Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Computational Intelligence Society, Vol. 11, Iss. 2, pp. 123-152, 2013

TRAINING A NEURAL NETWORK TO DETECT PATTERNS ASSOCIATED WITH SEVERE WEATHER

Glauston R. T. de Lima
Stephan Stephany

Instituto Nacional de Pesquisas Espaciais - INPE
Laboratório Associado de Computação e Matemática Aplicada - LAC
e-mails: glau11@gmail.com, stephan@lac.inpe.br
Avenida dos Astronautas 1758, Jardim da Granja
CEP 12227-010, São José dos Campos, SP, Brasil

Abstract - Early detection of severe convective events is essential to take preventive measures in order to reduce the related negative impacts. However, the increasing amount of data that meteorologists need to analyze nowadays precludes the chances of forecasting such events. Hence, the development of advanced data mining tools to support weather forecasting became a current research topic in Meteorology. This work presents the results obtained with an artificial neural network that was designed to recognize atmospheric patterns associated to severe convective activity. The network was trained with patterns given by a set of selected meteorological variables values of the numerical weather forecasting model Eta. The classes of atmospheric convective activity are defined by values of density of occurrence of atmospheric electrical discharges. It is assumed that such density can be correlated to the level of convective activity. The network architecture and training algorithm are based on heuristics derived from previous studies of the authors. Once validated, the proposed neural classifier would be able to screen the Eta model outputs and assist meteorologists in the early detection of severe convective events. In addition, it would also assess the Eta model skill to predict such events. In a first approach, only two data classes are considered, corresponding to severe convective activity and to moderate/weak/absent convective activity. The convenience of defining more classes will be evaluated in further work. The classification results are promising.

Keywords - supervised neural network, data mining, classification, convective activity, weather forecasting.

1 Introduction

Atmospheric severe convective events result in heavy rainstorms of mid or long term with potential for flooding and other impacts. In recent years, these phenomena became frequent causing enormous environmental and socio-economic damages in several Brazilian regions. Brazil is one of the world's largest producers of food and some crops are very sensitive to this kind of events. Moreover, severe convective events are the leading cause of deaths due to climate-related phenomena in the country. Therefore, an early and semi-automatic tool for the detection of severe convective events would be useful to assist meteorologists in order to avoid, or at least mitigate such negative impacts.

However, the enormous amount of meteorological data currently available provided by sensors like those on-board of satellites, and numerical simulations with increasing spatial and temporal resolutions, makes it difficult, if not impossible, its prompt analysis by meteorologists. In addition, severe convective events are related to physical atmosphere processes whose genesis and evolution are sometimes very fast making even more difficult their early and accurate prediction. In this scenario, the meteorologist could be assisted by advanced tools able to analyze data and provide support in decision-making concerning weather forecasting. The development of methodologies for this purpose became a current topic of research in Meteorology.

In this scope, this paper presents a new artificial neural network that was designed to recognize atmospheric patterns associated to severe convective events. The supervised training of the proposed network is done using patterns associated to different levels of convective activity. These patterns are vectors containing the values of 26 meteorological variables that were selected among the hundreds of meteorological variables generated by the regional meteorological model Eta [1]. The model outputs such variables every six hours of simulated time (6, 12, 18 and 24 hours) for the considered spatial grid. This reduced set of 26 variables was selected with the help of meteorologists and climatologists. The class of convective activity for a given time and point of the grid is assigned according to the corresponding value of density of occurrence of cloud-to-ground atmospheric

electrical discharges (henceforth referred to as density of discharges). In this work, it is assumed that such density can be correlated to the level of convective activity. In a first approach, only two data classes are considered corresponding to severe convective activity (referred to as **SCA** class) and moderate/weak/absent convective activity (referred to as **NSCA** class). The convenience of defining more classes will be evaluated in further work.

The network architecture and training algorithm are based on heuristics proposed in previous studies of the authors about classification approaches in the same scope of identifying patterns associated to atmospheric severe convective activity [2]. These studies also employed a meteorological database composed of Eta model and density of discharges data. As it will be discussed in Section 3, there was a strong overlap of classes that required the exploitation of the information embedded in specific subgroups of the 26 Eta variables in order to distinguish **NSCA** patterns from **SCA** patterns.

The use of neural networks in weather forecasting goes back to the 60's when a Widrow's adaptive linear network was used for this purpose [3]. This pioneering work yielded limited results due to lack of more sophisticated multilayer network training algorithms at that time. Since then, numerous applications of networks in weather forecasting have been reported employing different meteorological variables, such as temperature, wind speed components, humidity, geopotencial height or rainfall estimation. For instance, different neural paradigms like Multi-Layer Perceptron (MLP) and Kohonen Self Organized Maps (KSOM) were used separately [4, 5] or jointly [6] for atmospheric temperature prediction from temporal series data. The latter work demonstrated that neural networks may have sensitivity to predict seasonal temperature variations. In another work [7], the performance of MLP, Radial Basis Functions (RBF), Hopfield and Elman recurrent neural networks are tested and compared to each other and also compared with one ensemble of the same networks in predicting temperature, wind speed and relative humidity. It was concluded that the ensemble had the best performance, but individually, the RBF performance was better. A neural network was also used for climate change prediction in order to forecast phenomena like El Niño that affect the global climate [8]. A weather forecast system was modeled in [9] employing also a neural network to estimate the values of some meteorological variables. Prediction of heavy rainfall or extreme temperatures are addressed in [10] using a modified version of the time-delay neural network. A performance comparison between MLP e RBF in rainfall prediction is made in [11] and Bayesian Networks are used in [12] for the same purpose. Models for rainfall forecast have been also addressed in [13, 14, 15] evaluating the performance of neural networks with different training algorithms and data. In [16], neural networks are used in order to improve the forecast accuracy concerning thunderstorms development and location. In order to perform a climatic prediction of atmospheric temperature based on a temporal series, a hybrid approach with neural network, fuzzy logic and genetic algorithms was used in [17]. Support vector machine (SVM) was employed in [18] to predict climatic changes applied in Agrometeorology. SVM is also applied to predict daily maximum temperatures using a temporal series in [19] and also [20].

Our proposal of detecting severe convective activity patterns from a numerical weather forecast model is new and the related research started a few years ago [27]. However, in an attempt to improve the classification performance of our former classifier and take advantage of the potential of neural networks to solve data mining and data modeling problems [21, 22, 23] we started this work. The neural network approach adopted in this work provided good classification performance for three selected mini-regions of Brazil using data corresponding to two summer months of 2007-2011, as discussed in section 5. The final goal of this research is to develop a set of neural network-based classifiers that are specific for different mini-regions and different seasonalities. Once validated, the neural network presented in this paper would be able to screen the outputs of the meteorological model Eta. It will eventually represent an ancillary tool for meteorologists in the early identification of atmospheric conditions associated to the occurrence of severe convective events.

The remainder of the paper is organized as follows: Section 2 describes the meteorological database used in the training and test phases of the proposed neural network; Section 3 summarizes the previous data mining approaches developed by the authors to identify severe convective patterns, Section 4 shows the architecture and training algorithm of the proposed neural network while Section 5 presents the classification results and discussion. Finally, Section 6 are the concluding remarks.

2 Meteorological Database

The meteorological database employed in this work includes output data of the numerical weather forecast mesoscale model Eta that performs 24 to168-hour forecasts. The boundary conditions of this model are given by a global numerical model and

observational data. The outputs refer to the months of January and February of five years, 2007-2011 (total of 59 days per year, except for 2008 that has 60 days) for an area of 20° by 20° bounded by the coordinates (15°S to 35°S) and (40°W to 60°W). This corresponds to a square of 101 by 101 grid points, considering the resolution of 0.2° (≈ 20 km) of the model, and covers the Midwest, Southeast and Southern of Brazil. The Eta data outputs are daily and for every 6 hours (6, 12, 18 and 24 hours) for all points of the considered grid. Such data includes hundreds of variables and some of them are given for several levels of atmospheric pressure. A reduced set of 26 variables was defined with the help of meteorologists and climatologists, based on its relevance for the detection of severe convective events. It is shown in Table 1. Therefore, the patterns used in the training of the neural network are these 26-variable vectors. Each vector is related to a particular grid point and time.

In previous studies, we adopted the full 20° by 20° square, but such an extense area presents different climate conditions like moisture or temperature, and hampered the classifiers to perform well. Therefore, in order to make feasible the training of the neural network we selected three mini-regions of 1° by 1° (containing 36 grid-points) each one with a homogeneous microclimate. These selected mini-regions, denoted as mini-regions A, B, and C, are defined by the following coordinates:

mini-region A: (18.5°S to 19.5°S), (56.5°W to 57.5°W)

mini-region B: (21.5°S to 22.5°S), (49.5°W to 50.5°W)

mini-region C: (23.0°S to 24.0°S), (45.0°W to 46.0°W)

The full 20° by 20° area would provide a number of training vectors given by 59 x 4 x 101 x 101 (about 2.5 million vectors). On the other hand, each particular 36 grid-point mini-region provides only 59 x 4 x 36, or 8,496 training vectors per year. This may be a disadvantage, but on the other hand, each mini-region may have its own particular patterns.

In addition to the variables of the Eta model, used as information attributes, the density of discharges was used as the decision attribute. We assumed that such density can be correlated to the level of convective activity, and therefore a class of convective activity was assigned to each training pattern according to its corresponding density. The choice of the density of discharge was also due to the availability in Brazil of a network of discharge detectors that provides discharge occurrence data over most of the country with millisecond resolution.

There are other well-known indicators of convective activity, such as images from weather radar or meteorological satellites. Satellite images provide cloud coverage and cloud-top brightness temperatures, that are good indicators [24]. In addition, cloud coverage allows estimating rainfall rates [25, 26]. However, satellite orbit coverage and periodicity are limited over Brazil and there are only few weather radars. Therefore, we adopted the density of discharges as indicator.

Table 1 : The 26 selected variables of the Eta model.

One-level variables
pslm (MSL pressure, Mesinger method [hPa])
pslc (surface pressure [hPa])
dp2m (metre dewpoint temperature [K])
cape (convective available potential energy [m^2/s^2])
bli (best lifted index to 500 hPa [K])
agpl (instantaneous precipitable water [kg/m^2])
Multi-level variables at different pressure levels

vvel (meridional wind [m/s])	**at 925, 500 and 300 hPa**
uvel (zonal wind [m/s])	**at 925, 500 and 300 hPa**
zgeo (geopotential height [gpm])	**at 1000, 500 and 300 hPa**
temp (absolute temperature [K])	**at 925, 700, 500 and 300 hPa**
Omega [Pa/s]	**at 925, 500 and 300 hPa**
umes (specific humidity [kg/kg])	**at 925, 700, 500 and 300 hPa**

A simple statistical analysis of the densities of discharges for known cases of severe convective activity allowed to define a suitable threshold of density (normalized value of 0.01) in order to distinguish between classes **NSCA** and **SCA**. Consequently, training patterns (26-variable vectors) generated by the Eta model with densities less than this threshold were assigned to class **NSCA** and the remaining, to class **SCA**. Table 2 shows the distribution of data in the two classes for the three mini-regions considering this threshold.

This neural network was designed for recognizing patterns of atmospheric convective activity. The meteorological database was divided into training and test sets, that are mutually exclusive. A third one, the validation set, is obtained by detaching randomly a number of vectors from the training set, but preserving class balance. It is used during the training phase in order to identify the epoch/iteration that presented the minimum validation error. The corresponding weights and biases define the neural network to be employed in the test phase. Typically, If network error for the validation set starts to increase after an appreciable number of iterations, this is assumed as an indication of overfitting and training is halted. However, for this problem and data, such behavior was not observed and the training proceeds until reaching the limit number of epochs. Concerning model validation schemes, a chronological split, that corresponds to a non-random holdout, was adopted in order to simulate the prediction of severe convective activity.

Table 2 : Number of vectors/patterns *per* class for each mini-region.

