

# Automatic Analysis of Radio Meteor Events Using Neural Networks

Victor Ștefan Roman<sup>1</sup> · Cătălin Buiu<sup>1</sup>

Received: 10 October 2014 / Accepted: 14 July 2015 / Published online: 18 July 2015  
© Springer Science+Business Media Dordrecht 2015

**Abstract** Meteor Scanning Algorithms (MESCAL) is a software application for automatic meteor detection from radio recordings, which uses self-organizing maps and feedforward multi-layered perceptrons. This paper aims to present the theoretical concepts behind this application and the main features of MESCAL, showcasing how radio recordings are handled, prepared for analysis, and used to train the aforementioned neural networks. The neural networks trained using MESCAL allow for valuable detection results, such as high correct detection rates and low false-positive rates, and at the same time offer new possibilities for improving the results.

**Keywords** Automatic meteor detection · Self-organizing map · Multi-layered perceptron

## 1 Introduction

On its trip around the Sun, the Earth encounters important quantities of cosmic dust, debris or small rocks, which are small extraterrestrial objects called meteoroids. These objects interact with the Earth's atmosphere, either burning up through ablation, or falling on the planet's surface. It is estimated that between 4.4 and 7.4 tons (Mathews et al. 2001) of space material enters Earth's atmosphere every day, and while most of it burns up, some of the material may survive the interaction and fall on our planet. Meteoroids have important research value because they hold information about our Solar System, how it came to be,

---

**Electronic supplementary material** The online version of this article (doi:10.1007/s11038-015-9473-y) contains supplementary material, which is available to authorized users.

---

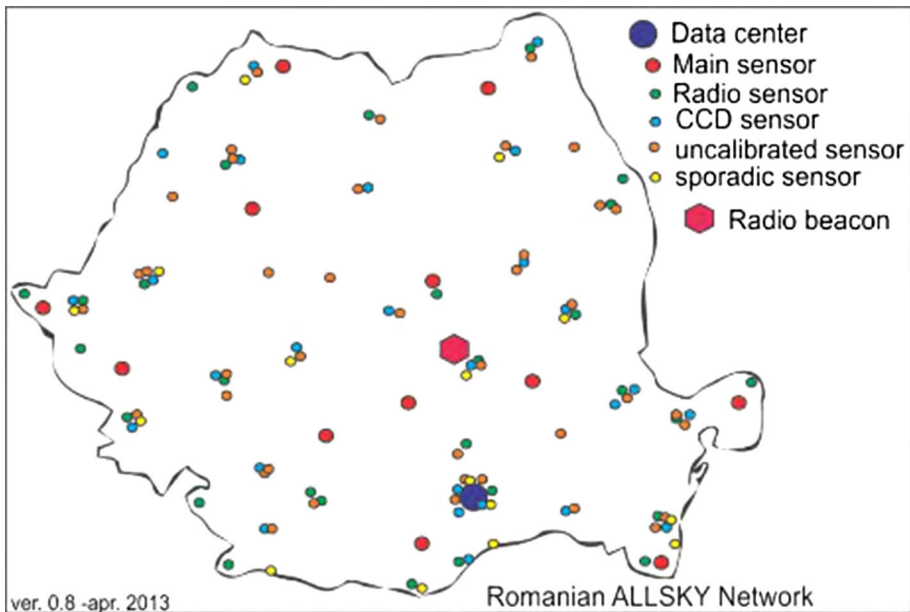
✉ Victor Ștefan Roman  
victor.roman@acse.pub.ro

<sup>1</sup> Department of Automatic Control and Systems Engineering, Politehnica University of Bucharest, Bucharest, Romania

and how it evolved to the state we know today. They are also regarded as samples from the early days of the Solar System.

With the development of radio/video technology and its ease of use in monitoring the Earth's atmosphere, there are more and more studies done on our planet's daily interactions with meteoroids, or on the meteoroids themselves. One of the important topics of these studies is the automatic detection of meteors. The most common meteor detection solutions implemented today involve the use of either a radio detection approach, with examples found in Hocking et al. (2001), Wen et al. (2005) and in Weryk and Brown (2012), or the use of a video detection approach, examples of which can be found in Gural (2007), Trigo-Rodriguez et al. (2008), Jenniskens et al. (2011) or Weryk et al. (2013). These meteor detection solutions all present a recurring theme: they all employ signal processing techniques and utilize a threshold function to decide whether a particular event is an occurrence of a meteor or not.

