False-nearest-neighbors algorithm and noise-corrupted time series

Carl Rhodes*
Chemical Engineering, 210-41, California Institute of Technology, Pasadena, California 91125

Manfred Morari†
Institut für Automatik, ETH-Z/ETL, CH-8092 Zürich, Switzerland

(Received 14 March 1996; revised manuscript received 14 January 1997)

The false-nearest-neighbors (FNN) algorithm was originally developed to determine the embedding dimension for autonomous time series. For noise-free computer-generated time series, the algorithm does a good job in predicting the embedding dimension. However, the problem of predicting the embedding dimension when the time-series data are corrupted by noise was not fully examined in the original studies of the FNN algorithm. Here it is shown that with large data sets, even small amounts of noise can lead to incorrect prediction of the embedding dimension. Surprisingly, as the length of the time series analyzed by FNN grows larger, the cause of incorrect prediction becomes more pronounced. An analysis of the effect of noise on the FNN algorithm and a solution for dealing with the effects of noise are given here. Some results on the theoretically correct choice of the FNN threshold are also presented. [S1063-651X(97)01605-X]

PACS number(s): 07.05.Kf, 05.45.+b

I. THE FALSE-NEAREST-NEIGHBORS ALGORITHM

The false-nearest-neighbors (FNN) algorithm is a tool to determine if a given ''input'' vector contains enough information to predict another ''output'' value directly from properties of the data. More specifically, for the description

$$y(n)=G[x_1(n),x_2(n),\ldots,x_l(n)] \tag{1}$$

FNN answers the question of whether a single-valued function G exists relating the x_i variables and the output variable y for a given data set.

The FNN algorithm was originally developed for determining the number of time-delay coordinates needed to recreate autonomous dynamics [1,2], but FNN has also been extended to examine the problem of determining the proper embedding dimension for input-output dynamics and for inferential measurement selection [3]. More information about the theory of using time-delay coordinates for modeling of input-output systems can be found in [4,5]. For autonomous systems described by state space equations, a scalar output at any point in time can be predicted by a function involving time-delayed versions of the same output. The following relationship holds for autonomous systems if $l>2d$, where d is the dimension of the state space dynamics [6,7]:

$$y(t)=G[y(t-\tau),y(t-2\tau),\ldots,y(t-l\tau)]. \tag{2}$$

However, given only a time series $y(t)$ from an unknown dynamical system, this theoretical result is little help in determining the proper embedding dimension l.

The FNN algorithm was developed to determine the smallest number of time-delay coordinates l needed to determine the output $y(t)$ directly from time-series data. FNN is based on the following fact. If the number of time-delay coordinates l is too small, then points close in the time-delay coordinates may only be close because of projection rather than the dynamics of the system. In this case, points which are close together in the time-delay coordinates may have very different outputs $y(t)$. Points which are only close because of projection are known as *false neighbors*. When l is large enough to represent the dynamics of the system, points which are close in the time-delay coordinates will always have outputs $y(t)$ which are ''close'' in some sense.

Here is a short outline of the FNN algorithm for autonomous systems.

(1) Identify the closest point (in the Euclidean sense) to a given point in the time-delay coordinates. That is, for a given time-delay point

$$\mathbf{z}_l(k)=[y(k-\tau),\ldots,y(k-l\tau)] \tag{3}$$

find the point $\mathbf{z}_l(j)$ in the data set such that the following distance d is minimized:

$$d=\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2. \tag{4}$$

$\mathbf{z}_l(j)$ is known as the nearest neighbor to $\mathbf{z}_l(k)$.

(2) Determine if the following expression is true or false:

$$\frac{|y(k)-y(j)|}{\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2}\leqslant R, \tag{5}$$

where R is some previously chosen threshold value. If expression (5) is true, then the neighbors are *true* neighbors. If the expression is false, then the neighbors are *false* nearest neighbors.

(3) Continue the algorithm for all points k in the data set. Calculate the percentage of points in the data set which have false nearest neighbors.

(4) Continue the algorithm for increasing l until the percentage of false nearest neighbors drops to zero (or some acceptably small number).

While this single threshold test works quite well for cases where there are ''sufficient data'' to fill out the embedding space, for cases where the distance between nearest neigh-

*Present mailing address: Institut für Automatik, ETH-Z/ETL, CH-8092 Zürich, Switzerland. Electronic address: car@aut.ee.ethz.ch

†Electronic address: morari@aut.ee.ethz.ch

 © 1997 The American Physical Society

bors $\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2$ is large the distance between outputs $|y(k)-y(j)|$ can be quite large and still satisfy the above threshold test [Eq. (5)]. If the distance between the nearest neighbors embedded in the space of outputs and regressors is roughly the same magnitude as the size of the attractor, then the neighbors should also be considered false neighbors. For this reason, a second threshold test which only becomes important in cases of sparse data was also utilized in the original FNN algorithm [1,10].

The second threshold test is defined as

$$\frac{R_{d+1}}{R_A}<A_{\mathrm{tol}}, \tag{6}$$

where

$$R_{d+1}^2=[y(k)-y(j)]^2+\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2^2, \tag{7}$$

$$R_A^2=\frac{1}{N}\sum_{n=1}^{N}[y(n)-\overline{y}]^2, \tag{8}$$

$$\overline{y}=\frac{1}{N}\sum_{n=1}^{N}y(n). \tag{9}$$

The recommended threshold of $A_{\mathrm{tol}}=2$ is used in all of the examples given here. Failing this additional threshold test means that the nearest neighbors are far apart in the extended space of R_{d+1} and that the neighbors should be considered false. Since a failure of the above threshold test implies a failure of the threshold test given in Eq. (5) when nearest neighbors are close [when $\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2^2$ is small], this test is only important when the nearest neighbors are relatively far apart.

