UNIVERSITY OF MARYLAND

Emergent Detection of Concept Drift within the Glia-Inspired 'Rhythmic Sharing' Algorithm

Ian Whitehouse, Hoony Kang, and Wolfgang Losert

ICRC 2025

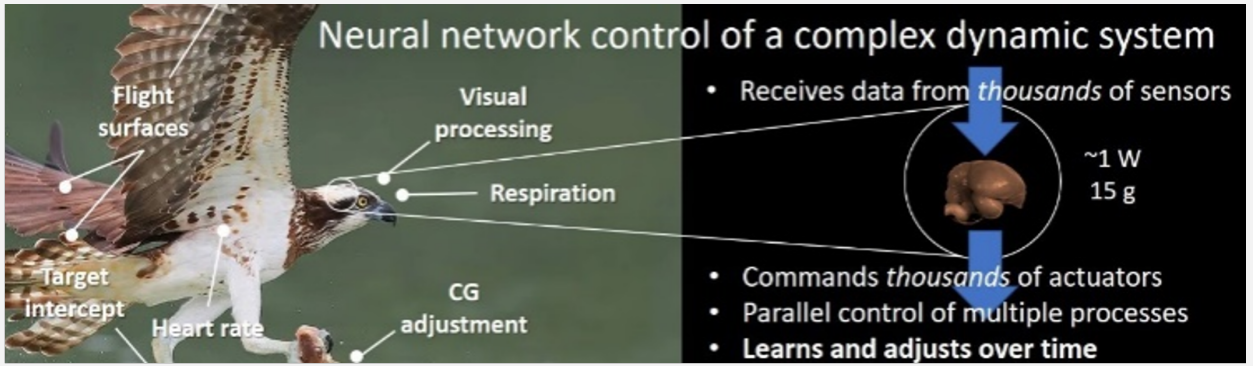
Overview

Summary – Neuromorphic Algorithms

Why do we look to the brain when developing algorithms?

Summary – Neuromorphic Algorithms

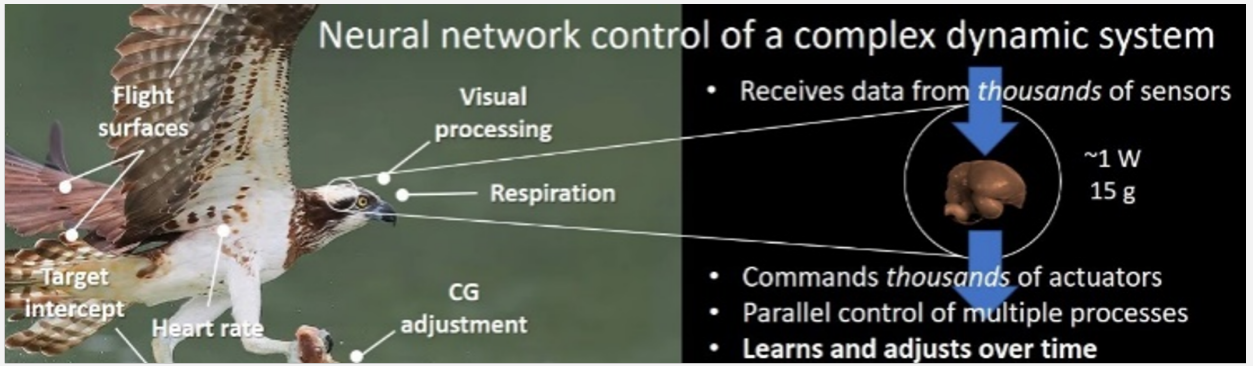
Why do we look to the brain when developing algorithms?



From: ResCon LLC

Summary – Neuromorphic Algorithms

Why do we look to the brain when developing algorithms?



From: ResCon LLC

The brain outperforms machine learning in several tasks, including

- learning and extrapolating from limited data with different modalities
- anticipating, detecting, and adapting to concept drift

Summary – Our Lab

Losert Lab works on the intersection of neuroscience research and computation. We're interested in:

- astrocytes as conductors of neuronal cognition
- hybrid systems that combine neuronal cultures and digital computers
- Neuromorphic algorithms unify theories of neural cognition and computation

Summary – Our Lab

Losert Lab works on the intersection of neuroscience research and computation. We're interested in:

- astrocytes as conductors of neuronal cognition
- hybrid systems that combine neuronal cultures and digital computers
- Neuromorphic algorithms unify theories of neural cognition and computation

