Learning the Chaotic Hénon Map with a Multilayer Perceptron: Prediction Accuracy and Dynamical Analysis

HACENE GHAROUT
Université de Bejaia
Faculty of Exact Sciences and Faculty of Life and Natural Sciences
Laboratory of Applied Mathematics
06000 Bejaia,
ALGERIA
ORCID: https://orcid.org/0000-0002-3052-7968

Abstract: This paper investigates the ability of a multilayer perceptron (MLP) to approximate the dynamics of the classical Hénon map. The reference system is analyzed through its fixed points, Jacobian matrix, local stability, phase portrait, bifurcation diagrams, and maximum Lyapunov exponent. The MLP is trained to learn the transition from consecutive states, $(x_n, y_n) \mapsto (x_{n+1}, y_{n+1})$, using two hidden layers and the Levenberg-Marquardt algorithm. The training results yield a minimum validation mean squared error of $2 \quad .1488 \times 10^{-10}$. Model performance is further assessed through free-running trajectories, phase-space representations, residual distributions, parameter-dependent bifurcation diagrams, and Lyapunov-exponent estimates. The results indicate qualitative similarities between the original system and the learned model. However, divergence in iterated trajectories highlights the limitations of one-step prediction accuracy for reproducing long-term behavior in chaotic systems. Parameter-dependent comparisons are interpreted cautiously because the role of the control parameter in the learned model is not fully specified.

Keywords: Hénon map, chaotic dynamical systems, multilayer perceptron, supervised learning, attractor reconstruction, bifurcation diagram, Lyapunov exponent, Levenberg--Marquardt

Received: April 14, 2026. Revised: June 7, 2026. Accepted: September 11, 2026. Published: November 16, 2026.

1 Introduction

Discrete dynamical systems describe temporal evolution by repeatedly applying a map to the current state. Despite their compact formulation, nonlinear maps can exhibit fixed points, periodic orbits, bifurcations, and chaotic attractors. In a chaotic regime, nearby initial states may separate rapidly, so deterministic evolution does not necessarily imply reliable long-term trajectory prediction [1, 2]. Understanding this combination of deterministic structure and practical unpredictability remains an important problem in nonlinear dynamics.

The Hénon map, introduced by Hénon [3], is a standard low-dimensional model for investigating these phenomena. Its two equations are simple enough to permit detailed mathematical analysis and fast numerical simulation, but its parameter-dependent dynamics are sufficiently rich to provide a demanding test for data-driven models. In particular, the classical parameter pair $(a, b) = (1.4, 0.3)$ produces the familiar folded Hénon attractor.

Artificial neural networks offer a complementary way to represent nonlinear transformations. A multilayer perceptron (MLP) can learn an input--output relation from examples without explicitly being given the analytical expression of the map [4]. However, learning a map from sampled transitions and reproducing its dynamical properties are distinct objectives. A network may have a very small one-step prediction error while its iterated predictions drift away from the reference trajectory. For this reason, evaluation of a learned chaotic model should combine conventional prediction metrics with geometric and dynamical diagnostics.

This study has four objectives: (i) characterize the reference dynamics of the Hénon map; (ii) formulate the transition-learning problem and describe the MLP training procedure; (iii) compare the original and learned systems using trajectories, attractors, errors, bifurcation diagrams, and Lyapunov exponents; and (iv) assess the limits of the numerical results and identify the details needed for reproducibility. Where implementation details are not specified by the available figures and description, the corresponding limitations are stated explicitly.

2 The Hénon Map and Its Dynamical Properties

2.1 Definition and parameter values

The Hénon map is the discrete-time transformation

$$\begin{aligned} x_{n+1} &= 1 - a x_n^2 + y_n, \\ y_{n+1} &= b x_n, \end{aligned} \tag{1}$$

where $(x_n, y_n) \in \mathbb{R}^2$ is the state at iteration n, and $a, b \in \mathbb{R}$ are control parameters. The classical case considered here is

$$a = 1.4, \qquad b = 0.3, \tag{2}$$

with initial condition $(x_0, y_0) = (0.1, 0.1)$ for the illustrative orbit. Transient iterations are removed when the asymptotic attractor or parameter-dependent behavior is examined.

The map combines a quadratic nonlinearity in the first coordinate with a linear coupling between the two coordinates. The parameter a controls the strength of the nonlinear folding, while b controls the contribution of the previous x-coordinate to the next y-coordinate. Changing either parameter can alter the asymptotic behavior, so parameter scans provide information that a single trajectory cannot supply.

2.2 Fixed points

A fixed point $P^* = (x^*, y^*)$ satisfies

$$x^* = 1 - a(x^*)^2 + y^*, \qquad y^* = bx^*. \tag{3}$$

Substituting $y^* = bx^*$ gives

$$a(x^*)^2 + (1-b)x^* - 1 = 0. \tag{4}$$

For $a \neq 0$, the two candidate fixed-point coordinates are

$$x^*_{1,2} = \frac{-(1-b) \pm \sqrt{(1-b)^2 + 4a}}{2a}, \qquad y^*_i = bx^*_i. \tag{5}$$

The roots are real when the discriminant $(1-b)^2 + 4a$ is nonnegative. For the classical parameters in Equation (2), the fixed points are approximately

$$P_1 = (0.63135,\ 0.18941), \tag{6}$$
$$P_2 = (-1.13135,\ -0.33941). \tag{7}$$

These equilibria are algebraic solutions of the fixed-point equation; their existence alone does not determine whether they attract nearby trajectories.

