the definition presented in Eq. (5), it is concluded that:

$$\sum_{p=1}^n \{GLPF_p(h)\} = 1. \quad (7)$$

Thus, if n is large, low values of $GLPF_p(h)$ are more likely, which can consequently damage the performance of the ANN training. In this case, a multiplication factor (mf_p) for each $GLPF_p(h)$ is suggested, as the result of this multiplication shows the most appropriate values for the ANN training. When predicting, an adjustment must be made, i.e., the division of $GLPF_p(h)$ by mf_p . For example, the $GLPF$ is considered small if its mean is less than 0.05 (5%). However, it is emphasized that every \mathbf{Ib} (ARTb input vector) has the same size M , and small values of the components are compensated with the concept of complementation (Eq. (A.3)). Therefore, there were no inaccuracies introduced, as already observed in experiments.

This strategy, based on the use of $GLPF$, represents economy of calculus and faster execution, mainly in the training phase. It implements a small variation spectrum around a mean value for every interested node considered. Therefore, the ANN has more stable behavior and converges to a solution more quickly. Fig. 1 shows the typical behavior of a $GLPF$, corresponding to the substations of the system for a 24-h period.

This way, the interested nodes can be selected for the nodal prediction defined within:

$$\mathcal{I} = \{\mathcal{I}_I, \mathcal{I}_A\} \quad (8)$$

where:

\mathcal{I}_I : set of interested nodes;

\mathcal{I}_A : aggregate nodes.

The \mathcal{I}_I are individually treated as associated with the interested multinodal loads. The load related to \mathcal{I}_A corresponds to the sum of every load that does not belong to the \mathcal{I}_I . Therefore, if $\dim\{\mathcal{I}_I\} = m$, then $\dim\{\mathcal{I}\} = m + 1$.

3.4. Load forecasting calculation

Another advantage of the local modules that can be emphasized is that the exogenous variables are dispensable, which are passed to the modules of the MLF by load $L(h + 1)$ (1 h in advance). Thus, the FANN is stable, with short execution time compared to a multilayer feedforward ANN trained by backpropagation [8], which is one of the most commonly used in the literature, as discussed in the Results and Discussion section.

Finally, the predicted values for 24 h are defined by the ANN as:

$$\hat{TL}(h) = b'_1(h) \times L_{base} \quad (9)$$

$$L_p(\hat{h}) = \hat{GLPF}_p(h) \times \hat{TL}(h) \quad (10)$$

for $h = 1, 2, \dots, 24$;

where:

$\hat{TL}(h)$: total predicted load for hour h ;

$b'_1(h)$: ANN output value (global predictor) for hour h ;

L_{base} : total load value taken as base of the normalization process;

$\hat{L}_p(h)$: output value of the p -th ANN (local predictor) corresponding to hour h ;

$\hat{GLPF}_p(h)$: predicted value of the $GLPF$ associated with the p -th local module corresponding to hour h .

The execution training time, which represents the largest part of the execution time considering the parallel prediction system, is defined according to expression (11):

$$T_{total} = T_{GLF} + T_{MLF} < (m + 1)T_{GLF} \quad (11)$$

where:

T_{total} : total execution time of the training predictor system (global and multinodal).

From Eq. (11), it is clear that the total time to execute the FANN (T_{total}) is less than $(m + 1)T_{GLF}$, as the mean execution time of the local modules is less than the global execution time, because the