II. THEORETICAL CHOICE OF FNN THRESHOLD

Assume that the minimal representation in the time-delay coordinates is known. Let n be the smallest integer for which there exists a function G uniquely determining the output coordinate $y(t)$ for all time-delay vectors.

$$y(t)=G[y(t-\tau),y(t-2\tau),\ldots,y(t-n\tau)] \tag{10}$$

$$=G[\mathbf{z}_n(t)]. \tag{11}$$

Also assume that the function G is known. How should the threshold R for the ratio test be chosen when the time series is noise free? The following choice of threshold should be made if the data are sufficiently ''dense'' over the region of interest.

Lemma 1. $R=\max_t\|DG(\mathbf{z}_n(t))\|_2$, where $DG(\mathbf{x})$ is the Jacobian of the function G at the point $\mathbf{x}$, is the smallest choice of the threshold which will give 0% FNN at the proper dimension n for all data sets.

Proof: If sufficient data are available, the nearest neighbor to each point will be in a region where a local linear approximation to the function G can be made. Using a linear approximation around the point $\mathbf{z}_n(k)$, the output of the nearest neighbor $y(j)$ is given by

$$y(k)-y(j)=DG(\mathbf{z}_n(k))[\mathbf{z}_n(k)-\mathbf{z}_n(j)]+O([\mathbf{z}_n(k)-\mathbf{z}_n(j)]^2). \tag{12}$$

Ignoring the higher order terms, by the Cauchy-Schwarz inequality we know

$$|y(k)-y(j)|\leqslant\|DG(\mathbf{z}_n(k))\|_2\|\mathbf{z}_n(k)-\mathbf{z}_n(j)\|_2, \tag{13}$$

$$\frac{|y(k)-y(j)|}{\|\mathbf{z}_n(k)-\mathbf{z}_n(j)\|^2}\leqslant\|DG(\mathbf{z}_n(k))\|_2, \tag{14}$$

$$\frac{|y(k)-y(j)|}{\|\mathbf{z}_n(k)-\mathbf{z}_n(j)\|^2}\leqslant\max_t\|DG(\mathbf{z}_n(t))\|_2 \quad \forall k \text{ and nearest neighbor } j. \tag{15}$$

For any choice of R smaller than $\max_t\|DG(\mathbf{z}_n(t))\|_2$, the gain of the system may cause the FNN algorithm to record a false nearest neighbor [see Eq. (5)] at the time-delay point $\mathbf{z}_n(t)$ if the nearest neighbor happens to make the equation $\alpha DG(\mathbf{z}_n(t))=[\mathbf{z}_n(t)-\mathbf{z}_n(j)]$ true for some $\alpha\in\mathbb{R}$. In other words, the equality of Eq. (15) will hold when the nearest neighbor to $\mathbf{z}_n(t)$ happens to lie in the direction of maximum gain of the Jacobian

Therefore a proper lower bound on the choice of the threshold R for the FNN algorithm is dependent on knowledge of the function which needs to be identified. However, if an infinite amount of data is available the threshold R can be chosen arbitrarily large.

Lemma 2. For an amount of data approaching infinity, any finite threshold value R will lead to a nonzero percentage of FNN when the embedding dimension is smaller than n.

Proof: The function G has a unique output for all inputs only for values of embedding dimension $l\geqslant n$. For $l<n$, the implied function relating the time-delay coordinates to the output may have multiple outputs for a given time-delay vector. Take a point $\mathbf{z}_n(k)$ where the function does not have a unique output and a sequence of points $\mathbf{z}_n(j_i)$ where $\lim_{i\to\infty}\mathbf{z}_n(j_i)=\mathbf{z}_n(k)$ but $\lim_{i\to\infty}y(j_i)-y(k)=\kappa\neq 0$. Now using the FNN threshold [Eq. (5)] we see

$$\lim_{i\to\infty}\frac{|y(k)-y(j_i)|}{\|\mathbf{z}_l(k)-\mathbf{z}_l(j_i)\|_2}=\infty \tag{16}$$

or that the threshold R must be infinitely large for the threshold inequality [Eq. (5)] to be true. ■

When analyzing real, finite time series the results of these lemmas are not helpful. However, further analysis of the results of Lemma 2 leads us to believe that when there are more points in a data set, a larger threshold value can be chosen safely. On the other hand, if the threshold is chosen too large and nearest neighbors are not close in the time-delay space, outputs may be in completely different regions of the attractor and still be considered true neighbors by the ratio test. This problem will not be encountered with large

data sets where the space is ''filled out'' with data.

Another fact to remember when choosing the threshold value R is that points which are false neighbors tend to move very far apart in the output space. This is a consequence of the previously mentioned fact that false neighbors are only close because of projection. It has been our experience (as well as that of Abarbanel *et al.*) that the percentage of false nearest neighbors tends to remain fairly constant for a fairly wide range of $R(10 \leqslant R \leqslant 30)$. Once R is large enough to account for the local gain of the system, the false neighbors move far enough apart to cause the ratio test to fail for even fairly large values of R.

