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

Ian Whitehouse[1], Hoony Kang[2], Wolfgang Losert[2*]

[1]Department of Computer Science, University of Maryland, College Park, 20742, Maryland, United States of America.
[2]Department of Physics, University of Maryland, College Park, 20742, Maryland, United States of America.

*Corresponding author(s). E-mail(s): wlosert@umd.edu;
Contributing authors: ianjw@umd.edu; kang.hoony.2@gmail.com;

Abstract

Biological rhythms coordinate adaptive sensing and computation, spontaneously demonstrating concept drift detection abilities. We demonstrate that our oscillatory learning scheme, called rhythmic sharing, can autonomously detect concept drift. Inspired by astrocytic oscillations, the algorithm has recurrent links that vary sinusoidally, producing emergent sensitivity to distributional drift. We introduce a new measure, called per-input synchrony, which harnesses this sensitivity to enable early and precise detection of hidden or complex drifts. Across three datasets, NASA C-MAPSS, SWaT, and WADI, the output of our per-input synchrony features improves detector performance, **culminating in new state-of-the-art F1-scores on the complex SWaT and WADI datasets**. These industrial datasets highlight the ability of our model to detect drift in highly-complex, real-world systems. Additionally, these results suggest that oscillatory link dynamics may serve as a general computational principle for adaptive sensing, with implications for neuromorphic hardware and astrocytic network biology.

Keywords: Concept drift, Dynamic systems, Bioinspired algorithms, Neuromorphic computing

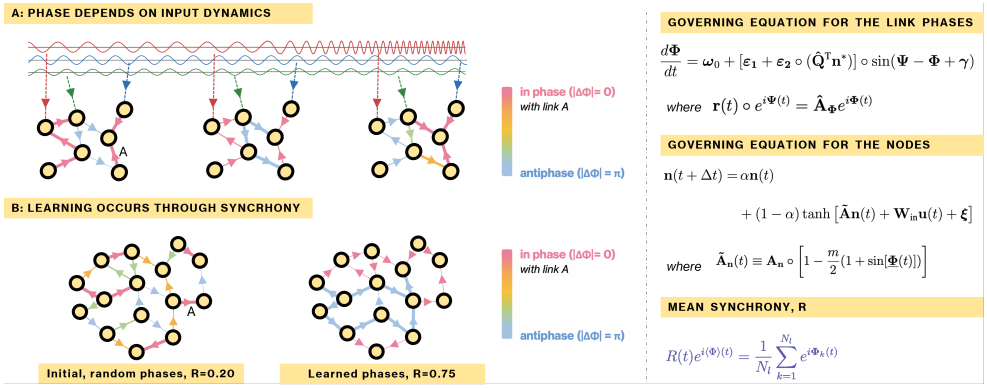

Fig. 1: Overview of the rhythmic sharing learning scheme. **A**: Instead of learning fixed weights, the model learns phases from the input dynamics. In this example, the model has learned two subnetworks, until the dynamics change, forcing it to change its learned phases. **B**: Synchrony is central to our learning scheme. The system starts with random phases and a low R value, but quickly learns new phases from the data, increasing the synchrony value R and developing logical subnetworks. The equations that govern this behavior are shown on the right.

Introduction

From early neuron models to modern neuromorphic approaches, biological principles have long informed unconventional computing, leading to neuromorphic hardware and algorithms that accurately model neuronal activity [1–5]. Often, wetlab experiments inspire novel neuromorphic computing schemes. An example of this is O'Neill et al., who proposed that astrocytes enable coordination between neurons through rhythmic activity [6]. This directly led to the rhythmic sharing algorithm, presented in Kang and Losert, which defines a new learning paradigm based on evolving oscillations that strengthen and weaken links within a reservoir computer [7]. In this extension of their work, we show that their algorithm has emergent, biologically-realistic, and useful concept drift detection capabilities, and we introduce a new metric, per-input synchrony, which capitalizes on those abilities, providing improved features to drift detection algorithms.

The discovery that the neuromorphic rhythmic sharing algorithm, where network topology is governed by changing oscillations, can recognize and adapt to concept drift corresponds with research on living neurons, which have the ability to drastically adjust their topology in response to drift in their environment [8]. Concept drift refers to spontaneous changes in the distribution of streaming data. It harms the performance of machine learning models, most of which lack the ability to detect or adapt to the drift [9]. Furthermore, state-of-the-art (SOTA) detection methods, like the maximum mean discrepancy or the Bidirectional Dynamic Model, rely on statistical tests or reconstruction accuracy, which are not effective for minute, hidden drifts, while our per-input synchrony identifies even tiny changes in each channel's dynamics [10, 11].

Therefore, our biologically plausible algorithm is able to extract these tiny changes, providing features to other detectors that improve their performance.