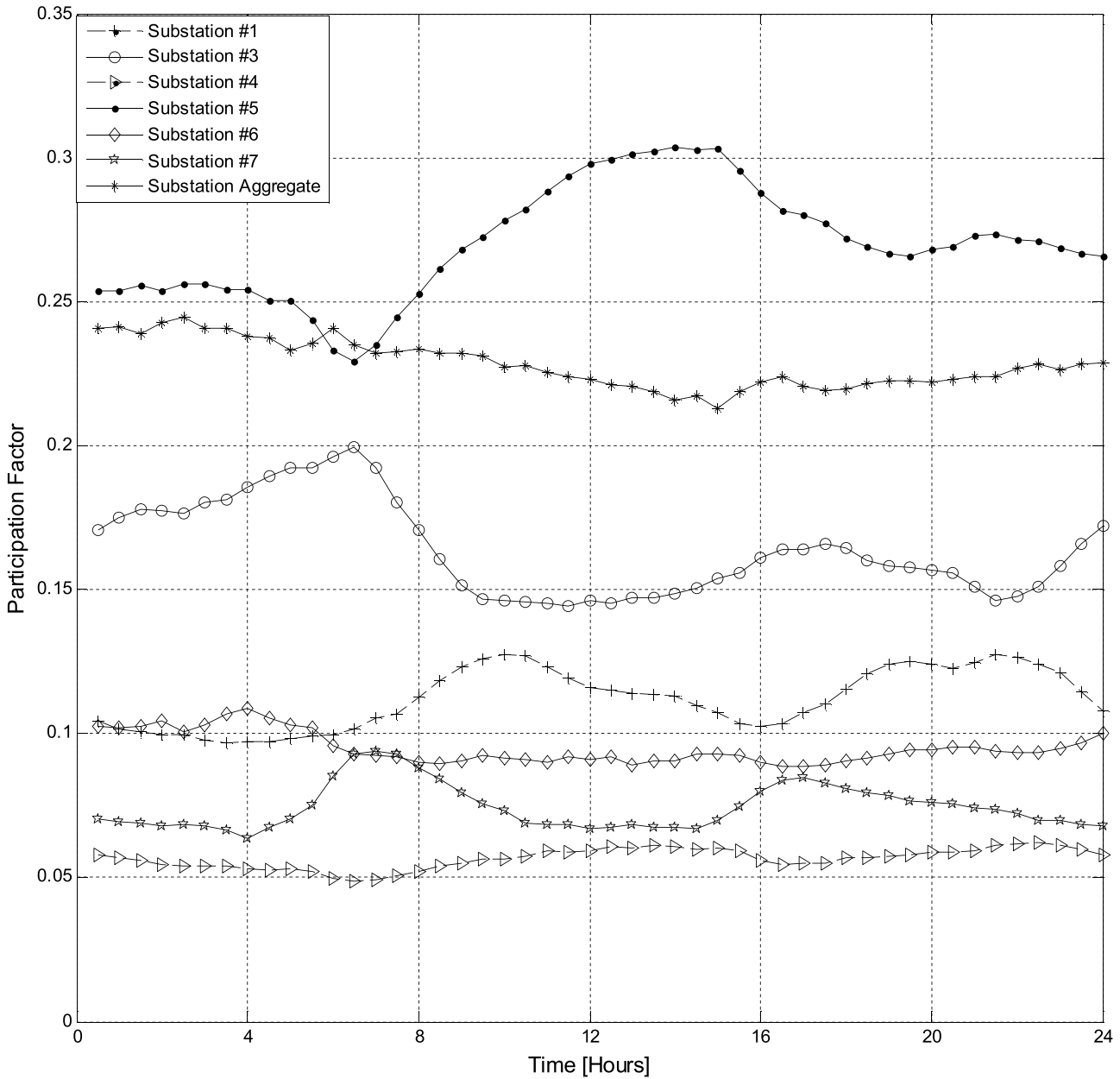


Fig. 1. Typical behavior of a GLPF.

dimension of vector \mathbf{a}_k (Eq. (3)) is less than that of vector \mathbf{a} (Eq. (1)). According to the literature, the FANN is more than a hundred times faster when compared to a multilayer feedforward ANN trained by backpropagation [8], which is the most commonly used method to solve the load forecasting problem.

Fig. 2 shows the load forecasting model as proposed in this paper. The values calculated in this prediction system are the total global load and the normalized loads on the several interested busses. Therefore, these values must be converted into real values (in watts/megawatts).

3.5. Proposed procedure for correction of multinodal forecasting

The first step of the prediction model is obtaining a global prediction. This prediction can then be used to determine the interested node predictions. According to simulations, multinodal predictions present greater error when compared to global predictions. To improve the accuracy of the multinodal prediction, an

algorithm is proposed to adjust the calculations as they are performed.

Step 1: Considering the forecast corresponding to the time $(h + 1)$, obtain the predicted values of: $GLPF_p(h + 1)$, for $p \in \mathcal{J}$;

Step 2: Compute: $sum = \sum_{p \in \mathcal{J}} GLPF_p(h + 1)$;

Step 3: Compute: $error = 1 - sum$;

Step 4: $|error| \leq tol?$ If yes, stop. Else, go to Step 5;

Step 5: $GLPF_p(h + 1) = (1 + error) GLPF_p(h + 1)$, for $p \in \mathcal{J}$. Return to Step 2;

where:

tol is a predefined tolerance.

4. Results and discussion

It is necessary to normalize the results (Eq. (A.1)) in consideration of the definitions of the input and output vectors of module GLF (Eqs. (1) and (2)) of the FANN processing. This is necessary to main-

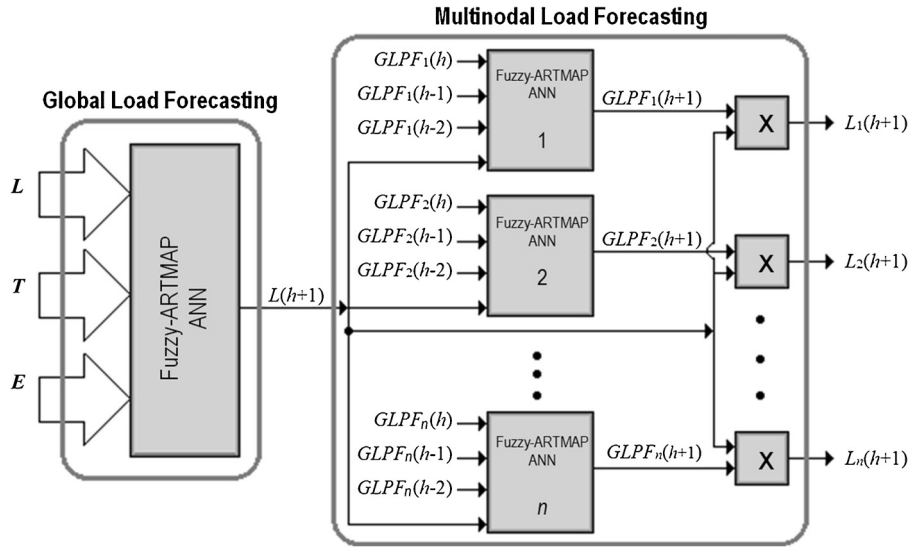


Fig. 2. Load forecasting system.

tain every component of vectors \mathbf{a} and \mathbf{b} pertaining to the interval $[0,1]$. As part of input vector \mathbf{a} corresponds to the binary codification vector \mathbf{E} , the normalization can only be done in relation to the data provided on the global load (vector \mathbf{L}) and temperature (vector \mathbf{T}). For convenience, the loads and temperature must be normalized separately. Therefore, Eq. (A.1) of Step 1, Appendix A, is not necessary once the constraint $a_i \in [0,1]$ is already satisfied. The other vectors \mathbf{b} , \mathbf{a}_k , and \mathbf{b}_k , for $k = 1, 2, \dots, n$, satisfy this constraint as well.

The precision of the load forecast can be evaluated according to the concept of MAPE [31]:

$$MAPE = \frac{1}{NA} \sum_{h=1}^{NA} \left| \frac{y_{actual}(h) - y_{forecast}(h)}{y_{actual}(h)} \right| \times 100 \quad (12)$$

where:

- MAPE: mean absolute percent error;
- NA: number of hourly samples of the time series;
- $y_{actual}(h)$: reference value (comparison) related to hour h ;
- $y_{forecast}(h)$: predicted value at hour h .

To evaluate the performance of the prediction, the maximum error parameter can also be used (E_{max}). These two parameters (MAPE and E_{max}) are the most frequently used in the literature on load forecasting. Additionally, they are used for comparative analysis with the prediction methods.

