韓國經營科學會誌
第23卷 第1號
1998年 3月

Robustness of Data Mining Tools under Varying Levels of Noise : Case Study in Predicting a Chaotic Process

Steven H. Kim* · Churlmin Lee* · Heungsik Oh*

Abstract

Many processes in the industrial realm exhibit stochastic and nonlinear behavior. Consequently, an intelligent system must be able to adapt to nonlinear production processes as well as probabilistic phenomena. In order for a knowledge based system to control a manufacturing process, an important capability is that of prediction: forecasting the future trajectory of a process as well as the consequences of the control action. This paper examines the robustness of data mining tools under varying levels of noise while predicting nonlinear processes, including chaotic behavior. The evaluated models include the perceptron neural network using backpropagation (BPN), the recurrent neural network (RNN) and case based reasoning (CBR). The concepts are crystallized through a case study in predicting a chaotic process in the presence of various patterns of noise.

1. Introduction

Recent years have witnessed a growing recognition of the need to develop intelligent manufacturing systems which can improve their performance through experience. Moreover, such systems should be competent in dealing not only with simple processes such as linear input-output functions, but stochastic and nonlinear behavior.

To this end, an intelligent manufacturing system may draw on techniques from disparate fields, involving knowledge in both explicit and implicit forms. An example of explicit knowledge is a set of rules or procedures, while an example of implicit know-how lies with neural networks.

* Graduate School of Mangnement, Korea Advanced Institute of Science and Techonlogy, Seoul, Korea.

In order for a knowledge based system to control a manufacturing process, an important capability is that of prediction: forecasting the future trajectory of a process as well as the consequences of the control action. This paper presents a study of robustness of data mining tools under varying levels of noise to predict nonlinear processes, including chaotic behavior. The evaluated models include perceptron neural networks using backpropagation (BPN), recurrent neural network (RNN) and case based reasoning (CBR). The concepts are crystallized through a case study in predicting a chaotic process.

MOTIVATION

Manufacturing systems operate in complex environments. The complexity arises from novelty, nonlinearities, and the multitude of interactions which arise when attempting to control various activities. In such a milieu, important process variables can remain unidentified. Even when they are identified, their interactions may remain uncertain. This complexity and the uncertainties which are often its derivatives limit the effectiveness of traditional control methods.

Fortunately, this situation can be remedied by an adaptive control methodology using knowledge integration [17]. The integration involves a judicious mixture of model based reasoning, information sharing, and case based reasoning. Over the past decade, a popular methodology for implementing adaptive systems has lain in the neural network. Despite its many advantages such as autonomous learning in specific contexts, the neural approach has its limitations. Among the limitations are the slow rates of learning and perhaps even more importantly, the implicit nature of the learned skill. More specifically, a neural network may yield the correct response to a query but it cannot explain the result or justify its "reasoning".

In contrast, the use of explicit knowledge allows for explanation and justification for the benefit of other entities, including an interested human observer. Examples of such high-level representation, also called the knowledge level, lies with declarative logic or production rules. A sophisticated learning system should provide for a fusion of both implicit and explicit methods of knowledge representation. In this way, it can build on the respective advantages of disparate techniques.

METHODOLOGY

An adaptive system should have the ability to accommodate a diversity of adaptive techniques among its component subsystems. In the area of production supervision, existing programs largely address the regulation of simple tasks. To illustrate, software packages for system regulation rely on classical methods of control: these methods model each component of the associated plant in simplified fashion, as linearizable or other elementary functions. In real systems, however, the components tend to be highly nonlinear: examples of nonlinearities are found in the step functions of static friction, the cycles of hysteresis, or the gaps of dead zones. Moreover, the plant is often poorly understood and therefore inadequately modeled.

All these limitations can be addressed by advanced methods of knowledge representation and manipulation. To illustrate, nonlinear plant dynamics can be modeled by arbitrary functions in the form of callable procedures. Moreover, poorly modeled plants can be handled by learning systems. By building on their experience, such adaptive systems can build enhanced models - whether explicit or implicit - of the plant and thereby improve system performance over time.

These capabilities can be further enhanced by incorporating complementary techniques. An example in this category lies in the use of predictive statistics. For instance, drift in the depth of cut of a milling machine can be estimated by regression on the data stream from previous operations. Any discernible drift can then be compensated for in the control system; or the endogenous plant could be reconfigured; or a human supervisor might be notified of an impending problem.

Neural Network. Neural networks are learning systems characterized by robustness and graceful degradation. The most common type of neural network and training procedure takes the form of backpropagation (BPN). A backpropagation neural network with standard connections responds to a given input pattern with exactly the same output pattern every time the input pattern is presented. A recurrent neural network (RNN) may respond to the same input pattern differently at different times, depending upon the patterns that have been presented as inputs in the past.

However, a key limitation of neural nets lies in the excessively long training periods. Thousands of trials are often required for satisfactory performance in various tasks. The time and effort required for training have hindered their widespread application to many practical domains.

Case Based Reasoning. A learning system should make increasingly useful decisions as it accumulates experience. This is the express goal of the work in case-based reasoning (CBR).

The CBR methodology can be effective even if the knowledge base is imperfect. Certain techniques of automated learning, such as explanation-based learning, work well only if a strong domain theory exists. In contrast, CBR can use many examples to overcome the gaps in a weak domain theory while still taking advantage of the domain theory [24]. CBR can also be used when the descriptions of the cases, as well as the domain theory, are incomplete [28]. A further advantage of CBR is the relative ease of combining techniques with other approaches. An example of such compatibility is a system which uses case reasoning to solve problems whenever possible; otherwise it resorts to heuristics to decompose a problem into a simpler one [21].

Perhaps the main limitation of CBR is its susceptibility to the misinterpretation of the knowledge in its case base. This is a perennial hazard in any field of endeavor, automated or otherwise. One way to address the problem in CBR is to encode deeper-level domain knowledge in addition to the surface features of various cases.

CASE STUDY

The utility of a learning approach to manufacturing supervision may be demonstrated through a case study in predicting a Henon process. The basic motivation and methodology behind the case study is encapsulated in [Figure 1].

The overall structure of the simulation experiment is shown in [Figure 2]. The inputs into the predictive system consists of two sources: a chaotic signal and a noise source. The chaotic signal takes the form of the Henon model shown in [Figure 3].

The noise process was generated in two steps, as indicated in [Figure 4]. Once the probability mass function was constructed, the data stream itself was generated through a Monte Carlo simulation using a pseudorandom number generator.

The pseudorandom generator is a computational process employing data structures of fixed size and therefore can assume only a finite number of potential values. Consequently, the generated sequence is itself a chaotic stream. However, since the inherent dimensionality of the sequence is high, the resulting process is statistically indistinguishable from pure noise such as that originating from natural sources in the physical world

The primary data streams consisted of the Henon process H_t, the noise process v_t and mixtures of the two. Each mixture consisted of a convex combination of the Henon and noise processes:

$$x_t = \lambda_t H_t + (1-\lambda_t) v_t$$

The convex weight λ_t was a function of time. As shown in [Figure 5], the three mixed modes consisted of a downward step function, a square function, and a tent function. Each of these 3 modes, plus the pure Henon and pure noise processes, constituted a total of 5 input data patterns.

Each of the 5 signal modes was digested by several learning methods, then predicted out-of-sample. One of the learning techniques was backpropagation (BPN), as illustrated in [Figure 6]. Another was the recurrent neural network (RNN) architecture, presented in [Figure 7]. The third was a case based reasoning (CBR) approach, as indicated in [Figure 8].

As shown in [Figure 5], a discontinuity occurred in the weight λ_t for all three mixing modes at period t = 1500. A series of exploratory charts are shown in Figure [9] through [13] in the vicinity of the discontinuity. The mixed signal modes displayed in Figures [10], [11] and [12] appear to change character to a certain degrees.

The test phase involved 200 forecasts from period 2800 to 3000. Out of these forecasts, the first 50 results are portrayed in [Figure 14] for the pure Henon process. The forecasts for this chaotic process, although generated out of sample, are remarkably accurate. However, the accuracy drops dramatically for each of the 4 other signal modes, as shown in Figures [15] through [18].

The difference in performance across signal modes is highlighted in [Figure 19], which presents the results according to the metric of mean absolute percent error (MAPE). As expected, the worst performance was due to pure noise, while the best accuracy resulted from the pure Henon process.

[Figure 20] compares results according to the hit rate (HR), or proportion of correct directional forecasts regarding a subsequent rise or fall in the value of a particular signal mode. The same qualitative inferences can be made here as for the differential results according to MAPE.

A more accurate set of results are listed in <Table 1> according to the metric of MAPE. <Table 2> presenteds a similar chart for the hit rate.

In <Table 3> an analysis of variance for the data behind <Table 1> indicates that the differences due to signal mode and to learning technique are both significant. Moreover, the interaction effects are also statistically significant.

<Table 4> presents a chi-square test for the hit rates from <Table 2>. According to the test,

the interaction effects are significant at level $p < 0.03$.

One interesting issue relates to the choice of a good architecture for the CBR model. [Figure 21] presents the metric of MAPE as function of the locality L for each signal mode, holding the input vector size fixed at $D = 2$. The results indicate that the optimal architecture depends in part on the particular signal mode to be predicted.

[Figure 22] shows a similar chart for $D = 4$. The qualitative inferences are the same as for $D = 2$ in the previous figure.

[Figure 23] presents the hit rate for CBR as a function of the locality L when the input vector size is fixed at $D = 2$. On the whole, the accuracy tends to rise with the size of the neighborhood. A similar chart for $D = 4$ in [Figure 24], however, reveals that accuracy depends both on the signal mode and the size of the locale.

〈Table 5〉 presents a set of pairwise comparisons across environmental scenarios for BPN. For instance, the first cell indicates that the MAPE for the pure Henon process using BPN is 0.011%, while that for the step mode is 0.113%. Further, the difference is significant at $p < 0.001$. The step mode is similar to the pure Henon process, except that half of the signal is comprised of noise after $t = 1500$. Consequently, a 50% dilution of the Henon process by noise results in a significant difference in forecasting accuracy. To take another instance from 〈Table 5〉, there is no significant difference in performance between the step and tent signal modes when BPN is used.

〈Table 6〉 presents similar data for BPN according to hit rates. Tables 〈7〉 and 〈8〉 enumerate the pairwise differences in environmental scenarios for RNN according to the criteria of MAPE and HR, respectively. A similar pair of charts is presented in Tables 〈9〉 and 〈10〉 for CBR.

The hit rate measures the accuracy of forecasts but ignores the mistakes. The proportions of Type I and Type II mistakes for the pure Henon process are listed for each learning technique in 〈Table 11〉. The results indicate that CBR dominates the other two techniques for the Henon signal. Similar charts in Tables 〈12〉 to 〈15〉 enumerate respectively the comparative performance in forecasting the four other signal modes.

〈Figure 25〉 offers a pictural representation of the Type I and Type II errors for the Henon signal. The chart highlights the fact that all techniques perform well for the pure Henon mode.

Figures 〈26〉 through 〈29〉 present the mistake charts due to the four other signal modes. Overall, the charts indicate the absence of a clear-cut winner in the context of classification mistakes.

CONCLUSION

The complexity inherent in a learning system for manufacturing automation can be addressed by the judicious use of a spectrum of advanced methodologies from data mining. A number of predictive methodologies were evaluated in the context of prediction for production supervision. The utility of these approaches was tested through a simulation model. The results support the feasibility of developing learning systems to support the prediction and control of general manufacturing processes, including chaotic behavior.

Future work will involve more sophisticated models of system behavior and predictive methods as well as their incorporation into control strategies. The longer-term goal is to develop a general knowledge-based system which can learn to supervise manufacturing processes regardless of the type of behavior exhibited by the production plant.