The ALLSKY Project (AP) (Georgescu and Georgescu 2013), a work in progress, is an attempt to build the first automatic meteor detection system in Romania. It involves deploying several meteor detection stations in Romania and connecting them in a network operated from a central data station, as can be seen in Fig. 1. The detection stations will allow for a dual type of meteor surveillance. The first means of scanning for meteors is through the use of radio receivers. To scan the radio waves for meteor echoes, the detection stations in the AP network will be deployed as radio receiver stations and they will work using the forward scattering of radio waves. There will be two kinds of radio signal sources that the AP network will listen for: firstly, there will be a series of local emitting stations, installed in different parts of Romania, and secondly, there will be more distant radio emitting stations, such as TV antennas from Russia, Greece or other close countries. The



**Fig. 1** The proposed ALLSKY meteor detection network (Georgescu and Georgescu 2013). It presents the meteor detection system's coverage of Romania's territory

principle behind the forward scattering of radio waves is that when a meteor enters the Earth's atmosphere, it will begin to leave a trail of ionized particles behind it. This trail, due to ionization, will reflect the radio waves emitted in the vicinity of the event. Depending on the distance between the emitting station, the meteor trail, and between the meteor trail and the receiver station, the reflected radio waves may be received by some stations in the ALLSKY project. The radio stations will operate 24 h a day, recording all received radio signals and sending them to the central data station, where they will be analyzed for meteor echoes.

The second means through which the AP will scan for meteors is through the use of dome-like, allsky CCD cameras, which will visually survey the night sky. The monitoring stations that will have CCD cameras installed will be deployed only in remote places, where light pollution levels are low, and the night skies are as cloudless as possible. The video monitoring stations with CCD cameras will scan the sky, and the recorded data will be saved and sent to the central data station.

In the present paper, a novel automatic meteor detection approach, that uses artificial neural networks (ANNs), is described. ANNs are implicit mathematical models that emulate the structure, functions or characteristics of biological neural networks in order to solve modeling or classification problems. Typically, an ANN is a collection of interconnected artificial neurons organized in one or more layers. In order to be able to solve a given problem, ANNs are required to undergo a training process, during which the ANNs build an internal model of the sought after objects. In the application presented here, ANNs are trained to differentiate between meteor and non-meteor samples and be able to compute new, previously unknown samples correctly. The ANNs were chosen for this study for their high processing speed, their ability to learn new, complex patterns and for their parallel processing capabilities.

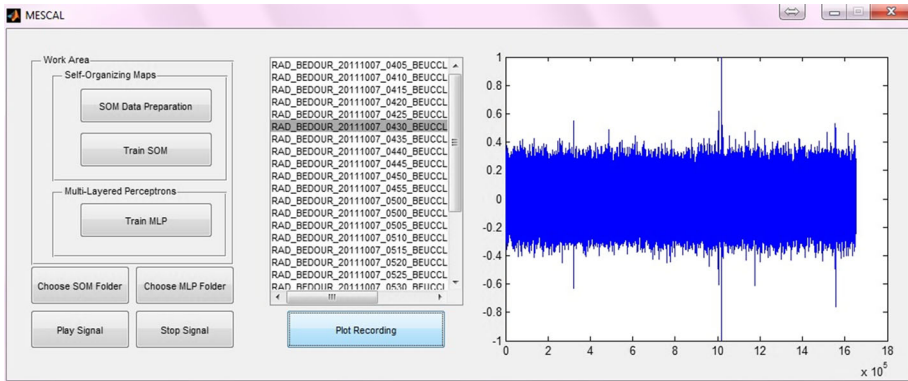
## 2 The MESCAL Concept

To tackle the problem of automatic meteor detection in radio recordings, a software application was developed to allow the training and testing of solutions which can be applied to the detection of meteors. Meteor Scanning Algorithms (MESCAL) is the software application built to address this problem. MESCAL's role is to allow the user to access and visualize radio data and to use this data to train meteor detection algorithms, or to test previously developed solutions. Figure 2 shows MESCAL's main interface which is used to access all of MESCAL's functions. More details on MESCAL's mode of operation and functions can be found in the ancillary material.

In order to test MESCAL's abilities in meteor detection, it was decided to use radio recordings of meteor data stored as sound (.wav) files and to train and test two types of ANNs to search for the meteor echoes in the input data. The two neural networks presented in this study are the self-organizing map (SOM) and the multilayer perceptron.

