Is Chaos Predictable? Learning to Predict Chaotic Systems

Tammy Lam, Faculty Mentors: William Ott, Bhargav Karamched
Department of Mathematics
UNIVERSITY of HOUSTON

Abstract

The dynamics of physiological systems are significantly impacted by delay. The time-delay caused by the transport and processing of chemical components and signals may be of significant consequence. Biological systems present a challenge to model, analyze and predict.

The utilization of machine learning to build mathematical models of complex systems has rapidly grown. For time-dependent series, generally a **recurrent neural network** (RNN), capable of returning past states, is used. In most common RNN implementations, multiple hidden layers are rebalanced during training to achieve adequate results. However, these implementations can be computationally expensive and may require extensive training data.

Here, we utilize a type of RNN called an **echo state network**[1] (ESN), a static, randomly initialized reservoir of nodes. We study two types of physiological systems exhibiting delay: **degrade-and-fire**[2] circuits and the **insulin-glucose**[3] cycle. Manipulating only signal propagation and input smoothing, we model both systems with generated reservoirs.

In future works, we will further tune the reservoir, addressing stability and noise. We will also research the use of other machine learning techniques in determining the optimal parameters for reservoir generation.

Source Code

https://github.com/minbin/esn

Interactive Data

https://minbin.github.io/uh-surf

References

[1] Echo State Network dx.doi.org/10.4249/scholarpedia.2330

[2] W. Mather et al., "Delay-Induced Degrade-and-Fire Oscillations in Small Genetic Circuits" doi:10.1103/PhysRevLett.102.068105

[3] B. Karamched et al., "Delay-induced Uncertainty in Physiological Systems" arXiv:2007.09309

[4] Mackey-Glass Equation doi:10.4249/scholarpedia.6908

Acknowledgments

This research project was funded by the UH Office of Undergraduate Research.

Thank you Wendy for your guidance and support.

RESERVOIR CONSTRUCTION

Our goal is to implement a random, large, fixed recurrent neural network. We utilize 100% neuron connectivity in our hidden layer and do not employ noise injection or other hyper-parameters.

The reservoir is governed by the **state update equation**

$$x_{n+1} = \tanh(W_{in}u_{n+1} + Wx_n + W_{fb}y_n)$$

During state harvesting, we drive the ESN with input sequence $u(1), ..., u(n_{nmax})$ using our state update equation. We obtain the **extended system state** by concatenating reservoir and input states $z(n) = [x(n); u(n)]$. The extended system states are filed row-wise into a **state collection matrix** S.

Data is generated from two models. These serve as both the teacher output and the **desired output** $d(n)$. The **output collection matrix** D is filled row-wise with our desired outputs.

The **desired output weights** W^{out}, used to transform reservoir states to outputs, is computed

$$W^{out} = (S^{\dagger}D)^T \quad {}^{\dagger}{}_{pseudoinverse}$$

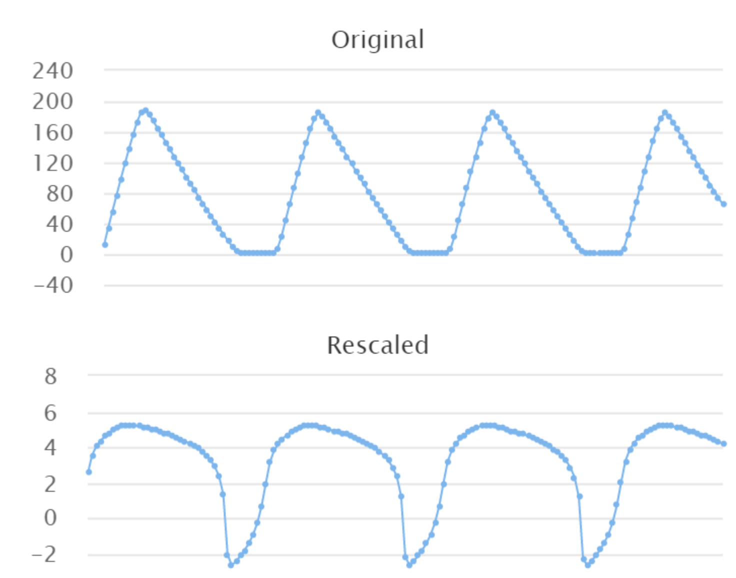
DATA MODEL

DEGRADE-AND-FIRE

We utilize a constructed **synthetic gene oscillator** for our degrade-and-fire model. The mass-action kinetics of our delayed production and degradation is expressed by the delay-differential equation

$$\dot{r} = \frac{\alpha C_0^2}{(C_0 + r_\tau)} - \frac{\gamma_r r(t)}{R_0 + r(t)} - \beta r(t)$$

The degrade-and-fire model displays clustering and excess amplitude. To compensate, we rescale the data using $\ln(r)$.



