NOISE REDUCTION USING GRADIENT DESCENT

Sreya K, Neha Pramod K, Muhammed Anshif K T, Adil Rabeu T, Arshida Mahmooda, Sishu Shankar Muni[1]

[1]School of Digital Sciences, Digital University of Kerala, Thiruvananthapuram, Kerala, India

Abstract

This project explores reducing noise in chaotic and nonlinear time series data using a simple gradient descent method. Real-world data, whether from heart rhythms, climate systems, or financial markets, is often messy and filters out noise without losing the essential underlying patterns is a real challenge. To tackle this, we treat noise reduction as an optimization problem, minimizing the errors between what the system predicts and what we observe. By applying the steepest descent algorithm, we gradually remove noise from the data without disrupting the underlying chaotic structure, preserving essential features such as the attractor's shape and the system's exponents. We tested our approach on the well-known Hénon map and found it significantly improved the clarity of the signal without overfitting. We also show, through theoretical results, that in certain systems (specifically hyperbolic ones), our method can exactly recover the original system behavior. Finally, we compare our method with the traditional Schreiber–Grassberger approach, discussing where each method performs best. Overall, this project presents a practical and reliable way to clean up noisy, complex data without losing what makes the system interesting. Keywords:Noise Reduction, Gradient descent, Optimization algorithm, Gaussian noise

Introduction

Real-world time series data tend to be generated by extremely nonlinear systems, in which very small changes in initial conditions may produce radically different results [1], [2], [3]. Such are heart rhythms, climate patterns, financial markets, and mechanical vibrations, all of which display complicated, chaotic behavior [4], [5], [6]. Classical linear models, although historically significant, are frequently not up to the task of explaining these complex dynamics [7], [8], [9]. The evolution of nonlinear time series analysis has provided the researcher with an even stronger array of tools for investigating and reconstructing the concealed structure from which observations hide [10], [11]. Experimental data, however, are nearly always corrupted by noise-random fluctuations or distortions that mask the actual underlying behavior [12], [13]. Such disturbances could arise from instrumentation limitations, environmental fluctuations, or the inherent stochasticity of the system itself. As a result, there is an urgent need for advanced noise reduction techniques that can retain the key system dynamics while removing extraneous fluctuations [14], [15], [16].

Several approaches have been suggested to solve this problem. Schreiber and Grassberger [17] presented a technique similar to gradient descent, in which noisy trajectories are successively smoothed to clearly expose the structure of the underlying attractor. Other methods, like delay coordinate embedding [18], [19] and local linear prediction [20], [21], take advantage of the internal geometrical organization of dynamical systems to filter signal from noise. A notable contribution is that of shadowing theory, in which one looks for a noiseless trajectory near the noisy data that follows the dynamics of the system to perfection [22], [23], [24]. Methods like strong state-space reconstruction [25], manifold learning [26], [27], and machine learning models [28] have extended further the researcher's options. Improved methods also involve statistical techniques [29], [30] and physics-based techniques such as noise-assisted data analysis [31], which purposefully utilizes stochasticity to enhance signal reconstruction.

The method proposed in this project attempts to combine straightforward noise removal and strict shadowing by reformulating the problem of noise removal as an optimization problem with a mathematical definition. Applying a steepest descent algorithm, the algorithm iteratively optimizes the signal without violating the natural constraints of the system, in an effort to maintain important dynamical properties, e.g., attractor shape, Lyapunov exponents, and correlation dimension [32], [33], [34]. Whereas conventional smoothing techniques have a risk of erasing important nonlinear features or shadowing techniques could overfit sparse or noisy data, the steepest descent algorithm determines the optimal trade-off between fidelity and dynamical consistency [35]. This type of approach has broad applications, ranging from early detection of disease in bio-medical signals [36], [37] to forecasting extreme events [38], [39] and structural health monitoring [40], [41]. With greater convergence of AI and big data into physical sciences, strong nonlinear analysis that can withstand noise will remain essential [42], [43], [44]. Key Contributions are:

- We devised a process that tidies up noisy data by approaching noise reduction as a math problem—a specific optimization problem. Rather than presupposing that clean data is virtually identical to noisy data, we incrementally correct the data using gradient descent to smooth it and make it more accurate. This allows us to preserve meaningful patterns in data and eliminate random noise.
- We tried our approach on a well-known chaotic system known as the Hénon map with a small amount of noise added to it. Once we had applied our method, we could recover the system's original shape and dynamics quite accurately. Even better, we demonstrated that in some systems (hyperbolic systems), our approach does not guess at all; it can exactly replicate the original path of the system. We are not just scrubbing the data; we are restoring what occurred.
- We benchmarked our procedure with an earlier well-established technique (Schreiber–Grassberger technique). Although their approach is more straightforward and quicker, our procedure is steadier in very noisy data. Furthermore, our procedure doesn't have to postulate the clean data approximating the noisy data, and that makes our method more general and realistic under everyday circumstances when data is muddled—such as in healthcare information or climatological measurements.

The following sections will discuss in detail Section 1, where we discuss how we tackle noise reduction as an optimization problem by constructing a cost function out of dynamic errors and attempting to recover a smoother path that more accurately represents the underlying system behavior. Section 2 explains how we implement the gradient descent algorithm to minimize this cost step-by-step. We also provide some mathematics behind it, and along with a practical example on the Hénon map, illustrate how well the method performs based on error metrics. Lastly, in Section 3, we provide the theoretical explanation for our methodology, specifically in hyperbolic systems, showing that it can precisely follow the system's actual path by a method called exact shadowing. Section 4 provides a summary of how our method compares to the widely used Schreiber–Grassberger technique, indicating whether they are successful or not. Finally, in Section 5, we explain why our method performs well in practice- it is strong, simple to implement, and handles noisy, dirty data better than many traditional approaches.

