

NEURAL FUZZY NETWORK MODEL WITH EVOLUTIONARY LEARNING ALGORITHM FOR MYCOLOGICAL STUDY OF FOODBORNE FUNGI

WEN-HSIEN HO¹, JINN-TSONG TSAI² AND HUE-YU WANG^{3,*}

¹Department of Healthcare Administration and Medical Informatics
Kaohsiung Medical University
No. 100, Shi-Chuan 1st Road, Kaohsiung 807, Taiwan
whho@kmu.edu.tw

²Department of Computer Science
National Pingtung University of Education
No. 4-18, Ming Shen Road, Pingtung 900, Taiwan
jttsai@mail.npue.edu.tw

³Department of Pharmacy
Chi-Mei Medical Center
No. 901, Chung Hwa Road, Yong Kang, Tainan 701, Taiwan

*Corresponding author: cmh5500@mail.chimei.org.tw

Received April 2011; revised August 2011

ABSTRACT. *This study developed a neural fuzzy network (NFN) model with evolutionary learning algorithm for use in the field of food mycology for predicting growth in foodborne fungi. The evolutionary learning algorithm in the proposed model is a hybrid Taguchi-genetic algorithm (HTGA) that simultaneously finds the optimal antecedent and consequent parameters by directly minimizing root-mean-squared error (RMSE), which is a key performance criterion. The minimum RMSE is then used to optimize the number of fuzzy rules for the NFN. Experimental results show that the proposed HTGA-based NFN model with eight fuzzy rules outperforms recently reported neural networks in terms of accuracy in predicting the maximum specific growth rate of foodborne *Monascus ruber*.*

Keywords: Fungal growth, Food mycology, Neural fuzzy network, Genetic algorithm

1. Introduction. Advances in prediction modeling now enable rapid and cost-effective assessment of microbial growth. In the food manufacturing industry, many applications of microbial growth prediction models include product development, risk assessment of food-borne illness, and education [1]. Recently, these models have been used to assess shelf life in food products by forecasting the growth of micro-organisms that cause spoilage. Fungal spoilage of food commodities causes significant economic losses. Although industrial standards have greatly improved in recent years, food spoilage by fungi remains a major concern for both food producers and regulatory agencies. In recent years, newly developed food manufacturing, processing and packaging techniques have increased the importance of characterizing foodborne fungi [2]. Fungi can adversely affect the organoleptic value of food commodities due to the appearance of visible mycelium and off-flavour development. In addition to diminishing the nutritional value of many foods, they are a public health risk because they can produce toxic metabolites that are potentially carcinogenic [3]. Improving food quality and safety requires appropriate tools for predicting fungal growth. In recent years, many studies in predictive microbiology have analyzed foodborne pathogens, but few have developed predictive models of growth in filamentous fungi [4]. The few reported predictive models for food mycology include those developed

by Panagou and Kodogiannis [5], one of which combined neural network with multilayer perceptron (PKNN + MLP) and one of which combined neural network with radial basis functions (PKNN + RBF), to model the relationship between growth parameters (temperature, pH and water activity) and the desired output (maximum specific growth rate) of *Monascus ruber*. The models were compared with the conventional polynomial model in terms of accuracy in predicting the growth of *Monascus ruber*. When using the full data set, accuracies with root mean square error (RMSE) were 0.167, 0.110 and 0.092 in the polynomial model, the PKNN + MLP model, and the PKNN + RBF model, respectively. Thus, the proposed PKNN models outperformed the conventional polynomial model in terms of prediction accuracy. Their key parameters for predicting the maximum specific growth rate of *Monascus ruber* were, in order of sensitivity, temperature, water activity, and pH. A recent literature review, however, shows that the PKNN models are the only reported use of neural networks to predict the growth of foodborne *Monascus ruber*.

Although PKNN models have demonstrated good accuracy, the current study shows that a neural fuzzy network (NFN) model is even more accurate for predicting the growth of *Monascus ruber*. Because they combine the semantic transparency of rule-based fuzzy systems with the learning capability of neural networks, NFNs have become a popular research topic in recent years [6-20]. The main advantage of NFNs is their resolution of the black box nature of the neural network paradigm as the connectionist structure of an NFN essentially defines the IF-THEN rules. However, parameter training remains problematic. For example, an unresolved problem in NFN design is simultaneously finding the optimal antecedent and consequent parameters by directly minimizing the RMSE performance criterion. One solution is genetic algorithm (GA), which has proven to be a powerful technique for training system parameters and for finding a global solution while optimizing the overall structure by minimizing the error function [21].

This study proposes a novel NFN with hybrid Taguchi-genetic algorithm (HTGA-based NFN) that improves accuracy in predicting *Monascus ruber* growth by simultaneously optimizing the antecedent and consequent parameters and by directly minimizing the RMSE performance criterion. The HTGA is applied because it reportedly outperforms conventional genetic algorithms [22,23]. The minimum RMSE is then used to optimize the number of fuzzy rules for the HTGA-based NFN. After the experimental results are given, the model is compared with the PKNN models in terms of prediction accuracy.

The present study therefore developed a novel HTGA-based NFN and compared its performance with PKNN + MLP and PKNN + RBF approaches in terms of accuracy in predicting the growth of this fungus [5]. Specifically, the objectives were (i) to develop an intelligent methodology for using HTGA-based NFNs to predict the combined effects of temperature, pH and water activity on the maximum specific growth rate of *Monascus ruber*, (ii) to compare prediction accuracy between the proposed HTGA-based NFNs and the PKNN + MLP and PKNN + RBF approaches, and (iii) to evaluate the number of fuzzy rules required when using HTGA-based NFN to predict fungal growth. To our knowledge, this is the first application of an NFN for predicting fungal growth. Compared with conventional methodologies used in predictive mycology, the proposed approach is simpler yet equally effective.