### 2.1 Data Preparation

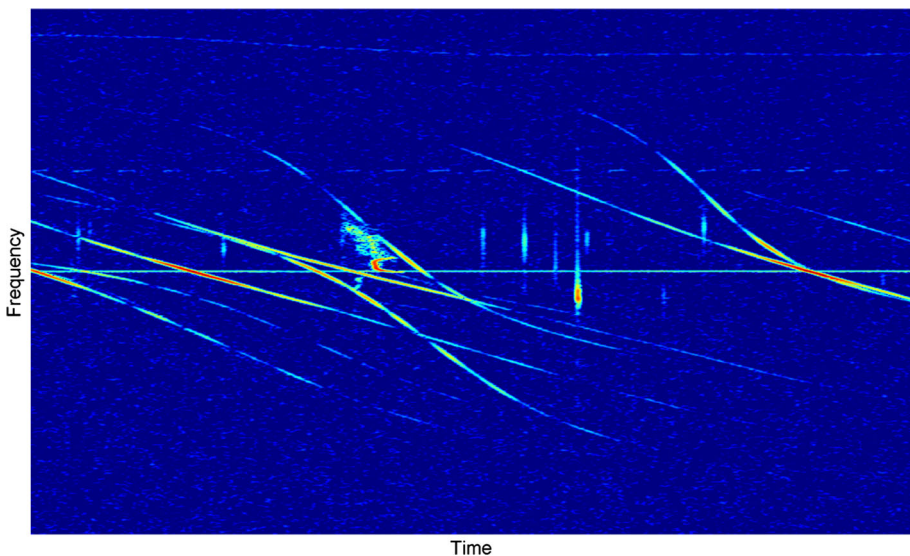
The radio recordings used for this study were obtained from BRAMS, the Belgian radio detection system. The BRAMS network consists of one transmitting beacon and 26 radio receiving stations spread around Belgium, and utilizes the forward scattering of radio waves to record meteors. More details on the BRAMS network can be found in Calders and



**Fig. 2** MESCAL's user interface, which allows for the visualization of the radio recordings, the sampling of the available data and for the training and testing of the two types of ANNs employed by MESCAL

Lamy (2012) and in Lamy et al. (2013). The radio stations in the BRAMS network record data continuously, saving it every 5 min as .wav audio files. Figure 3 depicts the spectrogram representation of such a recording.

Considering that the radio files used in this study have been recorded over a bandwidth of 2.5 kHz and are stored as 5 min long recordings, it became clear that before the available data can be used to train any ANN, a process of filtering and sampling of the radio recordings had to be undertaken.



**Fig. 3** An example of a spectrogram representation of a radio recording from BRAMS. The duration of the recording is 5 min, the frequency range is 200 Hz centered around the beacon's frequency, while the power of the signal is color coded. The horizontal line in the middle of the picture is the beacon's directly received signal. The inverted S-shapes and diagonal lines are radio reflections on airplanes. The vertical lines are underdense meteors, while the complex shape close to the middle of the picture is an example of an overdense meteor. (Color figure online)

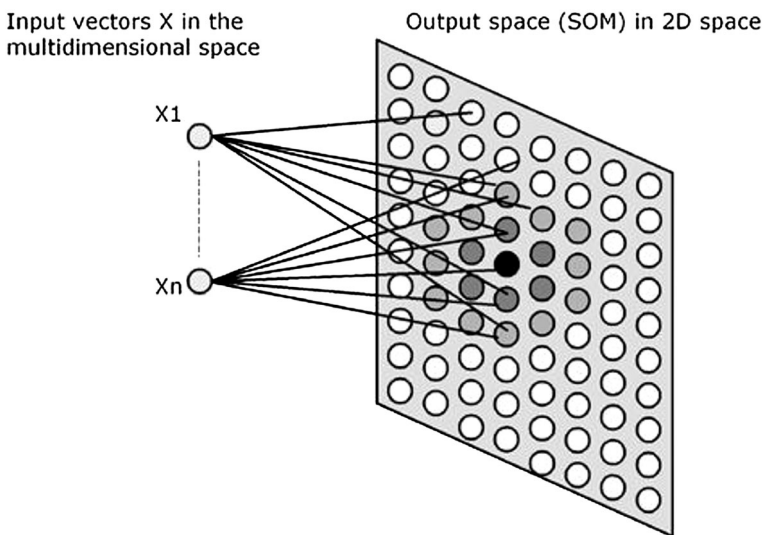
The filtering of the radio recordings was done because experience with the BRAMS recordings has shown that the vast majority of meteors are recorded in the vicinity of the network's beacon signal. Therefore, the recordings used for this study utilize only 200 Hz of the full bandwidth, and are centered around the beacon's signal.

The sampling of the BRAMS recordings was completed taking two aspects into account. Firstly, the majority of meteors detected and recorded are underdense meteors, most of which have been discovered, through manual inspection of the radio recordings, to span for a very short duration, typically somewhere around 0.1 s. Secondly, both types of ANNs used in this study would benefit from being trained with clear examples of meteor echoes. Therefore, it was decided that the recordings should be sampled using a 0.1 s long sampling window, which has a 0.05 s slide. This particular sampling window was chosen because a smaller window might not have extracted enough information and would not have captured the shape of the underdense meteor echoes, while a larger window size might have included other signal contributions that may have detracted from the contribution of the meteors in the recordings.