In this work, we exhibit and discuss these emergent abilities, answering the question "Can oscillation-based learning algorithms use their emergent sensitivity to subtle distributional changes to improve SOTA methods' performance on complex datasets?"

This approach exemplifies how biologically-inspired unconventional computing can provide alternatives to conventional weight-based learning. Rather than converging to static parameters, the oscillatory rhythmic sharing model mirrors input dynamics and naturally reconfigures through phase synchronization, as shown in Figure 1. The spontaneous emergence of drift detection from these dynamics highlights a hallmark of neuromorphic computing, where complex behaviors can arise without explicit design, as seen in prior works such as Miconi and Yang et al. [3, 4]. These findings also suggest that astrocytic rhythms may contribute to drift detection in living neurons, providing inspiration for potential future experiments.

Results

Overview

To demonstrate how the algorithm's emergent detection of concept drift amplifies small drifts within an input, the methodology was applied to three datasets: the NASA C-MAPSS dataset and the SWaT and WADI datasets [12–14]. These are standard datasets that are dynamic, multisensor/multivariate, and noisy simulations of real-world industrial processes. Prior work has employed these datasets for drift-detection methodologies, and there are existing SOTA measures from other works.

The NASA C-MAPSS dataset is a simulated set of turbofan engines, with sensors recording the engines' degradation at the end of their lifetime [12]. Following prior work, we consider the last 40% of each recording anomalous [15, 16]. Measurements are taken from the same sensors in the `FD001` split as those baseline results for comparability.

The highlight of our results are on the Secure Water Treatment testbed (SWaT) dataset and water distribution testbed (WADI) dataset [13, 14]. In these, our algorithm monitors sensors, setpoints, actuators, and other datapoints within complicated, physical simulators of a water utility company. Our success on these datasets highlight the utility of our algorithm's emergent abilities in a complex industrial environment. We use the same splits used by Wang et al. and Feng and Tian [11, 17]. In Table 2, the AE, MMD, Clustering and RS results were calculated by us using Feng and Tian's exact protocol, while the SOTA NSIBF and BDM results were taken from their manuscripts.

NASA C-MAPSS Dataset

Our tests on the NASA C-MAPSS dataset highlight the effectiveness of our approach and the features generated by our algorithm. In Table 1, we show that adding rhythmic sharing preprocessing improves the performance of generic models, elevating them to

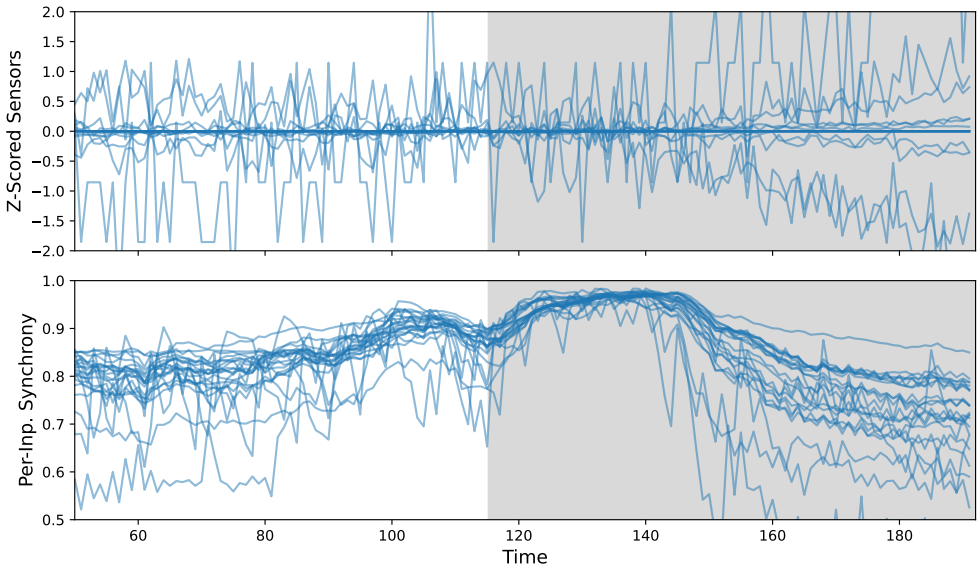

Fig. 2: Imperceptible drifts in sensor readings (top) are amplified by our work (bottom). The amplified drifts, which began occurring before the shaded section (representing the last 40% of samples), are noticed with per-input synchrony values increasing and moving closer together. Then, when the sensor readings begin moving erratically, the per-input synchronies begin to decrease. This plot suggests that our model recognizes the changing dynamics before the 60%/40% cutoff suggested by the literature, which is confirmed by how our model reduces delay in Table 1 [15].

closer to the performance of SOTA detectors built for the dataset, like Bataineh et al. [15].

