

# Unsupervised Online Learning for Seizure Detection and Prediction

*Adelson Chua*



Electrical Engineering and Computer Sciences  
University of California, Berkeley

Technical Report No. UCB/EECS-2025-22

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-22.html>

May 1, 2025

Copyright © 2025, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Unsupervised Online Learning for Seizure Detection and Prediction

By

Adelson Chua

A dissertation submitted in partial satisfaction of the  
requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Rikky Muller, PhD, Chair  
Michael I. Jordan, PhD  
Daniela Kaufer, PhD

Summer 2023



## Abstract

### Unsupervised Online Learning for Seizure Detection and Prediction

by

Adelson Chua

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Rikky Muller, PhD, Chair

Implantable devices that record neural activity and detect seizures have been adopted for issuing warnings or triggering neurostimulation to suppress epileptic seizures. Traditional seizure detection systems rely on high-accuracy offline-trained machine learning classifiers that need manual retraining when seizure patterns change over time. For an implantable seizure detection system, a low-power, at-the-edge, online learning algorithm can be used to dynamically adapt to neural signal drifts, maintaining high accuracy without external intervention. This dissertation describes an energy-efficient classification algorithm based on logistic regression that incorporates stochastic gradient descent for online and unsupervised updates, ensuring sustained high classification accuracies over long periods of time. The online learning framework was implemented on two different on-chip variants: one being focused solely on seizure detection (which is referred to as SOUL); and another that combines both seizure detection and prediction (which is referred to as SPIRIT), which leverages the detector's outputs to continually improve the prediction accuracy without additional external inputs. The systems' performance was evaluated using long-term datasets, including cases with drifting seizure features, demonstrating high prediction and detection accuracies over extended periods through on-chip adaptation.

Both SOUL and SPIRIT managed to achieve comparable, if not better, detection and prediction accuracies versus other on-chip state-of-the-art work in this field. The online learning approach described in this thesis enabled the systems to maintain high accuracies over long periods of time while being very energy efficient. For SOUL, the combination of the proposed algorithmic approach and circuit-level optimizations resulted in an energy efficiency of 1.5 nJ/classification, which is at least 24x better than the state-of-the-art. It also consumes 0.1 mm<sup>2</sup> of area making it the smallest seizure detector classifier in the literature by a factor of 10x. For SPIRIT, using the same architectural optimizations that made an energy-efficient SOUL, the energy efficiency for prediction was 17.2 nJ/classification, which is at least 5.6x better than the only other on-chip seizure predictor in the literature. Compared to the same work, SPIRIT's power consumption is about 134x smaller at 17.2  $\mu$ W, while also being 28x smaller at 0.14 mm<sup>2</sup>. SPIRIT is the first on-chip seizure predictor that can retrain in an unsupervised manner while being more energy efficient than state-of-the-art.

# Table of Contents

I. Introduction .....	1
A. Epilepsy and EEG.....	1
B. Detecting seizures through EEG.....	3
II. Review of Related Work .....	8
A. Feature extraction units.....	8
B. Neural network-based classifiers .....	9
C. SVM-based classifiers .....	9
D. Decision-tree based classifiers.....	11
E. Online learning classifiers.....	12
III. Algorithm Design.....	14
A. Logistic regression as a classifier .....	14
B. Stochastic gradient descent for logistic regression .....	15
C. Enabling unsupervised learning.....	18
D. Making the unsupervised online learning robust.....	18
E. Tuning the online learning hyperparameters.....	19
F. Classifier stability.....	20
IV. On-chip Seizure Detection.....	22
A. Classifier features description.....	22
B. Feature extraction hardware.....	24
C. Classification and online learning hardware.....	27
V. On-chip Seizure Prediction .....	31
A. Classifier features description.....	33
B. Feature extraction hardware.....	35
C. Classification and online learning hardware.....	37
VI. Classifier Performance.....	41
A. Dataset description.....	41
B. Seizure detection performance on long-term iEEG data .....	43
C. Seizure detection performance on scalp EEG data .....	45
D. Seizure prediction performance .....	49

VII. Results and Discussions .....	52
A. Discussion on SOUL and SPIRIT chip results .....	52
B. Discussion on the use of logistic regression .....	54
C. Discussion on seizure detection performance.....	54
D. Discussion on seizure prediction performance .....	55
VIII. Future Work .....	56
IX. Conclusions.....	58
References.....	59
Appendix.....	63

## Acknowledgements

This work was supported by different foundations and funding agencies both in the US and the Philippines. The author would like to acknowledge the following:

- National Science Foundation (NSF) Faculty Early Career Development Program (CAREER) under Grant 1847710
- Wagner Foundation
- Weill Neurohub
- Army Research Office under Contract W911NF-16-1-0368
- Sponsors of the Berkeley Wireless Research Center (BWRC)
- Savio computational cluster resource provided by the Berkeley Research Computing (BRC) program
- Department of Science and Technology - Science Education Institute (DOST-SEI)
- University of the Philippines Diliman

The author would like to thank Dr. Mark Cook and Dr. Dean Freestone for graciously providing the intracranial EEG patient dataset which had been a crucial part in testing the algorithms developed in this thesis.

