Forecasting chaotic dynamics

Lecture 19 of “Mathematics and AI”

Outline

1. Nonlinear dynamics

   fixed points, stable manifolds, chaos, Lyapunov exponent, strange attractor

2. Time-delayed embedding

   feature mapping, Taken’s theorem

3. Reservoir computing

Nonlinear dynamics

Dynamical system

- Discrete dynamical systems

$$x_{t+1} = f(x_t, x_{t-1}, \dots, \varepsilon_t, \varepsilon_{t-1} \dots, t)$$

- Discrete deterministic system

$$x_{t+1} = f(x_t, x_{t-1}, \dots, t)$$

- Discrete deterministic Markov system

$$x_{t+1} = f(x_t, t)$$

- Continuous dynamical systems

$$\frac{dx}{dt} = \int_{-\infty}^{t} f(x(\tau), \varepsilon_\tau, \tau)\, d\tau$$

- Continuous deterministic system

$$\frac{dx}{dt} = \int_{-\infty}^{t} f(x(\tau), \tau)\, d\tau$$

- Continuous deterministic Markov system

$$\frac{dx}{dt} = f(x(t), t)$$

More spaces!

Feature space

- Axes correspond to quantities that we ***observe***
- A point in space corresponds to a ***measurement***

1-1 correspondence if model has exactly 1 parameter per observed variable

Parameter space

- Axes correspond to ***model parameters***
- A point in space corresponds to a ***model configuration***

1-1 correspondence if we observe ***exactly*** as many features as are needed to describe the system's state

State space

- Axes correspond to all quantities that are ***necessary to characterize*** a system's state
- A point in space corresponds to a ***system state***

same

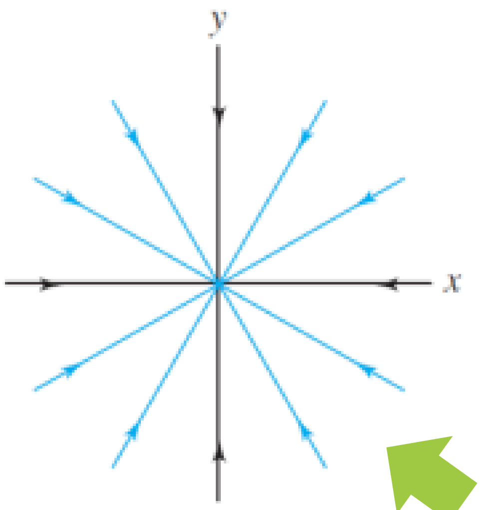
Phase space

- Axes like state space
- A ***point*** in space corresponds to a ***system state***
- A ***curve*** in space corresponds to a ***trajectory***

Image sources: https://logancollinsblog.com/2018/01/27/notes-on-dynamical-systems, https://www.cnet.com/science/biology/features/what-is-the-butterfly-effect-how-scientists-find-beauty-in-mathematical-chaos/

Stability

Stable dynamics

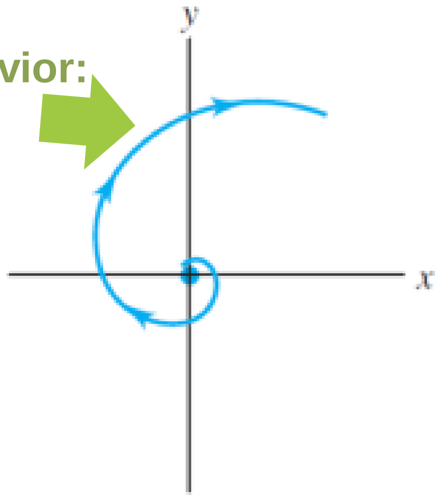


Node
(asymptotically stable)

Short-term behavior:
easy to predict
Long-term behavior:
easy to predict

Unstable dynamics

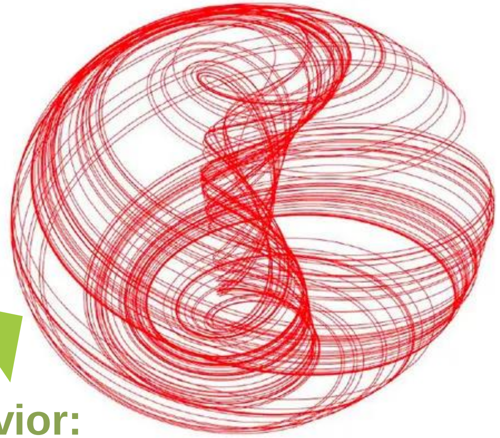
Short-term behavior:
easy to predict
Long-term behavior:
not interesting



Spiral
(unstable)

Chaotic dynamics

(special type of unstable)



Short-term behavior:
hard to predict
Long-term behavior:
Impossible to predict

Image sources: https://logancollinsblog.com/2018/01/27/notes-on-dynamical-systems, https://www.cnet.com/science/biology/features/what-is-the-butterfly-effect-how-scientists-find-beauty-in-mathematical-chaos/

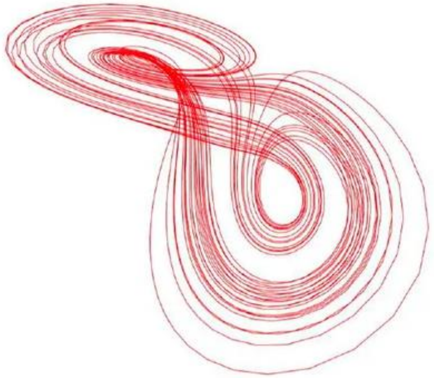
Chaotic dynamics

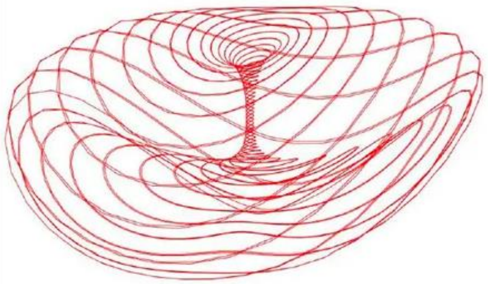
- Chaos is aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions (Strogatz 2024)
- Dynamics “stretch and fold” volumes in phase space

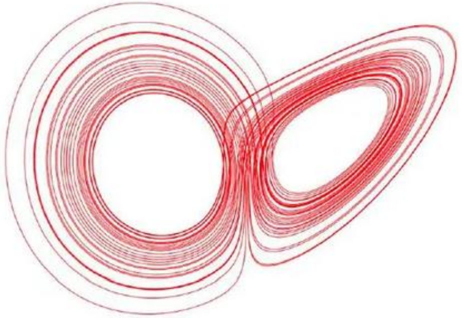
Video source: https://www.youtube.com/watch?v=kN1fpqlSleE&t=33s

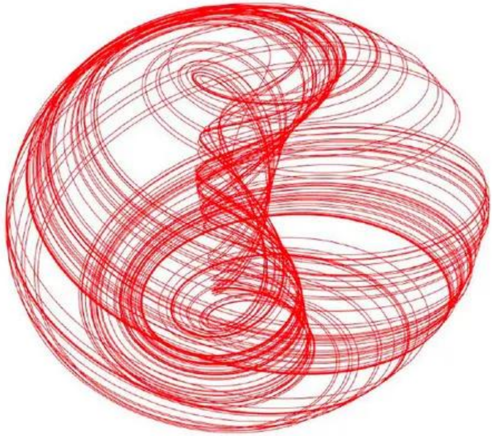
Strange attractors

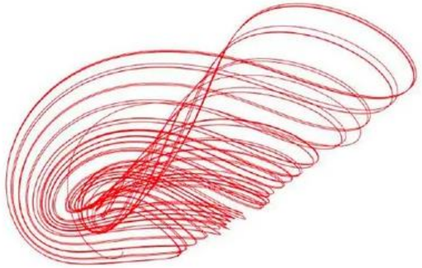
- Trajectories stay within a bounded region ("box")
- Quasiperiodic dynamics
- Minimal box dimension is 3