REFERENCES

[1] Aboutalib, A.O. "Embedded Knowledge-based Systems for Automatic Target Recognition." R.C. Harney, ed., *Sensor Fusion III*, Bellingham, WA: SPIE, Vol. 1036(1990), pp.205-219.

[2] Cameron, W.L., H. Fain J.C. Bezdeck. "An Intelligent System for AutonomousNavigation of Airborne Vehicles." P.S. Schenker, ed., *Sensor Fusion*, Bellinggham, WA: SPIE, Vol. 1003(1988), pp. 451-469.

[3] Chandra, D.V. S. "Multisensor Seeker for Medium Range Air-to-Air Missiles." R.C. Harney, ed., *Sensor Fusion III*, Bellingham, WA: SPIE, Vol. 1306(1990),pp.180-186.

[4] Croteau, A., D. Laurendeau, and J. Lessard. "Building a Space OccupancyModel for a power-line Maintenance Robot Using a Range Data Sensor." *Proc. Intel. Robots and Comp. Vision X: Algorithms and Techniques* (1991), Bellingham, WA:SPIE, Vol. 1607(1992), pp.720-725.

[5] Egilmez, K. and S. H. Kim. "A Logical Approach to Knowledge Based Control." *J. of Intelligent Manufacturing*, Vol. 1(1990), pp.59-76.

[6] Egilmez, K. and S. H .Kim. "Deployment of Robotic Agents in Uncertain Environments: Game Theoretic Rules and Simulation Studies." *Artif. Intel. for Engineering Design, Analysis and Manufacturing*, Vol. 6, No. 1(1992), pp.1-17.

[7] Forsyth, D., J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. "Invariant

Descriptors for 3-D Object Recognition and Pose." *IEEE Trans. Pattern Anal. and mach. Intel.*, Vol. 13 No. 10(1991), pp.971-991.

[8] Gentner, D. "Structure-Mapping: A Theoretical Framework for Analogy." *Cognitive Science*, Vol. 7, No. 2(1883), pp.155-170.

[9] Goldberg, K.Y. and B.A. Pearlmutter. "Using Backpropagation with Temporal Windows to Learn the Dynamics of the CMU Direct-Drive Arm II.". In D.S. Touretzky, ed., *Advances in Neural Information Processing Systems, I*, M. Kaufmann, San Maeto, CA, 1989, pp.356-363.

[10] Hall, R.P. "Computational Approaches to Analogical Reasoning: A Comparative Analysis." *Artificial Intelligence*, Vol. 39, No. 1(1989), pp.39-120.

[11] Holland, J.H. *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: Univ. of Michigan Press, 1975.

[12] Kedar-Cabelli, S. "Purpose-Directed Analogy." In *Programm of the Seventh Annual Conf. of the Cognitive Science Society*, Irvine, CA, 1985, pp.150-159.

[13] Kim, S. H. *Designing Intelligence.* New York: Oxford Univ. Press, 1990a.

[14] Kim, S. H. *Essence of Creativity.* New York: Oxford Univ. Press, 1990b.

[15] Kim, S. H. *Knowledge Systems through Prolog.* New York: Oxford Univ. Press, 1991.

[16] Kim, S. H. *Statistics and Decisions.* New York: Van Nostrand Reinhold, 1992.

[17] Kim, S. H. *"Learning Systems for Process Automation through Knowledge Integration." Proc. Second World Congress on Expert Systems*, Lisbon, Portugal, 1994a.

[18] Kim, S. H. *Learning and Coordination.* Dordrecht, Netherlands: kluwer, 1994.

[19] Kim, S. H. and M.B. Novick. "Using Clustering Techniques to Support Case Reasoning." *Int. J. of Computer Applications in Technology*, Vol. 6, No. 2/3(1993), pp.57-73.

[20] Lenat, D. "EURISKO: A Program that Learns New Heuristics and Design Concepts: The Nature of Heuristics, III: Program Design and Results." *Artificial Intelligence*, Vol. 21, No. 2(1983), pp. 61-98.

[21] Maher, M.L. "Machine Learning in Design Expert Systems." In J. Leibowitz, ed., *Expert Systems World Congress Proc.*, v. 1, Oxford, UK, Pergamon, 1991, pp.728-36.

[22] Mel, B.W. *Connectionist Robot Motion Planning. NY*: Academic, 1990.

[23] Pomerleau, D.A. "ALVINN: An Autonomous Land Vehicle in a Neural Network.". In D.S. Touretzky, ed., *Advances in Neural Information Processing Systems, I*, M. Kaufmann, San Maeto, CA, 1989, pp.305-313.

[24] Porter, B.W., E.R. Bareiss, and R.C. Holte, "Concept Learning and Heuristic Classification in Weak-Theory Domains." *Artificial Intelligence*, Vol. 45, No. 1/2(1990), pp.229-263.

[25] Qian, L. and J.S. Gero. "A Design Support System Using Analogy." In J.S. Gero, ed., *Artificial*

Intelligence in Design '92, Dordrecht, Netherlands: Kluwer, 1992, pp.795-816.

[26] Riekki, J., J. Roning, O. Silven, M. Pietikainen, and V. Koivunen. "Pick-and-Place Guidance Utilizing an Integrated Control Method and Structured Light Ranging." *Proc. Intel. Robots and Comp. Vision X: Algorithms and Techniques (1991)*, Bellingham, WA: SPIE, Vol. 1607(1992), pp.689-699.

[27] Shibazaki, H. and S.H. Kim, "Learning Systems for Manufacturing Automation: Integrating Explicit ad Implicit Knowledge." Presented at International Conference on the Manufacturing Science and Technology of the Future, Stockholm, Sweden, June 1989. Also in *Robotics and Computer-Integrated Manufacturing*, Vol. 9(1992).

[28] Sycara, K. and D. Navinchandra. "Index Transformation Techniques for Facilitating Creative Use of Multiple Cases." *Proc. 12th Int. Joint Conf. on AI*, Morgan Kaufman, Los Altos, CA, 1991, pp.347-352.

[29] Tecuci, G. and Y. Kodratoff. "Apprenticeship Learning in Imperfect Domain Theories." In Y. Kodratoff and R. S. Michalski, eds., *Machine Learning: An Artificial Intelligence Approach, v. III, San Maeto*, CA: M. Kaufmann, 1990, pp.514-551.

[30] Winston, P.H. "Learning New Principles from Precedents and Exercises." *Artificial Intelligence*, Vol. 19, No. 3(1982), pp.321-350

Purpose

◆ To investigate the robustness of learning methods when the underlying system changes character

Methodology

◆ Simulation of system events (e.g. service request, malfunctions, etc.) with chaotic interarrival times and stochastic noise

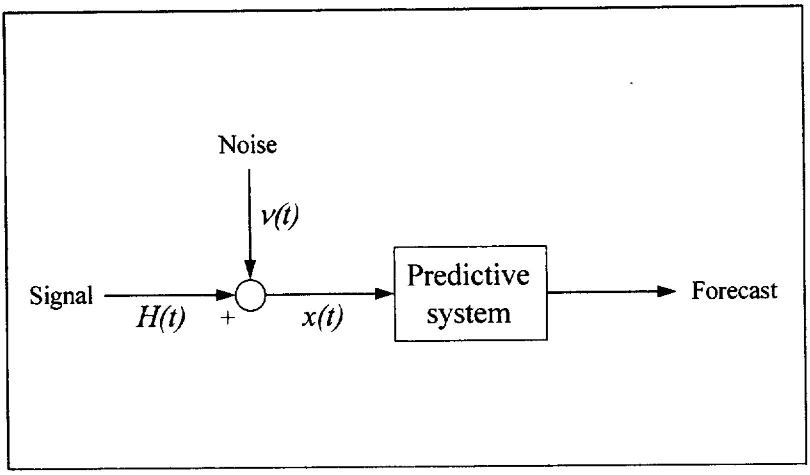
Types of Inputs

◆ Primary inputs
- Henon : H_t
- Noise : v_t

◆ Weighting factor : λ_t
- Observed time series is weighted sum of primary inputs :

$$x_t = \lambda_t H_t + (1-\lambda_t)\, v_t$$

[Figure 1] Highlights of the study.

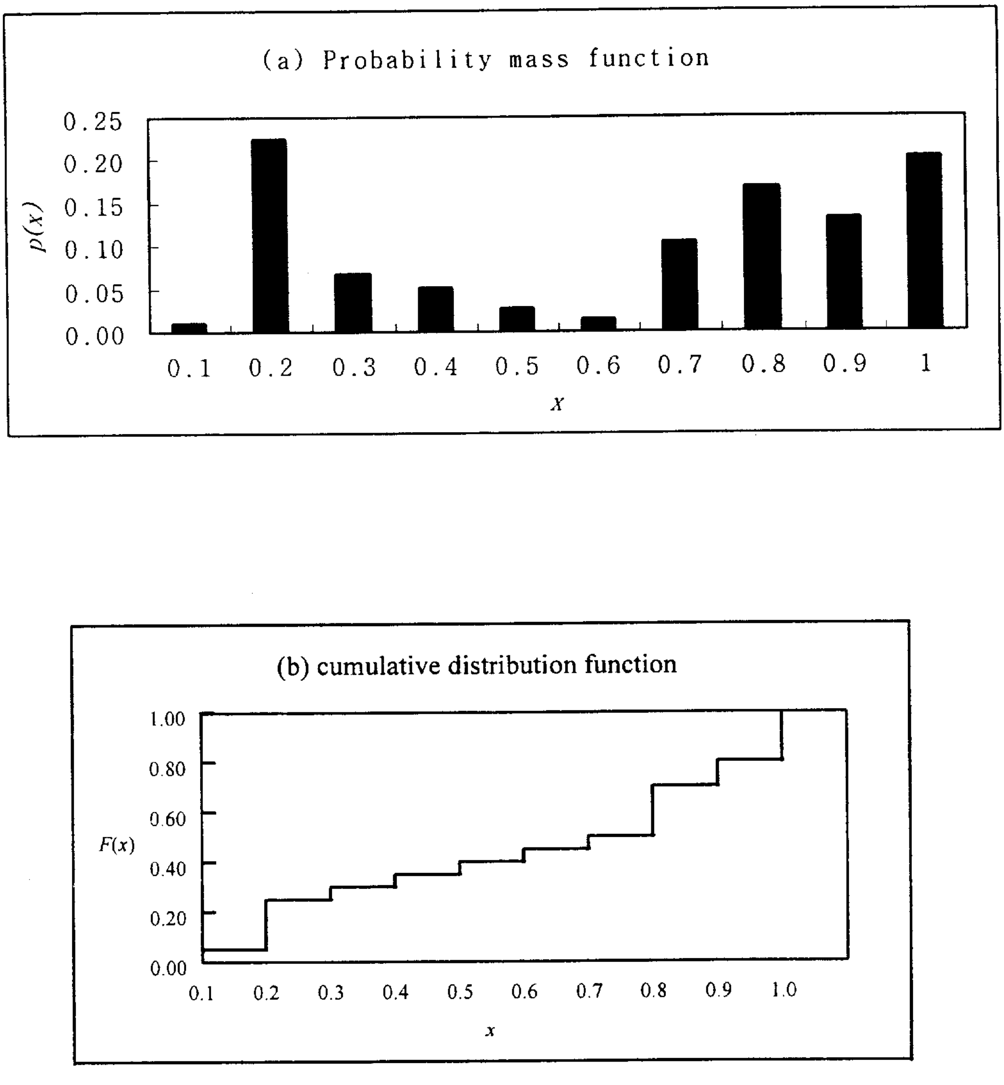


[Figure 2] Schenatic of the simulation model.

