A Comparative Analysis of Machine Learning Models for Predicting the Lorenz System

Rushi Chaudhary
Memorial University of Newfoundland
rrchaudhary@mun.ca

October 27, 2025

1 Introduction

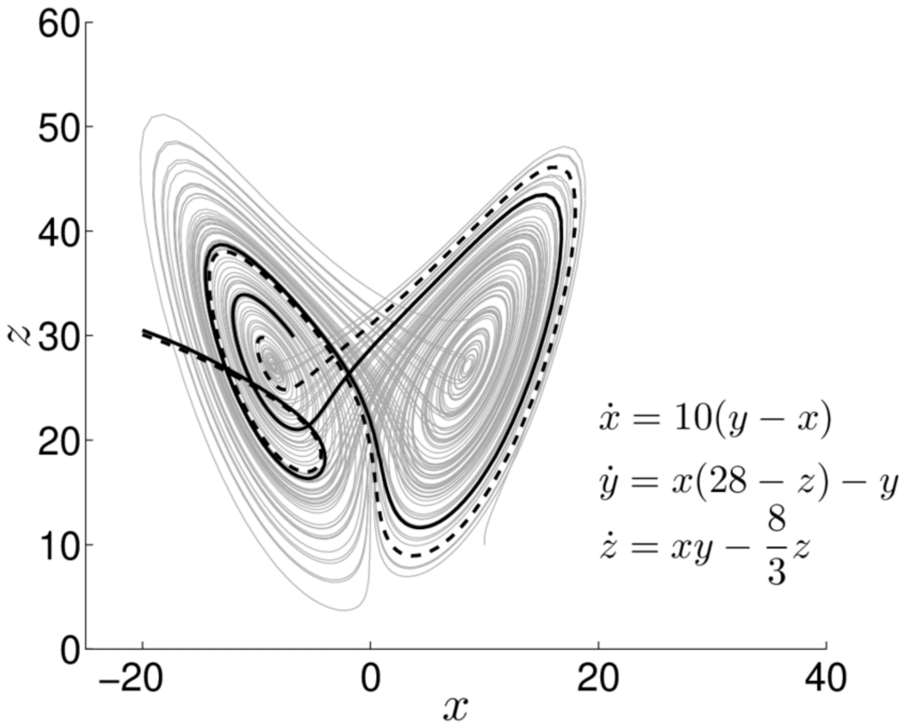
The Lorenz system, introduced by Edward N. Lorenz in his seminal 1963 paper on deterministic nonperiodic flow, serves as a foundational model in the study of chaotic dynamics[4]. Originally derived as a simplified mathematical representation of atmospheric convection, the system consists of three coupled nonlinear ordinary differential equations:

$$\dot{x} = \sigma(y - x), \tag{1}$$
$$\dot{y} = x(\rho - z) - y, \tag{2}$$
$$\dot{z} = xy - \beta z, \tag{3}$$

where x, y, and z represent variables related to convective motion, horizontal temperature variation, and vertical temperature variation, respectively. The parameters σ, ρ, and β control the system's behavior. The Lorenz system exhibits chaotic attractors characterized by sensitive dependence on initial conditions—a phenomenon popularly known as the "butterfly effect"[2]. This sensitivity implies that small changes in starting points can lead to vastly different trajectories, making long-term prediction inherently challenging despite the system's deterministic nature.

In recent decades, machine learning (ML) has emerged as a powerful tool for modeling and forecasting complex dynamical systems. Unlike traditional methods that rely on explicit integration of known equations, ML techniques can learn directly from data, identifying hidden structures in time-series observations. For chaotic systems, recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) architectures, have shown promise in handling sequential data and capturing temporal dependencies[1]. These models excel in short-term predictions by learning from historical trajectories but may struggle with long-term stability due to the amplification of small errors in chaotic regimes. To address these limitations, physics-informed approaches integrate domain knowledge into ML frameworks. Physics-Informed Neural Networks (PINNs), for instance, embed the governing differential equations into the loss function, ensuring that predictions adhere to physical laws while fitting the data.