This paper is organized as follows. Section 2 describes the proposed HTGA for solving the parameter adjustment problem in NFN models of *Monascus ruber* growth. For comparison purposes, Section 3 then describes the results of a performance test of the proposed HTGA-based NFN model using the same data sample used in Panagou and Kodogiannis [5]. Finally, Section 4 concludes the study.

2. Materials and Methods. In Panagou and Kodogiannis [5], 73 data sets were obtained in experimental studies of the maximum specific growth rate μ_{\max} of *Monascus ruber*. Of these, the current study selected 60 data sets (Table 1) to train the proposed HTGA-based NFN, where $R = \ln(\mu_{\max})$; after training, the remaining 13 data sets (Table 2) were then used to verify the accuracy of the predicted value y . Three growth parameters that have a major impact on fungal growth include temperature (x_1), pH value (x_2) and water activity (x_3) [24]. Therefore, as in the work of Panagou and Kodogiannis [5], these three growth parameters were used as input variables for the NFN, and the predicted value y was selected as the output variable of the NFN. Figure 1 shows the nodes in the NFN architecture, including those in the layers for input, membership function, rules, consequent, and output (5 layers). An NFN consists of a set of fuzzy IF-THEN rules that describe the input-output mapping relationship of the network. In the Takagi-Sugeno-Kang model, the antecedents of fuzzy rules partition the input space into linguistic term sets whereas the consequent constituent functions as a linear combination of input variables plus a constant value [13].

The n fuzzy rules can be given in the following form:

$$R^j : \quad \text{IF } x_1 \text{ is } M_{1j} \text{ AND } x_2 \text{ is } M_{2j} \text{ AND } x_3 \text{ is } M_{3j}, \tag{1}$$

$$\quad \text{THEN } y = a_{0j} + a_{1j}x_1 + a_{2j}x_2 + a_{3j}x_3,$$

where R^j denotes the j th implication; n is the number of fuzzy rules; x_i is the input variable; y is the output variable; M_{ij} is the linguistic term of the precondition part; a_{kj} is the coefficient of consequent part; $i = 1, 2, 3$; $j = 1, 2, \dots, n$; and $k = 0, 1, 2, 3$. Depending on their measurement ranges, the values for variables x_1, x_2, x_3 and y are scaled in the rank $[0, 1]$.

The resulting output of NFN inferred from Equation (1) is represented as

$$y = \sum_{j=1}^n h_j (a_{0j} + a_{1j}x_1 + a_{2j}x_2 + a_{3j}x_3), \tag{2}$$

in which $h_j = w_j / \sum_{j=1}^n w_j$, $w_j = \prod_{i=1}^3 M_{ij}$, M_{ij} are the grades of Gaussian membership of x_i in the antecedent fuzzy sets M_{ij} ($i = 1, 2, 3$ and $j = 1, 2, \dots, n$), and

$$M_{ij} = \exp\left(-\frac{[x_i - m_{ij}]^2}{\sigma_{ij}^2}\right), \tag{3}$$

in which m_{ij} and σ_{ij} are the center and the width, respectively, of the Gaussian membership function of the i th implication of the j th input variable x_i .

After training, antecedent parameters (m_{ij} and σ_{ij}) and consequent parameters (a_{kj}) ($i = 1, 2, 3$, $j = 1, 2, \dots, n$, and $k = 0, 1, 2, 3$) can be optimized by directly minimizing the following RMSE performance criterion:

$$J = \sqrt{\frac{\sum (R - y)^2}{N}}, \tag{4}$$

where N denotes the number of training data sets.

Equation (4) shows that the value of the performance criterion J actually depends on the set $\{m_{ij}, \sigma_{ij}, a_{kj}\}$ ($i = 1, 2, 3$, $j = 1, 2, \dots, n$, and $k = 0, 1, 2, 3$). Therefore,

$$J = f(m_{ij}, \sigma_{ij}, a_{kj}). \tag{5}$$

TABLE 1. Experimental results for 60 training data sets [5]

Temperature ($^{\circ}\text{C}$) (x_1)	pH (x_2)	water activity (x_3)	$\ln(\mu_{\max})$ (R)
20	3.5	0.173	1.3483
20	4.0	0.173	1.1731
20	4.5	0.173	1.1346
20	5.0	0.173	0.9247
25	4.0	0.173	1.9864
25	4.5	0.173	2.1961
25	5.0	0.173	2.1372
30	3.5	0.173	2.4632
30	4.0	0.173	2.7704
30	5.0	0.173	2.5810
35	3.5	0.173	2.5864
35	4.5	0.173	2.8113
40	3.5	0.173	2.3165
40	4.0	0.173	2.1162
40	4.5	0.173	2.3842
20	3.5	0.207	1.2921
20	4.0	0.207	1.1752
20	4.5	0.207	0.8983
20	5.0	0.207	0.6296
25	3.5	0.207	2.0487
25	4.5	0.207	1.7376
25	5.0	0.207	1.5829
30	3.5	0.207	2.4361
30	4.0	0.207	2.3366
30	5.0	0.207	2.0407
35	3.5	0.207	2.4896
35	4.0	0.207	2.4532
35	4.5	0.207	2.3217
35	5.0	0.207	2.0677
40	3.5	0.207	2.4623
40	4.5	0.207	2.3635
40	5.0	0.207	2.1958
20	3.5	0.238	0.9383
20	4.0	0.238	0.7944
20	4.5	0.238	0.5462
25	3.5	0.238	1.7595
25	4.5	0.238	1.3797
25	5.0	0.238	1.3886
30	3.5	0.238	2.1991
30	4.0	0.238	2.1593
30	4.5	0.238	1.9997
30	5.0	0.238	1.7117
35	3.5	0.238	2.2086
35	4.0	0.238	2.1058
35	4.5	0.238	2.0274
40	3.5	0.238	2.0452
40	4.0	0.238	2.2202
40	4.5	0.238	2.1272
25	3.5	0.250	1.2073
25	4.0	0.250	0.9179
30	3.5	0.250	1.7978
30	4.0	0.250	1.7615
30	4.5	0.250	1.7459
30	5.0	0.250	1.6355
35	4.0	0.250	1.9248
35	4.5	0.250	1.8596
35	5.0	0.250	1.7971
40	3.5	0.250	1.3653
40	4.0	0.250	1.7113
40	4.5	0.250	1.7120