INSULIN-GLUCOSE

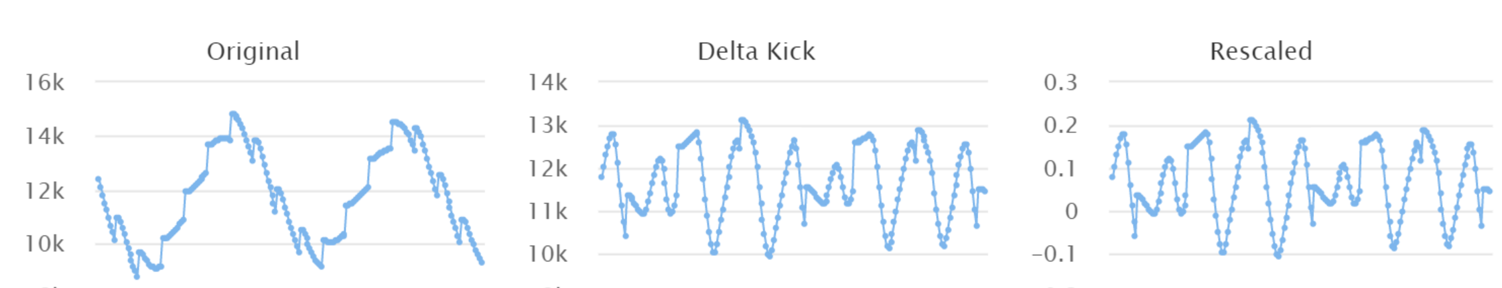
We expand the **Ultradian model** for glucose-insulin production with **delta kicks** for our chaotic series.

Our nutritional driver $I_G(t)$ includes basal signal and pulsatile kicks

$$I_G(t) = I_0 + \sum_{n=0}^{\infty} A_n \delta(t - T_n)$$

We rescale the data by a factor of 10^{-4} and then shift by -1.1 to avoid neuron saturation.

$$\frac{dI_p}{dt} = f_1(G) - E\left(\frac{I_p}{V_p} - \frac{I_i}{V_i}\right) - \frac{I_p}{t_p}$$
$$\frac{dI_i}{dt} = E\left(\frac{I_p}{V_p} - \frac{I_i}{V_i}\right) - \frac{I_i}{t_i}$$
$$\frac{dG}{dt} = f_4(h_3) + I_G(t) - f_2(G) - f_3(I_i)G$$
$$\frac{dh_1}{dt} = \frac{1}{t_d}(I_p - h_1)$$
$$\frac{dh_2}{dt} = \frac{1}{t_d}(h_1 - h_2)$$
$$\frac{dh_3}{dt} = \frac{1}{t_d}(h_2 - h_3)$$



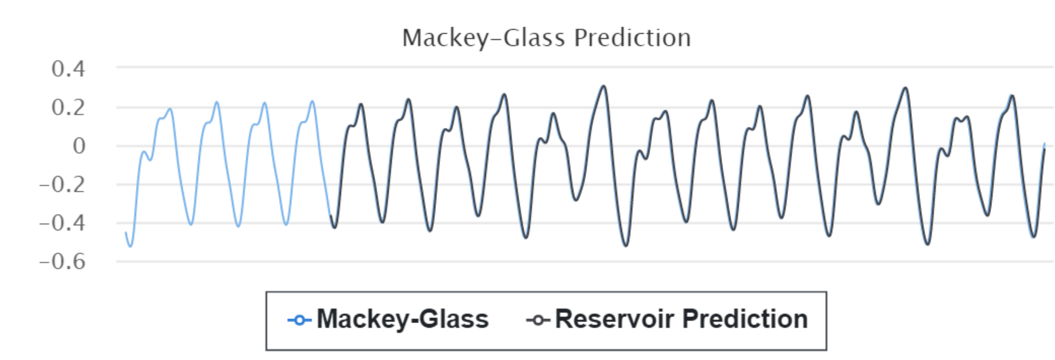
RESERVOIR PREDICTION

Input scaling and spectral radius codetermine the nonlinearity of the reservoir dynamics. Larger amplitudes and spectral radii imply longer-range interactions between neurons slower signal decay. Balancing these two parameters is crucial to obtaining a usable prediction.