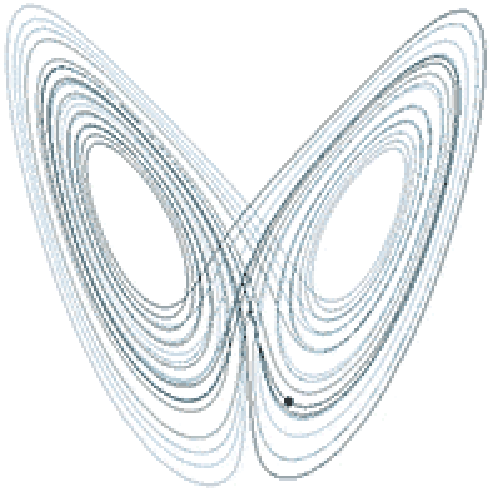


Image source: Wikipedia, https://www.cnet.com/science/biology/features/what-is-the-butterfly-effect-how-scientists-find-beauty-in-mathematical-chaos/

Lyapunov exponents

- Rate λ of divergence of trajectories with an initially small displacement $\delta X(0)$

$$|\delta X(t)| = e^{\lambda t}|\delta X(0)|$$

- Can change with time/ along a trajectory
- Used to characterize chaotic dynamics
- Poses limit on predictability
- Predictability horizon is $1/\lambda$

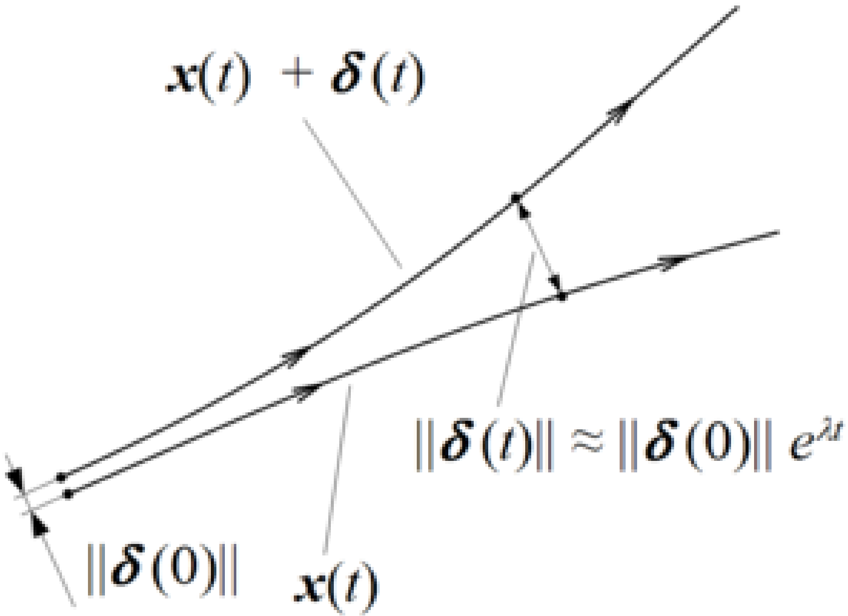
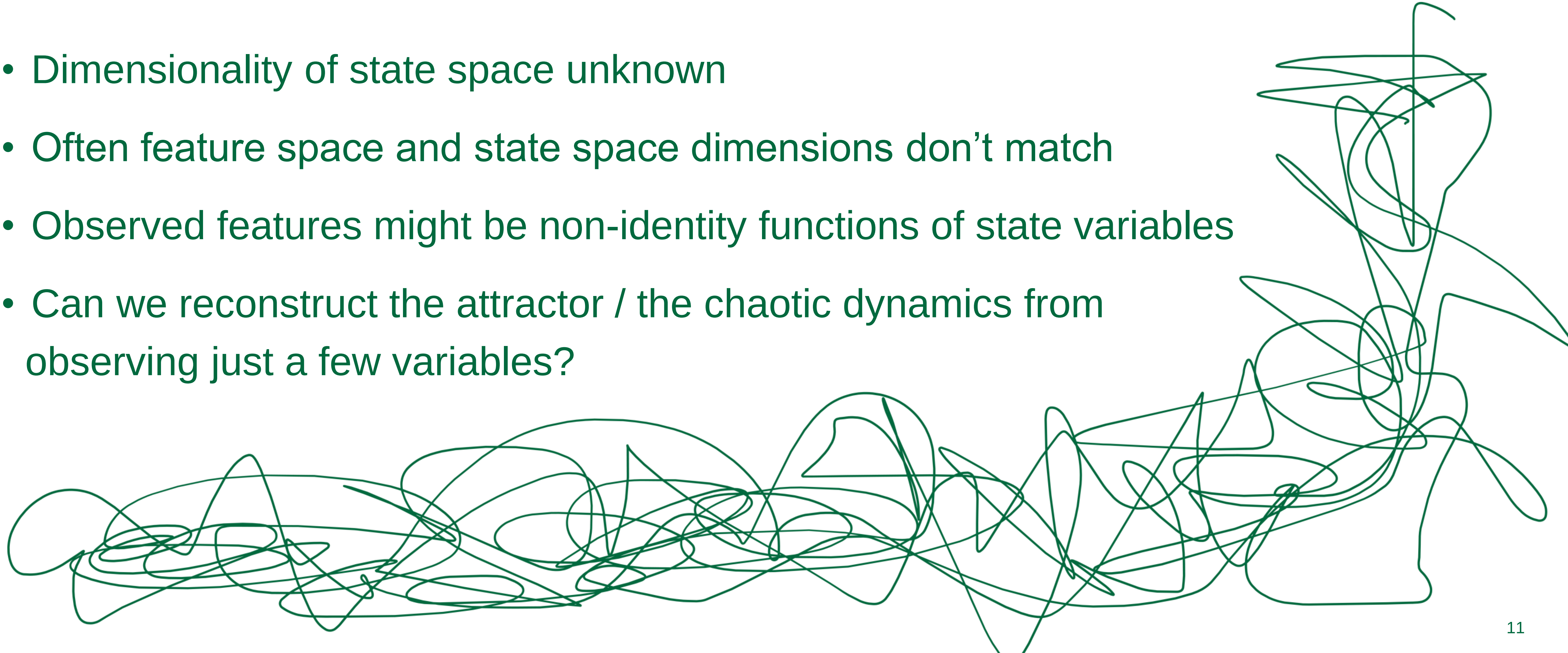


Image source: Wikipedia

Time-delayed embedding

Challenges in predicting chaotic dynamics

- Dimensionality of state space unknown
- Often feature space and state space dimensions don’t match
- Observed features might be non-identity functions of state variables
- Can we reconstruct the attractor / the chaotic dynamics from observing just a few variables?



Taken's theorem

> Let M be a compact manifold of dimension m. For pairs (ϕ, y), where $\phi : M \to M$ is a smooth diffeomorphism (an invertible function that maps one differentiable manifold to another such that both the function and its inverse are smooth) and $y : M \to \mathbb{R}$ a smooth function, it is a generic property that the $(2m+1)$--delay observation map $\Phi_{(\phi,y)} : M \to \mathbb{R}^{2m+1}$ given by
>
> $$\Phi_{(\phi,y)}(x) = \left(y(x), y \circ \phi(x), \ldots, y \circ \phi^{2m}(x)\right)$$
>
> is an embedding; by `smooth' we mean at least C^2.

$\Phi_{(\phi,y)}$

- is invertible and has invertible first derivative
- preserves structure

Attractor reconstruction via time-delayed embedding

1. Measure time series (1D feature space)
2. Feature augmentation via time-delayed embedding with dimension d and time lag Δt
3. Reconstruct attractor in augmented feature space
4. Use augmented features for prediction

