

Forecasting Short-Term Time Series using an Artificial Immune System

Vasilis-Thanos Anagnostopoulos

Piraeus, 2015

Contents

1	Introduction	1
2	Artificial Immune Systems	1
3	Implementation	2
3.1	Training the AIS	3
3.2	Forecast	5
3.3	Parameters	6
4	Case study	6
	Bibliography	10

List of Tables

3.1	The input parameters of the algorithm.	6
3.2	The output parameters of the algorithm.	6
4.1	The RMSE of the models	8

List of Figures

4.1	Linear model	7
4.2	Exponential model	7
4.3	AIS model	8
4.4	AIS model(2)	9

List of Algorithms

2.1	Basic AIS [2, p. 427]	1
3.1	The immune memory creation algorithm [1]	3

1 Introduction

In the following sections, we will review how short-term forecasting in time series can be accomplished using an artificial immune system. In section 2 we will mention how an artificial immune system works Afterwards we will present our implementation and some results in CO_2 measurements.

2 Artificial Immune Systems

The natural immune system is a complex system for the operation of which many publications have been made [6]. Apart from its ability to fight infections and foreign cells towards the body, it also has a memory, a particularly important property, since memory allows it to identify and manage more directly an invasion by pathogens that have infected the organism in older periods.

An artificial immune system (AIS) models the ability of the natural immune system of the vertebrates to detect foreign cells in the organism. The result is a new computing model that is capable of recognizing patterns and this new computing model is applied mainly to detect anomalies [2, 5].

The implementation of an AIS includes a variety of topics such as [2, 3]:

- hybrid structures and algorithms which take into account the mechanisms of the immune system such as the detection of foreign cells with a certain affinity for the storage of informations and reuse of that particular informations.
- computational algorithms based on the principles of the immune system, such as distributed processing, the principle of clone selection and the theory of the immune system.

Using the above, Engelbrecht [2] lists the basic step for creating a basic AIS (see algorithm 2.1).

Algorithm 2.1: Basic AIS [2, p. 427]

```
1 Initialize a set of artificial lymphocytes (ALC) as population  $C$  ;
2 Determine the antigen patterns as training set  $D_T$  ;
3 while some stopping condition(s) not true do
4   for each antigen pattern  $z_p \in D_T$  do
5     Select a subset of ALCs for exposure to  $z_p$ , as population,  $S \leq C$  ;
6     for each ALC  $x_i \in S$  do
7       Calculate the antigen affinity between  $z_p, x_i$  ;
8     Select a subset of ALCs with the highest calculated antigen affinity as
      population  $H \leq S$  ;
9     Adapt the ALCs in  $H$  with some selection method, based on the calculated
      antigen affinity and/or the network affinity among ALCs in  $H$  ;
10    Update the stimulation level of each ALC in  $H$  ;
```

According to Engelbrecht [2] the algorithm is expanded below:

Initialize C and determining D_T : The population C may be populated from randomly generated artificial lymphocytes¹ (ACLs) or with ALCs that are initialized with a cross section of the data set to be learned.

¹The lymphocytes is a type of white blood cell which is found on the natural immune system and is responsible for defending the body against infections [7].

Stopping condition for the while: Most models of AIS have a stopping condition based on the convergence of the population of the population of ACL or after a specified number of iterations.

Selecting a subset S of ALCs: The subset S can be the entire set of P or a number of randomly selected ALCs from P .

Calculating the antigen affinity: The antigen affinity is the measurement of similarity or dissimilarity between the ACLs and the antigen patterns.

Selecting a subset of H of ALCs: In some of the AIS models, the selection of highest affinity ALCs is based on a preset affinity threshold. Thus, the selected subset H can be the entire set S , depending on the preset affinity threshold.

Calculating the network affinity: This is the measurement of affinity between two ALCs. The different measures of network affinity are the same as those for antigen affinity. A preset network affinity threshold determines whether two or more ALCs are linked to form a network.