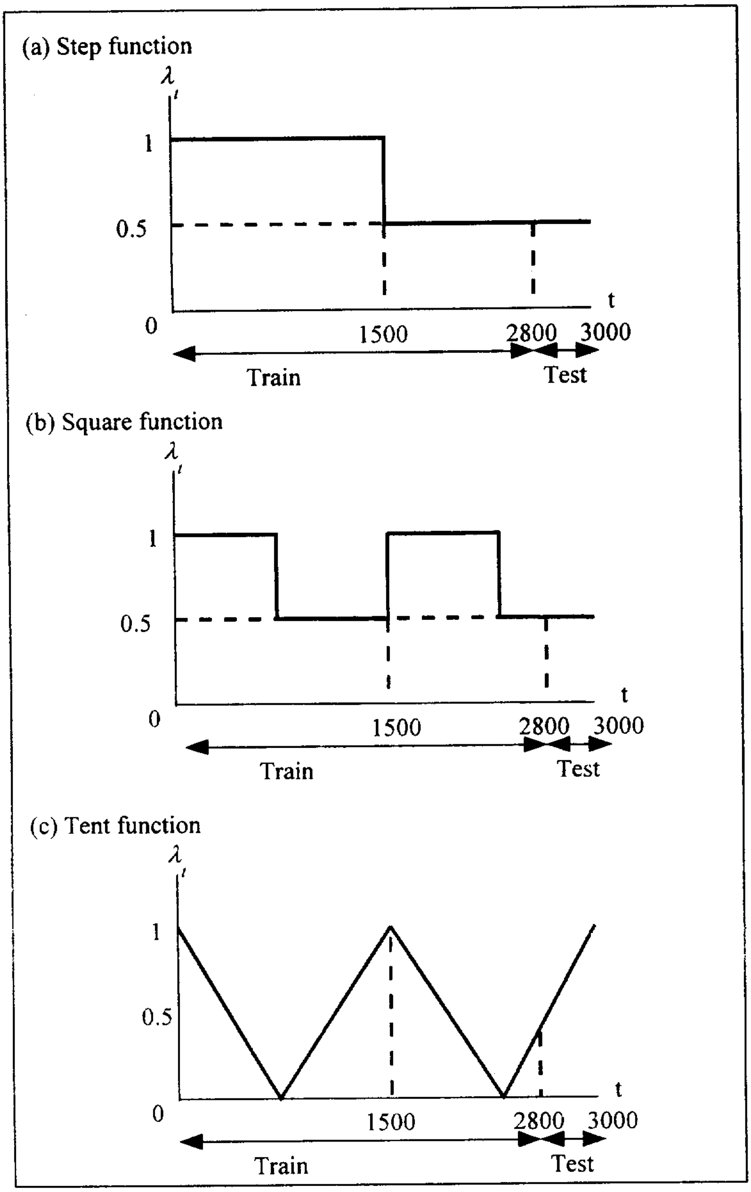
$$x(t+1) = 1 - ax(t)^2 + y(t)$$

$$y(t+1) = bx(t)$$

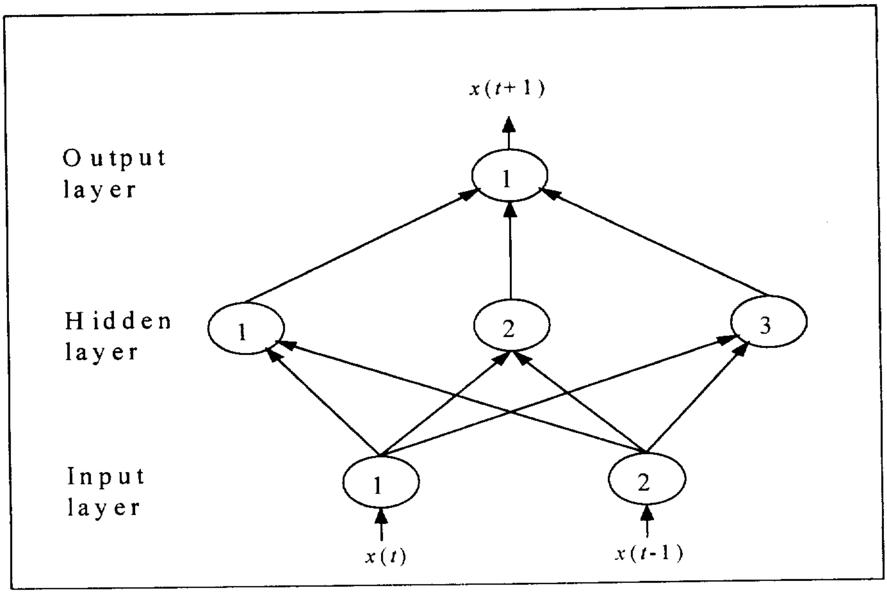
[Figure 3] The Henon model. The parameter values a = 1.4 and b = 0.3 result in chaotic behavior.



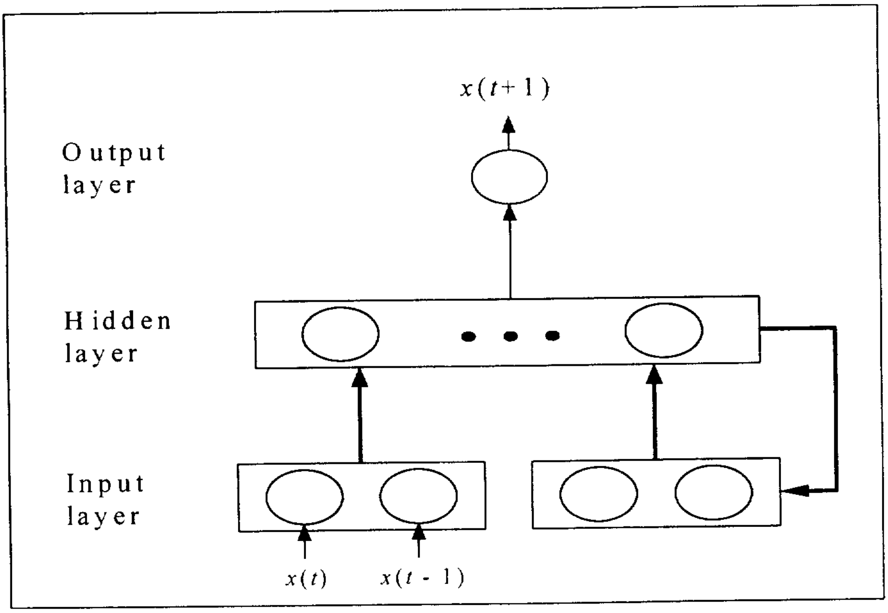
[Figure 4] Generation of random inputs representing noise. (a) The probability mass function was generated by procuring a sequence of numbers from a random number table, then dividing by their collective sum. (b) The cumulative distribution function resulted from summing the probability mass function. Monte Carlo simulation was used to obtain a value for F(x), from which the inverse $F^{-1}(x)$ yielded a value for the noise input.



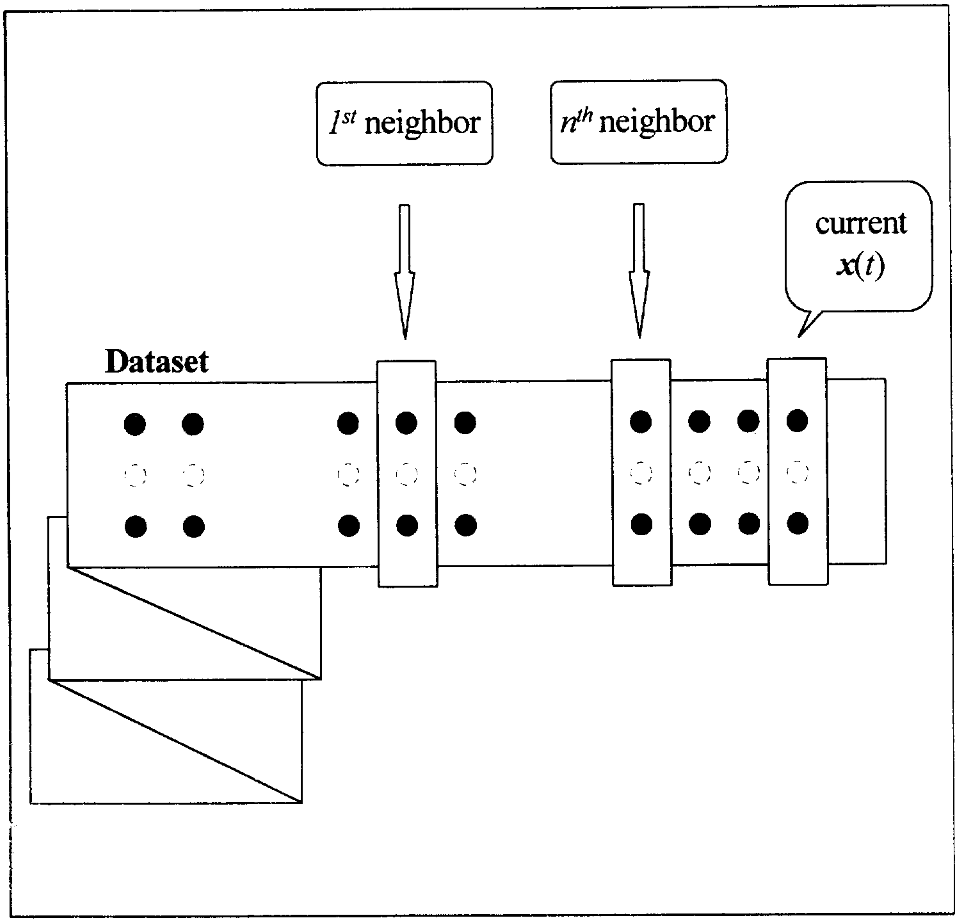
[Figure 5] Variation in weighting factor as a function of time. The weighting factor λ_t is used to generate a mixed input stream composed of primary and noise sources.



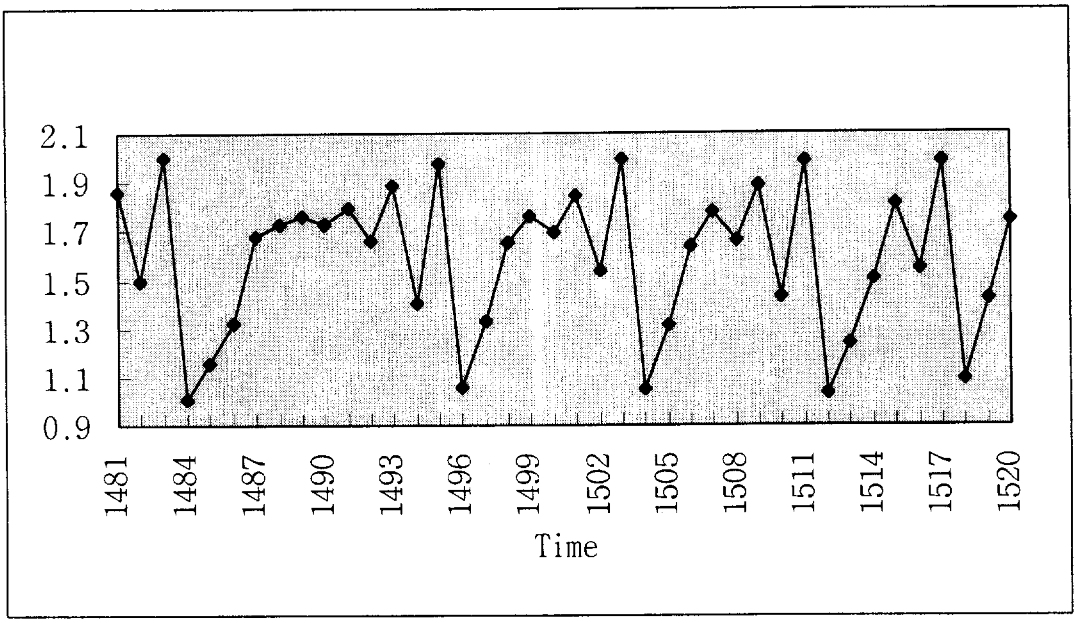
[Figure 6] Architecture of the backpropagation (BPN) neural network for the case of two inputs, 3 hidden nodes, and a single output cell.



