

Identification of Discrete Chaotic Maps with Singular Points

P. G. AKISHIN^a, P. AKRITAS^{b,c}, I. ANTONIOU^{b,c,*} and V. V. IVANOV^{a,b}

^aLaboratory of Computing Techniques and Automation, Joint Institute for Nuclear Research, Dubna, 141980, Russia;
^bInternational Solvay Institute for Physics and Chemistry, CP-231, ULB, Campus Plaine, Bd. du Triomphe, 1050,
Brussels, Belgium; ^cTheoretische Natuurkunde, Free University of Brussels, Brussels, Belgium

(Received 22 September 2000)

We investigate the ability of artificial neural networks to reconstruct discrete chaotic maps with singular points. We use as a simple test model the Cusp map. We compare the traditional Multilayer Perceptron, the Chebyshev Neural Network and the Wavelet Neural Network. The numerical scheme for the accurate determination of a singular point is also developed. We show that combining a neural network with the numerical algorithm for the determination of the singular point we are able to accurately approximate discrete chaotic maps with singularities.

Keywords: Neural networks; Chaotic maps; Wavelets

1. INTRODUCTION

In the past few years new methods for the analyzing and simulating complex processes based on artificial neural networks are being intensively developed. Feed-forward layered neural networks based on back-propagation learning have been applied to a number of problems [1–3].

We have proposed [4] a new approach for the reconstruction and prediction of discrete chaotic maps based on the Chebyshev Neural Network (CNN) network. We have compared the performance of the CNN network with a conventional Multi-Layer Perceptron (MLP) and demonstrated

that the CNN exceeded significantly the MLP in the learning rate and in the accuracy of approximation. As the CNN provides better approximation of time series analyzed, we have new possibilities for long-term prediction.

During the last years a new technique, the wavelet networks [5], based on the recently developed wavelet theory (*e.g.*, see [6,7]) and neural networks has been developed for chaotic time series prediction and data classification [8,9].

In [10] we proposed a feed-forward neural network realizing the approximation of one-dimensional function based on the wavelet decomposition of the corresponding time series. This

*Corresponding author. e-mail: iantonio@vub.ac.be

Wavelet Neural Network (WNN) has been applied to the electrocardiogram (ECG) analysis problem (see, for instance, [11]).

We investigate here the abilities of different neural networks to identify discrete chaotic maps containing singular points. We compare the MLP, CNN (Section 3) and Wavelet Neural Network (WNN) (Section 4) for the identification/reconstruction of the well-known Cusp map which arises as an approximation of the Poincaré section of the Lorenz attractor [12] (Section 2). The numerical scheme for the determination of singular point location is developed in Section 5.

2. ACCURATE COMPUTATION OF THE CUSP MAP

We shall consider a simple but important map, namely, the Cusp map [12], on the interval $[-1, 1]$:

$$X_{n+1} = S[X_n] = 1 - 2|X_n|^{1/2}, \quad n = 0, 1, 2, \dots \quad (1)$$

By assuming a starting value X_0 in $[-1, 1]$, the n -th iteration is

$$X_n = S^n[X_0].$$

Although the dynamical law $S[\cdot]$ is simple (from the computational point of view), the resulting evolution is complex, because the Cusp map with invariant density $\rho_{eq}(x) = (1/2)(1-x)$ is an exact dynamical system [13] with positive Kolmogorov-Sinai entropy production [14, 15].

If N is the number of digits corresponding to the required accuracy and M is the maximal number of digits available by our machine (limited by the memory of the computer), then there is a maximal number τ of iterations preserving the accuracy N . This number τ is known as the horizon of Predictability [16]. For $N > \tau$ there is no reliable prediction with respect to the accuracy N .

In order to compute the reliable number of iterations for the initial value $X_0 = 0.51$ with accuracy $N = 10$, we used the **Multiple-Precision Floating-Point Arithmetic**: the MPA package, Algorithm A105 from the CERN Library [17]. This package provides the possibility to perform all necessary operations with variables of arbitrary base and number of digits, and as a consequence to make computations with needed precision.

In this way we obtained a reliable time series, see Figure 1, for at least 2000 iterations for the chaotic Cusp map which will be processed by the three neural networks.