1 Noise Reduction as a Minimisation Problem

Time series is a representation of a sequence of data points collected over time from an experimental dynamical system, such as weather patterns, stock market prices, or physiological signals; it often contains measurement noise. Before attempting any noise reduction, the data is embedded into the reconstructed phase. This process, known as *embedding*, involves constructing vectors using *delay coordinates*, where each vector is formed by taking a data point along with its delayed versions.

Once the embedding is complete, the dynamics of the system can be approximated. However, conventional function approximation techniques typically do not account for measurement noise. As a result, all approximation errors are effectively treated as dynamic noise, making it difficult to distinguish between the true deterministic behavior of the system and the noise introduced during measurement.

Recovering a system's true (deterministic) trajectory from noisy data is generally challenging. Here, the *trajectory* or *system path* refers to the sequence of states, the system follows over time. Although direct recovery is difficult, an alternative approach is possible by focusing on a small neighborhood of the trajectory. It is reasonable to assume that the system's dynamics behave approximately linearly within such a localized region. This assumption is known as *linearization*, where simpler linear relationships locally approximate otherwise complex non-linear behavior. An approximate solution can be sought by assuming that the true deterministic path lies somewhere within this nearly linear neighborhood. This process involves solving a system of linear equations, typically requiring the inversion of a matrix that captures the local dynamics. In essence, the method reconstructs a less noisy version of the true behavior of the system by exploiting the local linearity of the dynamics. Normally, earlier approaches assumed that the real (deterministic) and noisy paths are very close together inside a small, almost straight (linear) neighborhood. However, this is a strong assumption and may not always hold.

A more straightforward approach is suggested here: Instead of assuming that the true path and the noisy path are nearly identical, we instead search for a path within the neighborhood that reduces the noise compared to the original data. It is not necessary for this new path to exactly match the true deterministic path; rather, the important goal is to obtain a trajectory that is smoother, i.e., with less noise. This process can be iteratively repeated: after obtaining a less noisy version of the data, further adjustments are made until no significant noise reduction can be achieved, effectively reaching a minimum. Thus, the noise reduction task is reformulated as a minimization problem: We minimize a function of the dynamic errors for adjustments in the data. Here, the dynamic error represents the deviation between the predicted behavior of the system based on its underlying dynamics and the actual noisy data observed.

Dynamic error is defined as:

$$\epsilon_i = x_i - F(x_{i-1}, x_{i-2}, \dots, x_{i-d}) \tag{1}$$

The total error can be expressed as a cost function H.

$$H = \sum_{i=d+1}^{n} \epsilon_i^2. \tag{2}$$

To minimize the cost function H, a trajectory that closely follows the original data, is determined. Given that multiple solutions can minimize The goal is to select the one that accurately reflects the true data by applying an efficient optimization method.

2 Gradient Descent

The gradient descent method is a technique for minimizing nonlinear functions and is widely used in optimization problems. The fundamental idea of the algorithm is to take small steps in the direction where the function decreases the fastest, which is indicated by the negative of the gradient.

The update rule for gradient descent is given by:

$$\mathbf{x} = \mathbf{x} - \delta \nabla H \tag{3}$$

Where $\mathbf{x}$ represents the current trajectory (the parameter set being optimized), ∇H is the gradient of the cost function H, indicating the direction of steepest ascent,δ is a small positive constant known as the step size, controlling how large each step is.

In this context, $\mathbf{x}$ denotes the system trajectory we aim to adjust to achieve noise reduction. The gradient ∇H provides the necessary information on how the error or cost changes for small changes in the trajectory, guiding how the adjustments should be made. By iteratively updating $\mathbf{x}$ using the rule above, the algorithm progressively reduces the noise in the data. The step size δ ensures that the updates are gradual enough to approach the minimum of the cost function without overshooting. To apply gradient descent to the noise reduction problem described earlier, it is necessary to calculate ∇H explicitly in terms of the system function F, the trajectory points x_i, and the measurement errors ϵ_i.