The database used for the simulations is from a company in New Zealand [10], and is available online. It is a system composed of nine substations. The substations are organized as follows, considering the sets \mathcal{J}_I and \mathcal{J}_A (Eq. (8)):

$$\mathcal{J}_I = \{Sub_1, Sub_3, Sub_4, Sub_5, Sub_6, Sub_7\}$$

$$\mathcal{J}_A = \{Sub_2, Sub_8, Sub_9\}.$$

where:

- Sub_i : i -th substation;
- \mathcal{J}_A : aggregate nodes.

The database is divided into two parts: (1) training and (2) prediction. For the training phase, the period is from December 8-th, 2007 to January 7-th, 2008, forming 1488 vectors with a discretization of 30 min. The prediction (24 h in advance) is defined as January 8-th, 2008. The parameters used in this experiment are shown in Table 1.

Fig. 3 shows the real and predicted global loads (24 h in advance).

Table 1
Parameters of the FANN.

Parameter	Value
β	1
A	0.7
$\rho a_{baseline}$	0.1
ρ_b	0.99
ρ_{ab}	1
E	0.05

Table 2
MAPEs.

Substation	MAPE (%)
Sub_1	4.99
Sub_3	3.42
Sub_4	5.63
Sub_5	4.76
Sub_6	4.75
Sub_7	3.31
$Sub_{aggregate} (Sub_2, Sub_8, Sub_9)$	3.85
Global	2.91

Figs. 4 and 5 show the local load prediction (substations) with Sub_7 and $Sub_{aggregate}$, respectively, taken as examples.

Table 2 shows the MAPE for global and local predictions.

According to the results presented in previous literature, the MAPEs are satisfactory (less than 5%, except for Sub_4 , which is slightly higher at 5.63%).

This experiment has been realized using a Dell XPS PC with a Core i7-4770 Intel processor @3.40 GHz and 8.0 GB RAM, running a Microsoft Windows 64-bit operating system

We consider that the tests presented in this work show that this proposal is promising, in addition to being fast and stable, and provides good results with precision similar to those obtained by backpropagation neural networks (BNNs) (benchmark in precision, though very slow).

According to reference [32], a classical experiment elaborated in reference [33] (two spirals apart), a BNN needed 20,000 epochs to conclude the training, whereas, in relation to the same experiment, the FANN was trained with only five epochs. This is obviously a significant difference with respect to processing time, in favor of the FANN. Another advantage is its plasticity; for example, we can use new data and change the structure, such as by increasing or reduc-

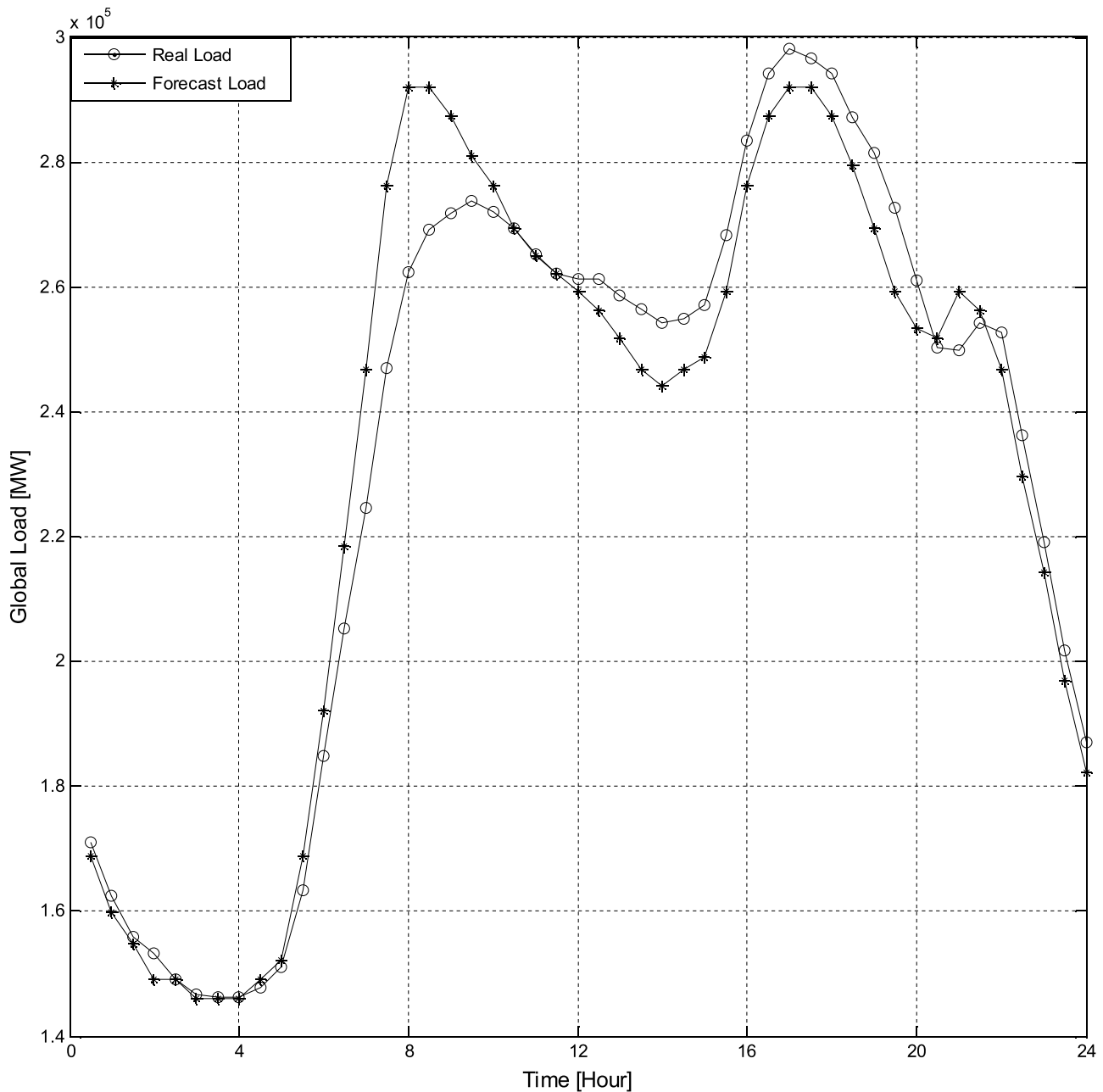


Fig. 3. Global load (real and predicted).

ing the quantity of input data components, without destroying the knowledge previously acquired. It is emphasized that most of the neural networks available in the literature do not exhibit plasticity.