However, as the visualization in Figure 2 shows, and as the delay column in Table 1 confirms, our algorithm begins detecting the engine's failure before the 60%/40% cutoff. The table shows detection delay calculated as the average number of timesteps before the first correct detection. A strict 60%/40% cutoff harms our method's performance, does not match the variety of failures in the dataset, and is a limitation of the authors' approach. The authors of the two algorithms did not provide source code, limiting our ability to test our methodology with their algorithms.

SWaT and WADI Datasets

The SWaT and WADI datasets are more compelling tests of the methodology, and the methodology significantly improves the performance of Feng and Tian's NSIBF algorithm on both datasets, beating the previously-better BDM algorithm [11, 17]. The new SOTA scores achieved by our approach are shown in Table 2. This jump in performance is in line with qualitative observations, like Figure 3, which shows that the model has clear reactions to two cyberattacks. This figure also shows that the model

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	0.511	**1.000**	0.676	**0.00**
with RS	0.860	0.804	0.831	0.58
Bataineh et al. [15]	0.896	0.724	0.801	-
Somma et al. [18]	**0.960**	0.952	**0.956**	-

Table 1: Precision, Recall, F1-Score, and Delay on the NASA C-MAPSS Dataset. As shown, using the rhythmic sharing algorithm improves the detection capabilities of out-of-the-box detectors, improving their results and reducing their distance to SOTA models.

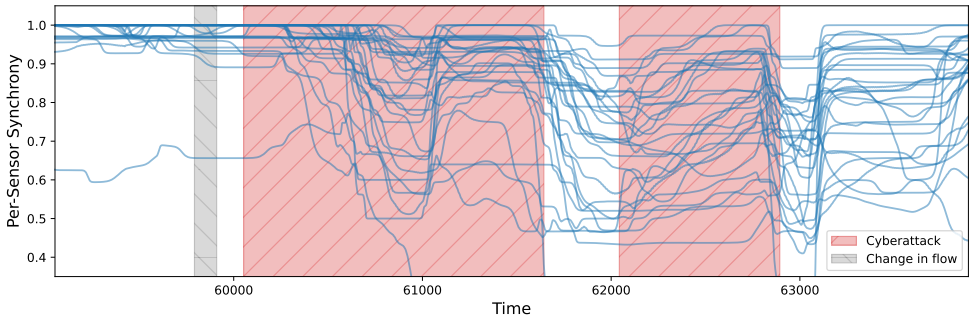

Fig. 3: Per-input synchrony over a large amount of time within the WADI dataset. As shown, the per-input synchrony has a large reaction to both the two cyberattacks (shown in red), and a smaller reaction to an ordinary change in requested flow (shown in grey). Per-input synchrony also shows that not every input experiences the concept drift, which potentially harms the detectors using the raw dataset.

reacts to an ordinary change in the requested flow of the facility, suggesting that the model understands the underlying dynamics of the system. In Section "Discussion", we provide intuition for why rhythmic sharing is able to beat SOTA baselines using the NSIBF algorithm but does not improve the performance of the BDM model.

Method	SWaT Dataset			WADI Dataset		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
AE	0.028	0.492	0.052	0.202	0.55	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
DAGMM [19]	0.957	0.643	0.769	**0.904**	0.131	0.228
USAD [20]	**0.995**	0.629	0.771	0.243	0.462	0.319
BDM [11]	0.991	0.685	0.811	0.276	0.593	0.377
with RS	0.972	0.631	0.765	0.130	0.557	0.210
NSIBF [17]	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**

Table 2: Precision, recall, and F1-score on the SWaT and WADI datasets. As shown, using the rhythmic sharing algorithm to preprocess the dataset dramatically improves the F1-Score of the baseline algorithms, doubling it in some cases. Additionally, it improves the NSIBF algorithm, achieving new SOTA benchmarks on SWaT and WADI. It does not equally improve the BDM algorithm, leading to the hypothesis that the state-space nature of the NSIBF algorithm is advantaged by the rhythmic sharing process. All methods were evaluated on identical train/test splits using the same metric implementation.

Discussion

We show that the rhythmic sharing algorithm spontaneously detects concept drift by encoding system dynamics through oscillatory link coordination. Monitoring per-input synchrony enhances sensitivity to subtle, channel-specific changes that detectors miss on raw data, mirroring astrocytic networks, where rhythmic activity enables rapid responses to environmental cues.

Effectiveness is demonstrated on three well-established datasets. On the NASA C-MAPSS dataset, the rhythmic sharing algorithm boosts the performance of generic drift detection schemes, raising their performance closer to the level of SOTA detectors built for the dataset, like Bataineh et al. [15]. Unfortunately, the source code for the two SOTA algorithms on the NASA C-MAPSS dataset were not provided, stopping us from running the same experiments that we ran for the SWaT and WADI datasets.