One of these algorithms is rhythmic sharing, today's topic

This Presentation

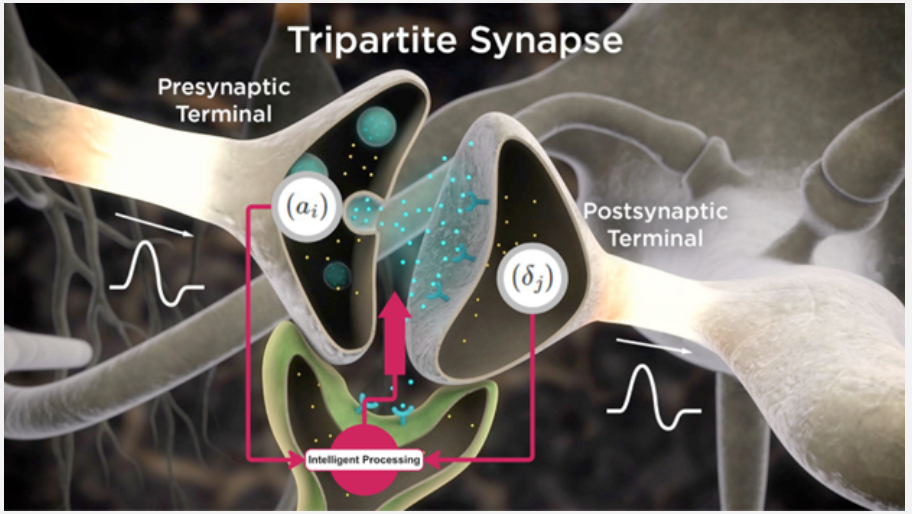
I'll give an overview of:

1. The astrocytic inspiration of our rhythmic sharing algorithm
2. The rhythmic sharing algorithm, and how per-input synchrony enhances drift in a dataset
3. The concept drift detection task
4. The result of combining rhythmic sharing with drift detection algorithms on complex datasets

Biological Inspiration

Astrocytes and Glial Cells

Astrocytes are glial cells involved in cognition through the tripartite synapse [1].



Astrocytes and Glial Cells

Through the tripartite synapse, astrocytes are essential to cognition:

- Each astrocyte is connects to between 270,000 and 2 million synapses, coordinating and monitoring them [2]
- An astrocyte can be incredibly long, connecting to spatially-distant neurons
- The astrocytes respond to the body's state and to external stimulation [3]

Astrocytes and Glial Cells

Through the tripartite synapse, astrocytes are essential to cognition:

- Each astrocyte is connects to between 270,000 and 2 million synapses, coordinating and monitoring them [2]
- An astrocyte can be incredibly long, connecting to spatially-distant neurons
- The astrocytes respond to the body's state and to external stimulation [3]

One way they potentially control neurons is through rhythmic, physical force on the synapses [4]. This inspired the rhythmic sharing algorithm.

Rhythmic Sharing

RS Algorithm

The rhythmic sharing algorithm was originally proposed in Kang and Losert [5].

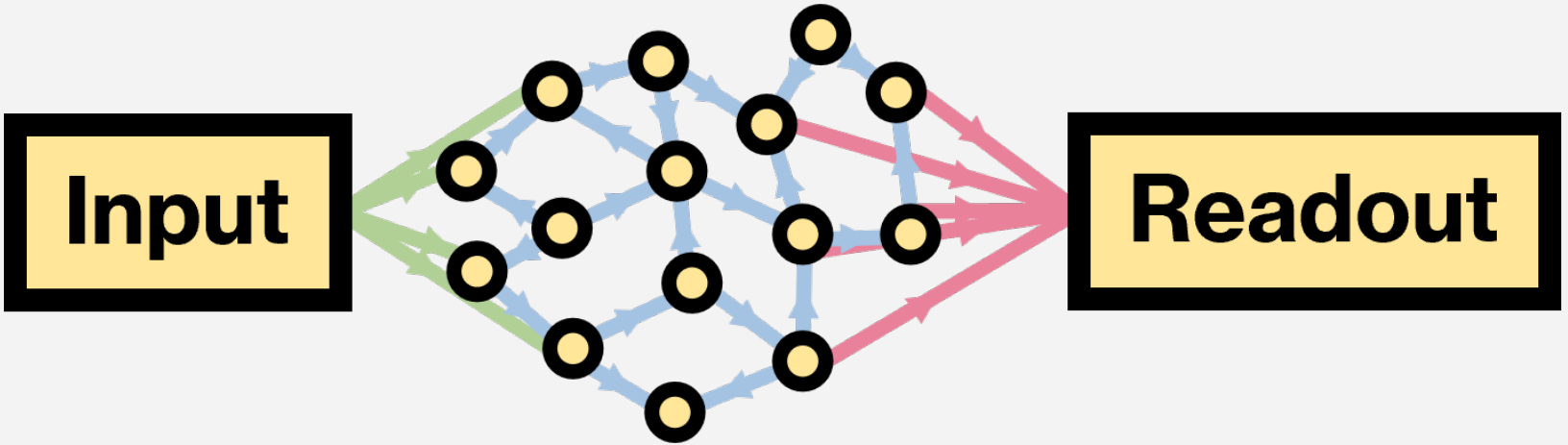
We proposed two hypotheses about learning in neurons after observing astrocytes' rhythmic activity:

- Learning involves rhythmic variations in link strength, and,
- Learning occurs via coordination of the phases of these rhythmic variations.

