World Journal of Agricultural Sciences 4 (1): 114-120, 2008
ISSN 1817-3047
© IDOSI Publications, 2008

Artificial Neural Network Selection for the Detection of Plant Viruses

[1]D. Frossyniotis, [1]Y. Anthopoulos, [2]S. Kintzios, [2]G. Moschopoulou and [1]C. P. Yialouris

[1]Laboratory of Informatics, Department of Science,
Agricultural University of Athens, Iera Odos 75, 118 55, Athens, Greece
[2]Laboratory of Plant Physiology and Morphology, Department of Agricultural Biotechnology,
Agricultural University of Athens, Iera Odos 75, 118 55, Athens, Greece

Abstract: The aim of this work is the development of an intelligent system for the detection of plant viruses, using biosensors and Artificial Neural Networks. The system is based on the Bioelectric Recognition Assay (BERA) method for the detection of viruses, developed by our team. BERA sensors detect the electric response of culture cells suspended in a gel matrix, as a result to their interaction with virus's cells, rendering thus feasible its identification. Currently this is achieved empirically by examining the biosensor's response data curve. In this paper, we used specialized Artificial Neural Networks that were trained to recognize plant viruses according to biosensors' responses. Moreover, in order to increase the stability and the generalization capability of the classification model we applied a smoothing technique of the data. In addition, we used an advanced energy function for the training of the ANN network to reduce the complexity of the model.

Key words: Neural networks . intelligent systems . neural networks . biosensors . plant viruses

INTRODUCTION

The powerful tools of biotechnology are replacing the guesswork of early 20th Century medicine with 21st Century diagnostic skills that increasingly rely on knowledge of physiology at the molecular level. In the last two decades, medical diagnostics have been shaped by breakthroughs in immunology-fueled development of Enzyme-Linked Immunosorbent Antibody (ELISA) and other tests that use antibodies and chemical tags to find evidence of microorganisms in diagnostic samples. Although they are used as standard routine assays, these methods have a relative reliability (which rarely exceeds 70%) and a moderate sensitivity [1].

More recently, Reverse Transcriptase-Polymerase Chain Reaction (RT-PCR) evolved from a profound understanding of the structure and function of DNA and RNA. These last techniques are very sensitive and selective, although they are associated with risks for the persons working with them, due to the frequent use of radioactive labels. Generally, the equipment required for conducting determinations by conventional methods is expensive and space-consuming, while there is an additional requirement for trained personnel and laboratory infrastructure [2]. Moreover, the time needed for running a complete analysis varies from a few hours to several weeks, thus hindering the application of these methods to routine analysis. Therefore, conventional methods of determination have considerable disadvantages as far as the issues of practicality; time and cost of each analysis are concerned [3].

The extraordinary evolution of analytical technology now promises to make available an especially sensitive and accurate biosensor called the Bioelectric Response Assay (BERA) system. Indeed, it was evolution itself that provided the basic assay system at the core of biosensors--living cells.

BIOELECTRIC RECOGNITION ASSAY

A biosensor can be defined as a device incorporating a biological sensing element connected to a transducer [4]. Biosensors can play an important role to biosecurity, homeland security, food safety, environmental monitoring and medical diagnostics. Cell biosensors are based on the measurement of cellular responses to various compounds, such as measuring the cellular electrophysiological properties, in particular the electric potential, which reflects changes of a network of inter-related metabolic reactions.

The Bioelectric Recognition Assay (BERA) is a technology that detects the electric response of culture cells, suspended in a gel matrix, to various ligands, which bind to the cell and/or affect its physiology. In

Corresponding Author: Dr. C.P. Yialouris, Laboratory of Informatics, Department of Science, Agricultural University of Athens, Iera Odos 75, 118 55, Athens, Greece

previous studies [5-8] the potential application of the method for ultra rapid (some minutes) and ultra cheap tests for the detection of human and plant viruses has been demonstrated. Assays have been carried in an entirely crude sample and a high sensitivity of the method (0.1 ng) has been indicated, making it an attractive option for routine sample screening that could help reduce the exceeding use of advanced and costly molecular techniques, such as the Reverse Transcription Polymerase Chain Reaction (RT-PCR).

After producing a series of different biosensor generations, BERA biosensors were radically redesigned in order to produce the fifth generation which is optimal for diagnostic applications. Fifth generation biosensors are extremely miniaturized, consisting of a disposable array of gel beads loaded with cells. They are characterized by a very high degree of reproducibility (>99.9%), extremely low cost and high speed of manufacturing (with a production performance of approx. 1000 sensors per technician per hour). A further variation of the method, called the "6^{th} sensor generation" employs 5^{th} generation sensors which contain engineered cells expressing target-specific antibodies on their membrane [9-11].