2.3 Linear stability

The Jacobian matrix of the map is

$$J(x, y) = \begin{pmatrix} -2ax & 1 \\ b & 0 \end{pmatrix}. \tag{8}$$

At a fixed point, the eigenvalues satisfy

$$\det(\lambda I - J(x^*, y^*)) = \lambda^2 + 2ax^*\lambda - b = 0. \tag{9}$$

A fixed point is locally asymptotically stable if both eigenvalues have modulus strictly less than one. If at least one eigenvalue lies outside the unit circle, the fixed point is unstable. The eigenvalues describe local behavior near the equilibrium, not the full global structure of the attractor. Consequently, fixed-point analysis is complemented by direct orbit simulation and parameter-dependent diagnostics.

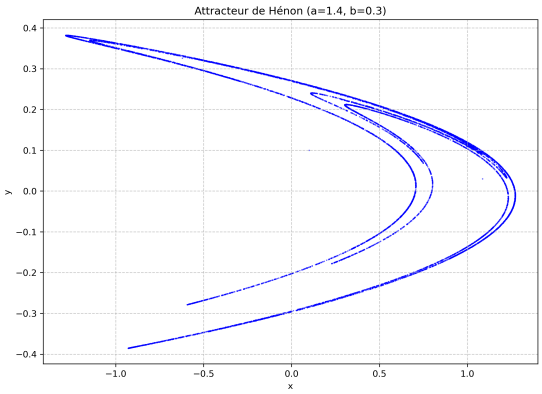


Figure 1: Reference phase portrait generated from the Hénon map at $a = 1.4$ and $b = 0.3$, with initial condition $(0.1, 0.1)$. The retained trajectory and transient length should be verified against the simulation code.

2.4 Attractor and bifurcation diagrams

For the classical parameters, iterating Equation (1) after discarding a transient is expected to produce the well-known Hénon strange attractor. The supplied phase portrait is retained as the numerical reference used in this article; however, its smooth, sparsely populated branches do not visibly show the densely folded structure usually associated with the classical attractor. The simulation settings and retained iteration count should therefore be checked before the plot is treated as evidence of asymptotic chaotic geometry.

Figure 1 presents the reference phase portrait used in the subsequent comparisons. The plotted orbit is treated as the numerical baseline for assessing the spatial organization generated by the MLP. Because the figure's visible geometry should be consistent with the underlying simulation, the initial condition, transient length, and retained iteration count must be identical or explicitly stated when comparing the two systems; visual similarity alone does not establish equality of their invariant measures.

A bifurcation diagram is constructed by varying one parameter while fixing the other, iterating the map for each parameter value, discarding transient points, and plotting the retained values of a state coordinate. The numerical results include a scan of b with $a = 1.4$ and scans of a with $b = 0.3$. These diagrams reveal the coexistence of periodic windows and irregular regions over the investigated parameter intervals.

Figure 2 shows the retained values of x_n as b varies while $a = 1.4$ remains fixed. Vertical bands of points indicate parameter values for which the orbit visits a broad range of x-values, whereas separated branches can indicate periodic behavior. This diagram therefore complements the single attractor in

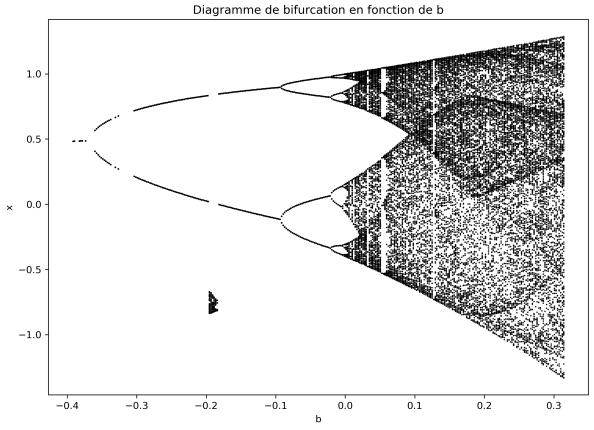


Figure 2: Bifurcation diagram of the original Hénon map in the (b, x) plane with $a = 1.4$.

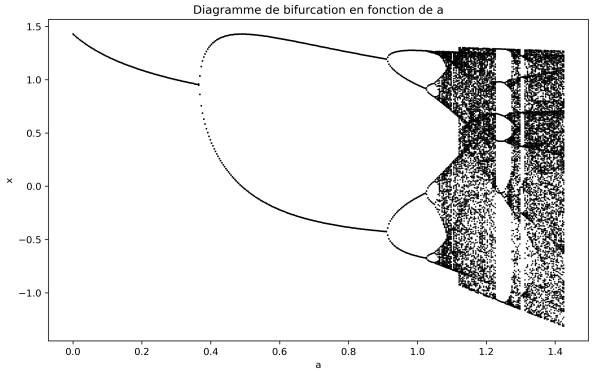


Figure 3: Bifurcation diagram in the (a, x) plane with $b = 0.3$.