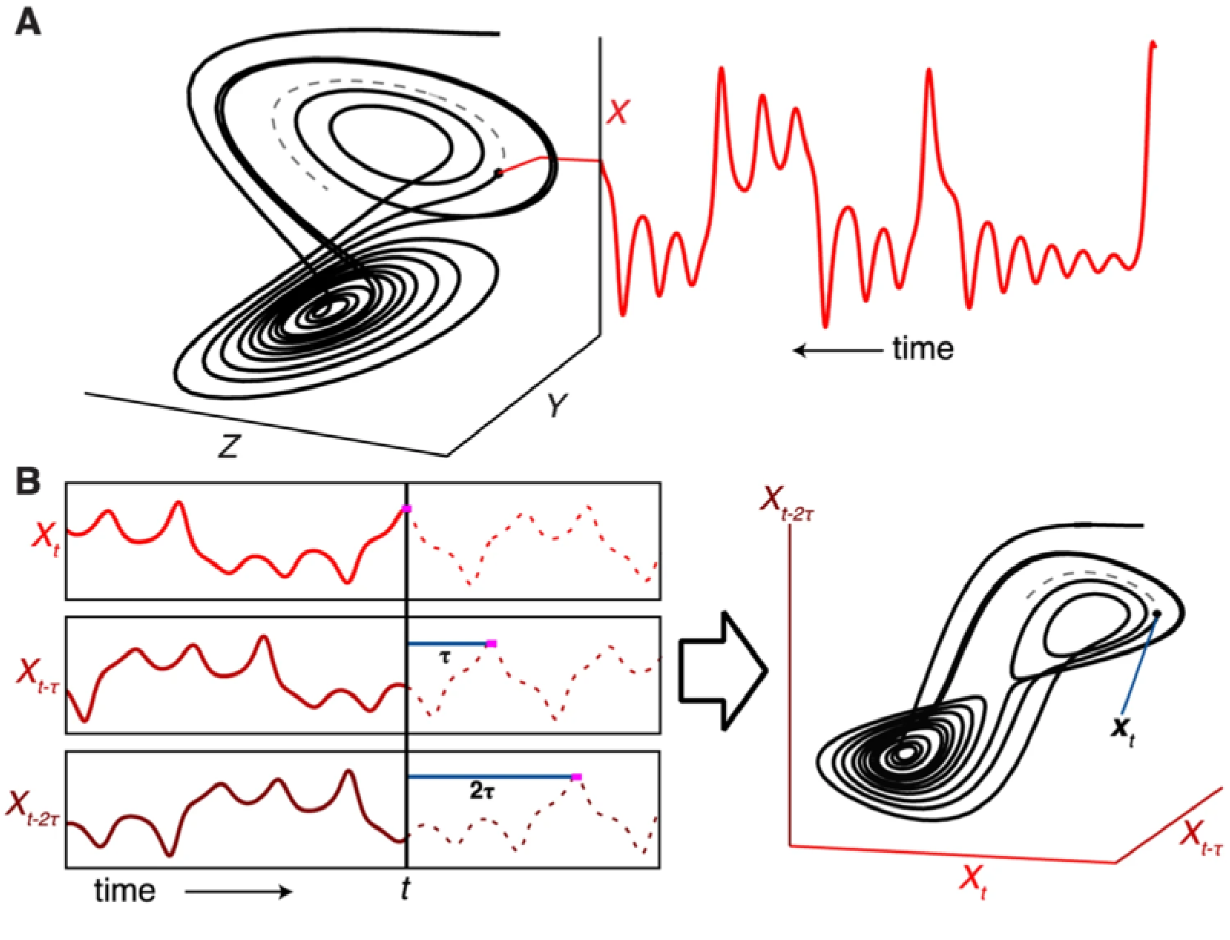


Image source: Giron-Nava, A., Munch, S.B., Johnson, A.F. *et al.* Circularity in fisheries data weakens real world prediction. *Sci Rep* **10**, 6977 (2020). https://doi.org/10.1038/s41598-020-63773-3

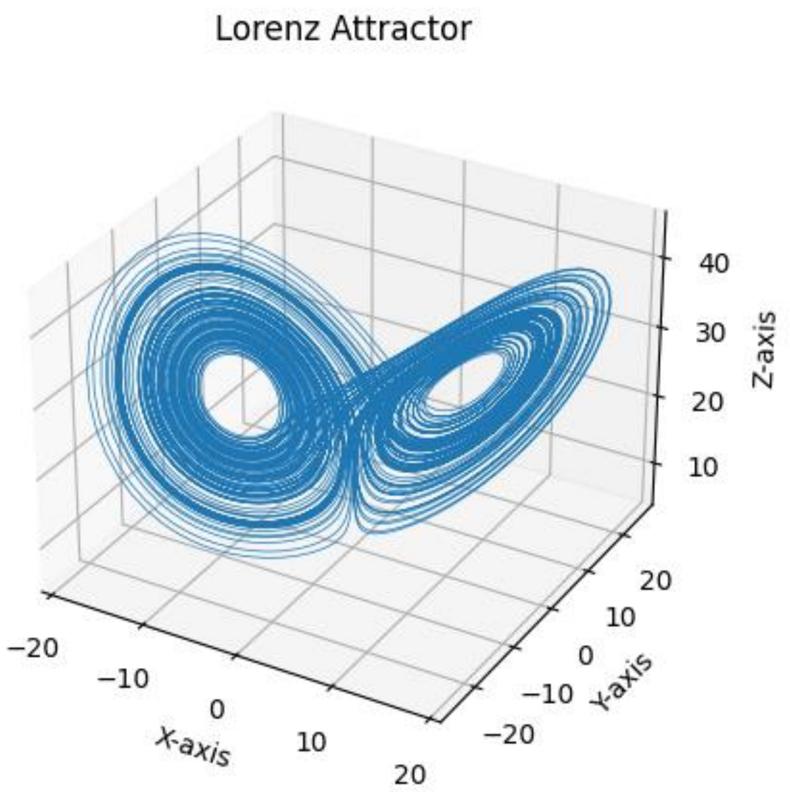
Example: Lorenz attractor

1. Synthesize chaotic dynamics

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

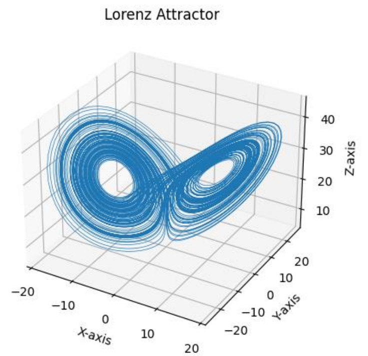
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