This project undertakes a comparative analysis of various ML models—regression, LSTMs, and PINNs—for predicting the evolution of the Lorenz system. The primary goal is to assess their efficacy in both short-term and long-term forecasting horizons, quantified through metrics such as mean squared error (MSE) and root mean squared error (RMSE). Additionally, we aim to explain the theoretical reasonings of their performances



2 Methods

2.1 Problem formulation

We consider the Lorenz system

$$\dot{x} = \sigma(y - x), \tag{4}$$
$$\dot{y} = x(\rho - z) - y, \tag{5}$$
$$\dot{z} = xy - \beta z, \tag{6}$$

where (x, y, z) denote the state variables and (σ, ρ, β) are constant parameters. The learning task is one-step-ahead prediction: given a recent history of states, predict the next state $(x_{t+\Delta t}, y_{t+\Delta t}, z_{t+\Delta t})$.

2.2 Data generation and preprocessing

Two trajectories were generated by numerically integrating the Lorenz system using a variable–step solver and then resampling to a uniform time grid using interpolation to obtain evenly spaced samples:

- Training/validation trajectory: 10,000 samples from initial condition [1, 1, 1].
- Test trajectory : 10,000 samples from initial condition [5, 10, 15].

The state vectors were then standardized (zero mean and unit variance), and the same transformation was applied to validation and test sets. Generating separate test and training data sets ensures that we're testing true generalization, not memorization.

2.3 Supervised framing

We construct rolling-window sequences of fixed length $L = 50$ to predict the next time step. For a standardized time series $\{\mathbf{s}_t\}_{t=1}^{N}$ with $\mathbf{s}_t \in \mathbb{R}^3$, define input–target pairs

$$\mathbf{X}_t = \left[\mathbf{s}_{t-L+1}, \ldots, \mathbf{s}_t\right] \in \mathbb{R}^{L\times 3}, \qquad \mathbf{y}_t = \mathbf{s}_{t+1} \in \mathbb{R}^3, \tag{7}$$

for $t = L, \ldots, N-1$. This yields $N - L$ sequences per split.

2.4 Models

We compare three approaches that take $\mathbf{X}_t$ as input and predict $\widehat{\mathbf{y}}_t$.

Baseline regression (Random Forest): An ensemble of decision trees trained on the flattened $L \times 3$ window. Each tree learns simple split rules and the forest averages their predictions to capture nonlinear relations while reducing variance.
Expected behavior:

- Handles nonlinear mappings on tabular features but does not model temporal continuity explicitly.
- Good baseline for one-step prediction; typically worse than sequence models on smooth dynamical systems.

Long Short-Term Memory (LSTM): A recurrent neural network that ingests the $L \times 3$ sequence in order, compresses it into a hidden state, and maps that to the next state via a dense layer. Trained with MSE using Adam and early stopping.
Expected behavior:

- Learns nonlinear temporal dependencies leading to better performances on time series.
- Performance depends on sequence length L, hidden size, regularization, and the match between training and test distributions.

Physics-Informed Neural Network (PINN): A neural network $f_\theta(t) = (x_\theta, y_\theta, z_\theta)$ trained to both fit observed data and satisfy the Lorenz ODE via automatic differentiation. The loss combines data fidelity and physics residuals:

$$\mathcal{L}_{\text{data}} = \frac{1}{N_d} \sum_i \left\| f_\theta(t_i) - \mathbf{s}(t_i) \right\|_2^2, \tag{8}$$

$$\mathcal{L}_{\text{phys}} = \frac{1}{N_c} \sum_j \Big(\|\dot{x}_\theta - \sigma(y_\theta - x_\theta)\|_2^2 + \|\dot{y}_\theta - (x_\theta(\rho - z_\theta) - y_\theta)\|_2^2 + \|\dot{z}_\theta - (x_\theta y_\theta - \beta z_\theta)\|_2^2 \Big), \tag{9}$$

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{phys}} \mathcal{L}_{\text{phys}}. \tag{10}$$

Expected behavior:

- Encodes correct physics, improving data efficiency and physical consistency.
- May not minimize one-step MSE as effectively as LSTM but long Term fitting outclasses LSTM with careful tuning.

