

ORIGINAL RESEARCH

Solar radiation: Cloudiness forecasting using a soft computing approach

Vassiliki H. Mantzari¹, Dimitrios H. Mantzaris²

1. Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens, Athens, Greece. 2. Informatics Laboratory, Department of Molecular Biology and Genetics, Democritus University of Thrace, Alexandroupolis, Greece.

Correspondence: Dimitrios H. Mantzaris. Address: Informatics Laboratory, Department of Molecular Biology and Genetics, Democritus University of Thrace, GR-68100, Alexandroupolis, Greece. Telephone: 30-697-938-9130. Email: dmantzar@med.duth.gr

Received: February 23, 2012

Accepted: June 25, 2012

Online Published: December 20, 2012

DOI: 10.5430/air.v2n1p69

URL: <http://dx.doi.org/10.5430/air.v2n1p69>

Abstract

Solar energy is one of the most important energy sources with increasing penetration into the power supply systems of many countries, due to the reduced environmental impact of its operation. One of the important factors of the efficiency of photovoltaic systems is the predictability of solar radiation, which depends on the clouds and the meteorological conditions, the occurrence of which is a non-linear process. Prediction of clouds' amount that affects solar radiation using Artificial Neural Networks (ANNs) is presented in this paper. To our knowledge, this approach is the first computational intelligence method that refers to cloudiness forecasting. Input parameter selection is very critical in ANN design. In this study, ten meteorological and temporal variables were selected in the implementation of the ANN structure. The appropriate ANN architecture consists of a hidden layer with hyperbolic tangent sigmoid transfer function and an output layer with a saturating linear transfer function. The most important benefits of the proposed method are the cloudiness forecasting in the next half hour, the embedment of temporal parameters for cloudiness prediction and the highest quantity of data comparatively with other similar studies.

Key words

Cloudiness forecasting, Solar radiation, Photovoltaic efficiency, Computational intelligence, Feed forward neural networks, Multi-layer perceptrons

1 Introduction

Solar energy is one of the most important energy sources with increasing penetration into the power supply in the form of heat or photovoltaic systems, providing reduced environmental impacts. The solar energy systems are preferred to their source of energy because the sun is a renewable source; although, their penetration is reduced due to various factors. One of the most important factors is the predictability of solar radiation, which is a complicated task. The weather variation is a continuous, data-intensive, multidimensional, dynamic and chaotic process, so that these properties make weather forecasting a complicated procedure. Additionally, the solar radiation depends on the clouds and the meteorological conditions are non-linear processes, which affect the efficiency of photovoltaic systems.

The empirical and the dynamic approaches are used in weather forecasting and were introduced by Lorenz in 1969^[2]. Specifically, the cloudiness prediction is based on image processing analysis, where the use of threshold for pixel intensity, determines the clear as well as the cloudy sky^[1]. The spectral measurements of solar irradiance are another technique for cloudiness estimation. Furthermore, numerical methods have been developed in order to face the weather prediction problem^[3-5]. The most weather prediction systems use a combination of empirical and dynamic techniques^[6].

The accurate weather prediction is crucial for the planning of energy production. Specifically, the solar radiation affects the performance of photovoltaic systems. The latitude, the season of the year, the altitude, the reflection, the weather conditions, such as cloud amount, air humidity, air pollution, and the path of radiation in the atmosphere are main parameters for the estimation of solar radiation. The cloudiness is one of the most important factors for the predictability of solar irradiation.

Soft Computing is a set of methodologies implementing computational intelligence systems for solving non-linear real world problems, without prior knowledge and symbolic representation of their rules^[7]. Artificial Neural Networks, Fuzzy Logic, Evolutionary Programming, Genetic Algorithms, Mimetic Algorithms and Artificial Immune Systems are subfields of soft computing.

Artificial Neural Networks (ANNs) are powerful alternative - data modeling - tools that are able to correlate complex input and output data. In the recent past, the ANNs have been applied to model large data with large dimensionality^[8-10].

The ANN approach has several advantages over conventional phenomenological or semi-empirical models, since it requires known input data set without any assumptions^[10, 11]. It exhibits rapid information processing and is able to develop a mapping of the input and output variables. Such a mapping can subsequently be used to predict desired outputs as a function of suitable inputs^[10].

The ANNs are used in ample scope applications, such as weather prediction^[12-14], heat transfer prediction^[15], short term load forecasting^[16-19], numerical simulation of nonlinear equations^[20, 21], financial prediction^[22, 23], telecommunications, medicine^[24, 25], signal processing and other fields^[26-28].

In an ANN design stage, the implementation of the appropriate ANN architecture to solve a real-world problem is relatively complex. A neural network with few neurons implies inadequate lore, while a big one leads to poor generalization ability, presenting overfitting^[29]. Usually, trial and error strategy is the most common way for specification of an ANN architecture^[30].