Adapting the ALCs in subset H : The adaptation of ALCs can be seen as the maturation process of the ALC.

Updating the stimulation of an ALC: The stimulation level is calculated in different ways in existing AIS models.

According to Engelbrecht [2], several algorithms have been proposed to solve such AIS such as:

classical view: of the immune system is that the immune system distinguishes between self and non-self, using lymphocytes produced in the lymphoid organs. These lymphocytes "learn" to bind to antigen.

clonal selection theory: where an active B-Cell produces antibodies through a cloning process. The produced clones are also mutated.

network theory: where the immune system has the ability to distinguish between dangerous and non-dangerous antigen.

danger theory: where it is assumed that B-Cells form a network. When a B-Cell responds to an antigen, that B-Cell becomes activated and stimulates all other B-Cells to which it is connected in the network.

Artificial immune systems have been successfully applied in many areas, such as the detection of anomalies, in classification of data, in detecting viruses, e.t.c. [5, 2, 3].

3 Implementation

In this section we present the implementation of the AIS. It was decided the algorithm to be developed using MATLAB® [4]. MATLAB® (MATrix LABoratory) is a high-level and fourth-generation programming language designed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. It is used in many domains such as signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology [8, 4].

This algorithm could have been written in a low-level computer language such as C++, which would have freed it from dependency on other software. However, although low-level

languages are very powerful and flexible, they are not as conducive to rapid program development or easy implementation of graphical interfaces. Similar with interpreted languages, such as Python, which allow easier program development than low-level languages do, still they do not include all the prepackaged algorithmic and graphical tools in the basic version of MATLAB®.

So even though MATLAB® is only available commercially, it is commonly used in research, and the advantages that offers overcome the disadvantage of being a commercial product. With MATLAB® the developed software can be executed in every operating system (Windows, Linux or Mac OS) where MATLAB® is installed. In a later stage someone could also replace MATLAB® with GNU Octave².

3.1 Training the AIS

The implementation of the AIS was based on the algorithm described by Dudek [1] (see algorithm 3.1).

Algorithm 3.1: The immune memory creation algorithm [1]

- 1 Loading of the training set of antigens ;
 - 2 Generation of the initial antibody population ;
 - 3 Antibody affinity calculation ;
 - 4 Activated antibody detection and evaluation ;
 - 5 **while** *some stopping condition(s) not true* **do**
 - 6 Clonal selection ;
 - 7 Clone hypermutation ;
 - 8 Antibody affinity calculation ;
 - 9 Activated antibody detection and evaluation ;
 - 10 Selection of the best antibodies ;
-

The detailed algorithm of the immune memory creation in the proposed model is described below:

Loading of the training set of antigens: The whole dataset is divided into two subsets (testing and training). Immune memory is trained using the training set and the model afterwards is tested using the testing subset [1].

To simplify the our model, we need to preprocessed the dataset in order to get rid of the time series trend and seasonality [1].

If $P_x = [P_x(1), \dots, P_x(n)]$ and $P_y = [P_y(1), \dots, P_y(n)]$ is a vector of the values we want to forecast and a vector of the forecast values accordingly with $n \in \mathbb{N}^+$ being the period of the time series then the input pattern $x = [x(1), \dots, x(n)]$ and the output pattern $y = [y(1), \dots, y(n)]$ presented in our model are vectors with components defined in equation (1) and (2):

$$x(i) = \frac{P_x(i)}{\bar{P}_x} \quad (1)$$

$$y(i) = \frac{P_y(i)}{\bar{P}_x} \quad (2)$$

with $i = 1, \dots, n$. The model learns to map $x \rightarrow y$. Pattern x is presented to the model and the pattern y is obtained as the output of the model. Equation (2) is used to receive the forecasted load curve P_y [1].