The major applications of BERA technology are for detection of viruses and metabolic changes linked to disease; and for screening candidate molecules for use as commercial pharmaceutical agents [12]. The combination of simplicity, reliability and sensitivity make BERA the assay of choice for mass screening programs and environmental monitoring. Results are fast and offer an invaluable first look at infection, disease and contamination. In this work, BERA biosensors are used to detect plant viruses, such as the Tobacco Rattle Virus (TRV) and the Cucumber Green Mottle Mosaic Virus (CGMMV), using appropriate plant cells as the sensing elements. In respect to virology applications, each virus demonstrates a unique pattern of biosensor response over a specific range of concentrations, like a 'signature.' That is, individual viruses leave each one a characteristic "signature", which can be read as a graphical curve.

In order to develop an intelligent system for the detection of plant viruses, we applied Artificial Neural Networks (ANN) with different architecture. Next, we describe Artificial Neural Networks and more specifically Multilayer perceptrons which are the most popular feedforward classification models.

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks or simply Neural Networks refer to a group of algorithms that typically operate on a large number of simple interconnected components (or neurons). This networking enables the entire algorithm to perform much more powerful computations by combining the limited processing power of the separate components [13, 14]. The body of literature on Artificial Neural Networks (ANN) is intractably vast, so here only some very general comments will be made. An illustration of the structure of an Artificial Neural Network is illustrated in Fig. 1. Input data enter into the neural network from the input layer. The hidden layer associates the input with the output at the output layer.

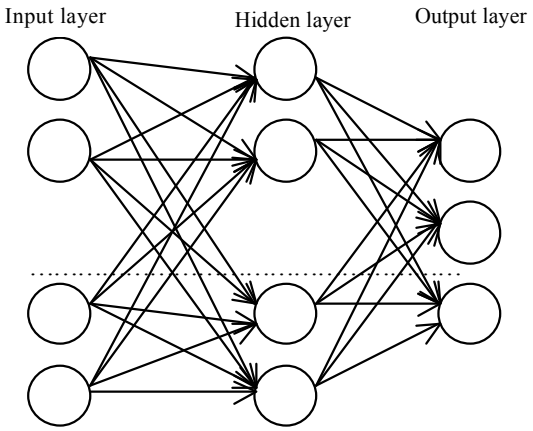


Fig. 1: The structure of an artificial neural network

The development of a neural network can be distinguished into two phases. The first phase is the training or learning one. A neural network learns in a way that is specified by the training method, the neurons' behaviour and neurons' interconnections. Many examples pairs of inputs and outputs are given to the neural network for training. For each pair of input-output, as a response to presented inputs, the synaptic weights change until the neural network learns to associate the inputs with the given output. The second phase of the neural network development is called 'testing', when it generates an output signal as a response to previously unknown inputs, i.e. it generalizes. The effectiveness of generalization can be expressed as the ratio of the correctly recognized input patterns to the total number of presented patterns during the test phase.

The generalization property as well as the convergence of the empirical risk to its expected value depend on the Vapnik-Chervonenkis (VC) dimension h of the machine. The VC dimension of a set of indicator functions is the maximum number of patterns that can be shattered by the set of functions.

The VC dimension of a set of real functions *f(x,w)* is the VC dimension of the set of indicator functions

$$I(x,w,\beta) = H(g(x,w) - \beta),\ w \in W,\ \beta \in (A,B),$$

Where, $A \le g(x,w) \le B, w \in W$ and $A, B \in \overline{\Re}$ and H is the step function

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \ge 0 \end{cases}.$$

In any case however, the results show that the generalization error R_{gene} is lower than a guaranteed risk [15]

$$R_{guarant} = R_{emp} + C(n,h,R_{emp},\eta)$$

where n is the number of training examples and the confidence interval C is a measure of the complexity of the machine and goes to zero as $n \to \infty$.

Thus, given a fixed number of training examples n, in order to minimize the generalization error we have to match the network complexity to the training set. This can be done by defining a structure of nested network families with increasing VC-dimension and then choosing the S_{opt} for which R_{gene} is minimized and subsequently minimize R_{emp}. The two questions rising are how to define a good structure and how to select S_{opt}.

The two most popular ways of defining structures are either by the architecture of the network or by the learning process. In the first case we vary the number of neurons of one of the hidden layers. We can start with a small number of neurons and add new ones when a certain criterion is met (growing algorithms) or conversely we can start with a large number of nodes and delete some of them under certain conditions (pruning algorithms). There is a large number of such algorithms [16-21]. In general all of them use some criterion of the saliency of the weights and prune the ones with small saliency. In addition, they use some information theoretic or statistical quantity to measure the increase in complexity that adding a node causes, in order to decide whether to add a node or not.