BASIC CASE

We start by using the Mackey-Glass[4] nonlinear time delay differential equation

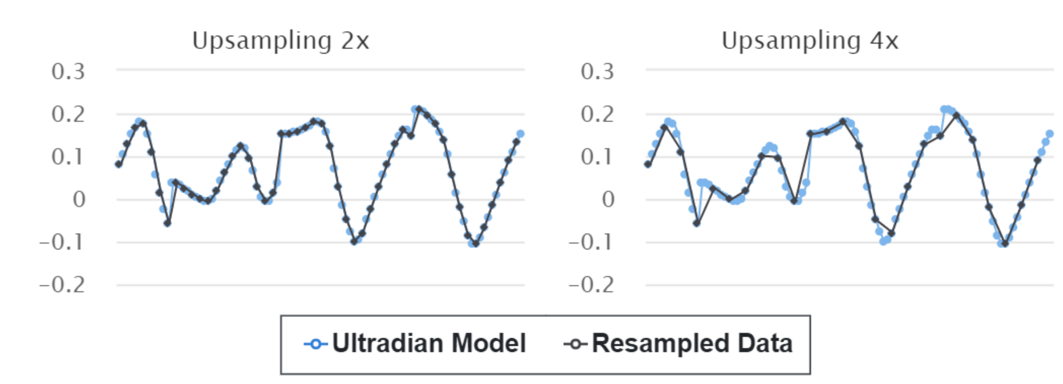
$$\frac{dx}{dt} = \beta\frac{x_\tau}{1 + x_\tau^n} - \gamma x, \quad \gamma, \beta, n > 0$$



Using 500 training data points, 1024 reservoir units and a spectral radius of 1.5, we establish a basis for predicting an oscillating series.

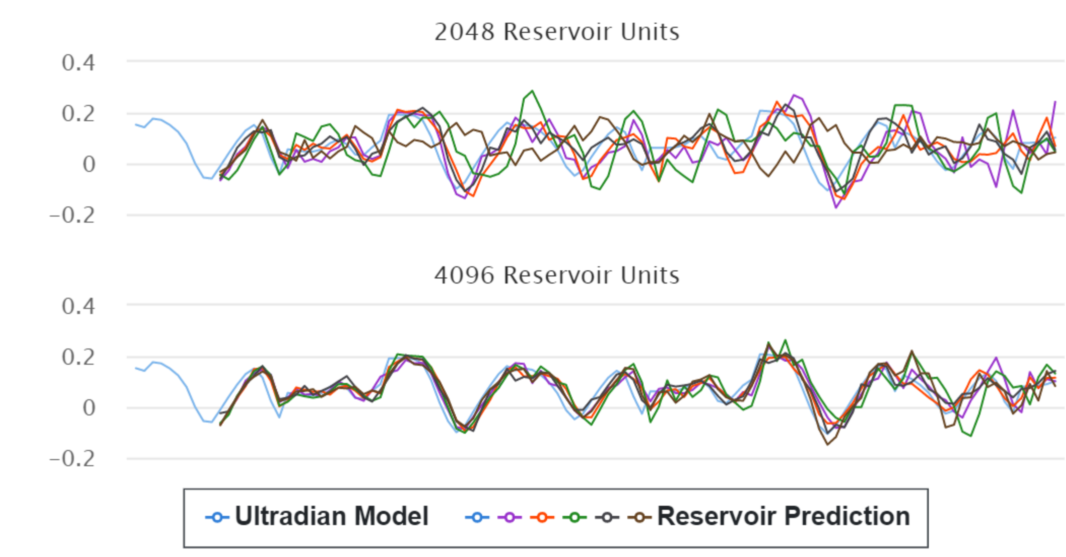
SAMPLING RATE & SATURATION

The arbitrary long periods of oscillations in the degrade-and-fire model compared to delay time are especially difficult to predict. Upsampling by a small factor helps mitigate clustering points and noise while closely tracking the underlying data.