III. THE EFFECT OF NOISE

Now consider the case where the data set to be examined is corrupted with noise. Assume that for each time t, what we observe is $y^m(t)=y(t)+\delta$ where δ is magnitude bounded ($|\delta| \leqslant \delta_\infty$). The time-delay coordinates will also be corrupted with noise, $\mathbf{z}_n^m(k)=[y^m(k-\tau),\ldots,y^m(k-n\tau)] =[y(k-\tau)+\delta_1,\ldots,y(k-n\tau)+\delta_n]$,

$$|y^m(k)-y^m(j)| \leqslant |G(\mathbf{z}_n(k))-G(\mathbf{z}_n(j))|+2\delta_\infty \tag{17}$$

$$\leqslant \|DG(\mathbf{z}_n(k))\|_2 \|\mathbf{z}_n(k)-\mathbf{z}_n(j)\|_2+2\delta_\infty \tag{18}$$

$$\leqslant \|DG(\mathbf{z}_n(k))\|_2 [\|\mathbf{z}_n^m(k)-\mathbf{z}_n^m(j)\|_2+2\sqrt{n}\delta_\infty]+2\delta_\infty \tag{19}$$

$$\leqslant \max_t \|DG(\mathbf{z}_n(t))\|_2 [\|\mathbf{z}_n^m(k)-\mathbf{z}_n^m(j)\|_2+2\sqrt{n}\delta_\infty]+2\delta_\infty . \tag{20}$$

Converting this final result into a form similar to the threshold function,

$$\frac{|y^m(k)-y^m(j)|}{\|\mathbf{z}_n^m(k)-\mathbf{z}_n^m(j)\|_2} \leqslant \max_t \|DG(\mathbf{z}_n(t))\|_2 + \frac{2\delta_\infty(\sqrt{n}\ \max_t \|DG(\mathbf{z}_n(t))\|_2+1)}{\|\mathbf{z}_n^m(k)-\mathbf{z}_n^m(j)\|_2}. \tag{21}$$

Note that there are two terms on the right hand side of the above equation. The first term accounts for the maximum possible local gain of the system at a point $\mathbf{z}_n(k)$, and the second term is due only to the effects of noise. Also note that the second term is inversely proportional to the separation of the nearest neighbors in the time-delay coordinates.

If a ratio test with a threshold independent of the density of the points is used to analyze a time series which contains noise [such as the original FNN test given in Eq. (5)], the percentage of false nearest neighbors arising only from noise should increase proportionally with the density of the points in the system.

This effect can be seen by using a time series from a well studied example. Let us use the FNN algorithm to examine data from integration of the Lorenz equations:

$$\begin{aligned} \dot{x} &= \sigma(y-x), \\ \dot{y} &= -xz+\rho x-y, \\ \dot{z} &= xy-\beta z, \\ \sigma &= 10, \quad \rho = 45.92, \quad \beta = 4. \end{aligned} \tag{22}$$

A 50 000 point scalar time series was found by taking the x output from the integration of the above equations using a sampling time of 0.1. Two noisy data sets were also developed by adding uniformly distributed noise of maximum absolute value 0.5 and 1.0, respectively, to the original data set (the x variable of the Lorenz signal has a standard deviation of 12.36).

The FNN algorithm was utilized on each of the three data sets using time series of three different lengths (500, 5000, 50 000 points). The standard choice of threshold recommended in [1] ($R=17.3$) was used. The results of the algorithm can be seen in Figs. 1–3. When the percentage of FNN drops to 0, the embedding dimension is large enough to represent the dynamics.

For noise-free data, the percentage of false nearest neighbors for a given dimension remains constant as the amount of data increases. However, for data corrupted with noise, the percentage of false nearest neighbors for a given embedding dimension *increases* as the amount of data is increased. For the FNN algorithm working on noise-corrupted data, more data are not necessarily better. This is contrary to the common belief in identification that more data lead to more accurate results. Larger time series lead to ''false'' false nearest neighbors, neighbors which are a result of noise corruption rather than an incorrect embedding dimension.

IV. A POSSIBLE SOLUTION

A possible solution to this problem is to account for noise by using a FNN threshold which includes both a constant term (as in the original FNN formulation) and another term to account for noise.

Instead of using Eq. (5) for the threshold, a logical test for nearest neighbors based on the previous analysis is

$$\frac{|y(k)-y(j)|}{\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2} \leqslant R+\frac{2\epsilon R\sqrt{l}+2\epsilon}{\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2}. \tag{23}$$