Other comparative studies, considering different ANNs, have been presented in the specialized literature with this emphasis, e.g., in [34–37]. The results from the comparative studies show that the ANN model is a superior method for load forecasting, owing to its ability to handle load data and its lower MAPE. These publications also emphasize the advantages of applying neural networks in relation to classical methods. The principal advantage is the lack of necessity to model the load, i.e., it is not necessary to determine a correlation between the cause (input data) and effect (output, which is the load forecasted one hour in advance). Neural networks can use the quantity and quality of several types of information, once the importance (sensitivity) of the load to be predicted on the data used in input is determined automatically.

Moreover, the sensitivity can be extracted by neural networks [38] after concluding the training phase with the values of the internal parameters of the neural network (from the information field of synapse weights and nonlinearities). This procedure can be used to select important (more sensitive) input variables and eliminate others that contribute little or not at all to obtaining a good load forecasting. Considering this assumption, we think that the proposal is innovative.

5. Conclusion

This paper presents a new prediction model for load prediction up to 24 h in advance, based on the use of a FANN. The load forecasting is constituted by global prediction, i.e., considering the sum of all loads on the system, as well as in each node (buses of the electrical system). It is achieved without additional com-

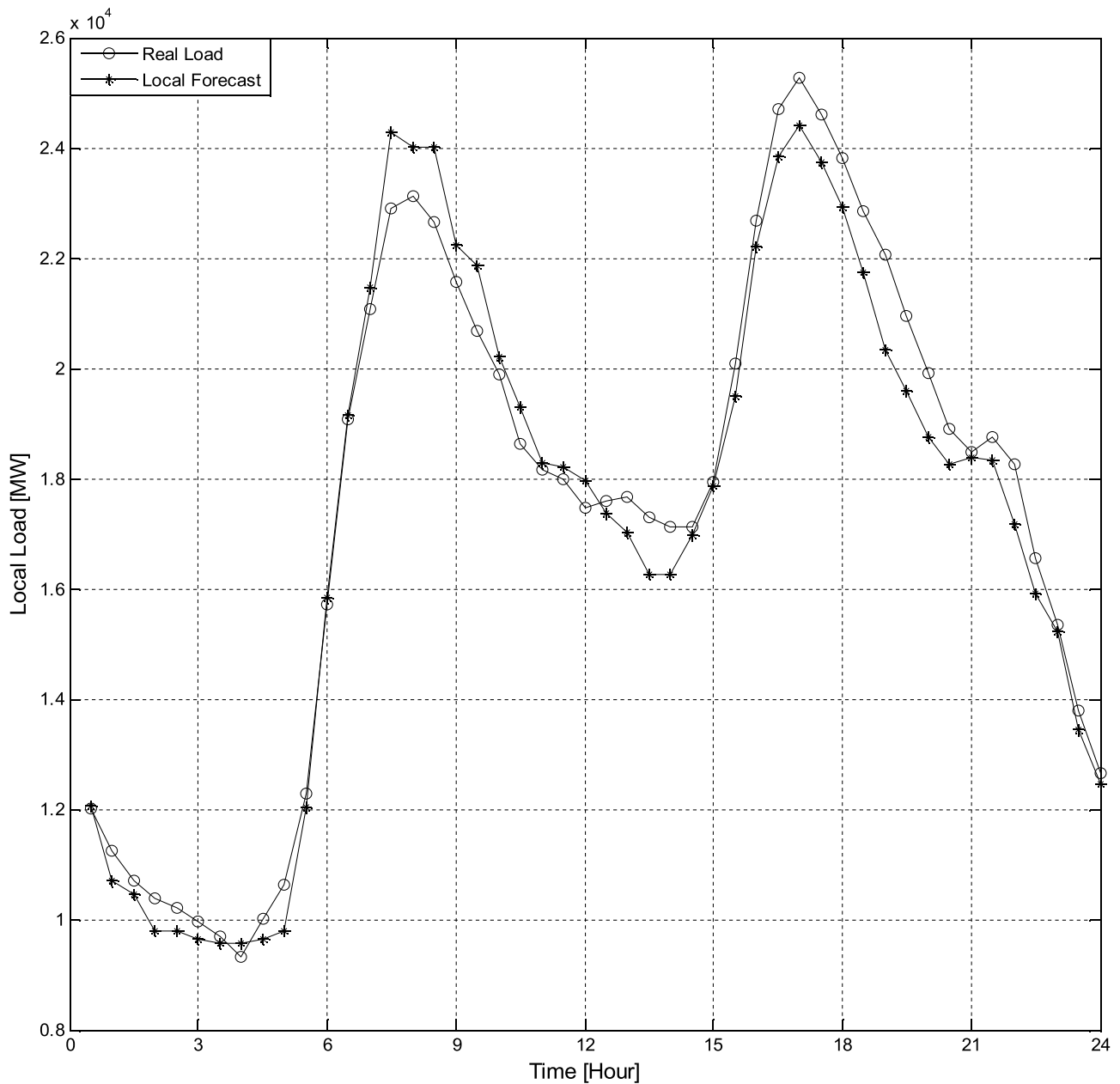


Fig. 4. Local load prediction associated with substation *Sub₇*.

putational processing time costs, as the modules are processed in parallel. A reduced quantity of data is used to execute the local node training, thus reducing the time for calculating each local module. The total training execution time is half of that required for global prediction training. The time for training the FANN is substantially less than that required for traditional backpropagation training [8], which is considered a benchmark in the literature. The advantages of the FANN are its quality in terms of stability, plasticity, and results that satisfy the exigencies of the users. The use of a GLPF presents a better behavior (less variation around a mean value) than the corresponding local load. This increases the precision of the results, providing errors of generally less than 5% (similar to those presented in the literature), with only the error for *Sub₁* above this threshold. Another improvement is the adjustment procedure of the GLPF based on the application of *pro rata* corrections. This concept is intended to demonstrate that it is possible to realize adjustments without prejudicing the results and encourage

new upgraded algorithms. One of the qualities of the ART family architecture (including the FANN) is that application-tailored adjustments can be made to provide the best results.