RESERVOIR UNITS

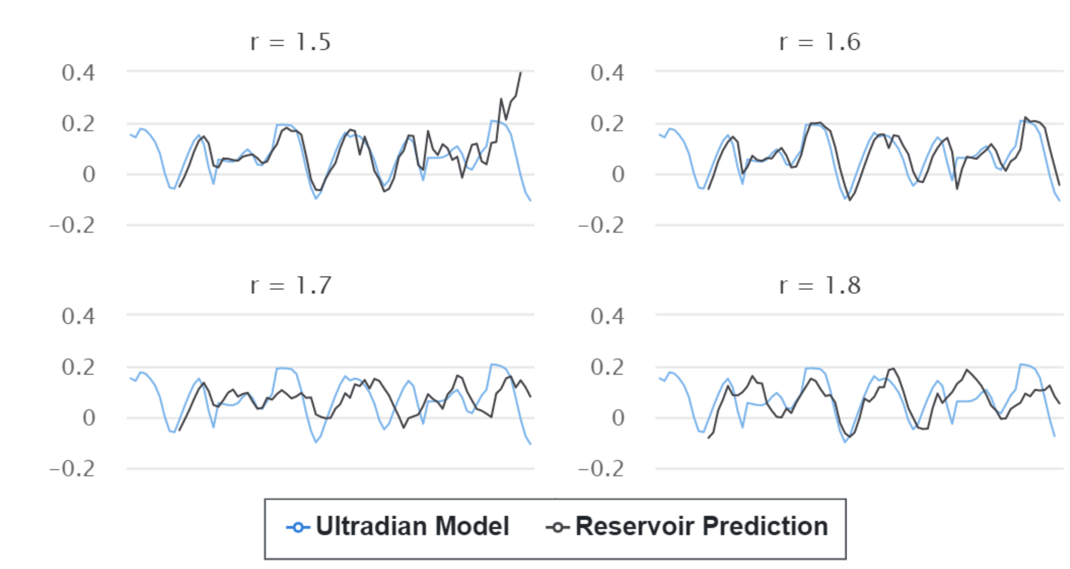
An increase in the amount of reservoir units improves the stability of the prediction.