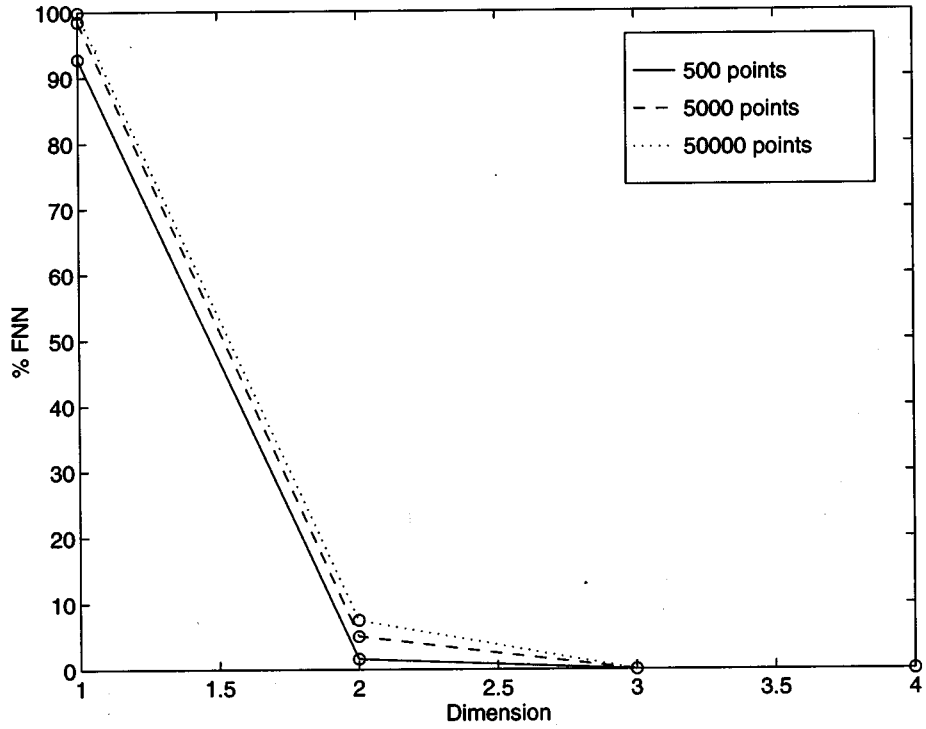

FIG. 1. FNN plots for noise-free Lorenz time series.

By examining Eq. (23) further, it can be seen that the threshold test has two distinct limits depending on the size of the term $\|\mathbf{z}_l(k)-\mathbf{z}_l(j)\|_2$. When the distance between the nearest neighbors is relatively large, the first term on the right hand side of the inequality will dominate and this ratio test is equivalent to the original FNN test. When the distance between nearest neighbors approaches zero, the second term on the right hand side dominates. In this limit, the new test is equivalent to asking whether the two observed outputs lie within a certain noise threshold of each other for identical inputs.

The main problem with this choice of threshold is that there are now two variables which must be tuned, namely, R and ϵ. Optimally we should choose $\epsilon=\delta_\infty$ and R as suggested before. However, for time series where δ_∞ is unknown, some physical arguments based on the size of ex-

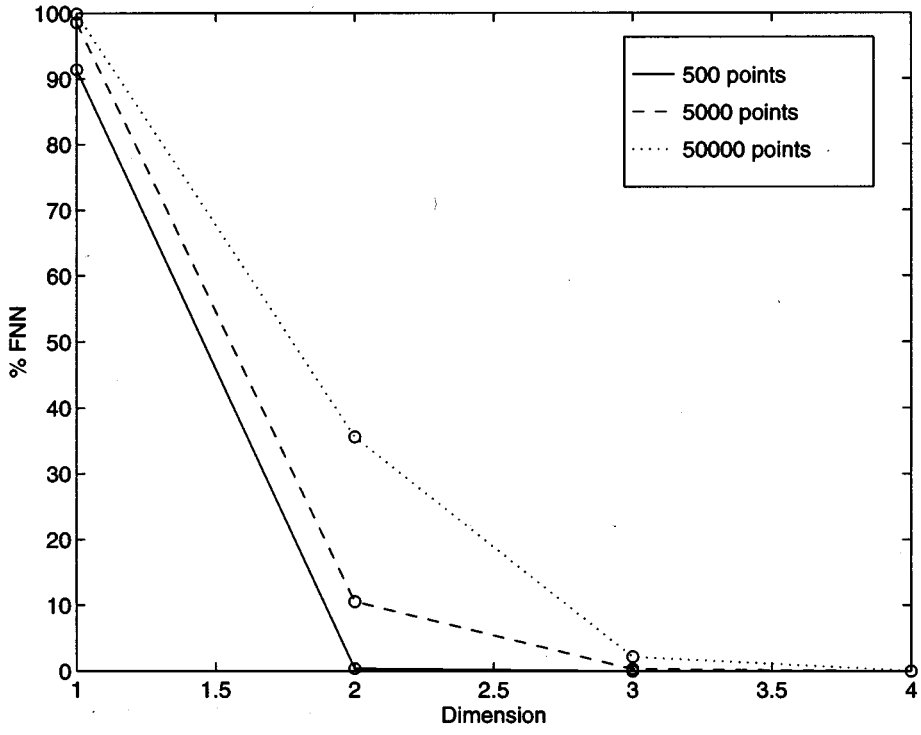

FIG. 2. FNN plots for Lorenz time series with magnitude 0.5 noise.

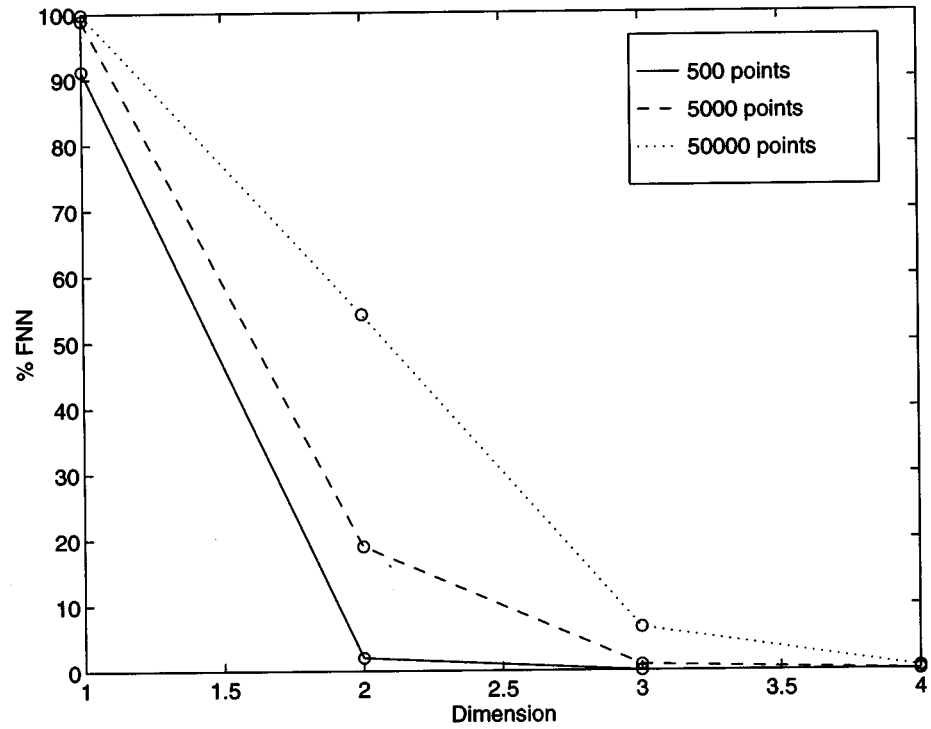

FIG. 3. FNN plots for Lorenz time series with magnitude 1.0 noise.