2.5 Training, validation, and testing

From the standardized trajectory with initial condition $[1, 1, 1]$, the first 3000 samples form the training series and the next 7000 samples form the validation series. The test set is the separate standardized trajectory (10,000 samples) generated from $[5, 10, 15]$ and is never used for model fitting or selection.

2.6 Evaluation

One-step-ahead mean squared error (MSE) on the test set:

$$\text{MSE} = \frac{1}{M} \sum_{t=1}^{M} \left\| \widehat{\mathbf{y}}_t - \mathbf{y}_t \right\|_2^2, \tag{11}$$

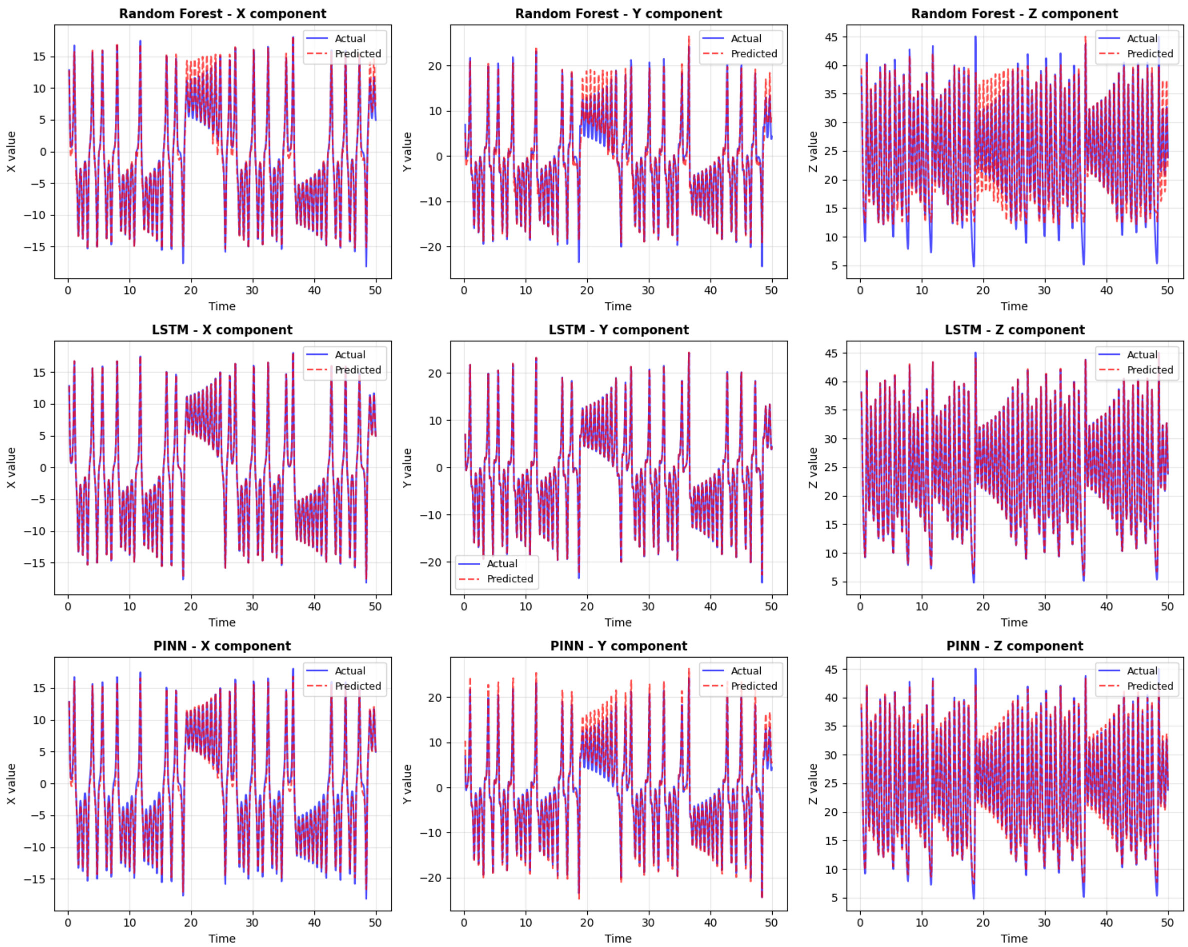
computed for windows of length $L = 50$.