This study focuses on the development and assessment of ANN architectures based on Multi-Layer Perceptrons (MLPs) and the application of these implementations to the problem of cloudiness forecasting. This problem is important as the clouds influence the photovoltaic cells' efficiency and therefore additional generating reserves are required to cover the energy requirements. The specified ANN and the obtained results are presented in this paper.

Meteorological stations measure data useful to evaluate the weather conditions of a specific area. The proposed neural system, which is a computational intelligent implementation and uses stored experimental knowledge, forecasts the amount of clouds. The combination of meteorological and temporal parameters as inputs of the proposed artificial neural system results the development of an artificial neural technique for the cloudiness prediction, instead of four implementations according to the year seasons. Moreover, the ability of the suggested neural network structure to estimate the cloudiness trend in the next half hour is one important technological innovation.

2 Artificial neural networks theory

The aim of this study is the implementation of a non-symbolic learning approach, called ANN, for cloudiness forecasting. An ANN is a computational system, consisting of simple processing elements, called neurons, operating in parallel,

interconnected to each other, and has the ability to use stored experimental knowledge^[31]. The acquired knowledge of an ANN from its environment through a learning process, the use of interneuron connection strengths (synaptic weights) for storing this knowledge, and the generalization ability based on the information from the input data, are evidences that an ANN resembles human brain in forecasting rainfall^[32] from structural as well as functional point of view.

ANNs are robust systems as their performance doesn't degrade appreciably if some of their neurons or interconnections are lost. Moreover, ANNs are capable to recall information based on incomplete or noisy or partially incorrect inputs and they can be self-organizing. More specifically, some neural network architectures can generalize from data patterns used in training without being provided with specific instructions on exactly what to learn^[33].

Parallel processing and neural network's training are the differences between von Neuman computer and ANN architecture^[34].

The object function of an ANN is the mapping from a given data set to features or relationships among the data and based on examples containing trusted set of input/output data, where the training algorithm iteratively adjusts the values of connection weights (synapses), and these synaptic weights store the necessary knowledge to solve specific problems^[32, 35, 36]. The process that ensures that ANN estimates the correct output in all known situations is called training^[37], similar to biological systems and it is based on learning rules^[31].

Many learning algorithms have been proposed for the development of ANN, such as Hebb's rule, the Delta rule also called Least Mean Square (LMS) learning rule or Widrow – Hoff learning rule, the Gradient Descent Rule, Kohonen's learning algorithm, and Hopfield law^[38, 39]. However, when the input data does not contain information related to the desired output, then the ANN never learns^[38].

Beyond the learning rules, many ANN models have been proposed. The most usual architectures are Perceptrons, Multi-Layer Perceptrons (MLPs), Radial Basis Functions (RBF) networks, Self-Organizing Maps (SOM), Learning Vector Quantization (LVQ) networks and recurrent networks^[39]. The forenamed architectures^[28, 38-40] are not recommended for all application fields. The suitable ANN architecture is specified by the requirements of each application.

Perceptrons include one hidden layer of neurons and are used as classifiers. However, they can classify linear separable input vectors, so the nonlinear separable input vectors classification is feasible using other types of networks such as Multi-Layer Perceptrons (MLPs)^[39].

The Multi-Layer Perceptron network consists of a set of sensory units that constitute the input layer, one or more hidden layers of computational nodes, and an output layer of computational nodes^[38-42]. Multilayer feed-forward neural networks (MFFNNs) are very often called backpropagation networks because of the typical training algorithm^[43, 44]. The architecture of a multi-layer feed-forward network is not completely constrained by the problem to be solved^[45].

The RBF networks have similar structure and different neuron model and learning algorithms with multi-layer perceptrons. The RBF networks are embedded into a three-layered feedforward neural network comprised of input, hidden/memory, and output neurons^[45]. The hidden layer consists of radial basis transfer function, while the output layer uses linear transfer function^[46]. The RBF networks are popular for their fast and easy training and adaptation. However, these advantages bring some drawbacks too. The main disadvantage of RBF network is the high memory requirement, because the number of neurons in the hidden layer is equal to the number of training data^[44]. The key factor that was chosen for testing was a spread parameter that defines the smoothness of the approximation function^[46].

Two variants of RBF networks are Generalized Regression Neural Networks (GRNN) and Probabilistic Neural Networks (PNN). A GRNN has a radial basis function layer and a special linear layer. It is often used for function approximation. PNN consists of a radial basis function layer and a competitive layer, and can be used for classification problems. In spite

of the GRNN and PNN advantages, they have a major drawback. They operate slower because of using more computation than other kinds of networks to do their function approximation or classification ^[31, 39].

The Self-Organizing Maps (SOMs) have the ability to detect regularities and correlations in their input. In particular, SOMs learn to classify input vectors according to how they are grouped in the input space. The SOM learn both the distribution and topology of the input vectors that they are trained on ^[31, 39].

