Advancing RFI-Detection in Radio Astronomy with Liquid State Machines

Nicholas J. Pritchard

International Centre for Radio Astronomy Research
University of Western Australia
Perth, Australia
0000-0003-0587-2310

Mohammed Bennamoun

School of Physics, Mathematics and Computing University of Western Australia Perth, Australia 0000-0002-6603-3257

Andreas Wicenec

International Centre for Radio Astronomy Research
University of Western Australia
Perth, Australia
0000-0002-1774-5653

Richard Dodson

International Centre for Radio Astronomy Research
University of Western Australia
Perth, Australia
0000-0003-0392-3604

Abstract-Radio Frequency Interference (RFI) from anthropogenic radio sources poses significant challenges to current and future radio telescopes. Contemporary approaches to detecting RFI treat the task as a semantic segmentation problem on radio telescope spectrograms. Typically, complex heuristic algorithms handle this task of 'flagging' in combination with manual labeling (in the most difficult cases). While recent machine-learning approaches have demonstrated high accuracy, they often fail to meet the stringent operational requirements of modern radio observatories. Owing to their inherently time-varying nature, spiking neural networks (SNNs) are a promising alternative method to RFI-detection by utilizing the time-varying nature of the spectrographic source data. In this work, we apply Liquid State Machines (LSMs), a class of spiking neural networks, to RFI-detection. We employ second-order Leaky Integrate-and-Fire (LiF) neurons, marking the first use of this architecture and neuron type for RFI-detection. We test three encoding methods and three increasingly complex readout layers, including a transformer decoder head, providing a hybrid of SNN and ANN techniques. Our methods extend LSMs beyond conventional classification tasks to fine-grained spatio-temporal segmentation. We train LSMs on simulated data derived from the Hydrogen Epoch of Reionization Array (HERA), a known benchmark for RFIdetection. Our model achieves a per-pixel accuracy of 98% and an F1-score of 0.743, demonstrating competitive performance on this highly challenging task. This work expands the sophistication of SNN techniques and architectures applied to RFI-detection, and highlights the effectiveness of LSMs in handling fine-grained, complex, spatio-temporal signal-processing tasks.

Index Terms—radio astronomy, spiking neural networks, neuromorphic computing, supervised learning, adaptive LiF

I. INTRODUCTION

Radio astronomy is a data-intensive science that heavily relies on sophisticated high-performance computing solutions to meet the demands of contemporary telescopes. Detecting and mitigating Radio Frequency Interference (RFI) from terrestrial and satellite communications is a growing challenge. The sheer

This work was supported by a Westpac Future Leaders Scholarship, an Australian Government Research Training Program Fees Offset and an Australian Government Research Training Program Stipend.

data volume and increased sensitivity of modern and upcoming radio telescopes are pushing the limit of traditional RFIdetection methods, which often combine complex heuristic algorithms and manual inspection.

Machine learning (ML) approaches, particularly convolutional neural networks (CNNs), have been applied to RFI-detection with some success [1]–[3]. However, these methods typically treat RFI-detection as a binary segmentation task on two-dimensional spectrograms and can be computationally expensive, limiting their real-time applicability in operational settings [4].

Spiking Neural Networks (SNNs) are time-varying systems inspired by biological neural systems [5]. Exploiting SNN's dynamic nature to process the temporal nature of radio astronomy observations is an appealing and promising alternative to ANN-based RFI-detection approaches. Recent studies have begun to explore the application of SNNs to RFI-detection by leveraging their dynamic nature to handle spatio-temporal data effectively [6]–[8].

Liquid State Machines (LSMs), a form of reservoir computing with SNNs [9], have shown potential in various signal processing tasks, including temporal pattern recognition [10]. Unlike other deep learning approaches, which require training all parameters in a network, LSMs require only training a readout layer, significantly reducing training complexity and computational cost. Their capacity to handle time-varying inputs coupled with lighter-weight training requirements makes them appealing candidates for addressing the challenges of RFI-detection in radio astronomy, where radio spectrograms evolve dynamically over time.

In this work, we extend the application of LSMs to the domain of RFI-detection by:

 Applying LSMs to RFI-detection formulated as timeseries segmentation with synthetic data emulating the Hydrogen Epoch of Reionization Array (HERA) telescope.