3 Results

3.1 One-step-ahead accuracy

We evaluated one-step-ahead prediction on the held-out Lorenz trajectory generated from initial condition $[5, 10, 15]$ using a rolling window of length $L = 50$. Mean squared error (MSE) was computed in standardized coordinates over the test sequences (9950 windows).

Model	Test MSE
Random Forest	0.063599
LSTM	0.000663
PINN	0.029885



3.2 Unexpected outcome relative to hypothesis

We hypothesized in the previous section that a physics-informed model (PINN), which enforces the Lorenz ODE through residual penalties, would outperform a purely data-driven sequence model (LSTM) data due to its stronger inductive bias and improved data efficiency. Contrary to this expectation, the LSTM achieved the lowest one-step MSE, while the PINN performed intermediate between the LSTM and the Random Forest.

3.3 Possible explanations

The observed ranking (LSTM < PINN < Random Forest in MSE) is consistent with several plausible factors:

- **Different training goal vs. evaluation score.** The PINN is trained to be *physics-consistent* [5], not to minimize the next-step error directly. Our test metric is one-step MSE, which favors models (like the LSTM) trained exactly for that target.
- **Overfitting vs. regularization.** The LSTM likely had just enough capacity and regularization to fit short-term patterns without overfitting noise. The PINN can *underfit* locally when the physics penalty is too strong or the network is small, and

the Random Forest can *overfit* local patterns yet still miss smooth dynamics. Recent studies document optimization and failure modes in PINNs. These studies emphasize the difficuly in optimising the loss functions in PINNs [3]

4 Conclusion

We compared three approaches for one-step prediction of the Lorenz system from short history windows ($L = 50$): a baseline (Random Forest), a sequence model (LSTM), and a physics-informed neural network (PINN). Using standardized data, a clean train/validation/test split, and a held-out trajectory from a different initial condition, the LSTM achieved the lowest one-step MSE, followed by the PINN, with the Random Forest performing worst.

This outcome differs from our initial expectation that the PINN would outperform the LSTM due to its physics prior. The likely explanations include the PINN's objective to balance data fit and ODE residuals rather than directly optimizing next-step error and loss tuning choices leading overfitting in data-only models or underfitting in physics-regularized ones.

Practically, if the goal is lowest next-step error on data from the same regime, LSTMs are a strong choice. If the goal is physical consistency or data efficiency PINNs remain attractive but require careful tuning of physics/data losses and training strategy.

5 Appendix

All code and LaTeX sources for this report are available on GitHub:

- Repository: RC1092/MATH_2130_project2
- Notebook: code.ipynb

References

[1] Pierre Dubois, Thomas Gomez, Laurent Planckaert, and Laurent Perret. Data-driven predictions of the lorenz system. *Physica D: Nonlinear Phenomena*, 408:132495, 2020.

[2] James Gleick. *Chaos: Making a New Science*. Viking, New York, 1987.

[3] Aditi Krishnapriyan, Amir Gholami, S. Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34, 2021. See also arXiv:2109.01050.

[4] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[5] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.