The algorithm implements these in a reservoir computing model

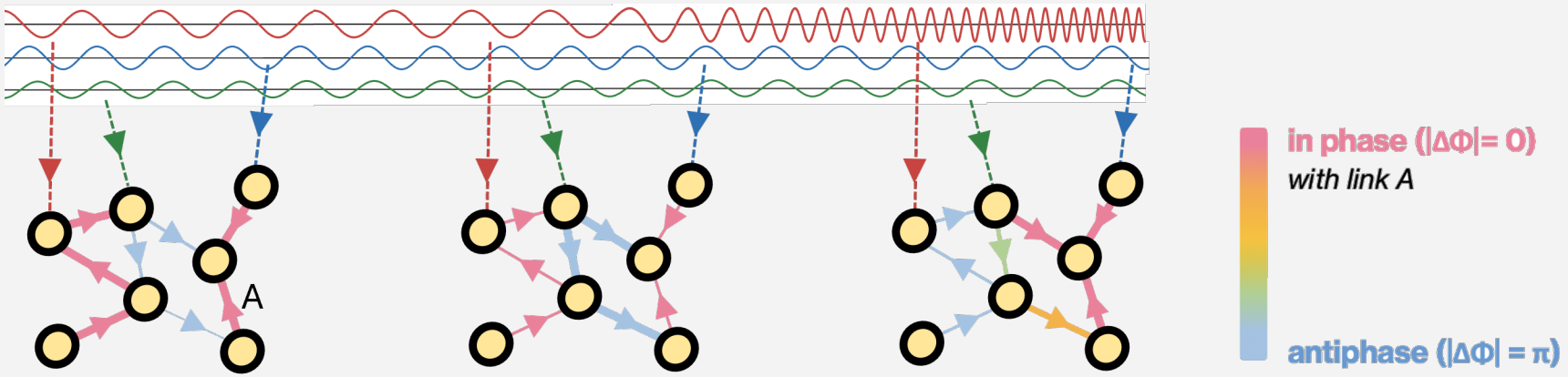
RS Algorithm – Reservoir Computing

The base of rhythmic sharing is an echo-state network with sparse connectivity



RS Algorithm – Oscillations

Initially, the strength of the links oscillate with random phases. A Kuramoto-inspired model controls synchronization of subgroups of phases based on the input [6]



RS Algorithm – Oscillations

This is the Kuramoto model function that controls the phases of the links.

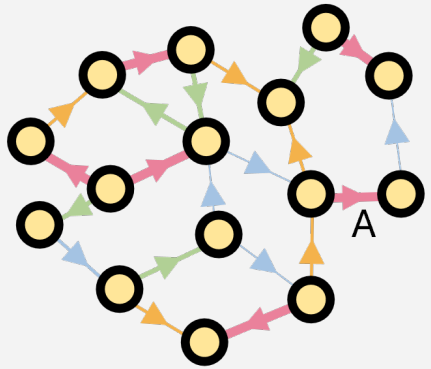
$$\frac{d\Phi}{dt} = \omega_0 + \left(\epsilon_1 + \epsilon_2 \hat{Q}^T n^*\right) \circ sin\left(\Psi - \Phi + \gamma\right) \quad (1)$$

Here, Φ denotes the phase matrix, ω_0 is the natural frequency of the nodes, Ψ is the local mean field, $\hat{Q}$ is the incidence matrix, ϵ_1 and ϵ_2 are coupling hyperparameters, and $\circ$ denotes element-wise multiplication.

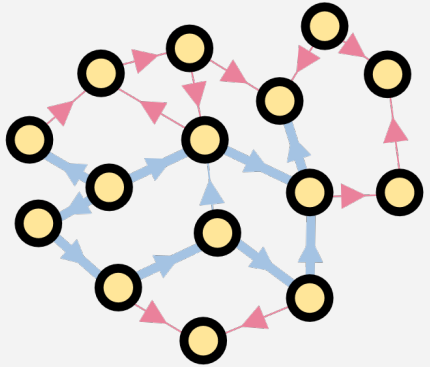
RS Algorithm – Synchrony

As proposed, the model learns as groups of links become synchronized.