## 2.2 Self-organizing Maps

The SOM, also known as the Kohonen neural network (Kohonen 1990), is the best known example of ANN used for the analysis and clustering of high-dimensional data. The SOM, depicted in Fig. 4, is a neural network based on competitive learning that clusters the input data onto a topographic map of interconnected neurons, in which similar input patterns are clustered in the same area of the map. The particularity of the SOM neural network is that it uses an unsupervised training algorithm to produce the final map, an algorithm which is based solely on the similarity between the input samples and the neurons in the SOM.

Training a SOM neural network is the process through which a random initial distribution of neurons is transformed, over many iterations (called epochs), into a map of areas



**Fig. 4** A typical graphic representation of a SOM neural network, where a high-dimensional input vector  $x$  is mapped onto a two-dimensional lattice of neurons

of similarity, with each area containing similar input patterns. This type of mapping, alongside the competitive training algorithm, ensures that the SOM preserves the topological relationships of the input space.

The training algorithm used to build a SOM occurs for a large number of epochs, with each epoch containing a series of steps:

1. A random vector is chosen from the training dataset and is presented to the SOM.
2. The training algorithm searches the neuron that best resembles the chosen input. The most alike neuron will be deemed the Best Matching Unit (BMU).
3. The BMU's topological neighborhood is determined. This neighborhood starts large, but diminishes every epoch. All the neurons that are closer to the BMU than the neighborhood's radius are considered to be part of the neighborhood.
4. The BMU and its neighbors' weights are adjusted to resemble the input sample more.
5. Repeat steps 1–4 for all the samples in the input dataset.

Determining the BMU in the map of neurons for a given input means finding the neuron whose weights are closest to the input sample. This is done by calculating the distance between the input sample and the neuron's weights. The distance measure used in this study is the popular Euclidean distance, which is given by:

$$d(x, w) = \sqrt{\sum_i (x_i - w_i)^2}, \quad (1)$$

where  $x$  is the input vector, while  $w$  is the weights vector of the analyzed neuron.

After having determined the BMU, its neighborhood has to be calculated. A typical choice for the topological neighborhood around the BMU is the Gaussian function (Haykin 1998):

$$h_{j,i}(t) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(t)}\right), \quad (2)$$

where  $d_{j,i}$  is the distance between the BMU  $i$  and neighboring neuron  $j$ , while  $\sigma(t)$  is the radius of the neighborhood centered on the BMU at time  $t$ . One of the unique features of the SOM is that its training algorithm evolves over time, through the constant shrinking of two parameters. The first one is the radius of the BMU's neighborhood,  $\sigma$ , which is modified every epoch following this equation:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \quad (3)$$

where  $\sigma_0$  is the initial neighborhood radius,  $\tau_1$  is a time constant and  $t$  is the current epoch.

After calculating which neurons are part of the BMU's neighborhood, their weights will be adjusted to resemble the input sample better, which is done with the following equation:

$$w_j(t+1) = w_j(t) + \eta(t)h_{j,i}(t)(x - w_j(t)), \quad (4)$$

where  $w_j(t)$  is the neuron  $j$ 's current weight vector,  $x$  is the input sample,  $h_{j,i}(t)$  is the neighborhood function and  $\eta(t)$  is the training algorithm's learning rate. This last parameter is the other parameter in the SOM's training algorithm that shrinks with time, following this equation:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right), \quad (5)$$

where  $\eta_0$  is the initial learning rate,  $t$  is the current epoch and  $\tau_2$  is another time constant of the SOM algorithm.

It should be noted that the weights adjusting formula in Eq. 4 is dependent not only on the learning rate  $\eta$ , but also on the distance between the BMU and the neuron  $j$ , which is given through the neighborhood function  $h_{j,i}(t)$ . This means that the farther neuron  $j$  is from the BMU, the less it will train using the current input vector.

Once the training of a SOM is finished, the resulting topographic map will contain areas of similarity that have been formed through adaptation of the neurons in those areas to some input samples in the training dataset. Therefore, if an area inside the SOM can be delimited from other areas, it will mean that all samples that are mapped in that area will be members of a particular class of inputs. This area separation is what will allow the SOM to later classify inputs into the classes that were defined during the training of the neural network.