Ano	**mini_region A**		**mini_region B**		**mini_region C**	
	NSCA	**SCA**	**NSCA**	**SCA**	**NSCA**	**SCA**
2007	**8362**	**134**	**8168**	**328**	**8319**	**177**
2008	**8489**	**7**	**8357**	**139**	**7805**	**691**
2009	**8456**	**40**	**8430**	**66**	**8189**	**307**
2010	**8323**	**173**	**8434**	**62**	**7869**	**627**
2011	**8450**	**10**	**8026**	**434**	**7871**	**589**

3 A Previously Developed Classifier

The current research to detect severe convective events was initiated in 2008 with the choice of the Eta model and the density of discharge to compose the meteorological database. This database was also employed in this work. The use of the precipitation (rainfall) estimation given by the Eta model itself was discarded, since it is very innacurate. The set of 26 Eta variables was then selected, composing a vector. It was assumed that if two vectors were similar, then the associated weather conditions would be also similar. Therefore, similarity metrics commonly used, such as the Manhattan or Euclidean distance, could be employed to compare two vectors.

However, preliminary tests performed with such database showed that the classification was unfeasible. Similar vectors, according to the considered metric, were actually associated to different classes of convective activity. Therefore, an instance of the **NSCA** class was eventually indiscernible of another of the **SCA** class considering only the corresponding vectors. This might indicate a wrong choice of the 26 meteorological variables or that such variables were not equally correlated to the

decision variable (density of discharges). Since the 26 variables were chosen using the knowledge of specialists in Meteorology and Climatology, we opted to maintain the set of variables and to seek more subtle information in subsets of these variables.

This resulted in our previous approach to tackle the current problem, named frequency of occurrence [2]. Since it would be virtually impossible to test all subsets in the 26-variable set in order to distinguish vectors belonging to different classes, we attempted to consider only pairs of variables. This is computationally feasible since for each vector with 26 variables there are 325 possible pairs of variables. In the "training phase", clusters of vectors are build based solely on the decision variable (density of discharges) using thresholds of density given by a simple statistics. Subsequently, in the "test phase", a given vector is classified according to its similarity to the vectors of each cluster/class. The similarity measure is given by the frequence of occurrence of similar discrete values of the 26 meteorological variables and also of similar discrete values of all possible pairs of variables. The related discretization scheme splits the original ranges of each variable into 16 bands. In order to classify a new vector, a similarity measure must be obtained. Matrices of frequence of occurrence are build comparing the vector to all vectors of each cluster. Each entry of these matrices represents a count of similar values (diagonal positions) or similar pairs of values (non-diagonal positions). These counts are the frequences of occurrence of similar discrete values. Three similarity measures are then obtained by summing up the elements of each matrix. The vector is then classified as pertaining to the cluster with the highest summation of frequencies of occurrence in the corresponding matrix. This approach is innovative and inspired the architecture of the neural network proposed in this work, that also considers pairs of input variables in the hidden layer, as described in the next section.

This frequency of occurrence approach yielded good results in the classification of patterns associated to severe convective events, but only when employing a 10-fold cross-validation scheme and three classes (weak/absent, moderate and severe convective activity. It did not perform as expected in the case of a chronological split, corresponding to a non-random holdout scheme that would correspond to a reduction. This would be the case of employing data of some weeks in the "training phase" and data of a subsequent week in the "test phase". This issue made us to consider further approaches to obtain a classifier able to predict such severe convective events, as is the case of neural networks, supposed to solve complex data classification problems. The next section presents the neural network that we propose for the prediction of severe convective events.

4 Methodology

The proposed neural network architecture and the training algorithm are described in the following subsections, while the results obtained in the classification of meteorological patterns are shown in Section 5.

4.1 Neural Network Architecture

As shown in Fig. 1, the new neural network was designed with 26 input nodes, corresponding to the meteorological variables of each training pattern. The proposed network architecture connects each of the 325 possible pairs of variables (taken among the 26 inputs) to a particular neuron in the hidden layer. The net input for each hidden neuron is then given by the values of its two input values multiplied by the respective weights plus its bias. This architecture that partially connects input nodes and hidden neurons was chosen in order to seek the information contained in pairs of variables, as explained in the previous section. Therefore, the network employs subsets given by the pairs of variables to identify what information is relevant for class separation.

It is common in classification tasks make a nonlinear mapping from the input space into a feature space of higher dimensionality under the assumption that, in this way, a nonlinear problem can be turned into a linear or "less nonlinear" one [28]. This characteristic is exploited here: the nonlinear mapping is provided by the bipolar sigmoid activation function for the hidden layer, while the 325 hidden activations (implying in a higher dimensionality than the set of 26 input nodes) are multiplied by the respective weights and summed up to produce the network output. Therefore, the identity function is used in the output layer. Thus, we may say that our network is trained according to a scheme of decomposition and synthesis of the training patterns.

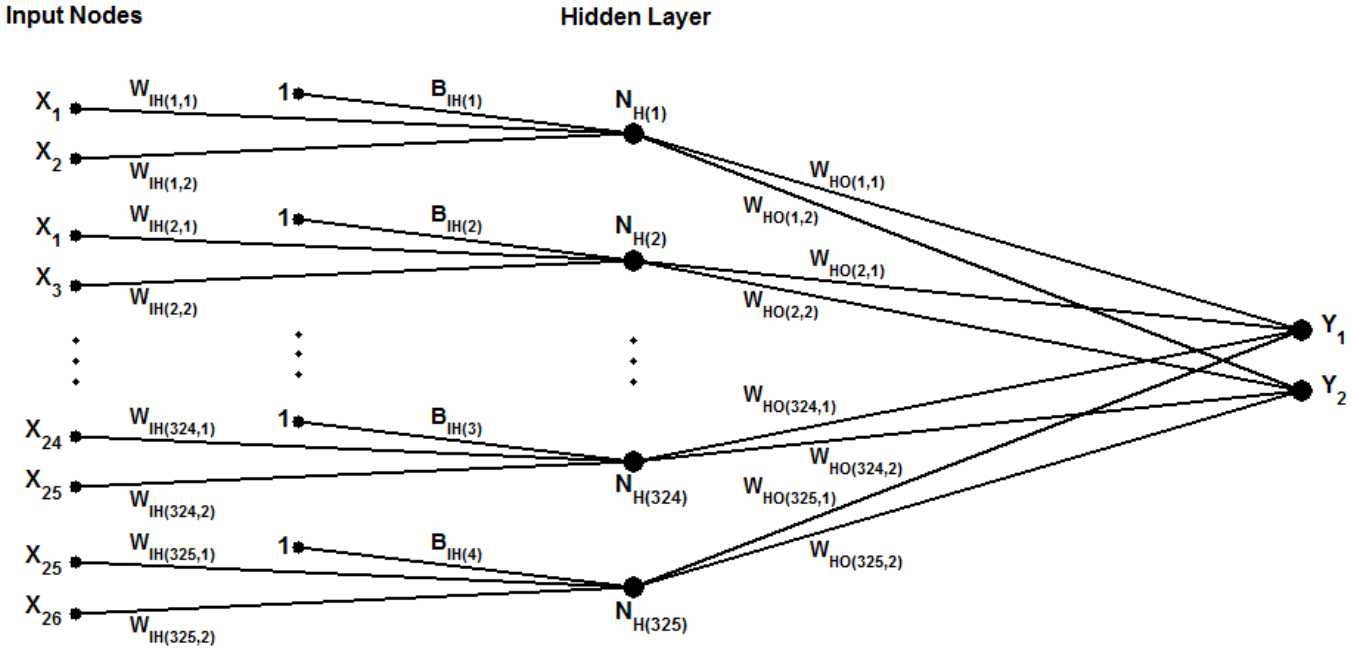

Figure 1: Proposed neural network architecture

4.2 Neural Network Training Algorithm

A normalization of the meteorological variables to the range [-1, 1] was required to use them as inputs of a neural network using the bipolar sigmoid activation function shown in Eq. 1. In the architecture described in the previous section, the hidden activations ***HA*** that result of each input pattern ***X*** presented to the network are given by:

$$\mathbf{HA} = \frac{\mathbf{1} - \exp(-(\Sigma\ (\mathbf{X} \circ \mathbf{W_{IH}}) + \mathbf{B_{IH}}))}{\mathbf{1} + \exp(-(\Sigma\ (\mathbf{X} \circ \mathbf{W_{IH}}) + \mathbf{B_{IH}}))} \quad (1)$$

where (each vector or matrix is followed by its dimension):

X (325x2) is the input pattern obtained by rearranging the corresponding input vector (26x1) into 325 pairs of variables;

HA (325x1) is the vector of hidden activations produced by input pattern ***X***;

$\boldsymbol{W_{IH}}$ (325x2) is the weight matrix connecting the input to the hidden layer;

$\boldsymbol{B_{IH}}$ (325x1) is the vector of biases applied to the hidden neurons;

X o $\boldsymbol{W_{HI}}$ (325x2) denotes the element-wise multiplication of ***X*** by $\boldsymbol{W_{IH}}$;

$\sum$ (***X*** o $\boldsymbol{W_{IH}}$) (325x1) is the summation of the pairs of weighted inputs in each row of ***X*** o $\boldsymbol{W_{HI}}$.

The network outputs Y_1 e Y_2 employ a linear activation function given by :

$$\mathbf{Y_1} = \sum_{\mathbf{j=1}}^{\mathbf{325}} (\mathbf{HA_j} \circ \mathbf{W_{HO\,(j,1)}})$$
$$\mathbf{Y_2} = \sum_{\mathbf{j=1}}^{\mathbf{325}} (\mathbf{HA_j} \circ \mathbf{W_{HO\,(j,2)}}) \quad (2)$$

where, $W_{HO(j,1)}$ and $W_{HO(j,2)}$ (325x1) are the weight vectors connecting the hidden layer to the outputs Y_1 and Y_2, respectively;

HA o W_{HO} (325x1) denotes the element-wise multiplication of HA by W_{HO};

$\sum$ (HA o W_{HO}) is the inner product of HA and W_{HO}.

Considering a threshold T, it is expected that the neural network produce an output Y_1 above T and an output Y_2 below $-T$ for input patterns of the class **SCA** and outputs Y_1 e Y_2 below $-T$ and above T, respectively, for input patterns of the class **NSCA**. Otherwise, there are errors E_1 and E_2 expressed by either $E_1 = (T - Y_1)$ and $E_2 = (-T - Y_2)$ or $E_1 = (-T - Y_1)$ and $E_2 = (T - Y_2)$. Such errors are then employed to adjust the network weights W_{IH} , W_{HO} and the biases B_{IH} . The adjustment is based on the gradient descent method. The thresholds of {-0.1, 0.1} and {0.1, -0.1} were adopted, since the typical ones ({1,0} and {0 ,1}) were not adequated for this problem.

In order to avoid excessive perturbation of the network during the training phase, we opted to use batch learning, adjusting the weights at the end of every epoch and taking into account the corrections due to all the input patterns. For the same reason, after computing the correction due to the i-th input pattern, we also chose not to adjust all the weights, but only the ones corresponding to the products in ($HA^i \circ W_{HO}{}^i$) that individually contributed to a misclassification. Therefore, if the i-th input pattern is a **NSCA** pattern, we consider (for correction purposes) only the weights corresponding to the products in ($HA_j^i \circ W_{HO(j,1)}{}^i$) which are larger than –T and the weights corresponding to the products in ($HA_j^i \circ W_{HO(j,2)}{}^i$) which are smaller than T; if the i-th input pattern is a **SCA** pattern, only the weights corresponding to the products in ($HA_j^i \circ W_{HO(j,1)}{}^i$) which are smaller than T and the weights corresponding to the products in ($HA_j^i \circ W_{HO(j,2)}{}^i$) which are larger than -T are corrected. This is achieved by means of a mask vector M^i (for the same i-th input pattern) containing ones for such elements and zeros for the remaining ones. At the end of the epoch the corrections from the input patterns are averaged separately, for **NSCA** input patterns and for **SCA** input patterns. This is done in order to reduce the effects of class imbalance. Finally, the network weights are adjusted from these two averages by employing a specific heuristic explained afterwards.