²<http://www.gnu.org/software/octave/>

Concatenated patterns x and y form antigens ($AG = [x(1), \dots, x(n), y(1), \dots, y(n)]$). Thus each antigen is composed of $2 \cdot n$ amino acids with which $P_x(i), P_y(i) \in \mathbb{R}$ and $i \in [1, n]$. It is assumed that the only components of the immune system are antibodies built analogously to antigens. Each antibody has two chains (x to detect the x -chain of the antigens and y to memorize the y -chain of detected antigens) [1].

The task of the immune system is to learn to map a set of antigens into a set of antibodies. For each sub - period of the period (for example if our period is a month, then our sub-period could be each day of the month) the separate immune memory is created using antigens representing only this sub-period (e.g. for forecasting the second day of the month, the AIS learns from antigens which x -chain represents the first day of the month and y -chain represents the second day pattern). [1].

Generation of the initial antibody population: An initial antibody population is created by copying all the antigens from the training set (antibodies and antigens have the same structure). This way of initialization prevents inserting antibodies in empty regions without antigens [1].

Antibody affinity calculation: The affinity measure is based on the distance between x -chains of antigens and antibodies. The Euclidean distance (see equation (3)) is used [1].

$$d = \sqrt{\sum_{i=1}^n [x_{Ab}(i) - x_{Ag}(i)]^2} \quad (3)$$

where x_{Ab} and x_{Ag} are the x -chains of the antibody and antigen, respectively [1]. Probably more affinity measures can be used and a more thorough search should be done.

Activated antibody detection and evaluation: If the affinity of the antibody for the antigen is smaller than or equal to the cross-reactivity threshold r , it means that the antigen lies in the antibody recognition region (the antibody is activated by the antigen). For this antibody the forecast error is calculated by equation (4) [1]:

$$d = \frac{1}{n} \cdot \sum_{i=1}^n \left| \frac{y_{Ab}(i) - y_{Ag}(i)}{y_{Ag}(i)} \right| \cdot 100\% \quad (4)$$

where y_{Ab} and y_{Ag} are the y -chains of the antibody and antigen activating this antibody. If several antigens lie in the antibody recognition region, the error is calculated for each of them. The mean error $\bar{\delta}$ is applied to evaluate the antibody and is minimized in the following iterations of the algorithm [1].

Stopping condition(s): The algorithm stops if the maximum number of iteration L is reached [1] or the population of antigens does not change (in two consecutive iterations the populations of the antigens remains the same) or there is not even one enable antigens. In the last case the AIS failed and the forecast is impossible.

Clonal selection: Each antibody secrets as many clones as the antigens that lying in its recognition region. Thus most clones are generated in the dense clusters of antigens [1].

Clone hypermutation: The main goal of hypermutation is to improve the diversity of the immune system in order to effectively recognize new antigens. The hypermutation is realized as follows. Each clone of the antibody is shifted towards different antigen lying in the recognition region of this antibody. The bigger the error δ for the given antigen is, the bigger shift toward this antigen is. The shift is calculated according to the equation (5) and (6) [1]:

$$x_{Ab}(i) = x_{Ab}(i) + \eta(i)[x_{Ag}(i) - x_{Ab}(i)] \quad (5)$$

$$y_{Ab}(i) = y_{Ab}(i) + \eta(i + n)[y_{Ag}(i) - y_{Ab}(i)] \quad (6)$$

where $\eta \in (0, 1)$ is a learning coefficient calculated from the hyperbolic tangent sigmoid function from equation (7):

$$\eta(i) = \frac{2}{1 + \exp[-\beta \cdot \delta \cdot N_i(1, 0.1)]} - 1 \quad (7)$$

where β is the shape parameter and $N_i(1, 0.1)$ are the independent normally distributed random numbers with mean 1 and standard deviation 0.1.

Random factor in equation (7) is introduced to avoid stagnation of the learning process caused by getting into local minimums of the error function. This type of hypermutation produces new antibodies only in the regions covered by antigens [1].