- Utilizing second-order Leaky Integrate-and-Fire (LiF) spiking neurons to RFI-detection with SNNs testing three encoding methods: latency, rate and direct current encoding.
- Testing three readout layer configurations: linear, ReLU and transformer-decoder. Utilizing the spiking behavior of reservoir as memory for the transformer layer.

Our results demonstrate that LSMs can achieve competitive performance on this challenging segmentation task, expanding the collection of tasks to which they have been successfully applied. This work advances the sophistication of SNN-based methods for RFI-detection in radio astronomy and establishes a precedent for applying LSMs to complex spatio-temporal problems. By leveraging LSMs' inherently temporal nature, we aim to develop more efficient data-driven and real-time RFI-detection methods.

II. RELATED WORK

RFI-detection and mitigation is essential to operating contemporary radio telescope observatories. RFI-detection, termed 'flagging' requires ingesting partially processed 'visibility' data from the telescope's interferometric correlator and outputting a boolean mask (flag) indicating the presence of RFI at a particular time-step in a particular channel. Contemporary RFI-detection methods rely on cumulative sum algorithms to identify regions of increased power within spectrograms in addition to expertly tuned heuristics and filters [11]. While these methods are operationally efficient, they require ongoing fine-tuning, expert oversight, and, in challenging cases, manual labeling, making them difficult to scale for next-generation radio telescopes such as the Square Kilometer Array (SKA) [12].

The sheer volume of available historical observation data and the promise of increasing future observation data volumes provide strong motivation for investigating data-driven and machine-learning techniques to aspects of radio astronomy processing. Many have tackled RFI-detection with machine learning as a two-dimensional segmentation problem similar to traditional algorithms. UNet-like architectures, in particular, have been widely investigated and achieve high accuracy in segmenting spectrograms using the outputs of well-tuned traditional algorithms as training supervision [1]–[3]. To address the undesirable reliance on using existing flagging techniques as supervision targets, unsupervised nearest-neighbor methods [13] and auto-encoder-based anomaly detection methods have additionally been investigated [4], [14]. Most recently, preliminary work into deploying vision-transformer models [15] shows promise, albeit at a high computational cost. Du Toit et al. [16] provide a comprehensive comparison of deep learning approaches to RFI-detection. While many methods provide excellent detection performance, they incur high operational requirements compared to traditional methods.

The promise of marrying the time-varying nature of radio astronomy observations and SNNs has been known to be appealing for several years [17], [18]; however, few have investigated applying SNNs and neuromorphic computing to

any tasks in radio astronomy observations or RFI-detection. Pritchard et al. [6] provided the first attempt to apply SNNs to RFI-detection by adapting an auto-encoder-based anomaly detection scheme [4] through ANN2SNN conversion. While accurate, this method only reduced the memory requirements compared to the original technique rather than providing improved detection performance or time efficiency. Later, the same team moved to supervised training small SNNs by reformulating RFI-detection as a time-series segmentation problem [7]. We borrow the problem formulation and datasets of this later work. While pioneering, these two works only use first-order leaky-integrate and fire (LiF) neurons and do little to consider biological plausibility or inspiration in their approaches. We aim to partially address these shortcomings in this work.

Liquid State Machines (LSMs) offer a biologically plausible time-series processing paradigm based on SNNs [9]. Despite their promise and elegance, correctly tuning the liquid state and neuron dynamics are key to achieving a suitably dynamic but not completely chaotic liquid [19]. LSMs have previously been shown to handle complex time-series processing tasks such as robot arm path prediction [20], generalized pattern recognition [10], uni-variate time-series classification [21], and speech recognition reaching performance competitive with backpropagation through time trained SNNs [22]. Moreover, the simplicity in defining the un-trained liquid reservoir makes LSMs amenable to hardware acceleration [23]-[26]. RFIdetection, formulated as a time-series segmentation task, provides an exceedingly challenging task for LSMs, which have largely been focused on classification problems. Moreover, this work is the first to explore the application of LSMs to radio astronomy and RFI-detection in particular. By extending LSMs to perform fine-grained RFI-detection, we aim to investigate their utility in addressing the unique challenges posed by radio astronomy.

