Turbulence state modelling using Machine Learning for fusion plasmas

Lucas Perrier
Supervisors: Mathieu Seurin   Alessandro Biancalani

Introduction & Motivation

Tokamak plasma turbulence is a key challenge in fusion research, as chaotic behavior leads to energy losses and limits confinement. More broadly, machine learning (ML) offers a new approach to forecast complex, nonlinear dynamics. In this work, we focus on the use of an Recurrent Neural Network (RNN) to predict the nonlinear predator–prey dynamics of the Lotka–Volterra system. This serves as preparation for studying the Chen system, a dynamical system used as a proxy for turbulence in tokamak plasmas.[1].

Dynamical System Equations

The Lotka–Volterra system equations are:

$$\frac{dx}{dt} = \alpha x - \beta xy, \qquad \frac{dy}{dt} = \delta xy - \gamma y. \tag{1}$$

where: x, y are state variables, and $\alpha, \beta, \delta, \gamma$ are system parameters.

For the Chen model:

$$\frac{dP}{dt} = P - 2ZS\cos(\Phi), \tag{2}$$

$$\frac{dS}{dt} = -\Gamma_d S + ZP\cos(\Phi), \tag{3}$$

$$\frac{dZ}{dt} = -\gamma_z Z + 2PS\cos(\Phi), \tag{4}$$

$$\frac{d\Phi}{dt} = \delta - \frac{PZ\sin(\Phi)}{S}. \tag{5}$$

where P, S, Z, Φ are state variables, and $\Gamma_d, \gamma_z, \delta$ are system parameters.

Data and Methods

Data Acquisition: We use a Runge-Kutta 4 numerical integrator to simulate the Lotka-Volterra system over time. By varying the parameters and initial conditions, we generate multiple time series, which we then format into supervised learning datasets.

RNN Benchmark:

- **Data Structure:** Each training sample consists of a 4-second sliding window extracted from the simulation data. The input features include the state variables (x, y), system parameters $(\alpha, \beta, \gamma, \delta)$, and time t, forming a 7×T tensor. The output targets are the immediate next step values of the state variables (x, y), represented as a 2-dimensional vector.
- **Architecture:** We tested a single-layer RNN with a hidden size of 32 units (with comparisons using 16 and 64 units), mapping inputs to a 2-dimensional output that predicts the next time step's values.
- **Optimization:** The RNN was trained using the Adam optimizer with mean squared error as the loss function.

Our experiments show that the RNN can predict the nonlinear dynamics of the Lotka–Volterra system, serving as a baseline for further work.

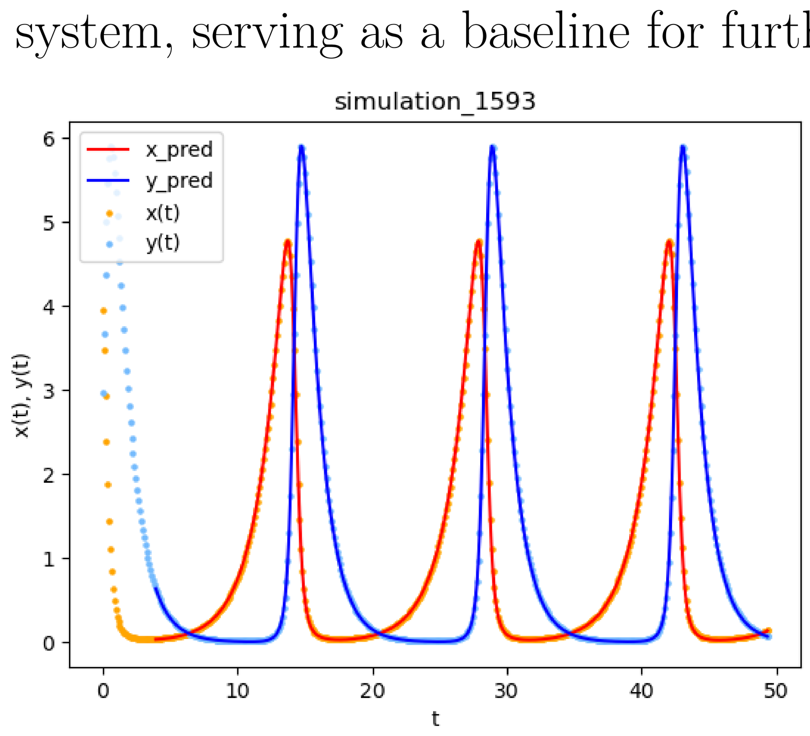

Figure 1: The RNN predicts the nonlinear dynamics of the Lotka-Volterra predator-prey model.