Thus, we highlight the following advantages and disadvantages of the methodology proposed for multinodal load forecasting. The advantages include fast training, as well as ease of changing the necessary strategies (different from those already established in the literature) to improve the results or including new data without destroying the previously acquired knowledge (incremental training). Another advantage is that the FANN architecture has no stability problems when employing a large database, which is a problem for most of the neural networks available in the literature; i.e., this neural network (FANN) always converges. The disadvantages include that to obtain high-quality solutions, it is necessary to elaborate a good plan of the input and output variables of the neural network, mainly owing to the use of binary codification with analogic quantities. Moreover, using FANN, the quantity of com-

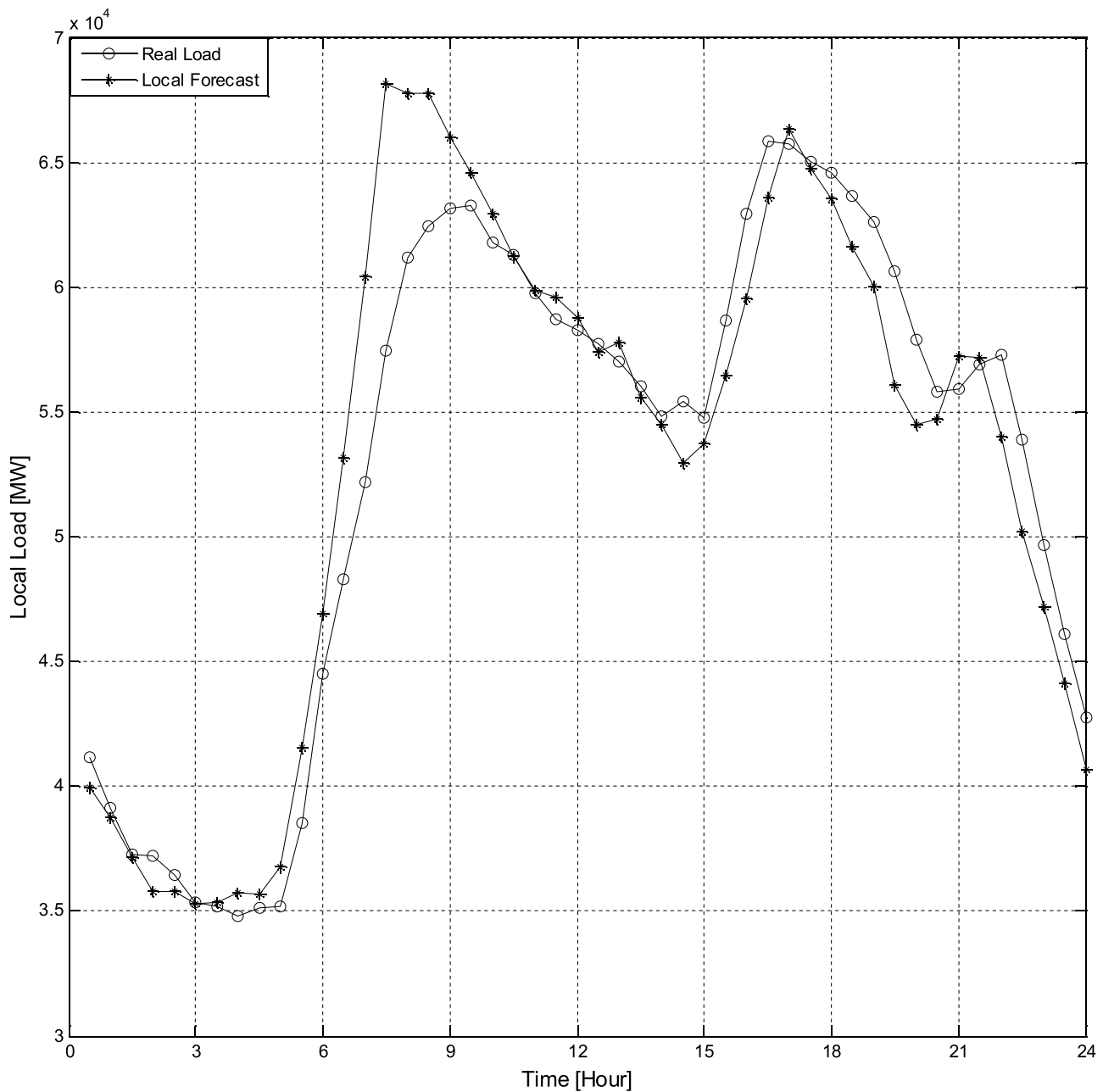


Fig. 5. Local load prediction referred to $Sub_{aggregate}$.

ponents of the input vector is doubled (see Eq. (A.4) – Appendix A) in comparison with those of other neural networks. Finally, we emphasize that considering the field of load forecasting, it is possible to investigate new techniques based on ANN, such as reservoir computing [39] and deep learning [40–42].

Acknowledgment

The authors thank CAPES (Brazilian Research Foundation) and CNPq (National Council for Scientific and Technological Development) for financial support.

Appendix A. A – fuzzy-artmap neural network

FANN is a combination of two techniques: adaptive resonance theory [1] and fuzzy logic [5]. It is constituted by two unsupervised

Fuzzy ART ANNs interconnected by a module called inter-ART. The neural modules ARTa and ARTb process the input vector (vector \mathbf{a}) and output (vector \mathbf{b}), respectively. The inter-ART module realizes the matching (input–output mapping) with the two modules ARTa and ARTb providing the extraction of the knowledge (training or learning) from the FANN, in a supervised mode, i.e., it is based on input and output stimuli. Fuzzy logic is aggregated to the ARTMAP architecture to improve the solutions, which provides an intelligent increment to the neural architecture. The input and output of the FANN must be pre-processed considering two principal objectives: (1) normalization, i.e., each component of the input and output vectors must be between 0 and 1; (2) the quality of the solutions is dependent on the ability of the user, and on the codification of the components \mathbf{a} and \mathbf{b} .