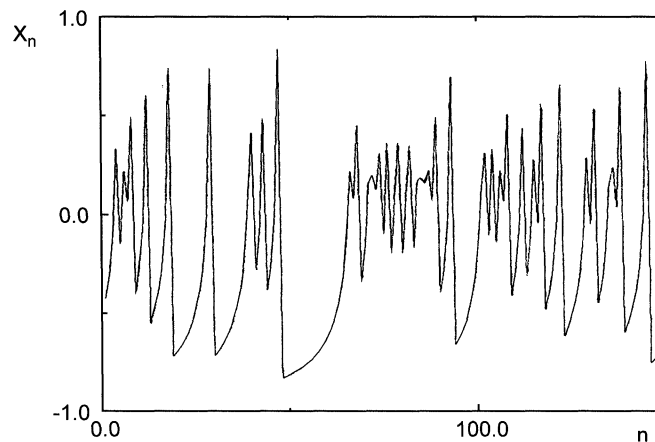


FIGURE 1 The initial part (150 iterations) of the Cusp-map time series.

3. IDENTIFICATION OF THE CHAOTIC MAP USING THE MLP AND CNN NETWORKS

The MLP and CNN networks were described in details in our previous paper [4]. Here we give their brief description, the procedure for the map identification and present the results of approximation of the analyzed time series.

3.1. The MLP Neural Network

The MLP is a feed-forward multi-layer network, which involves an input layer corresponding to the data to be processed, an output layer dealing with the results, and hidden layers. A network scheme is presented in Figure 2.

Here x_k , h_j and y_i denote the input, hidden and output neurons, respectively; w_{jk} and w_{ij} are the weights of connections between neurons.

The adaptation of the MLP to the problem to be solved is performed by the minimization of the error functional with respect to the weights

$$E = \frac{1}{2} \sum_p [\bar{y}(x_p) - \vec{f}(x_p)]^2 \quad (2)$$

applying the *back-propagation* algorithm [18]. Here $p = 1, 2, \dots, N_{\text{train}}$ is the number of training patterns, $\vec{f}(x_p)$ is the target value of the output signal.

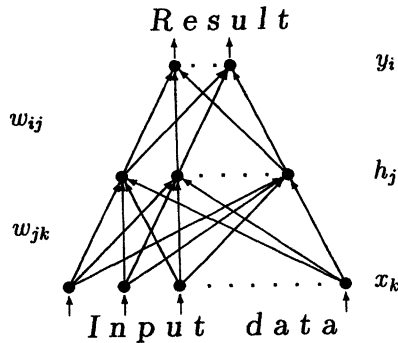


FIGURE 2 Scheme of the MLP network with one hidden layer.

In this work we used the MLP neural network from the JETNET package [19] with the following the architecture: one input neuron, one output neuron and a single hidden layer with a variable number of neurons.

3.2. The CNN Network

Consider a function $f(x)$ defined on a finite set of values of $x: f(x_0), f(x_1), \dots, f(x_n)$. We look for the interpolation function $\varphi_n(x)$ based on orthogonal Chebyshev polynomials $T_n(x)$ of the I-st kind:

$$\varphi_n(x) = c_0 T_0(x) + c_1 T_1(x) + c_2 T_2(x) + \dots + c_n T_n(x), \quad (3)$$

where $T_n(x) = \cos(n \arccos x)$, $|x| \leq 1$.

To calculate the coefficients c_k of (3) one have to know the values of the function $f(x)$ at the nodal points [20]. However, in practice this is hardly possible. In order to simplify the calculation of the coefficients c_i , we proposed the CNN network with architecture presented in Figure 3.

The argument x is sent from the input neuron to the neurons of the hidden layer. Each i -th neuron of the hidden layer transforms the received signal in accordance with the transfer function $g_i(x)$ in the form of a Chebyshev polynomial $g_i(x) = T_i(x)$, where $i = 0, \dots, n$. Then, the sum of weighted signals from the neurons of the hidden layer $a = \sum_{i=0}^n \omega_i \cdot T_i(x)$ is supplied to the output neuron

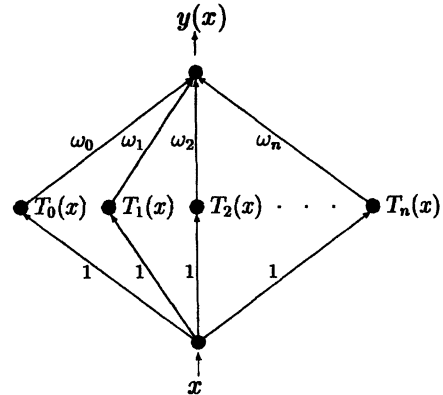


FIGURE 3 Architecture of the CNN network.

from which the computed value of function $y(x)$ is obtained.