When varying the amount of different parameter combinations used for training data and length of the training sequences, we maintain accurate forecasts.

Assumptions and Limitations

- **Simplified Model:** The Lotka–Volterra system is a simplified, non-chaotic predator–prey model.
- **Noise Sensitivity:** The RNN is not robust to noise; adding noise causes overfitting.
- **Generalization:** Generalization to unseen dynamical system parameter combinations has not been tested.
- **Plasma Context:** Real plasma data are not always evenly spaced in time. Our approach assumes uniform time steps, and the RNN functions as an oracle without uncovering underlying physics.

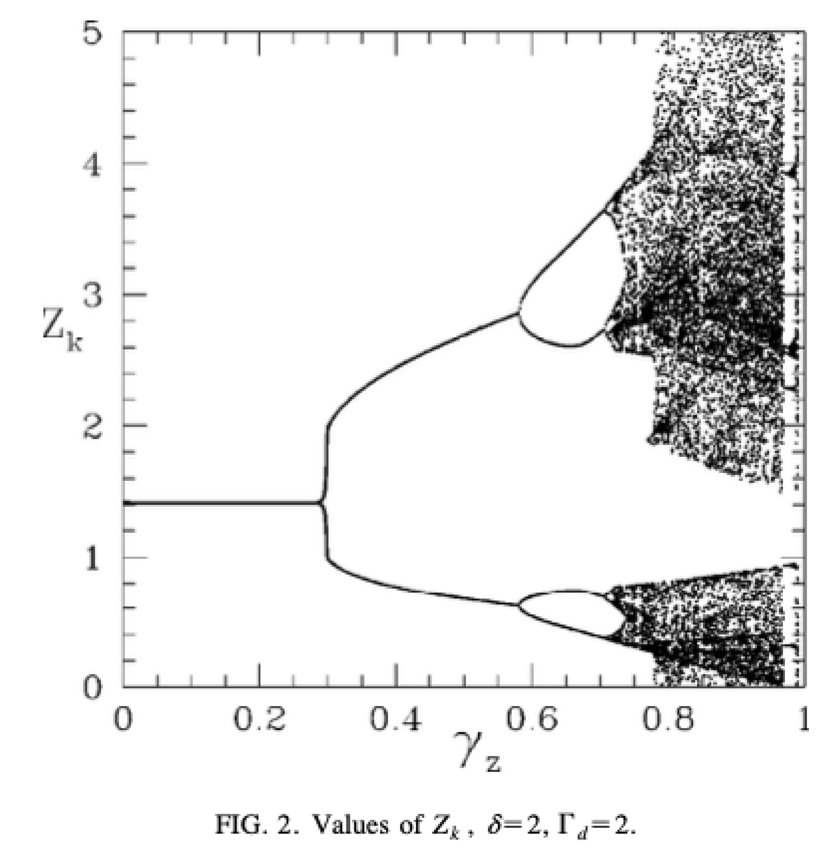

Figure 2: The Chen system exhibits chaotic behavior.[1]

Future Work

For the Chen system, which exhibits chaotic behavior, different algorithms may be more suitable depending on the specific challenges, such as capturing continuous dynamics or ensuring physical consistency. Our future work will explore the following approaches:

- **Neural ODE:** This approach allows the model to learn dynamics from the data while respecting the system's underlying continuous-time nature, potentially improving robustness to noise [2].
- **SINDy Algorithm:** Sparse Identification of Nonlinear Dynamical Systems (SINDy) automatically discovers the governing equations from the data. By leveraging sparsity-promoting techniques, SINDy can yield interpretable models that capture the essential dynamics of the system [3].
- **PINNs:** Physics-Informed Neural Networks (PINNs) incorporate the governing equations as constraints within the ML model's loss function. PINN's encourage the neural network to remain consistent with the underlying physical laws, and enables formulating the task as an inverse problem. [4].

Conclusions

We demonstrated that an RNN can predict the nonlinear predator–prey dynamics of the Lotka–Volterra system using sliding window forecasting. However, our approach has several limitations, including sensitivity to noise and assumptions about data regularity. Future work will compare different ML models and ways to frame the problem, with the aim of better forecasting chaotic transitions relevant to tokamak plasma turbulence.

References

[1] Liu Chen, Zhihong Lin, and Roscoe White. Excitation of zonal flow by drift waves in toroidal plasmas. *Physics of Plasmas*, 7(8):3129–3132, 2000.

[2] Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.

[3] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[4] M. Raissi, P. Perdikaris, and G.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.