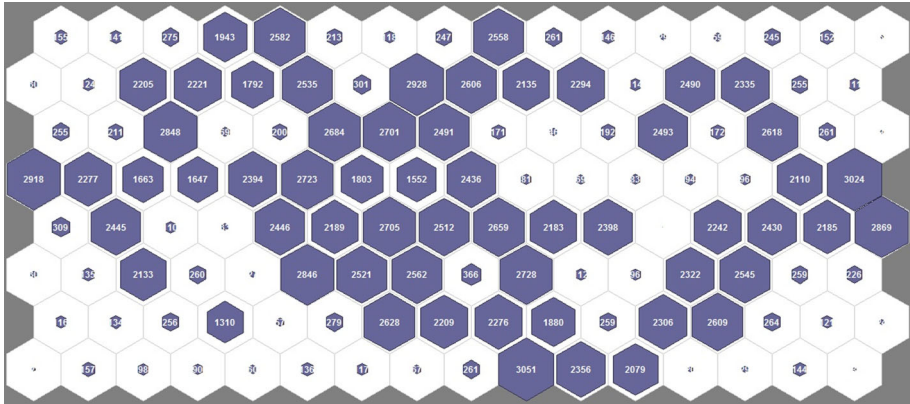
SOM neural networks train using an unsupervised algorithm that requires no previous information of the training samples to be available before or during the training. Therefore, the training dataset used to train the SOMs in this study was built after sampling 25 radio recordings, which resulted in a dataset containing around 150,000 samples. To be able to assess the quality of the SOMs trained in this study, the training dataset has been manually inspected and all meteor samples have been identified. This was required in order to discover the areas on the map where meteors are clustered later.

The SOMs used in this study were trained with two goals in mind: to offer results in correctly detecting radio samples and to minimize the values of two SOM quality metrics, the quantization error and the topographic error (Uriarte and Martin 2008). These two metrics assess a SOM's quality in relationship to its training dataset. More specifically, they determine if a SOM preserves the topographic properties existing in the input dataset. Practice has shown that obtaining good values for these two metrics, values close to or equal to 0, will lead to the training of a SOM that has good performances in meteor detection too.

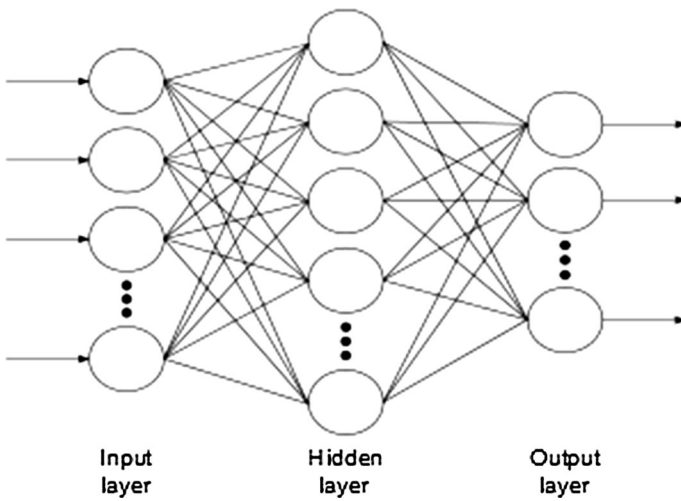
Using the training dataset mentioned above, several SOMs were trained in this study. By monitoring the values of the quantization and topographic errors, as well as checking the meteor detection capabilities of the trained networks, the best performing SOM found is presented in Fig. 5. That particular SOM was built with 128 neurons in the output lattice, trained for a number of 1000 epochs and generated the lowest values for the two quality metrics.

### 2.3 Multi-layer Perceptrons

Used very often in pattern recognition problems, the multi-layer perceptron (MLP) is an ANN that stacks neurons in one or more layers. Generally, a MLP consists of one input layer, one output layer and one or more processing layers of neurons, also known as hidden layers. Neurons in one layer are fully connected to all the neurons in the surrounding layers. For this study, a simple MLP with three layers and feedforward architecture was chosen, as depicted in Fig. 6. Using the feedforward architecture, the information in the MLP neural network will only flow from the input layer towards the output layer, where it will produce a specific output, generated through simple calculations: each neuron in a MLP neural network calculates the weighted sum of all inputs received and passes the result to a nonlinear activation function that determines the output of that particular neuron. Thus, the quality of the signal generated by a MLP's output layer will greatly depend on a well trained set of neuron weights.