pected measurement noise can be made for determining ϵ. The results of the algorithm should be fairly independent of the choice of R over a large range as before. Using this modified test for false nearest neighbors, the problem of an increasing number of false nearest neighbors with increasing amounts of data will not be encountered in noisy data sets.

In fact, as can be seen in the following example, by utilizing the new ratio test even a small ϵ term can dramatically affect the results of the FNN algorithm. The same data were examined with the FNN algorithm substituting Eq. (23) for the standard threshold inequality Eq. (5). The thresholds were set with $R=17.3$ (as before) and $\epsilon=0.05$. Figures 5 and 6 show the results of the modified algorithm on the noisy data. These figures can be compared to Figures 3 and 4, respectively, which are the results of the original FNN algorithm.

Notice that ϵ is well below the recommended δ_∞ values of 0.5 and 1.0 for the two data sets. However, the modified algorithm has the desired results of distinguishing those false nearest neighbors which are the result of noise from those

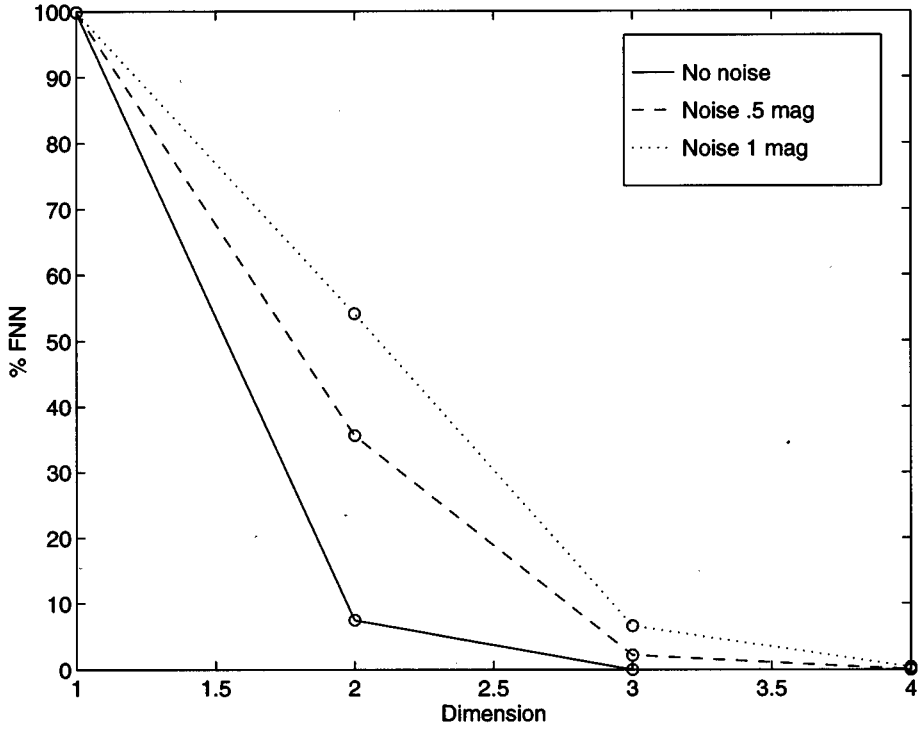

FIG. 4. FNN plots for the three different Lorenz time series of length 50 000.

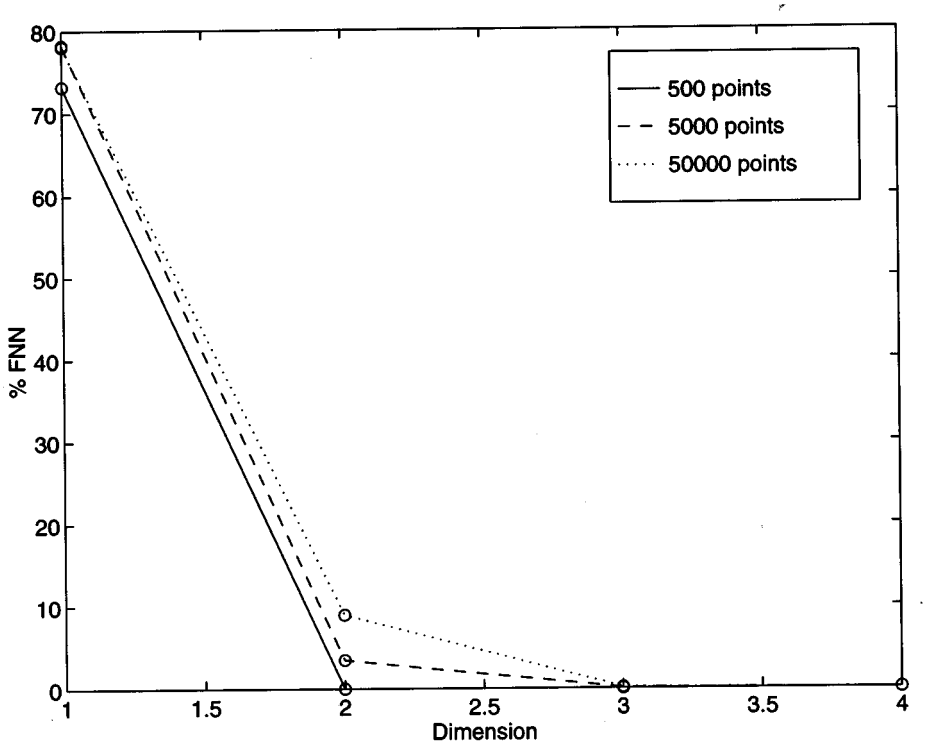

FIG. 5. FNN plots for Lorenz time series with magnitude 1.0 noise, modified algorithm