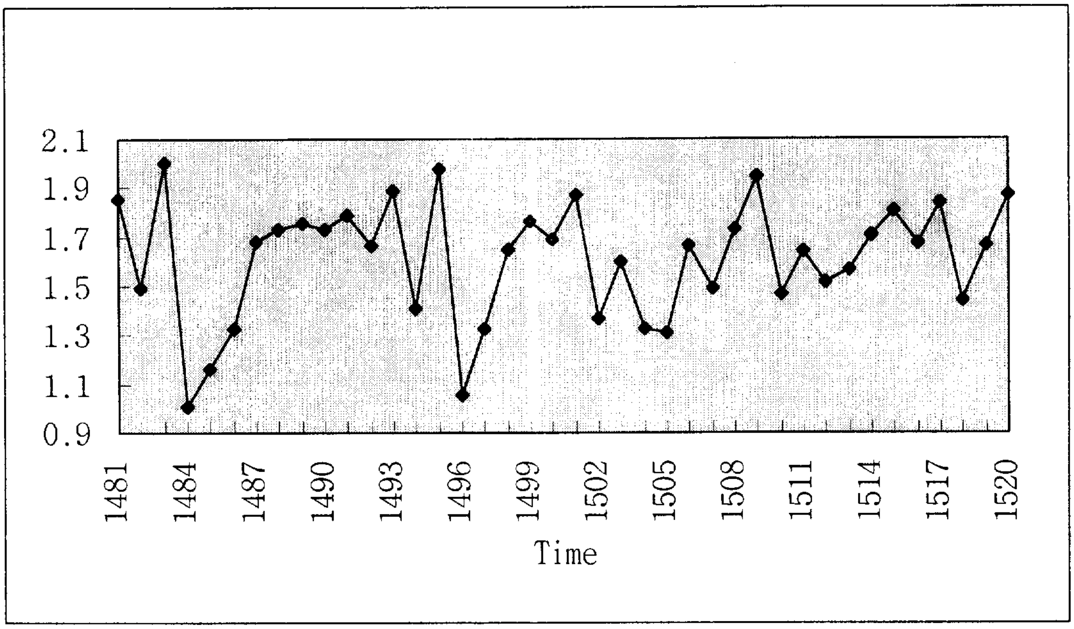
[Figure 7] Architecture of the recurrent neural network (RNN).



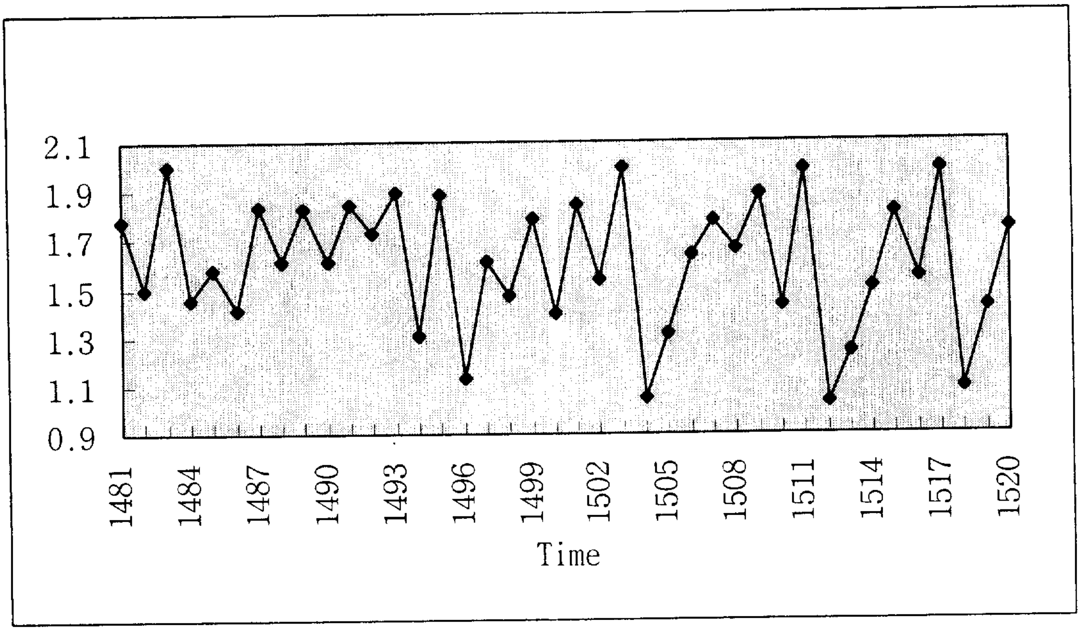
[Figure 8] Schematic of the case based reasoning (CBR) approach. In comparing CBR against the other learning methods, the following architecture was used: number of neighbors $L = 3$ and input vector size $L = 4$.



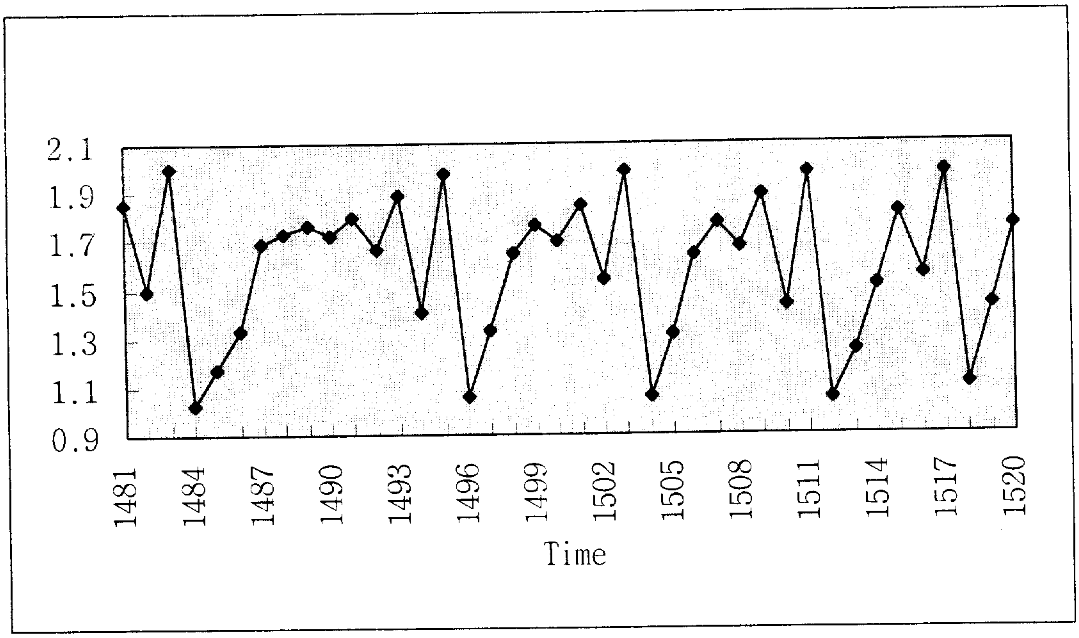
[Figure 9] Partial sequence of a deterministic chaotic series under the Henon model. For pictorial clarity, only a subset of the data series is shown: the 40 points straddling the critical period t = 1500, ranging from 1481 to 1520.



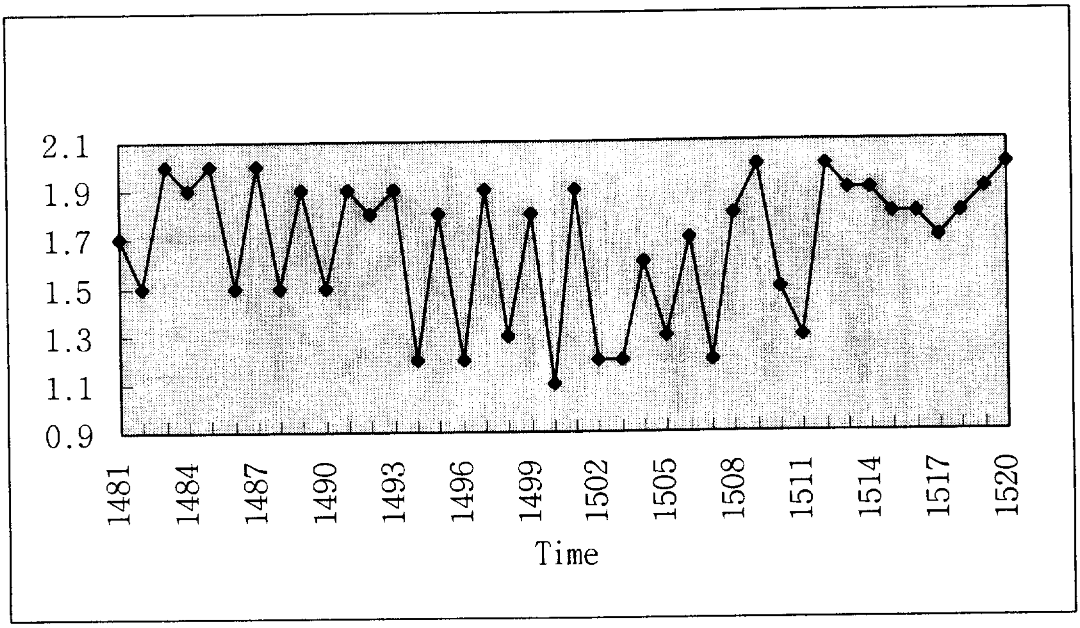
[Figure 10] Partial plot of the modified input stream due to the step function in Figure 5(a). The partial plot covers 20 points to either side of the discontinuity in λ_t at t = 1500.



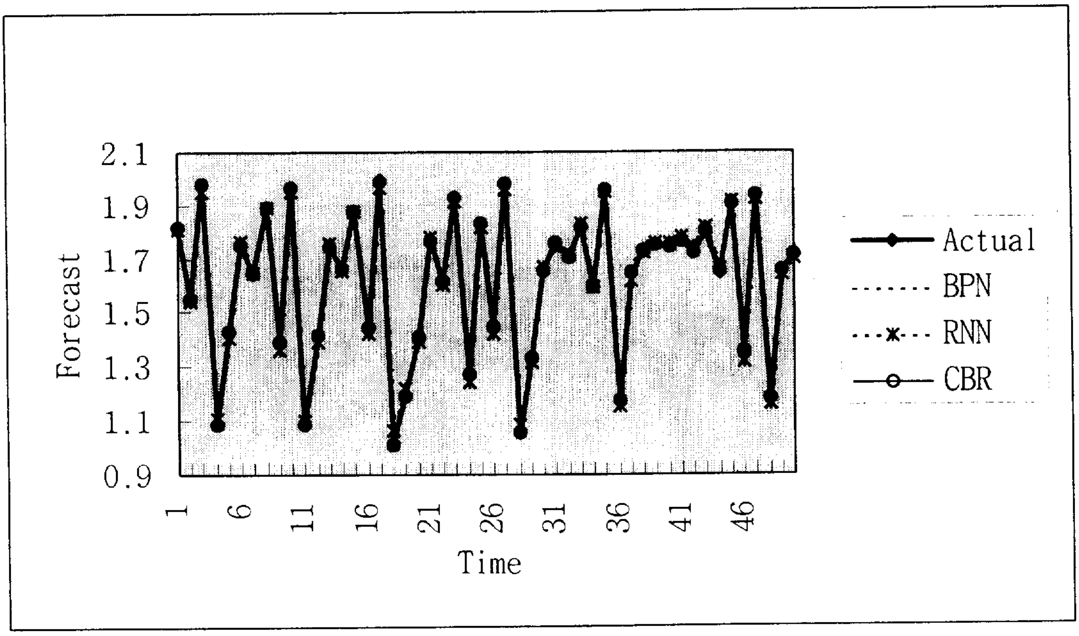
[Figure 11] Partial plot of the square signal mode. The partial plot covers 20 points to either side of the discontinuity at t = 1500.