In the case of the weights $W_{HO(j,k)}$, for the i-th input pattern, the correction vector $(\Delta W_{HO(j,k)})^i$ considers only the elements filtered by the mask M^i:

$$\Delta \mathbf{W_{HO\,(j,1)}}^{\mathbf{i}} = \mathbf{E_1}^{\mathbf{i}} (\mathbf{HA_j}^{\mathbf{i}} \circ \mathbf{M}^{\mathbf{i}})$$
$$\Delta \mathbf{W_{HO\,(j,2)}}^{\mathbf{i}} = \mathbf{E_2}^{\mathbf{i}} (\mathbf{HA_j}^{\mathbf{i}} \circ \mathbf{M}^{\mathbf{i}}) \quad (3)$$

where, $E_1{}^i$ and $E_2{}^i$ are, respectively, the network errors for the outputs values Y_1 *e* Y_2 due to the i-th input pattern.

Since we are using the gradient descent method, corrections for the hidden activations HA^i are calculated for the i-th input pattern, using the same mask vector M^i, by:

$$\Delta \mathbf{HA}^{\mathbf{i}} = [\mathbf{E_1}^{\mathbf{i}} (\mathbf{W_{HO\,(j,1)}}^{\mathbf{i}} \circ \mathbf{M}^{\mathbf{i}}) + \mathbf{E_2}^{\mathbf{i}} (\mathbf{W_{HO\,(j,2)}}^{\mathbf{i}} \circ \mathbf{M}^{\mathbf{i}})]/2 \quad (4)$$

Then, the vector $(\Delta W_{IH})^i$ that corrects the weights W_{IH} is calculated as follows:

$$\Delta \mathbf{W_{IH}}^{\mathbf{i}} = (\Delta \mathbf{HA}^{\mathbf{i}}) \circ \frac{\mathbf{1}}{\mathbf{2}} (\mathbf{1} - (\mathbf{HA}^{\mathbf{i}})^{\mathbf{2}}) \circ \mathbf{X}^{\mathbf{i}} \quad (5)$$

where, X^i is the i-th input patterns and the vector $(1/2)(1-(HA^i)^2)$ contains the 325 derivatives of bipolar sigmoid activation function with respect to the corresponding HA^i hidden activations.

Finally, the vector $(\Delta B_{IH})^i$ that corrects the biases B_{IH} is given by:

$$\Delta B_{IH}{}^i = (\Delta HA^i) \circ \frac{1}{2}\,(1-(HA^i)^2) \tag{6}$$

These corrections due to each input pattern, as given by the Eqs. 3, 5 and 6, are averaged at the end of each epoch. Assuming a total of $N = N^{(SCA)} + N^{(NSCA)}$ of training patterns, the average correction $\overline{\Delta W}_{HO}^{(NSCA)}$, due to all $N^{(NSCA)}$ input patterns of the class **NSCA**, for the weights W^{HO} is expressed by:

$$\overline{\Delta W}_{HO(j,1)}^{(NSCA)} = \frac{1}{N^{(NSCA)}} \sum_{i=1}^{N} ((\Delta W_{HO(j,1)})^i \circ \delta(i))$$

$$\overline{\Delta W}_{HO(j,2)}^{(NSCA)} = \frac{1}{N^{(NSCA)}} \sum_{i=1}^{N} ((\Delta W_{HO(j,2)})^i \circ \delta(i)) \tag{7}$$

$$\delta(i)=1 \text{ if class of } X^i = NSCA \text{ and } \delta(i)=0 \text{ if class of } X^i = SCA$$

A similar expression allows calculating the average corrections $\overline{\Delta W}_{HO(j,1)}^{(SCA)}$ and $\overline{\Delta W}_{HO(j,2)}^{(SCA)}$ concerning the input patterns of

class **SCA.** There are also pairs of such expressions to calculate the remaining average corrections $\overline{\Delta W}_{HI}^{(NSCA)}$, $\overline{\Delta W}_{HI}^{(SCA)}$, $\overline{\Delta B}_{HI}^{(NSCA)}$ and $\overline{\Delta B}_{HI}^{(SCA)}$.

At the end of the considered epoch, the proposed heuristic that balances the influence of both classes is applied for weights and biases adjustments. It employs the network learning rate α and a damping parameter λ. Denoting by W a generic network weight/bias (that may be an element of W_{HO} or W_{IH} or B_{IH}), the heuristic proceeds as:

- If the signs of the two average adjustments for W (derived from **NSCA** patterns and **SCA** patterns) are different of the W sign, such adjustments are multiplied by α (as seen in Eq. 8a).
- If the signs of the two average adjustments are the same as W, such adjustments are multiplied by α and λ (as seen in Eq. 8b).
- If the signals of the two adjustments calculated for W are different, the adjustment with the same sign as W is multiplied by α and λ, while the other is multiplied by only α (as seen in Eqs. 8c or 8d).

Therefore, the adjustment for the weight W will be performed, at the end of current epoch, by one of the following options:

$$W_{new} = W_{old} + \alpha\,(\overline{\Delta W}^{(NSCA)}) + \alpha\,(\overline{\Delta W}^{(SCA)}) \tag{8a}$$

$$W_{new} = W_{old} + (\alpha * \lambda) * (\overline{\Delta W}^{(NSCA)}) + (\alpha * \lambda) * (\overline{\Delta W}^{(SCA)}) \tag{8b}$$

$$W_{new} = W_{old} + (\alpha * \lambda)\,(\overline{\Delta W}^{(NSCA)}) + \alpha\,(\overline{\Delta W}^{(SCA)}) \tag{8c}$$

$$W_{new} = W_{old} + \alpha\,(\overline{\Delta W}^{(NSCA)}) + (\alpha * \lambda)\,(\overline{\Delta W}^{(SCA)}) \tag{8d}$$

where, the terms $\overline{\Delta W}^{(NSCA)}$ and $\overline{\Delta W}^{(SCA)}$ correspond to the two average adjustments calculated for the generic weight W due to **NSCA** and **SCA** patterns, respectively.

The idea underlying this heuristic is to prevent that any of the weights W_{HO} or W_{IH} or the biases B_{IH} result overcorrected due to the influence of one class, while not being influenced by the other class. Besides trying to avoid that the network training be biased by one of the classes, to a certain extent this heuristic also apply a kind of regularization similar to weight decay [29] since it keeps the values of the weights within a small range that avoids overfitting and poor generalization. It is worth to note that without applying this heuristic, the training convergence was slower and the generalization ability of neural network was poorer.

In the proposed training algorithm, the definition of initial values for the learning rate α and the damping parameter λ, as well as an update scheme were required. We empirically chose the initial values of α and λ as 0.10 and 0.75, respectively. These values are updated for epoch *j,* i.e. from epoch *(j-1)* to epoch *j*, using the following formulas, where *M* denotes the maximum number of epochs and the values α_min and λ_max were set to 0.001 and 1.000, respectively.

$$\alpha_{(new)} = \frac{\alpha\_min}{\alpha_{(old)}}^{\frac{1}{M-j}} \tag{9a}$$

$$\lambda_{(new)} = \frac{\lambda\_max}{\lambda_{(old)}}^{\frac{1}{M-j}} \tag{9b}$$

In addition, the maximum number of epochs M was set to 60000 defining the stopping criterion of the training phase. Adopting a threshold for the final mean square error (MSE) was not feasible since it would be very difficult to estimate a reasonable value for it. The network weights were initialized with small random values.

5 Results and Discussion

This section presents the results obtained with the proposed neural network architecture and training scheme applied to the classification of meteorological patterns assuming two classes (**NSCA** and **SCA**). Each classification test by an average of the confusion matrices corresponding to 20 executions. Minimum and maximum values, as well as the corresponding standard deviation are shown in an elementwise basis. In each matrix, diagonal elements denotes patterns of class **NSCA** or **SCA** that were correctly classified, while elements out-of-the diagonal, elements of class **NSCA** mistakenly classified as being **SCA** or vice-versa. The classification performance indexes, the Kappa index [30] and the classification accuracy, are calculated from the average confusion matrix. All these tests employed three datasets: training, validation and test. The training phase was always halted with 60,000 epochs. The corresponding learning curves (MSE x epoch) are also shown for the training and validation sets.

As described in Section 2, the meteorological database includes data of the Eta model for the months of January and February of 2007/2008/2009/2010/2011 for three mini-regions A, B, and C of Brazil. The training datasets are composed of data of January and the 18 first days of February of each year, while the test datasets are composed of the remaining 10 days of February of each year. In. short, 83% of the data were taken for training and the remaining 17% for test, in a holdout scheme, but preserving the chronological order of the data in order to simulate a prediction.

As shown in Table 2, the number of vector/patterns of class **NSCA** (moderate/weak/absent convective activity) in the training dataset is much higher than the vector/patterns of class **SCA** (severe convective activity). Therefore, in order to make feasible the training, a sampling was required for the vectors of the class **NSCA**. Table 3 shows, for the year 2007, the total number of

training patterns after the sampling that was actually used in the network training for each mini-region. Sampling of these patterns was performed randomly in a way to get number of patterns of class **NSCA** that is twice the number of patterns of class **SCA**. We considered this reduction acceptable and not affecting the representativeness of class **NSCA** in the training based on the assumption that information redundancy is higher among patterns associated to low density of discharge. All tests adopted this sampling for the training-validation dataset.

Table 3 : Distribution of training/validation patterns *per* class for the 3 mini-regions (2007).

mini-region	Number of patterns before sampling		Number of patterns after sampling			
	NSCA	SCA	NSCA		SCA	
			Training	Validation	Training	Validation
A	6945	111	200	22	100	11
B	6751	305	550	60	275	30
C	6977	79	142	16	71	8

In addition, Table 4 shows that a similar sampling was performed also in the test dataset. This sampling would be unnecessary, but was adopted in the test phase due to the Kappa index, since it is very sensitive to unbalance between the number of vectors/patterns of each class. However this sampling of the test set was only employed in few test cases, in Section 5.1 and 5.4.

Table 4 : Distribution of test patterns *per* class for the 3 mini-regions (2007).

mini-region	Number of patterns before sampling		Number of patterns after sampling	
	NSCA	SCA	NSCA	SCA
A	1417	23	46	23
B	1417	23	46	23
C	1342	98	52	26

A brief description of the results follows. Section 5.1 show results for each of the three mini-regions for 2007 using sampling in the test set. Section 5.2 show results for mini-region B without sampling in the test set for years 2007-2011 in a year-by-year scheme, i.e. training and testing with data of the same year. Section 5.3 show results for mini-region B without sampling for years 2007-2011, but training with data of all these years, but testing separately for each year. Finally, in order to serve as a benchmark, Section 5.4 presents results obtained with a standard backpropagation ANN with two output neurons for the three mini-regions for 2007 using sampling in the test set. Therefore, these results can be compared to those of Section 5.1.

5.1 Proposed ANN for mini-regions A, B and C (year 2007)

These results were obtained with the proposed ANN for the three mini-regions, but using sampling also in the test dataset. Results are good, showing that only a few (up to 3) SCA patterns were not detected, corresponding to false negatives. False positives are also a few (up to 5). In general, Kappa values are above 80% and accuracies, above 90%. Classification performance had the benefit of a sampled test set, that provided a 2:1 balance between NSCA and SCA patterns.