On the SWaT and WADI datasets, our algorithm significantly boosted the NSIBF algorithm by Feng and Tian, leading to it beating the previous SOTA by a large margin [17]. However, the same methodology did not improve the previous SOTA, which was the BDM algorithm by Wang et al. [11]. We hypothesize that rhythmic

sharing's per-input synchrony is a low-dimensional dynamical representation with a temporal evolution that closely matches the assumption of neural state-space models like NSIBF. The NSIBF algorithm combines LSTM-based system identification with Bayesian filtering, and is therefore particularly well suited to the per-input synchrony features. In contrast, BDM is reconstruction-based, and is likely able to smoothly reconstruct the simpler per-input synchrony, even in the presence of drifts.

These positive results suggest that oscillatory synchrony could act as a distributed anomaly-detection mechanism *in vivo*, providing hypotheses for neuroscience experiments and inspiration for neuromorphic design. Additionally, this emergent property of rhythmic sharing highlights the potential implications of biological inspiration and the power of applying findings from living neuronal networks to algorithm design. Limitations do exist, including that per-input synchrony has varying levels and types of reactions to drifts in different datasets. Additionally, the algorithm's success is heavily based on hyperparameter selection, presenting opportunities for refinement and greater robustness.

These positive results leave additional questions that we will answer in future works. A key one is whether, after detecting the concept drift, the model can overcome it and learn despite it. In Kang and Losert, we showed that rhythmic sharing could recover previous dynamics from the model's memory by freezing the links' synchrony, even after a drift has occurred [7]. This is incredibly powerful, showing both that the rhythmic sharing reservoir model can learn multiple sets of dynamics and that it can continue to predict those past dynamics even if its input is affected by a drift. More work is necessary to understand the mechanism of action and how that phenomenon can be applied in real-world datasets, however, we hope this enables lifelong learning for reservoirs in systems with concept drift.

We plan to address this while also improving the robustness of per-input synchrony as a drift detection methodology. Here, we are most interested in techniques that can distinguish abnormal from benign distributional shifts. This is a challenge in datasets like WADI, where the utility's requested flow rate drifts with similar effects to real cyberattacks, as shown in Figure 3.

Finally, the success of an astrocyte-inspired approach to concept drift detection shows the power of neuromorphic algorithms. As a scientific consensus is found on the role of astrocytes in cognition, they can continue to serve as inspiration, leading to improvements in this and other neuromorphic algorithms. Furthermore, observations and experiments on neuromorphic algorithms can hopefully provide testable hypotheses for biologists to design future experiments.

Methods

This work presents an emergent feature of Kang and Losert's rhythmic sharing algorithm, and shows how it can be harnessed for concept drift detection [7]. The rhythmic sharing algorithm was a direct result of work on living neuronal cultures, where O'Neill et al. discovered that astrocyte glial cells respond to chemical signaling through rhythmic oscillation [6]. In the original publication, Kang and Losert observed that the rhythmic sharing algorithm appears to react to changes in an input distribution. In

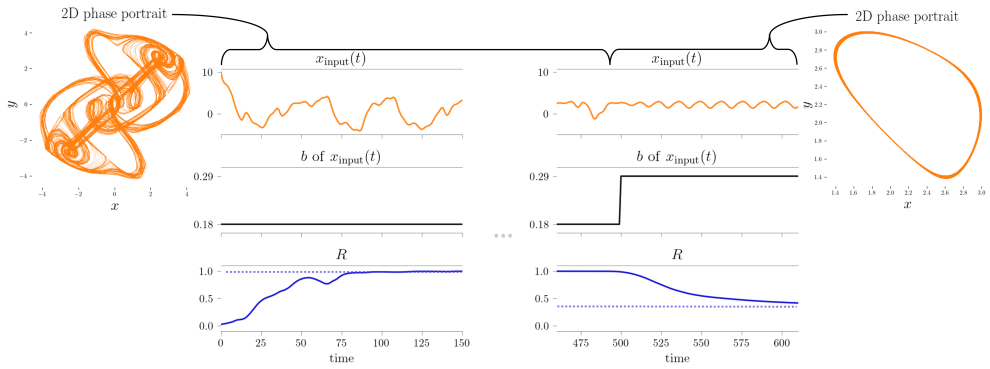

Fig. 4: Diagram showing how the model's global synchrony, R, responds quickly to changes in the input's dynamics, as discovered by Kang and Losert [7]. As shown, the b parameter of the Thomas System changes, leading to a change in the R value [21]. As discussed in Figure 1, this occurs as the model adjusts to the new dynamics. We leverage responses like this to detect concept drifts in our methodology.

this work, we validate that observation, introducing the novel measure **per-input synchrony**, a metric that reacts to subtle distributional shifts, and run head-to-head tests of the methodology against state-of-the-art (SOTA) concept drift detectors. We show examples of this methodology detecting drift on two toy problems and industry-standard concept drift datasets.