In the case of defining the structure by the learning process we use regularization techniques, i.e. an extra term is added to the error function. This type of function is designed to penalize the mappings that are not smooth and thus result in approximations with small variance. The error function then becomes

$$R_{emp}(w) = R_{emp}(w) + \lambda\Omega$$

where the parameter λ controls the effect of the regularization on training. Large values of λ result in decreased variance and increased bias and visa versa.

There are a lot of other regularizers proposed [22-25], many times derived from the nature of the problems the network has to solve, always with success.

Neural networks offer several advantages over conventional computing architectures. In this paper we present an extensive comparison among several feedforward neural network models in the context of the detection of different type of plant viruses. We present results from the application of Multilayer Perceptrons (MLP).

Multilayer perceptrons: The most popular feedforward neural network models are the Multilayer perceptrons which are trained with the Back-Propagation (BP) algorithm. This algorithm is a gradient descent procedure which minimizes the value of the Energy (Cost) function

$$E(w) = \sum_p \sum_j (t_{pj} - o_{pj})^2$$

where t_{pj} is the desirable output of node j corresponding to the p-th input pattern and o_{pj} the actual output of node j for the same input. The cost is minimized by iteratively updating the weights according to the following learning rule:

$$w_{ij}(n+1) = w_{ij}(n) - \eta \nabla E(w)$$

The transfer functions of network's nodes can be either continuously differentiable sigmoid functions saturating at 0/1 (or-1/1) or hard limiters. In order to improve the training speed and as well as the effectiveness of the neural network, we employed the *BFGS quasi-Newton* optimization algorithm [26]. As termination criteria we considered the:

- Maximum number of iterations or
- Mean square training error less than 0.01.

Once the neural network has been trained (i.e. its internal parameters are fine-tuned), it can accept new inputs (not previously seen) and attempt to compute an appropriate output. To produce an output, the trained network simply performs function evaluation. To assess the generalisation performance, a separate test is presented to the MLP after the completion of training. The ratio of correct recognitions to the total number of test patterns indicates the generalization capability of the MLP.

EXPERIMENTAL EVALUATION

Collection of the training data: The focus of our study is on the prediction of the presence of a virus. In our experiments, the measurements produced from the sensors are time series data. We used three types of BERA biosensors, one with antibodies of the TRV

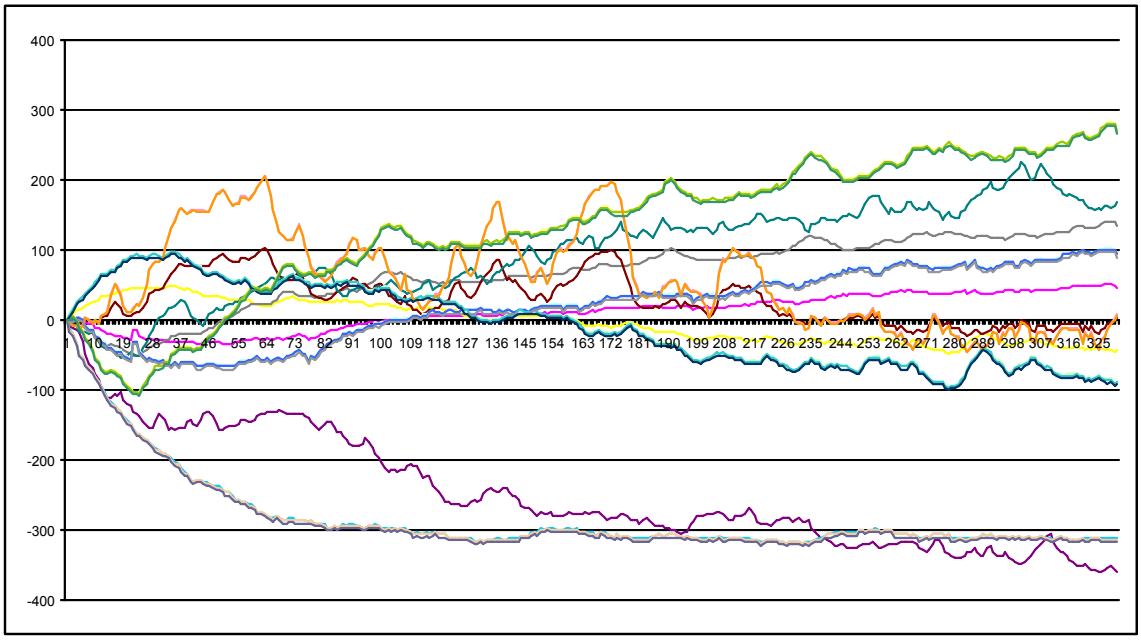


Fig. 2: Graphical representation of a part from the training data for the CGMMV

virus, one with antibodies of the CGMMV virus and one with no antibody. Each experiment contained the recording of the response of each one of the mentioned biosensors to a specific virus. That means a sample of an already known virus was tested by the three biosensors producing three different measurements for the same virus.