Table 5 : Confusion matrices, statistics and performance evaluation indices for mini-region A (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8134**
Actual	NSCA	**178/200**	**22/200**	**172/178/190 (5)**	**10/22/28 (5)**	**Accuracy:0.9133**
	SCA	**4/100**	**96/100**	**1/4/6 (1)**	**94/96/99 (1)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8125**
Actual	NSCA	**41/46**	**5/46**	**37/41/45 (2)**	**1/5/9 (2)**	**Accuracy:0.9130**
	SCA	**1/23**	**22/23**	**0/1/3 (1)**	**20/22/23 (1)**	

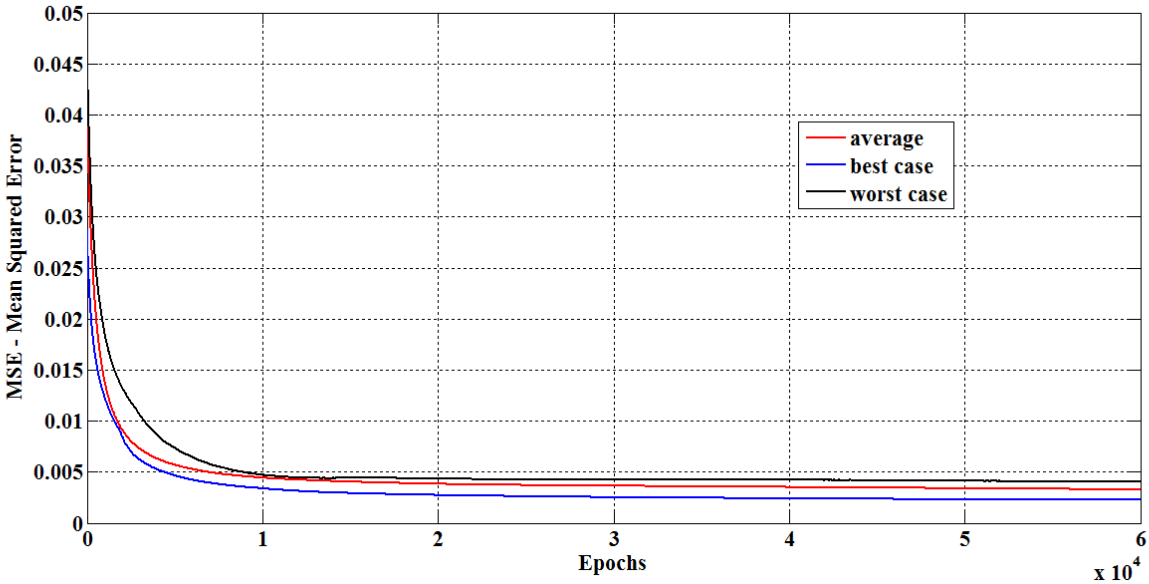

Figure 2 : **Learning curves for the training set of mini-region A (2007).**

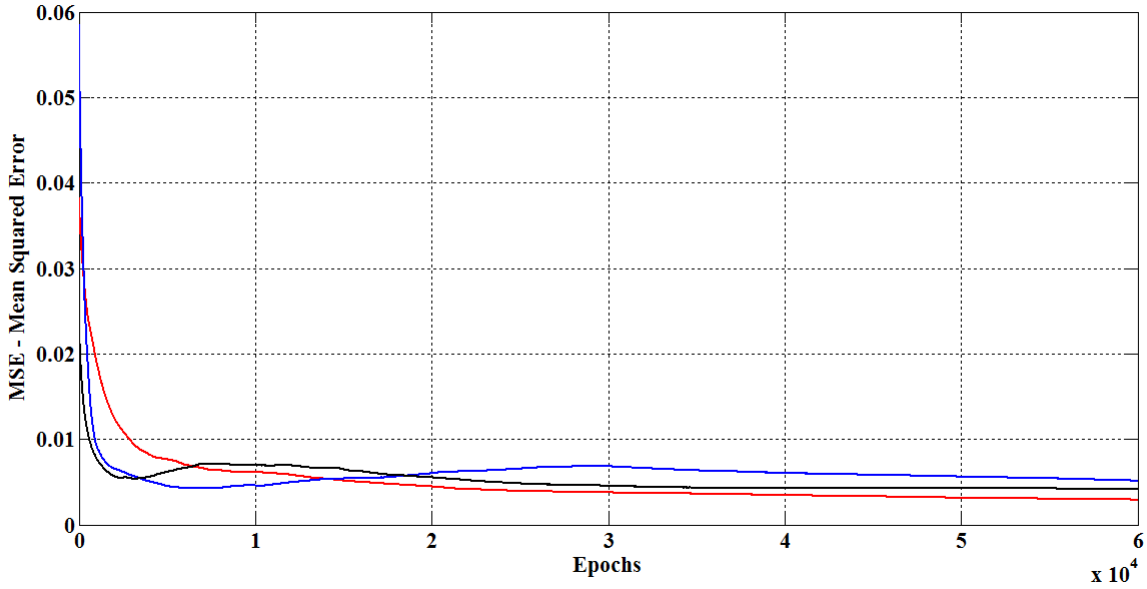

Figure 3 : **Some examples of error curves for the validation set of mini-region A (2007).**

Table 6 : Confusion matrices, statistics and performance evaluation indices for mini-region B (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8150**
Actual	NSCA	**506/550**	**44/550**	**498/506/520 (6)**	**30/44/52 (6)**	**Accuracy:0.9164**
	SCA	**25/275**	**250/275**	**20/25/39 (5)**	**236/250/255 (5)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8696**
Actual	NSCA	**44/46**	**2/46**	**42/44/45 (1)**	**1/2/4 (1)**	**Accuracy:0.9420**
	SCA	**2/23**	**21/23**	**0/2/4 (1)**	**19/21/23 (1)**	

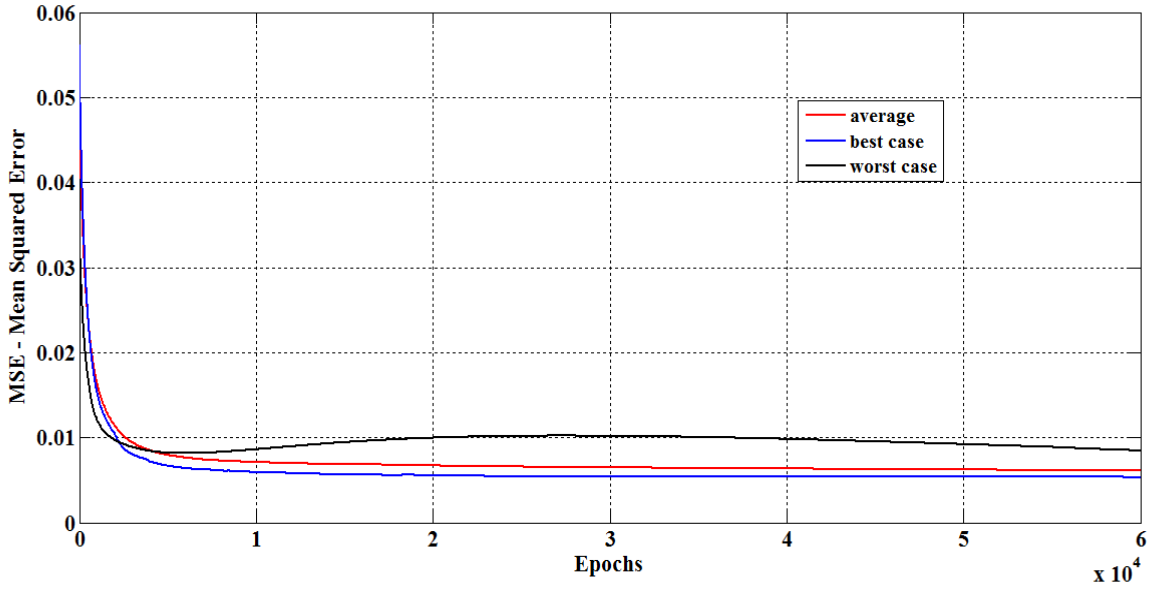

Figure 4 : Learning curves for the training set of mini-region B (2007).

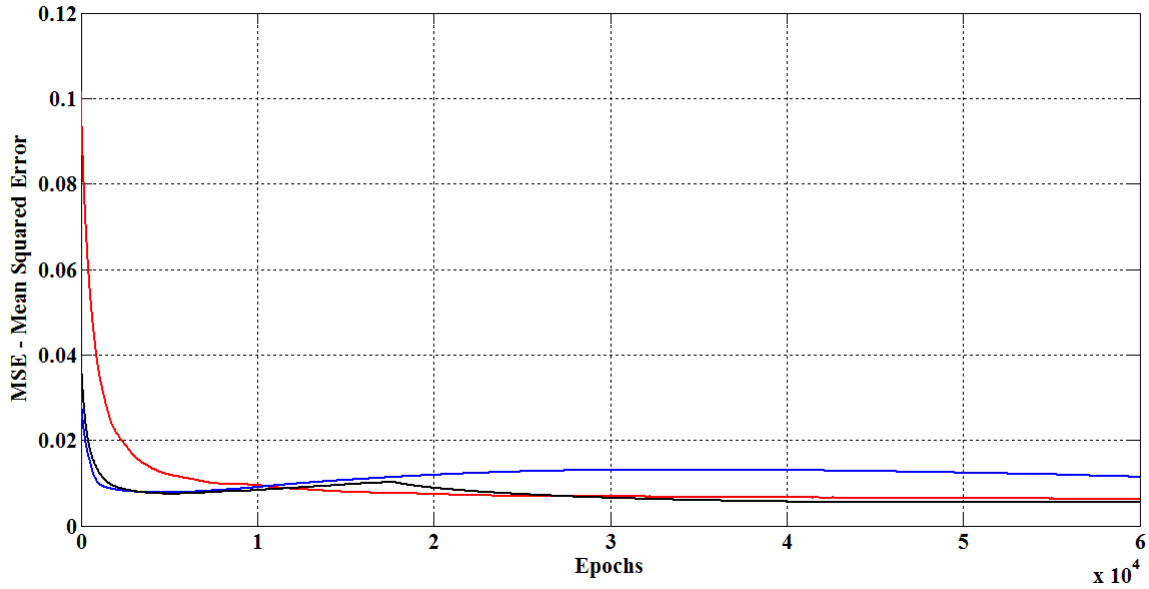

Figure 5 : Some examples of error curves for the validation set of mini-region B (2007).

Table 7 : Confusion matrices, statistics and performance evaluation indices for mini-region C (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8041**
Actual	NSCA	**129/142**	**13/142**	**119/129/136 (5)**	**6/13/23 (5)**	**Accuracy:0.9108**
	SCA	**6/71**	**65/71**	**2/6/11 (2)**	**60/65/69 (2)**	
Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7736**
Actual	NSCA	**47/52**	**5/52**	**41/47/50 (3)**	**2/5/11 (3)**	**Accuracy:0.8974**
	SCA	**3/26**	**23/26**	**0/3/8 (2)**	**18/23/26 (2)**	