Fuzzy-ART and FANN work according to the following phases: (1) training phase; and (2) analysis phase (prediction). The training phase is realized offline with the presentation of the input stimulus

(for unsupervised training) and input and output (supervised training). Such stimuli are elaborated from examples extracted from a database and/or simulations. Three layers constitute Fuzzy-ART modules: F_0 (input layer), F_1 (comparison layer), and F_2 (recognition layer for category filing), implemented according to the following steps:

Step 1: Input Data (Stimulus)

The input data are vector $\mathbf{a} = [a_1 a_2 a_3 \dots a_M]$, with dimension M . This vector is normalized to avoid a proliferation of categories (beyond the minimum necessary), therefore:

$$\bar{\mathbf{a}} = \mathbf{a} / \|\mathbf{a}\|_1 \tag{A.1}$$

where:

$\bar{\mathbf{a}}$: normalized input vector;

$$\|\mathbf{a}\|_1 = \sum_{i=1}^M |a_i| \text{ (norma - 1)}. \tag{A.2}$$

The application of an ART descendent ANN combined with Fuzzy Logic (the case addressed in this work), needs the variables and parameters pertaining to the interval $[0,1]$. Therefore, if these components originally satisfy this restriction, the execution of Step 1 could be dispensable, or at least optional.

Step 2: Codification of the Input Vector

The codification, called complement, is realized to preserve the magnitude of the information, i.e.:

$$\bar{a}_i^c = 1 - \bar{a}_i \tag{A.3}$$

where:

\bar{a}_i^c : i -th component of input normalized complementary vector.

Therefore, the dimension of this vector is equal to $2M$:

$$\mathbf{I} = [\bar{\mathbf{a}} \ \bar{\mathbf{a}}^c] = [\bar{a}_1 \ \bar{a}_2 \ \dots \ \bar{a}_M \ \bar{a}_1^c \ \bar{a}_2^c \ \dots \ \bar{a}_M^c] \tag{A.4}$$

$\|\mathbf{I}\|_1 = M$ (every vector with normalization and complementary codification has the same size M).

Step 3: Activity Vector

The activity vector of F_2 is represented by $\mathbf{y} = [y_1 y_2 y_3 \dots y_N]$, where N is the quantity of categories created in F_2 . Thus:

$$y_i = 1 \text{ if node } i \text{ of } F_2 \text{ is active; otherwise it is } 0. \tag{A.5}$$

Step 4: Selection of Neural Network Parameters

The parameters used on the ART-Fuzzy ANN processing are:

1. Choice Parameter: $\alpha > 0$;
2. Training Rate: $\beta \in [0,1]$;
3. Vigilance Parameter : $\rho \in [0,1]$.

Step 5: Weight initializing

First, every weight is equal to 1:

$$w_{j1}(0) = w_{j2}(0) = w_{j3}(0) \dots = w_{j2M}(0) = 1 \tag{A.6}$$

This shows that there is no previously assumed active category.

Step 6: Chosen Category

Considering the input vector \mathbf{I} in F_1 , for each node j in F_2 , the chosen function T_j is determined by:

$$T_j = \|\mathbf{I} \wedge \mathbf{w}_j\|_1 / (\alpha + \|\mathbf{w}_j\|_1) \tag{A.7}$$

where:

\wedge : fuzzy operator AND defined by:

$$(\mathbf{I} \wedge \mathbf{w})_i = \min(I_i, w_i), \quad i = 1, 2, \dots, N. \tag{A.8}$$

The category is chosen corresponding to the active J node, i.e.:

$$J = \arg\{\max T_j\}, j = 1, 2, \dots, N. \tag{A.9}$$

Observing Eq. (A.9), if there is more than one active category, that with the minimum index will be chosen.

Step 7: Resonance or Reset

Resonance occurs if the vigilance criterion (Eq. (A.10)) is satisfied:

$$\|\mathbf{I} \wedge \mathbf{w}_J\|_1 / \|\mathbf{I}\|_1 \geq \rho \tag{A.10}$$

If the criterion defined by Eq. (A.10) is not satisfied, the device called reset is set. At the reset, node J of F_2 is excluded from the searching process (according to Eq. (A.9)), i.e., one must set $T_J = 0$. Then, a new category is chosen using Eq. (A.9) in the resonance process. This procedure is repeated until a category satisfying Eq. (A.10) is found.

Step 8: Weight Updating (Training)

When input vector \mathbf{I} completes the resonance state, the weights of the Fuzzy-ART ANN are adjusted according to:

$$\mathbf{w}_J^{\text{new}} = \beta (\mathbf{I} \wedge \mathbf{w}_J^{\text{old}}) + (1 - \beta) \mathbf{w}_J^{\text{old}} \tag{A.11}$$

where:

J : index of active category;

$\mathbf{w}_J^{\text{new}}$: updated weight vector;

$\mathbf{w}_J^{\text{old}}$: weight vector referred to the previous updating.

It is emphasized that if $\beta = 1$, the training is faster.