The weights ω_i , which play the role of expansion coefficients, are obtained by the CNN “training” applying also the *back-propagation* algorithm [18].

3.3. Procedure of the Map Identification

A procedure of identification of the map underlying a series of numbers being the input to a neural network (NN) consists in the following.

- (1) We divide the series of numbers in two parts. For example the first 1000 numbers and the second 1000 numbers if we have 2000 numbers.
- (2) We use the first part of the series for the training of the NN and the second part of the series for the testing of the performance of the NN.
- (3) The training of the NN is performed by considering all couples (x_i, x_{i+1}) of the first part of the series of numbers.
- (4) We supply the first argument of each couple (x_i, x_{i+1}) as an input to the NN and obtain the output $y(x_i)$. The output is compared with x_{i+1} and the weights are corrected so that $y(x_i)$ coincides with x_{i+1} . A single run over all

points of the set composes one training epoch.

- (5) After each training epoch the weights are fixed and the quality of the map reconstruction (one-step prediction) by the neural network was estimated. The second part of the data was used for this purpose.
- (6) Then, the neural network training is repeated again. The procedure terminates when the desired correspondence between output signals $y(x)$ and target values $S[x]$ is achieved for the whole set of training data.

3.4. Results for the MLP and CNN Networks

In Figures 4 and 5 we present the distributions of errors for the MLP and CNN networks: (1) *Error* is the deviation of the actual value $S[x]$ from the value $y(x)$ predicted by the network, and (2) *Relative error* is the relative deviation: $\text{Error}/S[x]$. These distributions concern the testing set. The histograms were fitted by Gaussian functions and the value of the standard deviation (*Sigma* in figures) can be used as the estimator of the accuracy for one-step prediction.

Comparing Figures 4 and 5 we can see that the CNN network fits better the analyzed time series compared with the MLP network.

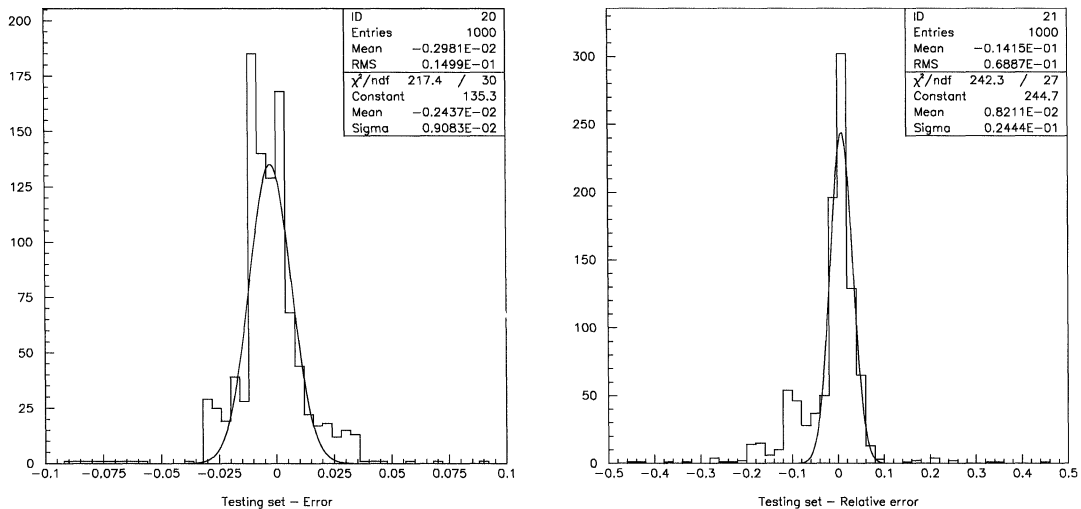
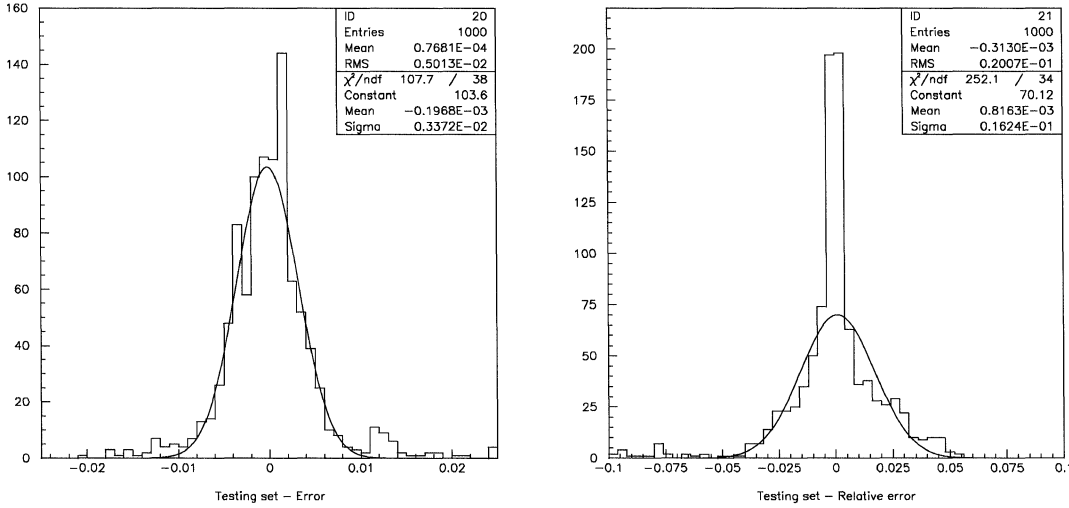