Figure 1 by showing how the observed asymptotic regime changes with a control parameter.

The complementary scan in Figure 3 varies a while keeping $b = 0.3$. Comparing Figures 2 and 3 helps distinguish the influence of the two parameters: each plot is a one-parameter slice of the family of Hénon maps, not a complete description of the two-parameter plane. The plotted branches should be interpreted together with the stated transient removal and retained-iteration count.

Figure 4 combines bifurcation plots and Lyapunov-exponent estimates for scans in the parameters a and b. The joint view helps compare changes in orbit structure with changes in the leading instability indicator. Dense regions in a bifurcation diagram alone do not establish chaos; a positive maximum Lyapunov exponent provides complementary numerical evidence, subject to finite-time and numerical-convergence limitations.

The additional view in Figure 5 is intended to make the parameter-dependent structure easier to inspect. It should be read as a numerical continuation

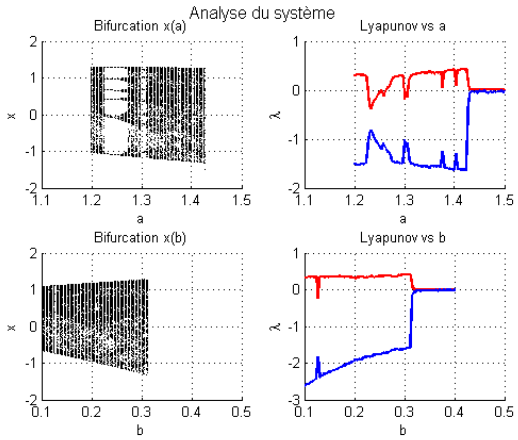


Figure 4: Combined bifurcation diagrams and Lyapunov-exponent estimates for parameter scans of the Hénon map.

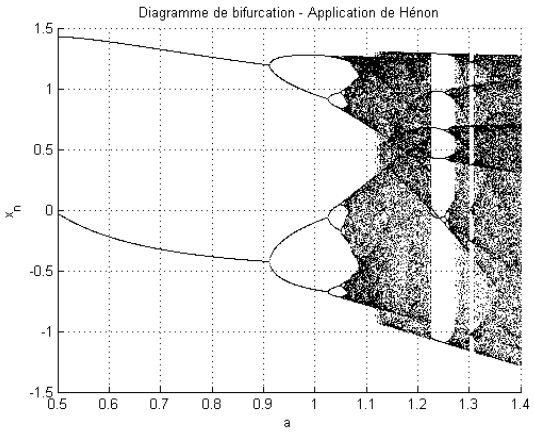


Figure 5: Bifurcation diagram as a varies with $b = 0.3$.

of the scan in Figure 3; the same values of b, transient length, and retained sample count must be used if the two views are to be compared quantitatively.

2.5 Maximum Lyapunov exponent

The maximum Lyapunov exponent (MLE) measures the average exponential growth rate of an infinitesimal perturbation along an orbit. A positive value is commonly associated with sensitive dependence on initial conditions, whereas a negative value indicates average contraction along the leading tangent direction. Values close to zero can occur near transitions or in non-hyperbolic regimes, and finite-time estimates should be interpreted with care.

For the Hénon map, tangent perturbations evolve according to the Jacobian in Equation (8). Let v_n be a nonzero tangent vector. A numerical estimate can be

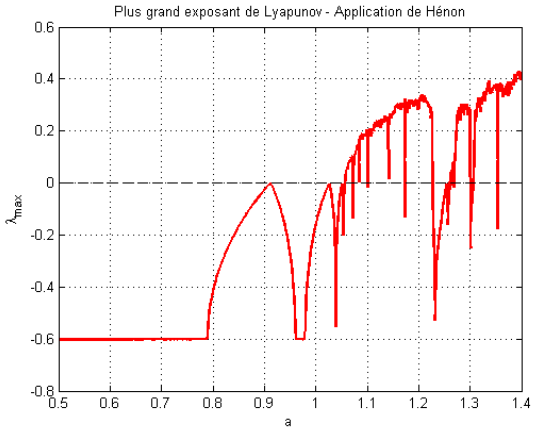


Figure 6: Maximum Lyapunov exponent of the original Hénon map as a function of a for fixed $b = 0.3$.

obtained by propagating

$$v_{n+1} = J(x_n, y_n)v_n \tag{10}$$

and periodically renormalizing the vector. If v_n is normalized before each step, an estimate over N retained iterations is

$$\lambda_{\max} \approx \frac{1}{N}\sum_{n=1}^{N} \log \left\| J(x_n, y_n)\, \widetilde{v}_n \right\|, \tag{11}$$

where $\widetilde{v}_n$ is the normalized tangent direction. This expression estimates the leading exponential growth rate; reliable results require a sufficiently long orbit and consistent transient removal.