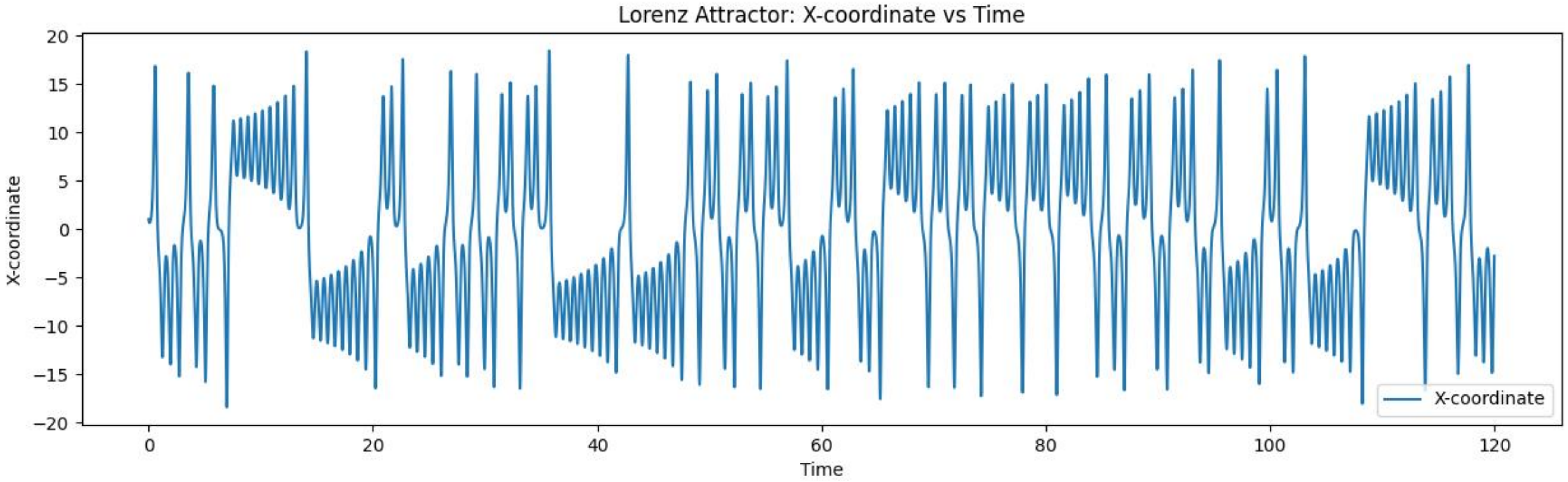


Source: Wikipedia, https://verzep.github.io/Takens/

Example: Lorenz attractor

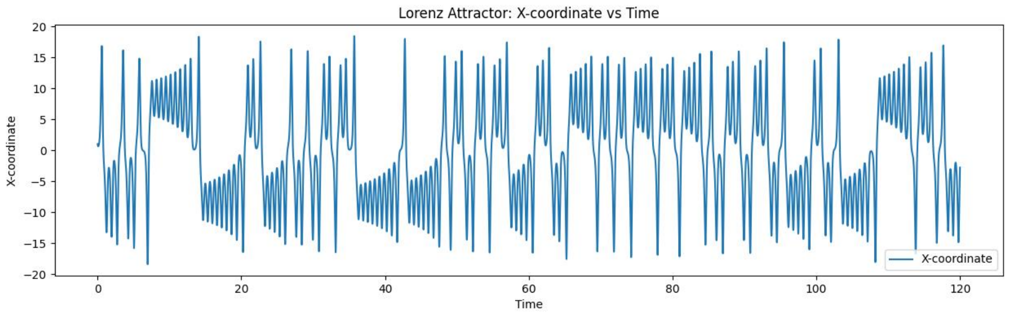
2. Observe one variable





Source: https://verzep.github.io/Takens/

Example: Lorenz attractor



3. High-dimensional time-delayed feature mapping

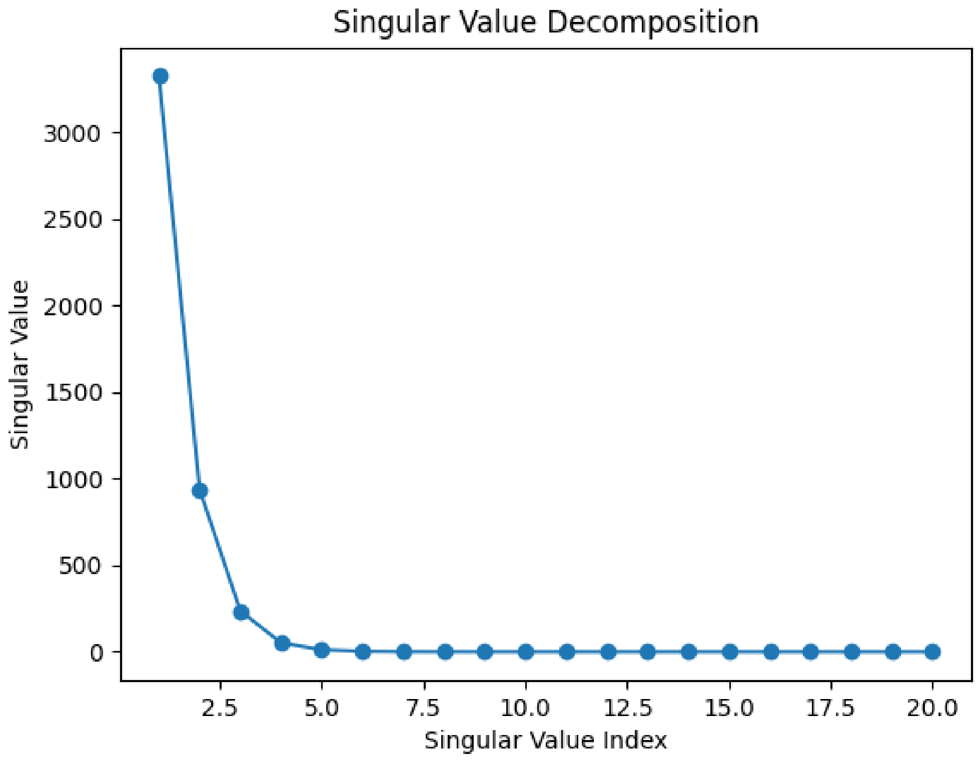
$$x_t = \begin{bmatrix} x_t^1 \\ x_t^2 \\ \dots \\ x_t^D \end{bmatrix} = \begin{bmatrix} u_t \\ u_{t-1} \\ \dots \\ u_{t-(D-1)} \end{bmatrix} = \begin{bmatrix} h(s_t) \\ h(s_{t-1}) \\ \dots \\ h(s_{t-(D-1)}) \end{bmatrix} =: F_D(s_t)$$

Source: https://verzep.github.io/Takens/

Example: Lorenz attractor

$$x_t = \begin{bmatrix} x_t^1 \\ x_t^2 \\ \dots \\ x_t^D \end{bmatrix} = \begin{bmatrix} u_t \\ u_{t-1} \\ \dots \\ u_{t-(D-1)} \end{bmatrix} = \begin{bmatrix} h(s_t) \\ h(s_{t-1}) \\ \dots \\ h(s_{t-(D-1)}) \end{bmatrix} =: F_D(s_t)$$

3. Dimension reduction

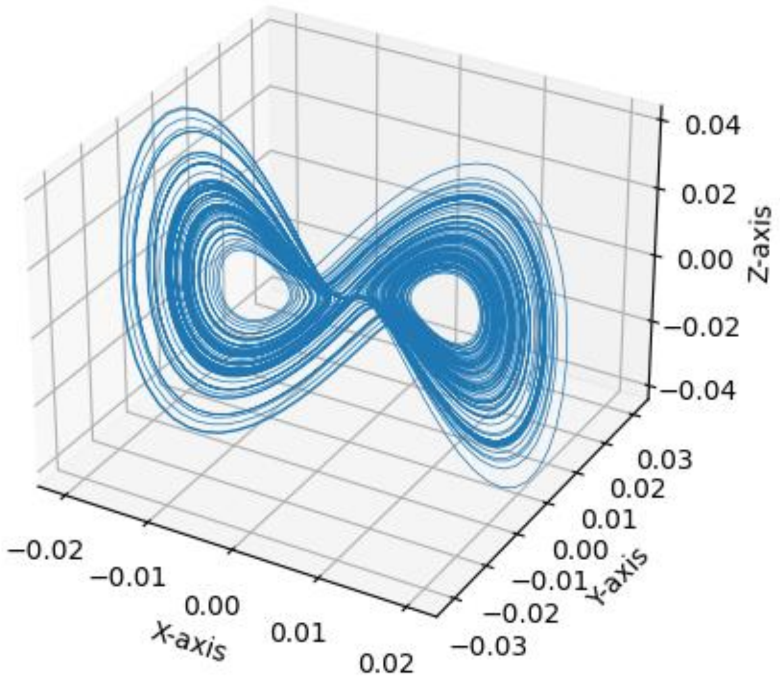


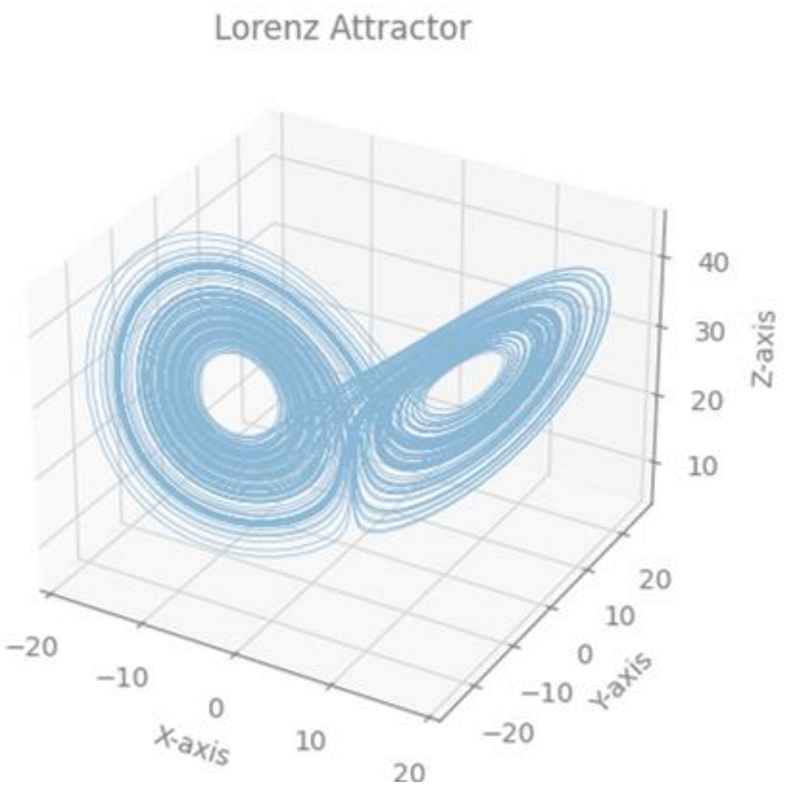
Source: https://verzep.github.io/Takens/

Example: Lorenz attractor

- What can we do with the embedded coordinates?