Rhythmic Sharing Algorithm

Oscillations play a large role in in-vitro neuronal networks, affecting everything from actin function to the networks' response to electrical fields [22–26]. Research by O'Neill et al. has shown that astrocytes, a type of glial cell, use rhythmic actin waves to respond to environmental cues [6]. Astrocytes are brain cells that are not electrically active, but surround multiple neuronal synapses, and may coordinate the information flow at these synapses, a concept known as the 'Tripartite Synapse.'

After observing oscillatory mechano-chemical activity in astrocytes within networks of neurons, we proposed two hypotheses in prior work:

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

These hypotheses were implemented in a Kuramoto-like model, with rhythmic links inside an Echo-State Network, proposed by Jaeger and Haas [27, 28]. Instead of fixed weights, the nodes within the ESN are connected by links with rhythmically-varying strength, with phases governed by Equation 1:

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

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. Figure 1 shows how this works within the rhythmic sharing model. The algorithm, and how global and input-specific synchrony demonstrate emergent drift detection abilities, are discussed fully in Supplemental Information Section S1.

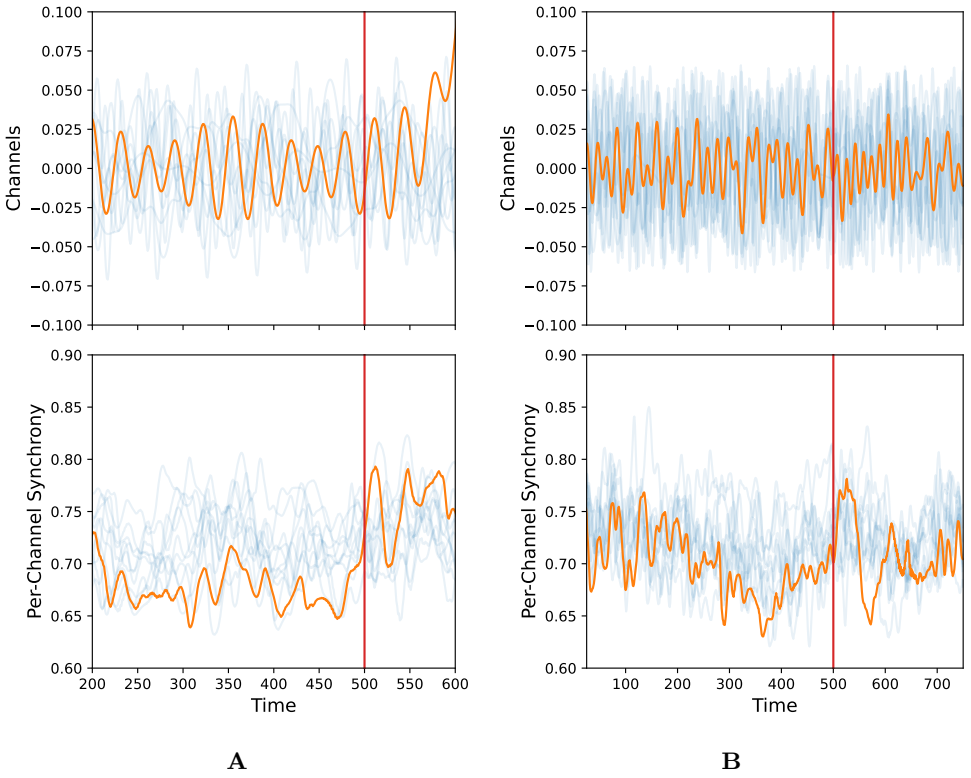

Fig. 5: Toy problems showing how per-input synchrony captures minute concept drifts. **A**: Reactions of the per-input synchronies to the concept drift in the first toy example. Here, the model immediately reacts with a jump when it notices a small upward drift, highlighting the algorithm's sensitivity to single channel drifts. **B**: Reactions of the per-input synchronies to the concept drift in the second toy example. The drift comes from one channel slowly moving from being the mean of channels $1, 2, 3$ to being the mean of $2, 5, 6$, which does not change its statistics.

Global and Input-Specific Synchrony

In this model, which we call 'rhythmic sharing', one immediately useful measure emerges: global synchrony from the Kuramoto order parameter R. It is a scalar measuring phase coordination across all of the links in the network. A change in the dynamics of the input system leads to a change in the R of the network, shown in Figure 4. However, this effect is absent or delayed in cases where only a subset of inputs drift or the drift is small, which is common in real-world datasets. To address this, we introduce **per-input synchrony**, a measure of how the links connected to each of the input nodes are synchronized. Biologically, this reflects how astrocytes that control a single input's connection to the rest of the dynamical system react to changes in that input's dynamics, and it acts like a microscope to reveal local drift relative to the entire system.