Selection of the best antibodies: For each antigen from the training set, the set of antibodies activated by this antigen is determined. Only one antibody from this set, with the best evaluation δ , is selected to the next population. So the clonal expansion, unnecessary in this model, is halted. The maximum number of antibodies in the next population is equal to the number of antigens, but the real number of antibodies is usually smaller because the same antibody could be selected by the several antigens (it depends on the value of the cross-reactivity threshold r). Outliers, i.e. antigens lying away from other antigens, is represented by the separate antibodies [1]. After the selection of the best antibodies the antibodies are re-evaluated again.

3.2 Forecast

The forecasting procedure applies the learned immune memory to new antigens consisting only of x-chain. The new antigen is detected by the antibodies with similar x-chains and the y-chain of the antigen is reconstructed from the y-chains of these antibodies [1].

The antibodies represent overlapping clusters of similar antigens. In the forecast procedure a new antigen having only x-chain is presented. The Ω set of antibodies, activated by this antigen, is determined. The y-chain of the input antigen is reconstructed from the y-chains of the antibodies contained in the Ω set (see equation (8)).

$$\hat{y}_{Ag} = \frac{\sum_{j=1}^{|\Omega|} w_j \cdot y_{Ab}^j(i)}{\sum_{i=1}^{|\Omega|} w_j} \quad (8)$$

where $w_j \in (0, 1)$ is the weight whose value is dependent on the distance d_j between the input antigen and the j-th antibody from the Ω set (see equation (9)).

$$w_j = 1 - \frac{d_j}{r} \quad (9)$$

Antibodies, closer to the antigen, have higher influence on the forecast. If an antigen is not recognized by the antibodies, it means that it represents a new shape, not contained in the training set. In this case the cross-reactivity threshold r is consistently being increased until the antigen is recognized by one or more antibodies. The level of confidence in the forecast in such a case is low and the forecast should be verified [1].

The quality criteria of the AIS is the forecast error. For that reason, the training data are used in the forecast and the forecast values are compared to the actual values. The smaller

Parameters	Explanation
original_data	the array of input data used for training the AIS. It should be an array with the lines representing the a hole period of measurements
threshold	declares the the cross-reactivity threshold r , for which an antigen is activated
relax_threshold	if an antigen does not react the threshold is relax according to this particular percentage
max_iterations	The maximum number of iterations for producing the antibodies
beta	The shape parameter
forecast_data:	The data for which a forecast from the AIS will be done. If left empty then only the next period will be forecasted.
input_antibodies	An initial population of antibodies that can be used
hypermutation	A parameter that instructs the AIS not to produce new antibodies. Should only be used if the antibodies are provided
diagnostics	If set to true diagnostics messages will be printed

Table 3.1: The input parameters of the algorithm.

Parameters	Explanation
forecast	the forecast values
confidence	the confidence of the forecast values. If there is not an antibody that reacts to the values then threshold is relaxed and that is shown in the confidence values that is between 0 and 1
antibodies	the produced antibodies from the AIS
iterations	the iterations that the AIS needed
total_time	the total running time of the algorithm
forecast_antigen	the forecast antigens that were used to produce the forecast
rmse	the rmse in the original data of the AIS
errors	the errors in the training data from which the rmse is produced

Table 3.2: The output parameters of the algorithm.

the error, the better the conversion of the AIS. So the RMSE from this procedure is an indication of how well can the AIS predict the values of the existing shapes. As already mention if a new shape is introduced to the AIS then the level of confidence is reduced.

