

Memetic Pareto Evolutionary Artificial Neural Networks for the determination of growth limits of *Listeria Monocytogenes*

J.C. Fernández¹, P.A. Gutiérrez¹, C. Hervás¹, and F.J. Martínez²

¹ Department of Computer Science of the
University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain
{fernandezcaballero@gmail.com, i02gupep@uco.es, chervas@uco.es}

² Department of Management and Quantitative Methods, ETEA,
Escritor Castilla Aguayo 4, 14005, Córdoba, Spain
{fjmestud@etea.com}

Abstract

The main objective of this work is to automatically design neural network models with sigmoidal basis units for classification tasks, so that classifiers are obtained in the most balanced way possible in terms of CCR and Sensitivity (given by the lowest percentage of examples correctly predicted to belong to each class).

*We present a Memetic Pareto Evolutionary NSGA2 (MPENSGA2) approach based on the Pareto-NSGAI evolution (PNSGAI) algorithm. We propose to augment it with a local search using the improved Rprop—IRprop algorithm for the prediction of growth/no growth of *L. monocytogenes* as a function of the storage temperature, pH, citric (CA) and ascorbic acid (AA). The results obtained show that the generalization ability can be more efficiently improved within a framework that is multi-objective instead of a within a single-objective one.*

1. Introduction

There are many fields of study, such as medicine and epidemiology, where it is very important to predict a binary response variable or, equivalently, the probability of occurrence of an event (success), in terms of the values of a set of explicative variables related to it. In this work, we discuss learning and generalization improvement of classifiers designed using a multi-objective evolutionary learning algorithm (MOEA). Specifically we investigate the generation of neural network classifiers based on two objectives: the correct classification rate, C, and the sensitivity, S. The basic structure of the MOEA has been modified by introducing an additional step, where each individual in the population has been enhanced by a local search method. For this purpose, a *Memetic Pareto Evolutionary NSGA2 (MPENSGA2) algorithm* has been developed.

Our MOEA can be applied for predictive microbiology, a specific application of the field of mathematical modelling for describing the behaviour of pathogen and spoilage micro-organisms under a given set of environmental conditions. The importance of this application is the demand for healthier and more convenient food products, as scientists recognize that there is an increasing need to model microbial growth limits [1]. Growth / no-growth models or boundary models quantify the probability of microbial growth and define combinations of factors that prevent growth.

The rest of the paper is organized as follows. In Section 2, background materials are covered followed by an explanation of accuracy and sensitivity. In Section 4, our problem is described as a multi-objective optimization. The MPENSGA2 algorithm is described in Section 5, followed by experimental design in Section 6 and the conclusions, which are drawn in Section 7.

2. Background materials

Evolutionary Artificial Neural Networks (EANNs) have been a key research area in Computer Science for the last decade. On one hand, methods and techniques have been developed to find better approaches for evolving ANNs, and more specifically, multi-layer feed-forward ANNs. On the other hand, finding a good ANN architecture has been a debatable issue as well in the field of Artificial Intelligence. Methods for network-growing denominated “constructive algorithms” [2,3] start with a small network (usually a single neuron). This network is trained until it is unable to continue learning, then new components are added to the network. This process is repeated until a satisfactory solution is found. Destructive methods, also known as “pruning algorithms” [4], start with a big network, that is able to learn but usually ends in over-fitting, and then some

processes are applied in order to remove the connections and nodes that are not useful. However, all these methods usually suffer from slow convergence and long training time. In addition, they are gradient-based techniques and, therefore, can easily get stuck in a local minimum. EANNs provide a more successful platform for optimizing both network performance and architecture simultaneously.

Evolutionary computation has been widely used in the last few years to evolve neural-network architectures and weights. There have been many applications for parametric learning [5] and for both parametric and structural learning [6]. This may indicate that there is an extensive need for finding better ways to evolve ANN. A major advantage of the evolutionary approach over traditional learning algorithms such as BP (Backpropagation) is the ability to escape a local optimum. More advantages include robustness and ability to adapt to changing environments. The major disadvantage of the EANN approach is that it is computationally expensive, as the evolutionary approach is usually slow. To overcome this slow convergence of the evolutionary approach, hybrid techniques were used to speed up convergence by augmenting evolutionary algorithms with a local search technique (i.e. memetic approach), such as BP [7].