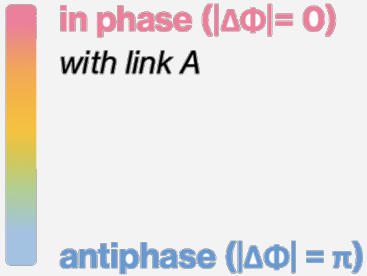
Synchrony provide a path for the information to flow through the model.
We measure synchrony with the Kuramoto order parameter R.



Initial, random phases, R=0.20

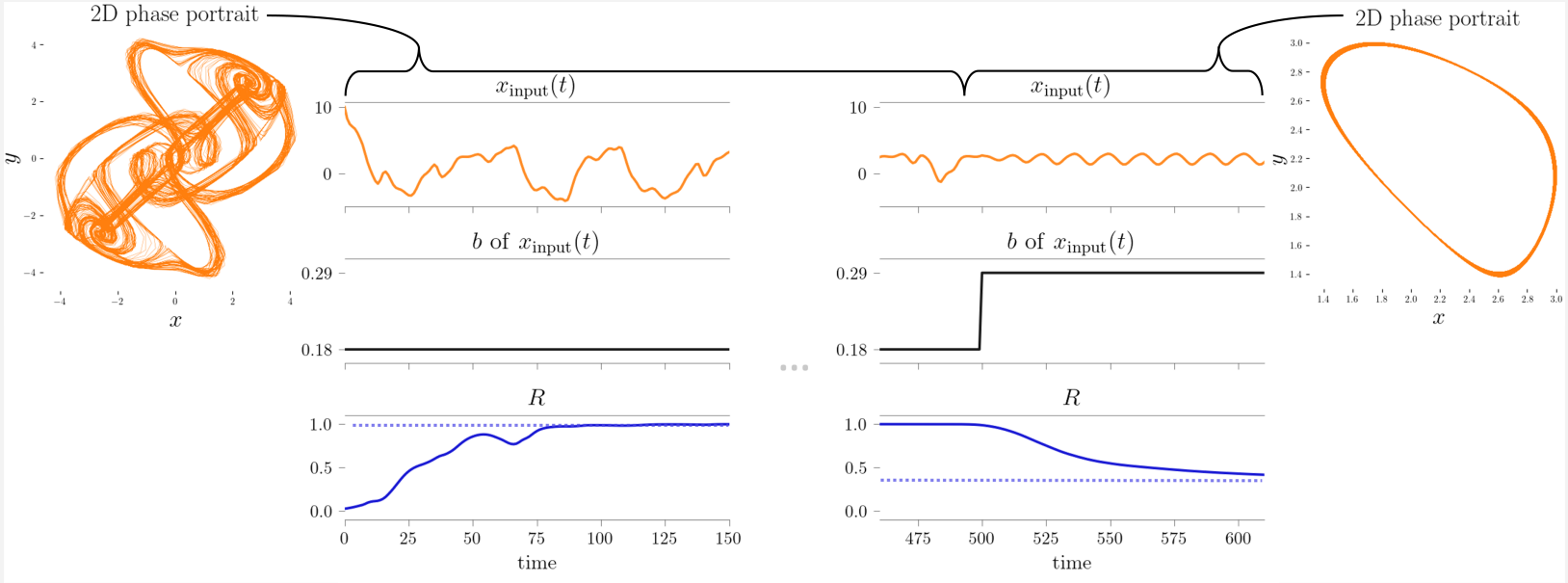


Learned phases, R=0.75



RS Algorithm – Reaction to Changing Dynamics

The synchrony reacts to different dynamics because the model adjusts which nodes are synchronized to match the new input.



RS Algorithm – Per-Input Synchrony

Initially, we believed that this changing synchrony was enough to detect concept drift. However, it only captured large dynamical shifts, not subtle changes.

RS Algorithm – Per-Input Synchrony

Initially, we believed that this changing synchrony was enough to detect concept drift. However, it only captured large dynamical shifts, not subtle changes.

Therefore, we introduced per-input synchrony, which only measures the synchrony of links connected to each inputs' nodes.

RS Algorithm – Per-Input Synchrony

Initially, we believed that this changing synchrony was enough to detect concept drift. However, it only captured large dynamical shifts, not subtle changes.

Therefore, we introduced per-input synchrony, which only measures the synchrony of links connected to each inputs' nodes.

We show that per-input synchrony generates rich features that amplify drifts, improving performance of detection algorithms.

Concept Drift Detection

Concept Drift

The concept drift detection task focuses on detecting when the distribution that an input is drawn from changes.

It is an important problem in machine learning since most models are brittle to it, and even minor drifts result in worse performance.

We utilize the ability of our model to highlight drifts to improve the performance of different algorithms on three datasets.