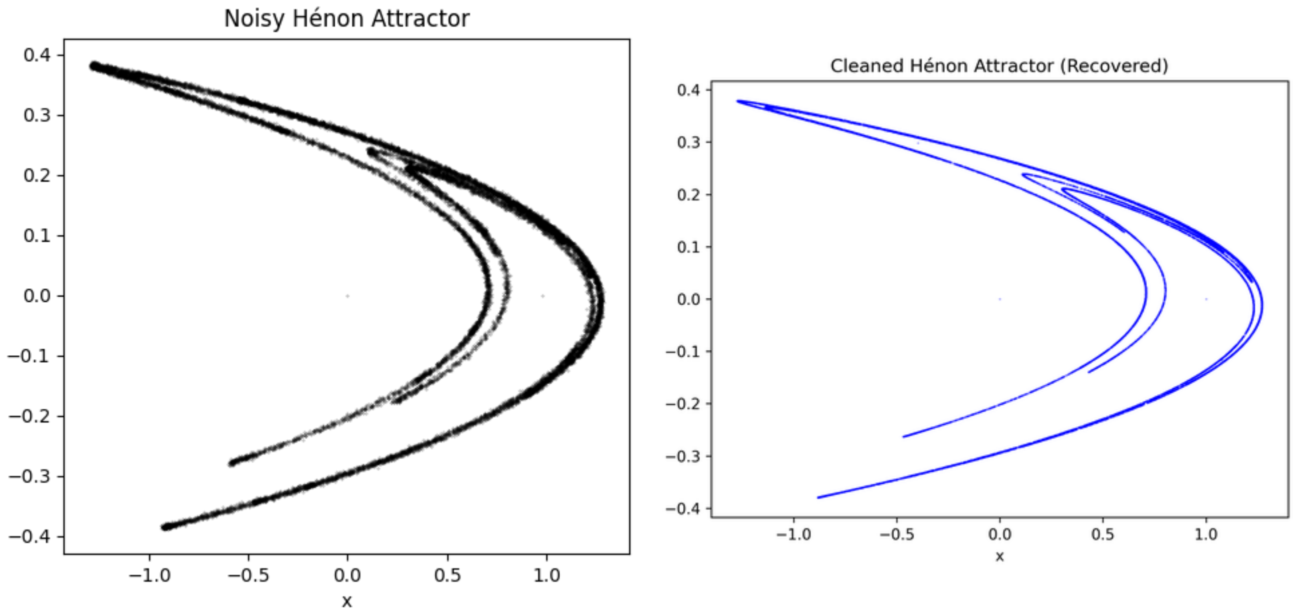
$$\frac{\partial H}{\partial x_i} = 2 \times \left(\epsilon_i - \frac{\partial F_{i+1}}{\partial x_i}\epsilon_{i+1} - \frac{\partial F_{i+2}}{\partial x_i}\epsilon_{i+2} - \cdots - \frac{\partial F_{i+d}}{\partial x_i}\epsilon_{i+d} \right) \tag{4}$$

To demonstrate the effectiveness of this method, the algorithm was applied to a time series, x_i, of 1000 points of data from the Hénon map:

$$x_n = 1 + y_{n-1} - a x_{n-1}^2, \tag{5}$$
$$y_n = b x_{n-1}, \tag{6}$$

with $a = 1.4$ and $b = 0.3$, which was corrupted with 1% additive Gaussian noise.

We analyze the dynamics of the Hénon attractor in the presence of noise and how it can be restored via gradient descent optimization. The attractor, after adding a small amount of random noise (1% of the signal's standard deviation), is depicted in the first subplot "Noisy Hénon Attractor". As can be observed, the stunning and intricate charm of the Hénon map undergoes a vehement distortion. This illustrates the fact that even low noise levels can decisively hinder the underlying mechanics of a chaotic system. After applying a noise reduction technique, figure 1(b), "Cleaned Hénon Attractor (Recovered)" was obtained. Here, using gradient descent, we estimated the system parameters a and b by minimizing the error produced between the prediction and actual measurements of consecutive points. Once the optimized parameters were found, the clean trajectory was reconstructed using the updated model. Compared to the noisy version, the cleaned attractor captures the sharp structure of the true Hénon map.



(a) (b)

Figure 1: **(a)** Noisy Hénon Attractor with 1% added noise. **(b)** Cleaned Hénon Attractor using Gradient descent, restoring the true dynamics.

To test how well different noise reduction methods work in chaotic systems, we look at both 2D and 3D versions of the Hénon map. We start with the 2D map and then extend our study to a 3D version to see how noise spreads in more complex systems. Figure 2 compares the 3D Hénon-like attractors with and without added noise. We add Gaussian noise to the system's state variables to simulate real-world disturbances.

We further investigate the dynamic behavior of the 3D discrete Lorenz-type Hénon map for various parameter M_2. Figure 3 illustrates attractors calculated with almost the same values of M_2, indicating the sensitivity of the system to parameter variation. A chaotic attractor is found for $M_2 = 0.85$, while a periodic or quasi-periodic structure is found for $M_2 = 0.815$. The color gradient in the plots indicates iteration steps, and it provides information about the evolution of the trajectory in time.

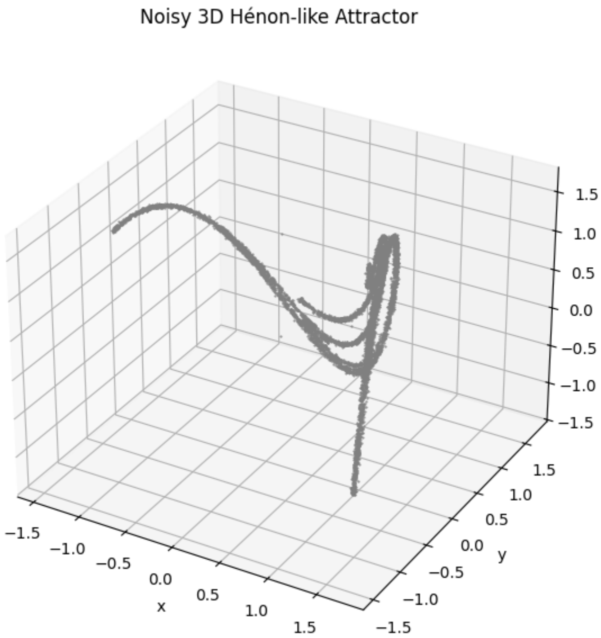
The process involves multiple iterative steps, each requiring operations on the order of $d \times n$, where d is the embedding dimension and n is the number of data points. Repeating the routine several times allows for progressive noise reduction.