3. Accuracy and sensitivity

To evaluate a classifier, the machine learning community has traditionally used the correct classification rate or accuracy to measure its default performance. However, the pitfalls of using accuracy have been pointed out by several authors [8]. Actually, it is enough to simply realize that accuracy cannot capture all the different behavioral aspects found in two different classifiers. Even in the simplest case where there are only two classes, accuracy states a one-dimensional ordering where you find two different types of errors. We consider traditionally-used accuracy C and the minimum of the sensitivities of all classes S , that is, the lowest percentage of examples correctly predicted as belonging to each class with respect to the total number of examples in the corresponding class. The sensitivity versus accuracy pair (S, C) expresses two features associated with a classifier: global performance C and the rate of the worst classified class S . The (S, C) pair tries to find an intermediate point between scalar measures and multidimensional ones based on misclassification rates, trying to evaluate two features of a classifier: global

performance in the whole dataset and the performance in each class.

One point in (S, C) space dominates another if it is above and to the right, i.e. it has more accuracy and greater sensitivity. Let us consider a Q -class classification problem. Let C and S be respectively the accuracy and the sensitivity associated with a classifier g , then $S \leq C \leq 1 - (1 - S)p^*$, where p^* the minimum of the estimated prior probabilities is. Therefore, each classifier will be represented as a point in the shaded region in Figure 1. Several points in (S, C) space are important to note. The lower left point $(0,0)$ represents the worst classifier and the optimum classifier is located at the $(1,1)$ point. Furthermore, the points on the vertical axis correspond to classifiers that are not able to predict any point in a concrete class correctly. Note that it is possible to find among them classifiers with a high level of accuracy, particularly in problems with small p^* .

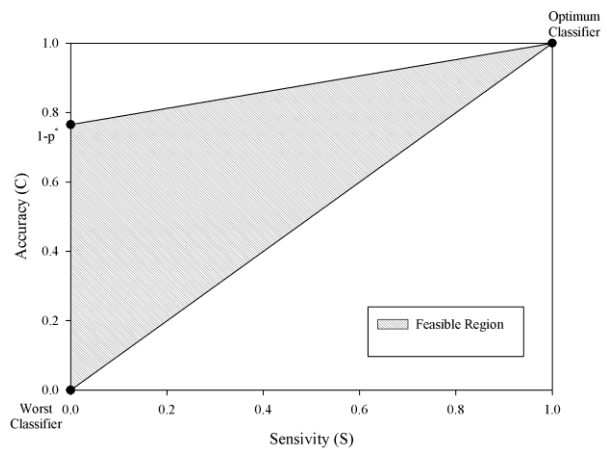


Figure 1. Feasible region in the two dimensional (S,C) space of a concrete classification problem.

4. Multi-objective optimization in classification problems

General multiobjective optimization problem (MOP) solution methods range from linear objective function aggregation to Pareto-based techniques. In an attempt to stochastically solve problems of this generic class in an acceptable timeframe, specific multiobjective evolutionary algorithms (MOEAs) were initially developed in the mid-eighties for application to the MOP domain being efficient in the evaluation of the Pareto-optimal set in difficult multiobjective

optimization problems. Several MOEA have been suggested that are capable to deal with a population of points to define an approximation to the Pareto set with a single run. There are already a number of good reviews on MOEA methods [9].

During the last few years, new methods called Memetic Algorithms (MAs) have been developed in order to improve the EAs using local optimization algorithms [10]. Some of the most important works in the literature about MOEAs and local optimizers used to speed up the convergence are [11-15].