III. METHODS

We continue from the RFI-detection formulation Pritchard et al. [7] present, which casts what is traditionally treated as a two-dimensional image-segmentation problem as a multivariate time-series segmentation problem more suited for SNN evaluation. By iterating on the model architectures tested, we uncover performance in spike-encoding methods previously considered unsuitable for this task. Our main contribution is expanding the array of architectures, and neuron models applied to this formulation of RFI-detection. Fig. 1 outlines our overall approach encompassing comparing encoding methods and LSM readout layers. The example encodings show that rate encoding exhibits the most sparsity, latency encoding provides an offset perturbation in each time step, and direct encoding retains input magnitude. Including a second-order LiF neuron model introduces complexity and temporal dynamics, which we balance against the biologically inspired LSM architecture, which is noted for its relative training ease. Comparing increasingly complex readout layers further expands the insight into this problem space and broadens the set of tasks to which LSMs have been applied.

This section outlines the time-series segmentation formulation of RFI-detection with SNNs, the dataset tested and pre-processing techniques applied, our LSM definition, parameters, and spiking neuron model. It briefly introduces the encoding methods tested, the readout layer variations tested, the hyper-parameter-optimization routine, and finally, the compute, training, and simulation environment details.

A. RFI-Detection with SNNs

We adopt the time-series segmentation approach for RFIdetection as Pritchard et al. [7] propose. In this formulation, the loss function supervises the spiking network to output a boolean mask of 'flags' as:

$$\mathcal{L}_{sup} = min_{\theta_n}(\Sigma_t^T \mathcal{H}(m_{\theta_n}(E(V(\upsilon,t,b))), F(G(\upsilon,t,b)))) \tag{1}$$

where V is the original visibility data, G is the supervised ground truth, E and F are spike-encoding and decoding functions, respectively, θ_n are the parameters of some classifier m and \mathcal{H} is a comparator function between the supervision masks and output of the network. Finally, v is the visibility frequency, t the time and t baseline (pair of antennae) in the original visibility data. This formulation introduced the concept of exposure-time t. To allow the network dynamics to produce an output per time step (in the original spectrogram), each time step is expanded (and correspondingly compressed) by t and t.

Three spike-encoding methods were tested:

- Latency encoding encodes inputs and outputs as a spike delay proportional to its value as described previously in [7].
- Rate encoding converts input and output intensity into spike frequency. Gradient updates are calculated on the average activity of the network per spectrogram time-step, unlike in [7].
- **Direct encoding** is an approach where *E* is an identity function, treating the visibility values as driving currents for each input channel, repeating the value for the desired exposure time. Output decoding is the same as rate decoding.

Previous methods making use of this formulation provide supervision on the output spike-rasters directly [7], [8]; in this work, we provide supervision on the decoded segmentation decisions, providing a stronger gradient influence. We investigate several encoding methods alongside readout layer designs as they encourage the network to represent information in different ways; some vastly more effective than others. There is no a priori way to know which method will perform best and it could be the case that a particular encoding method paired with a particular readout architecture performs especially well.

B. Dataset

The dataset we use originates from the HERA simulator [27] as prepared in [4] and has been used to benchmark several

other RFI-detection algorithms [16]. As is common in other machine-learning-based RFI-detection methods, the original 512×512 pixel spectrograms are split into 32×32 chunks to shrink the model size significantly. The specific version this work uses employs the divisive normalization techniques introduced in [8] as it has been shown to improve performance significantly. Like the original, this dataset contains 420 training and 140 test spectrograms with an overall RFI contamination of around 3%. See [4], [7] for a more detailed description of this dataset.

C. Liquid State Machine Parameters

The LSM architecture consists of a reservoir of recurrently connected spiking neurons with a random distribution of parameters to encourage a dynamic response to various inputs. In line with known literature [22], we initialize the reservoir with an 80/20 ratio of excitatory and inhibitory connections. We use second-order Leaky Integrate and Fire (LiF) neurons characterized by the dynamics:

$$I_{syn}^{t+1} = I_{syn}^{t} + S_{in}(t) + S_{rec} \cdot W_{rec}$$

$$I_{syn}^{t+1} = I_{syn}^{t} \cdot \exp(-dt/\tau_{syn})$$

$$V_{mem}^{t+1} = V_{mem}^{t} \cdot \exp(-dt/\tau_{mem})$$

$$V_{mem}^{t+1} = V_{mem}^{t} + I_{syn}^{t} + b + \sigma \zeta(t)$$
(2)

as per Rockpool's implementation [28] where V_{mem} is the membrane voltage potential, I_{syn} is the synaptic current, S_{in} are the incoming spikes, S_{rec} are recurrent spikes, W_{rec} is the current weight, τ_{syn} is the synaptic time-constant, τ_{mem} is the membrane time-constant, b is a bias current, c0 is a Wigner random noise function, and the neuron membrane is reset by subtraction. We initialize the reservoir neurons with time constants c1 ranging uniformly from 0.001 to 0.01 and uniformly random recurrent weights.