The Learning Vector Quantization (LVQ) networks classify input vectors into target classes by using a competitive layer to find subclasses of input vectors, and then by combining them into the target classes ^[39]. The LVQ networks can classify any set of input vectors, not just linearly separable sets of input data. The only requirement is that the competitive layer must have enough neurons ^[39].

A recurrent network has feedback connection either from a hidden layer or from the output layer to its input. Elman and Hopfield networks are two variants of recurrent networks ^[31, 39].

The Elman networks, known as partial recurrent networks or simple recurrent networks, are two-layer backpropagation networks with addition of an internal feedback loop. This feedback path, which connects the output of hidden layer to network's input, allows Elman networks to learn to recognize and generate temporal patterns, as well as spatial patterns, so Elman networks are useful in signal processing and prediction areas where the time plays an ascendant role ^[47].

The Hopfield networks are recursive in the way that the output is fed back to input. Hopfield networks can act as error correction or vector categorization networks. These networks are interesting from a theoretical aspect, but are seldom used in practice ^[39]. The feedback loops involve the use of particular branches composed of unit-delay elements, which result in a nonlinear dynamic behavior by virtue of the nonlinear nature of the neurons ^[45].

3 Data collection

Artificial Neural Networks have shown tremendous improvement over the traditional methods to handle the non-linear problems, especially in the field of weather forecasting. This study proposes a computational method namely MLP for cloudiness forecasting, as it is an important feature of the evaluation of solar radiation effectiveness of photovoltaic systems.

The cloud is a set of visible vapor, or very fine hydrodroplet, or very fine ice crystals, or a combination of the foregoing, from the condensation of water vapor in the atmosphere. Two ways are responsible for the clouds' creation. The first one is based on the release of electrons during a storm. The electrons create clusters with the molecules of atmosphere, and then the electrons are released in order to produce stable neutral bonds for creation of clouds. The second way is result of the ionization that binds water molecules in the atmosphere and the high concentration achieved results in clouds generation.

The weather data used at the implementation and testing of the MLP model was obtained from the Weather Underground ^[48] for a period of sixteen years (1996 - 2011). The data collection based on information about the cities of Alexandroupolis, Thessaloniki, Athens and Heraklion, Greece.

The database includes readings of the real time observation of the meteorological parameters recorded at every half hour interval. These meteorological variables are air temperature, dew point, air humidity, sea level pressure, visibility, wind speed, wind direction and amount of clouds. The aforementioned parameters selected because these meteorological variables are countable amounts and are measured directly in weather stations. Other parameters are related to these measured data and can be derived with the help of a direct or empirical relationship. Addition, studies that focus on weather forecasting ^[15, 49], predicting the weather, separated their available data in seasons (two or more), in order to

include the period of the year were recorded the meteorological data. This work took into account three temporal factors that are the month, the day and the time of recording meteorological data, facing the separation of data as mentioned in forenamed papers. Consequently, the implemented neural networks had ten input parameters which are depicted in Table 1. The 1st column of Table 1 corresponds the ANN inputs to the meteorological and temporal variables (2nd column), the units of which are presented in 3rd column of the same sample.

Table 1. Meteorological and temporal variables.

NN inputs	Parameter	Unit
1	Air temperature	Deg.C
2	Dew point	Deg.C
3	Air humidity	%H
4	Sea level pressure	hPa
5	Visibility	Km
6	Wind speed	Km/h
7	Wind direction	Deg
8	Month	Number: 1 – 12
9	Date	Number: 1 – 31
10	Time	Minutes: 0 - 1439

Neural networks generally provide improved performance with normalized data. The use of original data as the ANN input may cause a convergence problem^[50]. Consequently, the training and testing datasets are transformed into values between -1 and 1 by the following equation:

$$nv = \frac{(\min_value - act_value)}{(\max_value - \min_value)} \quad (1)$$

where max_value and min_value are maximum and minimum values, act_value is original dissimilarity and nv is normalized dissimilarity.

The main goal of normalization, in combination with weight initialization, is to allow the squashed activity function to work at least at the beginning of the learning phase. Thus, the gradient, which is a function of the derivative of the non-linearity, will always be different from zero. At the end of each algorithm, the outputs were demoralized into the original data format for achieving the desired result^[49, 51].

The aim of this work is to apply the best ANN implementation in estimating the amount of clouds. Therefore, the constructed ANNs have one output, which, corresponds to the amount of clouds.

The available data set, for intelligent cloudiness forecasting in the city of Alexandroupolis, Greece, is divided into a set of fifteen years (1996-2010) records and another set of one year (2011) records. The former was used for training of the ANNs, while the latter for testing.

4 ANN architecture for cloudiness forecasting

The MLP network's ability to map independent (input) and dependent (target) variables finding out the nonlinear cohesion between them is an important advantage in data processing^[52]. An MLP, of which a general scheme is presented in Fig. 1, with backpropagation learning algorithm, is the most widely used architecture for prediction^[49]. Cloudiness forecasting will be approached by an MLP neural network as well.