5. The MPENSGA2 algorithm

In this section we consider an MOEA with a local search, called MPENSGA2, that tries to move the classifier population towards the optimum classifier located at the (1,1) point in the (S,C) space. We consider standard feed forward MLP neural networks with one input layer with independent variables, one hidden layer and one lineal output layer, interpreting the outputs of neurons on the output layer from a probability point of view which considers the softmax activation function given by the following expression:

$$g_l(\mathbf{x}, \boldsymbol{\theta}_l) = \frac{\exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}{\sum_{l=1}^J \exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}, l = 1, 2, \dots, J$$

where J is the number of classes in the problem, $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$ the output of the j neuron for pattern \mathbf{x} (lineal output) and $g_l(\mathbf{x}, \boldsymbol{\theta}_l)$ the probability of pattern \mathbf{x} belonging to class j . Taking this consideration into account, it can be seen that the class predicted by the neuron net corresponds to the neuron on the output layer whose output value is the greatest. The optimum rule $C(\mathbf{x})$ is the following:

$$C(\mathbf{x}) = \hat{l}, \text{ where } \hat{l} = \arg \max_l g_l(\mathbf{x}, \hat{\boldsymbol{\theta}}), \text{ for } l = 1, 2, \dots, J$$

One of the fitness functions used in this research to evaluate a classification model is the function of cross-entropy error and is given by the following expression for J classes:

$$l(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \sum_{l=1}^J y_n^{(l)} \log g_l(x_n, \boldsymbol{\theta}_l) = \frac{1}{N} \sum_{n=1}^N \left[-\sum_{l=1}^J y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log \sum_{l=1}^J \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \right]$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J)$. The advantage of using the error function $l(\boldsymbol{\theta})$ instead of $(1-CCR)$ is that it is a continuous function, which allows training to converge towards more optimal solutions and that the evolutionary algorithm to converge more slowly. Then,

the first fitness measure to maximise is a strictly decreasing transformation of the entropy error $l(\boldsymbol{\theta})$

given by $A(g) = \frac{1}{1+l(\boldsymbol{\theta})}$, where g is a sigmoidal

basis function model given by the multievaluated function $g(\mathbf{x}, \boldsymbol{\theta}) = (g_1(\mathbf{x}, \boldsymbol{\theta}_1), \dots, g_J(\mathbf{x}, \boldsymbol{\theta}_J))$.

The second objective to maximice is the sensitivity S of the classifier as the minimum value of the sensitivities for each class $S = \min \{S_i; i = 1, \dots, Q\}$.

The algorithm evolves architectures and connection weights simultaneously, each individual being a fully specified ANN. The neural networks are represented using an object-oriented approach and the algorithm deals directly with the ANN phenotype. Each connection is specified by a binary value indicating whether the connection exists and a real value representing its weight. The crossover operator is not considered due to its potential disadvantages in evolving artificial networks [16], this object-oriented representation does not assume a fixed order among between the different hidden nodes. With these features, the algorithms fall into the class of evolutionary programming.

The MOEA proposed is NSGA2[17], adding the necessary mutators to obtain new individuals in the evolutionary process, and the local search algorithm is the *improved Rprop—IRprop+*[18]. The local search algorithm is applied when we combine parent and offspring population in NSGA2. Then only the individuals of the first pareto front of this combined population are optimized by *iRprop+*, reducing the computational cost considerably. *iRprop+* can be seen as a kind of life-time learning (the first objective only) within a generation. After learning, the fitness of each individual with regard to the approximation error is updated. In addition, the weights modified during life-time learning are encoded back to the chromosome, which is known as the Lamarkian type of inheritance. Life-time learning occurs in the generations 2/7, 4/7 y 6/7 of total generations.

Mutators used in this work are divided into strutral mutators: add/delete neurons, add/delete connections, and parametric mutators: in this case a new parametric mutation that involves the alteration of all weights of the network by adding a Gaussian noise, where the variance of the Gauss distribution follows a geometric decline (for details see [19,20]).

6. Determining the probability of growth for *L. Monocytogenes*

Listeria monocytogenes have been a serious problem concerning food industries due to their ubiquity in the natural environment [21] and the specific growth conditions of the pathogen that lead to its high prevalence in different kinds of food products. One impetus for this research has been the problem of listeriosis, and different strategies have been proposed to limit levels of contamination at the time of consumption to less than 100 CFU/g (European Commission, [22]).