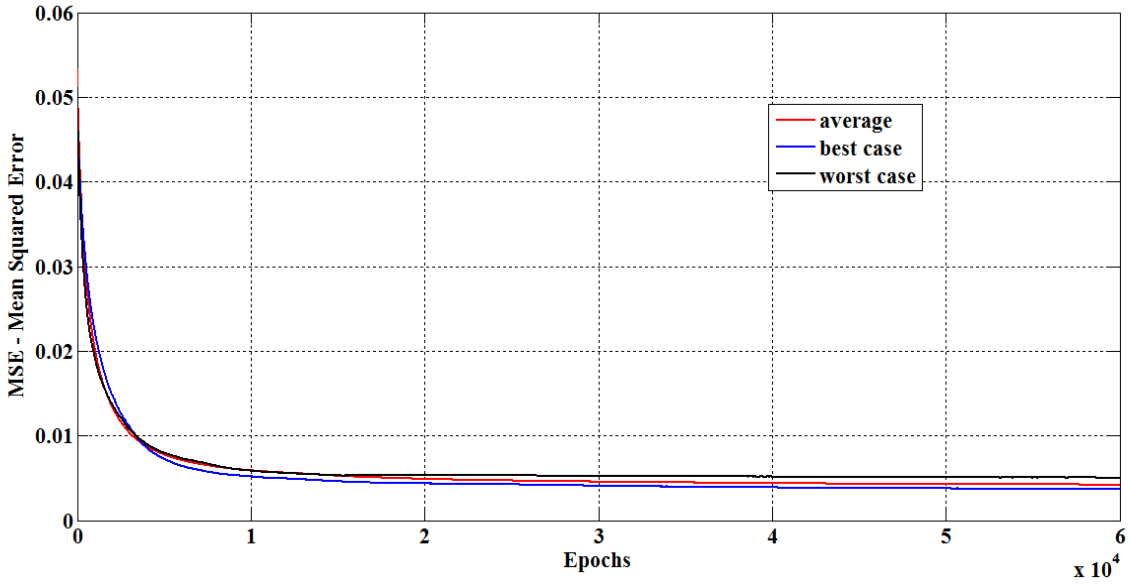

Figure 6 : **Learning curves for the training set of mini-region C (2007).**

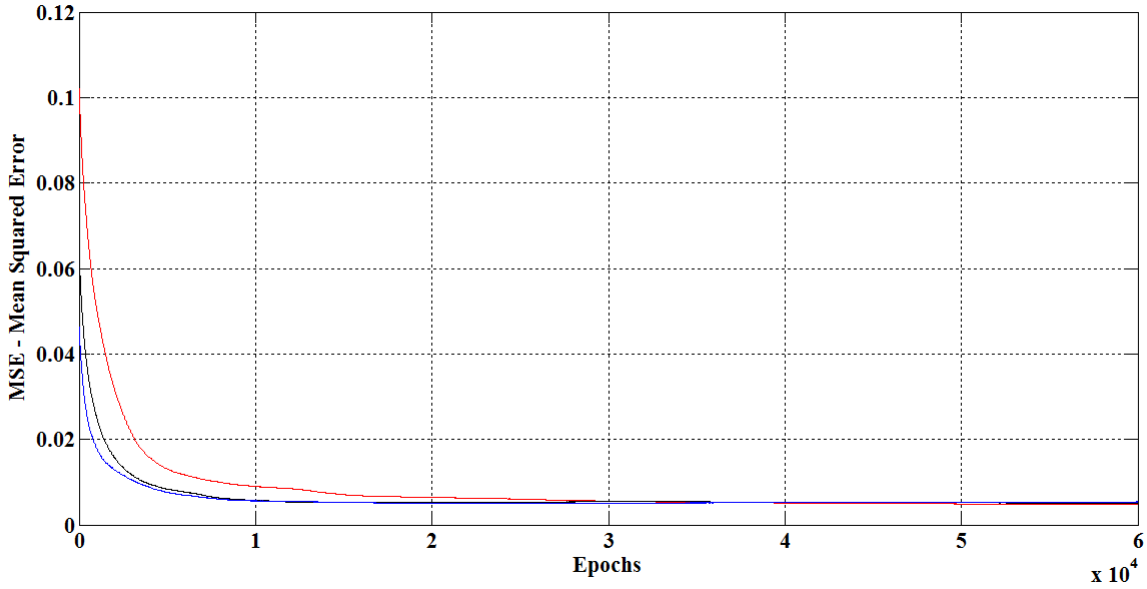

Figure 7 : Some examples of error curves for the validation set of mini-region C (2007).

5.2 Proposed ANN, mini-region B (2007-2011, year-by-year training)

These results were obtained with the proposed ANN only for mini-region B, but without using sampling in the test dataset. Training and test were performed for each year separately. Classification performances were not good as in the tests of the previous section, that employed sampled test sets. However, the proposed ANN was able to detect most of the SCA patterns and the number of false positives was acceptable. There were few SCA patterns in 2009 and 2010, but a higher number in 2011. The Kappa values were very low due to class imbalance.

Table 8 : Confusion matrices, statistics and performance evaluation indices for mini-region B (without sampling)(2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8150**
Actual	NSCA	**506/550**	**44/550**	**498/506/520 (6)**	**30/44/52 (6)**	**Accuracy:0.9164**
	SCA	**25/275**	**250/275**	**20/25/39 (5)**	**236/250/255 (5)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.1456**
Actual	NSCA	**1225/1417**	**192/1417**	**1142/1225/1277 (35)**	**140/192/275 (35)**	**Accuracy:0.8646**
	SCA	**3/23**	**20/23**	**1/3/5 (1)**	**18/20/22 (1)**	

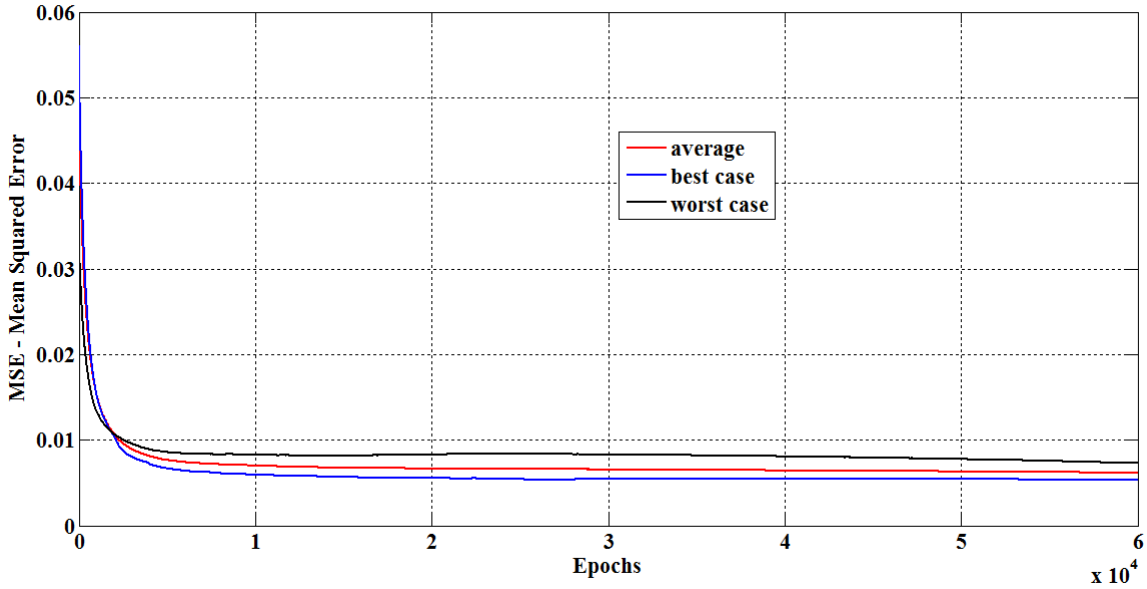

Figure 8 : Learning curves for training set of mini-region B (without sampling) (2007).

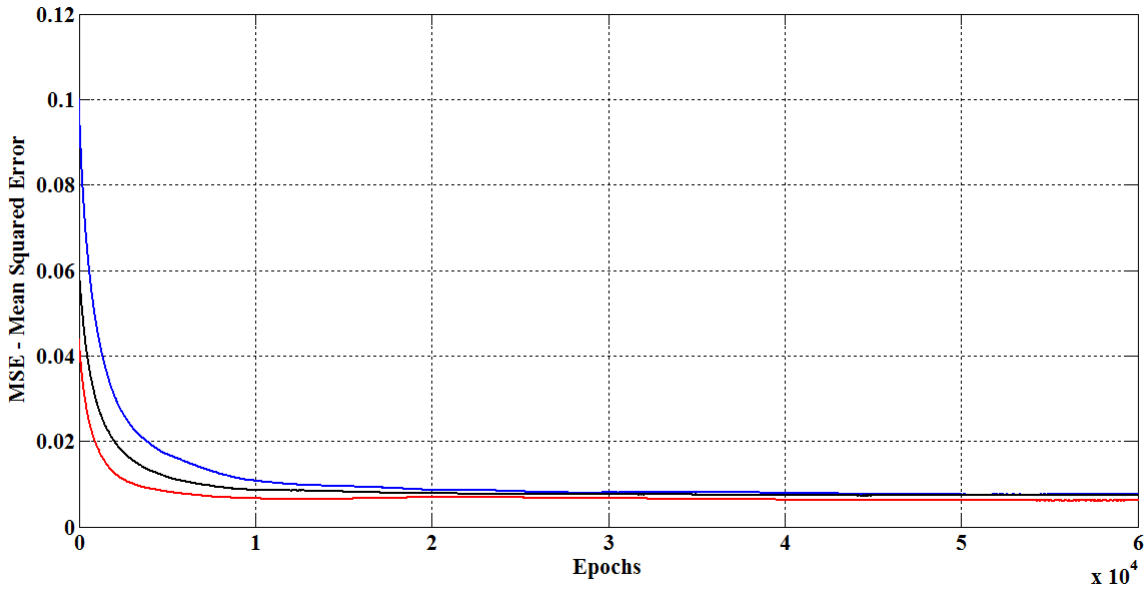

Figure 9 : Some examples of error curves for the validation set of mini-region B (without sampling) (2007).

Table 9 : Confusion matrices, statistics and performance evaluation indices for mini-region B (without sampling)(2008).

		Average of 20 executions		Min./Average/Max. (Standard deviation)		Performance indexes
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.8045**
Actual	NSCA	**195/216**	**21/216**	**191/195/200 (2)**	**16/21/25 (2)**	**Accuracy:0.9105**
	SCA	**8/108**	**100/108**	**5/8/10 (1)**	**98/100/103 (1)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.1078**
Actual	NSCA	**1209/1421**	**212/1421**	**1042/1209/1335 (68)**	**86/212/379 (68)**	**Accuracy:0.8507**
	SCA	**3/19**	**16/19**	**0/2/7 (2)**	**12/17/19 (2)**	

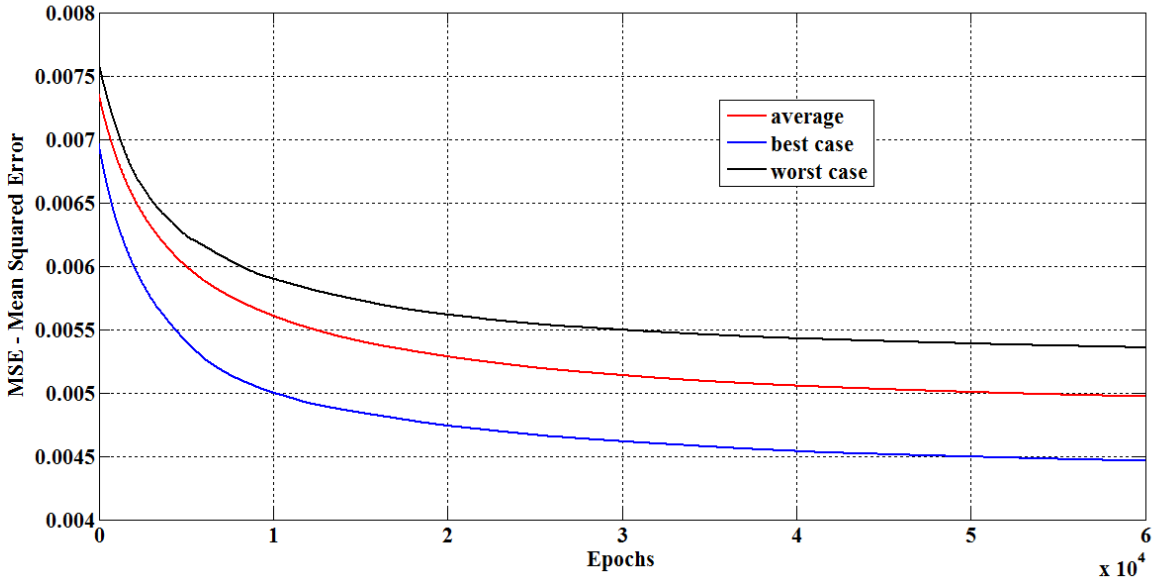

Figure 10 : **Learning curves for training set of mini-region B (without sampling) (2008).**

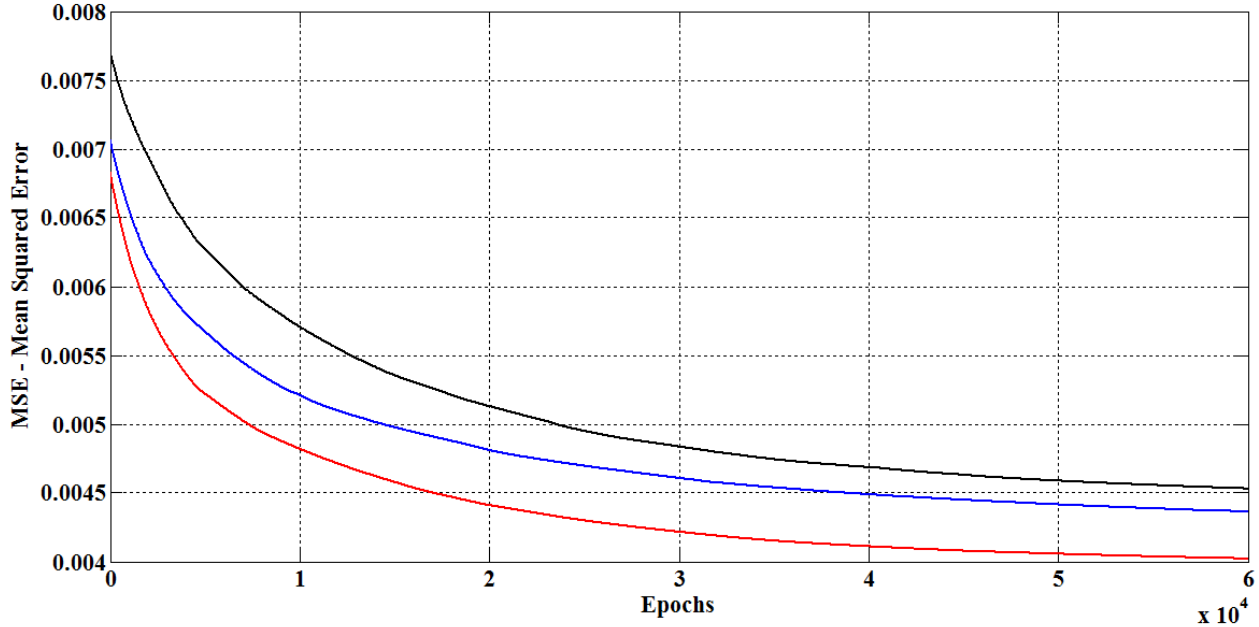

Figure 11 : **Some examples of error curves for the validation set of mini-region B (without sampling) (2008).**