FIGURE 4 Distributions of *Error* and *Relative error* for the MLP neural network.


 FIGURE 5 Distributions of *Error* and *Relative error* for the CNN neural network.

4. THE WAVELET NEURAL NETWORK

The “wavelet decomposition” of an one-dimensional square integrable function $f(x) \in L_2(\mathbb{R})$ at (multi-resolution) level J is given by the series:

$$f(x) \approx D_1 + D_2 + \dots + D_J + S_J, \quad (4)$$

where

$$S_j(x) = \sum_k c_k^j \phi(2^j x - k),$$

$$D_j(x) = \sum_k d_k^j \psi(2^j x - k), \quad j = \overline{1, J}$$

and the basis functions $\{\phi_{jk}(x) = \phi(2^j x - k), \psi_{jk}(x) = \psi(2^j x - k), j, k \in \mathbb{Z}\}$ are obtained from the *father* and *mother* wavelets $\phi(x)$ and $\psi(x)$, respectively, by the binary dilation 2^j and the dyadic translation $k/2^j$ [6].

In Eq. (4) terms D_1, D_2, \dots, D_J and S_J are responsible, respectively, for the *detail* and *smooth* approximations. The reduced sum of the expansion (4) restricted by the smooth approximation term is

$$f_J(x) = \sum_k c_k^J \phi(2^J x - k), \quad (5)$$

where $\phi(x)$ is also known as the scaling function corresponding to the *mother* wavelet function $\psi(x)$ [6, 7].

Often $\phi(x)$ is taken as the m -th order cardinal B -splines N_m , where m is an arbitrary positive integer. The B -spline has compact support $\text{supp} N_m = [0, m]$, it is positively determined inside this interval, symmetrical regarding the center of the support (where it has the maximum), and restricts the unit area. In each interval $[k, k+1]$, $k=0, \dots, m$, $N_m(x)$ is a polynomial of $(m-1)$ order. These peculiarities provide remarkable smoothing properties of B -splines [6, 21].

Suppose that the analyzed time series can be fitted by a second order polynomial $f(x) \approx a + b(x - \xi)^2$. So we can take the scaling functions as B -splines of the 3-rd order $N_3(x)$:

$$N_3(x) = \begin{cases} x^2/2 & \text{if } x \in [0, 1]; \\ -x^2 + 3x - 3/2, & \text{if } x \in [1, 2]; \\ x^2/2 - 3x + 9/2 & \text{if } x \in [2, 3]; \\ 0 & \text{elsewhere.} \end{cases}$$

The determination of the coefficients $\{c_k^j\}$ in formula (5) is obtained from the minimization of the difference between the function $f(x)$ and the

linear combination of splines $\{N_3(2^j x - k), k \in \mathbb{Z}\}$ in the L_2 -norm [22]:

$$E(\{\omega_k\}) = \|f(x) - \sum_k \omega_k N_3(2^j x - k)\|_{L_2}^2$$

where $\omega_k \equiv c_k^j$.