The time series x_i consists of 1000 points from the Henon map. To simulate measurement error, 1% additive Gaussian noise was added. Figure 1(a) displays the Poincaré map of the noisy data. Before applying noise reduction, the data is embedded into a two-dimensional delay space using a delay of one so that:

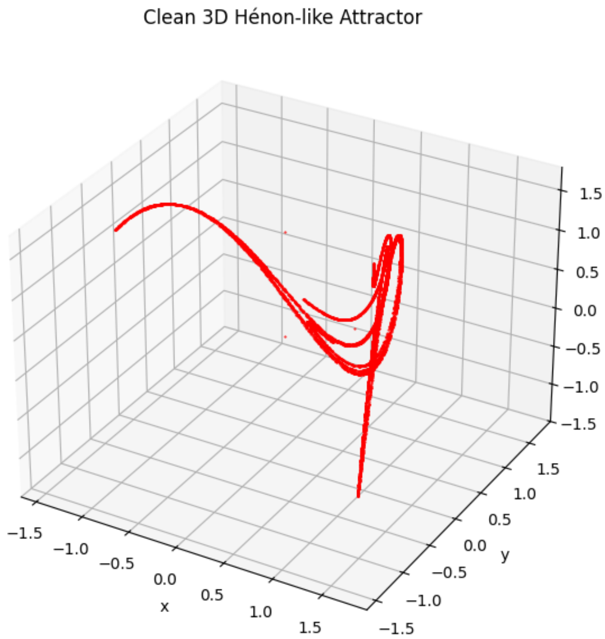
$$z_i = F(x_{i-1}, x_{i-2}) \tag{7}$$

The mapping function F is estimated through a global fit using a 5×5 grid of radial basis functions (RBFs) based on the method proposed by Broomhead & Lowe (1988). Each RBF has the form $\exp(-r^2)$, where r is the distance from the center of the function. The fitting minimizes the least squares error between predicted and actual values. Alternative fitting methods were tested and yielded similar results.

After approximating the function F, the Jacobian matrix of the map is calculated for each point in the series. From this, the gradient of the cost function H is derived, allowing adjustments to the

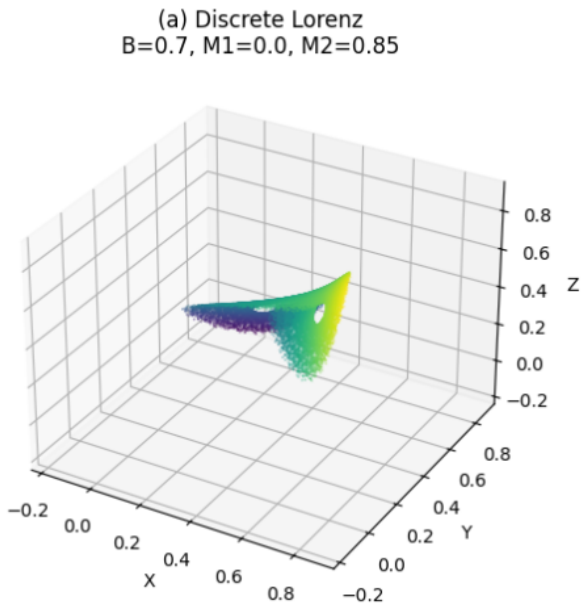


(a)

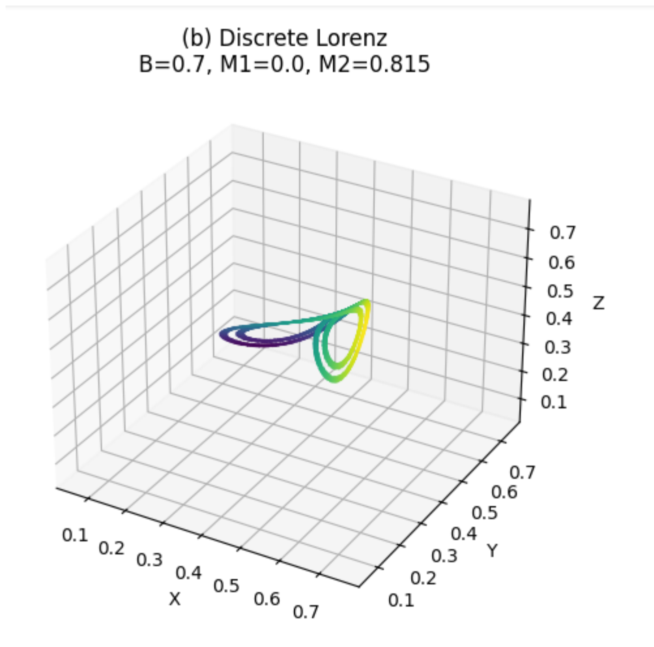


(b)

Figure 2: Comparison of noisy and clean versions of the 3D Hénon-like attractor. Gaussian noise reveals the attractor's structural robustness. **(a)** Noisy 3D Hénon Attractor **(b)** Clean 3D Hénon Attractor



(a)



(b)

Figure 3: 3D attractors of the discrete Lorenz-type Hénon map for different values of the parameter M_2. **(a)** Chaotic attractor observed for $B = 0.7$, $M_1 = 0.0$, $M_2 = 0.85$. **(b)** Periodic or quasi-periodic orbit for $B = 0.7$, $M_1 = 0.0$, $M_2 = 0.815$. The color gradient represents the iteration sequence, highlighting the structure and evolution of the attractors.

trajectory via gradient descent. This entire routine is repeated multiple times, in this case, 30 steps, until the resulting trajectory is sufficiently smooth and deterministic. The cleaned data is shown in Figure 1(b)