In our case 300 experiments per virus were performed producing 900 timeseries per virus which were used as training patterns for the neural network.

For any, given sequence of measurements for each experiment, we subtracted the value of the first measurement from each measurement of the sample. So the first measurement was always zero. A part of the training data set concerning CGMMV produced by a biosensor with antibodies against CGMMV is illustrated in Fig. 2.

We trained the first neural networks using as an input vector the whole time series (331 inputs). Several network topologies were applied with one hidden layer consisted of 10-40 hidden units. Unfortunately the performance of the trained neural networks was low. Then, we applied a resampling technique in order to extract the necessary features and reduce the input vector. According to the resampling rate we defined the number of the produced features and also the dimensionality of the problem. In addition, noise accompanies almost every real measurement and the presence of noise also affects the similarity significantly. Using smoothing techniques like a good resampling rate we could produce better quality of data without a considerable loss of information.

The system: The BERA biosensor diagnostic system was available as a desktop, laboratory-scale prototype that could be operated by expert users only. Further dissemination and/or commercialization of the device required refinements and engineering for a more compact, user-friendly unit. An essential element of this work was the development of a user-friendly software that will allow for a rapid and reliable recognition of the signature-like response of a BERA sensor against a sample containing a virus under detection.

Each biosensor was connected to an electrode made from 80% CuCopper, electrochemically coated with Silver an Ag/AgCl layer and having a diameter of 0.75 mm. Electrodes were connected to the data acquisition device, which comprised the PMD-1608FS A/D card (Measurement Computing, Middleboro, MA). The acquisition device was connected to a computer via USB port. So the signal (pattern) produced by the biosensor was stored in a computer file. This file may contain several signals (patterns) each of them composed of 331 records (data measurements). Each measurement of the record of this file contained the average voltage that the biosensor produced in a second. After the data recording was completed, we used an especially developed software, based on Artificial Neural Networks, for the identification of the virus. This software was developed using the Matlab package and it is trained to detect the CGMMV and TRV plant viruses. In this program (Fig. 3a) the user enters the name of the file to be examined. The program reads the data file and provides the user with an answer for virus identification (Fig. 3b). If the file contains

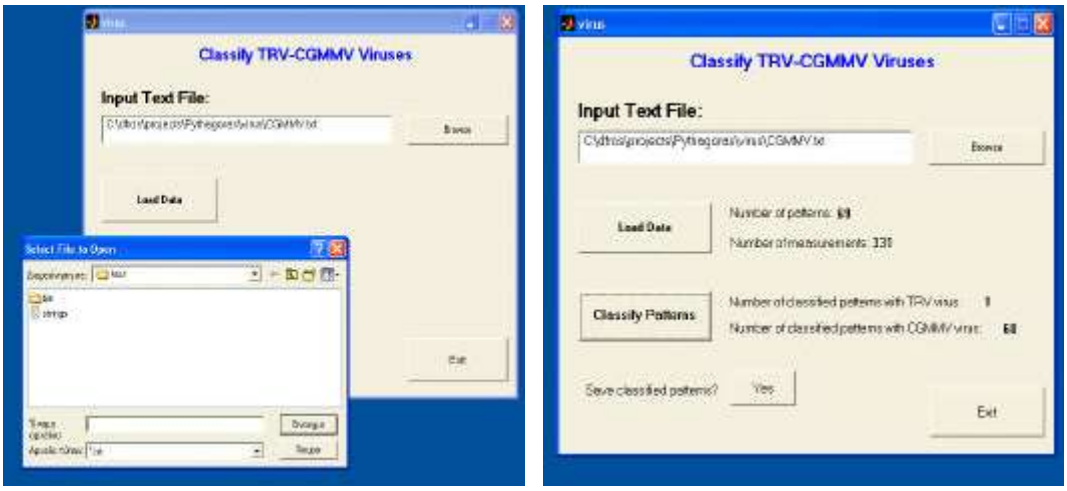


Fig. 3: The developed software for the identification of plant viruses (a) Entering the file name to be examined (b) The program reads the data file and provides the user with an answer for virus identification

Table 1: Average results using BFGS quasi-Newton algorithm for training MLP

Network architecture*	Resampling rate			
Number of units in				
the hidden layer	2	4	6	8
10	75.8%	84.3%	86.3%	78.9%
15	80.1%	86.1%	87.7%	81.2%
20	85.9%	87.2%	90.3%	81.7%

only one pattern the answer is directly given on the screen of the program.