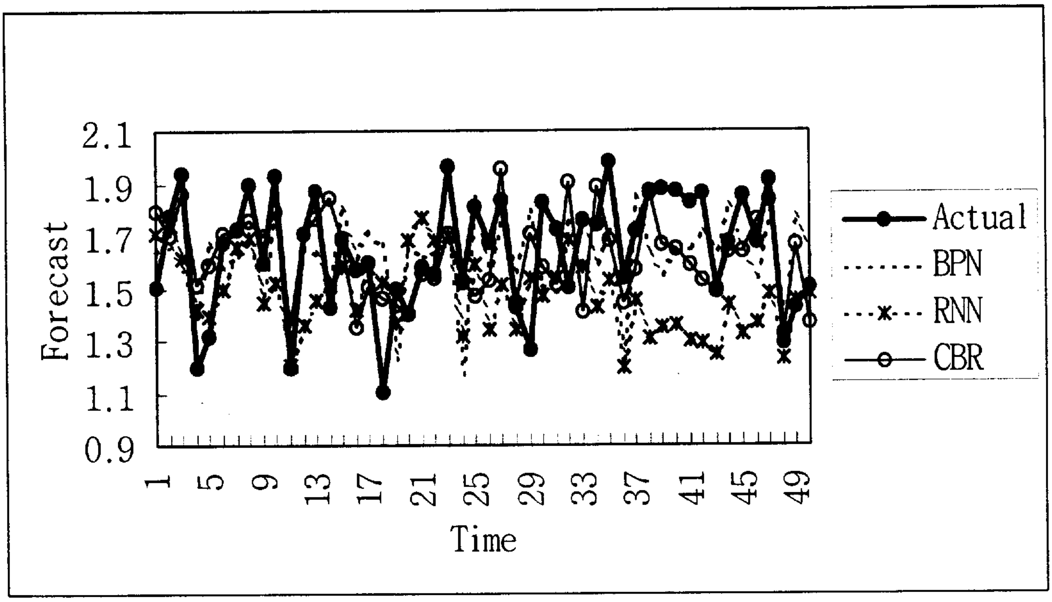
[Figure 12] Partial plot of the tent signal mode. The partial plot covers 20 points to either side of the discontinuity at t = 1500.



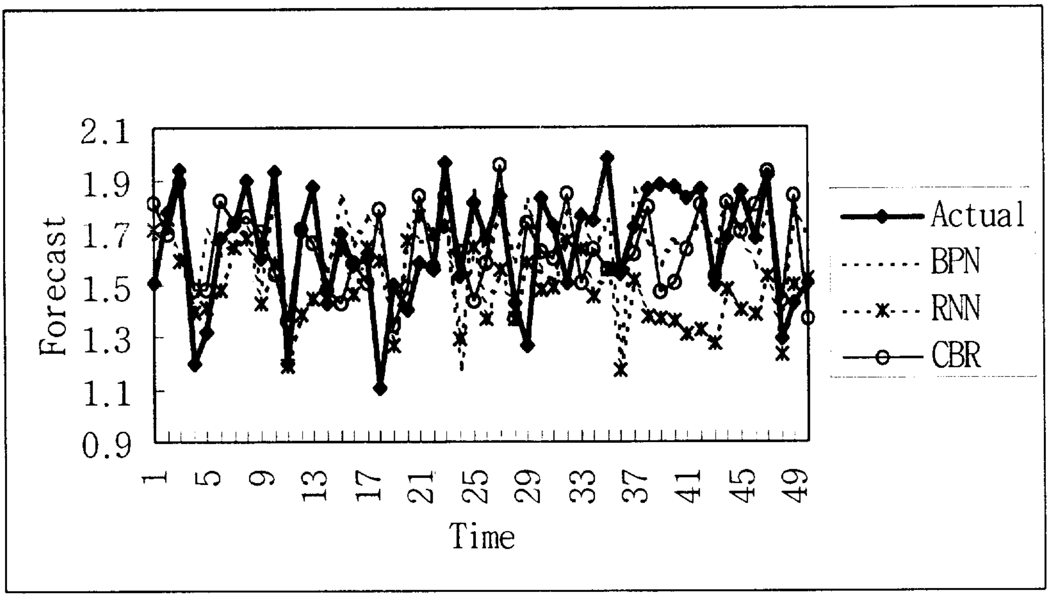
[Figure 13] Partial plot of the pure noise process.



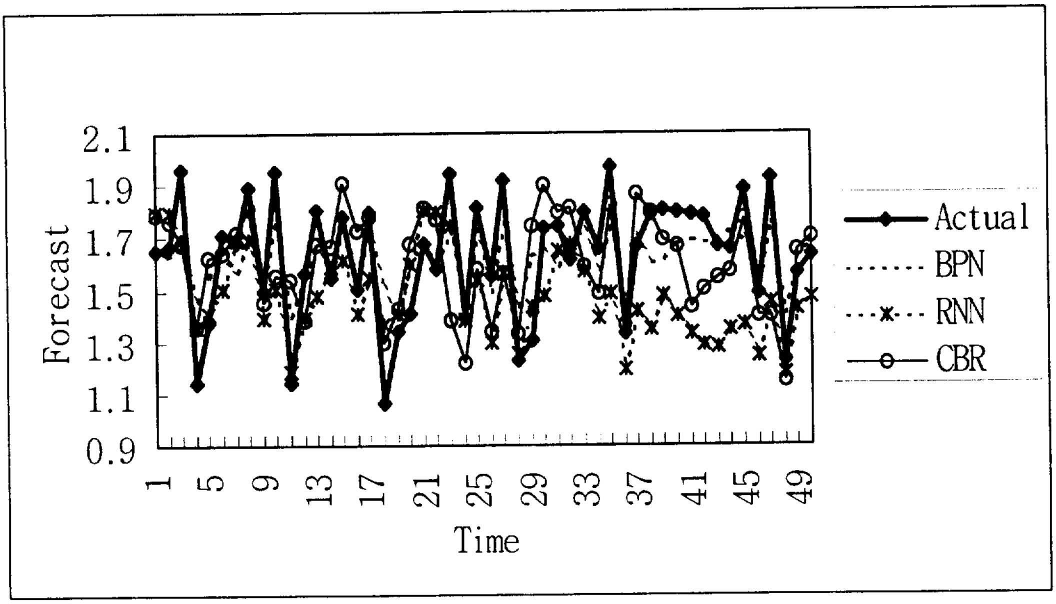
[Figure 14] Plot of forecasts for the pure Henon model using various methods. The test period runs from period 2801 to 3000. For pictorial clarity, however, only the first 50 forecasts are shown.



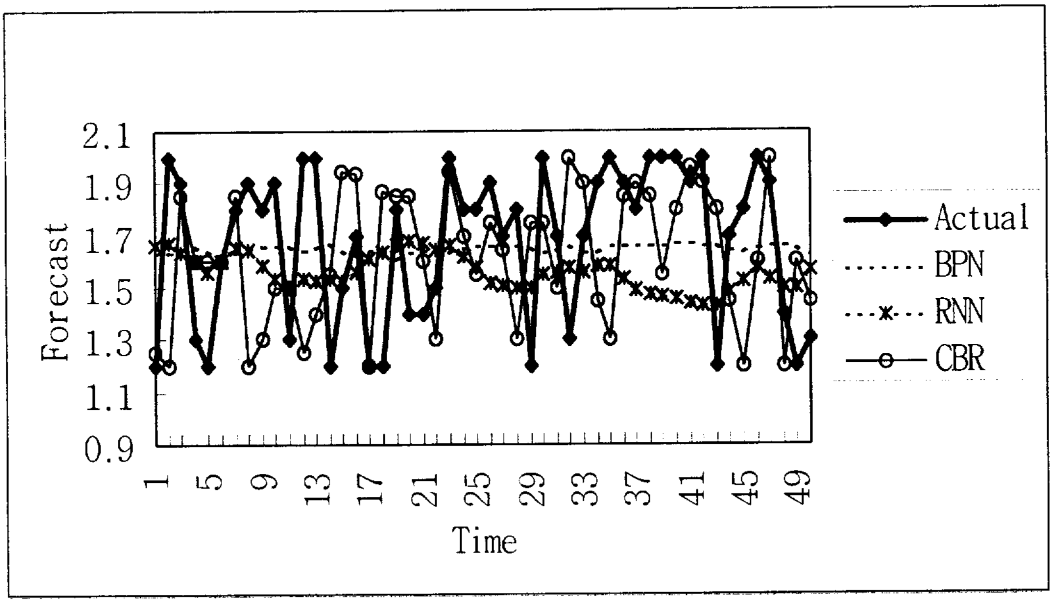
[Figure 15] Plot of forecasts for step model using various learning techniques. The test period runs from period 2801 to 3000, but only the first 50 forecasts are shown.



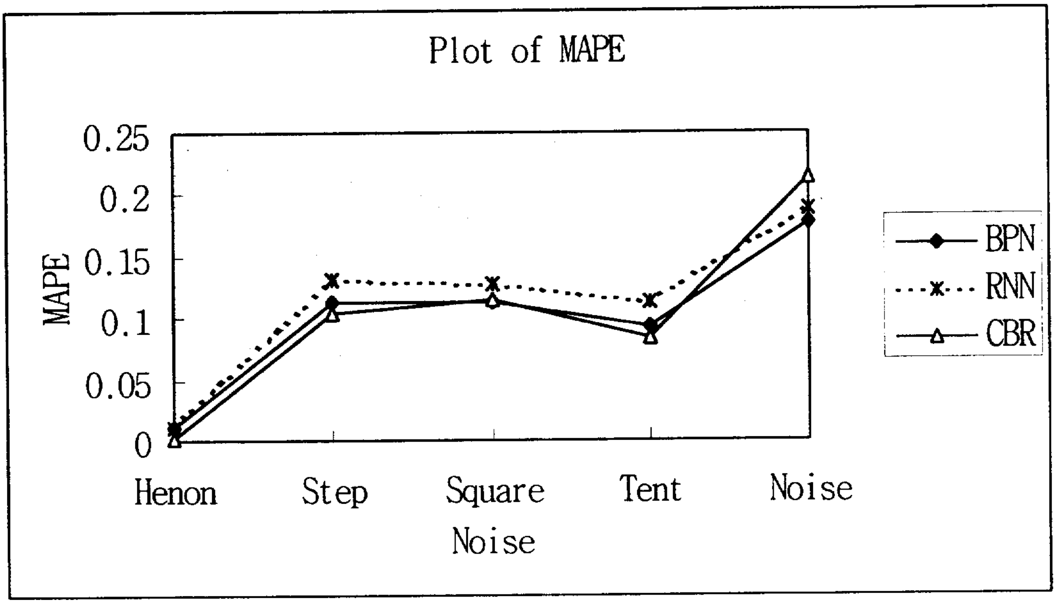
[Figure 16] Plot of forecasts for square model using various methods. The test period runs from period 2801 to 3000, but only the first 50 forecasts are shown.