D. Readout Layer Variations

We only train the readout layer in LSMs, leveraging the reservoir's dynamics. We evaluated three readout architectures of increasing complexity, each mapping directly to the 32 output channels:

- Linear Layer: A single layer of weights.
- Linear + ReLU Layer: Linear weights with ReLUactivated artificial neurons.
- Transformer Decoder: A transformer decoder block with four attention heads, ReLU-activated neurons and using the spiking history as memory.

Table I summarizes our model architectures, in all cases the input and output is 32 spectrogram channels. Since we train only the readout layer, these models significantly reduce computational overhead compared to other similarly scaled SNNs or ANNs, making them attractive for real-time applications.

E. Training Environment

All experiments were conducted using SynSense's Rockpool library [28] which is based on PyTorch, in combination with PyTorch Lightning [29] for efficient data-driven parallelism.

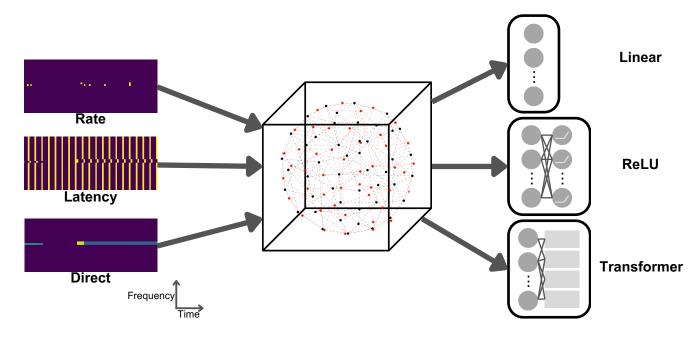


Fig. 1: Overall approach to RFI-detection with liquid state machines. One of several encoding methods (left) is fed into a recurrently connected LSM (center), the output of which feeds into one of three decoding architectures (right).

TABLE I: Neural Network Architectures. N_{LSM} refers to the size of the reservoir.

Layer	Input	Output
Input	32	$\overline{N_{LSM}}$
Reservoir	N_{LSM}	N_{LSM}
Readout (Linear, ReLU, Transformer)	N_{LSM}	32

Code is publicly available online¹. All trials were optimized with Adam with an initial learning rate of 0.0001 scheduled to reduce on validation loss plateaus by a factor of 0.5 and patience of 10. All models were trained for 100 epochs except hyper-parameter trials, which were trained for 50 epochs and 10% of the original training set. These training parameters are in-line with previous investigations [7], [8] using the same datasets, although it may be possible to extract further performance with more epochs.

F. Hyper-Parameter Optimization

We use Optuna's [30] tree-structured Parzan estimation algorithm to optimize for F1-Score. We select to optimize for F1-Score as it reflects the inherent imbalance present in the RFI-detection task; RFI typically comprises at most 10% of all pixels, and therefore, while per-pixel accuracy may be high, this can be boosted by a large proportional of true-negative predictions. Table II describes the parameters selected for optimization and their ranges. Table III contains the final parameters selected for each spike-encoding method

TABLE II: Ranges for parameters included in hyper-parameter optimization routines.

Parameter	Range
Input Sparsity	[0.0, 1.0]
Exposure Reservoir Size	[1, 2, 4, 8, 16, 32] [512, 1024, 2048, 4096, 8192]

TABLE III: Hyper-parameter optimization results

Readout Layer	Encoding Method	F1-Score	Input Sparsity	Exposure	Reservoir Size
Linear	Rate	0.55	0.19	16	8192
	Latency	0.1	0.298	4	4096
	Direct	0.654	0.074	1	4096
ReLU	Rate	0.552	0.178	16	1024
	Latency	0.100	0.377	16	4096
	Direct	0.481	0.149	4	2048
Transformer	Rate	0.778	0.285	4	2048
	Latency	0.094	0.382	2	4096
	Direct	0.785	0.230	32	1024

and readout layer over 25 trials each. Each of the 25 trials is run on 10% of the original training set and trained for 50 epochs each.