Implementing the FANN, the execution of modules ARTa and ARTb follows the same procedure (Steps 1–8), associating, respectively, the indices $j(J)$ and $k(K)$, *mutatis mutandis*. The indices J and K show the winner neurons in ARTa and ARTb, respectively. At module ARTb, the parameters α_b , β_b , and ρ_b are arbitrated based on experience. With respect to module ARTa, the algorithm is the same, where α_a and β_a are considered fixed and ρ_a is variable, taking the following value:

$$\rho_{a \text{ baseline}} = \|\mathbf{I} \wedge \mathbf{w}_J^a\|_1 / \|\mathbf{I} \wedge \mathbf{I}\|_1 \tag{A.12}$$

The parameter ρ_a is incremented from the initial value (A.12), by a small positive value (ε) until the value of ρ_a does not satisfy Eq. (A.10). The last value of ρ_a that satisfies Eq. (A.10) is the correct value of $\rho_{a \text{ max}}$, which corresponds to the matching of the categories of the input and output patterns. The parameter ρ_a , between $\rho_{a \text{ baseline}}$ and $\rho_{a \text{ max}}$, can be run as few iterations, according to Carpenter and Grossberg (1992). The input–output pairs (\mathbf{a}, \mathbf{b}) , associated to the modules ARTa and ARTb, are confirmed (the inputs \mathbf{a} and \mathbf{b} are referred to the categories J and K active, respectively) with the test of match tracking:

$$(x^{ab})_i = \|\mathbf{y}_i \wedge \mathbf{w}_J^{ab}\|_1 / \|\mathbf{y}_i\|_1 \tag{A.13}$$

$(x^{ab})_i \geq \rho_{ab} \rightarrow$ the training pair (\mathbf{a}, \mathbf{b}) must be confirmed on the weight matrices with indices J and K .

$(x^{ab})_i < \rho_{ab} \rightarrow$ one must search another index J (in relation to the input vectors \mathbf{I}_a), maintaining the index K , until the resonance criterion is satisfied. Where:

ρ_{ab} : vigilance parameter of module inter-ART.

The weights \mathbf{w}^a and \mathbf{w}^b must be adapted using Eq. (A.11). In relation to the weights \mathbf{w}^{ab} , the procedure follows:

$$w_{JK}^{ab \text{ new}} = 1 \tag{A.14}$$

$$w_{jk}^{ab \text{ new}} = 0, \text{ for } k \neq K. \tag{A.15}$$

References

- [1] S. Grossberg, Adaptive resonance theory: how a brain learns to consciously attend, learn, and recognize a changing world, *Neural Networks* 37 (2013) 1–47, <http://dx.doi.org/10.1016/j.neunet.2012.09.017>.
- [2] G.E. Box, G.M. Jenkins, *Times Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, USA, 1976.
- [3] T.M. O'Donovan, *Short Term Forecasting: An Introduction to the Box-Jenkins Approach*, John Wiley & Sons, New York, 1983.
- [4] S. Haykin, *Neural Networks and Learning Machines*, 3rd edition, Prentice-Hall, New Jersey, USA, 2008.