Finally, the author acknowledges his dissertation committee for making this work possible:

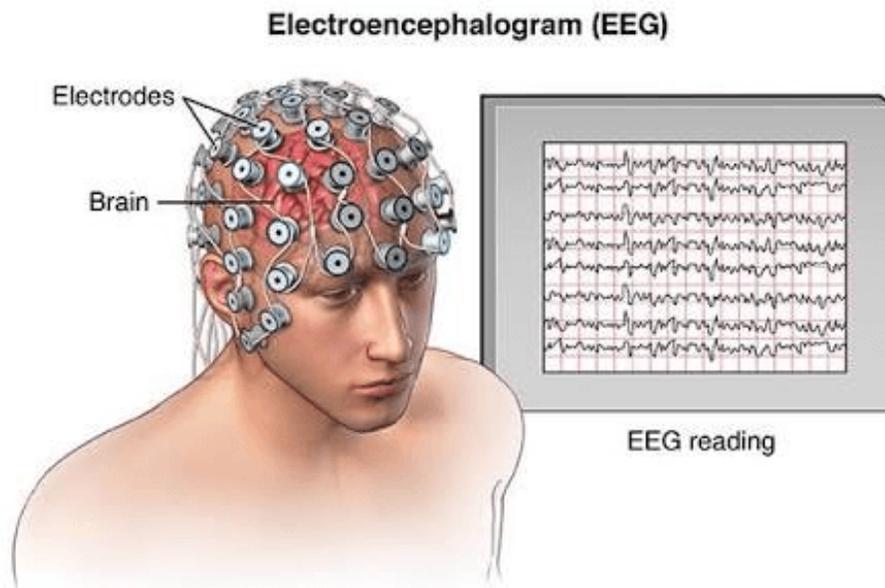
- Michael Jordan, for his input to the development of the machine learning algorithm proposed in this work. The initial idea of using logistic regression to enable unsupervised online learning was the one that jumpstarted this whole body of work.
- Daniela Kaufer, for providing a biological perspective on the target application of this work. Engineering would always aim for the best possible metric possible to compare against other works, but a different perspective is greatly appreciated.
- Rikky Muller, for believing in me and my capacity to make an impactful work in the field of integrated circuits, even if this work focused mainly on the digital domain. Thank you for pulling all the resources you had that would help me finish this work, which allowed me to collaborate with different people that I would not have contacted otherwise without your help.