Figure 6 complements the bifurcation diagram by reporting the maximum Lyapunov exponent as a varies. Positive values indicate average expansion of nearby trajectories along the leading tangent direction and are commonly associated with chaos; negative values indicate average contraction. The exponent should be interpreted jointly with the orbit diagram, especially near transitions where finite-time estimates may fluctuate. The standard tangent-dynamics approach is described in the literature on nonlinear dynamics [5].

3 MLP-Based Approximation of the Map

3.1 Learning objective

Equation (1) defines a deterministic transition function $F : \mathbb{R}^2 \to \mathbb{R}^2$:

$$F(x_n, y_n) = \begin{pmatrix} 1 - ax_n^2 + y_n \\ bx_n \end{pmatrix}. \tag{12}$$

The network is trained to approximate the one-step transition

$$\widehat{F}(x_n, y_n) \approx (x_{n+1}, y_{n+1}). \tag{13}$$

Each supervised example is a pair

$$\big(\mathbf{s}_n, \mathbf{s}_{n+1}\big), \qquad \mathbf{s}_n = (x_n, y_n)^{\mathsf{T}}. \tag{14}$$

The current state is the input, and the next state is the target. After training, the MLP can be used in free-running mode by feeding its output back as the next input. This distinction is central: one-step evaluation uses reference states as inputs, whereas free-running evaluation repeatedly uses predictions generated by the network itself.

3.2 Data generation and preprocessing

Training examples are generated by iterating the original Hénon map and recording consecutive states. The first 100 iterations are discarded to reduce the influence of the initial transient. The data are also normalized and randomly shuffled before training. Normalization can improve numerical conditioning by placing input and target variables on comparable scales; shuffling reduces dependence on the original sequential order during parameter updates.

The exact number of generated examples, the precise normalization transformation, the random seed, and the proportions assigned to training, validation, and test sets are not specified in the material available for this article. These settings should therefore not be inferred. For a reproducible implementation, the normalization parameters should be estimated from the training set only and then applied unchanged to validation and test data. The split strategy should also be stated clearly: randomly splitting adjacent points from one orbit may produce strongly correlated partitions, whereas a chronological or independently generated test trajectory can provide a more stringent evaluation.

3.3 Network architecture

The displayed MATLAB network configuration is a fully connected feed-forward MLP with two inputs, two hidden layers, and two outputs. The supplied training interface is shown in Figure 9. The input layer receives (x_n, y_n) and the output layer estimates $(\widehat{x}_{n+1}, \widehat{y}_{n+1})$. Each hidden layer applies an affine transformation followed by a nonlinear activation. A generic representation is

$$h_n^{(1)} = \phi_1(W_1\mathbf{s}_n + \mathbf{c}_1), \tag{15}$$

$$h_n^{(2)} = \phi_2(W_2h_n^{(1)} + \mathbf{c}_2), \tag{16}$$

$$\widehat{\mathbf{s}}_{n+1} = W_3h_n^{(2)} + \mathbf{c}_3, \tag{17}$$

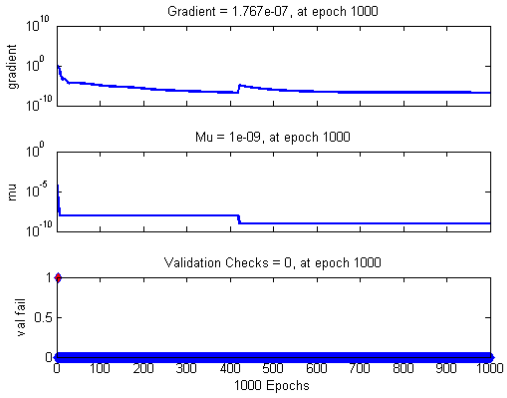


Figure 7: Training diagnostics showing the gradient, damping parameter, and validation checks during optimization.

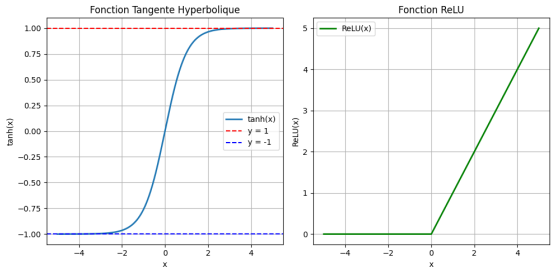


Figure 8: Activation functions discussed in the study: hyperbolic tangent and rectified linear unit (ReLU).

where W_i and $\mathbf{c}_i$ denote trainable weights and biases. A linear output layer is appropriate for continuous-valued regression. The hyperbolic tangent and ReLU activations are considered; the exact activation assigned to each hidden layer in the final trained model is not unambiguously specified in the text, and is therefore not assumed here. The number of neurons in each hidden layer is likewise not reported in the supplied description.

Figure 7 shows the gradient, damping parameter, and validation-check history during optimization. These diagnostics describe the optimizer's progress and stopping behavior; they do not directly measure the learned map's long-term dynamical fidelity. The architecture is assessed more directly through the prediction error and the dynamical comparisons that follow.

The two candidate hidden-layer nonlinearities are illustrated in Figure 8. The bounded, zero-centered tanh function and the piecewise-linear ReLU have different response ranges and gradient behavior. The figure illustrates the activation choices considered, but the final experimental report must identify which activation was actually used in the trained network.