which are the result of an incorrect embedding dimension. The reason that such a small ϵ works well in this case may be both because a conservative choice of R is made and the new FNN threshold is conservative in construction. False nearest neighbors which are the result of noise in the original algorithm are not counted as false nearest neighbors with this modified formulation.

For the data set of size 500, the modified algorithm incorrectly predicts an embedding of dimension 2 for the noisy data set. However, for small data sets problems arising from noise are not encountered when using the original FNN ratio test. For larger data sets, the problem with increasing data causing false nearest neighbors from noise is no longer present with the new ratio test and correct prediction of the embedding dimension is again possible.

V. ADDITIONAL EXAMPLES

In this section, two additional examples will be presented which illustrate the advantages of the proposed threshold

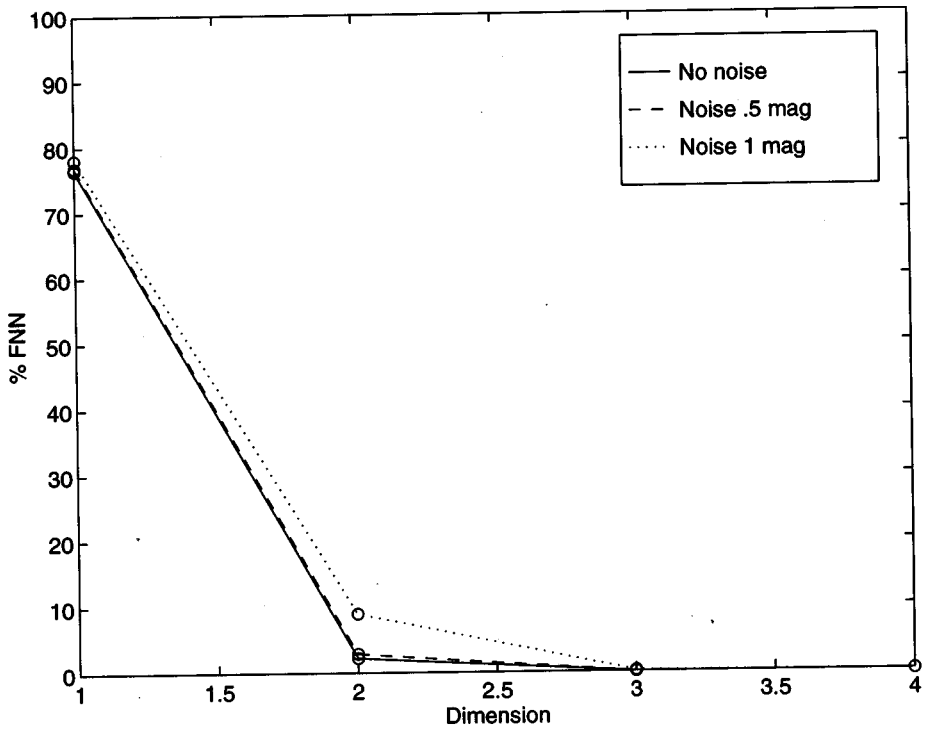

FIG. 6. FNN plots for modified algorithm (Lorenz time series, length 50 000)

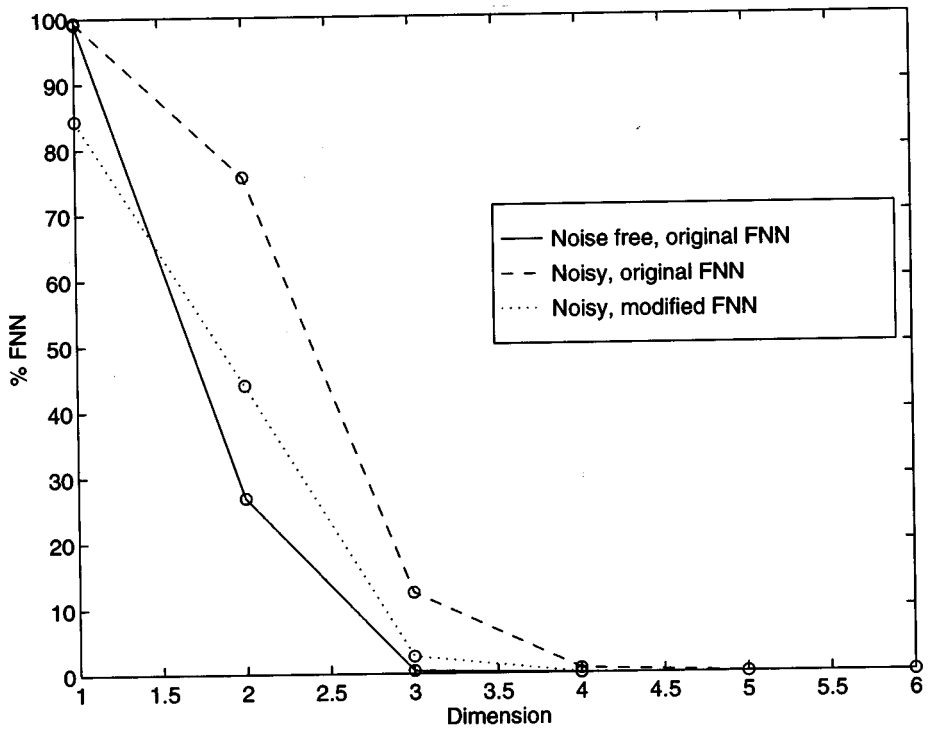

FIG. 7. FNN plots for Mackey-Glass time series.