➢ Attractor reconstruction





Source: https://verzep.github.io/Takens/

Example: Lorenz attractor

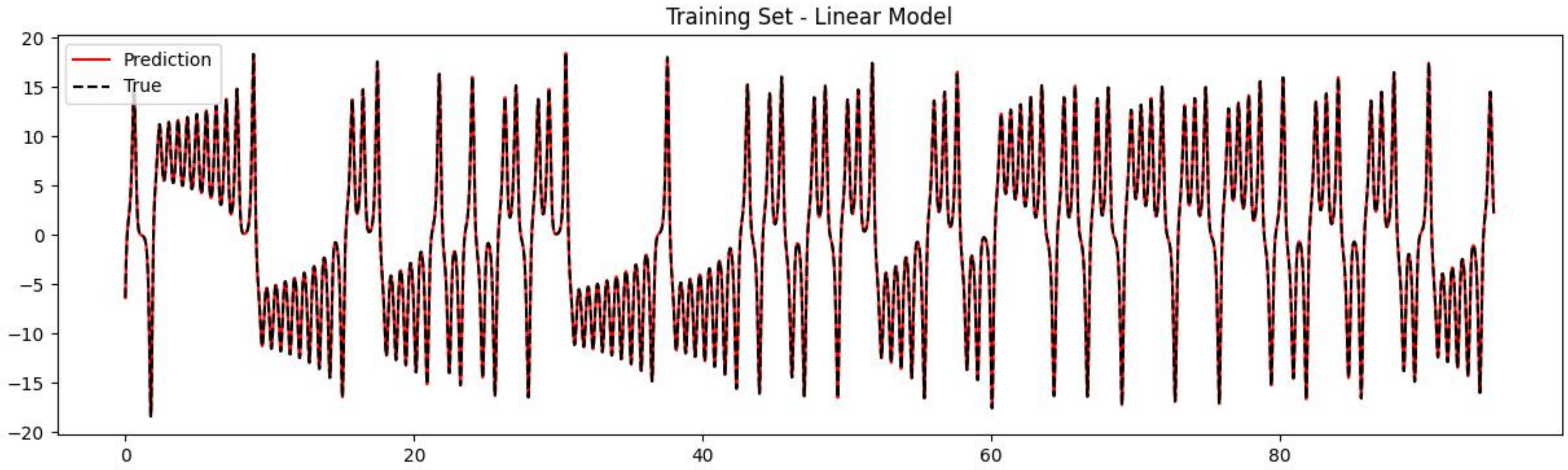
- What can we do with the embedded coordinates?

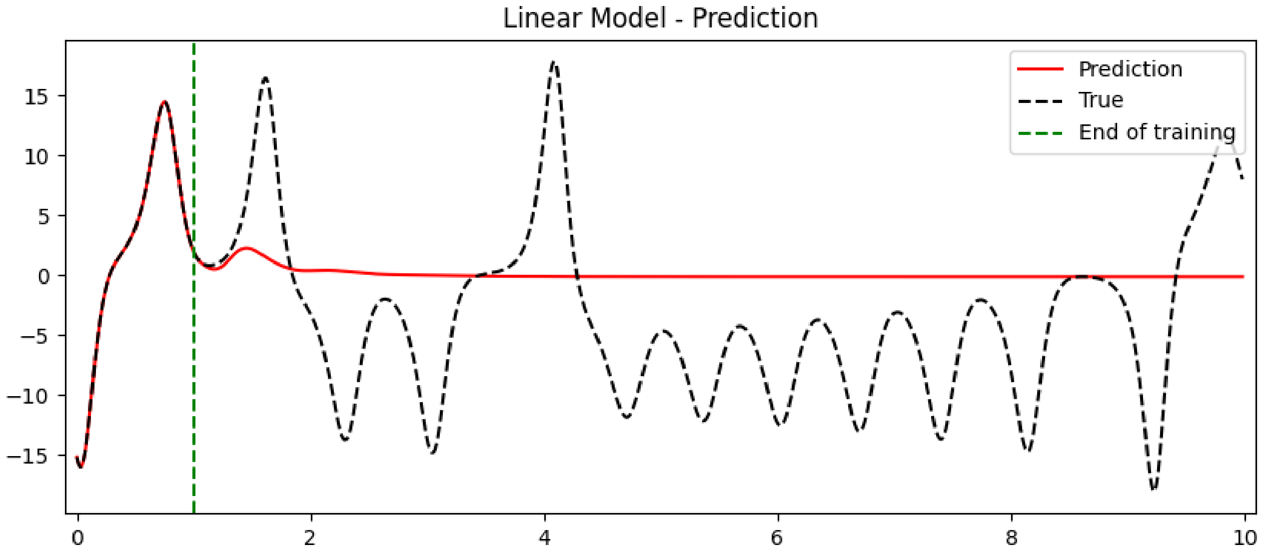
➢ Attractor reconstruction

➢ Prediction (with linear model)

```
from sklearn import linear_model

X = embedded_data[:-1,:]
Y = embedded_data[1:,-1]

linear_predictor = linear_model.Ridge(alpha = 1e-5)

linear_predictor.fit(X,Y)
```

Source: https://verzep.github.io/Takens/





Source: https://verzep.github.io/Takens/

Example: Lorenz attractor

- What can we do with the embedded coordinates?