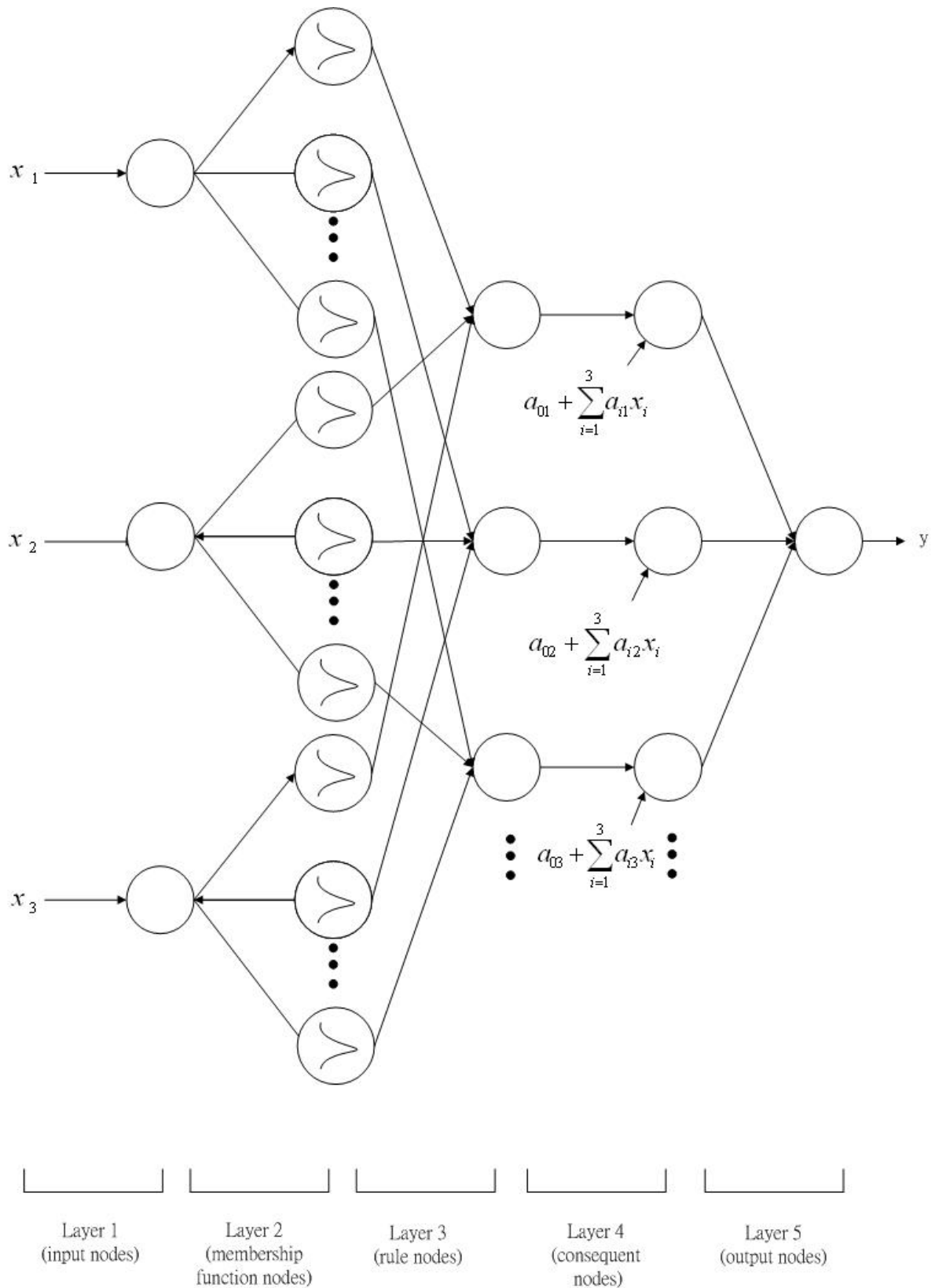


FIGURE 1. Three-input/output NFN architecture

TABLE 2. Experimental results for 13 test data sets [5]

Temperature ($^{\circ}\text{C}$) (x_1)	pH (x_2)	water activity (x_3)	$\ln(\mu_{\max})$ (R)
25	3.5	0.173	2.1051
25	4.0	0.207	1.8690
25	4.0	0.238	1.6340
25	4.5	0.250	0.5328
30	4.5	0.173	2.7427
30	4.5	0.207	2.3003
35	4.0	0.173	2.5109
35	5.0	0.173	2.6405
35	3.5	0.250	1.9633
40	5.0	0.173	2.1102
40	4.0	0.207	2.4161
40	5.0	0.238	1.9147
40	5.0	0.250	1.5193

This optimization problem is equivalent to the following:

$$\text{minimize } J = f(m_{ij}, \sigma_{ij}, a_{kj}). \quad (6)$$

The HTGA, which is described in detail below, can be used to search for the optimal solution for the optimization problem in Equation (6), where Equation (6) is a nonlinear function with continuous variables. The HTGA combines the genetic algorithm [25-31] with the Taguchi method [32-34]. In the HTGA, the Taguchi method is performed between the crossover and mutation operations of a TGA. The genetic algorithms are enhanced by using two major Taguchi tools (signal-to-noise ratio and orthogonal arrays) to optimize gene selection in crossover operations by incorporating the systematic reasoning capability of the Taguchi method (for a detailed description of the Taguchi method, see [32,33]; for a detailed discussion of HTGA, see [22,23,35,36]). The detailed steps of the HTGA are as follows:

Detailed Steps: HTGA

Step 1: Set parameters.