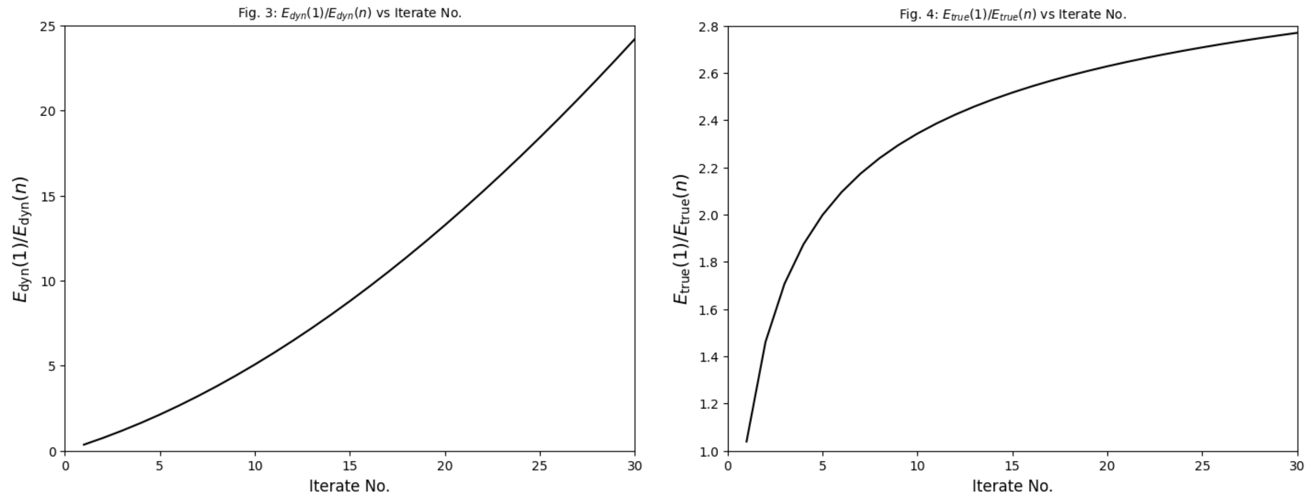
To measure the effectiveness of the method, two error measures are used:

$$E_{\text{dyn}} = \sqrt{\frac{1}{N}\sum_{i=d+1}^{N} \epsilon_i^2} \tag{8}$$

and

$$E_{\text{true}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - y_i)^2}, \tag{9}$$

where y_i is the original clean trajectory and x_i is the noisy trajectory after noise reduction. The first measure indicates the distance from determinacy, and the second one measures the distance from the original clean orbit. The effectiveness of a method can then be expressed as a ratio of the error.



(a) **(b)**

Figure 4: **(a)** A plot of $E_{\text{dyn}}(1)/E_{\text{dyn}}(n)$ against the iterate number, n. **(b)** A plot of $E_{\text{true}}(1)/E_{\text{true}}(n)$ against the iterate number, n.

The ratios of initial to final errors for the data are given as:

$$\frac{E_{\text{dyn}}(1)}{E_{\text{dyn}}(30)} = 24.2,$$
$$\frac{E_{\text{true}}(1)}{E_{\text{true}}(30)} = 2.77.$$

From this, we can infer the following values (assuming we take $E_{\text{dyn}}(30) = 1$ and $E_{\text{true}}(30) = 1$ for normalization)

In figure 4 (a),

$$\frac{E_{\text{dyn}}(1)}{E_{\text{dyn}}(n)} \tag{10}$$

is plotted against the iteration number n. This fraction indicates how much the dynamic error reduces as the process continues. The graph increases steadily, which implies the method continues to decrease the error. At the 30th iteration, the fraction is approximately 24.2, indicating a huge improvement from the noisy beginning.

In figure 4(b),

$$\frac{E_{\text{true}}(1)}{E_{\text{true}}(n)} \tag{11}$$

Across the iterations. This ratio shows how close the denoised signal gets to the original clean signal. The curve grows very fast at the start, after which it settles, reflecting that most gains are realized in the initial stages. At iteration 30, the ratio is roughly 2.77, with diminishing returns after the initial steps.

3 Exact Shadowing

In hyperbolic dynamical systems, the gradient descent method minimizes noise and resolves the shadowing issue. In particular, if the system is uniformly hyperbolic and the time series is finite with bounded dynamical errors, then any pseudo-trajectory that minimizes the defined cost function will match the system's actual deterministic orbit. The denoised trajectory *shadows* a real orbit due to the dynamical errors completely disappearing (a process called **exact shadowing**). As a result, the minimization procedure successfully recreates a real trajectory that could have theoretically generated the observed noisy data.

3.1 Theorem

Let z_i be a hyperbolic pseudo-trajectory of some mapping function F with bounded dynamic errors. Let H be the cost function defined in equation(2).If z_i is a minimum of H, then the dynamic errors,

$$\epsilon_i = z_i - F(z_{i-1}) \tag{12}$$

are all zero.

Interpretation: This theorem claims that if the pseudo-trajectory z_i is the minimizer of the cost function H, and if the system is hyperbolic (i.e., it does not have neutral directions and the trajectories diverge or converge exponentially), then the pseudo-trajectory must be an actual trajectory. In other words, every sequence point strictly satisfies the system's dynamics: $z_i = F(z_{i-1})$ for all i. The most important implication is that for hyperbolic systems, gradient descent does not merely simulate system behavior but reconstructs an exact orbit consistent with the system's evolution. This formalizes gradient-based noise reduction to reconstruct actual system dynamics from noisy observations.