- [5] L.A. Zadeh, A. Fuzzy Sets, *Advances in Fuzzy Systems – Applications and Theory*, vol. 6, 1965, pp. 394–432, <http://dx.doi.org/10.1142/9789814261302-0021>.
- [6] A.B. Altran, *Intelligent System for Multinodal Load Forecasting of Electrical Power Systems*, Ph.D. Thesis, Electrical Engineering Department, UNESP – São Paulo State University, SP, Brazil, 1999, February (In Portuguese).
- [7] K. Nose-Filho, A.D.P. Lotufo, C.R. Minussi, Short-term multinodal load forecasting using a modified general regression neural network, *IEEE Trans. Power Deliv.* 26 (October (4)) (2011) 2862–2869, <http://dx.doi.org/10.1109/TPWRD.2011.2166566>.
- [8] P.J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Thesis, Harvard University, 1974.
- [9] H. Zhang, Z. Wang, D. Liu, Comprehensive review of stability analysis of continuous-time recurrent neural networks, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (7) (2014) 1229–1262, <http://dx.doi.org/10.1109/TNNLS.2014.2317880>.
- [10] CDS-Centralized Dataset, *Electricity Commission for the Load Dataset*, New Zealand, 2010.
- [11] J.W. Taylor, R. Buizza, Neural network load forecasting with weather ensemble predictions, *IEEE Trans. Power Syst.* 17 (3) (August 2002) 626–632, <http://dx.doi.org/10.1109/TPWRS.2002.800906>.
- [12] S.C. Pandian, K. Duraiswamy, C.C.A. Rajan, N. Kanagaraj, Fuzzy approach for short term load forecasting, *Electr. Power Syst. Res.* 76 (6–7) (2006) 541–548, <http://dx.doi.org/10.1016/j.epsr.2005.09.018>, Lausanne.
- [13] G.C. Liao, T.P. Tsao, Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting, *IEEE Trans. Evol. Comput.* 10 (June (3)) (2006) 330–340, <http://dx.doi.org/10.1109/TEVC.2005.857075>.
- [14] C. Kocak, ARMA(p,q) type high order fuzzy time series forecast method based on fuzzy logic relations, *Appl. Soft Comput.* 58 (2017) 92–103, <http://dx.doi.org/10.1016/j.asoc.2017.04.021>.
- [15] G. Gross, F.D. Galiana, Short-term load forecasting, *Proc. IEEE* 75 (12) (1987) 1558–1573, <http://dx.doi.org/10.1109/PROC.1987.13927>.
- [16] I. Moghram, S. Rahman, Analysis and evaluation of five short-term load forecasting techniques, *IEEE Trans. Power Syst.* 4 (4) (1989) 1484–1491, <http://dx.doi.org/10.1109/59.41700>, New York.
- [17] A.A. El Desouky, M.M. El Kateb, Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA, *Proc. Gener. Transm. Distrib.* 147 (4) (2000) 213–217, <http://dx.doi.org/10.1049/ip-gtd:20000521>.
- [18] M.L.M. Lopes, C.R. Minussi, A.D.P. Lotufo, Electric load forecasting using a fuzzy ART&ARTMAP neural network, *Appl. Soft Comput.* 5 (January (2)) (2005) 235–244, <http://dx.doi.org/10.1016/j.asoc.2004.07.003>.
- [19] M.L.M. Lopes, A.D.P. Lotufo, C.R. Minussi, Application of the fuzzy ART&ARTMAP neural network to the electrical load forecasting problem, in: J.J. Zhu, G.P.C. Fung (Eds.), *Forecasting Models: Methods and Applications*, Vol. 1, iConcept Press Pty Ltd., 2010, pp. 179–190.
- [20] L.M. Saini, M.K. Soni, Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods, *IEEE Proc. Gener. Transm. Distrib.* 149 (5) (2002) 578–584, <http://dx.doi.org/10.1049/ip-gtd:20020462>, New York.
- [21] H.A. Malki, N.B. Karayiannis, M. Balasubramanian, Short-term electric power load forecasting using feedforward neural networks, *Expert Syst.* 21 (3) (2004) 157–167, <http://dx.doi.org/10.1111/j.1468-0394.2004.00272.x>, New York.
- [22] E.H. Barakat, M.A.M. Eisa, Forecasting monthly peak demand in fast growing electric utility using a composite multiregression-decomposition model, *IEE Proc. C-Gener. Transm. Distrib.* 136 (1) (1989) 35–41, <http://dx.doi.org/10.1049/ip-c.1989.0006>.
- [23] C.S. Chen, Y.M. Tzeng, J.C. Hwang, The application of artificial neural networks to substation load forecasting, *Electr. Power Syst. Res.* 38 (2) (1996) 153–160, [http://dx.doi.org/10.1016/S0378-7796\(96\)01077-2](http://dx.doi.org/10.1016/S0378-7796(96)01077-2).
- [24] C.C. Hsu, C.Y. Chen, Regional load forecasting in Taiwan—applications of artificial neural networks, *Energy Convers. Manag.* 44 (2002) 1941–1949, [http://dx.doi.org/10.1016/S0196-8904\(02\)00225-X](http://dx.doi.org/10.1016/S0196-8904(02)00225-X).
- [25] N.K. Singh, A.K. Singh, N. Paliwal, Neural network based short-term electricity demand forecasting for Australian states, in: *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, 2016, pp. 1–4, <http://dx.doi.org/10.1109/ICPEICES.2016.7853603>.
- [26] R.M. Salgado, T. Ohishi, R. Ballini, A short-term bus load forecasting system, in: *10-th International Conference on Hybrid Intelligent Systems (HIS)*, Atlanta, GA, USA, 2010, pp. 55–60, <http://dx.doi.org/10.1109/HIS.2010.5600075>.
- [27] D.F. Specht, A general regression neural network, *IEEE Trans. Neural Netw.* 2 (November (6)) (1991) 568–576, <http://dx.doi.org/10.1109/72.97934>.
- [28] T. Nagumo, H. Ito, T. Sano, Load current forecasting using statistical, analysis, 24th International Conference & Exhibition on Electricity Distribution (CIRED), June 2017 (2017) 1274–1277, <http://dx.doi.org/10.1049/oap-cired.2017.0304>.
- [29] N. Kong, M. Bocquel, T. Barbier, R. Girard, E. Magliaro, G. Kariniotakis, G. Pelton, P. Cauchois, Long-Term Forecast of Local Electrical Demand and Evaluation of Future Impacts on the Electricity Distribution Network, June 2017 (2017) 2401–2405, <http://dx.doi.org/10.1049/oapcired.2017.0743>.
- [30] Y. Liu, D. Li, H. Pei, K. Liu, Y. Li, L. Yang, Short-term load prediction method for power distributing method based on back-propagation neural network, 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), June 2017 (2017) 881–886, <http://dx.doi.org/10.1109/ICIEA.2017.8282964>.
- [31] D.C. Park, M.A. El-Sharkawi, R.J. Marks II, L.E. Atlas, M.J. Damborg, Electric load forecasting using an artificial neural networks, *IEEE Trans. Power Syst.* 6 (2) (1991) 442–449, <http://dx.doi.org/10.1109/59.76685>.
- [32] G.A. Carpenter, S. Grossberg, D.B. Rosen, ART: fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Netw.* 4 (1991) 759–771.
- [33] K.J. Lang, M.J. Witbrock, Learning to tell two spirals apart, Touretzky, in: G. Hinton, T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann Publishers, 1988, pp. 52–59.
- [34] B.-g. Koo, S.-W. Lee, W. Kim, J.H. Park, Comparative study of short-term electric load forecasting, 2014 Fifth International Conference on Intelligent Systems, Modelling and Simulation (2014) 463–467, <http://dx.doi.org/10.1109/ISMS.2014.85>.
- [35] N.M.G. Kumar, P.S.S. Raju, P. Venkatesh, Load forecasting for practical power systems by using artificial neural networks and classical techniques—a case study, *Int. J. Comput. Eng. Res.* 3 (5) (2013) 56–65.
- [36] P. Laureta, M. David, D. Calogine, Nonlinear models for short-time load forecasting, *Energy Procedia* 14 (2012) 1404–1409, <http://dx.doi.org/10.1016/j.egypro.2011.12.1109>.
- [37] S. Dwijayanti, M. Hagan, Short term load forecasting using a neural network based time series approach, *First International Conference on Artificial Intelligence, Modelling & Simulation*, 2013 (2013) 12–17, <http://dx.doi.org/10.1109/AIMS.2013.11>.
- [38] A.D.P. Lotufo, M.L.M. Lopes, C.R. Minussi, Sensitivity analysis by neural networks applied to power systems transient stability, *Electr. Power Syst. Res.* 77 (2007) 730–738, <http://dx.doi.org/10.1016/j.epsr.2005.09.020>.
- [39] A. Goudarzi, C. Teuscher, Reservoir computing: quo vadis? *Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication* (2016) 1–6, <http://dx.doi.org/10.1145/2967446.2967448>.
- [40] F. Ghasemi, A. Mehridehnavi, A. Fassihi, H. Pérez-Sánchez, Deep neural network in QSAR studies using deep belief network, *Appl. Soft Comput.* 62 (2018) 251–258, <http://dx.doi.org/10.1016/j.asoc.2017.09.040>.
- [41] S. Ryu, J. Noh, H. Kim, Deep neural network based demand side short term load forecasting, *Energies* 10 (3) (2017) 1–20, <http://dx.doi.org/10.3390/en10010003>.
- [42] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning: Adaptive Computation and Machine Learning Series*, MIT Press Book, 2016, ISBN: 9780262035613.