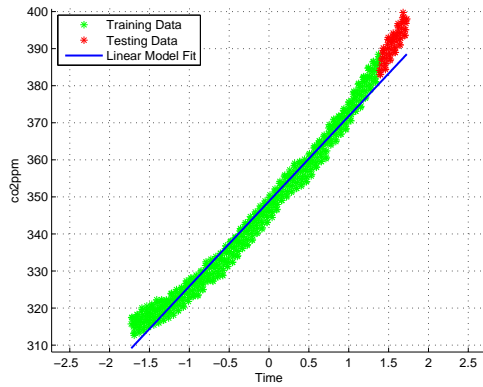
3.3 Parameters

In table 3.1 the input parameters of the algorithm are summarized and in table 3.2 the output parameters are summarized.

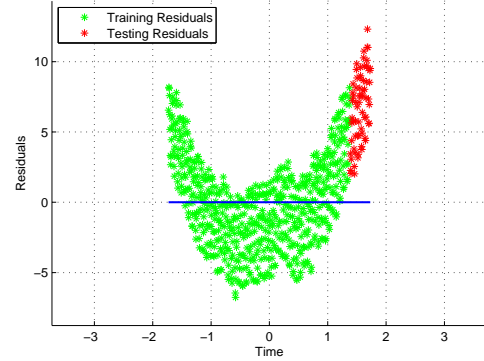
4 Case study

The Mauna Loa Observatory (MLO) is an atmospheric baseline station on Mauna Loa, on the island of Hawaii. MLO is an atmospheric research facility that has been continuously monitoring and collecting data related to atmospheric change since the 1950's and is known especially for the continuous monitoring of atmospheric carbon dioxide (CO_2), [9].

In our case, we will use the monthly CO_2 values from the MLO and we will combining them with the AIS, so we could forecast the CO_2 values of the next year.