Concept Drift – Detectors

We test the performance of our model applied to generic concept drift detection algorithms, including:

- Autoencoder reconstruction models [7]
- Clustering algorithms
- Maximum mean discrepancy [8]

Additionally, we test SOTA, dataset-tuned detectors

- Neural System Identification and Bayesian Filtering (NSIBF) [9]
- Bidirectional Dynamic Model (BDM) [10]

Our Results

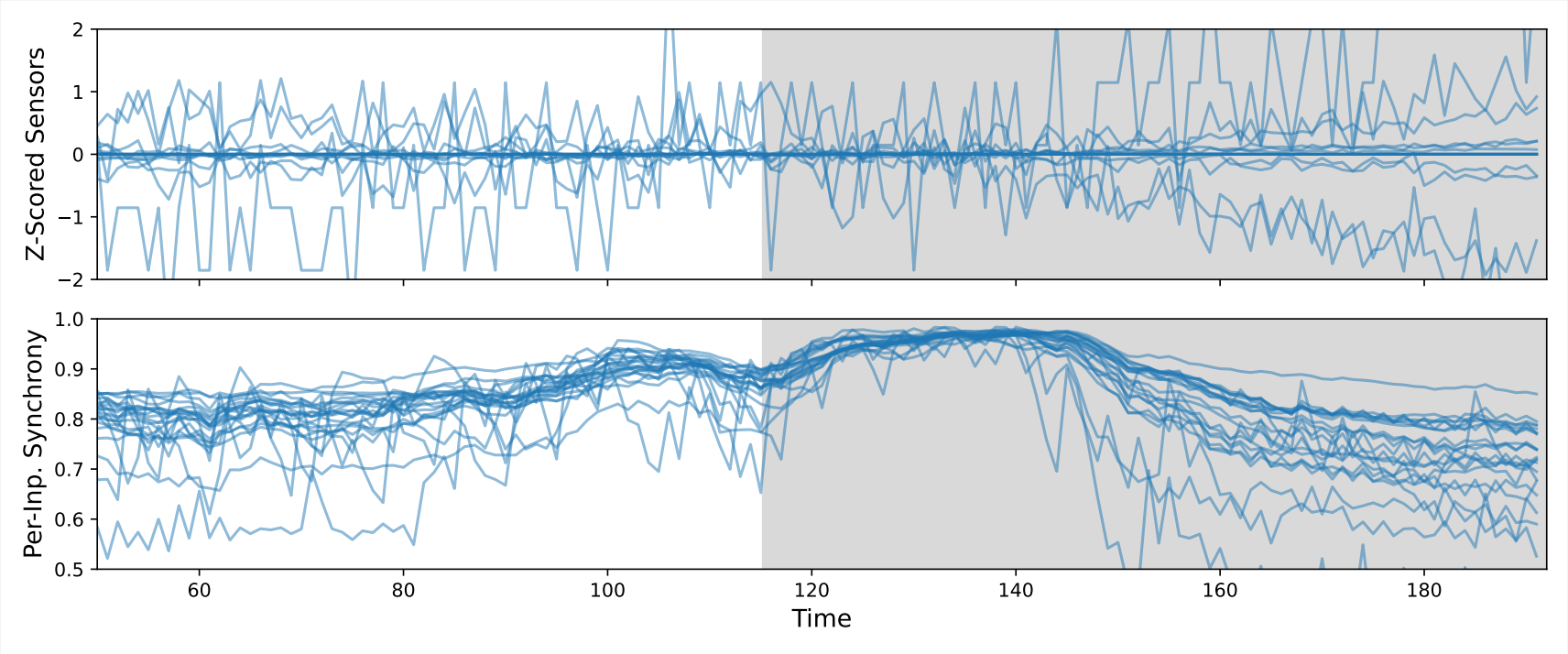
NASA C-MAPSS Dataset – Overview

The NASA C-MAPSS dataset includes recordings from sensors on a set of simulated turbofan engines as they catastrophically degrade.

Following prior work, we consider the last 40% of each recording anomalous [11, 12].