The activation functions considered are

$$\phi_{\tanh}(u) = \tanh(u), \qquad \phi_{\mathrm{ReLU}}(u) = \max(0, u). \tag{18}$$

The tanh function is bounded and centered at zero, which can be useful when the data are scaled to a symmetric interval. ReLU is computationally simple and is defined piecewise by a zero response for negative inputs and a linear response for positive inputs. These functions have different optimization properties; however, the final paper should identify the actual activation configuration used in the reported experiment rather than treating the discussion of alternatives as evidence that both were used in one model.

3.4 Loss function and optimization

The mean squared error (MSE) is used to quantify the discrepancy between network outputs and target states:

$$\mathrm{MSE} = \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{s}_{i+1} - \widehat{\mathbf{s}}_{i+1}\|_2^2 . \tag{19}$$

The reported best validation value is

$$\mathrm{MSE} = 2.1488 \times 10^{-10} \tag{20}$$

at epoch 1000, as indicated in Figure 10. This value is a validation result, not a test error. Its generalization significance still depends on how the training, validation, and test sets were formed and whether the test data were independent of the training trajectory.

The reported optimizer is Levenberg--Marquardt (LM), a method widely used for nonlinear least-squares training. In simplified form, an LM update for parameter vector $\boldsymbol{\theta}$ can be written as

$$\Delta\boldsymbol{\theta} = -(J_e^{\mathsf{T}} J_e + \mu I)^{-1} J_e^{\mathsf{T}} \mathbf{e}, \tag{21}$$

where $\mathbf{e}$ is the residual vector, J_e is its Jacobian with respect to the network parameters, and μ is a damping parameter. The damping term balances a gradient-like update with a Gauss--Newton-type step. This optimization description explains the reported training method, but does not replace the need to specify the actual network size and training/validation configuration.

Figure 9 is the MATLAB Neural Network Training interface and documents the training setup used for the network. The performance curves themselves are shown in Figure 10; their interpretation depends on the distinction between training, validation, and test errors.

Figure 10 displays the training, validation, and test performance curves. The plot labels the best

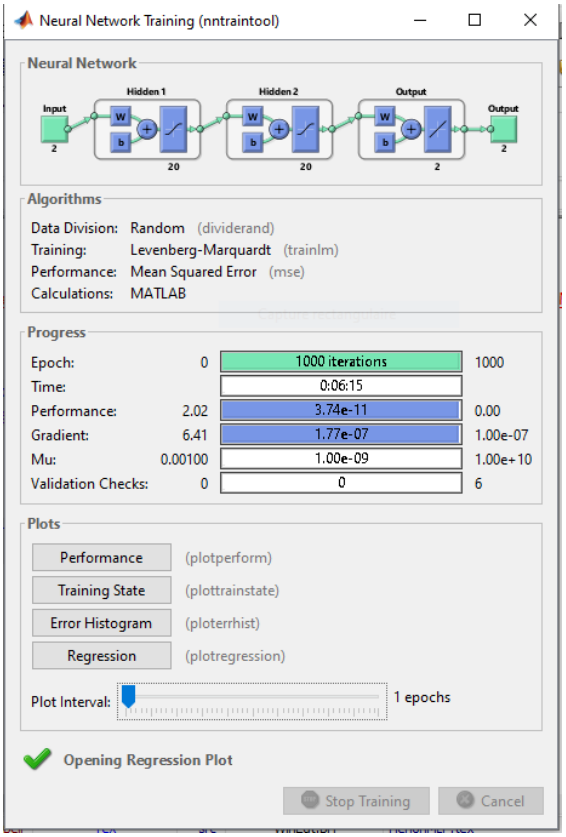


Figure 9: MATLAB Neural Network Training interface showing the network setup used in the reported experiment.

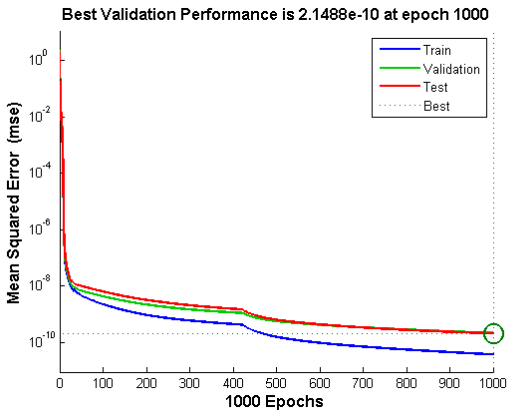


Figure 10: Training, validation, and test performance during 1000 epochs; the plot marks a best validation MSE of 2.1488×10^{-10} at epoch 1000.

validation performance as 2.1488×10^{-10} at epoch 1000. This is more informative than an unqualified MSE because it identifies the validation partition, although the split strategy and independence of the test set still need to be documented.