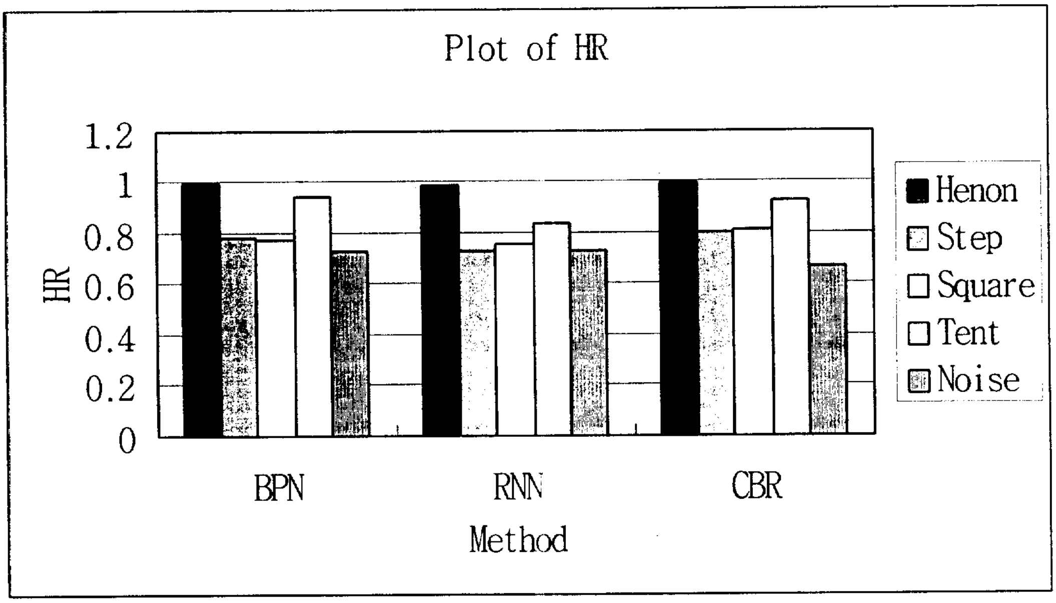
[Figure 17] Plot of forecasts for the tent mode using various methods. The test period runs from period 2801 to 3000, but only the first 50 forecasts are shown



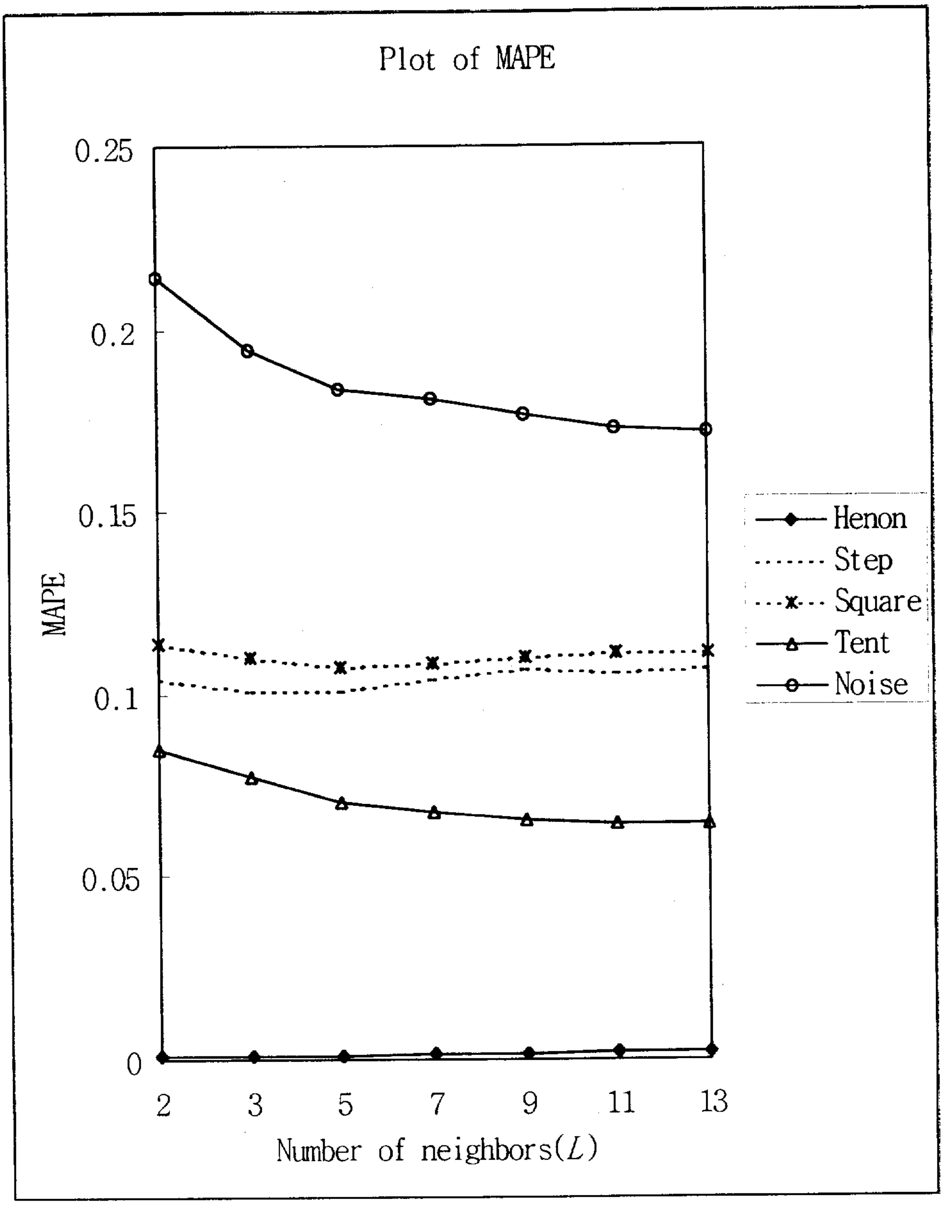
[Figure 18] Plot of forecasts for the pure noise mode using various methods. The test period runs from period 2801 to 3000, but only the first 50 forecasts are shown.



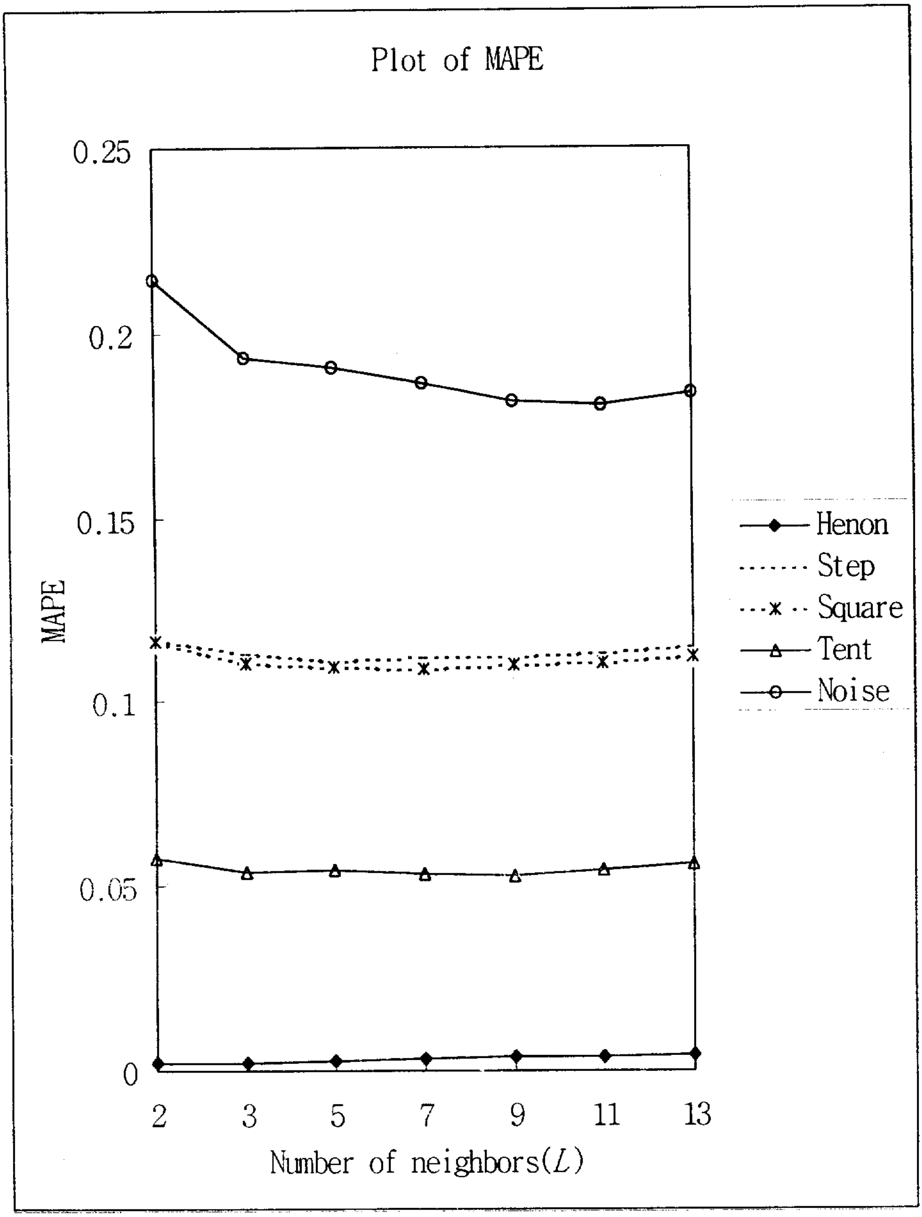
[Figure 19] Plot of mean square errors (MAPE) under varying levels of noise.



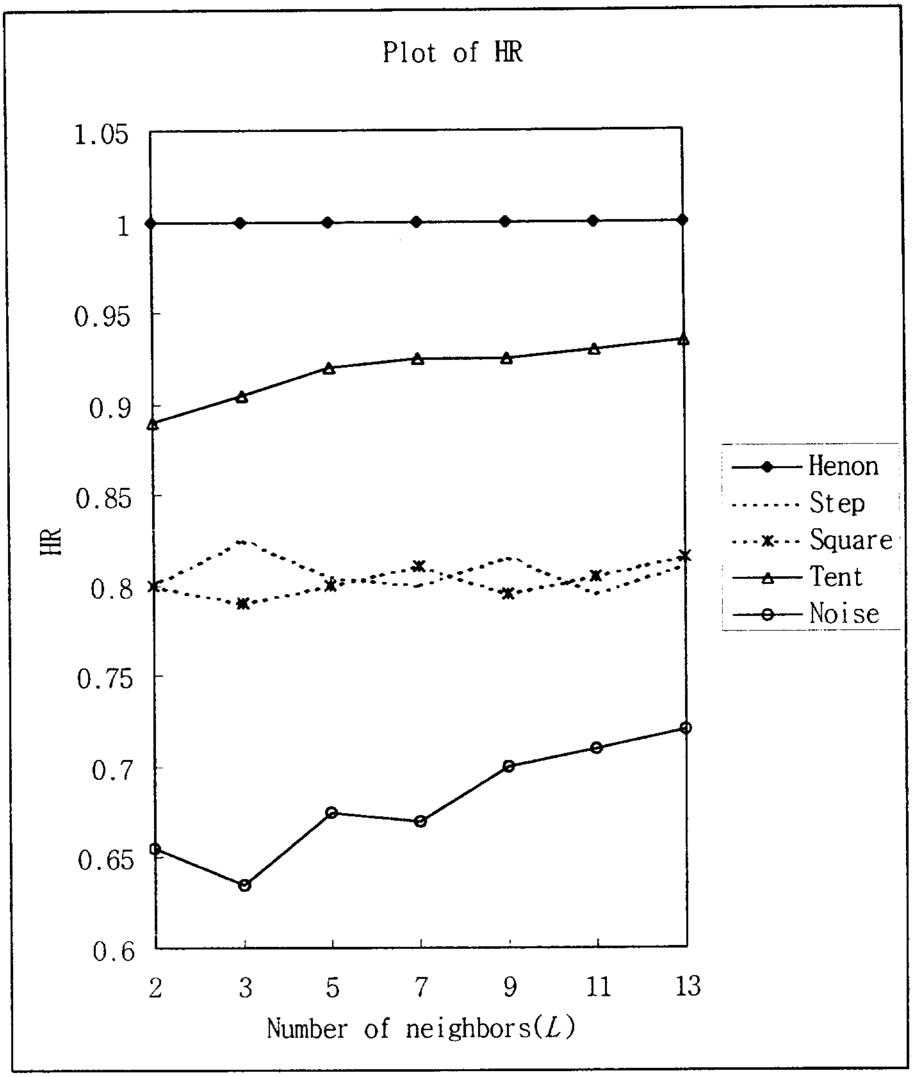
[Figure 20] Plot of hit rate (HR) under varying levels of noise.