As we have a limited number N of values of function $f(x)$ ($f_b = f(x_p), p = 1, 2, \dots, N$) the approximated values of the coefficients can be obtained by minimizing the functional:

$$E_N(\{\omega_k\}) = \sum_{p=1}^N \left[f_p - \sum_k \omega_k N_3(2^j x_p - k) \right]^2. \quad (6)$$

The functional (6) can be interpreted as the error functional of the feed-forward neural network of a specific architecture (see below). Its minimization is reduced to the network learning based upon known values of function $f(x)$.

The approximation of the signal at level j is:

$$f_j(x) = \sum_k \omega_k N_3(2^j x - k). \quad (7)$$

Let's suppose that experimental data correspond to the time interval $[0, 1]$. As in this interval only terms with numbers $k = -3, -2, -1, \dots, 2^j$ give their contribution into the expression (7) the

coefficients with other numbers k can be assumed to be equal to zero. The series (7) can be re-written in the terms of the three-layered perceptron. Then an argument x is given to its input and the fitted signal is taken from its output.

The architecture of this network looks similar to the architecture of the CNN network described in [4] (see Fig. 6).

The number of neurons in the hidden layer is equal to the number of terms in the expression (7), moreover, each k -th neuron corresponds to the k -th term in (7). The weights and thresholds between the input layer and the hidden layer are fixed and are set to be equal to the values $\alpha_k = 2^j, \theta_k = -k$. The transfer functions of hidden neurons are the third-order B -splines. The thresholds between the hidden layer and the output neuron are taken equal to zero, and each k -th coefficient corresponds to each k -th weight. Thus, training this neural network on the experimental data and using standard back-propagation method [18] for the minimization of functional (6) we can receive the needed set of coefficients.

Figure 7 presents the distributions of errors for the WNN network. These distributions demonstrate that the WNN network fits significantly better the analyzed time series compared with the MLP and CNN networks.

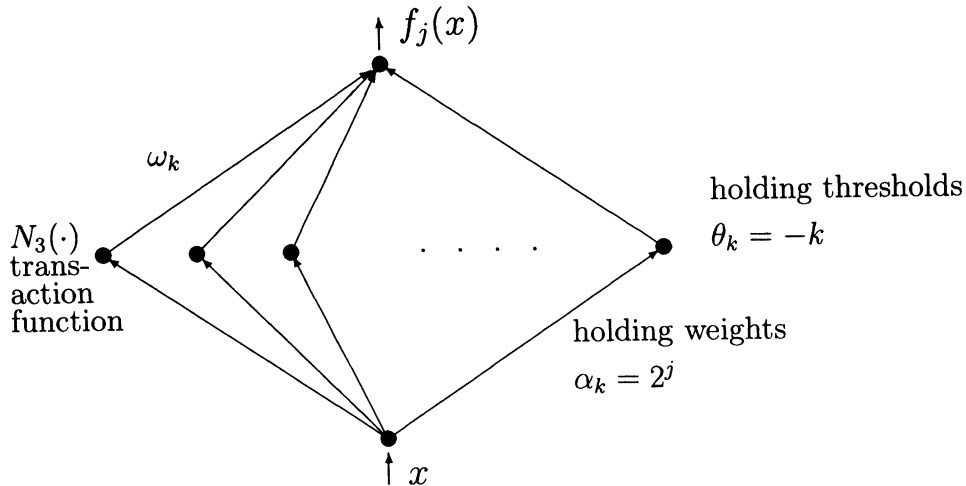


FIGURE 6 The architecture of the WNN neural network.