Table 10 : Confusion matrices, statistics and performance evaluation indices for mini-region B (without sampling)(2009).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7364**
Actual	NSCA	**102/116**	**14/116**	**97/102/106 (3)**	**10/14/19 (3)**	**Accuracy:0.8793**
	SCA	**7/58**	**51/58**	**5/7/9 (1)**	**49/251/53 (1)**	

		Predicted		**Predicted**		
Test		NSCA	SCA	NSCA	SCA	**Kappa: 0.0140**
Actual	NSCA	**1203/1438**	**235/1438**	**1091/1203/1310 (57)**	**128/235/347 (57)**	**Accuracy:0.8368**
	SCA	**0/2**	**2/2**	**0/0/0 (0)**	**2/2/2 (0)**	

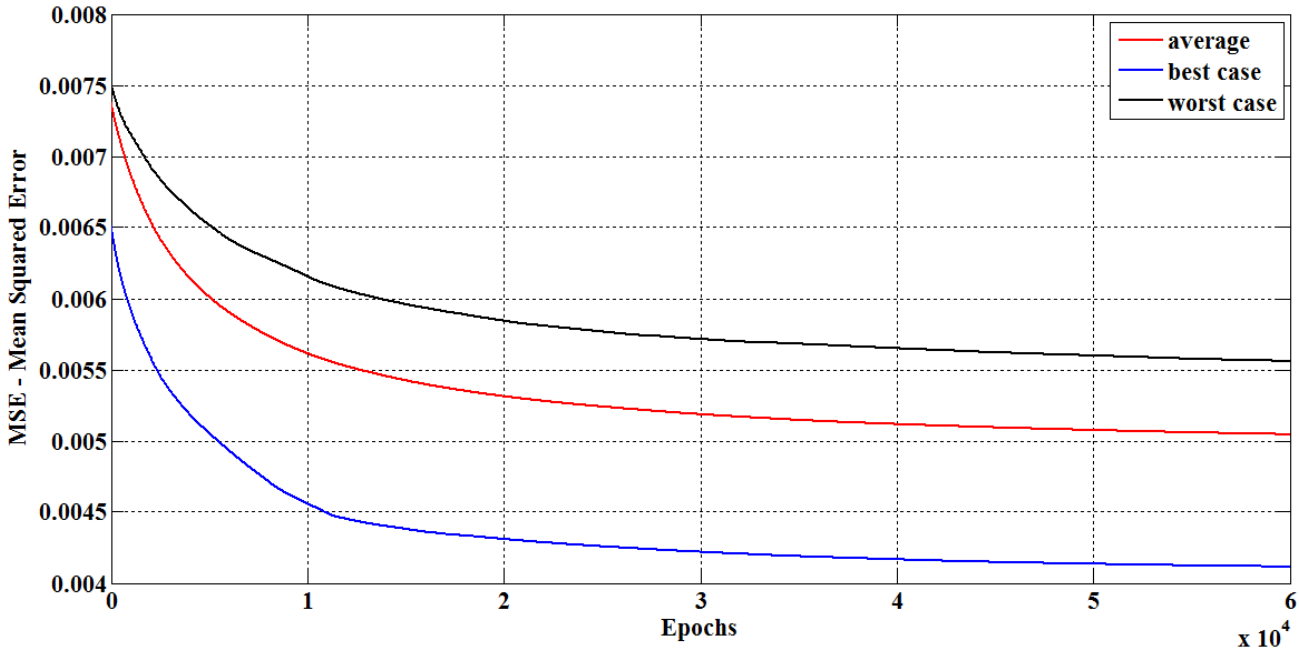

Figure 12 : Learning curves for training set of mini-region B (without sampling) (2009).

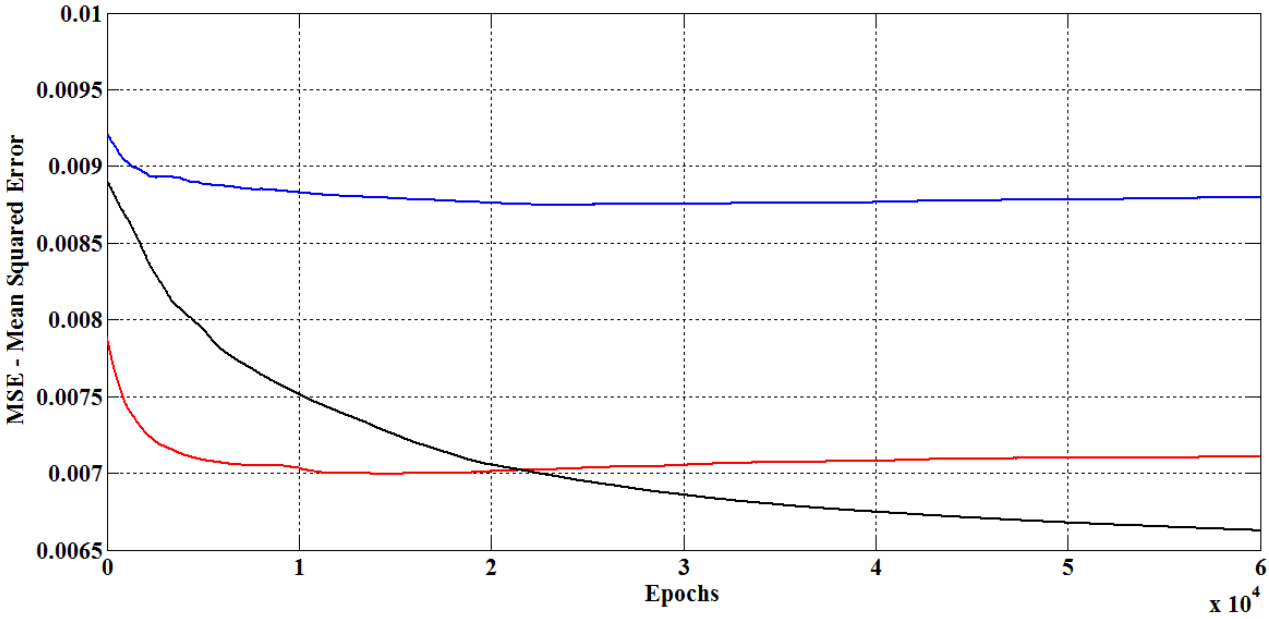

Figure 13 : Some examples of error curves for the validation set of mini-region B (without sampling) (2009).

Table 11 : Confusion matrices, statistics and performance evaluation indices for mini-region B (without sampling)(2010).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7652**
Actual	NSCA	**99/113**	**14/113**	**95/99/102 (2)**	**11/14/18 (2)**	**Accuracy:0.8896**
	SCA	**5/56**	**51/56**	**2/5/8 (2)**	**48/51/54 (2)**	

		Predicted		**Predicted**		
Test		NSCA	SCA	NSCA	SCA	**Kappa: 0.0000**
Actual	NSCA	**1166/1440**	**274/1440**	**1111/1166/1242 (39)**	**198/274/329 (39)**	**Accuracy:0.8097**
	SCA	**0/0**	**0/0**	**0/0/0 (0)**	**0/0/0 (0)**	

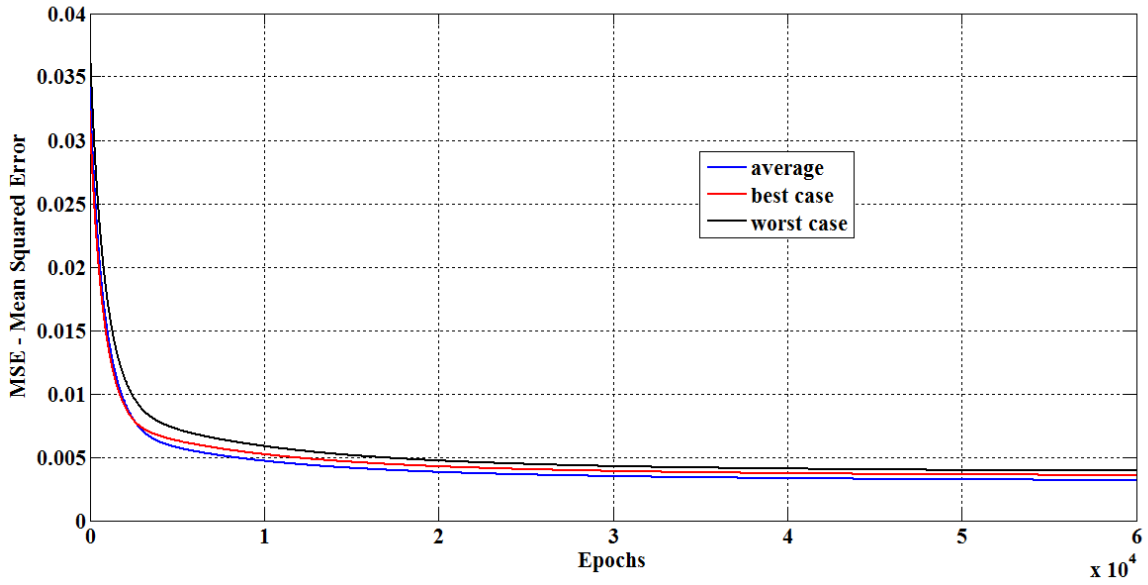

Figure 14 : **Learning curves for training set of mini-region B (without sampling) (2010).**

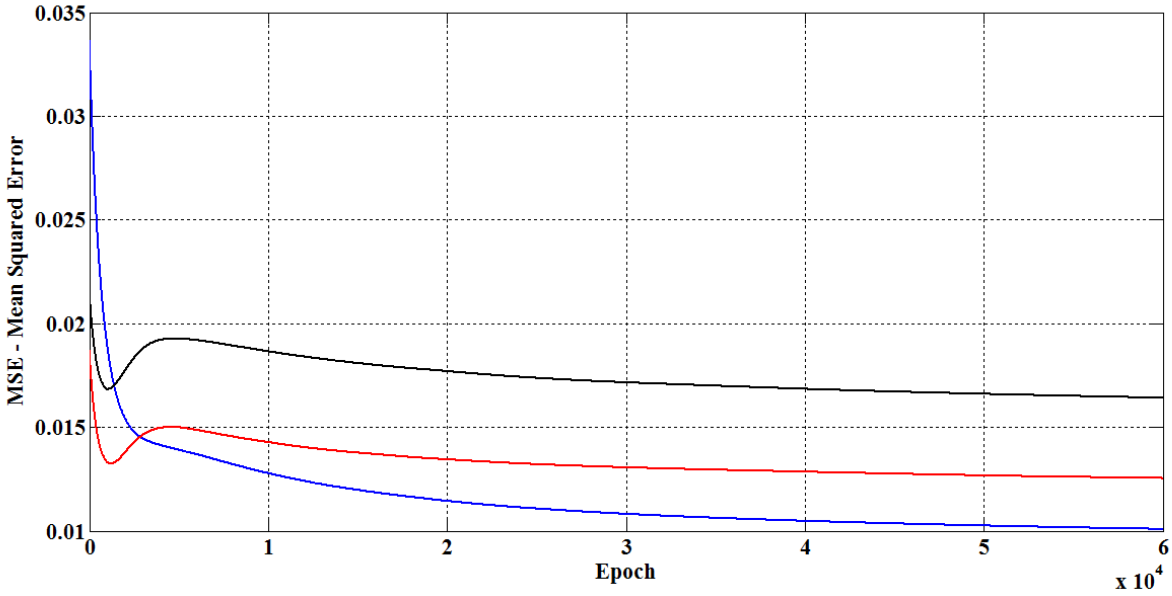

Figure 15 : **Some examples of error curves for the validation set of mini-region B (without sampling) (2010).**