Input: population size M , crossover rate p_c , mutation rate p_m , and number of generations.

Output: the set of $V = \{m_{ij}, \sigma_{ij}, a_{kj}\}$ and the value of J in Equation (5).

Step 2: Initialize by using J in Equation (5), which is the fitness function defined for the HTGA. Calculate the fitness values of the initial population, where randomly generated chromosomes in the initial population are given in the form $V = \{m_{ij}, \sigma_{ij}, a_{kj}\}$ for the problem in Equation (5).

Step 3: Perform selection operation by roulette wheel approach.

Step 4: Perform Crossover operation. The probability of the crossover is determined by the crossover rate p_c .

Step 5: Select a suitable two-level orthogonal array $L_{\gamma}(2^{\gamma-1})$ for the matrix experiments, where γ denotes the number of experimental runs, and $\gamma - 1$ is the number of columns in the orthogonal array.

Step 6: Execute the matrix experiments using randomly selected chromosome pairs.

Step 7: Calculate the fitness values of the γ experiments in the orthogonal array $L_{\gamma}(2^{\gamma-1})$ by using Equation (5).

Step 8: Calculate the effects of the various factors.

Step 9: Generate one optimal chromosome based on the results from Step 8.

- Step 10:** Repeat Steps 6 through 9 until the expected number $M \times p_c$ is met.
- Step 11:** Perform Taguchi method to generate the population.
- Step 12:** Perform mutation operation. The probability of the mutation is determined by the mutation rate p_m .
- Step 13:** Generate the offspring population.
- Step 14:** Sort the fitness values in increasing order among the parents and offspring populations.
- Step 15:** Compare feasible M chromosomes and select the best chromosomes for the constraint for use as the parents of the next generation.
- Step 16:** If the specified stopping criterion is met, go to Step 17. Otherwise, repeat Steps 3-16.
- Step 17:** Compare RMSE performance criteria J to determine if the stopping condition has been met. If so, go to Step 18. Otherwise, repeat Steps 2-17.
- Step 18:** Display the optimal chromosome and the optimal fitness value.

3. Results and Discussion. The proposed HTGA-based NFN model was directly compared with the PKNN models. In the evolution environment for the HTGA-based NFN, population size was 200, crossover rate was 0.9, mutation rate was 0.1, and generation number was 500. The training parameters for the HTGA-based NFN were $-1 \leq m_{ij} \leq 1$, $0.1 \leq \sigma_{ij} \leq 1$, and $-1 \leq a_{kj} \leq 1$ ($i = 1, 2, 3$, $j = 1, 2, \dots, n$, and $k = 0, 1, 2, 3$). For the training data, test data and full data sets, Table 3 shows that, in both the HTGA-based and conventional GA-based NFNs, RMSE performance criterion J obtained two to twelve fuzzy rules. The data in Table 3 reveal the following characteristics of the HTGA-based NFN: (i) the proposed HTGA obtains a smaller performance criterion J compared with conventional GA, (ii) when using the full data set, eight or nine fuzzy rules give the lowest RMSE (0.0865 or 0.0863, respectively); that is, eight or nine fuzzy rules is optimal for an HTGA-based NFN; (iii) when using the test data set, eight fuzzy rules gives a lower RMSE compared to nine fuzzy rules (RMSE = 0.1614 versus 0.1735, respectively); that is, eight fuzzy rules gives better prediction results compared to nine fuzzy rules; (iv) when using the training, test and full data sets, two fuzzy rules yields the worst prediction results; therefore, the smaller the number of fuzzy rules, the less accurate the NFN; (v) when using training data, twelve fuzzy rules gives the best RMSE (0.0432); when using test data, however, twelve fuzzy rules gives the worst RMSE (0.1959); therefore, an overly large number of fuzzy rules may result in a model over-fitted to recognizing specific subjects in the training data rather than learning general predictive values.

Models with two and eight fuzzy rules were also compared in terms of effects on computational performance. Figure 2 shows the convergence results for training RMSE performance criterion J in Equation (6) with two and eight fuzzy rules when using HTGA. Clearly, eight fuzzy rules outperforms two fuzzy rules in terms of the RMSE performance criterion. After training, entering three growth parameters of any one of the 13 test data sets into the HTGA-based NFN obtains the predicted value of y . In Figures 3 and 4, comparison of predictive performance when using different numbers of fuzzy rules in HTGA-based NFN shows that, compared to two fuzzy rules, eight fuzzy rules generally obtains a better model fit. In Figures 5 and 6, comparison of residual value spreads for two and eight fuzzy rules in HTGA-based NFN shows that residuals are distributed symmetrically around 0; that is, residuals do not tend to be either positive or negative. Compared to the HTGA-based NFN with two fuzzy rules, the one with eight fuzzy rules obtained a narrower spread of residual values and thus a better prediction performance. Therefore, the proposed HTGA-based NFN model with eight fuzzy rules obtains a smaller

TABLE 3. Comparison of training, test and full data for RMSE performance criterion J between HTGA-based NFNs and in conventional GA-based NFNs for different numbers of fuzzy rules

Number of fuzzy rules	RMSE performance criterion J					
	HTGA-based NFN			Conventional GA-based NFN		
	Training data	Test data	Full data	Training data	Test data	Full data
2	0.1641	0.2145	0.1741	0.1733	0.2599	0.1916
3	0.1324	0.2014	0.1471	0.1579	0.2937	0.1893
4	0.1181	0.2204	0.1418	0.1306	0.2604	0.1615
5	0.0699	0.2161	0.1111	0.1146	0.2500	0.1481
6	0.0590	0.2019	0.1006	0.1087	0.2413	0.1417
7	0.0609	0.1758	0.0925	0.1191	0.2455	0.1496
8	0.0588	0.1614	0.0865	0.1364	0.2248	0.1558
9	0.0503	0.1735	0.0863	0.1127	0.2327	0.1417
10	0.0517	0.1755	0.0876	0.1149	0.2882	0.1601
11	0.0594	0.1868	0.0954	0.1224	0.2556	0.1548
12	0.0432	0.1959	0.0915	0.1093	0.2606	0.1480