The residual distribution in Figure 11 provides

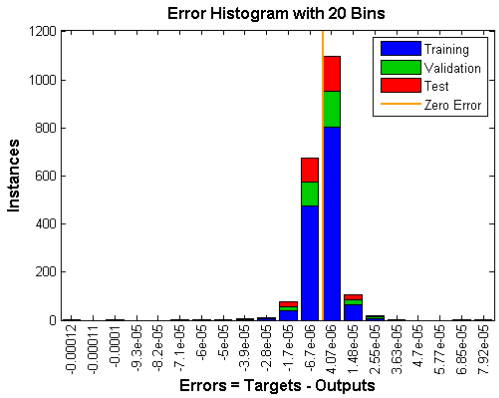


Figure 11: Residual histogram from the neural-network training output; preserve the original axes and residual definition.

a complementary view of the fitting quality. Concentration around zero indicates that many predictions have small signed residuals; it does not, by itself, exclude state-dependent bias or large errors in the distribution tails. The original axes and the distinction between training, validation, and test residuals should be preserved when reproducing the figure.

3.5 Evaluation protocol

The assessment uses several complementary tests:

1. *One-step prediction:* compare the MLP output with the true next state for a supplied input state.
2. *Free-running trajectory:* initialize the network and iterate its own outputs, then compare the predicted and reference trajectories.
3. *Attractor reconstruction:* discard a transient in the free-running sequence and plot the remaining states in the (x, y) plane.
4. *Residual distribution:* inspect the histogram of prediction errors to determine whether errors are centered near zero and whether large residuals occur.
5. *Parameter-dependent dynamics:* compare bifurcation diagrams and maximum Lyapunov exponents across the investigated parameter interval.

These tests probe different properties and should not be treated as interchangeable. A small MSE on one-step pairs evaluates local transition fitting on the sampled data; attractor geometry and Lyapunov exponents probe aspects of the dynamics generated after repeated application of the learned map.

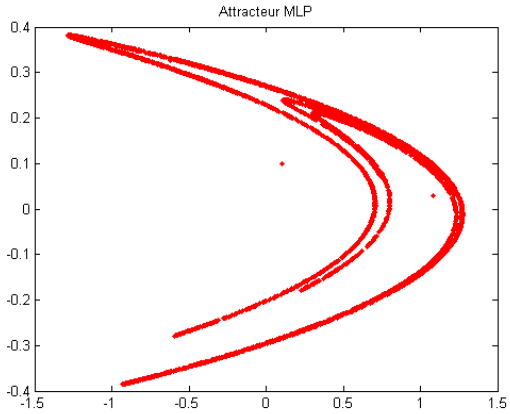


Figure 12: Phase portrait generated by iterating the trained MLP in free-running mode.

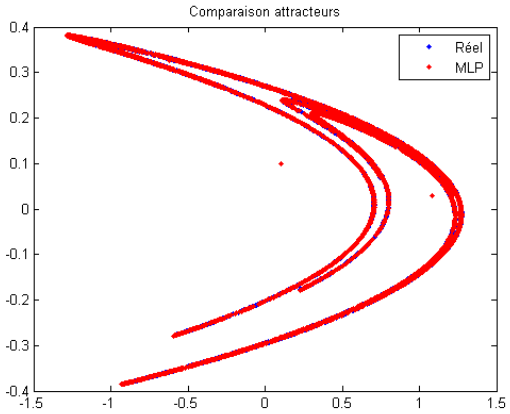


Figure 13: Comparison between the reference Hénon trajectory and the trajectory generated by the trained MLP.

4 Numerical Results

4.1 Reconstructed attractor

The MLP-generated orbit is compared with the supplied reference phase portrait. This comparison is a useful qualitative test because it examines the distribution of states produced after repeated iteration rather than only the pointwise error on isolated transitions. The learned attractor need not coincide point by point with the original trajectory: in a chaotic regime, two accurate trajectories can separate even while remaining on similar invariant sets.

As shown in Figure 12, the free-running MLP produces a phase portrait resembling the supplied reference plot in Figure 1. This indicates qualitative agreement between the two plotted outputs over the simulated horizon, but does not by itself establish that either plot resolves the asymptotic strange attractor. A stronger comparison would use the same number of retained points and axis limits for both systems, and would quantify differences in spatial occupancy rather than relying on visual resemblance alone.

A meaningful visual comparison requires the original and learned attractors to be generated with clearly stated initial conditions, transient lengths, and numbers of retained points. The axes should use the same ranges where possible. Differences in density or fine-scale geometry can result from finite sampling, small model errors, or a genuine change in the invariant dynamics. Accordingly, the reported visual similarity supports qualitative reconstruction but is not, by itself, proof of dynamical equivalence.

4.2 Reference trajectory versus MLP prediction

The reference trajectory is compared with the MLP-generated sequence. The reported result is close agreement at the beginning, followed by increasing divergence as the number of iterations grows. This behavior is consistent with iterative error accumulation, a limitation commonly encountered when approximating sensitive chaotic dynamics [2]. If the learned map is written as $\widehat{F} = F + \varepsilon$, then the state error at the next iteration depends both on the local approximation error ε and on the propagation of the existing state error through the derivative of the map. In a sensitive regime, the latter contribution can grow rapidly.