test. The first example consists of data from the Mackey-Glass delay-differential equation, and the second example examines white noise time-series data. The results given by the new threshold test are compared with the results of the standard FNN ratio test.

A. Mackey-Glass delay-differential equation

The Mackey-Glass equation is the delay-differential equation given as

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\Delta)}{1+x(t-\Delta)^{10}}. \tag{24}$$

This delay-differential equation exhibits chaotic behavior over a wide range of delay parameters Δ. For this study, a time series with 50 000 points was created by integrating Eq. (24) with $\Delta = 17$. The sampling time used to create the discrete time series is 6, as is suggested by a previous study of Casdagli [8].

In addition to the noise-free time series, a noise-corrupted time series is created by adding normally distributed noise with a standard deviation of 0.03 to the original time series. These two time series are then analyzed by the original and modified FNN algorithms, and the results are presented in Fig. 7 and Table I. For both the original and modified FNN algorithm the threshold $R = 17.3$ is used and in the modified FNN algorithm the threshold $\epsilon = 0.001$ is used. The results of the original FNN algorithm seem to suggest that the proper embedding dimension of the noise-free time series is 4. When the noise-corrupted time series is examined by the original FNN algorithm the proper embedding dimension appears to be 5. Again, by applying the modified threshold test, the proper embedding dimension (4) of the noise-corrupted time series can be recovered assuming one considers 0.05% false nearest neighbors is close enough to zero for the purpose of determining the embedding dimension.

What is especially important is that the modified FNN algorithm finds nearly the same percentage of false nearest neighbors for those embedding dimensions where the original FNN algorithm gives a small number of false nearest neighbors. This is important because accurate prediction of the percentage of false nearest neighbors is crucial when the percentage of false neighbors is small.

B. White noise

In order to confirm that the FNN threshold test presented here does not give spurious results, both the original and new FNN threshold tests are applied to a time series consisting of white noise. A normally distributed white noise time series

TABLE I. FNN analysis of data from Mackey-Glass equation.

	Embedding Dimension					
% FNN	1	2	3	4	5	6
Noise-free data, original FNN	99.12	26.81	0.42	0	0	0
Noisy data, original FNN	99.37	75.70	12.37	0.71	0.03	0.01
Noisy data, modified FNN	84.36	44.01	2.65	0.05	0	0

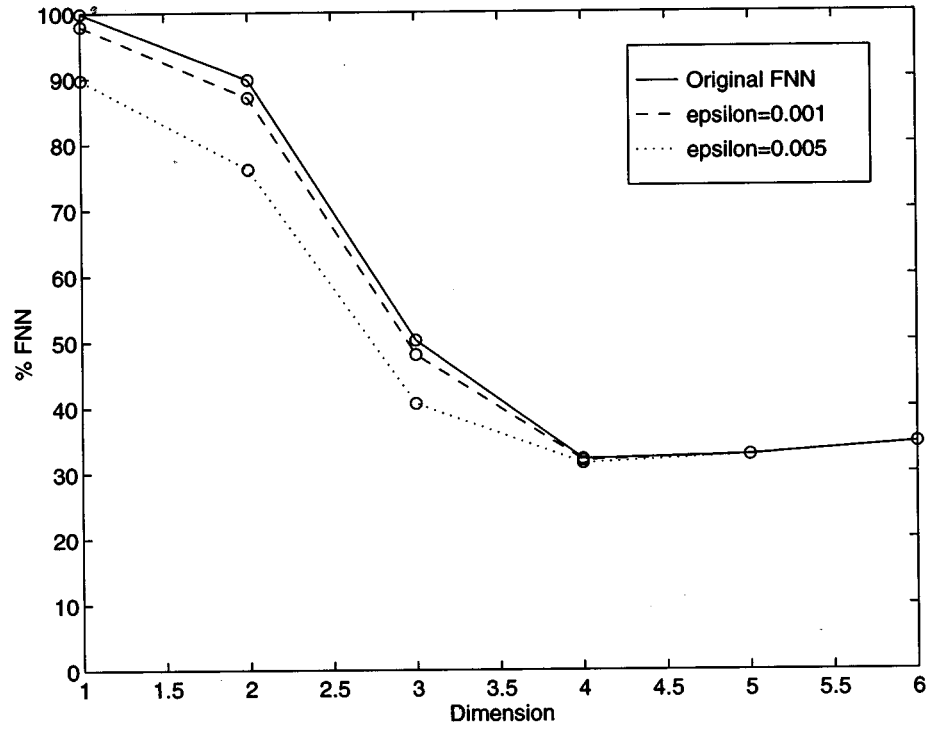

FIG. 8. FNN plots for white noise with all FNN tests.

of length 50 000 was generated by the MATLAB command RANDN [9]. The variance of the time series is one and the mean is zero.