NASA C-MAPSS Dataset – Per-Input Synchrony

Per-input synchrony begins emphasizing the drifts before the 60% cutoff



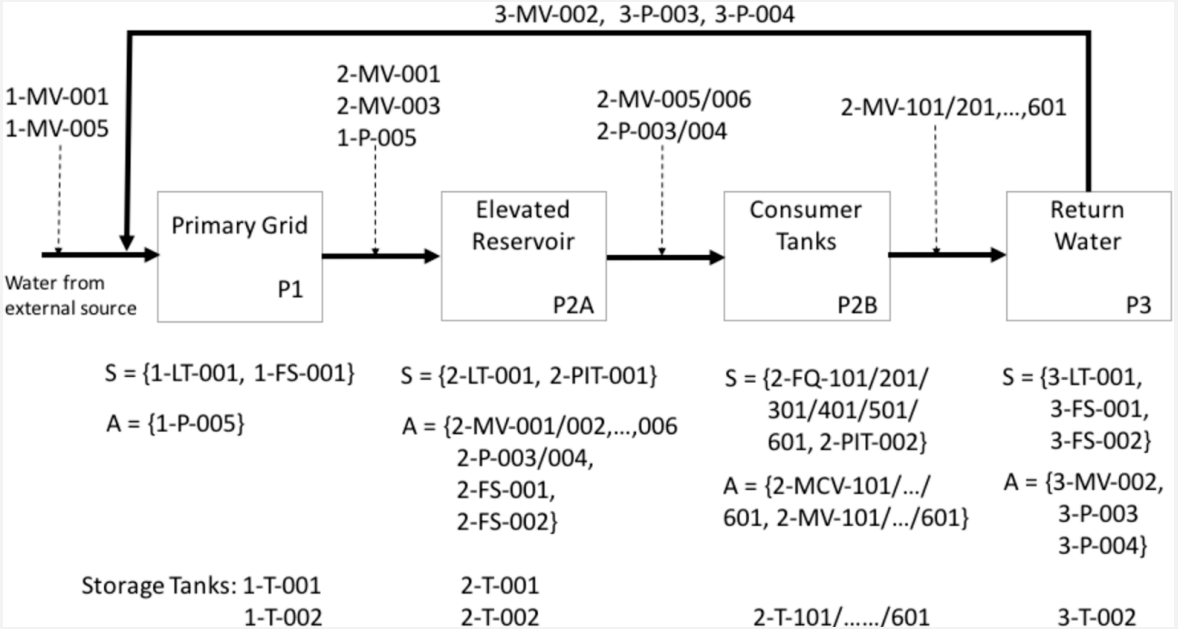
NASA C-MAPSS Dataset – Detector Performance

Per-input synchrony elevates the performance of the generic models.

Method	Prec.	Rec.	F1	Del.
AE	0.561	0.629	0.593	16.58
with RS	0.463	0.941	0.621	0.58
MMD	0.441	0.991	0.610	**0.00**
with RS	0.657	**0.822**	0.730	19.50
Clustering	0.413	1.000	0.585	**0.00**
with RS	**0.860**	0.804	**0.831**	0.58

WaDI and SWAT Datasets – Overview

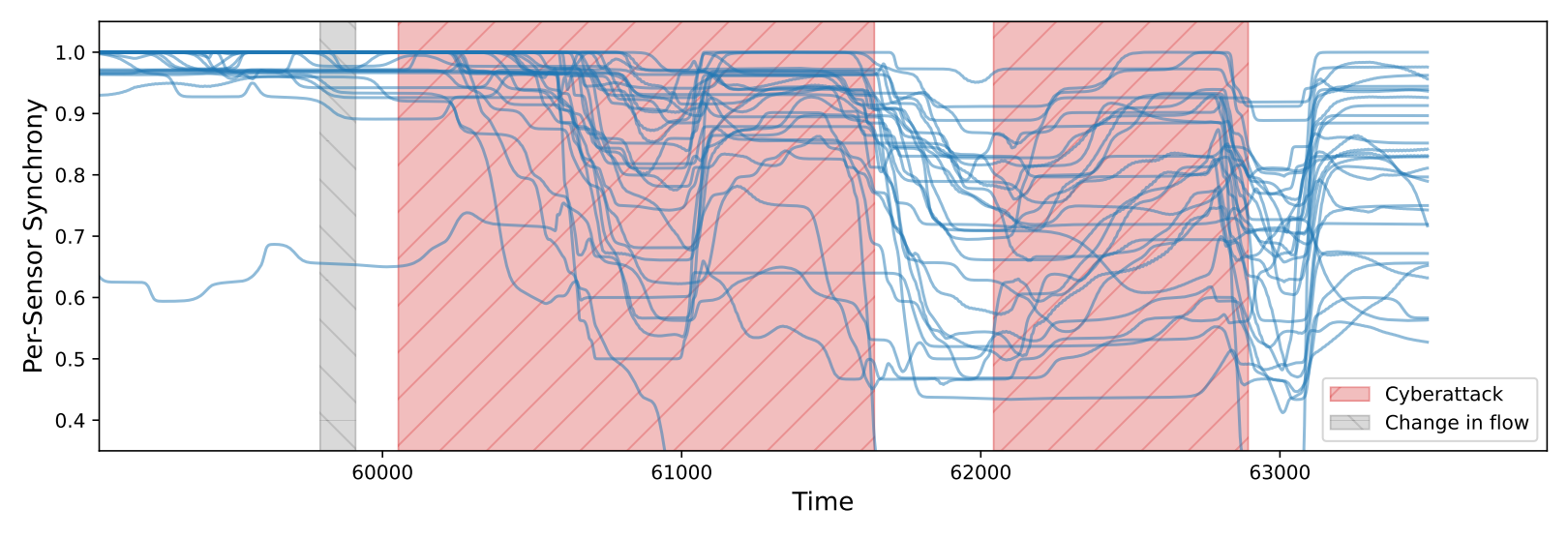
The highlights of our results are on the Secure Water Treatment testbed (SWaT) dataset and water distribution testbed (WADI) dataset [13, 14]