Optuna's fANOVA importance evaluator found input sparsity to be the most important variable by a significant margin, followed by exposure time and liquid size in all trials. Commenting briefly on the trial results, latency encoding exhibits the weakest performance, followed by rate encoding and then direct encoding. Input sparsity, in line with literature expectations [22], falls below 40% connectivity in

¹https://github.com/pritchardn/SNN_RFI_LSM

all cases. Moreover, latency encoding requires the highest input density, followed by rate encoding and, finally, direct encoding, indicating a potential relationship between encoding efficiency. As the readout layers increase in complexity, so too does input density. Exposure and reservoir size otherwise span the entire range of inputs, reflecting their lessened importance. Moreover, performance improves as the readout layer complexity increases resulting in direct encoding with a transformer readout layer providing the most promising parameters.

IV. RESULTS

Table IV contains the averaged performance of each encoding method and readout layer over five trials. We compare methods in Accuracy, Area Under the Receiver-Operator Curve AUROC, Area Under the Precision-Recall Curve AUPRC and F1-Score, paying greater attention to class balanced-performance metrics AUPRC and F1-Score. We discuss performance differences between encoding techniques and readout layer complexity and compare them to existing SNN-based RFI-detection methods.

A. Encoding Comparison

Generally, performance differences between the encoding methods in Table IV match the performance differences seen in the hyper-parameter trials in Table III. Latency encoding performs significantly worse than rate or direct encoding in all cases by quite some margin. We hypothesize this is due to the freezing of the reservoir parameters since the readout layer can only affect the precise spike timings so much. This deviates from prior results [7], which found latency encoding to perform better than other methods and rate encoding in particular. We expect this difference to arise from the fact that prior works directly supervised the output spike rasters rather than the decoded predictions, as we do in this work. Rate and direct encoding methods perform markedly better and similarly to each other, something we would expect, given they share a common decoding technique. Direct encoding, however, performs best independently of the readout layer used, providing an $\approx 5-10\%$ improvement in AUPRC and F1-score metrics over rate encoding. Moreover, we see that latency encoding is subject to the greatest variation in performance with the exception of the F1-Score of direct encoding with a linear decoding layer which is likely the result of employing a short exposure time of a single time-step. These results suggest that under an LSM framework, rate decoding with direct encoding provides the best performance in contrast to existing work [7].

B. Readout Layer Comparison

We additionally compare the differences between increasingly complex readout layers. We see relative subsequent improvements in accuracy between linear, ReLU and transformer readouts in rate and direct encoding. AUROC improves by 0.02 and then 0.07 for rate encoding across ReLU and transformer readouts, but we see a regression of 0.02 followed by an

improvement of 0.11 (over the linear readout result) for direct encoding. Moving from linear to ReLU and transformer layers yields 0.03 and 0.08 incremental improvements in the AUPRC score for rate encoding and 0.01 and 0.07 for direct encoding, respectively. Finally, F1 Scores improve by 0.08 when using a ReLU readout, 0.11 when moving to a transformer readout for rate encoding, and incrementally 0.07 and 0.08 for direct encoding. In combination with the increased input density suggested by the hyper-parameter optimizations in Table III, it is clear that the increasingly complex readout layers do provide significant performance improvements and, more notably, that the spiking output of the reservoir serves as a useful memory in the transformer decoder.

C. Comparison to Related Methods

Table V compares other ANN and SNN-based RFI-detection methods on this dataset. The other methods exhibit superior performance, however LSMs are typically employed for classification problems and not complex time-series segmentation tasks. We also note that the ANN and ANN2SNN auto-encoder examples [4], [6] are convolutional and therefore used at least an order of magnitude more parameters. The rate-coded LSM with a transformer layer readout shows results comparable to a latency-encoded SNN-MLP [7]. While the ANN2SNN auto-encoder-based method demonstrates the best F1-Score, it requires 512 exposure steps per 32 × 32 spectrogram patch and a significantly more intensive CNN-based auto-encoder architecture [6]. While our LSM approach does not surpass the best-performing ANN2SNN or SNN-MLP methods, the training process is significantly simpler as the spiking neurons do not need to be trained. Fig. 2 illuminates why performance from the LSM may lag that of the current state-of-the-art; the LSM struggles to deal with instantaneous, wide-band RFI. This may be resolved by employing a second reservoir with dynamics tuned to handle faster changing dynamics and integrating the output of both reservoir columns.