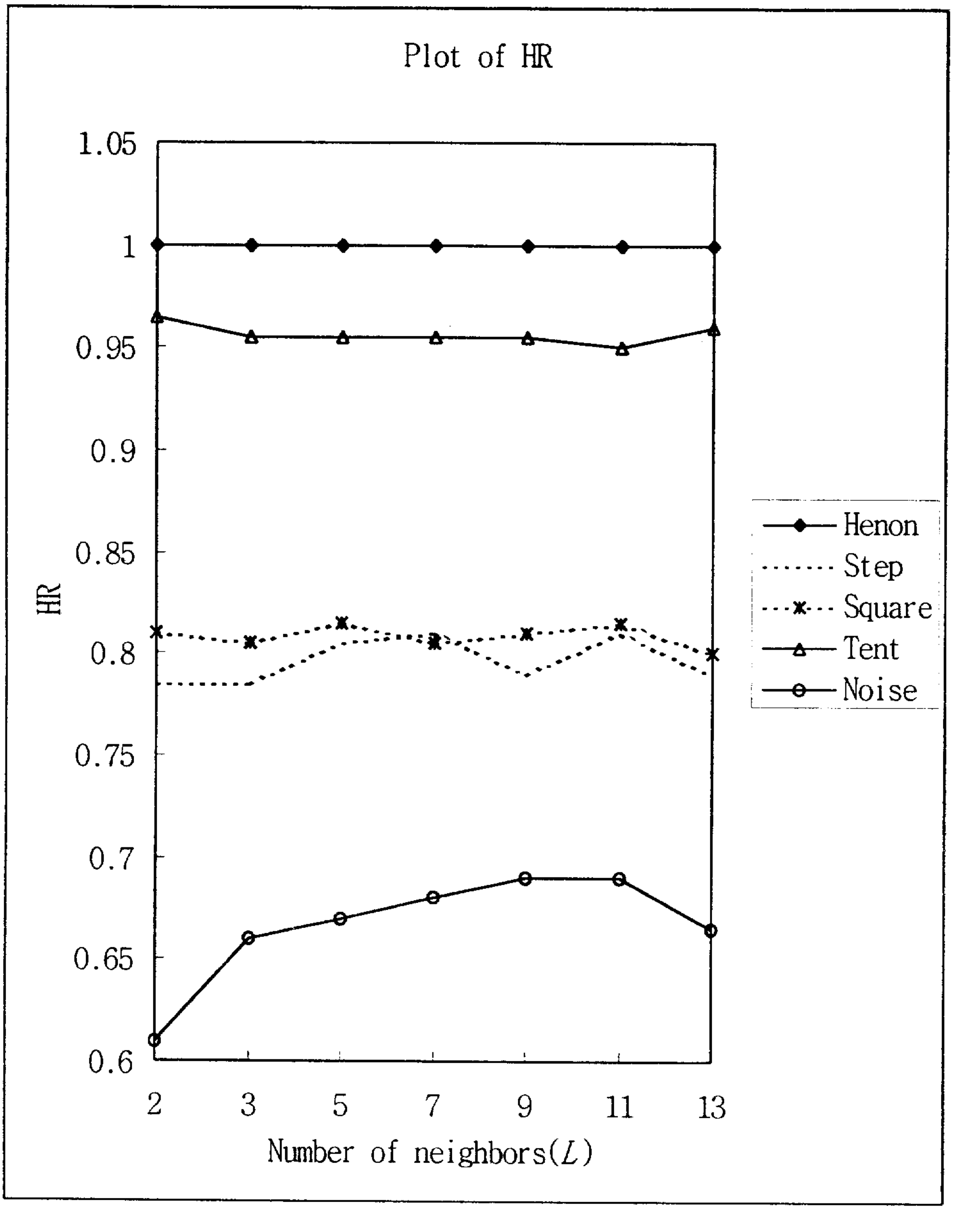
[Figure 21] MAPE of forecasts using CBR as a function of the number L of neighbors and a fixed size (D = 2) for the input vector.



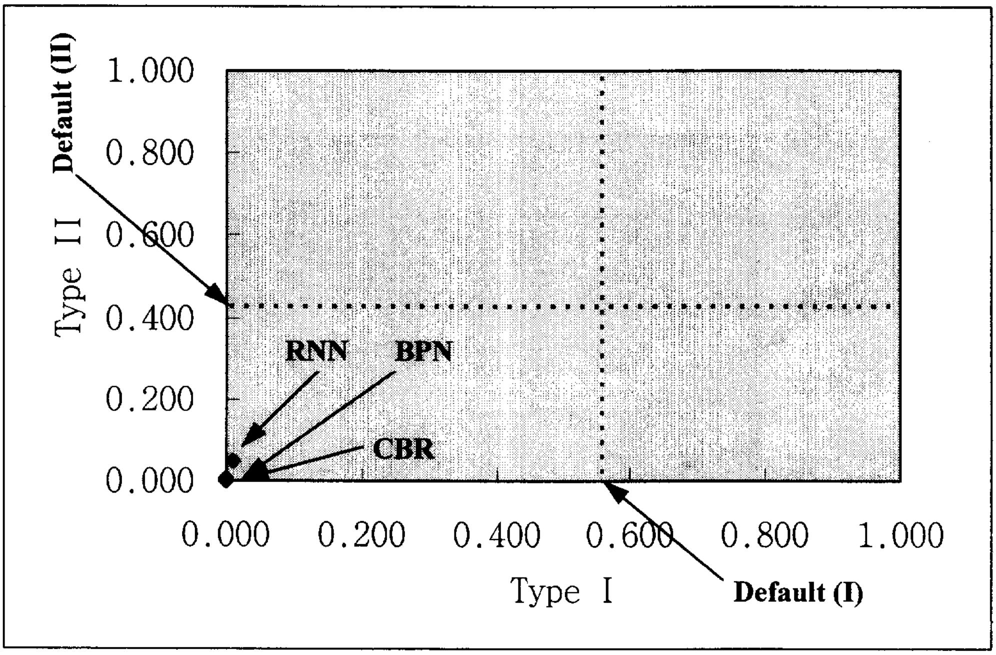
[Figure 22] MAPE of forecasts using CBR as a function of the number L of neighbors and a fixed size (D = 4) for the input vector.



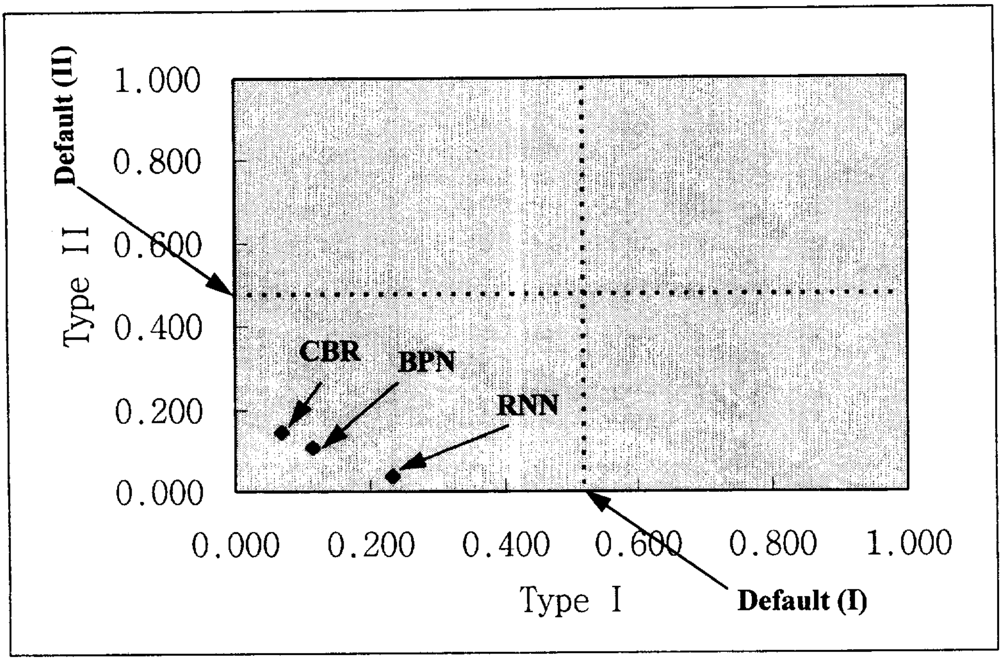
[Figure 23] HR of forecasts using CBR as a function of the number L of neighbors and a fixed size (D = 2) for the input vector.



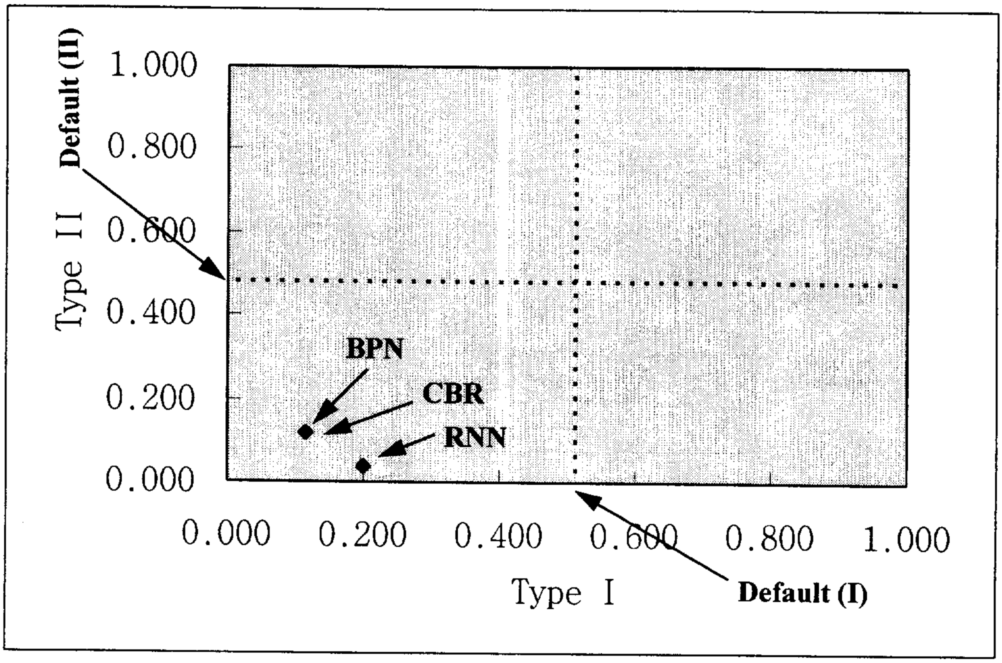
[Figure 24] HR of forecasts using CBR as a function of the number L of neighbors and a fixed size (*D* = 4) for the input vector.