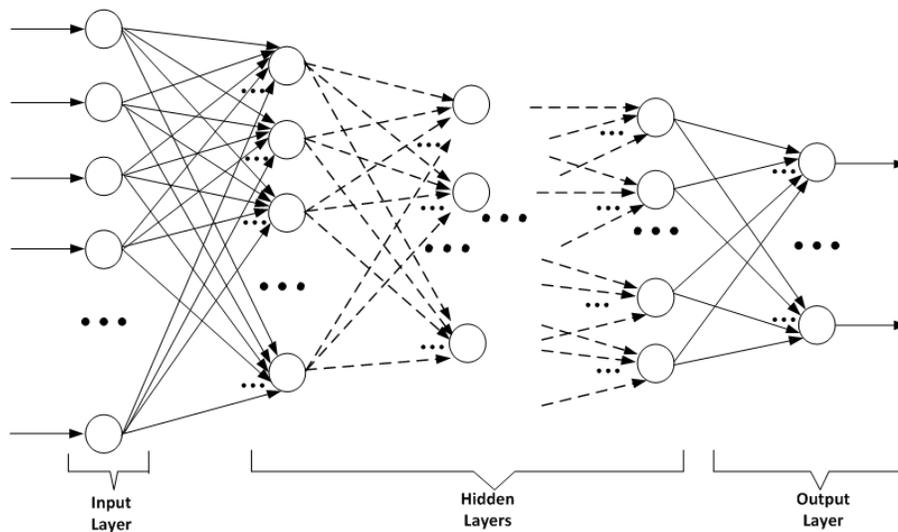


Figure 1. Multi-Layer Perceptron Network essential scheme

The name of backpropagation learning algorithm refers to the backward propagation of the error, which is transferred from the output nodes to the inner nodes. Technically, it is explained that back propagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. This gradient is always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Frequently the term “back propagation” is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent ^[32].

Multi-layer feed-forward networks transmit the provided data from input layer towards their output layer. The architecture of an MLP depends on the problem to be solved. The number of neurons in the input and output layer is constrained by the given number of inputs and the desired number of outputs, respectively, required by the problem ^[38]. According to recorded weather and temporal data, as they are described in section 3, the total number of input parameters is ten; therefore this is also the number of input nodes in the input layer of ANN. Moreover, as the cloudiness forecasting is based on the amount of clouds, there is a neuron in output layer that indicates it. However, the number and the size of layers between input and output layers are up to the designer.

The determination of appropriate number of hidden layers, hidden neurons, connections and transfer functions, which represents the ANN structure, was achieved by trial and error. The used transfer functions were four, namely hyperbolic tangent sigmoid, log sigmoid, positive linear and saturating linear ^[31, 39]. Mathematical equations of these transfer functions are depicted in Table 2.

During the MLP training process, the Levenberg-Marquardt backpropagation, the gradient descent with momentum backpropagation, and the gradient descent with adaptive learning rate backpropagation algorithms are selected for ANNs' training ^[31]. These learning algorithms based on backpropagation rule were used in order to speed up the convergence.

The mean absolute error (MAE) is a common measure of forecast error in time series analysis ^[53]. Therefore, the MAE is used as evaluation criterion of performance of ANNs. The MAE's mathematical notation is given by following equation:

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |t_k - a_k| = \frac{1}{N} \sum_{k=1}^N |e_k| \quad (2)$$

where N is the number of patterns-cases, t_k , a_k and e_k are the desired output, the ANN's calculated and the error value for each pattern, respectively.

Table 2. Transfer Functions.

Transfer Function	Mathematical Equation
Hyperbolic tangent sigmoid (tansig)	$f(x) = \frac{2}{1 + e^{-2x}} - 1$
Log sigmoid (logsig)	$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x \geq 0 \end{cases}$
Positive linear (poslin)	$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x \geq 1 \end{cases}$
Saturating linear (satlin)	$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x \geq 1 \end{cases}$

5 Experimental results

The development and performance assessment of ANN models were based on MATLAB Neural Network Toolbox, due to its effectiveness and user-friendly interface ^[39].

In Table 3, the results of best-implemented ANNs for the cloudiness forecasting are summarized. The implemented ANN architectures (2nd column), the transfer function (3rd column) the training algorithm (4th column) and the learning rate (if it exists, 5th column) of each architecture are presented.

The Nr 1-4 ANNs were trained by Levenberg-Marquardt backpropagation algorithm. The modification of transfer functions in ANNs' layers implies different obtained results. The gradient descent with momentum backpropagation algorithm is the training method for the 5th, the 6th and the 7th ANN. Again, the biased learning rate results modified ANN performance. The 8th and the 9th ANNs were trained by gradient descent with adaptive learning rate backpropagation algorithm. The last algorithm adapts the learning rate during training phase; however the appropriate modification of learning rate demands the precise adjustment of algorithm's parameters. The characteristic of the above analysis is that the used learning algorithm and its parameters specify the neural network's performance.