Table 12 : Confusion matrices, statistics and performance evaluation indices for mini-region B (without sampling)(2011).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7220**
Actual	NSCA	**549/640**	**91/640**	**542/549/558 (5)**	**82/91/98 (5)**	**Accuracy:0.8708**
	SCA	**33/320**	**287/320**	**24/33/44 (5)**	**276/287/298 (5)**	

		Predicted		**Predicted**		
Test		NSCA	SCA	NSCA	SCA	**Kappa: 0.2385**
Actual	NSCA	**1082/1326**	**244/1326**	**1010/1082/1147 (36)**	**179/244/316 (36)**	**Accuracy:0.8185**
	SCA	**11/79**	**68/79**	**6/11/16 (2)**	**63/68/73 (2)**	

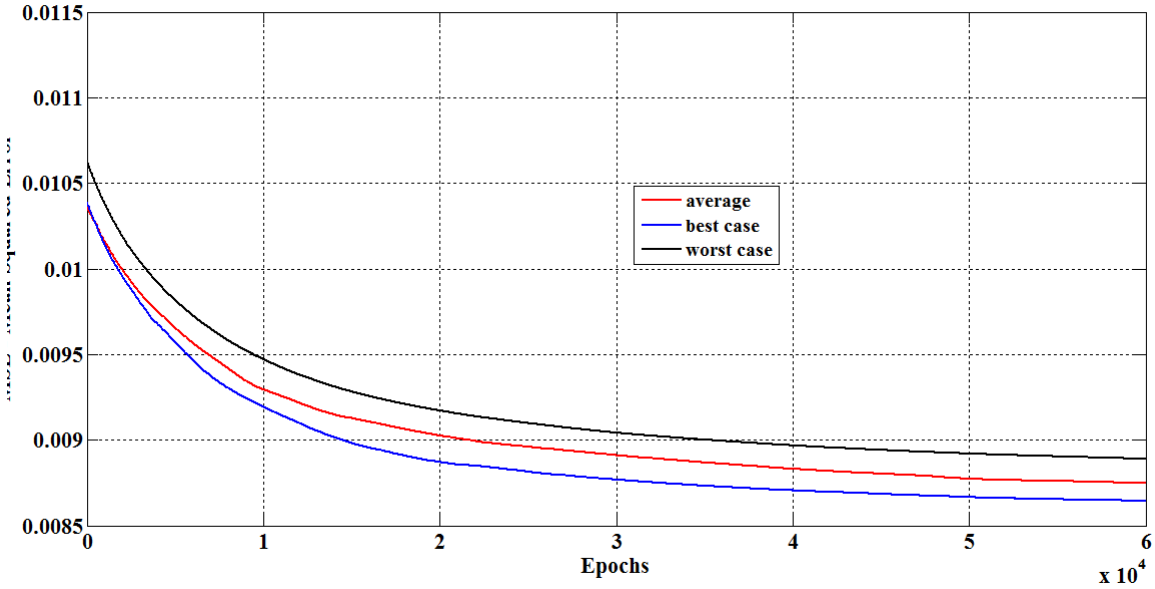

Figure 16 : Learning curves for training set of mini-region B (without sampling) (2011).

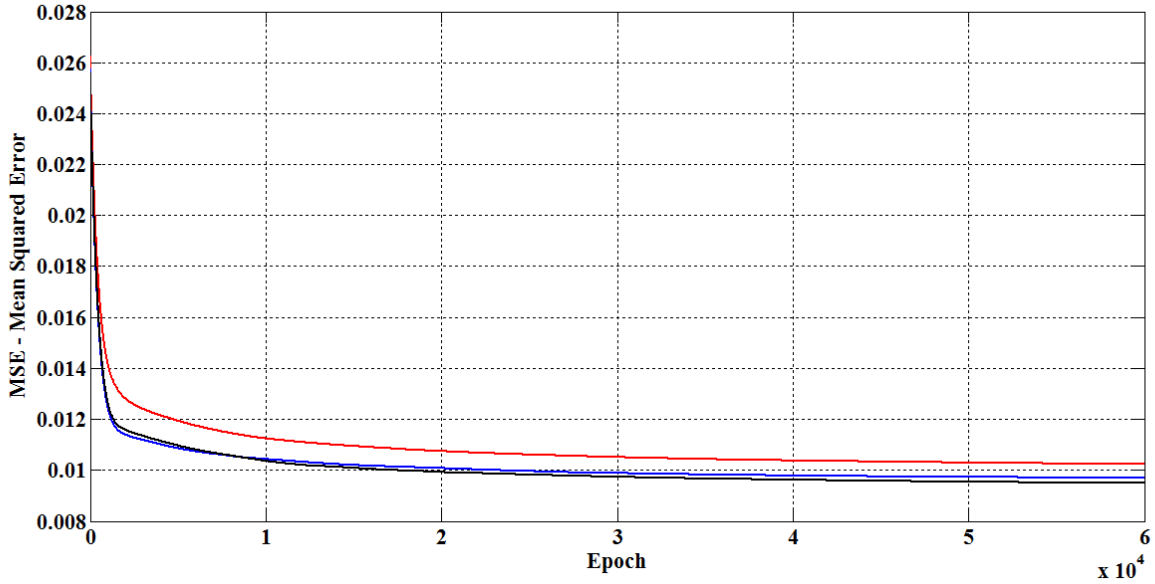

Figure 17 : Some examples of error curves for the validation set of mini-region B (without sampling) (2011).

5.3 Proposed ANN, mini-region B (years 2007-2011, joint training)

These results were obtained with the proposed ANN only for mini-region B, but without using sampling in the test dataset. Training was performed using data of all these years (2007-2011) and test was performed separately for each year. Classification performances were a little bit lower than those of previous section that employed a separate training for each year. Again, Kappa values were low due to class imbalance. This results suggest that there may be some particular meteorological issues for some years. These seasonal aspects are beyond the scope of this work.

Table 13 : Confusion matrices, statistics and performance evaluation indices for mini-region B (joint training) (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7321**
Actual	NSCA	**1432/1634**	**202/1634**	**1412/1432/1444 (13)**	**190/202/222 (13)**	**Accuracy:0.8771**
	SCA	**98/816**	**718/816**	**89/65/118 (11)**	**698/751/727 (11)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.1481**
Actual	NSCA	**1252/1420**	**168/1420**	**1238/1252/1270 (13)**	**150/168/182 (13)**	**Accuracy:0.8801**
	SCA	**5/23**	**18/23**	**3/5/8 (2)**	**15/15/20 (2)**	

Table 14 : Confusion matrices, statistics and performance evaluation indices for mini-region B (joint training) (2008).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7321**
Actual	NSCA	**1432/1634**	**202/1634**	**1412/1432/1444 (13)**	**190/202/222 (13)**	**Accuracy:0.8771**
	SCA	**98/816**	**718/816**	**89/65/118 (11)**	**698/751/727 (11)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.1184**
Actual	NSCA	**1243/1424**	**181/1424**	**1209/1243/1263 (22)**	**161/181/215 (22)**	**Accuracy:0.8718**
	SCA	**4/19**	**15/19**	**1/4/8 (3)**	**11/15/18 (3)**	

Table 15 : Confusion matrices, statistics and performance evaluation indices for mini-region B (joint training) (2009).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7321**
Actual	NSCA	**1432/1634**	**202/1634**	**1412/1432/1444 (13)**	**190/202/222 (13)**	**Accuracy:0.8771**
	SCA	**98/816**	**718/816**	**89/65/118 (11)**	**698/751/727 (11)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.0100**
Actual	NSCA	**1286/1440**	**154/1440**	**1264/1286/1302 (16)**	**138/154/176 (16)**	**Accuracy:0.8925**
	SCA	**1/2**	**1/2**	**0/1/2 (1)**	**0/1/2 (1)**	

Table 16 : Confusion matrices, statistics and performance evaluation indices for mini-region B (joint training) (2010).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7321**
Actual	NSCA	**1432/1634**	**202/1634**	**1412/1432/1444 (13)**	**190/202/222 (13)**	**Accuracy:0.8771**
	SCA	**98/816**	**718/816**	**89/65/118 (11)**	**698/751/727 (11)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.0000**
Actual	NSCA	**1255/1440**	**185/1440**	**1239/1255/1282 (19)**	**158/185/201 (19)**	**Accuracy:0.8715**
	SCA	**0/0**	**0/0**	**0/0/0 (0)**	**0/0/0 (0)**	

Table 17 : Confusion matrices, statistics and performance evaluation indices for mini-region B (joint training) (2011).

		Average of 20 executions		Min./Average/Max. (Standard deviation)		Performance indexes
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.7321**
Actual	NSCA	**1432/1634**	**202/1634**	**1412/1432/1444 (13)**	**190/202/222 (13)**	**Accuracy:0.8771**
	SCA	**98/816**	**718/816**	**89/65/118 (11)**	**698/751/727 (11)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.3395**
Actual	NSCA	**1146/1324**	**178/1324**	**1101/1146/1174 (34)**	**150/178/223 (34)**	**Accuracy:0.8626**
	SCA	**15/78**	**63/78**	**11/15/17 (3)**	**61/63/67 (3)**	

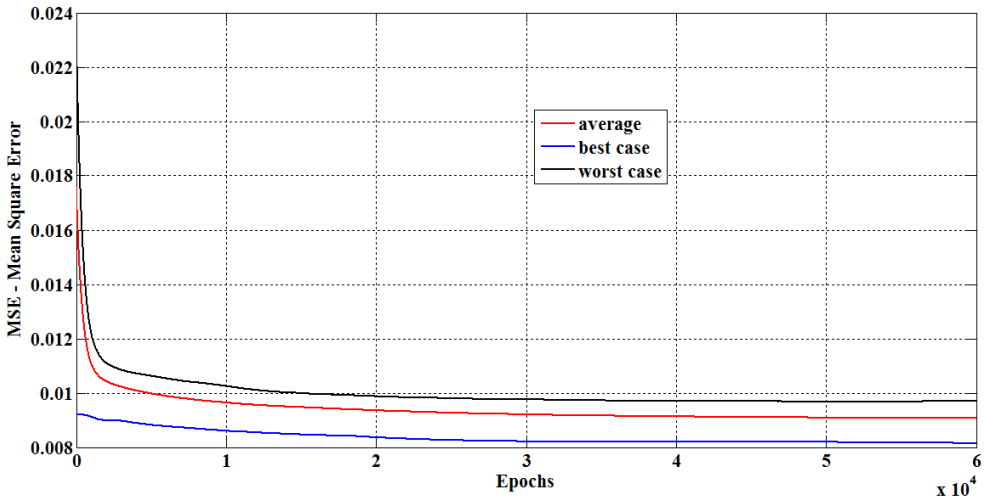

Figure 18 : Learning curves for training set of mini-region B (joint training) (2007).