# I. INTRODUCTION

## A. Epilepsy and EEG

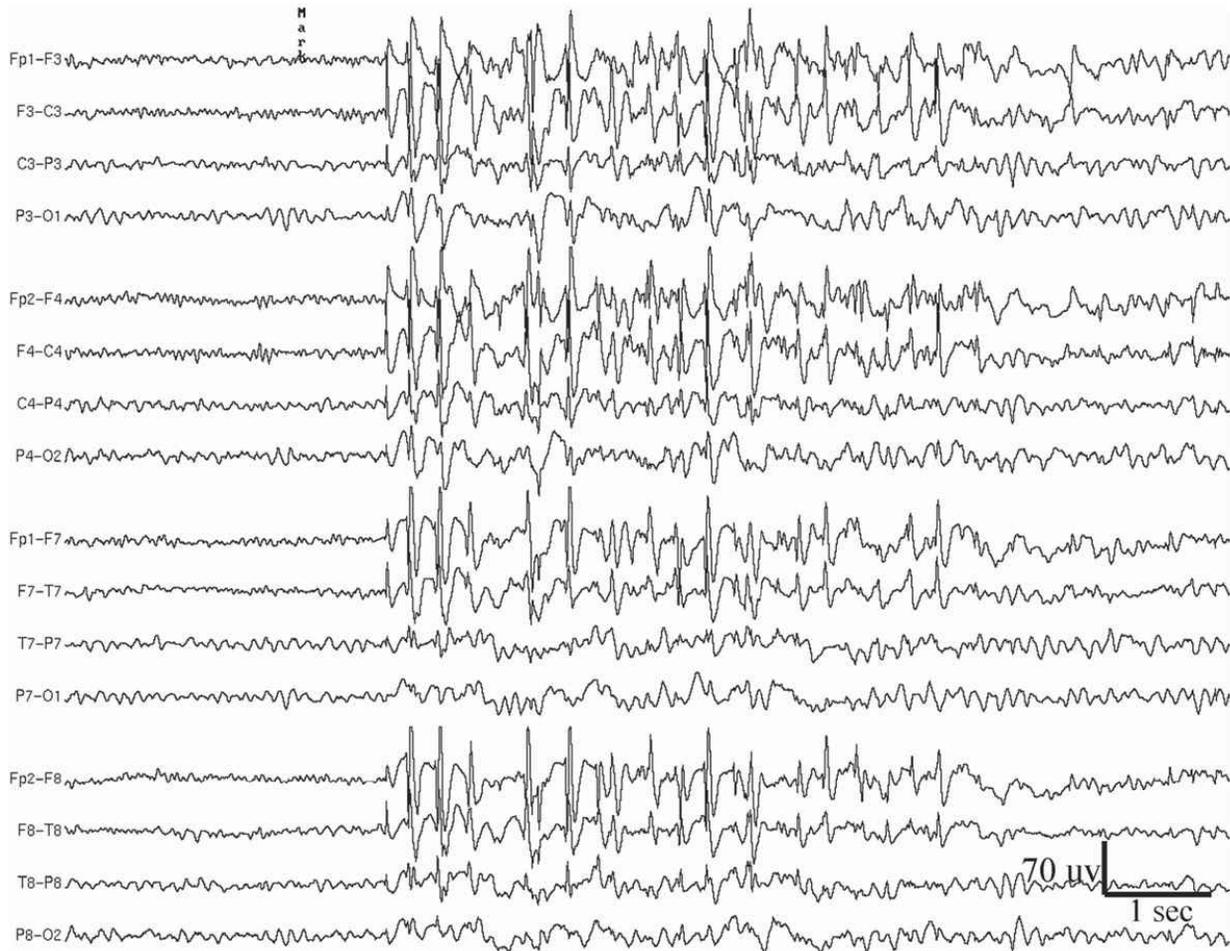
Epilepsy is a serious neurological disorder affecting around 50 million people worldwide [1] and is usually characterized by recurrent seizures. Epileptic seizures are manifestations of abnormal, excessive, or synchronous neuronal activity in the brain. They arise from the central nervous system and can lead to a variety of symptoms, ranging from brief lapses of attention or muscle jerks to severe and prolonged convulsions [2]. Seizure frequency varies greatly from person to person and can severely impact a person's quality of life. It can lead to physical injuries. Activities such as driving, swimming, or even walking can become hazardous. It also has some negative social impacts. People with epilepsy often face social stigma, which can lead to isolation, depression, and anxiety. Treating these seizures is crucial not only to improve the quality of life of those affected but also to prevent potential physical harm, reduce the risk of life-threatening conditions, and mitigate the psychosocial implications associated with the disorder.

Electroencephalography (EEG) is a non-invasive method used to record the electrical activity of the brain [3]. Fig. 1 illustrates how such devices are set up. It involves placing electrodes on the scalp to detect and record patterns of voltage fluctuations resulting from ionic current flows within the neurons of the brain. EEG is a primary tool for monitoring and detecting epileptic seizures due to its ability to capture the brain's electrical activity in real-time. This activity is displayed as a series of waveforms, with each waveform corresponding to the input from a specific electrode.



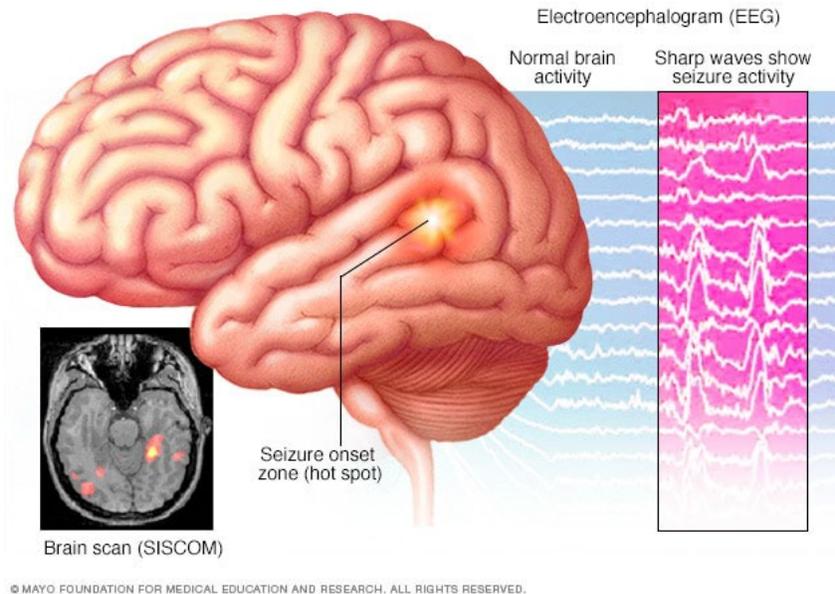
**Fig. 1.** EEG recording on the scalp and a typical EEG recording waveform. (Image credit: <https://eastneurology.com.au/eeeg-electroencephalogram-brain-wave-tests/>)