Proof

Suppose z_i is a minimum of the cost function

$$H = \sum_{i=d+1}^{N} \epsilon_i^2 \tag{13}$$

$$H = \sum_{i=d+1}^{N} \|z_i - F(z_{i-1})\|^2. \tag{14}$$

Then the gradient of H with respect to each $\{z_i\}$ must vanish:

$$\forall i \quad \frac{\partial H}{\partial z_i} = 0 \tag{15}$$

Define the dynamical error as

$$\epsilon_i = z_i - F(z_{i-1}) \tag{16}$$

Then, since F is differentiable, we can write

$$\forall i \quad \epsilon_i = J_{i-1}\epsilon_{i-1} \tag{17}$$

where J_i is the Jacobian matrix of F at z_i, i.e., $J_i = DF(z_i)$.

Now, assume that the system is hyperbolic. This means that the Jacobian J_i can be decomposed into stable and unstable components. The corresponding error ϵ_i can then be splits into stable and unstable parts:

$$\alpha_i = J_{i-1}^s \alpha_{i-1} \tag{18}$$

$$\beta_i = J_{i-1}^u \beta_{i-1} \tag{19}$$

where α_i and β_i are the stable and unstable components of ϵ_i and J_i^s and J_i^u are the stable and unstable components of the Jacobian J_i, respectively.

By forward iteration, we obtain:

$$\beta_n = \left(\prod_{j=1}^{n-1} J_j^u \right) \beta_1. \tag{20}$$

Since the system is hyperbolic, the unstable directions will cause β_n to grow exponentially unless $\beta_1 = 0$. However, we are given that ϵ_i is bounded for all i, so β_n must also be bounded. This can only occur if:

$$\beta_i = 0 \quad \text{for all } i.$$

A similar argument for stable directions (backward in time) gives:

$$\alpha_n = \left(\prod_{j=i}^{n} (J_j^s)^{-1} \right) \alpha_i \tag{21}$$

and again, to prevent α_n from growing unbounded, we must have:

$$\alpha_i = 0 \quad \text{for all } i.$$

Thus, both components of ϵ_i must be zero:

$$\epsilon_i = 0 \quad \text{for all } i.$$

Hence, $z_i = F(z_{i-1})$ for all i, and z_i is a true deterministic trajectory of the system.

4 The Method of Schreiber and Grassberger

The Schreiber and Grassberger technique is a powerful technique for data denoising. Researchers Schreiber and Grassberger developed this technique in 1991. Similar to the gradient descent method, this algorithm adjusts the noisy data slightly so that the dynamic error can be reduced. By making these adjustments, the consistency of the system's behavior is improved. The update rule in the method is:

$$x_{i,\text{new}} = x_{i,\text{old}} + \delta\varepsilon_i \tag{22}$$

Here, we adjust the current value x_i by moving in the direction of the error ϵ_i, scaled by a step size δ. In gradient descent, we move in the direction opposite to the gradient, but here we move in the direction of the error. So, it is simpler and cheaper to compute than gradient descent.

If the same prediction error as in Equation(1) is used, which only looks at past values, then the method would only be effective in directions where the system is stable. But, some directions are unstable in chaotic systems, i.e., small changes grow rapidly. So, the noise would remain if only the past is considered. To solve the problem, we improve the prediction function F by using both past and future values, for example

$$x_i = F(x_{i-d1}, x_{i-d1+1}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+d2}) \tag{23}$$

here, d1 and d2 are dimensions of the past and future parts of the delay vector.

Now on applying steepest gradient descent we get,

$$\frac{\partial H}{\partial x_i} = -\frac{\partial F_{i-d1}}{\partial x_i}\epsilon_{i-d1} - \ldots - \frac{\partial F_{i-1}}{\partial x_i}\epsilon_{i-1} + \epsilon_i - \ldots - \frac{\partial F_{i+d2}}{\partial x_i}\epsilon_{i+d2} \tag{24}$$

where everything is defined as above.

If we are not using a function that can predict future values, i.e., it is non-predictive, then most of those derivative terms $\frac{\partial F_j}{\partial x_i}$ for $j \neq i$ will be small. The approximate gradient could be calculated using the direct error, ignoring all the rest.

When the noise reduction methods are put side by side, the steepest descent and the Schreiber-Grassberger algorithm are similarly effective at quantitative noise reduction. The Schreiber–Grassberger algorithm also has the benefit of being simple to implement. While the steepest descent algorithm is more complicated in many cases, the computational complexity is less because local approximations of the function are precomputed, as the Jacobians at each point would have already been computed.

It was noted that the Schreiber–Grassberger algorithm continued to perform consistently even close to homoclinic tangencies, areas in chaotic systems where system dynamics trajectories intersect in a sensitive and fragile manner. In contrast, when using predictive function approximation, the steepest descent algorithm performed slowly in these areas. While Section 3 of this paper explains why the predictive version of steepest descent struggles, the reason why the Grassberger and Schreiber do not suffer from this is unknown. This area requires further research.