**Fig. 5** The hits plot of the best performing SOM. Each *hexagon* in the figure represents one of the SOM's neurons, while the *numbers* inside represent the number of samples mapped on that particular neuron. The neurons with *dark background* and high number of hits are the neurons where most of the non-meteor samples are mapped, while the neurons with *light background* and smaller number of hits are where most meteor samples are mapped



**Fig. 6** Graphic representation of a three layers MLP with feedforward architecture and fully connected neurons in all layers. Being a feedforward MLP, the information in such a neural network flows from the input layer to the output layer (from *left to right*)

Training a MLP involves using of a supervised training algorithm that will adjust the weights of all the neurons in the network in order to be able to recognize particular patterns in the training dataset. The training algorithm is called “supervised” because it requires a user to feed the network with input vectors and give the network a desired output for each input received, thus guiding the neural network to learn specific patterns. The best known MLP training algorithm, also used in this paper, is the Backpropagation algorithm (Rumelhart et al. 1986). This algorithm functions using an error measure, the difference



between the output signal and the desired output for a particular input vector. The training process of a MLP network takes place over several epochs, each epoch being formed by a set number of steps:

1. A random vector is chosen from the input dataset and is fed to the input layer of the MLP.
2. The input vector passes through all the layers of the neural network and generates an output signal.
3. The resulting output signal is compared to the desired output and an error is calculated, which is propagated backwards through the network.
4. The weight vectors of the neurons in the output and hidden layers are adjusted based on the error measure.
5. Steps 1–4 are repeated for all the input samples in the training dataset.

The error measure needed to adjust the weights of the neurons in the MLP is obtained through:

$$e_j(t) = d_j(t) - y_j(t), \quad (6)$$

where  $d_j(t)$  is the desired output, while  $y_j(t)$  is the actual output of neuron  $j$  in the output layer, at time  $t$ .

The weight correction equation used by the Backpropagation algorithm, which is defined by the delta rule, can be written in a general form as (Haykin 1998):

$$\Delta w_{ji}(t) = \eta \delta_j(t) y_i(t), \quad (7)$$

where  $\Delta w_{ji}(t)$  is the weight correction for the synapse (i.e. link) between neurons  $i$  and  $j$ ,  $\eta$  is the learning rate parameter,  $\delta_j(t)$  is the local gradient, which describes the changes needed for the weights of neuron  $j$ , while  $y_i(t)$  is the output signal of neuron  $i$  and input received by neuron  $j$ . In this previous equation, the parameter  $\delta_j(t)$  is dependent on the layer in which neuron  $j$  is situated. Thus, if the neuron  $j$  is part of the output layer, then  $\delta_j(t)$  will be deduced from:

$$\delta_j(t) = e_j(t) \varphi'_j(v_j(t)), \quad (8)$$

where  $e_j(t)$  is the error signal for neuron  $j$  and  $\varphi'_j(v_j(t))$  is the derivative of the neuron's activation function.

On the other hand, if neuron  $j$  is part of a hidden layer within the MLP, then  $\delta_j(t)$  will be calculated following:

$$\delta_j(t) = \varphi'_j(v_j(t)) \sum_k \delta_k(t) w_{kj}(t) \quad (9)$$

This last equation is a bit more complicated, where  $\delta_k(t)$  requires knowledge of the error signals in the layer that comes after the layer of neuron  $j$ , while  $w_{kj}(t)$  is the collection of the synaptic weights that connect neuron  $j$  with all neurons in the following layer.

After the supervised training process is finished, a MLP neural network will be able to classify received inputs based on the internal models built during training. Therefore, it is very important that the a priori class membership identification is thoroughly done by the user. The MLP network needs to be taught as many of the possible input patterns in order to be able to correctly identify previously unused input samples.

Due to the supervised nature of an MLP's training algorithm, the training dataset used to train the MLPs in this study had to be manually built. The final dataset comprised of 472 meteor samples and 386 non-meteor samples, along with the necessary expected outputs vector. The two numbers of meteor and non-meteor samples have thus been chosen because experience has shown that there should be a balance between the two for the resulting neural network to be able to correctly classify both types of inputs.

Searching for a MLP that offers good results in meteor detection means finding the appropriate number of neurons to be used in the MLP's hidden layer. While the number of neurons in the input layer was fixed to 551 neurons (the radio recordings use a 5512 Hz sampling rate, therefore 0.1 s long samples translate to vectors with 551 elements), while that of the output layer was chosen to be 2 (only two classes have been defined for this study, meteor and non-meteor), the number of neurons in the hidden layer was continuously varied, until a good performing MLP was found. In order to assess the quality of the MLPs trained during this study, the mean square error (MSE) measure was used. The MSE calculates the sum of differences between the MLP's output signals and their expected outputs. Therefore, the lower the MSE is, the better the MLP is at correctly classifying the input samples. For this reason, the MSE of the neural networks trained in this study was monitored while searching for a well performing MLP. Through testing, it was discovered that the best performing MLP trained for this study contained 250 neurons in the hidden layer and was trained for a number of 1000 epochs.