The performances of aforementioned ANN architectures, during training and simulating phase, are shown in Table 4. The mean absolute error (MAE) for training and testing phase is presented in the 2nd and 3rd column. The 4th column of Table 4 presents the percentage of successful amount of clouds forecasting over the testing set.

During the training phase of an ANN, the values of synaptic weights, between artificial neurons, are modified in order to decrease the error in the training data set. Moreover, it is expected that the error obtained by the used input data set for ANN testing, tends to decrease according to training error as the ANN performs a particular function.

Nevertheless, ANNs can exhibit overfitting behavior ^[29, 35]. An ANN that is too complex may fit the noise, not just the signal, leading to overfitting. Overfitting is especially dangerous because it can easily lead to forecasts that are far beyond the range of the training data in many of the common types of ANNs ^[35, 54]. The overfitting occurs in case of very small error during the training phase, but large error during the testing phase ^[29, 37].

Table 3. Results of different artificial neural network architectures

Nr of ANN	Architecture of Artificial Neural Network	Transfer Function	Training Algorithm	Learning Rate
1	10 – 7 – 3 – 1	tansig logsig poslin	Levenberg-Marquardt backpropagation algorithm	Does not exist
2	10 – 7 – 3 – 1	tansig logsig satlin	Levenberg-Marquardt backpropagation algorithm	Does not exist
3	10 – 4 – 2 – 1	tansig logsig satlin	Levenberg-Marquardt backpropagation algorithm	Does not exist
4	10 – 6 – 1	tansig satlin	Levenberg-Marquardt backpropagation algorithm	Does not exist
5	10 – 6 – 1	tansig satlin	Gradient descent with momentum backpropagation algorithm	0.01
6	10 – 6 – 1	tansig satlin	Gradient descent with momentum backpropagation algorithm	0.005
7	10 – 6 – 1	tansig satlin	Gradient descent with momentum backpropagation algorithm	0.1
8	10 – 6 – 1	tansig satlin	Gradient descent with adaptive learning rate backpropagation algorithm.	0.01
9	10 – 6 – 1	tansig satlin	Gradient descent with adaptive learning rate backpropagation algorithm.	0.009

Table 4. Experimental results using different artificial neural network architectures.

Nr	MAE Over the Training Set	MAE Over the Testing Set	Percentage (%) of Successful Forecasting Over the Testing Set
1	4.39E-09	2.3123	52.1
2	4.51E-20	0.3446	60.5
3	0.187000	0.1120	79.2
4	4.855E-22	0.0177	87.3
5	8.59E-19	0.2113	84.6
6	0.014567	0.2191	73.1
7	0.011278	0.0790	98.9
8	7.26E-11	0.2354	78.2
9	0.0005286	0.2321	85.5

From Table 4, it is obvious that the smaller the MAE in training set, the worse performance of ANN exists, keeping the other parameters constant. The network has memorized the training patterns, but it has not learnt to generalize new data. The 1st, and 2nd, and 5th ANNs present overfitting problem.

It is common the error of ANN after training to be smaller than the error at testing phase^[38], as the ANN does not know the test patterns a priori. The 3rd ANN diverges from the aforementioned mode.

The ANN that has the best performance over the overall test set (97.9%) is the 7th with 3 layers (architecture 10-6-1). The output layer contains one neuron, which generates an output value forecasting the amount of clouds.

The hyperbolic tangent sigmoid transfer function is used on hidden layer and the saturating linear transfer function is applied to output neuron. Both of the transfer functions are differentiable throughout its domain. The weights and the biases of the neurons were adjusted according to gradient descent with momentum backpropagation learning algorithm^[39], and with learning rate equal to 0.1. The meteorological and temporal data sets of cities of Thessaloniki, Athens and Heraklion were used as inputs of the Nr. 7 neural network in order to test its performance. The obtained results are summarized to Table 5, where the tested city is presented in the 1st column, the mean absolute error (MAE) for data sets is presented in the 2nd one and the 3rd column depicts the percentage of successful amount of clouds forecasting over the data sets. The latitude and longitude of the cities affect the performance of the best implemented artificial neural network. Consequently, the geographic coordinates are important factors for the estimation of cloudiness.

Table 5. The performance of the best MLP architecture over testing data sets

City	MAE Over the Data Sets	Percentage (%) of Successful Forecasting Over the Data Sets
Thessaloniki	0.2119	90.2
Athens	0.9632	83.6
Heraklion	1.7891	71.9

The most important stage of an ANN implementation is its training phase. Therefore, the 10-6-1 ANN structure trained using meteorological and temporal data of the four aforementioned cities. Specifically, the available data set, for intelligent cloudiness forecasting, is divided into a set of fifteen years (1996-2010) records and another set of one year (2011) records. The former was used for training of the ANNs, while the latter for testing. The experimental results are presented in Table 6.