The user, after the classification procedure, can save in a new file the examined data and for each pattern there is an extra label indicating the virus identification. This is necessary in case the file contains more than one pattern.

In our experimental study we wanted to discover the appropriate resampling rate and the MLP architecture (number of hidden units) that gives us the best results. To accomplish that we trained and tested several neural networks with different architectures and we also used several resampling rates to produce training data sets with different dimensionality. We considered MLP architectures consisting of the input layer (number of units according to the resampling rate), one hidden layer (10 to 20 sigmoid hidden units) and one output unit. Weights were randomly initialised in the range [-1, 1].

To compare the different network architectures, several series of experiments had to be conducted. For each type of MLP, we employed the 10-cross-validation method, in particular, ten experiments were performed with splits of data into training and test sets of fixed size. The average results were calculated from these ten trials and the best results are summarized in Table 1.

The experimental results in Table 1 indicate that increasing the resampling rate we got better results, by smoothing the data, until we have had a considerable loss of information, with resampling rate = 8, that leads to a poor performance.

The proposed energy function: Moreover, in order to increase the classification stability of the method and the generalisation performance, we used an advanced energy function [27] for the training algorithm of the neural network. The energy function to be minimized is the following:

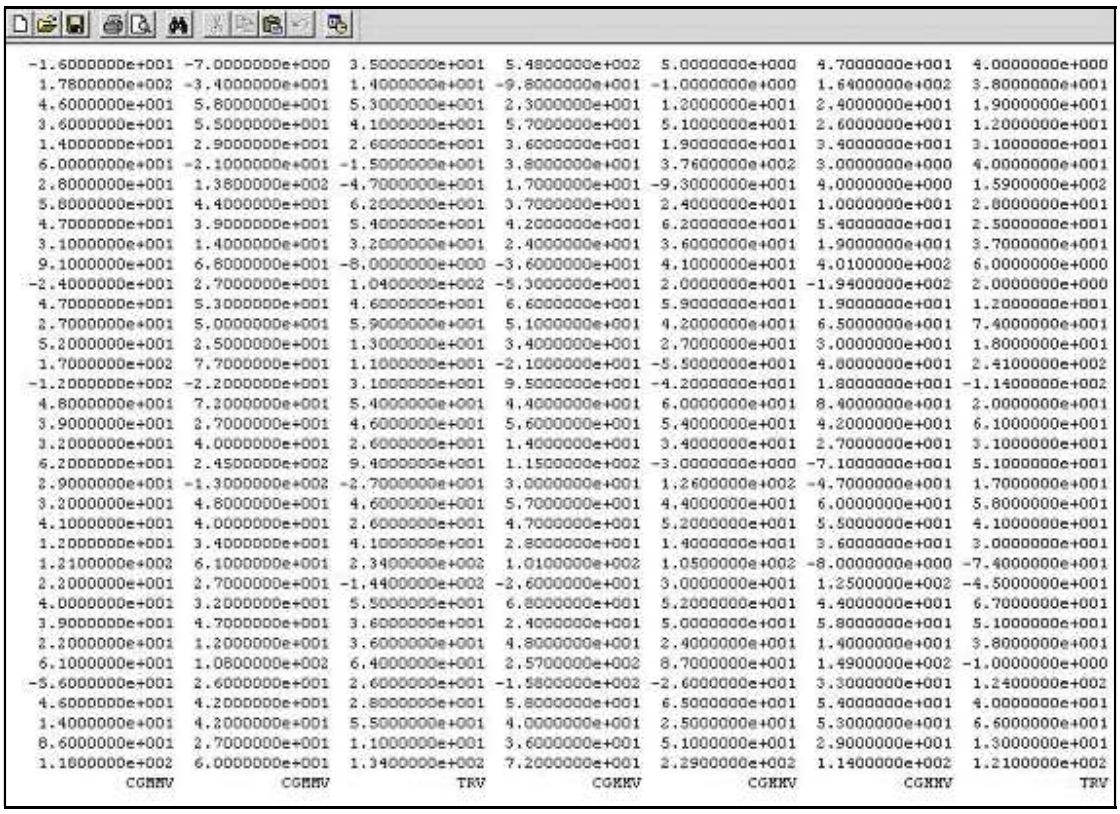
$$E = c_1 E_c + c_2 E_o + c_3 E_w$$

$$= c_1 \cdot \sum_p E_p + c_2 \cdot \sum_p \frac{n_o \cdot \sum_i o_{ip}^2}{\sum_i o_{ip}^2 + n_o \cdot k_1} + c_3 \cdot \frac{n_w \cdot \sum w_{ij}^2}{\sum w_{ij}^2 + n_w \cdot k_2}$$

Where, w_{ij} is the weight of the connection from node j to node i, o_{ip} is the output of node i and c_1, c_2, k_1, k_2, n_o, n_w are appropriate constants.