V. Conclusions

In this work, we demonstrate that liquid state machines can perform tasks beyond classification by tasking an LSM to detect RFI in a well-known synthetic dataset. We tested multiple encoding techniques and increasingly complex readout layers, finding that direct exposure with rate-decoding and a transformer decoder readout layer provide the best performance. We also find that utilizing the liquid's spiking output as the memory for the transformer layer significantly improves performance, providing a hybrid of both spiking and artificial neural networks to handle a task previously unseen for LSMs. While results lag the state of the art for trained SNNs and ANN2SNN conversions, this work recovers rate encoding as a viable scheme. The relative computational simplicity of training LSMs continues to make them an appealing direction for investigation. Moreover, this work is the first to employ second-order integrate and fire LiF neurons. This work demonstrates SNNs' fundamental flexibility and introduces a biologically inspired network model for RFI-detection. Future

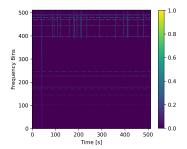
TABLE IV: Final results averaged over 5 trials presented as mean and standard deviation. Best scores are in bold.

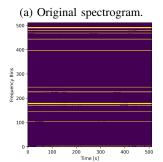
Readout Layer	Encoding Method	Accı	ıracy	AUI	ROC	AUI	PRC	F1-So	core
Linear	Rate	0.9742	0.0019	0.7152	0.0220	0.6429	0.0293	0.5569	0.0416
	Latency	0.3599	0.012	0.4959	0.0052	0.3501	0.0063	0.0795	0.0030
	Direct	0.9772	0.0049	0.7226	0.0576	0.7087	0.0552	0.5976	0.1058
ReLU	Rate	0.9748	0.0049	0.7335	0.0538	0.6778	0.0546	0.5942	0.0937
	Latency	0.9544	0.0036	0.5021	0.0018	0.1317	0.1724	0.0759	0.0013
	Direct	0.9781	0.0011	0.7085	0.0194	0.7175	0.0145	0.6658	0.0237
Transformer	Rate	0.9804	0.0007	0.8081	0.0068	0.7588	0.0038	0.7108	0.0088
	Latency	0.2412	0.0880	0.5079	0.0121	0.4237	0.0339	0.0790	0.0032
	Direct	0.9824	0.0010	0.8318	0.0107	0.7808	0.0082	0.7432	0.0144

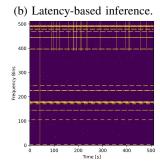
TABLE V: Results compared to other ANN and SNN-based RFI-detection methods

Work	Method	AUROC	AUPRC	F1-Score
[4]	ANN Auto-Encoder	0.981	0.927	0.910
[6]	ANN2SNN - Auto-Encoder	0.944	0.910	0.953
[7]	BPTT - SNN-MLP	0.929	0.785	0.761
[8]	BPTT - SNN-MLP	0.968	0.914	0.907
This work	LSM	0.842	0.781	0.743

directions could expand the number of columns present in the reservoir to handle both slow and fast-varying RFI, test an SNN-based decoding layer using the membrane potential as a decoding method in addition to the encoding demonstrated here, and finally expand the learning paradigms to include unsupervised or partial evolution of the reservoir itself. There is a growing understanding that SNNs can bring the computational power of connectionist methods while saving on energy usage, and this work adds to the body of work in the fledgling field of applying SNNs to astronomical sciences.







(c) Ground-truth RFI mask.

Fig. 2: Example original 512×512 sized spectrogram, output of a transformer-readout direct-encoded LSM and the supervised ground-truth. It is clear that the LSM struggles with wide-band, short-duration RFI and periodic RFI.

REFERENCES

[1] J. Akeret, C. Chang, A. Lucchi, and A. Refregier, "Radio frequency interference mitigation using deep convolutional neural networks," *Astronomy and Computing*, vol. 18, pp. 35–39, Jan. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S2213133716301056