Seizure activity in the EEG is often characterized by repetitive, evolving, and stereotyped waveforms. A real example of an EEG recording with seizures is shown in Fig. 2. These can manifest as spikes, sharp waves, or specific rhythmic patterns. These can be in the form of a series of high-amplitude, high-frequency electrical signals [4]. The exact appearance can vary based on the type of seizure and its location in the brain. Focal seizures, illustrated in Fig. 3, which start in one area of the brain, will show abnormal activity limited to electrodes overlying that area, while generalized seizures will show widespread abnormalities. The spatial distribution of seizure activity on the EEG can provide clues about the seizure's origin, which can also aid in targeted treatment.



**Fig. 2.** An actual EEG recording of an ongoing seizure event. Labels on the left indicate the electrode placement. Units are shown on the lower right.

(Image credit: <https://thoracickey.com/eeg-in-adult-epilepsy/>)

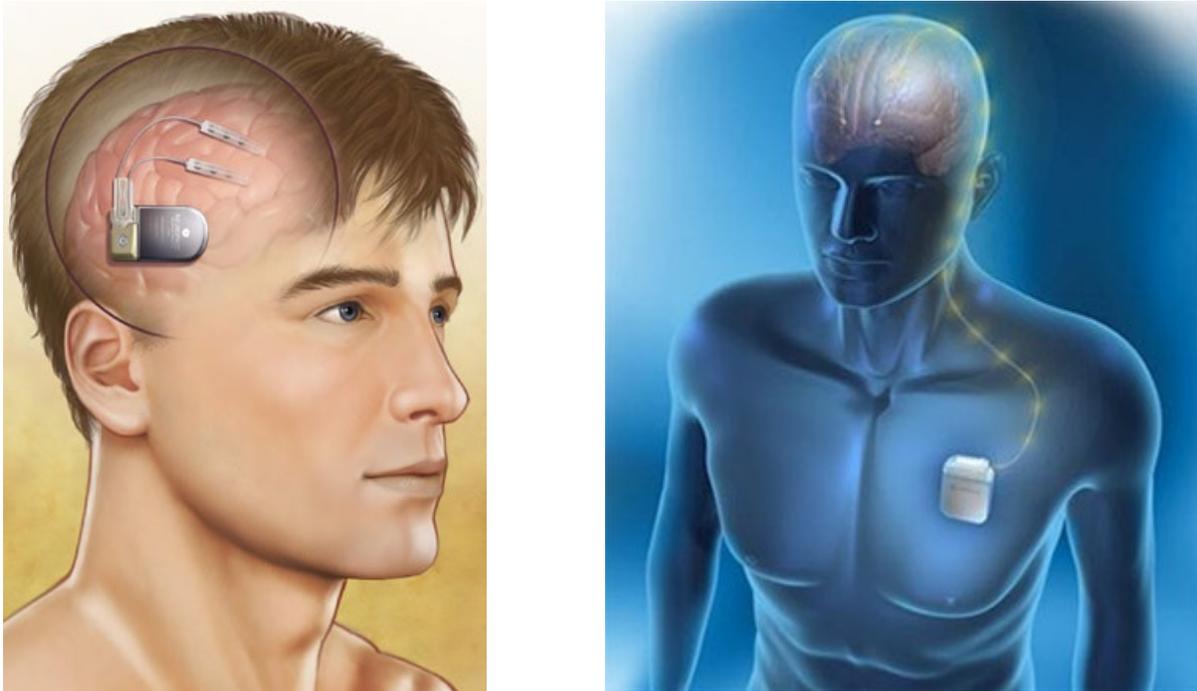


**Fig. 3.** Focal seizures start in one area of the brain (hot spot zone) and can be mapped through EEG recordings.

(Image credit: <https://www.mayoclinic.org/img-20456529>)

### *B. Detecting seizures through EEG*

Given the vast amount of data generated during continuous EEG monitoring, automated seizure detection algorithms have been developed. These algorithms analyze the EEG data to identify patterns consistent with seizure activity. Relevant features from the EEG data can be extracted allowing for better identification. These features can include time-domain characteristics, frequency-domain attributes, and statistical properties. Machine learning algorithms can then use these features to differentiate between normal and seizure activity. Commercially available advisory systems have been developed that warn patients when a seizure is about to occur. Closed-loop implantable neuromodulators have also been deployed for seizure treatment. These systems detect seizure events within an acceptable latency (typically <5 seconds [5,6]) and trigger neurostimulation to suppress the seizure. The NeuroPace Responsive neurostimulation (RNS) [5-7] and the Medtronic Deep-brain stimulation (DBS) [8,9], shown in Fig. 4, are two medically approved devices of this kind. These devices utilize a small, battery-powered pulse generator surgically implanted in the skull with two electrode leads that are implanted intracranially and/or epicortically. This treatment method has demonstrated clinical efficacy in terms of reducing long-term seizure occurrence, reporting a reduction of 66% of seizures by Year 6 for the NeuroPace RNS [5] and a 75% median reduction of seizures by Year 7 for the Medtronic DBS [9]. By automating the analysis process and potentially improving detection accuracy, machine learning techniques can play a pivotal role in the management and treatment of epilepsy.