Figure 13 compares the reference orbit with the neural prediction. The reported initial agreement followed by later divergence is consistent with the accumulation of local model errors under iteration. For interpretation, it is essential to distinguish free-running predictions from one-step predictions that use the exact reference state at every step; only the former measures the compounded effect of feeding network outputs back into the model.

The observed divergence should therefore be interpreted carefully. It does not automatically mean that the network has failed to learn the one-step map, nor does an initial overlap prove long-term predictability. A complete report should specify whether the comparison uses one-step predictions driven by reference inputs or a free-running simulation driven by predicted inputs, since these experiments answer different questions.

4.3 Prediction-error histogram

The prediction-error histogram is concentrated around zero. Such a distribution is consistent with small average prediction errors on the evaluated samples. However, a histogram alone does not indicate whether errors are correlated with the state,

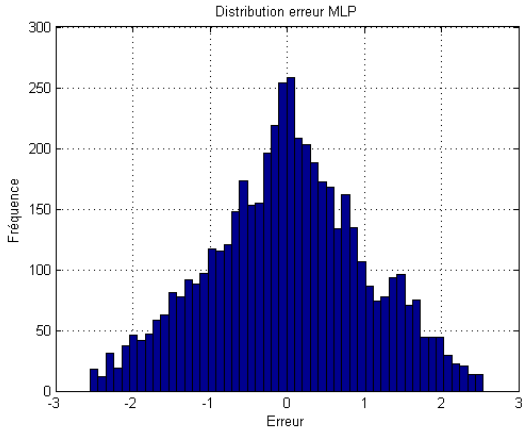


Figure 14: Histogram of prediction errors reported for the MLP.

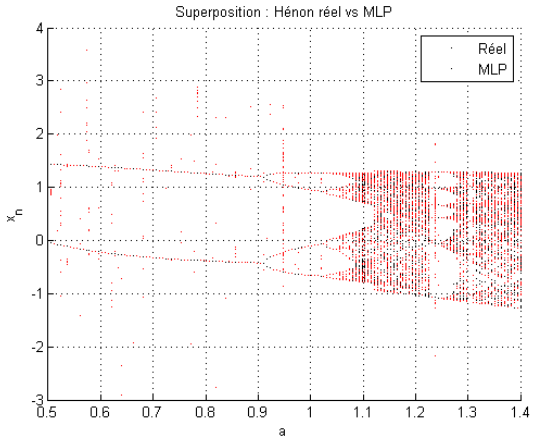


Figure 15: Comparison between the original-map bifurcation diagram and the diagram reported for the MLP model over $a \in [0.5, 1.4]$ with $b = 0.3$.

whether the tails contain occasional large errors, or whether the evaluated points are independent of the training data.

Figure 14 displays the prediction-error distribution used to assess the MLP. A distribution centered near zero is consistent with limited average bias, but its scientific interpretation depends on the residual definition and on the data partition from which it was computed. It should be reported separately from the training histogram in Figure 11 if the two plots represent different evaluation stages.

For reproducibility, the residual should be defined explicitly, for example $e_x = x_{n+1} - \widehat{x}_{n+1}$ and $e_y = y_{n+1} - \widehat{y}_{n+1}$. If the original plot combines both coordinates, the caption should state how the residuals were pooled. The MSE in Equation (19) should likewise be linked to a named data partition and to the same output scaling used during training.

4.4 Bifurcation diagrams: original map and learned model

The parameter scan uses

$$a \in [0.5, 1.4], \qquad b = 0.3, \tag{22}$$

using 1500 iterations per parameter value and discarding the first 500 as transient. The bifurcation diagram obtained from the original map is compared with a diagram attributed to the MLP. The two plots have been interpreted as showing qualitative agreement in broad periodic and irregular regions; this interpretation depends on how the parameter variation is implemented in the neural model.

Figure 15 presents the principal parameter-dependent comparison. Similar branch patterns would suggest that the learned model reproduces some qualitative changes in the observed orbit as a varies. However, this interpretation is valid only if the procedure used to vary a in the neural model is explicitly defined; a network trained at one fixed parameter pair cannot automatically represent a whole parameterized family unless the parameter is included in its inputs or the model is retrained for each value.

There is an important methodological detail to clarify before interpreting this plot as a parameter-generalization result. The training description presents the MLP as learning transitions generated at the classical parameter values, while the parameter-scan description varies a. If the parameter a is not supplied as an additional network input and the model is not retrained or otherwise modified for each value of a, a network trained only at one parameter pair does not represent the entire parameterized family $F_{a,b}$. The implementation used to produce the MLP scan must therefore be specified. For a genuinely parameterized surrogate, one possible formulation is

$$\widehat{F}(x_n, y_n, a, b) \approx (1 - ax_n^2 + y_n,\ bx_n), \tag{23}$$

with the parameters included in the input and training data covering the parameter domain. This is a possible formulation for a reproducible parameterized surrogate; it is not assumed to be the architecture used to generate the existing figure.