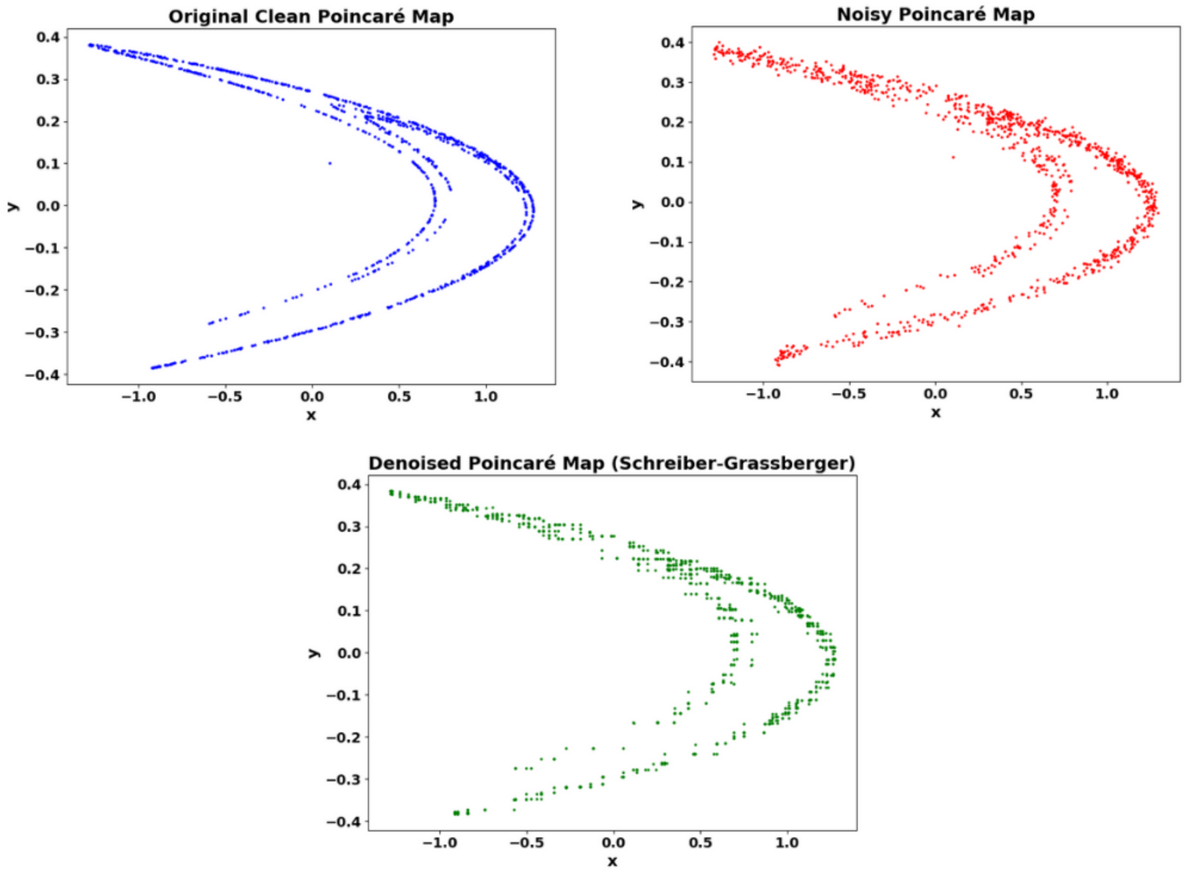


Figure 5: Poincaré maps illustrating the original clean data, the noisy data with some Gaussian noise added, and the denoised data with the Schreiber–Grassberger method. The series of plots shows how noise distorts the attractor and how the denoising procedure recovers its underlying structure. The last plot illustrates the method's capacity to restore dynamics that closely approximate the original clean system.

Upon measuring the effectiveness of this method using equation 8 and equation 9, we obtain the ratios of initial error to final error as

$$\frac{E_{\text{dyn}}(1)}{E_{\text{dyn}}(30)} = 1.00$$

and

$$\frac{E_{\text{true}}(1)}{E_{\text{true}}(30)} = 0.34$$

5 Discussion

This method for reducing noise in chaotic systems is based on a simple gradient descent strategy. Although it doesn't clean the data as quickly as some earlier methods, it's more stable and much easier to apply, especially when the noise levels are high. Unlike traditional techniques that assume the clean data is very close to the noisy version, this method doesn't need that assumption, which makes it a lot more practical for real-world messy data. This method also compares it to previous Newton-based methods, pointing out that while those can sometimes define the problem more clearly, they are more complex to implement when the system's exact mapping function isn't known, which is often the case. Interestingly, even when trying a more sophisticated cost function, the results didn't improve much, showing that the main limitation comes from how accurately we can approximate the actual system dynamics, not from the algorithm's complexity. Another critical insight discussed is how gradual improvements help prevent the model from getting stuck on wrong assumptions. When changes are minor, both the orbit and the system model can evolve together, leading to better overall noise reduction. In contrast, methods that force a perfect fit too quickly risk locking onto the errors from the noisy data itself. Finally, the paper reframes noise reduction in a bigger context. Instead of only trying to tell if the system is chaotic or random, we can ask how likely it is that the data came from a deterministic system. By comparing the size of the cleaned-up signal to the original noise level, researchers can get a reasonable measure of this likelihood. This approach feels more practical and honest, especially when working with real experimental data where nothing is perfect.

Conflict of interest

The authors declare that they have no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available within the article.

References

[1] E. N. Lorenz, "Deterministic nonperiodic flow 1," in *Universality in Chaos, 2nd edition.* Routledge, 2017, pp. 367–378.

[2] J. Gleick, *Chaos: Making a new science.* Penguin, 2008.

[3] A. Davey, K. Karasaki, J. Long, and M. Landsfeld, "Mandelbrot, bb, 1982. the fractal geometry of nature. references wh freeman and company, new york, 468 p. barker, j., 1988. a generalized radial-flow model for orbach, r., 1986. dynamics of fractal networks. science, pumping tests in fractured rock. water resour. res., v. 231, p. 814-819. v. 24, no. 10, p. 1796-1804. polek, j., 1990. studies of the hydraulic behavior of," *Earth SciencesDivision*, p. 40, 1991.