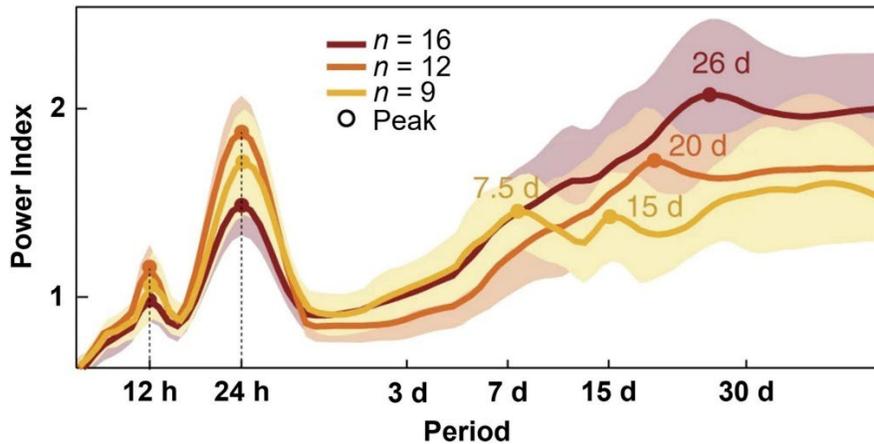
**Fig. 4.** NeuroPace RNS System (left) [5-7] and the Medtronic DBS System (right) [8,9]  
(Image credit: <https://www.medgadget.com/2013/11/neuropace-gets-fda-approval-for-rns-stimulator.html>,  
<https://www.tga.gov.au/news/safety-alerts/medtronic-deep-brain-stimulation-devices-multiple-models>)

While machine learning algorithms have found applications in seizure detection and treatment, there are challenges that need to be addressed for such an approach to be effective and efficient. We highlight three related challenges: 1) The changing seizure patterns over time that can degrade long-term detection accuracy; 2) The constant need for regular updates to these algorithms done by a medical professional to ensure that the machine learning models remain accurate; and 3) The trend of increasing computational complexity for seizure detection classifiers to remain accurate over long periods of time.

### **Challenge 1: The changing seizure patterns**

The dynamic nature of EEG seizure patterns poses challenges for machine learning-based seizure detection. EEG seizure patterns can exhibit variations over time due to a multitude of factors. As individuals age, their brain structures and functions evolve, leading to potential changes in EEG patterns. This is particularly pronounced in children and adolescents, whose brains are still developing. Factors such as stress, sleep deprivation, or hormonal changes can influence seizure patterns and their manifestation on EEG. The nature and characteristics of epilepsy can also change as the disease progresses. For instance, the focus of the seizures or the pathways they propagate through might shift over time. The introduction or alteration of antiepileptic drugs can also influence EEG patterns. Similarly, treatments like surgery or neurostimulation can lead to changes in the brain's electrical activity. Seizure patterns can vary based on the time of day, implying a patient-specific circadian profile [10]. Some individuals might be more prone to seizures during sleep or at specific times during the day, leading to variations in EEG patterns. Finally, shifting

electrode placement can also play a role as these can dramatically change the impedance seen by the EEG electrodes which can degrade signal quality [11]. Overall, as EEG seizure patterns change, the underlying data distribution that the machine learning model was trained on might no longer be representative. This phenomenon, known as model drift, can lead to decreased model accuracy over time.