Table 6. Experimental results using data sets of all cities for the 10-6-1 MLP training.

City	MAE Over the Training Set	MAE Over the Testing Set	Percentage (%) of Successful Forecasting Over the Testing Set
All cities	0.072136	0.0512	97.9
Alexandroupolis	0.011278	0.0790	98.9
Thessaloniki	0.193082	0.0790	98.1
Athens	0.196523	0.0913	97.5
Heraklion	0.199867	0.0921	97.2

The implemented ANN architectures (2nd column), the transfer function (3rd column) the training algorithm (4th column) and the learning rate (if it exists, 5th column) of each architecture are presented.

The experimental results are presented in Table 6. The 1st column of the Table summarizes the cities used in order to implement and test the MLP neural network. The Mean Absolute Error (MAE) for training and testing data sets are presented in 2nd and 3rd column of the Table 6, correspondingly. The percentage of successful cloudiness forecasting over testing sets. The obtained outcomes conclude that the performance of 10-6-1 MLP, after the training with data sets of four cities, is better contrary to the results of the Table 5. The cloudiness estimation using the proposed MLP neural network is improved for Thessaloniki, Athens and Heraklion, while it is unchanged for Alexandroupolis, Greece. This performance is

prospective because the meteorological data depend on the geographic coordinates. The proposed neural network, during the training phase, discovered the meteorological data correlation with the latitude and longitude of the cities.

The results presented above were considered quite satisfactory. The acquired knowledge is of the 10-6-1 ANN architecture; assist the experts to be informed about future weather conditions in order to schedule additional generating reserves to cover the energy requirements.

6 Conclusions

The implemented MLP neural network for half hour ahead cloudiness forecasting in Alexandroupolis, Greece, has satisfactory performance and reasonable prediction accuracy according to the obtained results. The mean absolute error between the exact and predicted value and the MLP's classification performance are quite satisfactory, so that the proposed MLP architecture faces the cloudiness forecasting. The use of temporal parameters, as MLP's inputs, is an attribute that improves the MLP's performance and reduces the hardware requirements contrary to existed artificial neural networks approaches for cloudiness forecasting. Consequently, the suggested MLP model could be an important tool for the estimation of cloudiness that affects solar radiation and hence the photovoltaic cells efficiency and the scheduling of additional generating reserves to cover the energy requirements.

References

- [1] Martins, I., Ferreira, P., and Ruano, A. Estimation and prediction of cloudiness from ground-based all-sky hemispherical digital images, Proceedings of CONTROLO'2010, the 9th Portuguese Conference on Automatic Control, Coimbra, Portugal. 2010.
- [2] Lorenz, E. Three approaches to atmospheric predictability, *Bull. Am. Meteorol. Soc.* 1969; 50: 345-351.
- [3] Shuman, F. Numerical methods in weather prediction: II. Smoothing and filtering, *Monthly Weather Review.* 1957; 357-361. [http://dx.doi.org/10.1175/1520-0493\(1957\)085<0357:NMIWPI>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1957)085<0357:NMIWPI>2.0.CO;2)
- [4] Warner, T. "Numerical Weather and Climate Predictions". Cambridge University Press. 2010. <http://dx.doi.org/10.1017/CBO9780511763243>
- [5] Glahn, H., and Lowry, D. "The use of model output statistics (MOS) in objective weather forecasting". *J. App. Meteorology.* 1972; 11: 1203-1211. [http://dx.doi.org/10.1175/1520-0450\(1972\)011<1203:TUOMOS>2.0.CO;2](http://dx.doi.org/10.1175/1520-0450(1972)011<1203:TUOMOS>2.0.CO;2)
- [6] Moro, I., Alonso, L., and Vivaracho C. Application of neural networks to weather forecasting with local data, *Appl. Inf.* 1972; 12: 68-70.
- [7] Mantzaris, D., Anastassopoulos, G., Iliadis L., and et al. H. A soft computing approach for osteoporosis risk factor estimation, in: *IFIP International Federation for Information Processing, AIAI2010, Springer.* 2010; 339: 120-127.
- [8] Gevrey, M., Dimopoulos, I., and Lek, S. Review and comparison of methods to study the contribution of variables in artificial neural network models, *Ecol. Modell.* 2003; 160: 249-264. [http://dx.doi.org/10.1016/S0304-3800\(02\)00257-0](http://dx.doi.org/10.1016/S0304-3800(02)00257-0)
- [9] Maier, H., Jain, A., Dandy, G., and Sudheer, K. Methods Used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions", *Environ. Modell. Software.* 2010; 25: 891-909. <http://dx.doi.org/10.1016/j.envsoft.2010.02.003>
- [10] Nagendra, S., and Khare, M. Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions, *Ecol. Modell.* 2006; 190: 99-115. <http://dx.doi.org/10.1016/j.ecolmodel.2005.01.062>
- [11] Gardner, M., and Dorling, S. Artificial neural network (multilayer perceptron) - A review of applications in atmospheric sciences, *Atmos. Environ.*, 1998; 32: 2627-2636. [http://dx.doi.org/10.1016/S1352-2310\(97\)00447-0](http://dx.doi.org/10.1016/S1352-2310(97)00447-0)
- [12] Chattopadhyay, S. "Multilayered feed forward artificial neural network model to predict the average summer-monsoon rainfall in India", *Acta Geophys.*, 2002; 55: 369-382. <http://dx.doi.org/10.2478/s11600-007-0020-8>
- [13] Maqsood, I., Khan, M., and Abraham, A. Neuro-computing based canadian weather analysis. 2nd Int. Workshop on Intell. Syst. Des. Appl. Comput. Intell. Appl., Dynamic Publishers. Atlanta Georgia. 2002: 39-44.
- [14] Maqsood, I., Khan, M., and Abraham, A. Intelligent weather monitoring systems using connectionist models. *Int. J. Neural Parallel and Sci. Comput.* 2000; 10: 157-178.
- [15] Hayati, M., and Mohebi, Z. Temperature forecasting based on neural network approach, *World Appl. Sci. J.* 2007; 2613-2620.