Per-input synchrony provides a vector of estimates of phase coordination. Per-input synchrony at time t is a length-C vector for C channels. Over time, it forms a $C \times T$ series, the same shape as the input datastream. Therefore, SOTA drift-detection methods, discussed in Section "Baseline Algorithms", can be applied to detect drifts within this vector. In two toy examples, we highlight how per-input synchrony is affected by two small drifts. These are shown in Figure 5. While per-input synchrony responds quickly to the changes, statistical detectors would struggle to identify them as they do not initially alter the statistical properties of the input.

Baseline Algorithms

Our algorithm, rhythmic sharing, provides per-input synchrony values for each sensor, which, as shown in Figure 2, amplify hidden or complex drifts. These work as features for other drift detectors, improving their performance relative to their performance on the raw data. To test the effectiveness of this approach, we employ both generic anomaly detectors and two SOTA algorithms purpose-made for the SWaT and WADI datasets.

We employ two generic, multi-channel drift detection algorithms implemented by Wolf and Windisch, based on clustering of the samples and the reconstruction error of an autoencoder [29]. Additionally, we consider the maximum mean discrepancy (MMD) algorithm, proposed by Gretton et al., which is a statistical test to determine if two sets of samples came from different distributions [10].

For the SWaT and WADI datasets, we also consider two recent SOTA algorithms, proposed by Feng and Tian and Wang et al. [11, 17]. The first, Neural System Identification and Bayesian Filtering (NSIBF) by Feng and Tian is a state-space model combined with a Bayesian filter, specifically designed for cyber-physical systems like SWaT and WADI. The second, called the Bidirectional Dynamic Model (BDM) by Wang et al., is an encoder-decoder model that uses LSTMs and measures reconstruction accuracy.

Declarations

Data Availability. The toy problems (Section "Rhythmic Sharing Algorithm") are generated in the code, available online at https://github.com/losertlab/rhythmic_sharing/tree/npj-concept-drift. The NASA C-MAPSS dataset is available from https://software.nasa.gov/software/LEW-18315-2. The SWaT and WADI datasets are available upon request from https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/, and the splits used are found at https://github.com/cfeng783/NSIBF/tree/main. The authors credit "iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design" for the SWaT and WADI datasets.

Code Availability. The code is available at https://github.com/losertlab/rhythmic_sharing/tree/npj-concept-drift.

Acknowledgments. This study was funded in part by the Air Force Office of Scientific Research Biophysics Program [Grant *FA9550-22-1-0405*]. The funder played no role in study design, data collection, analysis and interpretation of data, or the writing of this manuscript.

Author Contributions. HK developed the initial rhythmic sharing algorithm for the initial publication. IW added the per-input synchrony feature and conducted the experiments in this manuscript. WL supervised the algorithmic development and provided insight from wetlab experiments. All authors reviewed the manuscript.

Competing Interests. We declare pending patent application 19/379, 206, covering the algorithms and applications. The inventors are HK and WL and the applicant is the University of Maryland. The authors declare no other competing financial or non-financial interests.

References

[1] Mead, C.: Neuromorphic electronic systems. Proceedings of the IEEE **78**(10), 1629–1636 (1990) https://doi.org/10.1109/5.58356

[2] Schuman, C.D., Kulkarni, S.R., Parsa, M., Mitchell, J.P., Date, P., Kay, B.: Opportunities for neuromorphic computing algorithms and applications. Nature Computational Science **2**(1), 10–19 (2022) https://doi.org/10.1038/s43588-021-00184-y

[3] Miconi, T.: Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. eLife **6**, 20899 (2017) https://doi.org/10.7554/eLife.20899

[4] Yang, R., Ping, H., Xiao, X., Kiani, R., Bogdan, P.: Spiking dynamics of individual neurons reflect changes in the structure and function of neuronal networks. Nature Communications **16**(1), 6994 (2025) https://doi.org/10.1038/s41467-025-62202-1

[5] Zhang, Y., Mo, L., He, X., Meng, X.: Unsupervised spiking neural network based on liquid state machine and self-organizing map. Neurocomputing **620**, 129120 (2025) https://doi.org/10.1016/j.neucom.2024.129120

[6] O'Neill, K.M., Saracino, E., Barile, B., Mennona, N.J., Mola, M.G., Pathak, S., Posati, T., Zamboni, R., Nicchia, G.P., Benfenati, V., Losert, W.: Decoding natural astrocyte rhythms: Dynamic actin waves result from environmental sensing by primary rodent astrocytes. Adv. Biol. (Weinh.) **7**(6), 2200269 (2023)