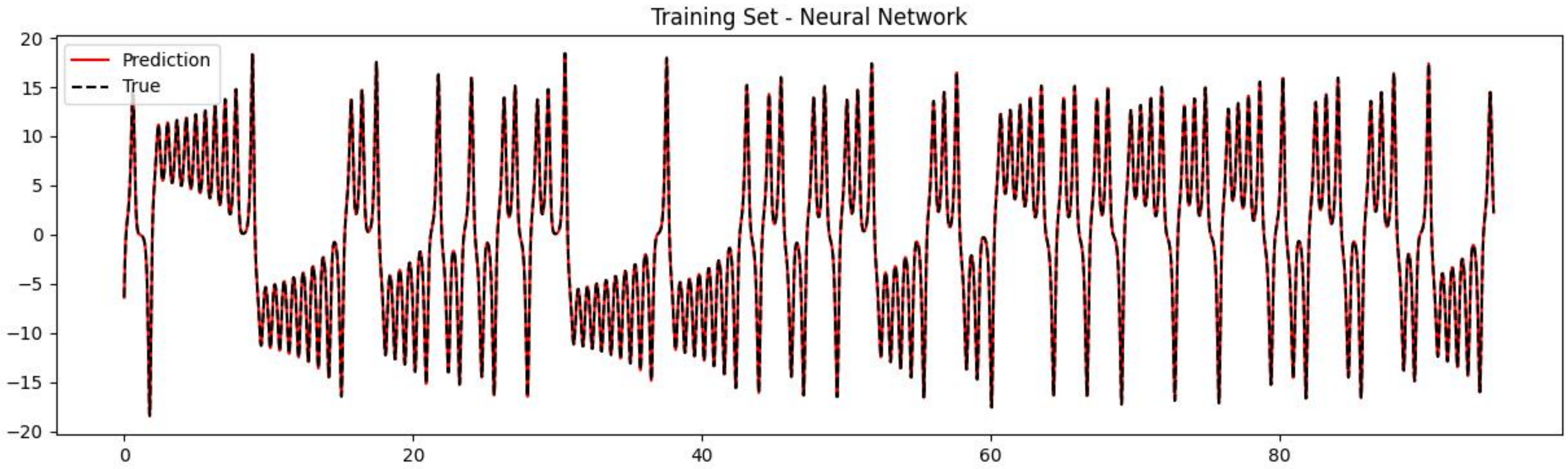
➢ Attractor reconstruction

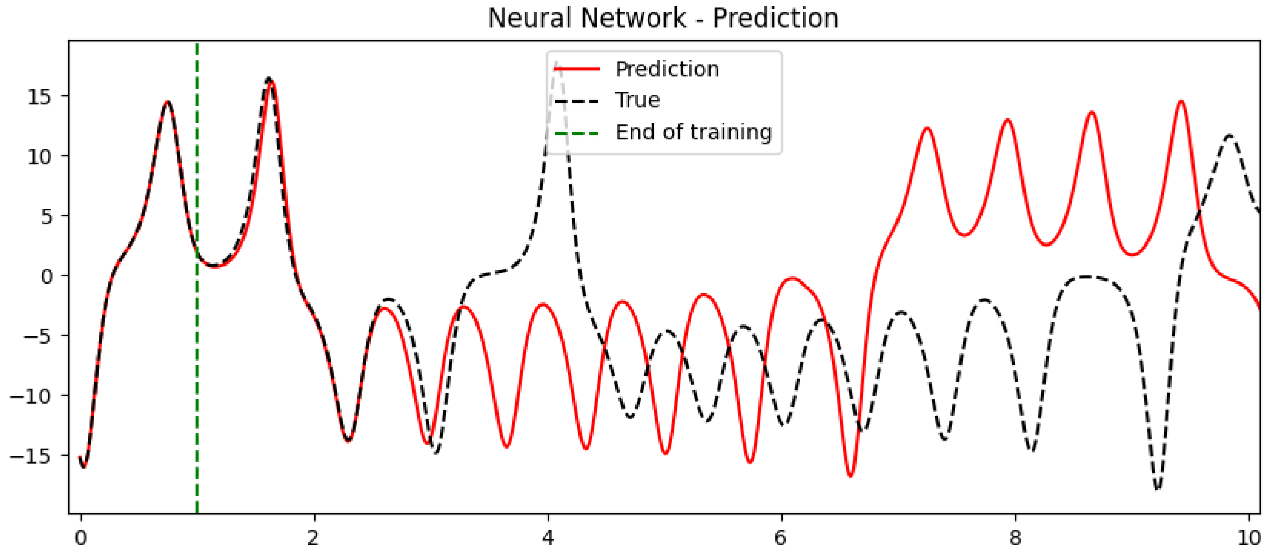
➢ Prediction (with linear model)