6.1. Experimental design

A fractional factorial design was followed in order to find out the growth limits of *L. monocytogenes*. Data were collected at citric acid, CA, and ascorbic acid, AA, concentrations between 0 and 0.4 % (w/v) at intervals of 0.05 %, at 4, 7, 10, 15 and 30°C and pH levels of 4.5, 5, 5.5 and 6. 539 different conditions were tested with 8 replicates per condition. This dataset was divided using a holdout cross-validation procedure, because in this kind of problem is difficult to justify the use of a k-fold cross-validation procedure. Thus 305 conditions were chosen for the model (training), and 234 for validation (generalization). Among the different conditions tested, there were 240 no-growth cases and 299 growth cases. To determine which data belong to model training and model validation, the conditions of organic acids used at the same level of temperature and pH were selected alternatively, as shown in Table 1. In [23] can be seen the growth medium preparation, the inoculation procedure and the growth/no-growth evaluation.

Table 1. Experimental design followed at the same level of temperature and pH

CA (%)	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
AA (%)	0	○	◆	○	◆	○	◆	○	◆
0.05	◆	○	◆	○	◆	○	◆	○	◆
0.1	○	◆	○	◆	○	◆	○	◆	○
0.15	◆	○	◆	○	◆	○	◆	○	◆
0.2	○	◆	○	◆	○	◆	○	◆	○
0.25	◆	○	◆	○	◆	○	◆	○	◆
0.3	○	◆	○	◆	○	◆	○	◆	○
0.35	◆	○	◆	○	◆	○	◆	○	◆
0.4	○	◆	○	◆	○	◆	○	◆	○

CA = Citric Acid; AA = Ascorbic Acid; ○ Training data ◆ Generalization data

To start processing data, each of the input and output variables were scaled in the ranks [0.1, 0.9] and [1, 2] respectively. The new scaled variables were named T^* , pH^* , CA^* and AA^* , for the input variables and $G^*=G+1$ for the output variable.

As our procedures are stochastic, the MPENSGA2 is run 30 times and presents the values of the average and the standard deviation obtained. The process for obtaining these values is as follows: Once the pareto front is calculated, the extreme values were chosen. First, (in training) the best individual belonging to the pareto front on Entropy (EI) is selected, after choosing the best individual in terms of sensitivity (SI). Once this is done, the values of CCR and sensitivity are obtained testing the individuals for EI and SI. Therefore we will have an individual $EI_{testing} = (C_{testing}, S_{testing})$ and an individual $SI_{testing} = (C_{testing}, S_{testing})$. This is repeated 30 times and then the average and standard deviation obtained from individuals is estimated, $\overline{EI}_{testing} = (\overline{C}_{testing}, \overline{S}_{testing})$, $\overline{SI}_{testing} = (\overline{C}_{testing}, \overline{S}_{testing})$. The first expression is the average obtained taking entropy into account as their primary objective, and the second taking sensitivity into account as their primary objective. So, the opposite ends of the pareto front are taken in each of the executions. Hence, the first procedure is called MPNSGA2E (Entropy) and the second MPNSGA2S (Sensitivity). In Figure 2 shows graphically the process followed.

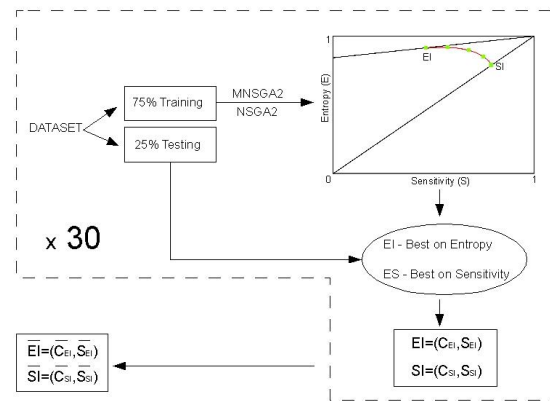


Figure 2. MPNSGA2E and MPNSGA2S procedure

Classifiers performance in several classification methodologies in the generalization set is presented in Table 2. The performance measures used are the correct classification rate and the sensitivity. The Pareto front obtained for *L. Monocytogenes* in training

and the values for CCR and Sensitivity in testing are shown in Figure 4 and Figure 5.