[7] Kang, H., Losert, W.: Rhythmic sharing: A bio-inspired paradigm for zero-shot adaptive learning in neural networks (2025). https://arxiv.org/abs/2502.08644

[8] Li, J., Bauer, R., Rentzeperis, I., Leeuwen, C.: Adaptive rewiring: a general principle for neural network development. Front. Netw. Physiol. **4**, 1410092 (2024)

[9] Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning **23**(1), 69–101 (1996) https://doi.org/10.1007/BF00116900

[10] Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research **13**(25), 723–773 (2012)

[11] Wang, F., Wang, K., Yao, B.: Time series anomaly detection with reconstruction-based state-space models. In: Iliadis, L., Papaleonidas, A., Angelov, P., Jayne, C. (eds.) Artificial Neural Networks and Machine Learning – ICANN 2023, pp. 74–86. Springer, Cham (2023)

[12] Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management, pp. 1–9 (2008). https://doi.org/10.1109/PHM.2008.4711414

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

[14] Ahmed, C.M., Palleti, V.R., Mathur, A.P.: 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, pp. 25–28. Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3055366.3055375

[15] Bataineh, A.A., Mairaj, A., Kaur, D.: Autoencoder based semi-supervised anomaly detection in turbofan engines. International Journal of Advanced Computer Science and Applications **11**(11) (2020) https://doi.org/10.14569/IJACSA.

2020.0111105

[16] Zhu, G., Huang, L., Li, D., Gong, L.: 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). https://doi.org/10.1109/ECNCT63103.2024.10704361

[17] Feng, C., Tian, P.: 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, pp. 2858–2867. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467137

[18] Somma, M., Gallien, T., Stojanovic, B.: Anomaly Detection in Complex Dynamical Systems: A Systematic Framework Using Embedding Theory and Physics-Inspired Consistency (2025). https://arxiv.org/abs/2502.19307

[19] Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=BJJLHbb0-

[20] Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: USAD: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20, pp. 3395–3404. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394486.3403392

[21] Thomas, R.: Deterministic chaos seen in terms of feedback circuits: Analysis, synthesis, "labyrinth chaos". Int. J. Bifurcat. Chaos **09**(10), 1889–1905 (1999)

[22] Yang, Q., Miao, Y., Campanello, L.J., Hourwitz, M.J., Abubaker-Sharif, B., Bull, A.L., Devreotes, P.N., Fourkas, J.T., Losert, W.: Cortical waves mediate the cellular response to electric fields. eLife **11**, 73198 (2022) https://doi.org/10.7554/eLife.73198

[23] Yang, Q., Miao, Y., Banerjee, P., Hourwitz, M.J., Hu, M., Qing, Q., Iglesias, P.A., Fourkas, J.T., Losert, W., Devreotes, P.N.: Nanotopography modulates intracellular excitable systems through cytoskeleton actuation. Proc. Natl. Acad. Sci. U. S. A. **120**(19), 2218906120 (2023)

[24] Ucar, H., Watanabe, S., Noguchi, J., Morimoto, Y., Iino, Y., Yagishita, S., Takahashi, N., Kasai, H.: Mechanical actions of dendritic-spine enlargement on presynaptic exocytosis. Nature **600**(7890), 686–689 (2021)

[25] Wu, M., Wu, X., De Camilli, P.: Calcium oscillations-coupled conversion of actin

travelling waves to standing oscillations. Proc. Natl. Acad. Sci. U. S. A. **110**(4), 1339–1344 (2013)

[26] Bull, A.L., Campanello, L., Hourwitz, M.J., Yang, Q., Zhao, M., Fourkas, J.T., Losert, W.: Actin dynamics as a multiscale integrator of cellular guidance cues. Front. Cell Dev. Biol. **10**, 873567 (2022)

[27] Kuramoto, Y.: Chemical Oscillations, Waves, and Turbulence, 1984 edn. Springer series in synergetics. Springer, Berlin, Germany (1984)

[28] Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science **304**(5667), 78–80 (2004)

[29] Wolf, E., Windisch, T.: A method to benchmark high-dimensional process drift detection. J. Intell. Manuf. (2025)

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

Ian Whitehouse[1], Hoony Kang[2], Wolfgang Losert[2]

[1]Department of Computer Science, University of Maryland, College Park, 20742, Maryland, United States of America
[2]Department of Physics, University of Maryland, College Park, 20742, Maryland, United States of America

Rhythmic Sharing Algorithm and Drift-Relevant Feature Extraction

This supplement describes the rhythmic sharing algorithm proposed by Kang and Losert, this paper's contribution (per-input synchrony), and how the model was practically applied to the datasets [7].