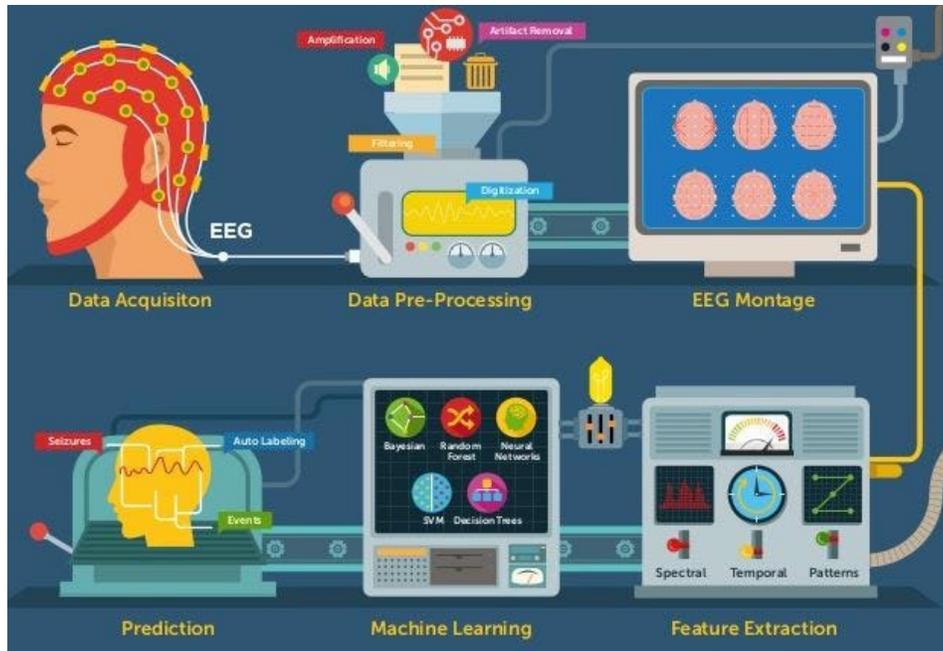


**Fig 5.** Spectral power of EEG over a 30-day period demonstrating a change in signal power that can cause misclassifications from a machine learning algorithm.

(Image credit: <https://www.frontiersin.org/articles/10.3389/fnins.2022.936104/full>)

## Challenge 2: The need for external intervention

Given the potential for changing EEG patterns, machine learning models might require regular retraining to maintain their performance. Requiring a medical professional to regularly update a machine learning algorithm to maintain high detection accuracies introduces several inefficiencies and challenges. The integration of machine learning into medical applications, especially in areas like EEG seizure detection, aims to automate and enhance the diagnostic and monitoring processes. However, frequent manual intervention can negate some of these benefits. Regular updates can lead to increased operational costs. It can also be an inconvenience to the user as a regular visit to a medical facility has to be scheduled and the entire retraining process can take time. A typical EEG retraining flow is shown in Fig. 6. Manual intervention can also increase the risk of errors. Mistakes in data labeling, model configuration, or other aspects of the update process can adversely affect the algorithm's performance. Finally, one of the primary advantages of machine learning algorithms is their potential for autonomy and efficiency. Requiring regular manual updates undermines this autonomy, making the system more dependent on external intervention. For the widespread adoption of machine learning-based seizure detection systems, scalability is essential. Manual updates by medical professionals are not scalable, especially when considering large patient populations or multiple healthcare facilities.



**Fig. 6.** Machine learning models need to be regularly updated to maintain high accuracy, especially if the signal patterns change over time. Updating such models takes time as EEG needs to be recorded first, then processed, and used for machine learning training.

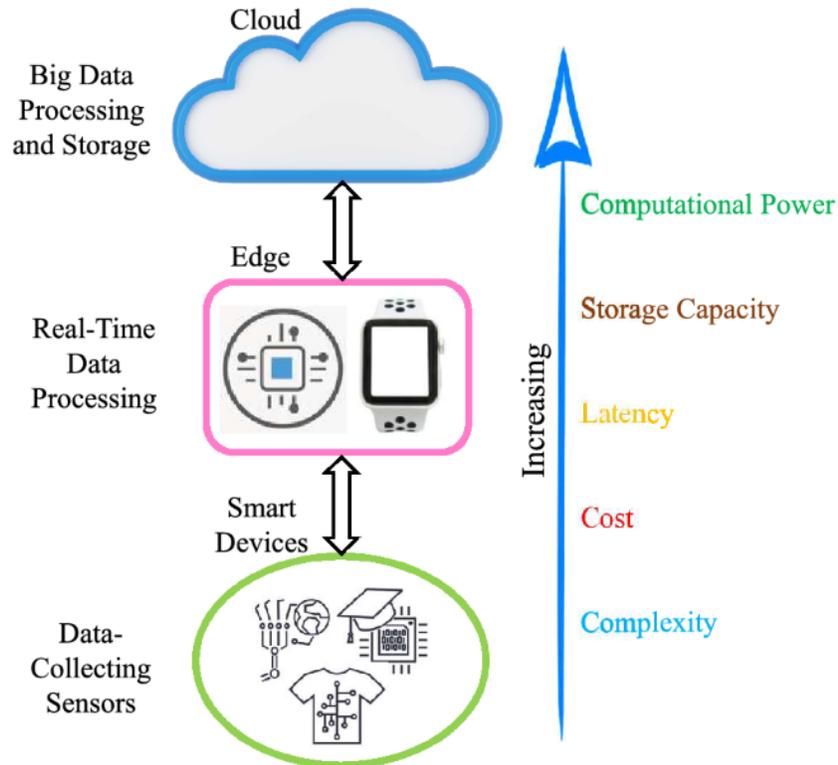
(Image credit: <https://towardsdatascience.com/disability-diagnostic-using-eeg-time-series-analysis-8035300837ac>)