### 3 Results and Discussion

The two neural networks previously proposed for handling the automatic detection of meteors were tested using a newly built data set in order to assess their ability to correctly classify meteor and non-meteor samples. A total of 144 previously unused BRAMS radio recordings (12 h of data) were sampled to build this test set. The data was then manually inspected before being presented to the ANNs to identify the meteor samples within. Through this procedure, 1496 samples were found to contain meteor echoes, while the other 858059 samples were labeled as non-meteor data.

Table 1 presents the detection results obtained through testing by the best performing SOM with the test dataset. These results, which were manually validated, show that the proposed SOM solution has the ability to recognize both meteor and non-meteor samples in the given test dataset. With a combined correct detection rate of over 89 %, the SOM is able to differentiate between the two types of samples, which shows that the self-organized map built during training contains areas of similarity that mostly contain either meteor or non-meteor samples, which allows the SOM to discriminate between the two types of data. But while the correct detection rate is high, it should be noted that the percentage of falsely classified non-meteor samples is quite important too.

**Table 1** The test results obtained using the best SOM

| Meteor samples (1496) |                | Non-meteor samples (858,059) |               |
|-----------------------|----------------|------------------------------|---------------|
| True positive         | False positive | False negative               | True negative |
| 1373                  | 123            | 97113                        | 760946        |
| 91.78 %               | 8.22 %         | 11.32 %                      | 88.68 %       |

**Table 2** The test results obtained using the best MLP

| Meteor samples (1496) |                | Non-meteor samples (858,059) |               |
|-----------------------|----------------|------------------------------|---------------|
| True positive         | False positive | False negative               | True negative |
| 1286                  | 210            | 116718                       | 741341        |
| 85.96 %               | 14.04 %        | 13.6 %                       | 86.4 %        |

There are several factors that influence the SOM's ability to detect meteors. Some are internal factors, like the size of the map, where more neurons may allow the SOM to better classify the existing patterns in the training data. The duration of training (i.e. the number of epochs) can also influence a SOM's abilities, through either undertraining, or through overtraining the neural network. There are also external factors, the most important of which being the data in the training set and how samples were extracted. It is very important that a user opts for a good sampling of the input data, because the SOM neural network has to be taught as many examples of class members as possible in order to later provide a high rate of correctly identified test samples. Therefore, a good sampling of the raw data, which allows for the extraction of sufficient examples of both classes used in this study, may improve the overall performance of the SOM.

To test the MLP's ability to detect meteors, the best performing ANN was tested with the data set described above and the results were manually analyzed in order to validate them. The results, presented in Table 2, show that the proposed MLP is also a good solution to detecting the meteors in the radio recordings used in this study. The total correct detection rate of the proposed SOM is equal to 86.4 %. And just like the SOM neural network tested before, the MLP can also correctly classify most samples in the test dataset, but displays the same problem of incorrectly detecting an important number of non-meteor samples.

The factors that influence the capabilities of a MLP to correctly classify the received inputs are similar to those that affect a SOM. Therefore, the size of the hidden layer, number of epochs of training and input data all contribute to the final results. But compared to the SOM case, the MLP neural networks are much more dependent on the training set. While testing for this study, it became obvious that the dataset used to train the MLPs has to include as many examples of both classes as possible, but at the same time it has to contain a balanced number of both meteor and non-meteor samples. Experience has shown that an unbalanced training set will lead to a MLP that will be better at recognizing either meteor or non-meteor samples, depending on which of the two classes had the majority of samples in the training set. Compared to this situation, the SOM approach has proven to be less dependent on the contents of the training set, but this must be a result of the different training approaches used by the two types of neural networks.

## 4 Conclusions

A software tool that uses two novel automatic meteor detection approaches, based on ANNs, was presented in this paper. Using MESCAL, a SOM or a multi-layered perceptron can be trained with radio data, and used to detect meteor echoes in radio recordings. The two ANNs work differently and are trained using distinct training algorithms. The SOMs are trained in an unsupervised manner, using a more general, automatically built training set. Meanwhile, the MLPs have to be trained in a supervised manner, where the training

inputs have to be handpicked to allow the neural network to build internal models of the different types of input signals. Both types of ANNs were trained with data extracted from radio recordings originating from the BRAMS radio network and both ANNs were tested with previously unused radio data.

The neural networks trained with the proposed strategies have provided very promising results in the field of automatic meteor detection. Both the SOM and the MLP presented in this paper have a high correct meteor detection rate, of more than 85 % of the included meteor samples. This proves that although different, both techniques are viable solutions for detecting meteors in radio recordings. At the same time though, both proposed ANNs have the same drawback: they both falsely classify an important number of non-meteor samples. Although the percentage of misclassified non-meteor samples is not high, the actual number of non-meteors that are classified as being meteors is significantly higher than the number of correctly detected meteors.