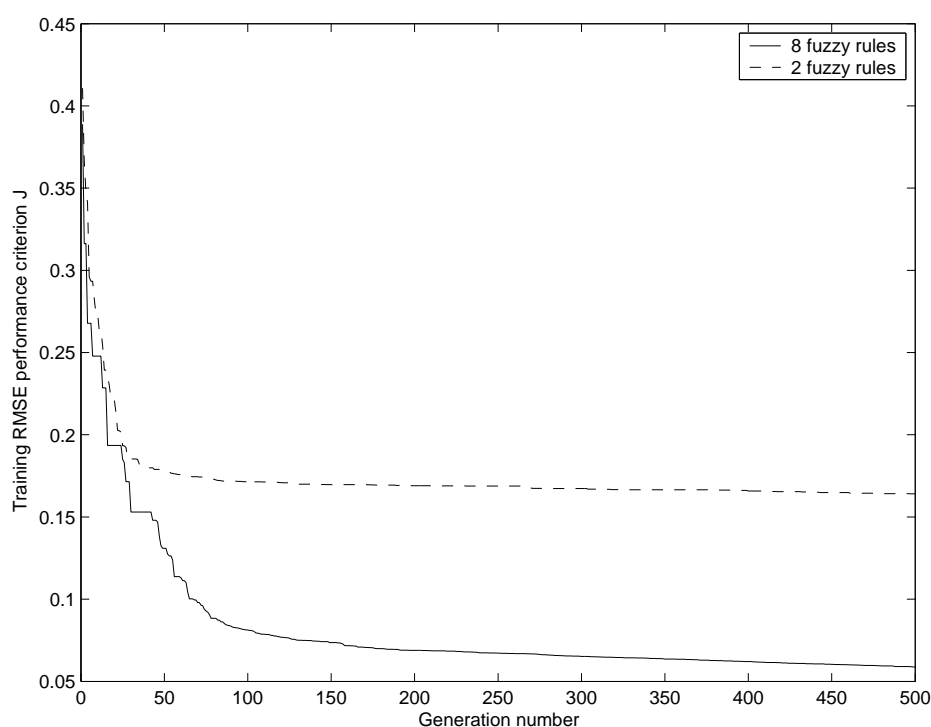


FIGURE 2. Convergence results for training RMSE performance criterion J when using HTGA

RMSE when using the optimal antecedent and consequent parameters shown in Tables 4 and 5, respectively.

Table 6 compares the RMSE in prediction obtained by the proposed HTGA-based NFN with eight fuzzy rules (0.0588 in training data, 0.1614 in test data, and 0.0865 in full data) with that obtained by PKNN + MLP (0.088 in training data, 0.179 in test data, and 0.110 in full data) and that obtained by PKNN + RBF (0.063 in training data, 0.167 in test data, and 0.092 in full data). The comparison shows that the predictive accuracy of the

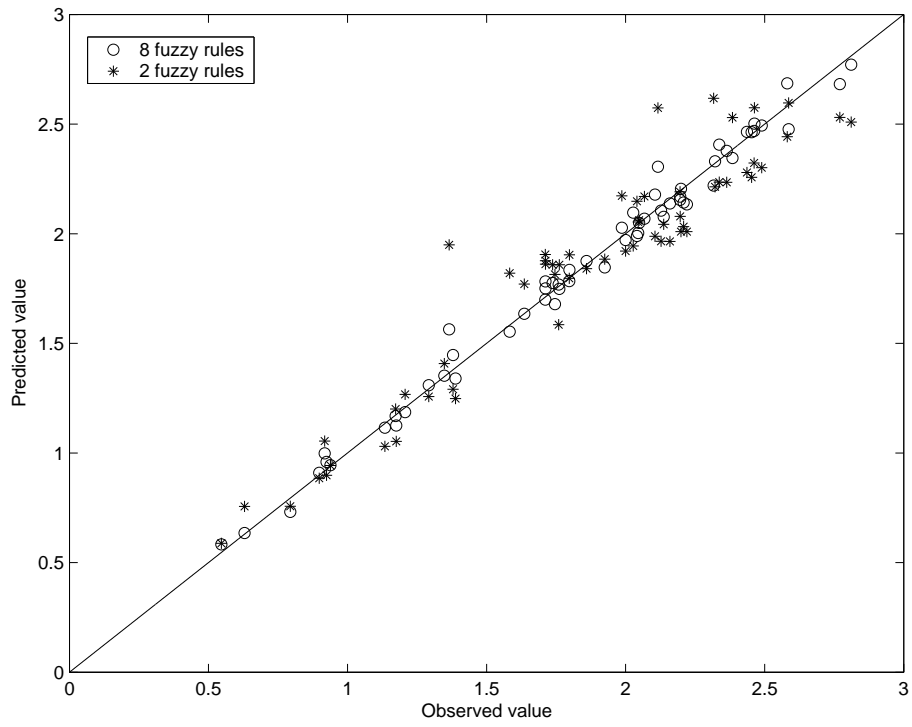


FIGURE 3. Comparison of predicted values versus observed values when using HTGA-based NFN in training data set