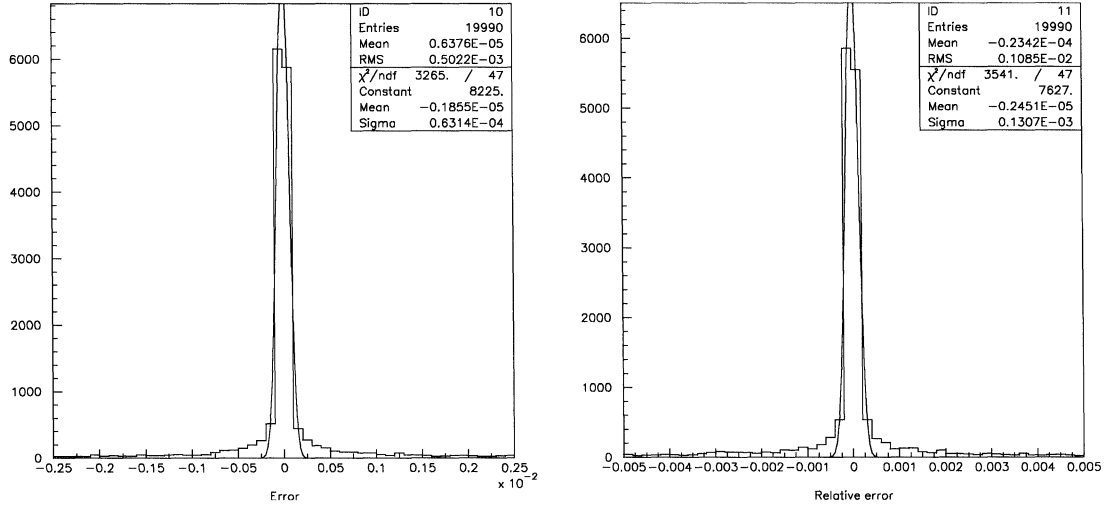


FIGURE 7 Distributions of Error and Relative error for the WNN network.

At the same time, one may notice long tails which are caused by couples concentrated in the region close to the singular point. This point is characterized by discontinuities of all derivatives, and, due to this reason, the Cusp-map can not be reconstructed and approximated by a neural network with accuracy close to that obtained for the Logistic map (see [4]).

5. NUMERICAL SCHEME FOR DETERMINATION OF THE LOCATION OF THE SINGULAR POINT

Due to discontinuities of the derivatives at the singular point, the approximation of time series in the neighborhood of this point using standard approach (based on smooth functions) gives large errors.

Here we present a numerical scheme for the accurate calculation of the position of the singular point based on specially chosen basis functions.

Let $G = \{(x_k, y_k), k = \overline{1, L}\}$ be the set of arguments and values of the unknown function $y(t)$ we are searching for in the neighborhood of the singular point. In our case $y(x_k) = y_k = x_{k+1}$. Let

us choose the set of basis functions $\{f_l(t, z), l = \overline{1, 2N_0}\}$ as follows:

$$f_l(t, z) = \begin{cases} 0, & t > z; \\ (t-z)^l, & t \leq z; \end{cases} \quad l = 1, \dots, N_0;$$

$$f_l(t, z) = \begin{cases} 0, & t \leq z; \\ (t-z)^{l-N_0}, & t > z, \end{cases} \quad l = N_0 + 1, \dots, 2N_0.$$

Let $F(\cdot)$ be the functional:

$$F(\alpha_0, \dots, \alpha_{2N_0}, z) = \sum_{k=1}^L \left| \sum_{l=0}^{2N_0} \alpha_l f_l(x_k, z) - x_{k+1} \right|^2. \quad (8)$$

Here $f_0(t) = 1$, the constant function. The minimum of $F(\cdot)$ is reached for $\{\alpha_i\}$ and z satisfying the following equations:

$$\frac{dF(\alpha_0, \dots, \alpha_{2N_0}, z)}{d\alpha_i} = 0, \quad i = \overline{0, 2N_0};$$

$$\frac{dF(\alpha_0, \dots, \alpha_{2N_0}, z)}{dz} = 0. \quad (9)$$

The nonlinear system (9) can be solved numerically applying Newton's method. The set $\{\alpha_i^k, i = \overline{0, 2N_0}\}$ and $\{z^k\}$ corresponding to the k -th

iteration satisfy the following linear equations:

$$\begin{aligned}
 & \sum_{j=1}^{2N_0} \Delta \alpha_j^k \frac{d^2 F(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{d\alpha_j d\alpha_i} \\
 & + \Delta z^k \frac{d^2 F(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{d\alpha_j dz} \\
 & = - \frac{dF(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{d\alpha_i}, \\
 & \sum_{j=1}^{2N_0} \Delta \alpha_j^k \frac{d^2 F(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{d\alpha_j dz} \\
 & + \Delta z^k \frac{d^2 F(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{d^2 z} \\
 & = - \frac{dF(\alpha_0^k, \dots, \alpha_{2N_0}^k, z^k)}{dz},
 \end{aligned} \tag{10}$$

where $\Delta \alpha_j^k = \alpha_j^{k+1} - \alpha_j^k$, $\Delta z^k = z^{k+1} - z^k$. The numbers $\{\alpha_i^0, i = \overline{0, 2N_0}\}$, z^0 are arbitrary chosen initial values.

Let x^* be the calculated value of the singular point, and let z^* and $\{\alpha_i^*, i = \overline{0, 2N_0}\}$ be the solution of system (10) obtained from the iterations.