The extra terms added in order to increase the stability of the algorithm and the generalization capability of the network by reducing the number of active weights and nodes (i.e. weights and nodes with values not practically equal to zero). The average results using the proposed energy function are summarized in Table 2.

Comparing the results in Table 1 with Table 2, we observe that using the proposed energy function in the training algorithm we can get more robust classification models with better generalization performance.



-1.6000000e+001	-7.0000000e+000	3.5000000e+001	5.4800000e+002	5.0000000e+000	4.7000000e+001	4.0000000e+000
1.7800000e+002	-3.4000000e+001	1.4000000e+001	-9.8000000e+001	-1.0000000e+000	1.6400000e+002	3.8000000e+001
4.6000000e+001	5.8000000e+001	5.3000000e+001	2.3000000e+001	1.2000000e+001	2.4000000e+001	1.9000000e+001
3.6000000e+001	5.5000000e+001	4.1000000e+001	5.7000000e+001	5.1000000e+001	2.6000000e+001	1.2000000e+001
1.4000000e+001	2.9000000e+001	2.6000000e+001	3.6000000e+001	1.9000000e+001	3.4000000e+001	3.1000000e+001
6.0000000e+001	-2.1000000e+001	-1.5000000e+001	3.8000000e+001	3.7600000e+002	3.0000000e+000	4.0000000e+001
2.8000000e+001	1.3600000e+002	-4.7000000e+001	1.7000000e+001	-9.3000000e+001	4.0000000e+000	1.5900000e+002
5.8000000e+001	4.4000000e+001	6.2000000e+001	3.7000000e+001	2.4000000e+001	1.0000000e+001	2.8000000e+001
4.7000000e+001	3.9000000e+001	5.4000000e+001	4.2000000e+001	6.2000000e+001	5.4000000e+001	2.5000000e+001
3.1000000e+001	1.4000000e+001	3.2000000e+001	2.4000000e+001	3.6000000e+001	1.9000000e+001	3.7000000e+001
9.1000000e+001	6.8000000e+001	-8.0000000e+000	-3.6000000e+001	4.1000000e+001	4.0100000e+002	6.0000000e+000
-2.4000000e+001	2.7000000e+001	1.0400000e+002	-5.3000000e+001	2.0000000e+001	-1.9400000e+002	2.0000000e+000
4.7000000e+001	5.3000000e+001	4.6000000e+001	6.6000000e+001	5.9000000e+001	1.9000000e+001	1.2000000e+001
2.7000000e+001	5.0000000e+001	5.9000000e+001	5.1000000e+001	4.2000000e+001	6.5000000e+001	7.4000000e+001
5.2000000e+001	2.5000000e+001	1.3000000e+001	3.4000000e+001	2.7000000e+001	3.0000000e+001	1.8000000e+001
1.7000000e+002	7.7000000e+001	1.1000000e+001	-2.1000000e+001	-5.5000000e+001	4.8000000e+001	2.4100000e+002
-1.2000000e+002	-2.2000000e+001	3.1000000e+001	9.5000000e+001	-4.2000000e+001	1.8000000e+001	-1.1400000e+002
4.8000000e+001	7.2000000e+001	5.4000000e+001	4.4000000e+001	6.0000000e+001	8.4000000e+001	2.0000000e+001
3.9000000e+001	2.7000000e+001	4.6000000e+001	5.6000000e+001	5.4000000e+001	4.2000000e+001	6.1000000e+001
3.2000000e+001	4.0000000e+001	2.6000000e+001	1.4000000e+001	3.4000000e+001	2.7000000e+001	3.1000000e+001
6.2000000e+001	2.4500000e+002	9.4000000e+001	1.1500000e+002	-3.0000000e+000	-7.1000000e+001	5.1000000e+001
2.9000000e+001	-1.3000000e+002	-2.7000000e+001	3.0000000e+001	1.2600000e+002	-4.7000000e+001	1.7000000e+001
3.2000000e+001	4.8000000e+001	4.6000000e+001	5.7000000e+001	4.4000000e+001	6.0000000e+001	5.8000000e+001
4.1000000e+001	4.0000000e+001	2.6000000e+001	4.7000000e+001	5.2000000e+001	5.5000000e+001	4.1000000e+001
1.2000000e+001	3.4000000e+001	4.1000000e+001	2.8000000e+001	1.4000000e+001	3.6000000e+001	3.0000000e+001
1.2100000e+002	6.1000000e+001	2.3400000e+002	1.0100000e+002	1.0500000e+002	-8.0000000e+000	-7.4000000e+001
2.2000000e+001	2.7000000e+001	-1.4400000e+002	-2.6000000e+001	3.0000000e+001	1.2500000e+002	-4.5000000e+001
4.0000000e+001	3.2000000e+001	5.5000000e+001	6.8000000e+001	5.2000000e+001	4.4000000e+001	6.7000000e+001
3.9000000e+001	4.7000000e+001	3.6000000e+001	2.4000000e+001	5.0000000e+001	5.8000000e+001	5.1000000e+001
2.2000000e+001	1.2000000e+001	3.6000000e+001	4.8000000e+001	2.4000000e+001	1.4000000e+001	3.8000000e+001
6.1000000e+001	1.0800000e+002	6.4000000e+001	2.5700000e+002	8.7000000e+001	1.4900000e+002	-1.0000000e+000
-5.6000000e+001	2.6000000e+001	2.6000000e+001	-1.5800000e+002	-2.6000000e+001	3.3000000e+001	1.2400000e+002
4.6000000e+001	4.2000000e+001	2.8000000e+001	5.8000000e+001	6.5000000e+001	5.4000000e+001	4.0000000e+001
1.4000000e+001	4.2000000e+001	5.5000000e+001	4.0000000e+001	2.5000000e+001	5.3000000e+001	6.6000000e+001
8.6000000e+001	2.7000000e+001	1.1000000e+001	3.6000000e+001	5.1000000e+001	2.9000000e+001	1.3000000e+001
1.1000000e+002	6.0000000e+001	1.3400000e+002	7.2000000e+001	2.2900000e+002	1.1400000e+002	1.2100000e+002
CGHHV	CGHHV	TRV	CGKKV	CGKKV	CGKKV	TRV