In their current form, the two proposed meteor detection approaches offer encouraging results, but also leave room for improvements. The primary test directions for improving the proposed approaches are clear. One would be to alter the size of the ANNs or that of the duration of their training, modifications that may allow the SOMs to develop maps which represent the patterns in the input data clearly, or allow the MLPs to build better internal models of the input samples. Another option would be to change the sampling window size and that of the window slide, and see if a different sampling approach will improve the ANNs capabilities to correctly detect the two types of input samples. Lastly, the two ANNs may benefit from larger training datasets, which could cover more examples of the input patterns the two ANNs have to learn. Also, it should be noted that the MESCAL application and the two meteor detection approaches concentrate on the correct identification of meteor echoes in a given dataset, but give no information on the meteor samples that they detect. Therefore, MESCAL could also be improved by including a tool that displays the exact time of the meteor samples that it detects.

**Acknowledgments** This work was supported by a grant from the Romanian National Authority for Scientific Research, CNDI-UEFISCDI, project number 205/2012. The work of V.Ş. Roman is supported by the Sectoral Operational Programme Human Resources Development (SOP-HRD), financed from the European Social Fund and the Romanian Government, under the contract number POSDRU/159/1.5/S/137390. V. Ş. Roman performed part of this work during a research stage at L’Institute d’Aéronomie Spatiale de Belgique and would like to thank Hervé Lamy for his kind support.

## References

- S. Calders, H. Lamy, Brams: status of the network and preliminary results, in *Proceedings of the International Meteor Conference (IMC 2011)* (2012), pp. 73–76
- A. Georgescu, T. Georgescu, Romanian AllSky Network: basic deployment, in *Proceedings of the 32nd International Meteor Conference (IMC 2013)* (2013). 22–25 Aug 2013
- P.S. Gural, Algorithms and software for meteor detection. *Earth Moon Planets* **102**, 269–275 (2007)
- S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edn. (Prentice Hall PTR, Upper Saddle River, NJ, 1998)
- W.K. Hocking, B. Fuller, B. Vandepuer, Real-time determination of meteor-related parameters utilizing modern digital technology. *J. Atmos. Sol. Terr. Phys.* **63**(2–3), 155–169 (2001)
- P. Jenniskens, P.S. Gural, L. Dynneson, B.J. Grigsby, K.E. Newman, M. Borden, M. Koop, D. Holman, CAMS: Cameras for Allsky Meteor Surveillance to establish minor meteor showers. *Icarus* **216**(1), 40–61 (2011)

- T. Kohonen, The self-organizing map. *Proc. IEEE* **78**(9), 1464–1480 (1990)
- H. Lamy, E. Gamby, S. Ranvier, Y. Geunes, S. Calders, J. de Kaiser, The BRAMS viewer: an on-line tool to access the BRAMS data, in *Proceedings of the International Meteor Conference (IMC 2012)* (2013). pp. 48–50
- J.D. Mathews, D. Janches, D.D. Meisel, Q.-H. Zhou, The micrometeoroid mass flux into the upper atmosphere: Arecibo results and a comparison with prior estimates. *Geophys. Res. Lett.* **28**(10), 1929–1932 (2001)
- D. Rumelhart, J. McClelland, PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol I, Foundations (MIT Press, Cambridge, MA, 1986)
- J.M. Trigo-Rodríguez, J.M. Madiedo, P.S. Gural, A.J. Castro-Tirado, J. Llorca, J. Fabregat, S. Viték, P. Pujols, Determination of meteoroid orbits and spatial fluxes by using high-resolution All-Sky CCD cameras. *Earth Moon Planets* **102**(1–4), 231–240 (2008)
- E. Uriarte, F. Martín, Topology preservation in SOM, in *World Academy of Science, Engineering and Technology, International Science Index 21*, vol 2, no. 9 (2008), pp. 867–870
- C.-H. Wen, J.F. Doherty, J.D. Mathews, D. Janches, Meteor detection and non-periodic bursty interference removal for Arecibo data. *J. Atmos. Sol. Terr. Phys.* **67**(3), 275–281 (2005)
- R.J. Weryk, P.G. Brown, Simultaneous radar and video meteors—I: metric comparisons. *Planet Space Sci.* **62**(1), 132–152 (2012)
- R.J. Weryk, M.D. Campbell-Brown, P.A. Wiegert, P.G. Brown, Z. Krzeminski, R. Musci, The Canadian automated meteor observatory (CAMO): system overview. *Icarus* **225**(1), 614–622 (2013)