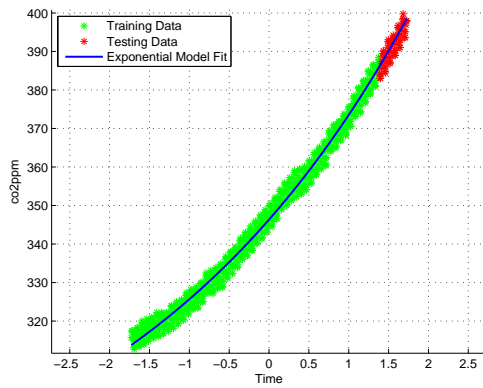


(a) Visualization of the model

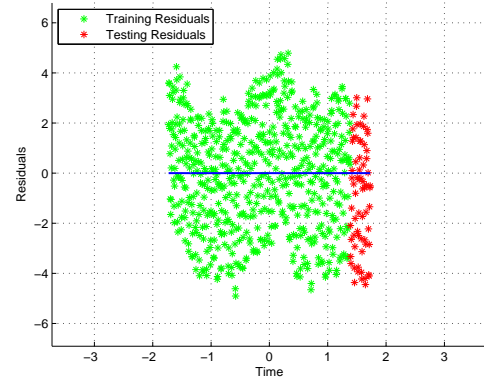


(b) Visualization of the residuals

Figure 4.1: Linear model



(a) Visualization of the model

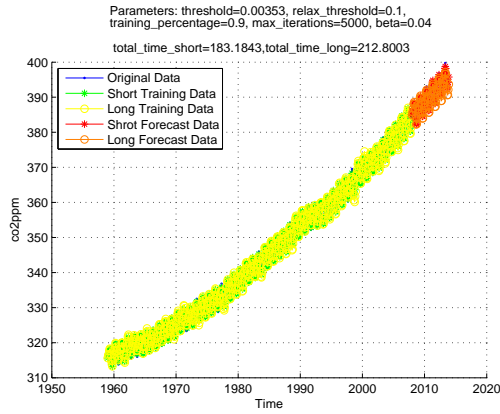


(b) Visualization of the residuals

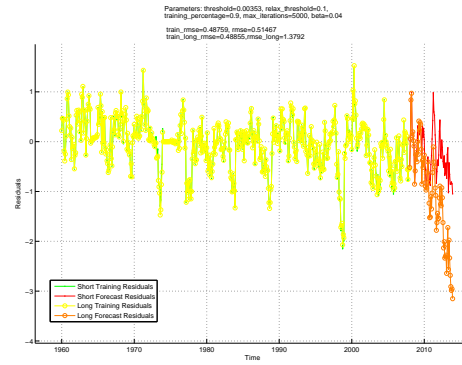
Figure 4.2: Exponential model

In a first stage the linear model (see fig. 4.1) and the exponential model (see fig. 4.2) were calculated so we could have a comparison model. Afterwards two AIS models (see fig. 4.3 and 4.4) were calculated using two different set of parameters (see table 4.1 for the parameters).

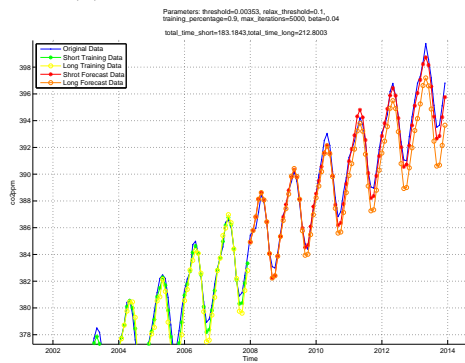
In all cases the AIS model has lower residuals from the linear and the exponential model. In a latter step, instead of using the values from the previous period to forecast the next period, we used the values from the forecast as input values (orange and yellow lines in the diagrams). In all cases the RMSE error (see table 4.1) were better. So the AIS can be used for short and long period forecasts.



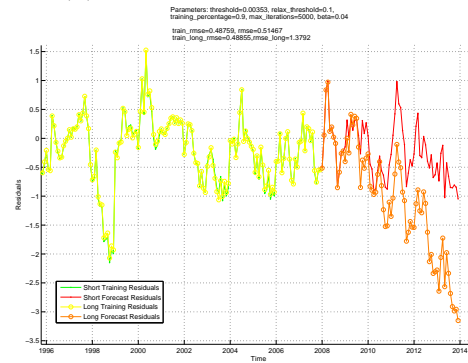
(a) Visualization of the model



(b) Visualization of the residuals



(c) Visualization of the model (2)

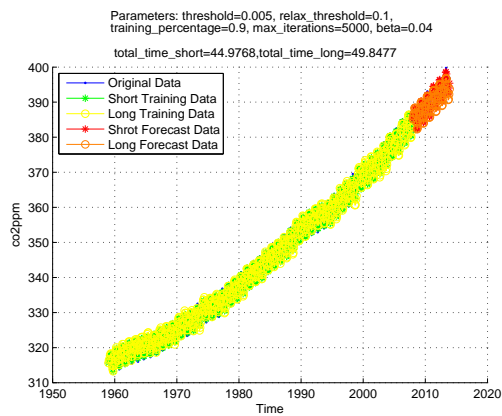


(d) Visualization of the residuals (2)

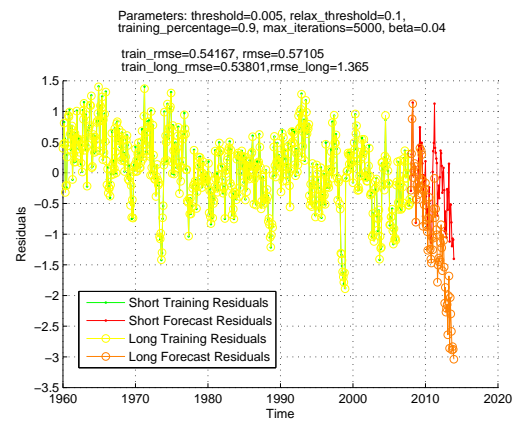
Figure 4.3: AIS model

Model	Parameters	RMSE(train)	RMSE(forecast)
Linear Model	-	3.183062	7.255561
Exponential Model	-	2.212280	2.315118
AIS Short model	threshold= 0.00353 relax_threshold = 0.1 beta = 0.04	0.48759	0.51467
AIS Long model	threshold= 0.00353 relax_threshold = 0.1 beta = 0.04	0.48855	1.3792
AIS Short model (2)	threshold= 0.005 relax_threshold = 0.1 beta = 0.04	0.54167	0.57105
AIS Long model (2)	threshold= 0.005 relax_threshold = 0.1 beta = 0.04	0.53801	1.365