Fig. 4: A part of the data file produced by the system after the classification procedure

Table 2: Average results using BFGS quasi-Newton algorithm with the proposed energy function for training MLP

Network architecture * Number of units in the hidden layer	Resampling rate			
	2	4	6	8
10	81.98%	87.90%	90.30%	83.90%
15	84.61%	90.68%	93.90%	85.80%
20	92.96%	93.41%	96.03%	93.61%

CONCLUSIONS

In this work, we applied Artificial Neural Networks (ANN) with different architecture in order to develop an intelligent system using biosensors for the detection of plant viruses. The system is based on already developed by the team method for detection of viruses named BERA. The main drawback of this method was the employment of an empiric way to detect the presence of a virus by examining the biosensor's response data curve. To overcome this problem, we used Artificial Neural Networks that are trained and specialized so that they recognize plant viruses in a selective pattern. In order to increase the classification stability of the method and the generalisation performance, we proposed an advanced energy function for the training algorithm of the neural network. We also used resampling as a smoothing technique to produce better quality of data without a considerable loss of information.

An important strength of the proposed classification approach is that it does not depend on the type of the classifier; therefore, it is quite general and applicable to a wide class of models including neural networks and other classification techniques. The next target of our work will be to develop a new neural network able to identify more than two viruses. Also, we will try to identify the concentration of the corresponding virus. Furthermore, we will train the system to classify human viruses.

ACKNOWLEDGEMENTS

The project is co-funded by European Social Fund & National Resources-O.P. "Education" II.

REFERENCES

1. Marquette, C.A. and L.J. Blum, 2006. State of the art and recent advances in immunoanalytical systems. Biosensors and Bioelectronics, 21 (8): 1424-1433.