### Challenge 3: The trend of increasing computational complexity

Machine learning algorithms, especially when aiming for high accuracy over large datasets (such as datasets comprising several hours of EEG recordings on multiple different patients), often trend towards increasing computational complexity. Fig. 7 illustrates where machine learning can take place for these types of systems. While complex models can capture intricate patterns and nuances in the data, they come with their own set of challenges, such as increased latency and cost of operation. Complex algorithms, especially deep learning or neural network-based models, require significant memory and computational resources and are typically relegated to cloud processing. Edge devices, on the other hand, are much more constrained in terms of computational complexity but allow for faster response times and lower operational costs. Edge devices are better suited for seizure detection systems so that immediate action can be done, such as neurostimulation or an in-brain drug delivery to suppress the seizures as they occur. For edge devices such as an implantable system, battery life is a primary concern. Increasing complexity of these devices would increase power consumption. Frequent recharging or battery replacements are not only inconvenient but can also pose risks and discomfort to the patient.

Prior art in seizure prediction and detection utilized long-term datasets to capture such variations [12], resulting in seizure detection accuracy greater than 90%. However, the classifier algorithms in those works were software-only implementations, where computational complexity and memory requirements were not a design consideration. For an at-the-edge, closed-loop, seizure detection systems, energy efficiency, area utilization, and long-term accuracy become important

design constraints. Balancing accuracy with simplicity, power efficiency, and adaptability is crucial. This work will address the three challenges highlighted earlier by developing an unsupervised online learning framework, that can be made energy-efficient, to dynamically adapt to changes in neural signal patterns over time and maintain high detection accuracy without external intervention.



**Fig. 7.** Edge devices allow for faster response times in exchange for reduced complexity. Cloud-based processing provides more computational power but increases latency.

(Image credit: <https://link.springer.com/article/10.1007/s42979-020-00272-2>)

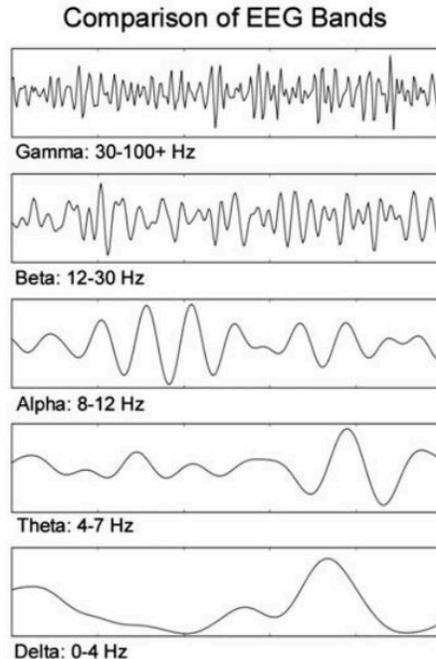
The remainder of this thesis is organized as follows. Chapter II reviews prior work on on-chip machine learning primarily for seizure detection, highlighting the type of classifiers used and the resulting energy efficiency, accuracy measurements, and area consumption. Chapter III introduces the unsupervised online learning framework, which will serve as the main innovation introduced in this work. Chapter IV describes the on-chip seizure detection architecture that leverages the online learning framework so that it maintains high accuracy over time while being very energy efficient. Chapter V extends the seizure detection hardware to support seizure prediction, which would be the first on-chip unsupervised online learning seizure predictor in the literature. Chapter VI showcases the experimental results of both seizure detector and predictor in terms of accuracy, highlighting the capability of the online learning scheme. Chapter VII provides additional discussion on the key findings and contributions of this work. Chapter VIII presents some ideas for potential future research direction that leverages the online learning framework which can be further explored. Chapter IX concludes the document.

## II. REVIEW OF RELATED WORK

When implementing an on-chip classifier, the memory and hardware requirements for machine learning need to be factored in. The resulting power and area needed to integrate these on-chip classifiers will limit the number of channels that the system can support, thereby reducing the recording granularity on a given power budget. Extending the edge device battery life would also be a good motivation for the need for low-power, energy-efficient systems. The machine learning power consumption is typically determined by the complexity of the feature extraction unit and the type of classifier being used [13].