- [2] Z. Yang, C. Yu, J. Xiao, and B. Zhang, "Deep residual detection of radio frequency interference for FAST," *Monthly Notices of the Royal Astronomical Society*, vol. 492, no. 1, pp. 1421–1431, Feb. 2020. [Online]. Available: https://doi.org/10.1093/mnras/stz3521
- [3] A. Vafaei Sadr, B. A. Bassett, N. Oozeer, Y. Fantaye, and C. Finlay, "Deep learning improves identification of Radio Frequency Interference," *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 1, pp. 379–390, Oct. 2020. [Online]. Available: https://doi.org/10.1093/mnras/staa2724
- [4] M. Mesarcik, A.-J. Boonstra, E. Ranguelova, and R. V. van Nieuwpoort, "Learning to detect radio frequency interference in radio astronomy without seeing it," *Monthly Notices of the Royal Astronomical Society*, vol. 516, no. 4, pp. 5367–5378, Nov. 2022. [Online]. Available: https://doi.org/10.1093/mnras/stac2503
- [5] T. P. Trappenberg, Fundamentals of computational neuroscience, 2nd ed. Oxford, UK; Oxford University Press, 2010, publication Title: Fundamentals of computational neuroscience.
- [6] N. J. Pritchard, A. Wicenec, M. Bennamoun, and R. Dodson, "RFI Detection with Spiking Neural Networks," *Publications of the Astronomical Society of Australia*, pp. 1–11, Apr. 2024. [Online]. Available: https://doi.org/10.1017/pasa.2024.27
- [7] —, "Supervised Radio Frequency Interference Detection with SNNs," in 2024 International Conference on Neuromorphic Systems (ICONS), Jul. 2024, pp. 102–109. [Online]. Available: https://ieeexplore.ieee.org/document/10766534
- [8] —, "Spiking Neural Networks for Radio Frequency Interference Detection in Radio Astronomy," Dec. 2024, arXiv:2412.06124 [cs]. [Online]. Available: http://arxiv.org/abs/2412.06124
- [9] W. Maass, T. Natschläger, and H. Markram, "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531– 2560, Nov. 2002, conference Name: Neural Computation. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6789852
- [10] E. Goodman and D. Ventura, "Spatiotemporal Pattern Recognition via Liquid State Machines," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, Jul. 2006, pp. 3848–3853, iSSN: 2161-4407. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1716628/
- [11] A. Offringa, "AOFlagger: RFI Software," Astrophysics Source Code Library, pp. ascl–1010, 2010.
- [12] E. Vermij, L. Fiorin, R. Jongerius, C. Hagleitner, and K. Bertels, "Challenges in exascale radio astronomy: Can the SKA ride the technology wave?" *The International Journal of High Performance Computing Applications*, vol. 29, no. 1, pp. 37–50, Feb. 2015, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/1094342014549059
- [13] C. J. Wolfaardt, "Machine learning approach to radio frequency interference(rfi) classification in radio astronomy," Thesis, Stellenbosch University, Mar. 2016, accepted: 2016-03-09T14:22:12Z. [Online]. Available: https://scholar.sun.ac.za:443/handle/10019.1/98464
- [14] D. J. van Zyl and T. L. Grobler, "Remove First Detect Later: a counter-intuitive approach for detecting radio frequency interference in radio sky imagery," *Monthly Notices of the Royal Astronomical Society*, vol. 530, no. 2, pp. 1907–1920, May 2024. [Online]. Available: https://doi.org/10.1093/mnras/stae979
- [15] X. Ouyang, H. Dreuning, M. Mesarcik, and R. V. van Nieuwpoort, "Hierarchical vision transformers for RFI mitigation in radio astronomy," Bariloche, Argentina, Oct. 2024.
- [16] C. D. Du Toit, T. L. Grobler, and D. J. Ludick, "A comparison framework for deep learning RFI detection algorithms," *Monthly Notices* of the Royal Astronomical Society, vol. 530, no. 1, pp. 613–629, May 2024. [Online]. Available: https://doi.org/10.1093/mnras/stae892
- [17] N. M. Scott, "Evolving Spiking Neural Networks for Spatioand Spectro- Temporal Data Analysis: Models, Implementations, Applications," Ph.D. dissertation, Auckland University of Technology, 2015. [Online]. Available: https://hdl.handle.net/10292/10601
- [18] N. Kasabov, N. M. Scott, E. Tu, S. Marks, N. Sengupta, E. Capecci, M. Othman, M. G. Doborjeh, N. Murli, R. Hartono, J. I. Espinosa-Ramos, L. Zhou, F. B. Alvi, G. Wang, D. Taylor, V. Feigin, S. Gulyaev, M. Mahmoud, Z.-G. Hou, and J. Yang, "Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications," *Neural Networks*, vol. 78, pp. 1–14, Jun. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608015001860