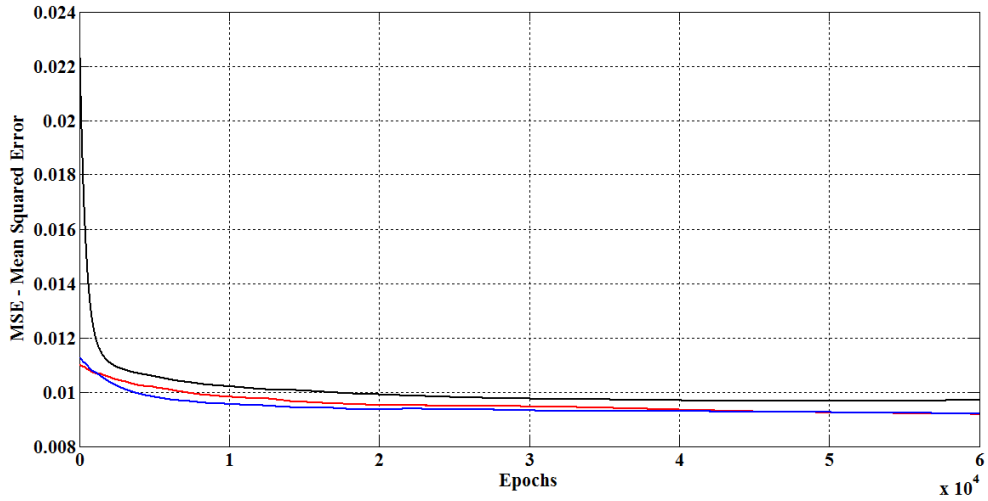

Figure 19 : Some examples of error curves for the validation set of mini-region B (joint training) (2007).

5.4 Standard backpropagation ANN for mini-regions A, B and C (year 2007)

In order to evaluate the proposed training algorithm, additional tests were performed with the same proposed network architecture that was shown in Fig. 1, but performing training using a standard backpropagation algorithm [22]. These tests employed sampling in the test set. The results clearly show that the standard backpropagation failed to prevent the predominance of **NSCA** patterns in the network learning, i.e. the network was "enslaved" by such patterns. Classification hits for **NSCA** class were high, in both training and validation, but classification performance for **SCA** class was very poor. It can be concluded that the standard backpropagation failed to find a good decision boundary between the two classes. This is probably due to its inability to get out of local minima in the error surface without disturbing the network learning [31, 32, 33].

Table 18 : Confusion matrices, statistics and performance evaluation indices for mini-region A (with sampling) (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.5532**
Actual	NSCA	**184/200**	**16/200**	**178/184/190 (3)**	**10/16/22 (3)**	**Accuracy:0.8133**
	SCA	**40/100**	**60/100**	**30/40/52 (6)**	**48/60/70 (6)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.5055**
Actual	NSCA	**39/46**	**7/46**	**36/39/42 (2)**	**4/7/10 (2)**	**Accuracy:0.7826**
	SCA	**8/23**	**15/23**	**2/8/15 (4)**	**8/15/21 (4)**	

Table 19 : Confusion matrices, statistics and performance evaluation indices for mini-region B (with sampling) (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.2246**
Actual	NSCA	**506/550**	**44/550**	**498/506/510 (4)**	**40/44/52 (4)**	**Accuracy:0.7042**
	SCA	**200/275**	**75/275**	**185/200/212 (8)**	**63/75/90 (8)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.2308**
Actual	NSCA	**43/46**	**3/46**	**40/43/45 (2)**	**1/3/6 (2)**	**Accuracy:0.7101**
	SCA	**17/23**	**6/23**	**12/17/20 (2)**	**3/6/11 (2)**	

Table 20 : Confusion matrices, statistics and performance evaluation indices for mini-region C (with sampling) (2007).

		Average of 20 executions		**Min./Average/Max. (Standard deviation)**		**Performance indexes**
Training		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.5963**
Actual	NSCA	**128/142**	**14/142**	**126/128/133 (2)**	**9/14/16 (2)**	**Accuracy:0.8263**
	SCA	**23/71**	**48/71**	**19/23/25 (2)**	**46/48/52 (2)**	

Test		**Predicted**		**Predicted**		
		NSCA	SCA	NSCA	SCA	**Kappa: 0.4124**
Actual	NSCA	**46/52**	**6/52**	**41/47/50 (3)**	**2/5/11 (3)**	**Accuracy:0.7564**
	SCA	**13/26**	**13/26**	**0/3/8 (3)**	**18/23/26 (3)**	

6 Concluding Remarks

This work presented a new neural network architecture and training algorithm for the detection of patterns associated to severe convective activity. The network was trained with patterns composed of selected meteorological variables values given by the numerical weather forecasting model Eta. The classes of atmospheric convective activity were defined by values of density of occurrence of atmospheric electrical discharges. Classification results show that the proposed architecture and training algorithm are promising, but still lack the performance required by its use as an ancillary tool for meteorologists in the detection of severe convective activity.

The frequency of occurrence approach [26] described in Section 3 performed well in the cross validation scheme, but was unable to perform classification using a prediction-like scheme as the current neural network made. However, test results with the proposed ANN show the existence of a number of false negatives (**SCA** patterns mistakenly classified as being **NSCA** ones) indicate that the network design can still be improved. We intend to do further work in order to improve the classification performance. Training patterns could be preprocessed using techniques such as Rough Sets, Principal Component Analysis or even Kohonen Self-Organized Maps, in order to reduce the class overlap. Another alternative would be to imbed in the training algorithm concepts derived from other computational intelligence paradigms such as Fuzzy Logic. This may be important to improve the network capacity to handle data at the boundaries between classes.

Another approach would be to adopt a three-class classification (low/absent, moderate and severe convective activity). In such a scheme, the middle class would act as a kind of cushion between the other two classes. This would minimize the impact of classification errors: for instance, a pattern corresponding to severe convective activity may be misclassified as being associated to moderate instead of low/absent. It is possible that such three-class neural network be more robust since the main goal is to indentify patterns corresponding to severe convective activity. However, it is difficult to say that its overall performance would be better.

A large number of tests was performed showing that the proposed ANN was able to detect most of the SCA patterns, as expected. The number of false positives and false negatives were relatively low. Values of the Kappa index and classification accuracy were typically above 80%, but the operational use of this ANN would require accuracies above 95%. The proposition of detecting patterns associated to severe convective activity from a weather forecast model is relatively new. Considering the results presented in this work, we may conclude that the proposed ANN formed better than our previous approach (*frequence of occurrence* classifier [2]) and has still room for improvements.

7 Acknowledgments

Authors thank CNPq (National Council for Scientific and Technological Development of Brazil) for the post-doctorate grant 159767/2010-5 (Glauston R. Teixeira de Lima) and grant PQ 313729/2009-3, currently PQ 305639/2012-9 (Stephan Stephany). Authors also thank the Center for Weather Forecasts and Climate Studies (CPTEC/INPE) for the Eta numerical weather forecasting model meteorological data and for the atmospheric electrical discharge data. Finally, authors thank, in memoriam, to Jose Demisio Simoes da Silva for his leadership and companionship, and also his overwhelming contribution to this work.

8 References

1. Black T L (1994) NMC notes: The new NMC mesoscale Eta model: description and forecast examples. Weather and Forecasting 9(2):265-278
2. Lima G R T, Stephany S (2013) A new classification approach for detecting severe weather patterns. Computer and Geoscience (in press)
3. Hu M J C (1964) Application of the Adaline system to weather forecasting. Master Thesis, Technical Report 6775-1, Standard Electronic Laboratories, Stanford-CA
4. Koskela T, Lehtokangas M, Saarinen J, Kaski K (1996) Time series prediction with multilayer perceptron, FIR and Elman neural networks. In: Proceedings of the World Congress on Neural Networks, San Diego
5. Corchado J, Fyfe C (1999) Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering 13(4):351-357
6. Pal N R, Pal S, Das J, Majumdar K (2003) SOFM – MLP: A hybrid neural network for atmospheric temperature prediction. IEEE Transactions on Geoscience and Remote Sensing 41(12):2783-2791
7. Maqsood I, Khan M R, Abraham A (2004) An ensemble of neural networks for weather forecasting. Neural Computing & Applications 13:112–122

8. Shikoun N, El-Bolok H, Ismail M A (2005) Climate change prediction using data mining. International Journal of Intelligent and Cooperative Information Systems 5(1):365-379

9. Paras M S, Kumar A, Chandra M (2007) A feature based neural network model for weather forecasting. In: Proceedings of World Academy of Science, Engineering and Technology, vol. 34, pp 66-73

10. Lai L L, Braun H, Zhang Q P, Wu Q, Ma Y N, Sun W C, Yang L (2004) Intelligent weather forecast. In: IEEE Proceedings of the Third International Conference on Machine Learning and Cybernetics, vol 7, pp-4216-4221

11. Santhanam T, Subhajini A C (2011) An efficient weather forecasting system using radial basis function neural network. Journal of Computer Science 7 (7): 962-966

12. Cofiño A S, Cano R, Sordo C, Gutiérrez J M (2002) Bayesian networks for probabilistic weather prediction. In: Proceedings of the 15th European Conference on Artificial Intelligence, pp 695-700

13. Hall T, Brooks H E, Doswell III C. A (1998) Precipitation forecasting using a neural network. Weather and Forecasting, 14:338-345

14. Kuligowski R J, Barros A P (1998) Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. Weather and Forecasting 13(4):1194-1204

15. Hung N Q, Babel M S, Weesakul S, Tripathi N K (2009) An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrology and Earth System Sciences 13:1413–1425

16. Collins W G, Tissot P (2007) Use of an artificial neural network to forecast thunderstorm location. Preprints of the fifth Conference on Artificial Intelligence Applications to Environmental Science at AMS Annual Meeting, San Antonio, TX, Paper number 2.2

17. Mandal S N, Choudhury J P, Chaudhuri S R B, De D (2008) Soft computing approach in prediction of a time series data. Journal of Theoretical and Applied information Technology 4(12):1131-1141

18. Deji W, Bo X, Faquan Z, Jianting L, Guangcai L, Bingyu S (2009) Climate prediction by SVM based on initial conditions. In: IEEE Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, pp-578-581

19. Radhika Y, Shashi M (2009) Atmospheric temperature prediction using SVM. International Journal of Computer Theory and Engineering 1(1):55-58

20. Latha B C, Paul S, Kirubakaran E, Sathianarayanan A (2010) Service oriented architecture for weather forecasting using data mining. International Journal of Advanced Networking and Applications 2(2):608-613

21. Fausett, L. 1994: Fundamentals of neural networks: architectures, algorithms and applications. Upper Saddle River-NJ: Prentice-Hall, Inc., 461p

22. Haykin S O (2008) Neural networks and learning machines. 3rded. Upper Saddle River- NJ: Pearson/Prentice-Hall, Inc., 936p

23. Samarasinghe S (2006) Neural Networks for applied sciences and engineering: from fundamentals to complex pattern recognition. New York-NY:Auerbach Publications, 570p

24. Vicente G A, Scofield R A, Menzel W P (1998) The operational GOES infra-red estimation technique. Bulletin of the American Meteorological Society, 79:1883-1898

25. Huffman G J, Adler R F, Bolvin D T, Gu G, Nelkin E J, Bowman K P, Hong Y, Stocker E F, Wolff D B (2007) The TRMM multisatellite precipitation analysis (TMPA): quasi-Global, multiyear, combined-sensor precipitation estimates at fine scales. Journal of Hydrometeorology 8(1):38-55

26. Rozante J R, Moreira D S, Goncalves L G G, Vila D A (2010) Combining TRMM and surface observations of precipitation: technique and validation over South America. Weather and Forecasting, 25(3):885-894

27. Pessoa A S A, Lima G R T, Silva J D S, Stephany S, Strauss C, Caetano M, Ferreira N J (2012) Mining of meteorological data to forecast severe convective events (in Portuguese). Brazilian Journal of Meteorology 27(1):

287-294

28. Cover T M (1965): Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers EC-14: 326–334

29. Krogh A, Hertz J A (1992) A simple weight decay can improve generalization. In: Advances in Neural Information Processing Systems. J E Moody, S J Hanson and R P Lippmann eds. Morgan Kauffmann Publishers, San Mateo, CA, pp. 950-957.

30. Congalton R G, Green K (2008): Assessing the accuracy of remotely sensed data: principles and practices. Boca Raton-FL: Taylor and Francis/CRC Press, 183p

31. Gori M, Tesi A (1992) On the problem of local minima in backpropagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(1):76–86

32. Bi W, Wang X, Zeng T, Tamura H (2005) Avoiding the local minima problem in backpropagation algorithm with modified error function. IEICE Transactions Fundamentals, E88-A (12):3645–3653

33. Xiong J-J, Zhang H (2003) Research on the problem of neural network convergence. In: IEEE Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi-an, p. 1132-1134