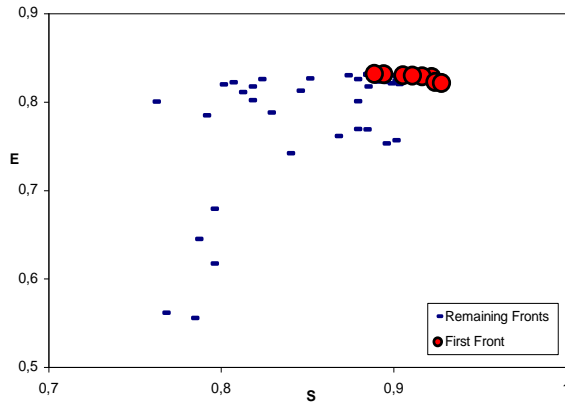


Figure 4. Pareto front for E-S in training for L. Monocytogenes

Of the methodologies tested, the model obtained using NBTree correctly predicted 87.40 % of the cases for the generalization set, with a sensitivity of 83.33 % whereas the worst model using the CART methodology obtained a C% of 79.25% with a sensitivity of 70.66%.

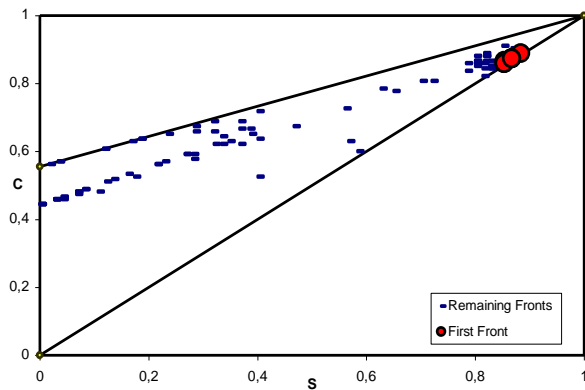


Figure 5. C-S in testing for L. Monocytogenes

In Table 2 presents the values of the average and the standard deviation of both measures; these results show that the best methodology is the MPENSGA2 (HNSGA2E) with an average value of 85.95% for C and of 84.53% for S, the second best result being on C and the first on S.

These results are in line with those obtained by Hajmeer and Basheer [24], who carried out hybrid approaches that integrate ANNs and statistical Bayesian conditional probability estimation, or the use of probabilistic neural networks in comparison to linear and non-linear logistic regression models. They observed that these new approaches outperformed linear and non-linear logistic regression models in terms of both classification accuracy and ease.

Table 2. Classification table obtained for the growth limits of L. monocytogenes.

Methodology	C(%)	S(%)
Mlogisitc	82.96±0.00	78.33±0.00
Slogistic	81.48±0.00	76.66±0.00
C45	85.92±0.00	78.33±0.00
NBTree	87.40±0.00	83.33±0.00
CART	79.25±0.00	70.66±0.00
SVM	80.74±0.00	73.33±0.00
HNSGA2E	85.95±1.57	84.53±1.93
HNSGA2S	85.60±1.86	82.81±2.91

C(%) = % of Correct classified rate; S(%) = Sensitivity

7. Conclusions

In this paper we study the improvement of the generalization ability of neural classifiers with multiple learning objectives. The inclusion of the two-objective (S,C) approach reveals a new point of view for dealing with classification problems.

It can be noted that the sensitivity obtained by the HNSGA2E model fits the data observed better than that obtained with other methodologies, and produced greater classification accuracy in generalization data (exception for NBTree). In conclusion, the use of the HNSGA2E model to determine growth probability under a set of conditions could constitute a valuable alternative method for mathematical modelling.

ACKNOWLEDGMENT

This work has been partially subsidized by TIN2005-08386 C05-02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) and FEDER funds. The research of J.C. Fernández and P.A. Gutiérrez has been backed respectively by the FPI and the FPU Predoctoral Programs (Spanish Ministry of Education and Science).

8. References

- [1] T.A. McMeekin, K. Presser, D.A. Ratkowsky, T. Ross, M. Salter, and S. Tienungoon. Quantifying the hurdle concept by modelling the bacterial growth/ no growth interface. *International Journal of Food Microbiology* 55, pp. 93-98, (2000).
- [2] S. Gidlant. *Neural-Network Learning and Expert Systems*. MIT Press, Cambridge, MA, 1993.