- [16] Hayati, M. Short-term load forecasting using artificial neural networks for the west of Iran, *J. Appl. Sci.* 2007; 7: 1582 - 1588. <http://dx.doi.org/10.3923/jas.2007.1582.1588>
- [17] Hayati, M., and Karami, B. Application of neural networks in short – term load forecasting, 7th WSEAS Int. Conf. Math. Methods Comput. Tech. Electr. Eng., Sofia, Bulgaria. 2007; 37 - 41.
- [18] Hayati, M., and Karami, B. Application of computational intelligence in short – term load forecasting, *WSEAS Trans. on Circuits Syst.* 2005; 1: 1594-1599.
- [19] Hayati, M., and Shirvany, Y. Artificial neural network approach for short term load of forecasting for Illam region, *Int. J. Electr., Comput. Syst. Eng.* 2007; 1: 121-125.
- [20] Hayati, M., Karami, B., and Abbasi, M. Numerical simulation of fuzzy nonlinear equation by feedforward neural networks, *World Appl. Sci. J.* 2007; 2: 229-234.
- [21] Rao, M., and Ravindran, K. An evolutionary approach for solving differential equations, *WSEAS Trans. Circuits*, 2002: 1, 100-105.
- [22] Perkins, R., and Brabazon, A. Predicting credit ratings with a GA-MLP hybrid, *Artif. Neural Networks in Real-Life Appl.*, 2006; 220-237. PMID:16785794
- [23] Sethuraman, J. Soft computing approach for bond rating prediction, *Artificial Neural Networks in Real-Life Appl.*, 2006; 202-219.
- [24] Mantzaris, D., Anastassopoulos, G., and Adamopoulos, A. Genetic algorithm pruning of probabilistic neural networks in medical disease estimation, *Neural Networks*, 2011; 24: 831-835. PMID:21723704 <http://dx.doi.org/10.1016/j.neunet.2011.06.003>
- [25] Mantzaris, D., Anastassopoulos, G., and et al. A probabilistic neural network for assessment of the vesicoureteral reflux's diagnostic factors validity, in: Diamantaras, K., Duch, W., Iliadis L. (Eds.), *ICANN 2010, Part I, LNCS 6352*, Springer-Verlag Berlin Heidelberg. 2010; 241-250.
- [26] Askariazad, M., and Wanous, M. Artificial neural network approach for modelling overall customer satisfaction, *Artif. Intell. Appl. (AIA)*, Innsbruck, Austria. 2009.
- [27] Dayhoff, J., and DeLeo, J. Artificial neural networks opening the black box, *Cancer Suppl.* 2001; 91: 1615-1635.
- [28] Riziotis, C., and Vasilakos, A. Computational intelligence in photonics technology and optical networks: a survey and future perspectives, *Inf. Sci.* 2007; 177: 5292-5315. <http://dx.doi.org/10.1016/j.ins.2007.06.012>
- [29] Mantzaris, D., Anastassopoulos, G., and et al. A non-symbolic implementation of abdominal pain estimation in childhood, *Inf. Sci.* 2008; 178: 3860-3866. <http://dx.doi.org/10.1016/j.ins.2008.06.015>
- [30] Georgopoulos, E., Likothanassis, S., and Adamopoulos, A. Evolving artificial neural networks using genetic algorithms, *Neural Network World.* 2000; 4: 565-574.
- [31] Mantzaris, D., Anastassopoulos, G., and et al. Intelligent prediction of vesicoureteral reflux disease, *WSEAS Trans. Syst.* 2005; 4: 1440-1449.
- [32] Hung, N., Babel, and et al. N. An artificial neural network model for rainfall forecasting in Bangkok, Thailand, *Hydrol. Earth Syst. Sci.* 2009; 13: 1413-1425. <http://dx.doi.org/10.5194/hess-13-1413-2009>
- [33] Young II, W., Millie, D., and et al. Modeling net ecosystem metabolism with an artificial neural network and Bayesian belief network, *Environ. Modell. Software.* 2011; 26: 1199-1210. <http://dx.doi.org/10.1016/j.envsoft.2011.04.004>
- [34] Lennernäs, B., Sandberg, D., and et al. The effectiveness of artificial neural networks in evaluating treatment plans for patients requiring external beam radiotherapy, *Oncol. Rep.* 2004; 12: 1065-1070. PMID:15492794
- [35] Astion, M., Wener, M., and et al. Overtraining in neural networks that interpret clinical data, *Clin. Chem.* 1993; 39: 1998-2004. PMID:8375090
- [36] Tripathy, A., Mohapatra, S., and et al. Weather forecasting using ANN and PSO, *Int. J. Sci. Eng. Res.* 2011; 2: 1-5.
- [37] Baboo, S., and Shereef, I. An efficient weather forecasting system using artificial neural network, *Int. J. Environ. Sci. Dev.*, 2010; 1: 321-326.
- [38] Anderson, D., and McNeil, G. *Artificial Neural Networks Technology*, Kaman Sci. Corp. 1992.
- [39] Howard, D., and Mark, B. *Neural network toolbox for use with MATLAB*, Math Works Inc. 2012.
- [40] Nayak, R., Jain, L., and Ting, B. Artificial neural networks in biomedical engineering: a review, *Proc. 1st Asian-Pacific Congr. Comput. Mech.* 2001; 887-892.
- [41] Vourkas, M., Papadourakis, G., and et al. Effects of segmentation on the discrimination of three mental stages using ANN and different EEG signal representations, *Proc. 4th Int. Con. Neural Networks and Expert Syst. Med. Healthcare.* 2001; 65-68.
- [42] Papik, K., Molnar, B., and et al. Application of neural networks in medicine - a review, *Med. Sci. Monit.*, 1998; 538-546.
- [43] Johari, D., Khawa, T., and et al. Artificial neural network based technique for lightning prediction, 5th Stud. Conf. Res. Dev., Malaysia. 2007.
- [44] Samek, D., and Manas, D. Artificial neural networks in artificial time series prediction benchmark, *Int. J. Math. Models Methods in Appl. Sci.* 2011; 5: 1085-1093.