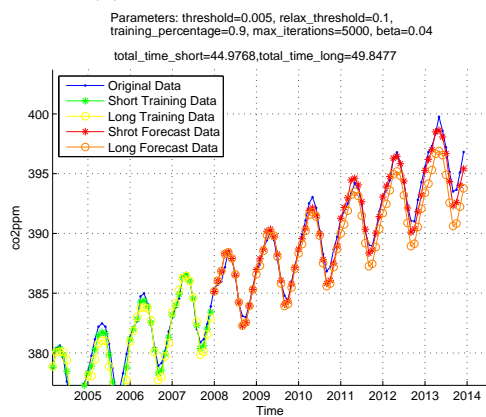
Table 4.1: The RMSE of the models .



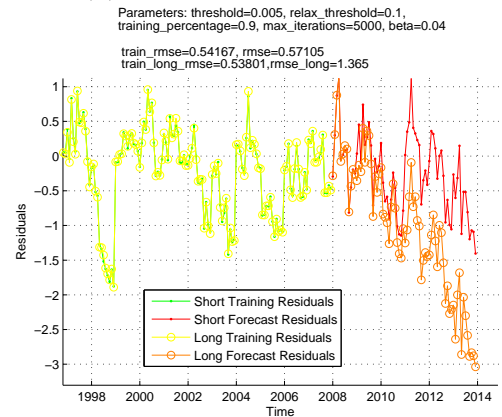
(a) Visualization of the model



(b) Visualization of the residuals



(c) Visualization of the model (2)



(d) Visualization of the residuals (2)

Figure 4.4: AIS model(2)

References

- [1] Grzegorz Dudek. "Artificial immune system for short-term electric load forecasting". In: *Artificial Intelligence and Soft Computing–ICAISC 2008*. Springer, 2008, pp. 1007–1017.
- [2] Andries P. Engelbrecht. *Computational Intelligence: An Introduction*. 2nd. Wiley Publishing, 2007. ISBN: 0470035617.
- [3] Vasilios K. Karakasis. "Models of artificial immune systems for extracting knowledge from data sets (in Greek)". MA thesis. National Technical University of Athens, 2005. URL: http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2005-0229/DT2005-0229.pdf.
- [4] MATLAB. *version 8.3.0 (R2014a)*. Natick, Massachusetts: The MathWorks Inc., 2014.
- [5] Wikipedia. *Artificial immune system — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/wiki/Artificial_immune_system. [Πρόσβαση στις 20 Ιανουαρίου 2015]. 2015. URL: http://en.wikipedia.org/wiki/Artificial_immune_system.
- [6] Wikipedia. *Immune system — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/wiki/Immune_system. [Πρόσβαση στις 24 Ιανουαρίου 2015]. 2015. URL: http://en.wikipedia.org/wiki/Immune_system.
- [7] Wikipedia. *Lymphocyte — Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/Lymphocyte>. [Πρόσβαση στις 24 Ιανουαρίου 2015]. 2015. URL: <https://en.wikipedia.org/wiki/Lymphocyte>.
- [8] Wikipedia. *MATLAB — Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/MATLAB>. [Online; accessed 27-June-2015]. 2015. URL: <https://en.wikipedia.org/wiki/MATLAB>.
- [9] Wikipedia. *Mauna Loa Observatory — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Mauna_Loa_Observatory. [Online; accessed 23-July-2015]. 2015. URL: https://en.wikipedia.org/wiki/Mauna_Loa_Observatory.