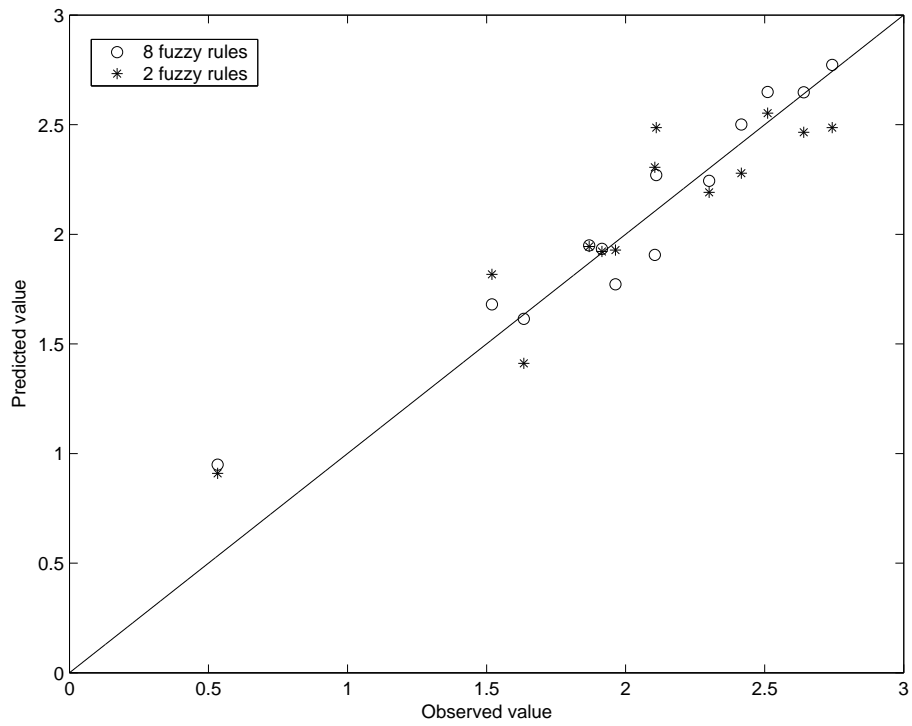


FIGURE 4. Comparison of predicted values versus observed values when using HTGA-based NFN in test data set

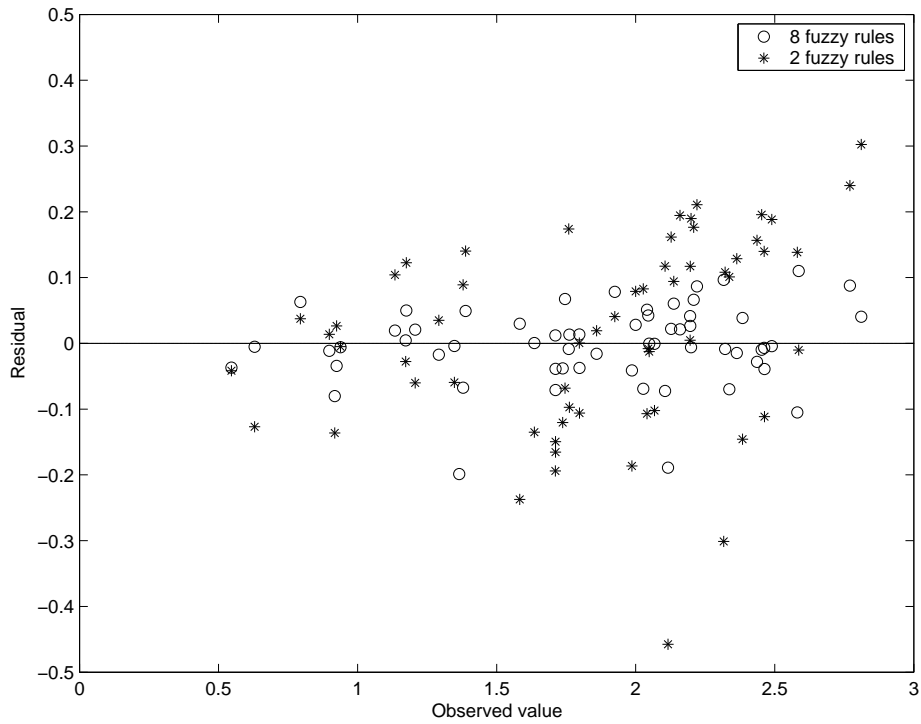


FIGURE 5. Comparison of residual values: predicted versus observed values when using HTGA-based NFN in training data set

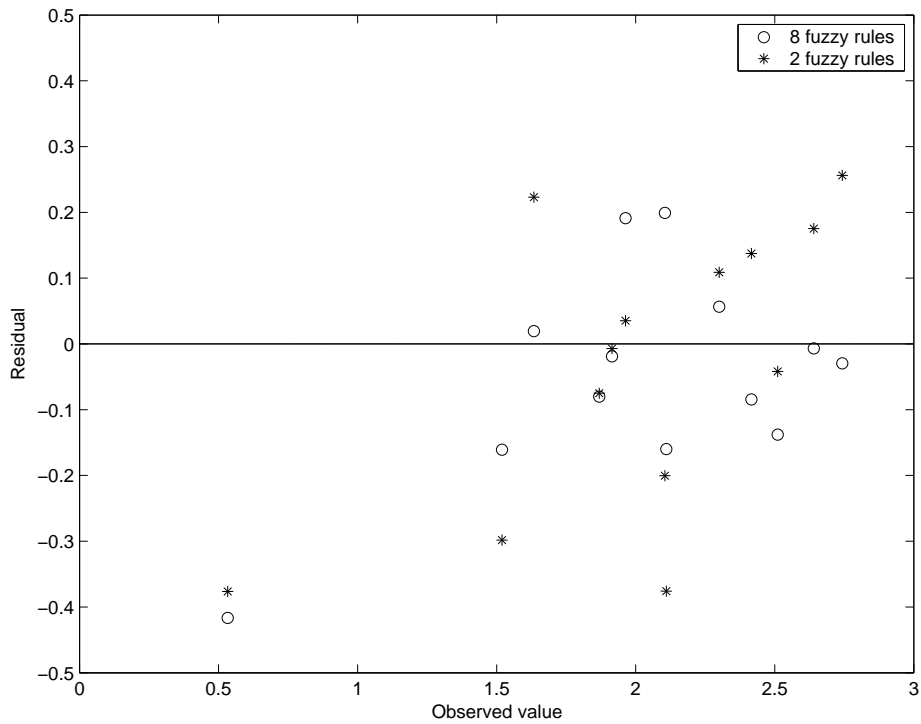


FIGURE 6. Comparison of residual values: predicted versus observed values when using HTGA-based NFN in test dataset