- [45] 45 Radhika, Y., and Shashi, M. Atmospheric temperature prediction using support vector machines, *Int. J. Comput. Theor. Eng.* 2009; 1: 1793-8201.
- [46] 46 Cury, D., and Segatto, E. Pattern recognition to distinguish magnetizing inrush from internal faults in power transformers, *WSEAS Trans. Syst.* 2004; 3: 1258-1264.
- [47] 47 Tasoulis, D., Spyridonos, P., Pavlidis, N., Cavouras, D., Ravazoula, P., Nikiforidis, G., and Vrahatis, M. Urinary Bladder tumor grade diagnosis using on-line trained neural networks, *Proc. Knowl. Based Intell. Inform. Eng. Syst.*, Springer-Verlag. 2003; 199-206.
- [48] Weather Underground [Internet]. 2012. Available from: <http://www.wunderground.com/history/airport/LGAL/1996/7/1> (access 23/04/2012).
- [49] Wongseeree, W., Chaiyaratana, N., and et al. Thalassaemia classification by neural networks and genetic programming, *Inf. Sci.* 2007; 177: 771-786. <http://dx.doi.org/10.1016/j.ins.2006.07.009>
- [50] Maqsood, I., Khan, M., and Abraham, A. An ensemble of neural networks for weather forecasting, *Neural Comput. and Appl.* 2004; 13: 112-122. <http://dx.doi.org/10.1007/s00521-004-0413-4>
- [51] Khan, M., and Ondrusek, C. Short-term Electric demand prognosis using artificial neural networks, *Electr. Eng.* 2000; 51: 296-300.
- [52] Anad, S., Dash, A., and et al. Prediction and classification of thunderstorms using artificial neural networks, *Int. J. Eng. Sci. Technol.* 2011; 3: 4031-4035.
- [53] Hayati, M., and Mohebi, Z. Application of artificial neural networks for temperature forecasting, *World Acad. Sci., Eng., Technol.* 2007; 28: 275-279.
- [54] 54 Narayan, S., and Tagliarini, G. An analysis of underfitting in MLP networks, *IEEE Neural Networks.* 2005; 2: 984-988.