Network Architecture and State Update

The rhythmic sharing algorithm is based on an echo-state network, which was proposed by Jaeger and Haas [28]. The reservoir is constructed with a sparse connection matrix, with the number of nodes N and the average degree of the nodes controlled by hyperparameters (shown in Table S1), and uses a linear, ridge-regression readout scheme. The links between the nodes have a base connection strength, set through uniform random sampling, scaled by the spectral radius, and stored as adjacency matrix $A \in \mathbb{R}^{N\times N}$.

For each input sample, a leak-rate update with tanh nonlinearity is performed:

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

where $W_{in} \in \mathbb{R}^{N\times C}$ is the input weight matrix (scaled by an input weight hyperparameter) and $u(t) \in \mathbb{R}^{C}$ is the input to the system. This is identical to a reservoir computer, with the exception of the modulated adjacency matrix, $A^{\sim} \in \mathbb{R}^{N\times N}$. This is calculated

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

where m is the link strength change ratio hyperparameter and $\Phi \in \mathbb{R}^{N\times N}$ are the links' phases.

Phase Dynamics and Synchrony Measures

The phases of the oscillators, Φ, are initially set with random uniform values, however, it changes as a function of time, the inputs, and the rhythmic sharing hyperparameters. Using a Kuramoto-like model, each oscillator is coupled to a subset of other oscillators in the network. Equation 1 shows how they evolve over time, and it is implemented in discrete time:

$$\Phi(t+\Delta t) = \Phi(t) + \Delta t * \frac{d\Phi}{dt} \tag{3}$$

The natural frequencies ω_0 are randomly and uniformly sampled when the reservoir is initiated. This Kuramoto-style relationship between the strengths of the nodes is the analog of the astrocyte cells, which control the rhythmic connections between neurons through the tripartite synapse.

Synchrony, R, is defined as the Kuramoto order parameter:

$$R(t)e^{i\langle\Phi\rangle(t)} = \frac{1}{N_l}\sum_{k=1}^{N_l} e^{i\Phi_k(t)} \tag{4}$$

In this work, we propose the per-input synchrony metric. First, we separate links associated with input channel c: L_c. Then, we calculate the per-input order parameter as

$$R_c(t)e^{i\langle\Phi\rangle_c(t)} = \frac{1}{|L_c|}\sum_{k\in L_c} e^{i\Phi_k(t)} \tag{5}$$

Downsampling can be performed to smooth the global or per-input synchrony, but is not necessary and tests on its effectiveness were inconclusive.

Hyperparameters, Reservoir Training, and Synchrony Monitoring

To maximize performance of the per-input synchrony methodology, hyperparameters were chosen in a two-step process. First, link strength change (m) was set to 0, making the reservoir a standard echo-state network reservoir, and the reservoir was repeatedly trained with different ESN parameters (Table S1, column A). Then, rhythmic sharing parameters (Table S1, column B) were chosen to maximize sensitivity. Hyperparameters were chosen on training samples. Since the training split of the SWaT and WADI datasets did not include cyberattacks, the hyperparameters were chosen to maximize sensitivity to changes in actuators and requested flow, respectively. The fact that hyperparameters tuned using actuator/flow changes (rather than cyberattack labels) still yielded strong detection performance highlights the robustness of the per-input synchrony features.

For each trial, input sample, or hyperparameter test, the rhythmic sharing reservoir was advanced in two stages. First, the nodes were updated using equation 1. Then, the phases were updated using equations 1 and 3. The inputs were normalized by demeaning and dividing by their standard deviations. For the test splits of the SWaT and WADI datasets, the train sets' mean and standard deviation were used. The testing was performed on two machines, with an `i7-6700k` and an `i7-5820k`, and the model was CPU-bottlenecked. Despite that, the algorithm processed samples with a sample rate of $\approx 40\ Hz$ on the SWaT and WADI datasets with 768 nodes, making it practical for industrial settings and significantly faster than the compared SOTA algorithms in Section "Baseline Algorithms".

As the model ran, the link phases and node states were saved, and, after all of the data was processed, the per-input synchrony was calculated.

A: **Echo State Network Hyperparameters**		**B**: **Rhythmic Sharing Hyperparameters**	
Parameter	**Range**	**Parameter**	**Range**
Nodes (C-MAPSS; SWaT/WADI)	512; 768	Δt	10
Node Degree	8	Link Strength Change (m)	$0.25 \rightarrow 1.00$
Input Weight	$0.25 \rightarrow 1.00$	ω_0	$0.01 \rightarrow 0.40$
Leakage (α)	$0.50 \rightarrow 0.95$	ϵ_1	$-0.2 \rightarrow -0.1$
Spectral Radius	$0.20 \rightarrow 0.80$	ϵ_2	$0.3 \rightarrow 1.0$

Table S1: Range of hyperparameters searched for **A**: the echo state network and **B**: the rhythmic sharing algorithm. Grid search maximized reconstruction accuracy and sensitivity to changing dynamics.