- [19] E. Goodman and D. Ventura, "Effectively using recurrently-connected spiking neural networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005., vol. 3, Jul. 2005, pp. 1542–1547 vol. 3, iSSN: 2161-4407. [Online]. Available: https://ieeexplore.ieee.org/document/1556107
- [20] H. Burgsteiner, M. Kröll, A. Leopold, and G. Steinbauer, "Movement prediction from real-world images using a liquid state machine," *Applied Intelligence*, vol. 26, no. 2, pp. 99–109, Apr. 2007. [Online]. Available: https://doi.org/10.1007/s10489-006-0007-1
- [21] R. Gaurav, T. C. Stewart, and Y. Yi, "Reservoir based spiking models for univariate Time Series Classification," Frontiers in Computational Neuroscience, vol. 17, Jun. 2023, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/journals/ computational-neuroscience/articles/10.3389/fncom.2023.1148284/full
- [22] L. Deckers, I. J. Tsang, W. Van Leekwijck, and S. Latré, "Extended liquid state machines for speech recognition," Frontiers in Neuroscience, vol. 16, Oct. 2022, publisher: Frontiers. [Online]. Available: https://doi.org/10.3389/fnins.2022.1023470
- [23] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A Digital Liquid State Machine With Biologically Inspired Learning and Its Application to Speech Recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2635–2649, Nov. 2015, conference Name: IEEE Transactions on Neural Networks and Learning Systems. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7024132
- [24] V. Saraswat, A. Gorad, A. Naik, A. Patil, and U. Ganguly, "Hardware-Friendly Synaptic Orders and Timescales in Liquid State Machines for Speech Classification," in 2021 International Joint Conference on Neural Networks (IJCNN), Jul. 2021, pp. 1–8, iSSN: 2161-4407. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9534021
- [25] L. Wang, Z. Yang, S. Guo, L. Qu, X. Zhang, Z. Kang, and W. Xu, "LSMCore: A 69k-Synapse/mm2 Single-Core Digital Neuromorphic Processor for Liquid State Machine," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 5, pp. 1976–1989, May 2022, conference Name: IEEE Transactions on Circuits and Systems I: Regular Papers. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9704873
- [26] N. Lin, S. Wang, Y. Li, B. Wang, S. Shi, Y. He, W. Zhang, Y. Yu, Y. Zhang, X. Qi, X. Chen, H. Jiang, X. Zhang, P. Lin, X. Xu, Q. Liu, Z. Wang, D. Shang, and M. Liu, "Resistive memory-based zero-shot liquid state machine for multimodal event data learning," Jul. 2023, arXiv:2307.00771. [Online]. Available: http://arxiv.org/abs/2307.00771
- [27] D. R. DeBoer, A. R. Parsons, J. E. Aguirre, P. Alexander, Z. S. Ali, A. P. Beardsley, G. Bernardi, J. D. Bowman, R. F. Bradley, C. L. Carilli, C. Cheng, E. d. L. Acedo, J. S. Dillon, A. Ewall-Wice, G. Fadana, N. Fagnoni, R. Fritz, S. R. Furlanetto, B. Glendenning, B. Greig, J. Grobbelaar, B. J. Hazelton, J. N. Hewitt, J. Hickish, D. C. Jacobs, A. Julius, M. Kariseb, S. A. Kohn, T. Lekalake, A. Liu, A. Loots, D. MacMahon, L. Malan, C. Malgas, M. Maree, Z. Martinot, N. Mathison, E. Matsetela, A. Mesinger, M. F. Morales, A. R. Neben, N. Patra, S. Pieterse, J. C. Pober, N. Razavi-Ghods, J. Ringuette, J. Robnett, K. Rosie, R. Sell, C. Smith, A. Syce, M. Tegmark, N. Thyagarajan, P. K. G. Williams, and H. Zheng, "Hydrogen Epoch of Reionization Array (HERA)," Publications of the Astronomical Society of the Pacific, vol. 129, no. 974, p. 045001, Mar. 2017, publisher: The Astronomical Society of the Pacific. [Online]. Available: https://dx.doi.org/10.1088/1538-3873/129/974/045001
- [28] D. R. Muir, F. Bauer, and P. Weidel, "Rockpool Documentaton," Sep. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3773845
- [29] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning
- [30] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.