4.5 Maximum Lyapunov exponent comparison

Maximum Lyapunov exponents are also compared for the original map and the MLP model. The reported curves are not perfectly superposed, although their overall trends are described as similar, particularly for $a \in [1.1, 1.4]$. The comparison is relevant because

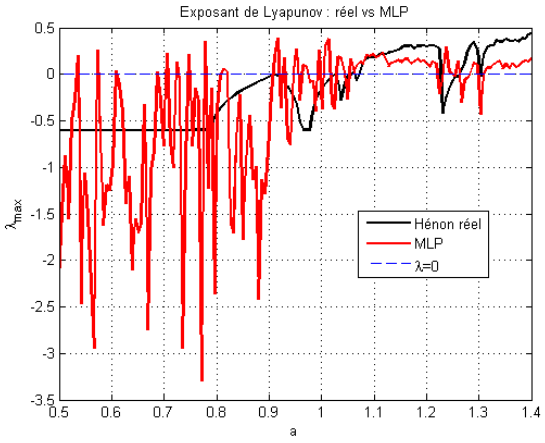


Figure 16: Comparison of the maximum Lyapunov exponent for the original Hénon map and the MLP model.

the sign of the largest exponent is often used as an indicator of local expansion and chaotic behavior.

Figure 16 compares the reported maximum Lyapunov-exponent curves. Agreement in their broad trends would support a similarity in average tangent-space expansion, whereas discrepancies may indicate differences in the learned Jacobian, the generated orbit, or the numerical estimation procedure. For a fair comparison, the neural exponent should be calculated from the Jacobian of the learned transition with respect to (x, y) and with the same transient removal and renormalization strategy as the original map.

The Jacobian for the original map is known analytically from Equation (8). For the MLP, a comparable exponent should be computed using the derivative of the learned transition with respect to its state input:

$$J_{\widehat{F}}(\mathbf{s}_n) = \frac{\partial \widehat{F}(\mathbf{s}_n)}{\partial \mathbf{s}_n}. \tag{24}$$

This Jacobian can be evaluated analytically by differentiating the network or numerically by finite differences. It should then be propagated along the MLP-generated orbit using tangent-vector renormalization, in the same way as for the original map. If the plotted neural exponent was instead calculated from the original map's Jacobian along MLP states, that alternative procedure must be stated, because it answers a different question. The available description does not fully specify this detail.

5 Discussion

The experiments provide evidence at several levels. First, the best validation MSE of 2.1488×10^{-10} indicates a close fit to the validation samples. Second, the MLP-generated phase portrait resembles the supplied reference plot, and the residual histogram is concentrated near zero. The supplied phase portraits should nevertheless be checked against a sufficiently long simulation, because their smooth branches do not visibly resolve the standard folded Hénon strange attractor. Third, the trajectory comparison illustrates that small local errors can accumulate under repeated iteration. Finally, the bifurcation and Lyapunov comparisons suggest broad qualitative similarities, but their interpretation depends on the precise procedure used to vary the parameter and calculate the MLP exponent.

These findings emphasize that data-driven modeling of chaotic systems requires more than a single error score. A model may approximate the transition function accurately over a sampled region yet generate a slightly different invariant set under iteration. Conversely, pointwise divergence between two chaotic trajectories is not by itself a sufficient measure of failure, because small perturbations can cause large phase differences. Evaluation should therefore distinguish at least three targets: one-step accuracy, statistical or geometric similarity of the attractor, and parameter-dependent dynamical fidelity.

Several limitations should be resolved in a final reproducible version. The hidden-layer widths and final activation configuration should be recorded; the training, validation, and test sizes should be reported; the normalization transformation and random seed should be supplied; and the meaning of the reported MSE should be identified. The procedure used to construct the MLP bifurcation diagram must also be documented, especially whether the parameter values are inputs to the network. Finally, the calculation of the neural maximum Lyapunov exponent should use and describe the Jacobian of the learned transition. These are not merely editorial details: they determine what the experiments establish and whether they can be independently reproduced.

6 Conclusion

This study investigated the ability of a multilayer perceptron (MLP) to approximate the chaotic Hénon map. The network achieved a minimum validation mean squared error of 2.1488×10^{-10}, demonstrating high accuracy in one-step prediction. The reconstructed phase portrait and prediction-error distribution provided additional insight into the agreement between the learned model and the reference system. Comparisons of trajectories, bifurcation diagrams, and maximum Lyapunov exponents were also conducted to assess the dynamical properties of the learned map.

Despite the excellent one-step prediction accuracy, the divergence observed during free-running simulations highlights the sensitivity of chaotic dynamics to small prediction errors and the difficulty of reproducing long-term trajectories. These findings demonstrate that prediction accuracy alone is insufficient to establish dynamical fidelity. Overall, the results illustrate the potential of MLPs for approximating nonlinear chaotic maps while emphasizing the importance of combining prediction-error metrics with dynamical analysis.

References

[1] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Addison-Wesley, Redwood City, CA, 1989.

[2] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 3rd ed. CRC Press, 2024.

[3] M. Hénon, ``A two-dimensional mapping with a strange attractor," *Communications in Mathematical Physics*, vol. 50, no. 1, pp. 69--77, 1976. doi: 10.1007/BF01608556.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[5] E. Gon□alves Da Silva, *Introduction aux systémes dynamiques et chaos*. Institut Polytechnique de Grenoble, 2004.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US