These are complicated datasets of real sensors and actuators treating water

WaDI and SWAT Datasets – Per-Input Synchrony

Per-input synchrony reacts to both real cyberattacks and normal drifts, showing its usefulness but potentially posing a problem for detectors



WaDI and SWAT Datasets – Simple Detectors

Our methodology improves performance on the simple, generic detectors

Method	SWaT Dataset			WADI Dataset		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
AE	0.028	0.492	0.052	0.202	0.550	0.252
with RS	0.038	**1.000**	0.073	0.187	0.673	0.284
MMD	0.045	0.747	0.084	**0.225**	0.012	0.024
with RS	0.142	0.172	0.156	0.215	0.611	0.318
Clustering	0.039	0.998	0.075	0.154	1.000	0.268
with RS	**0.139**	0.840	**0.238**	0.192	**0.962**	**0.321**

WaDI and SWAT Datasets – SOTA Detectors

Finally, our method achieves SOTA results with the NSIBF model [9]

Method	SWaT Dataset			WADI Dataset		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
DAGMM [15]	0.957	0.643	0.769	**0.904**	0.131	0.228
USAD [16]	**0.995**	0.629	0.771	0.243	0.462	0.319
BDM [10]	0.991	0.685	0.811	0.276	0.593	0.377
with RS	0.972	0.631	0.765	0.13	0.557	0.21
NSIBF [9]	0.892	0.712	0.792	0.234	0.496	0.318
with RS	0.943	**0.810**	**0.871**	0.574	**0.876**	**0.694**

Per-input synchrony is a low-dimensional dynamical representation that matches the assumption of neural state-space models like NSIBF but not the reconstruction-based BDM

Conclusions

Contributions

We previously proposed an algorithm that, from observations in in-vitro systems, tested two hypotheses of neural learning.

Contributions

We previously proposed an algorithm that, from observations in in-vitro systems, tested two hypotheses of neural learning.

Here, we show that the algorithm is sensitive to drift in an input, and we introduce a new method, per-input synchrony, that emphasizes these drifts.

Contributions

We previously proposed an algorithm that, from observations in in-vitro systems, tested two hypotheses of neural learning.

Here, we show that the algorithm is sensitive to drift in an input, and we introduce a new method, per-input synchrony, that emphasizes these drifts.

We show that using per-input synchrony as features for detection algorithms improves their performance, leading to SOTA results on the SWaT and WADI datasets.

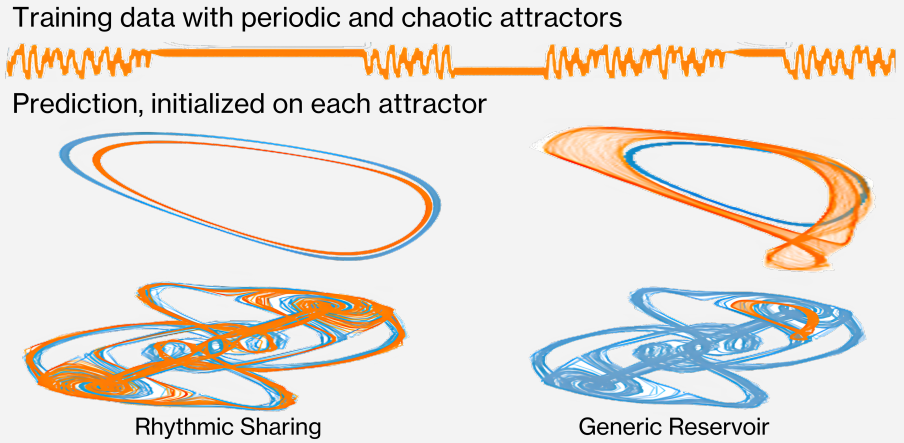
Future Work

We plan to continue working on understanding the rhythmic sharing algorithm. Specifically, we want to focus on:

- How hyperparameter selection changes performance.
- How we can perform drift detection in more complex environments.
- How rhythmic sharing changes our understanding of neuronal cultures or reservoir systems.
- How rhythmic sharing works with multidynamic systems

Future Work – Multidynamic Systems

Previously, we saw that a rhythmic sharing reservoir trained on two different attractors can reproduce both, while a generic reservoir reproduces a combination of the two.