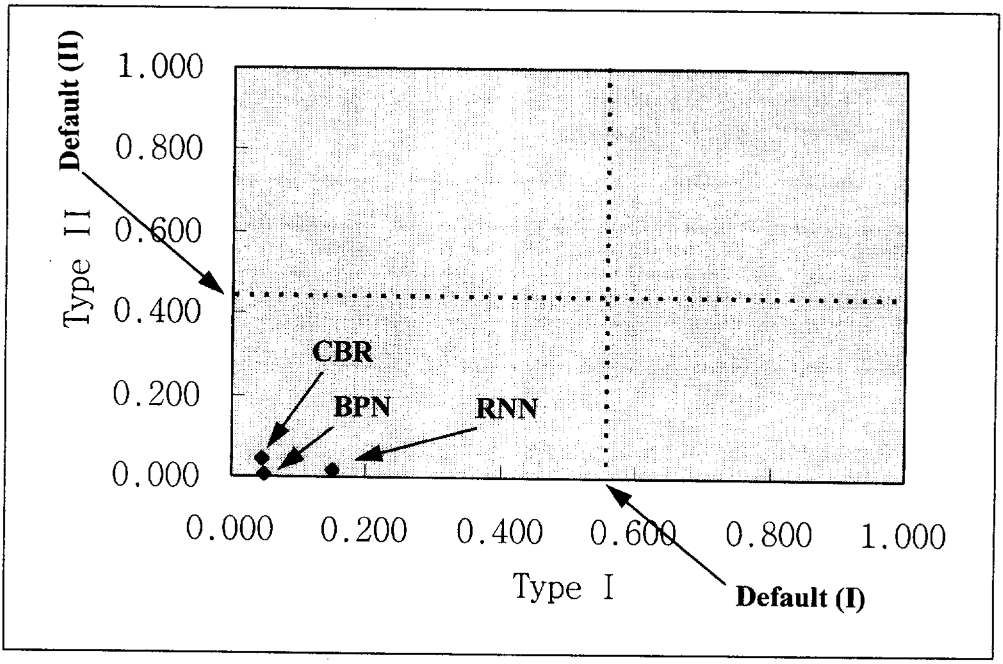
[Figure 25] Mistake chart for the pure Henon model. Dashed lines indicates default mistakes based on a constant prediction of Down or Up. For instance, Default (I) is the expected Type I error due to a constant forecast of Down. All the methodologies in the southwest region outperform the default predictions.



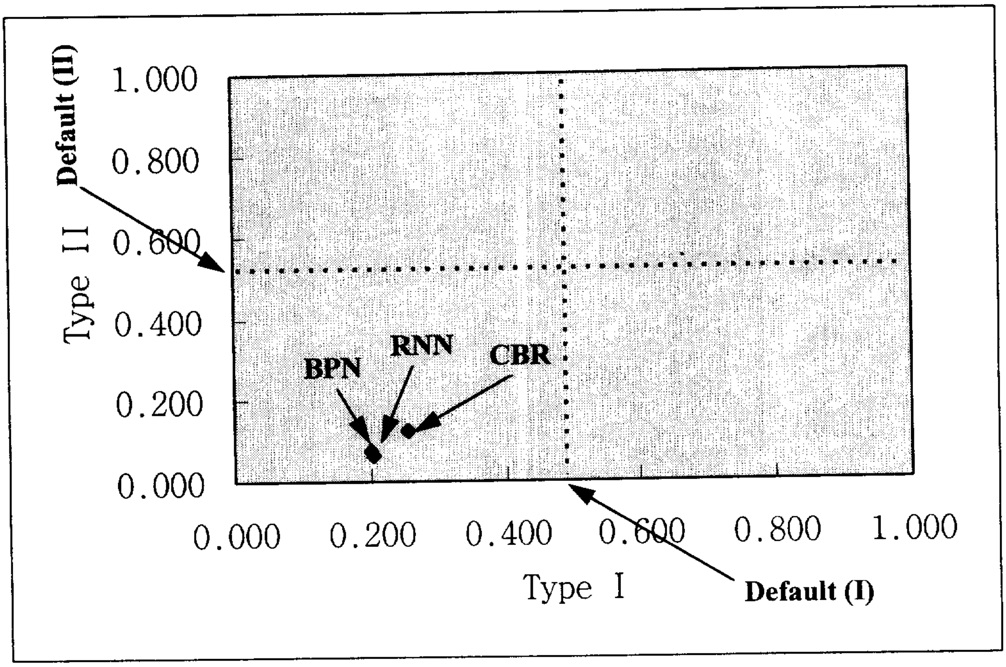
[Figure 26] Mistake chart for the step mode. Dashed lines indicates default mistakes based on a constant prediction of Down or Up. For instance, Default (I) is the expected Type I error due to a constant forecast of Down. All the methodologies in the southwest region outperform the default predictions..



[Figure 27] Mistake chart for the square mode.



[Figure 28] Mistake chart for the tent mode.



[Figure 29] Mistake chart for the noise mode.

〈Table 1〉 Performance by technique and mixing mode according to the metric of mean absolute percentage error (MAPE). Tables 1 through 6 examine the issue of temporal robustness; that is, the stability of performance when a technique is trained under a particular set of conditions, then faces similar or different circumstances.

	Henon	Step	Square	Tent	Noise
BPN	0.011366	0.113030	0.112430	0.092981	0.177675
RNN	0.010869	0.132187	0.127962	0.112924	0.187926
CBR	0.001123	0.104016	0.114119	0.084794	0.214353

〈Table 2〉 Performance by technique and mixing mode according to the metric of hit rate (HR).

	Henon	Step	Square	Tent	Noise
BPN	0.995	0.780	0.770	0.945	0.725
RNN	0.985	0.730	0.760	0.835	0.730
CBR	1.000	0.800	0.810	0.925	0.670

〈Table 3〉 Two-way ANOVA for the data in Table 1: the input observations pertain to the absolute percentage error (APE).

Source of Variation	D.F.	Sum of Squares	Mean Squares	F	Sig of F
Main Effects	6	10.660	1.777	236.133	0.000
Technique	2	0.095	0.047	6.311	0.002
Mode	4	10.565	2.641	351.044	0.000
2-Way Interaction	8	0.257	0.032	4.273	0.000
Explained	14	10.917	0.780	103.641	0.000
Residual	2985	22.459	0.008		
Total	2999	33.376	0.011		

3000 cases (200*15 cases) were processed.

〈Table 4〉 Chi-square test for the data in Table 2. Each cell entry denotes the number of hits out of 200 trials.

	Henon	Step	Square	Tent	Noise	Totals
BPN	199	156	154	189	145	843
RNN	197	146	152	167	146	808
CBR	200	160	162	185	134	841
Totals	596	462	468	541	425	2492

Chi-square statistic for independence:

$$x^2_{(r-1)(c-1)} = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}$$

where E_{ij} = (row i total)(col. j total)/grand total.

$\chi^2_8 = 2.289405$. $p-value = 0.029171$

〈Table 5〉 Stability of BPN across environmental scenarios, according to the metric of MAPE. Each cell contains 3 numbers in the format a : b (c). Here a is the performance metric in the training phase, b the metric in the test phase, and c the level of significance due to a t-test for the difference of means. Tables 5 through 10 represent an analysis of cross-sectional robustness for each learning technique.

	Henon	Step	Square	Tent	Noise
Henon	-	0.011:0.113 (.000)	0.011:0.112 (.000)	0.011:0.092 (.000)	0.011:0.177 (.000)
Step	-	-	0.113:0.112 (.594)	0.113:0.092 (.072)	0.113:0.177 (.000)
Square	-	-	-	0.112:0.092 (.021)	0.112:0.177 (.002)
Tent	-	-	-	-	0.092:0.177 (.000)
Noise	-	-	-	-	-

〈Table 6〉 Stability of BPN across environmental scenarios, according to the metric of HR. Each cell contains 3 numbers in the format a : b (c). Here a is the performance metric in the training phase, b the metric in the test phase, and c the level of significance due to a test of proportions.

	Henon	Step	Square	Tent	Noise
Henon	-	0.995:0.780 (1.02E-11)	0.995:0.770 (2.82E-12)	0.995:0.945 (0.003378)	0.995:0.725 (7.33E-15)
Step	-	-	0.780:0.770 (.810738)	0.780:0.945 (1.66E-06)	0.780:0.725 (.202505)
Square	-	-	-	0.770:0.945 (5.56E-07)	0.770:0.725 (.300295)
Tent	-	-	-	-	0.725:0.725 (3.1E-09)
Noise	-	-	-	-	-

〈Table 7〉 Stability of RNN across environmental scenarios, according to the metric of MAPE.

	Henon	Step	Square	Tent	Noise
Henon	-	0.010:0.132 (.000)	0.010:0.127 (.000)	0.010:0.112 (.000)	0.010:0.187 (.000)
Step	-	-	0.132:0.127 (.935)	0.132:0.112 (.182)	0.132:0.187 (.591)
Square	-	-	-	0.127:0.112 (.230)	0.127:0.187 (.658)
Tent	-	-	-	-	0.112:0.187 (.477)
Noise	-	-	-	-	-

〈Table 8〉 Stability of RNN across environmental scenarios, according to the metric of HR.

	Henon	Step	Square	Tent	Noise
Henon	-	0.985:0.730 (3.01E-13)	0.985:0.760 (1.53E-11)	0.985:0.835 (1.6E-07)	0.985:0.730 (3.01E-13)
Step	-	-	0.730:0.760 (.491268)	0.730:0.835 (.010922)	0.730:0.730 (3.01E-13)
Square	-	-	-	0.760:0.835 (.061998)	0.760:0.730 (.491268)
Tent	-	-	-	-	0.835:0.730 (.010922)
Noise	-	-	-	-	-

〈Table 9〉 Stability of CBR across environmental scenarios, according to the metric of MAPE.

	Henon	Step	Square	Tent	Noise
Henon	-	0.001:0.104 (.000)	0.001:0.114 (.000)	0.001:0.084 (.000)	0.001:0.214 (.000)
Step	-	-	0.104:0.114 (.079)	0.104:0.084 (.698)	0.104:0.214 (.000)
Square	-	-	-	0.114:0.084 (.200)	0.114:0.214 (.000)
Tent	-	-	-	-	0.084:0.214 (.000)
Noise	-	-	-	-	-

〈Table 10〉 Stability of CBR across environmental scenarios, according to the metric of HR.

	Henon	Step	Square	Tent	Noise
Henon	-	1.000:0.800 (2.63E-11)	1.000:0.810 (2.63E-11)	1.000:0.925 (1.4E-06)	1.000:0.670 (.000)
Step	-	-	0.800:0.810 (1.000)	0.800:0.925 (.012888)	0.800:0.670 (.001128)
Square	-	-	-	0.810:0.925 (.012888)	0.810:0.670 (.001128)
Tent	-	-	-	-	0.925:0.670 (2.08E-08)
Noise	-	-	-	-	-

〈Table 11〉 Types of error by methodology for the pure Henon mode. Type I (false rejection) refers to a down prediction when the actual index rises; and Type II (false acceptance) refers to an up prediction when the actual index falls. For Tables 11 to 15, each entry denotes the proportion of mistakes over the trial period of 200 cases. The best performance for each mistake type is highlighted in bold.

Henon	Type I	Type II
BPN	**0.000**	0.005
RNN	0.010	0.050
CBR	**0.000**	**0.000**

〈Table 12〉 Types of error by methodology for the step mode.

Step	Type I	Type II
BPN	0.115	0.105
RNN	0.235	**0.035**
CBR	**0.070**	0.145

〈Table 13〉 Types of error by methodology for the square mode.

Square	Type I	Type II
BPN	**0.115**	0.115
RNN	0.200	**0.040**
CBR	**0.115**	0.115

〈Table 14〉 Types of error by methodology for the tent mode.

Tent	Type I	Type II
BPN	0.050	**0.005**
RNN	0.150	0.015
CBR	**0.045**	0.045

〈Table 15〉 Types of error by methodology for the pure noise mode.

Noise	Type I	Type II
BPN	**0.200**	0.075
RNN	0.205	**0.065**
CBR	0.255	0.125