Again, both the original FNN algorithm and the modified threshold test were applied to the time series with the threshold $R=17.3$. For the modified threshold test, thresholds $\epsilon=0.001$ and 0.005 were used. The results of the FNN analysis are given in Fig. 8. It appears that qualitatively the results are identical for the two different FNN threshold tests. More importantly, none of the tests predict that the time series is deterministic. However, for dimensions larger than 3 a large number of the false nearest neighbors come as a result of the R_{d+1} threshold test [Eq. (6)]. To be sure the results of the original threshold test [Eq. (5)] are not affected by the modification, a second study was conducted excluding the R_{d+1} threshold test. The results of the previous examples (Lorenz, Mackey-Glass) are not affected by excluding the R_{d+1} threshold test [Eq. (6)].

When the R_{d+1} test is excluded (Fig. 9), the percentage of false nearest neighbors for embedding dimensions 4 and

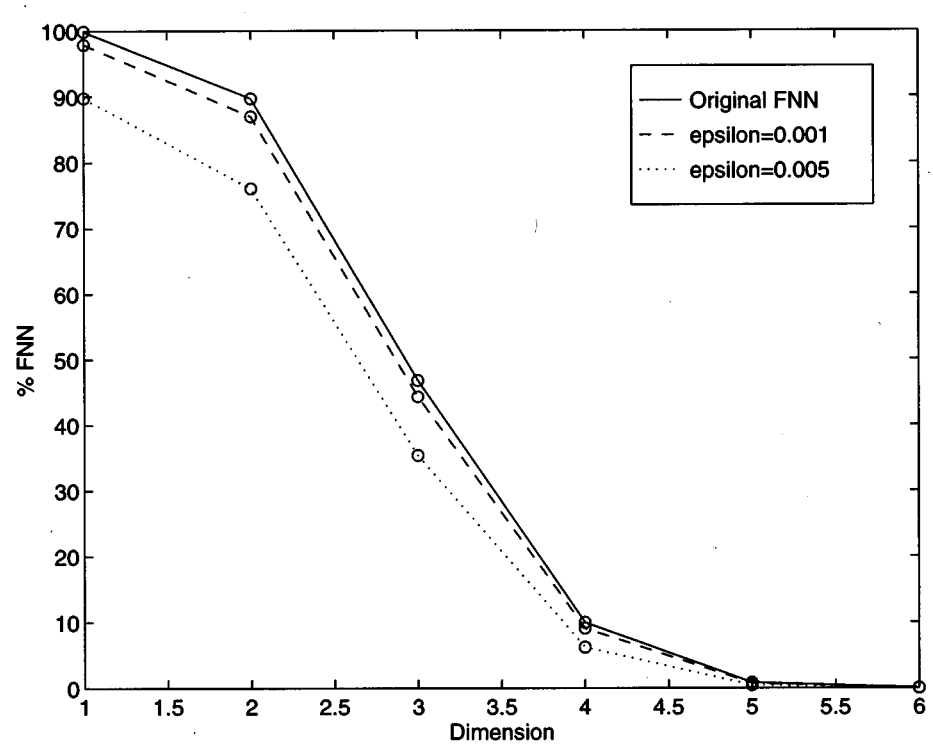

FIG. 9. FNN plots for white noise, excluding R_{d+1} distance test.

larger is significantly different than those given in Fig. 8. However, the results of FNN with the original [Eq. (5)] and modified [Eq. (23)] threshold tests are nearly identical. The modified FNN threshold test does not significantly change the percentage of false neighbors given by the original FNN algorithm.

VI. CONCLUSIONS

The problem of analyzing noisy time series with the FNN algorithm has been discussed and illustrated using data from the Lorenz attractor. The problem of false nearest neighbors which arise from noise rather than incorrect embedding dimension is one which will be encountered when using FNN to analyze time series from physical systems. A ratio test which solves this problem is proposed. However, an easy method of determining the correct choice of thresholds is a problem which remains to be solved (as it does with the original FNN algorithm). Some theoretical results and other guidelines are given to aid the proper choice of the R and ϵ thresholds. The modified threshold test is then applied to time-series data from the Mackey-Glass equation and to a white noise time series and the results are analyzed.

ACKNOWLEDGMENT

Partial support from the Department of Energy is gratefully acknowledged.

[1] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, Rev. Mod. Phys. **65**, 1331 (1993).

[2] H. D. I. Abarbanel and M. B. Kennel, Phys. Rev. E **47**, 3057 (1993).

[3] C. Rhodes and M. Morari, in *Proceedings of the American Control Conference, Seattle, 1995* (American Automatic Control Council, Evanston, IL, 1995), pp. 2190–2195.

[4] M. Casdagli, in *Nonlinear Modeling and Forecasting*, edited by M. Casdagli and S. Eubank, *A Proceedings Volume in the Santa Fe Institute Studies in the Sciences of Complexity* (Addison-Wesley, Reading, MA, 1992), Vol. XII, pp. 265–281.

[5] A. Poncet, J. L. Poncet, and G. S. Moschytz, in *Proceedings of the IEEE Conference on Circuits and Systems, Seattle, 1995* (IEEE, New York, 1995), pp. 1500–1503.

[6] F. Takens, in *Dynamical Systems and Turbulence, Warwick 1980*, edited by D. A. Rand and L. S. Young, Lecture Notes in Mathematics Vol. 898 (Springer-Verlag, Berlin, 1981), pp. 366–381.

[7] T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. **65**, 579 (1991).

[8] M. Casdagli, Physica D **35**, 335 (1989).

[9] *MATLAB Reference Guide* (The Math Works, Inc., Natick, MA, 1992).

[10] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, Phys. Rev. A **45**, 3403 (1992).