Then the function

$$\hat{y}(t) = \sum_{l=0}^{2N_0} \alpha_l^* f_l(t, z^*) \tag{11}$$

gives the approximation of function $y(t)$ in the neighbourhood of point x^* .

We applied the above described procedure to the sample, which was prepared on the basis of the original time series containing N couples: $\{x_i, x_{i+1}, i = \overline{1, N}\}$. The couples forming the new sample were selected using the following condition: $|x_i| \leq \Delta$, where $\Delta = 0.2$. The value of Δ was chosen empirically aiming to calculate the singular point position with minimal error.

TABLE I Results of the calculations for time series of different size N

n/n	N	$L(N)$	x^*
1	125	34	0.2968E-01
2	250	63	0.3712E-01
3	500	126	0.3769E-01
4	1000	242	0.1694E-03
5	2000	455	-0.11357E-03
6	5000	1064	0.11394E-03

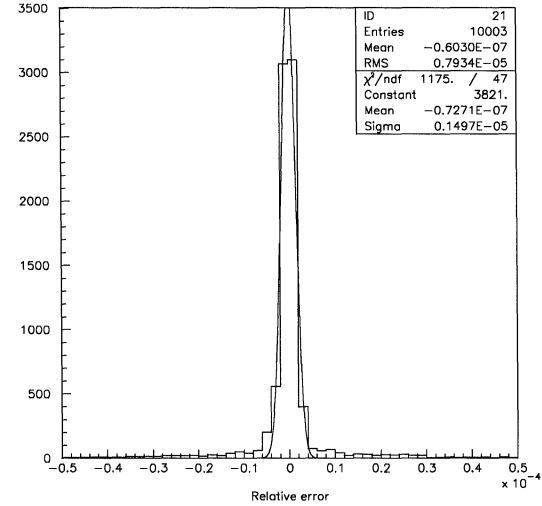
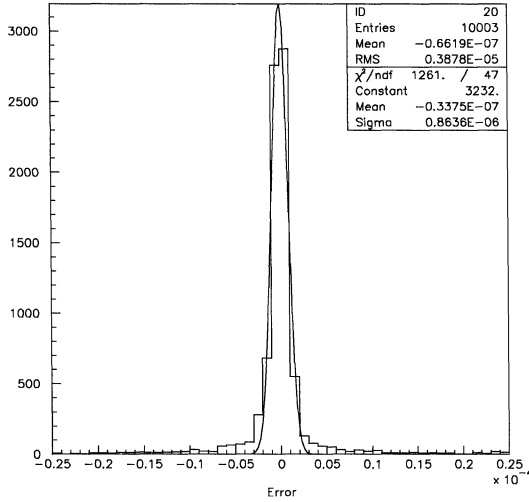


FIGURE 8 Distributions of *Error* and *Relative error* for the WNN network: only first part (see text) of the Cusp map has been used.

In Table I we present the results of our calculations for time series of different size N and for $N_0=3$. Here $L(N)$ is the size of a new sample formed from the initial time series of N elements, and x^* is the calculated value of the singular point.

6. CONCLUSION

When the location of the singular point is determined with a reliable accuracy, we may sub-divide the whole time series into two sets: the *first* one, for which the arguments less than the singular point value, and the *second* one with arguments greater than the singular point value. Then, these two sets can be approximated independently and with higher accuracy using traditional numerical schemes or applying neural networks [4].

We applied the above described algorithm to the Cusp map with 2000 couples and selected 2 sets: the *first* set included 1436 pairs, and the *second* had 564 pairs. Then the *first* set (as it included more patterns) was processed with the help of the WNN network. Figure 8 shows the distributions of the deviations and the relative deviations for the WNN network. These distributions show that the *first* set can be significantly better approximated compared with the whole time series.

Our work is summarized as follows:

- (1) We used neural networks (MLP, CNN, WNN) for a qualitative identification of the dynamical process underlying a complex time series.
- (2) We roughly estimated the location of the singular point.
- (3) We developed a numerical algorithm for the accurate determination of the location of the singular point.
- (4) Afterwards, we proceed to the identification of the smooth dynamics between the singularity.

Acknowledgments

We acknowledge the European Commission for partial support under the ESPRIT project 21042-CTIAC and the Belgian Government through the Inter-university Attraction Poles. We are grateful to Prof. I. Prigogine for encouragement and support.