➢ Prediction (with nonlinear model)

```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import TensorDataset, DataLoader
import matplotlib.pyplot as plt

# Convert NumPy arrays to PyTorch tensors
X_tensor = torch.tensor(X, dtype=torch.float32)
Y_tensor = torch.tensor(Y, dtype=torch.float32)

# Create a simple neural network model
class NeuralNetwork(nn.Module):
    def __init__(self, input_dim, hidden_dim):
        super(NeuralNetwork, self).__init__()
        self.fc1 = nn.Linear(input_dim, hidden_dim)
        self.fc2 = nn.Linear(hidden_dim, hidden_dim)
        self.fc3 = nn.Linear(hidden_dim, 1)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Source: https://verzep.github.io/Takens/





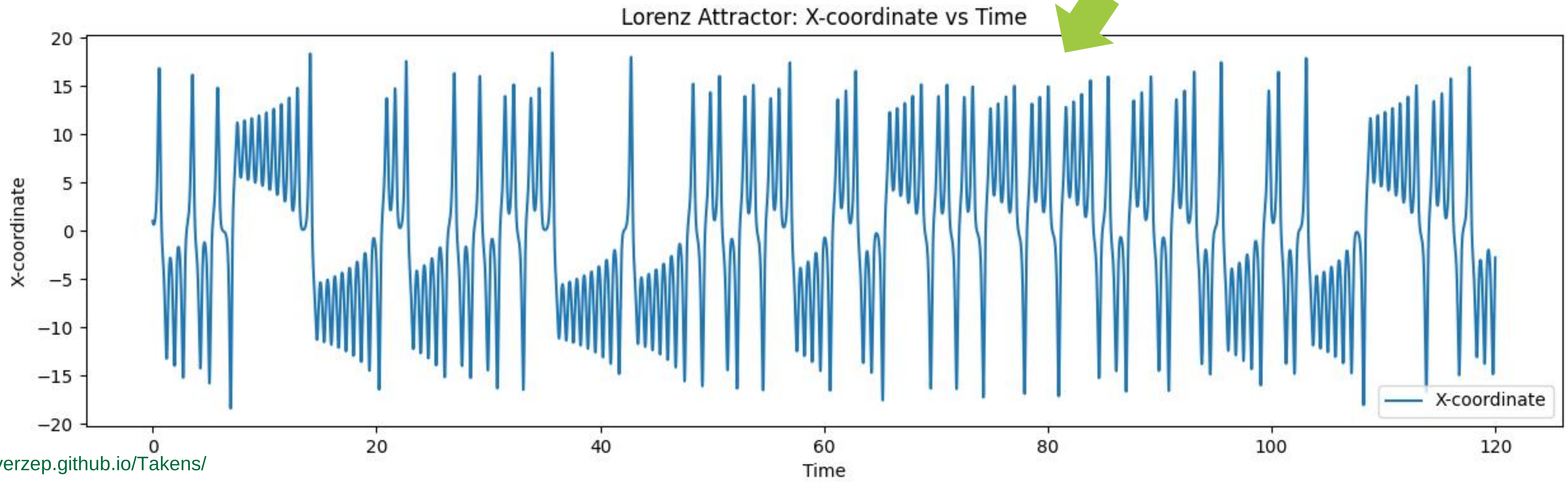
Source: https://verzep.github.io/Takens/

Reservoir computing

Digital twins of chaotic systems

- Create a digital chaotic system to model a real chaotic system
- How similar do the two chaotic systems need to be?

aperiodic/ quasiperiodic



Source: https://verzep.github.io/Takens/

Reservoir computing

- Observation: The weights that change the most when training RNNs are the weights to the output layer
- Idea: Create an RNN in which most weights are fixed

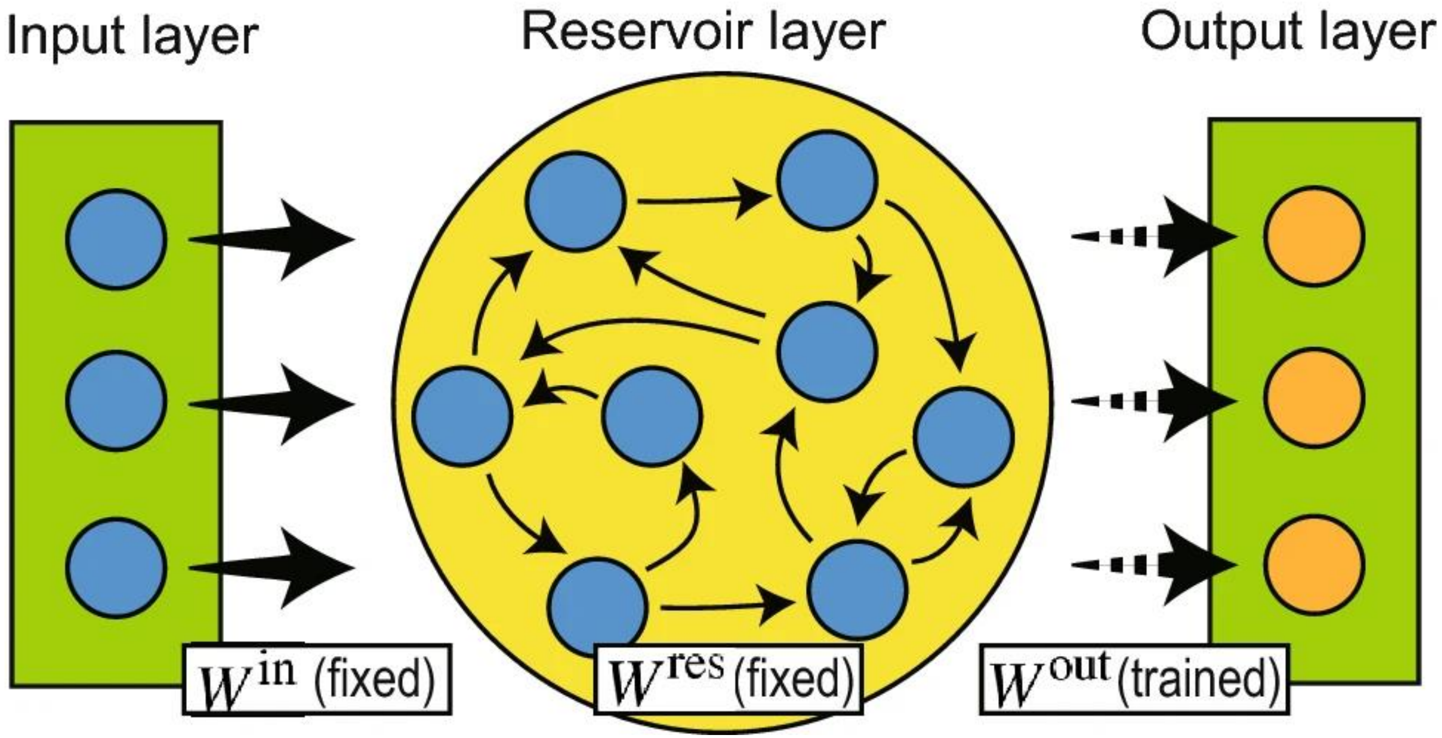
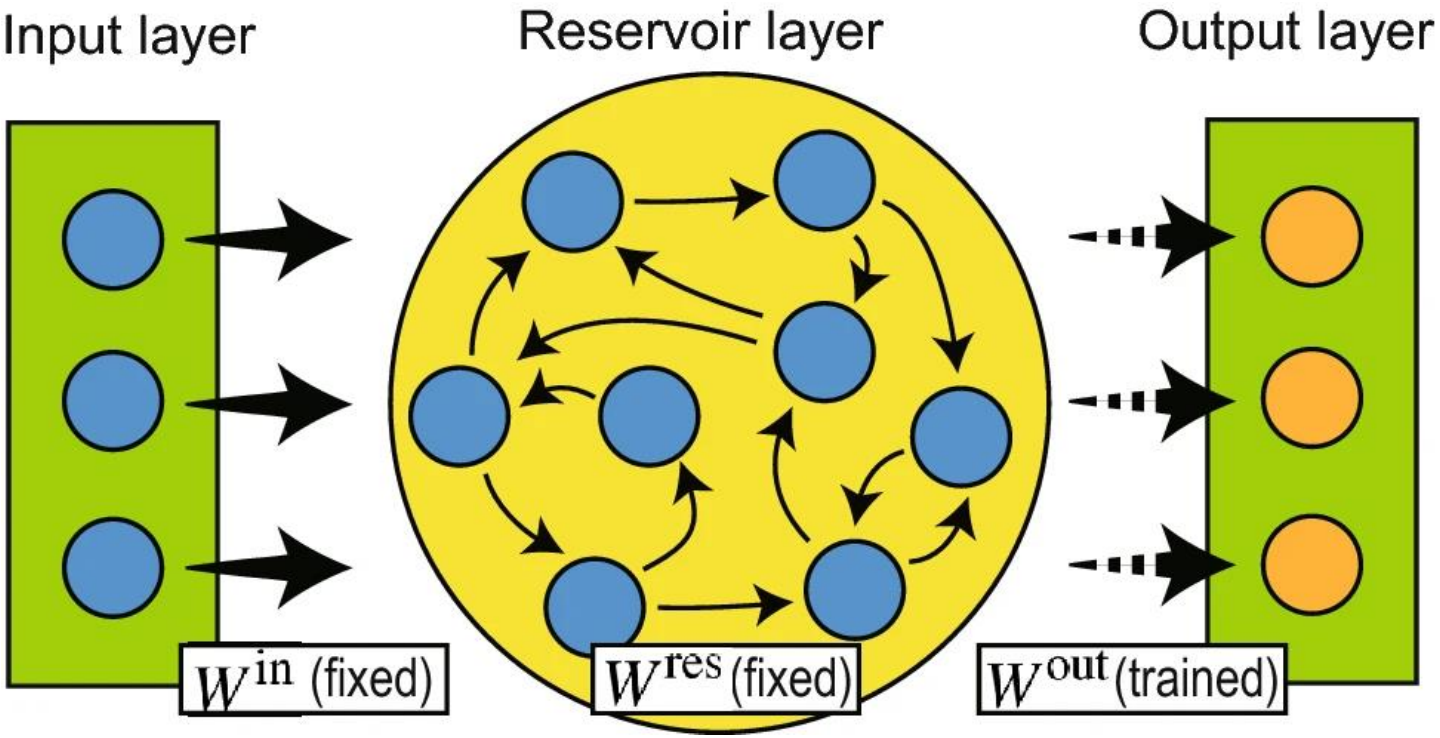


Image source: Sakemi, Y., Morino, K., Leleu, T. *et al.* Model-size reduction for reservoir computing by concatenating internal states through time. *Sci Rep* **10**, 21794 (2020). https://doi.org/10.1038/s41598-020-78725-0