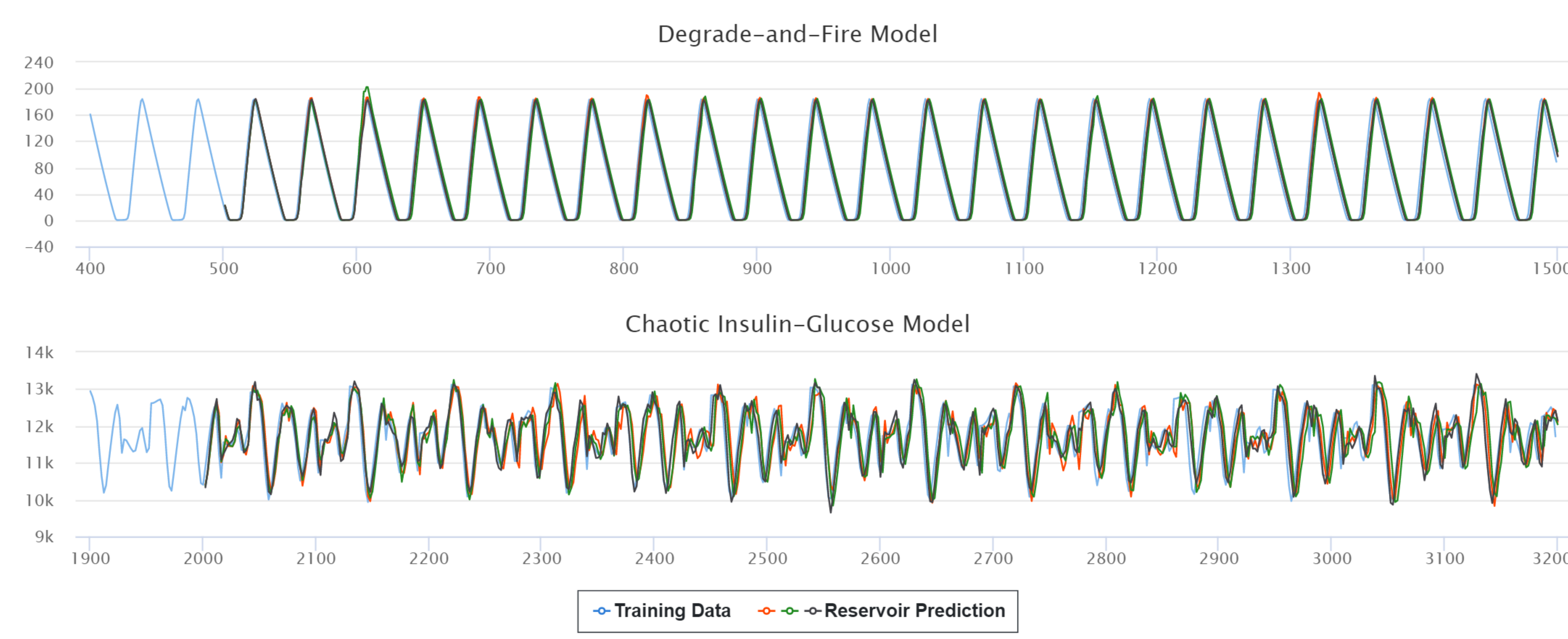
SPECTRAL RADIUS

Spectral radius effectively controls the decay rate and range of interactions. Picking a spectral radius r, we tune the reservoir weight matrix using the ratio

$$W = W * \frac{r}{|\lambda_1^{\dagger} W|}$$



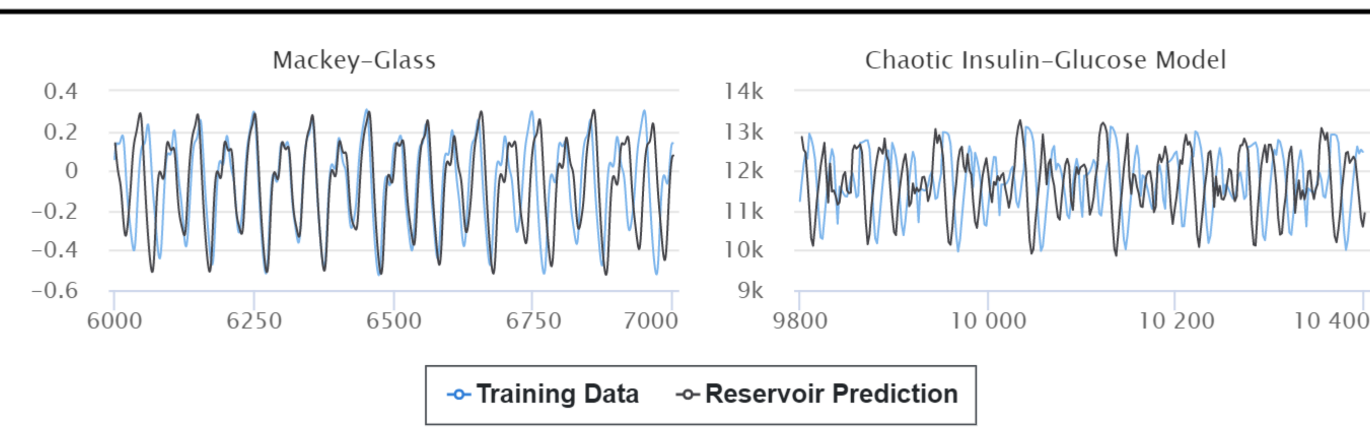
RESULTS



DISCUSSION & FUTURE WORK

With the properly tuned parameters, we can predict our generated models to several thousand steps into the future using a much smaller pool of training data points.

However, after thousands of steps, the predicted model begins to drift from the underlying data while maintaining amplitude. More reservoir units lengthen the period of accurate predictions. Additionally, a chaotic Gaussian insulin-glucose model we generated could not be accurately predicted by our current reservoir past a couple dozen steps. We hope to study and understand the cause of these behaviors in future works.



We want to implement and train a **convolutional training network** (CNN) to recognize the parameters a reservoir needs to predict a series. The two parameters we configured, reservoir units and spectral radius, can not be exhaustively tuned due to the resources needed to generate large reservoirs. We hope to discover a trained solution for picking reservoir generation parameters.