References

- [1] Lapedes, A. and Farber, R. (1987). "Nonlinear Signal Processing using Neural Networks: Prediction and System Modeling", Los Alamos Report LA-UR 87-2662.
- [2] Denby, B., "Tutorial on Neural Networks Applications in High Energy Physics: The 1992 Perspective". In *Proc. of the Second International Workshop on "Software Engineering, Artificial Intelligence and Expert Systems in High Energy Physics"*. January 13–18, 1992 L'Agelaude France-Telecom La Londe-les-Maures (France). New Computing Techniques in Physics Research II, Edited by Perret-Gallix, D., World Scientific, p. 287.
- [3] Bonushkina, A. Yu., Ivanov, V. V. and Zrellov, P. V. (1997). "Multidimensional Data Analysis Based on the ω_n^k -Criteria and Multilayer Perceptron", *Int. J. Comput. Math. Appl.*, **34**(7–8), 677–686.
- [4] Akritas, P., Antoniou, I. and Ivanov, V. V. (2000). "Identification and Prediction of Discrete Chaotic Maps Applying a Chebyshev Neural Network", *Chaos, Solitons and Fractals*, **11**, 337–344.
- [5] Zhang, Q. and Benveniste, A. (1992). "Wavelet Networks", *IEEE Trans. on Neural Networks*, **3**, 889–898.
- [6] Chui, Ch. (1992). "An Introduction to Wavelets", Academic Press, Inc.
- [7] Daubechies, I. (1998). "Orthonormal Bases of Compactly Supported Wavelets", *Comm. on Pure and Appl. Math.*, **XLI**, 909–996.
- [8] Szu, H. H., Yang, X.-Y., Telfer, B. A. and Sheng, Y. (1993). "Neural Network and Wavelet Transform for Scale-Invariant Data Classification", *Phys. Rev. E*, **48**(2), 1497–1501.
- [9] Cao, L., Hong, Y., Fang, H. and Ha, G. (1995). "Predicting Chaotic Time Series with Wavelet Networks", *Physica*, **D85**, 225–238.
- [10] Bonushkina, A. Yu., Ivanov, V. V., Selin, A. and Zrellov, P. V., "Wavelet Neural Network and its Application to the ECG Analysis" (under preparation).
- [11] Walter, G. G., "Wavelets and Other Orthogonal Systems with Applications", CRC Press, Boca Raton, Florida.
- [12] Hemmer, P. C. (1984). "The Exact Invariant Density for a Cusp-Shaped Return Map", *J. Phys. A: Math. Gen.*, **17**, L247–L249.
- [13] Antoniou, I., Shkarin, S. and Yarevsky, E., "Exactness and Probabilistic Description of the family of Cusp Maps", *Physica A* (submitted).
- [14] Rokhlin, V. (1964). "Exact Endomorphisms of Lebesgue Spaces", *Am. Math. Soc. Transl.*, **39**, 1–36; *Izv.*

- Akad. Nauk. SSSR Ser. Mat.*, **25**, 499–530 (1961), (in Russian).
- [15] Cornfeld, I., Fomin, S. and Sinai, Ya., “*Ergodic Theory*”, Springer-Verlag, Berlin, 1992.
- [16] Lighthill, J. (1986). “The recently recognized failure of predictability in Newtonian Dynamics”, *Proc. Roy. Soc.*, London, **A407**, 35–50.
- [17] CERN Library: Algorithm A105.
- [18] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., “Learning Internal Representations by Error Propagation” In: Rumelhart, D. E. and McClelland, J. L. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, **1**, Foundations. MIT Press (1986).
- [19] Lönnblad, L., Peterson, C. and Rognvaldsson, T. (1992). “Pattern Recognition in High-Energy Physics with Artificial Neural Networks: JETNET 2.0”, *Comp. Phys. Commun.*, **70**, p. 167.
- [20] Berezin, I. S. and Zhidkov, N. P. (1959). “*Computing Methods*”, Vol. I (Translated by Blunn, O. M.), Pergamon Press; this is a translation of the original Russian “*Metodi Vychislenii*” published by Fizmatgiz, Moscow, 1959.
- [21] Ueda, M. and Lodha, S., “Wavelet: An Elementary Introduction and Examples”, *Baskin Center for Computer Engineering & Information Sciences*, University of California, Santa Cruz, CA 95064, USA, UCSC-CRL 94–47, January 17, 1995.
- [22] Paty, Y. C. and Krishaprasad, P. S. (1993). “Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformation”, *IEEE Trans. on Neural Networks*, **4**(1).



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