Echo state networks (ESNs)

- Three states:
  - Input vector $x^{(in)}$
  - Reservoir state $x^{(res)}$
  - Output vector $x^{(out)}$



$x^{(in)}$ $x^{(res)}$ $x^{(out)}$

Image source: Sakemi, Y., Morino, K., Leleu, T. *et al.* Model-size reduction for reservoir computing by concatenating internal states through time. *Sci Rep* **10**, 21794 (2020). https://doi.org/10.1038/s41598-020-78725-0

Echo state networks (ESNs)

- Forward pass

$$x_t^{(res)} = (1-\alpha)x_{t-1}^{(res)} + \alpha f\left(W^{(in)}x_t^{(in)} + W^{(res)}x_{t-1}^{(res)}\right)$$

$$x_t^{(out)} = g\left(W^{(out)}x_{t-1}^{(res)}\right)$$

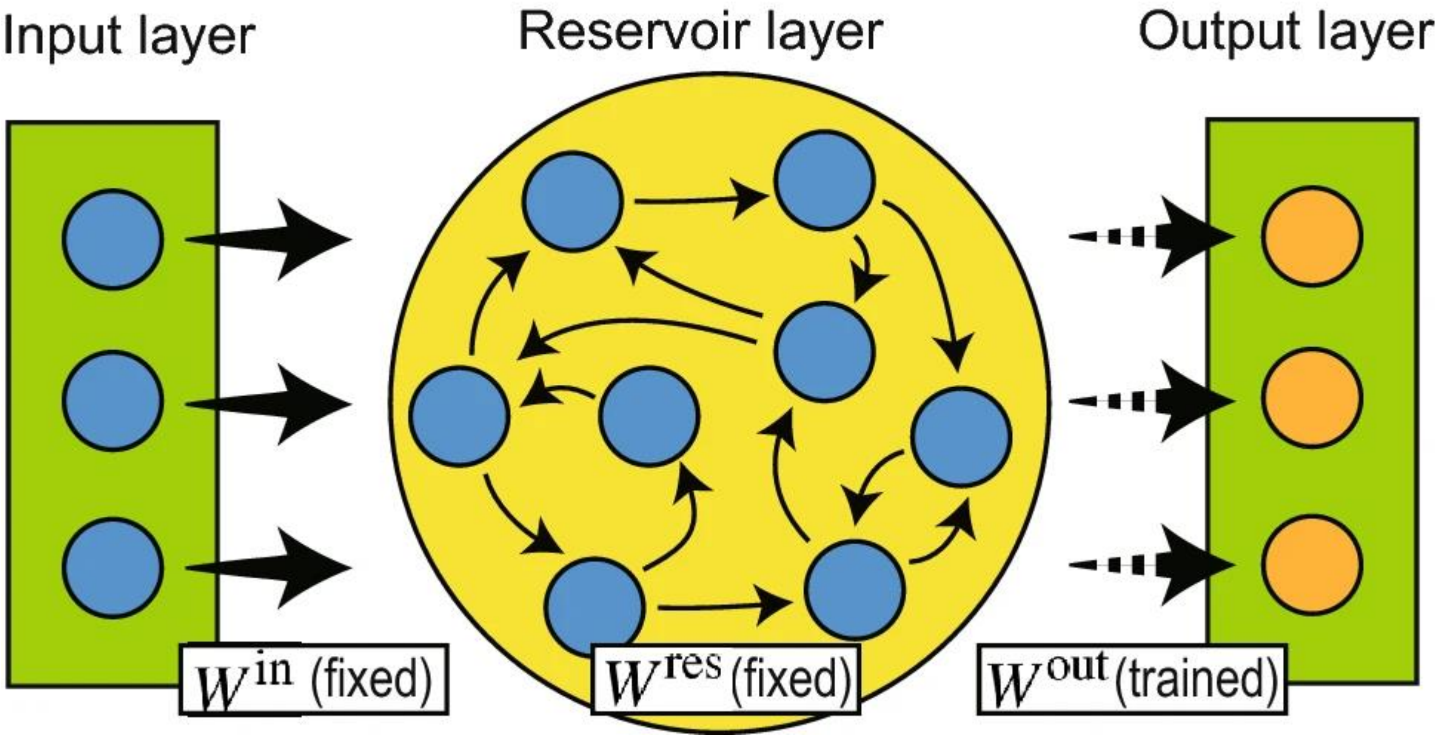


Image source: Sakemi, Y., Morino, K., Leleu, T. *et al.* Model-size reduction for reservoir computing by concatenating internal states through time. *Sci Rep* **10**, 21794 (2020). https://doi.org/10.1038/s41598-020-78725-0

Echo state networks (ESNs)

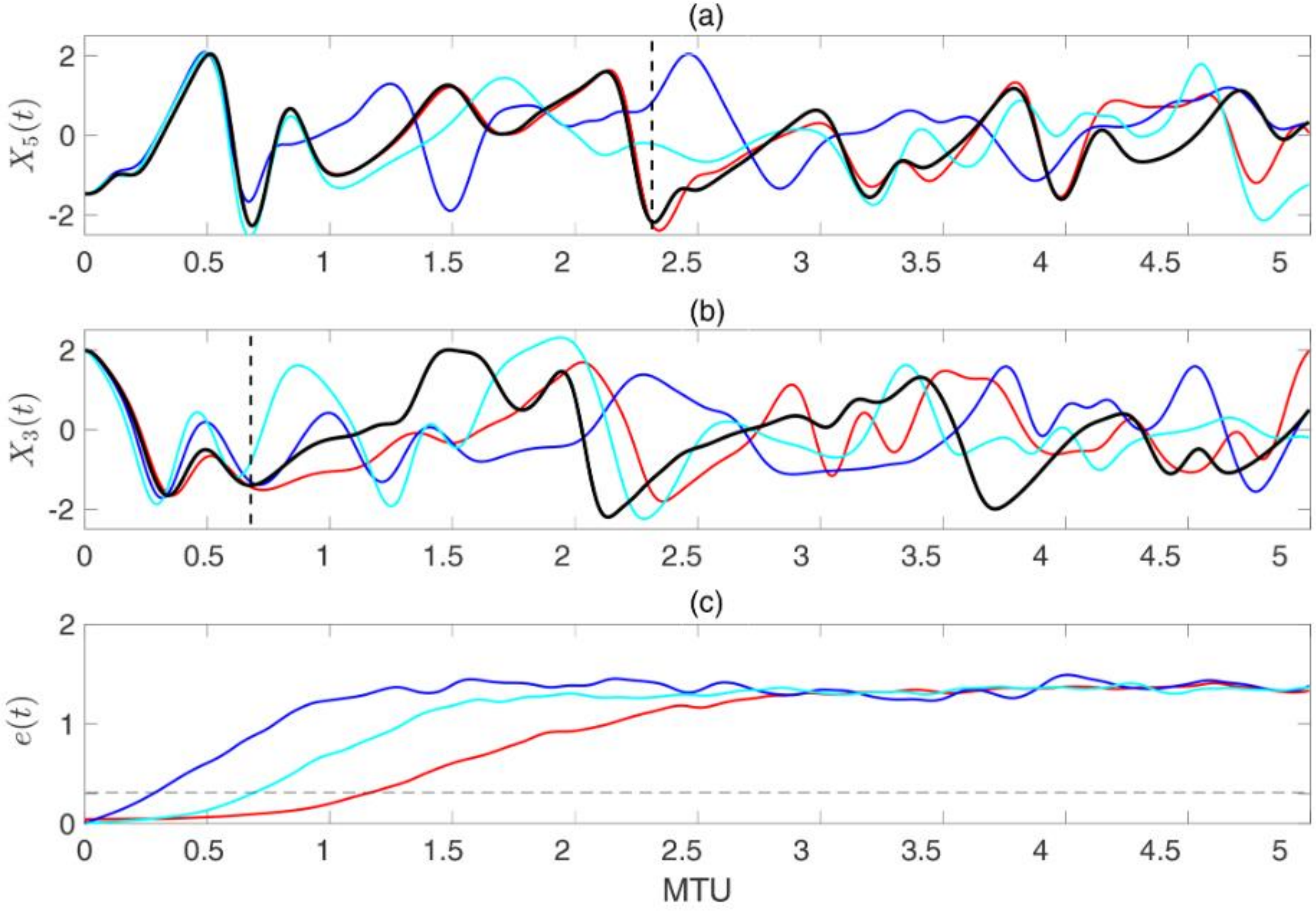


Image source: Chattopadhyay, A., Hassanzadeh, P., and Subramanian, D.: Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network, Nonlin. Processes Geophys., 27, 373–389, https://doi.org/10.5194/npg-27-373-2020, 2020.

Predicting spatio-temporal systems with ESNs

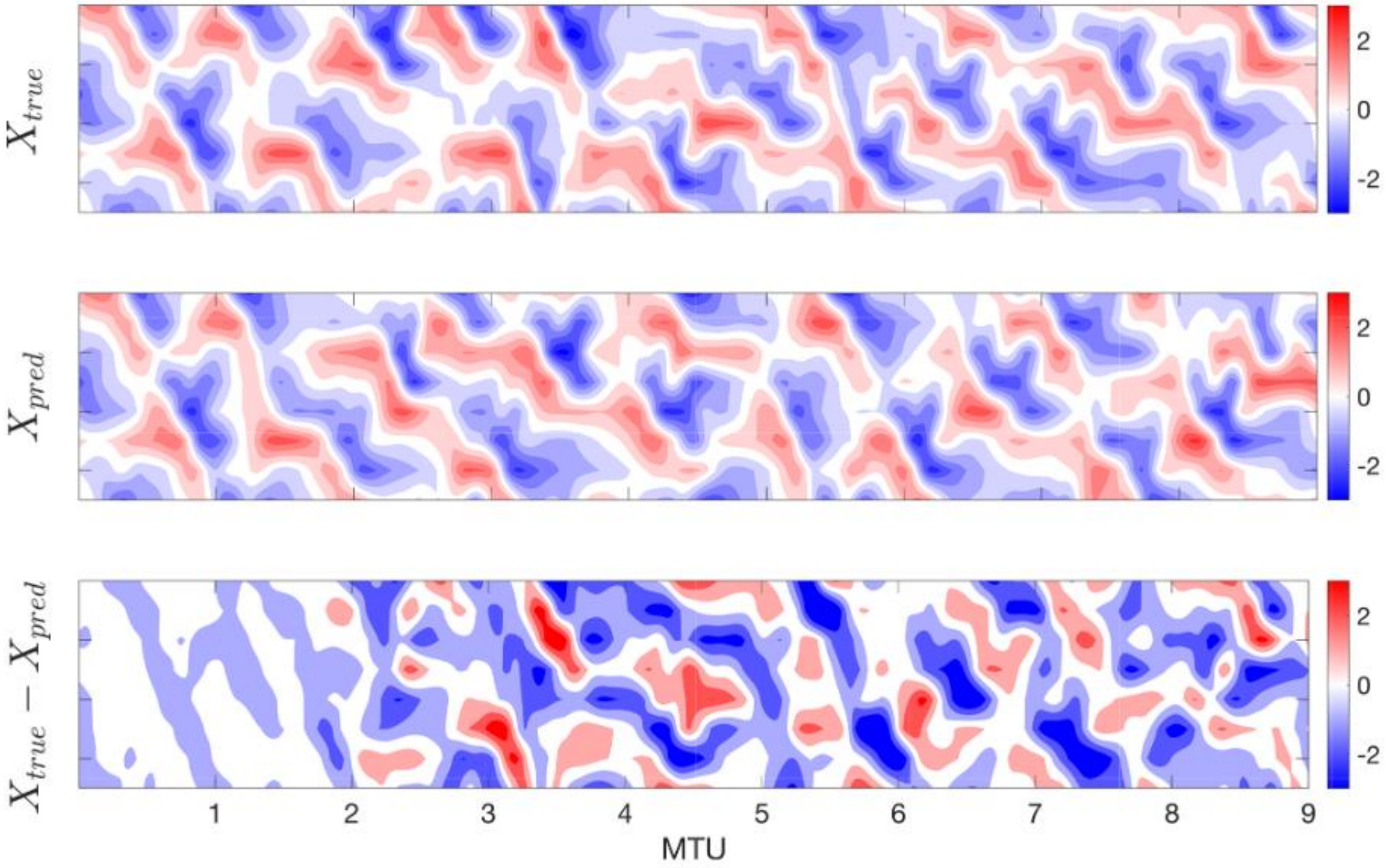


Image source: Chattopadhyay, A., Hassanzadeh, P., and Subramanian, D.: Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network, Nonlin. Processes Geophys., 27, 373–389, https://doi.org/10.5194/npg-27-373-2020, 2020.