- [3] R. Parekh, J. Yang, and V. Honavar. Constructive neural-network learning algorithms for pattern classification. *IEEE Transaction on Neural Networks* 11, pp. 436-450, (2000).
- [4] R. Reed. Pruning algorithms. A survey. *IEEE Transactions on Neural Networks* 4, pp. 740-747, (1993).
- [5] A.J.F.v. Rooij, L.C. Jain, and R.P. Johnson. Neural Networks Training Using Genetic Algorithms. *World Scientific, Series in Machine Perception and Artificial Intelligence* 26(1996).
- [6] X. Yao, and Y. Liu. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks* 8(3), pp. 694-713, (1997).
- [7] W. Yan, Z.Z. Z, and R. Hu. *Hybrid genetic/BP algorithm and its application for radar target classification*, in: Proceedings of the 1997 IEEE National Aerospace and Electronics Conference, NAECON, USA, pp. 981-984, 1997.
- [8] F. Provost, and T. Fawcett. *Robust classification system for imprecise environments*, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp. 706-713, 1998.
- [9] C.A. Coello, G.B. Lamont, and D.A.V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
- [10] C. Cotta, and P. Moscato, A Gentle Introduction To Memetic Algorithms. in: F. Glover, and G.A. Kochenberger, (Eds.), *Handbook On Metaheuristics*, Kluwer Academics, 2001.
- [11] H. Abbass. Speeding Up Backpropagation Using Multiobjective Evolutionary Algorithms. *Neural Computation* 15, pp. 2705-2726, (2003).
- [12] Y. Jin, T. Okabe, and B. Sendhoff. *Neural Network Regularization and Ensembling Using Multi-Objective Evolutionary Algorithms*, in: Proceedings of 2004 Congress on Evolutionary and Ensembling Using Multi-Objective Evolutionary Algorithms, Portland, pp. 1-8, 2004.
- [13] Y. Jin, B. Sendhoff, and E. Körner. Simultaneous Generation of Accurate and Interpretable Neural Network Classifiers. *Studies in Computational Intelligence (SCI)* 16, pp. 291-312, (2006).
- [14] S. Wiegand, and C. Igel. Evolutionary Multi-Objective Optimization of Neural Networks for Face Detection. *International Journal of Computation Intelligence and Applications* 4(3), pp. 237-253, (2004).
- [15] S. Roth, A. Gepperth, and C. Igel, (Eds.), *Multi-objective neural network optimization for visual object detection*, Springer-Verlag, 2006.
- [16] P.J. Angeline, G.M. Saunders, and J.B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks* 5(1), pp. 54-65, (1994).
- [17] K. Deb, A. Pratab, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA2. *IEEE Transactions on Evolutionary Computation* 6(2), pp. 182-197, (2002).
- [18] C. Igel, and M. Hüsken. *Improving the Rprop Learning Algorithm*, in: Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), Berlin, pp. 115-121, 2000.
- [19] C. Hervás, and F.J. Martínez-Estudillo. Logistic regression using covariates obtained by product-unit neural network models. *Pattern Recognition* 40, pp. 52-64, (2007).
- [20] A. Martinez-Estudillo, F. Martinez-Estudillo, C. Hervas-Martinez, and N. Garcia-Pedrajas. Evolutionary product unit based neural networks for regression. *Neural Networks* 19(4), pp. 477-486, (2006).
- [21] C.A. Hwang, and M.L. Tamplin. The influence of mayonnaise pH and storage temperature on the growth of *Listeria monocytogenes* in seafood salad. *International Journal of Food Microbiology* 102(277-285)2005).
- [22] E. Commission, Opinion of the scientific committee on veterinary measures relating to public health on *Listeria monocytogenes*, <http://www.europa.eu.int/comm/food/fs/sc/scv/out25>, 1999.
- [23] A. Valero, C. Hervás, R.M. García-Gimeno, and G. Zurera. Product unit neural network models for predicting the growth limits of *Listeria monocytogenes*. *Food Microbiology* 24, pp. 452-464, (2007).
- [24] M.N. Hajmeer, and I.A. Basheer. A hybrid Bayesian-neural network approach for probabilistic modeling of bacterial growth/no-growth interface. *International Journal of Food Microbiology* 82, pp. 233-243, (2003).