TABLE 4. Optical antecedent parameters

Input variables					
x_1		x_2		x_3	
$m_{11} = 0.4259$	$\sigma_{11} = 0.2410$	$m_{21} = 0.6952$	$\sigma_{21} = 0.2775$	$m_{31} = -0.0961$	$\sigma_{31} = 0.7232$
$m_{12} = -0.0690$	$\sigma_{12} = 1.0000$	$m_{22} = 0.4113$	$\sigma_{22} = 0.5045$	$m_{32} = 0.3905$	$\sigma_{32} = 0.2794$
$m_{13} = 1.0000$	$\sigma_{13} = 0.9535$	$m_{23} = -0.2436$	$\sigma_{23} = 0.9848$	$m_{33} = -0.3781$	$\sigma_{33} = 0.8667$
$m_{14} = -0.2085$	$\sigma_{14} = 0.3209$	$m_{24} = 0.0873$	$\sigma_{24} = 0.2271$	$m_{34} = 0.0196$	$\sigma_{34} = 0.2299$
$m_{15} = -0.7439$	$\sigma_{15} = 0.6437$	$m_{25} = 0.3558$	$\sigma_{25} = 0.4032$	$m_{35} = 1.0000$	$\sigma_{35} = 0.1492$
$m_{16} = -0.7941$	$\sigma_{16} = 0.4831$	$m_{26} = 0.0517$	$\sigma_{26} = 1.0000$	$m_{36} = -1.0000$	$\sigma_{36} = 0.7746$
$m_{17} = 0.4459$	$\sigma_{17} = 0.2211$	$m_{27} = 0.0560$	$\sigma_{27} = 0.5922$	$m_{37} = 0.1417$	$\sigma_{37} = 0.4423$
$m_{18} = 0.6448$	$\sigma_{18} = 0.2202$	$m_{28} = 0.4948$	$\sigma_{28} = 0.9734$	$m_{38} = 0.5699$	$\sigma_{38} = 0.1627$

TABLE 5. Optical consequent parameters

Fuzzy rule				
Rule 1	$a_{01} = 0.9892$	$a_{11} = -0.0757$	$a_{21} = -0.5409$	$a_{31} = 0.6146$
Rule 2	$a_{02} = 0.9240$	$a_{12} = -0.5543$	$a_{22} = 0.5352$	$a_{32} = 0.1945$
Rule 3	$a_{03} = 0.2114$	$a_{13} = 0.0855$	$a_{23} = -0.3640$	$a_{33} = 0.4179$
Rule 4	$a_{04} = -1.0000$	$a_{14} = -0.4426$	$a_{24} = -1.0000$	$a_{34} = 0.3598$
Rule 5	$a_{05} = 0.5064$	$a_{15} = -0.2614$	$a_{25} = -0.6628$	$a_{35} = 0.2937$
Rule 6	$a_{06} = 1.0000$	$a_{16} = -0.4451$	$a_{26} = -1.0000$	$a_{36} = -0.0984$
Rule 7	$a_{07} = 0.8182$	$a_{17} = 0.5060$	$a_{27} = 0.2489$	$a_{37} = 0.8766$
Rule 8	$a_{08} = 0.6462$	$a_{18} = -0.2004$	$a_{28} = -0.3709$	$a_{38} = 0.3783$

TABLE 6. Comparison of results

Performance index	Model	Data set		
		Training	Test	Full
RMSE	PKNN + MLP	0.088	0.179	0.110
	PKNN + RBF	0.063	0.167	0.092
	HTGA-based NFN with eight fuzzy rules	0.0588	0.1614	0.0865

proposed HTGA-based NFN approach with eight fuzzy rules is higher than that of both PKNNs.

4. Conclusions. The proposed HTGA-based NFN accurately predicts the maximum specific growth rate of *Monascus ruber* in terms of three growth parameters (temperature, pH and water activity). When applied in the NFN, the HTGA simultaneously optimized antecedent and consequent parameters by directly minimizing the RMSE performance criterion. Comparisons of experimental results for full data showed that the RMSE of 0.0865 obtained by the proposed HTGA-based NFN approach with eight fuzzy rules was superior to the RMSEs of 0.110 and 0.092 obtained by PKNN + MLP and PKNN + RBF, respectively.

Acknowledgment. This work was in part supported by a grant from the Chi-Mei Medical Center and Kaohsiung University Research Foundation (100CM-KMU-06), and the National Science Council, Taiwan, under grant number NSC 99-2320-B-037-026-MY2. We also thank the editor and the reviewers for their suggestions and comments.