2. Vernet, G., 2004. Molecular diagnostics in virology. J. Clin. Virol., 31: 239-247.
3. Kintzios, S., 2007. Cell-based sensors in clinical chemistry. Mini Rev. Med. Chem (In Press).
4. Eggins, B., 1996. Biosensors-An Introduction. Wiley and Teubner, Chichester.
5. Kintzios, S., E. Pistola, P. Panagiotopoulos, M. Bomsel, N. Alexandropoulos, F. Bem, I. Biselis and R. Levin, 2001a. Bioelectric recognition assay (BERA). Biosensors and Bioelectronics, 16: 325-336.
6. Kintzios, S., E. Pistola, J. Konstas, F. Bem, T. Matakiadis, N. Alexandropoulos, I. Biselis and R. Levin, 2001b. Application of the Bioelectric Recognition Assay (BERA) for the detection of human and plant viruses: Definition of operational parameters. Biosensors and Bioelectronics, 16: 467-480.
7. Kintzios, S., E. Makrygianni, E. Pistola, P. Panagiotopoulos and G. Economou, 2003. Effect of amino acids and amino acid analogues on the in vitro expression of glyphosate tolerance in Johnsongrass (*Sorghum halepense* L. pers.). J. Food Agric. Environ., 3: 180-184.
8. Kintzios, S., F. Bem, O. Mangana, K. Nomikou, P. Markoulatos, N. Alexandropoulos, C. Fasseas, V. Arakelyan, A.-L. Petrou, K. Soukouli, G. Moschopoulou, C. Yialouris and A. Simonian, 2004. Study on the mechanism of Bioelectric Recognition Assay: evidence for immobilized cell membrane interactions with viral fragments. Biosensors and Bioelectronics, 20: 907-916.
9. Kintzios, S., J. Goldstein, A. Perdikaris, G. Moschopoulou, I. Marinopoulou, O. Mangana, K. Nomikou, I. Papanastasiou, A.-L. Petrou, V. Arakelyan, A. Economou and A. Simonian, 2005. The BERA Diagnostic System: An all-purpose cell biosensor for the 21th Century. 5th Biodetection Conference, Baltimore, MD, USA, 9-10/06/05.
10. Moschopoulou, G. and S. Kintzios, 2005. Membrane engineered Bioelectric Recognition Cell sensors for the detection of subnanomolar concentrations of superoxide: A novel biosensor principle. International Conference on Instrumental Methods of Analysis (IMA) 2005, Crete, Greece, 1-5/10/2005.
11. Moschopoulou, G. and S. Kintzios, 2006. Application of "membrane-engineering" to bioelectric recognition cell sensors for the detection of picomole concentrations of superoxide radical: a novel biosensor principle. Anal. Chim. Acta, pp: 90-96, 573-574.
12. Kintzios, S., I. Marinopoulou, G. Moschopoulou, O. Mangana, K. Nomikou, K. Endo, I. Papanastasiou and A. Simonian, 2006. Construction of a novel, multi-analyte biosensor system for assaying cell division. Biosensors and Bioelectronics 21(7): 1365-1373.
13. Bishop, C., 1995. Neural Networks for Pattern Recognition, Oxford University Press.
14. Haykin, S., 1999. Neural Networks: A comprehensive foundation, Second Edition, Prentice Hall.
15. Guyon, I., V. Vapnik, B. Boser, L. Bottou and S.A. Solla, 1992. Structural Risk Minimization for Character Recognition, Advances in Neural Information Processing Systems 4.
16. Bartlett, E.B., 1994. Dynamic node architecture learning: an information theoretic approach, Neural Networks. 7: 129-140.
17. Wynne-Jones, M., 1992. Node splitting: A constructive algorithm for feed-forward neural networks. Advances in Neural Information Processing Systems 4.
18. Yasui, S., 1997 Convergence suppression and divergence facilitation: Minimum and joint use of hidden units by multiple outputs. Neural Networks, 10: 353-367.
19. Le Cun, Y., J.S. Denker and S.A. Solla, 1990. Optimal Brain Damage. Advances in Neural Information Processing Systems 2.
20. Reed, R., 1993. 'Pruning algorithms-A survey' IEEE Trans. On Neural Networks, 4: 740-747.
21. Levin, A.U., T.K. Leen and J.E. Moody, 1994. Fast pruning using principal components. Advances in Neural Information Processing Systems 6.
22. Weigend, A.S., D.E. Rumelhart and B.A. Huberman 1991. Generalization by weight-elimination with application to forecasting. Advances in Neural Information Processing Systems 3.
23. Jean, J.S.N. and J. Wang 1994. Weight smoothing to improve network generalization. IEEE Trans. Neural Networks, 5: 752-763.
24. Wu, L. and J. Moody, 1996. A smoothing reguralizer for feedforward and recurrent neural networks. Neural Computation, 8: 461-489.
25. Lang, K.J. and G.E. Hinton, 1990. Dimensionality reduction and prior knowledge in E-set recognition. Advances in Neural Information Processing Systems 2.
26. Dennis, J.E. and R.B. Schnabel, 1983. Numerical methods for unconstrained optimization and nonlinear equations. Englewood Cliffs, NJ: Prentice-Hall.
27. Tzafestas, S.G., P. Dalianis and G. Anthopoulos, 1996. On the Overtraining Phenomenon of Back-Propagation Neural Networks-IMACS Transactions: Mathematics and Computers in Simulation, 40 (5-6): 507-521.