We want to apply this, with our drift detection ability, to real-world datasets

Acknowledgements and Questions

Thank you to Hoony Kang, Wolfgang Losert, Noah Chongsiriwatana, Kate O'Neill, and the AFOSR Biophysics Program for their support.

Please reach out to learn more:

Ian Whitehouse: `ianjw@umd.edu`
Wolfgang Losert: `wlosert@umd.edu`

References – I

A. Araque, V. Parpura, R. P. Sanzgiri, and P. G. Haydon, "Tripartite synapses: glia, the unacknowledged partner," *Trends in Neurosciences*, vol. 22, p. 208–215, May 1999.

N. A. Oberheim, T. Takano, X. Han, W. He, J. H. C. Lin, F. Wang, Q. Xu, J. D. Wyatt, W. Pilcher, J. G. Ojemann, B. R. Ransom, S. A. Goldman, and M. Nedergaard, "Uniquely hominid features of adult human astrocytes," *The Journal of Neuroscience*, vol. 29, p. 3276–3287, Mar. 2009.

G. Perea, M. Navarrete, and A. Araque, "Tripartite synapses: astrocytes process and control synaptic information," *Trends in Neurosciences*, vol. 32, no. 8, pp. 421–431, 2009.

K. M. O'Neill, E. Saracino, B. Barile, N. J. Mennona, M. G. Mola, S. Pathak, T. Posati, R. Zamboni, G. P. Nicchia, V. Benfenati, and W. Losert, "Decoding natural astrocyte rhythms: Dynamic actin waves result from environmental sensing by primary rodent astrocytes," *Adv. Biol. (Weinh.)*, vol. 7, p. e2200269, June 2023.

H. Kang and W. Losert, "Rhythmic sharing: A bio-inspired paradigm for zero-shot adaptive learning in neural networks," 2025.

Y. Kuramoto, *Chemical oscillations, waves, and turbulence*.
Springer series in synergetics, Berlin, Germany: Springer, 1984 ed., Sept. 1984.

References – II

E. Wolf and T. Windisch, “A method to benchmark high-dimensional process drift detection,” *J. Intell. Manuf.*, Apr. 2025.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012.

C. Feng and P. Tian, “Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, (New York, NY, USA), p. 2858–2867, Association for Computing Machinery, 2021.

F. Wang, K. Wang, and B. Yao, “Time series anomaly detection with reconstruction-based state-space models,” in *Artificial Neural Networks and Machine Learning – ICANN 2023* (L. Iliadis, A. Papaleonidas, P. Angelov, and C. Jayne, eds.), (Cham), pp. 74–86, Springer Nature Switzerland, 2023.

A. A. Bataineh, A. Mairaj, and D. Kaur, “Autoencoder based semi-supervised anomaly detection in turbofan engines,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 11, 2020.

References – III

G. Zhu, L. Huang, D. Li, and L. Gong, "Anomaly detection for multivariate times series data of aero-engine based on deep lstm autoencoder," in *2024 6th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT)*, pp. 190–194, 2024.

J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security* (G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, eds.), (Cham), pp. 88–99, Springer International Publishing, 2017.

C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: a water distribution testbed for research in the design of secure cyber physical systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, CySWATER '17, (New York, NY, USA), p. 25–28, Association for Computing Machinery, 2017.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, (New York, NY, USA), p. 3395–3404, Association for Computing Machinery, 2020.

Appendix – Equations

Node Update:

$$n(t + \Delta t) = \alpha n(t) + (1 - \alpha)\, tanh(A^{\sim} n(t) + W_{in} u(t))$$

Modulated adjacency matrix, $A^{\sim} \in \mathbb{R}^{N \times N}$:

$$A^{\sim}(t) = A \circ \left(1 - \frac{m}{2}(1 + sin[\Phi(t)])\right)$$

Appendix – Equations

Link phase update:

$$\Phi(t + \Delta t) = \Phi(t) + \Delta t * \frac{d\Phi}{dt}$$

Synchrony:

$$R(t)e^{i\langle\Phi\rangle(t)} = \frac{1}{N_l}\sum_{k=1}^{N_l} e^{i\Phi_k(t)}$$

Per-input synchrony:

$$R_c(t)e^{i\langle\Phi\rangle_c(t)} = \frac{1}{|L_c|}\sum_{k\in L(c)} e^{i\Phi_k(t)}$$


McKELDIN LIBRARY
A R M O