REFERENCES

- [1] T. Ross and T. A. McMeekin, Review paper: Predictive microbiology, *Int. J. of Food Microbiology*, vol.23, pp.241-264, 1994.
- [2] A. M. Gibson and A. D. Hocking, Advances in the predictive modelling of fungal growth in food, *Trends in Food Science and Technology*, vol.8, pp.353-358, 1997.
- [3] P. A. Murphy, S. Hendrich, C. Landgren and C. M. Bryant, Food mycotoxins: An update, *J. of Food Science*, vol.71, pp.51-65, 2006.
- [4] P. Dantigny, A. Guilmart and M. Bensoussan, Basis of predictive mycology, *Int. J. of Food Microbiology*, vol.100, pp.187-196, 2005.
- [5] E. Z. Panagou and V. S. Kodogiannis, Application of neural networks as a non-linear modelling technique in food mycology, *Expert Systems with Applications*, vol.36, pp.121-131, 2009.
- [6] J. S. R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. on Systems Man and Cybernetics*, vol.23, pp.665-685, 1993.
- [7] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, MIT Press, Cambridge, MA, 1996.
- [8] C.-J. Lin and C.-T. Lin, An ART-based fuzzy adaptive learning control network, *IEEE Trans. on Fuzzy Systems*, vol.5, no.4, pp.477-496, 1997.
- [9] C.-F. Juang and C.-T. Lin, An online self-constructing neural fuzzy inference network and its applications, *IEEE Trans. on Fuzzy Systems*, vol.6, no.1, pp.12-31, 1998.
- [10] J. Zhang and A. J. Morris, Recurrent neuro-fuzzy networks for nonlinear process modeling, *IEEE Trans. on Neural Networks*, vol.10, pp.313-326, 1999.
- [11] A. Canuto, G. Howells and M. Fairhurst, An investigation of the effects of variable vigilance within the RePART neuro-fuzzy network, *J. of Intelligent and Robotic Systems*, vol.29, pp.317-334, 2000.
- [12] S. Osowski and T. H. Linh, ECG beat recognition using fuzzy hybrid neural network, *IEEE Trans. on Biomedical Engineering*, vol.48, pp.1265-1271, 2001.
- [13] G. Castellano, A. M. Fanelli and C. Mencar, A neuro-fuzzy network to generate human-understandable knowledge from data, *Cognitive Systems Research*, vol.3, pp.125-144, 2002.
- [14] J.-H. Chiang and P.-Y. Hao, A new kernel-based fuzzy clustering approach: Support vector clustering with cell growing, *IEEE Trans. on Fuzzy Systems*, vol.11, pp.518-527, 2003.
- [15] C.-J. Lin and C.-C. Chin, Prediction and identification using wavelet-based recurrent fuzzy neural networks, *IEEE Trans. on Systems Man and Cybernetics, Part B*, vol.34, no.5, pp.2144-2154, 2004.
- [16] G. Leng, T. M. McGinnity and G. Prasad, An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, *Fuzzy Sets and Systems*, vol.150, pp.211-243, 2005.
- [17] R.-J. Wai and P.-C. Chen, Robust neural-fuzzy-network control for robot manipulator including actuator dynamics, *IEEE Trans. on Industrial Electronics*, vol.53, pp.1328-1349, 2006.
- [18] C.-J. Lin and Y.-C. Hsu, Reinforcement hybrid evolutionary learning for recurrent wavelet-based neurofuzzy systems, *IEEE Trans. on Fuzzy Systems*, vol.15, pp.729-745, 2007.
- [19] C.-J. Lin, C.-H. Chen and C.-T. Lin, Efficient self-evolving evolutionary learning for neurofuzzy inference systems, *IEEE Trans. on Fuzzy Systems*, vol.16, pp.1467-1490, 2008.
- [20] J.-F. Chang, L.-Y. Wei and C.-H. Cheng, ANFIS-based adaptive expectation model for forecasting stock index, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1949-1958, 2009.
- [21] C.-J. Lin and Y.-J. Xu, A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications, *Fuzzy Sets and Systems*, vol.157, pp.1036-1056, 2006.
- [22] J.-T. Tsai, T.-K. Liu and J.-H. Chou, Hybrid Taguchi-genetic algorithm for global numerical optimization, *IEEE Trans. on Evolutionary Computation*, vol.8, no.4, pp.365-377, 2004.
- [23] J.-T. Tsai, J.-H. Chou and T.-K. Liu, Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm, *IEEE Trans. on Neural Networks*, vol.17, pp.69-80, 2006.
- [24] L. Rosso, J. R. Lobry, S. Bajard and J. P. Flandrois, Convenient model to describe the combined effects of temperature and pH on microbial growth, *Applied and Environmental Microbiology*, vol.61, pp.610-616, 1995.
- [25] T. Liu, C.-H. Chen and J.-H. Chou, Intelligent design of the automobile fixtures with genetic algorithms, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2533-2550, 2008.
- [26] C.-H. Hsieh, W.-H. Ho and J.-H. Chou, Optimal PID controllers design of PWM feedback time-varying systems using orthogonal-functions approach and genetic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.387-397, 2009.

- [27] C.-L. Wang, T.-P. Hong, G. Horng and W.-H. Wang, A GA-based key-management scheme in hierarchical wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4693-4702, 2009.
- [28] L. Yao and C.-C. Lin, On a genetic algorithm based scheduled fuzzy PID controller, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(B), pp.3593-3602, 2009.
- [29] W.-H. Ho, S.-H. Chen, T.-K. Liu and J.-H. Chou, Design of robust-optimal output feedback controllers for linear uncertain systems using LMI-based approach and genetic algorithm, *Information Sciences*, vol.180, pp.4529-4542, 2010.
- [30] F. Saadatjoo, V. Derhami and S. M. Karbassi, Stabilization of simultaneous linear multivariable systems while improving time-response using genetic algorithms, *International Journal of Innovative Computing, Information and Control*, vol.7, no.1, pp.151-160, 2011.
- [31] W.-H. Ho, Takagi-Sugeno fuzzy model of nonlinear HIV dynamics: Chebyshev-series approach integrated with genetic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.8, no.2, pp.1439-1452, 2012.
- [32] G. Taguchi, S. Chowdhury and S. Taguchi, *Robust Engineering*, McGraw-Hill, New York, 2000.
- [33] Y. Wu, *Taguchi Methods for Robust Design*, The American Society of Mechanical Engineers, New York, 2000.
- [34] W.-H. Ho, J.-H. Chou and C.-Y. Guo, Parameter identification of chaotic systems using improved differential evolution algorithm, *Nonlinear Dynamics*, vol.61, pp.29-41, 2010.
- [35] W.-H. Ho, J.-X. Chen, I.-N. Lee and H.-C. Su, An ANFIS-based model for predicting adequacy of vancomycin regimen using improved genetic algorithm, *Expert Systems with Applications*, vol.38, pp.13050-13056, 2011.
- [36] W.-H. Ho and C.-S. Chang, Genetic-algorithm-based artificial neural network modeling for platelet transfusion requirements on acute myeloblastic leukemia patients, *Expert Systems with Applications*, vol.38, pp.6319-6323, 2011.