[4] H. Kantz and T. Schreiber, *Nonlinear time series analysis.* Cambridge university press, 2003.

[5] E. J. Kostelich and T. Schreiber, "Noise reduction in chaotic time-series data: A survey of common methods," *Physical Review E*, vol. 48, no. 3, p. 1752, 1993.

[6] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D: Nonlinear Phenomena*, vol. 35, no. 3, pp. 335–356, 1989.

[7] W. Brock and W. Dechert, "Statistical inference theory for measures of complexity in chaos theory and nonlinear science," in *Measures of complexity and chaos.* Springer, 1989, pp. 79–97.

[8] D. A. Hsieh, "Chaos and nonlinear dynamics: application to financial markets," *The journal of finance*, vol. 46, no. 5, pp. 1839–1877, 1991.

[9] F. Takens, "Dynamical systems and turbulence, warwick 1980, vol. 898, chap. detecting strange attractors in turbulence, 366–381," 1981.

[10] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, pp. 579–616, 1991.

[11] M. Casdagli and S. Eubank, *Nonlinear modeling and forecasting.* Westview Press, 1992, vol. 12.

[12] J. D. Farmer and J. J. Sidorowich, "Optimal shadowing and noise reduction," *Physica D: Nonlinear Phenomena*, vol. 47, no. 3, pp. 373–392, 1991.

[13] A. S. Pikovsky and J. Kurths, "Coherence resonance in a noise-driven excitable system," *Physical Review Letters*, vol. 78, no. 5, p. 775, 1997.

[14] J. H. Friedman, "Exploratory projection pursuit," *Journal of the American statistical association*, vol. 82, no. 397, pp. 249–266, 1987.

[15] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2009, vol. 2.

[16] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning.* Springer, 2006, vol. 4, no. 4.

[17] T. Schreiber and P. Grassberger, "A simple noise-reduction method for real data," *Physics letters A*, vol. 160, no. 5, pp. 411–418, 1991.

[18] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80.* Springer, 2006, pp. 366–381.

[19] J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Reviews of modern physics*, vol. 57, no. 3, p. 617, 1985.

[20] D. Kugiumtzis, O. Lingjærde, and N. Christophersen, "Regularized local linear prediction of chaotic time series," *Physica D: Nonlinear Phenomena*, vol. 112, no. 3-4, pp. 344–360, 1998.

[21] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[22] D. Ruelle, *Chaotic evolution and strange attractors.* Cambridge University Press, 1989, vol. 1.

[23] T. Sauer, "Interspike interval embedding of chaotic signals," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, no. 1, pp. 127–132, 1995.

[24] E. Ott, *Chaos in dynamical systems.* Cambridge university press, 2002.

[25] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," *Physica D: Nonlinear Phenomena*, vol. 51, no. 1-3, pp. 52–98, 1991.

[26] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[27] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[28] M. Burgin, *Theory of information: fundamentality, diversity and unification.* World Scientific, 2010, vol. 1.

[29] T. J. Hastie, "Generalized additive models," *Statistical models in S*, pp. 249–307, 2017.

[30] W.-C. Chen, "Nonlinear dynamics and chaos in a fractional-order financial system," *Chaos, Solitons & Fractals*, vol. 36, no. 5, pp. 1305–1314, 2008.

[31] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using support vector machines," in *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop.* IEEE, 1997, pp. 511–520.

[32] A. Wolf *et al.*, "Quantifying chaos with lyapunov exponents," *Chaos*, vol. 16, pp. 285–317, 1986.

[33] J. F. Gibson, J. D. Farmer, M. Casdagli, and S. Eubank, "An analytic approach to practical state space reconstruction," *Physica D: Nonlinear Phenomena*, vol. 57, no. 1-2, pp. 1–30, 1992.

[34] T. Schreiber and A. Schmitz, "Improved surrogate data for nonlinearity tests," *Physical review letters*, vol. 77, no. 4, p. 635, 1996.

[35] W.-L. You, Y.-W. Li, and S.-J. Gu, "Fidelity, dynamic structure factor, and susceptibility in critical phenomena," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 2, p. 022101, 2007.

[36] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* Chapman and Hall/CRC, 2024.

[37] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms.* John Wiley & Sons, 2011.

[38] K. Judd and A. Mees, "On selecting models for nonlinear time series," *Physica D: Nonlinear Phenomena*, vol. 82, no. 4, pp. 426–444, 1995.

[39] W. Zhang, "Financial time series forecasting using neural networks," Ph.D. dissertation, Swinburne, 2019.

[40] J. H. Stock and M. W. Watson, "A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series," 1998.

[41] L. A. Smith, C. Ziehmann, and K. Fraedrich, "Uncertainty dynamics and predictability in chaotic systems," *Quarterly Journal of the Royal Meteorological Society*, vol. 125, no. 560, pp. 2855–2886, 1999.

[42] I. Prigogine, "Time, structure, and fluctuations," *Science*, vol. 201, no. 4358, pp. 777–785, 1978.

[43] C. W. Gardiner *et al.*, *Handbook of stochastic methods.* springer Berlin, 1985, vol. 3.

[44] R. Hegger, H. Kantz, and T. Schreiber, "Practical implementation of nonlinear time series methods: The tisean package," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 9, no. 2, pp. 413–435, 1999.