### A. Feature extraction units

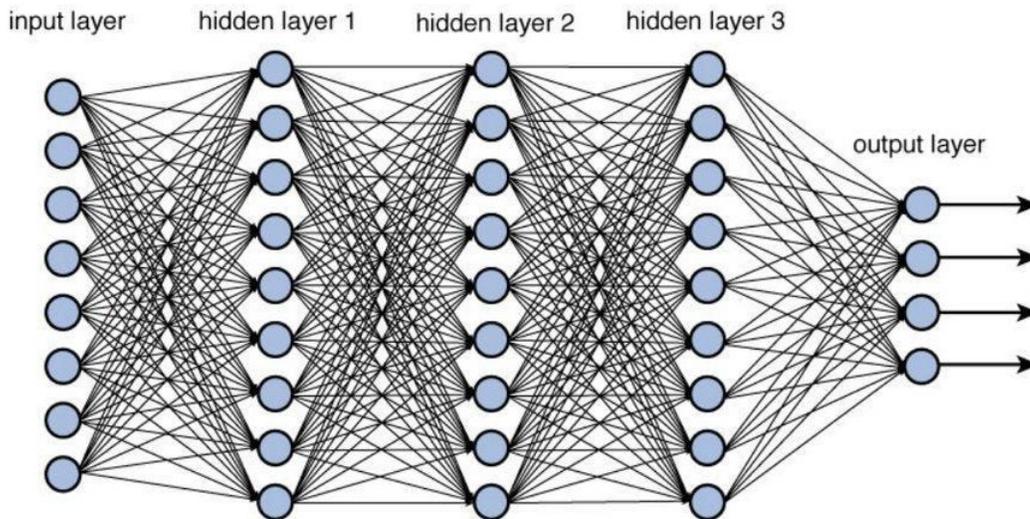
Feature extraction is the process of transforming raw data into another form that can highlight some inherent time or frequency domain characteristics which can help a classifier to differentiate between classes. A common method of feature extraction is through input data filtering so that the low and high-frequency characteristics of a signal can be segregated. EEG-based signals are typically subdivided into various groups [14], as illustrated in Fig. 8. Depending on the bandpass filter specifications and chosen topology, implementing them in hardware can be costly due to the multiply-and-accumulate operations. Various methods of reducing filter hardware overhead have been explored, such as time-division multiplexing [15], where different channels use the same filter hardware, and frequency-time division multiplexing [16], which improves on the first one by multiplexing the filter coefficients as well to reduce the filter multiplier hardware.



**Fig. 8.** Different frequency bands typically used when analyzing EEG signals. (Image credit: <https://neurosky.com/2015/05/greek-alphabet-soup-making-sense-of-eeb-bands/>)

## B. Neural network-based classifiers

Classification complexity also matters when implementing machine learning algorithms on-chip. Neural networks are one of the popular machine learning classifiers in the field of artificial intelligence due to the significantly high accuracies that they can achieve. It utilizes a network of parameters, typically hundreds, connected into different layers, which are then multiplied and added together to perform classification [17]. This type of classifier is widely used in software-based applications but is rarely considered in the context of on-chip classification due to the hardware-intensive implementation, as illustrated in Fig. 9, requiring large amounts of memory to store the parameters and the parallel multipliers and adders needed to combine them. The work from [18] had an energy efficiency of 1.24 mJ/classification and consumes 31.25 mm<sup>2</sup> of area. It will be seen later how these numbers are significantly larger compared to more energy-efficient approaches.

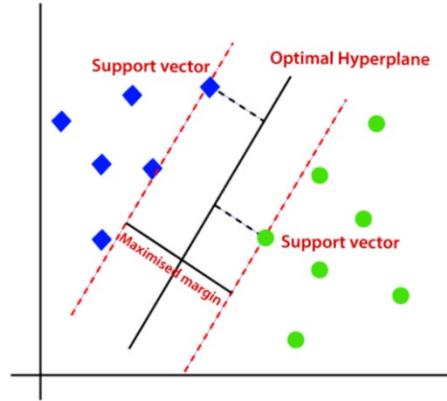


**Fig. 9.** A representative neural network topology. Each circle represents a parameter, each line represents multiplication, and lines converging to a point represents an addition.

(Image credit: <https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964>)

## C. SVM-based classifiers

The most common on-chip seizure event classifiers typically employ support vector machines (SVMs) due to their high accuracy and relatively simple implementations. As a classifier, an SVM attempts to create a dividing line (or a hyperplane in n-dimensions) that separates the two classes together [19], as shown in Fig. 10. The points where the margin lands on are called the support vectors. The more complex the classification task is (in higher dimensions), the more support vectors are needed to define the dividing hyperplane.



**Fig. 10.** SVM classification demonstration between two classes.

(Image credit: <https://medium.com/@viveksalunkhe80/support-vector-machine-svm-88f360ff5f38>)

The general classification function for an SVM is as follows:

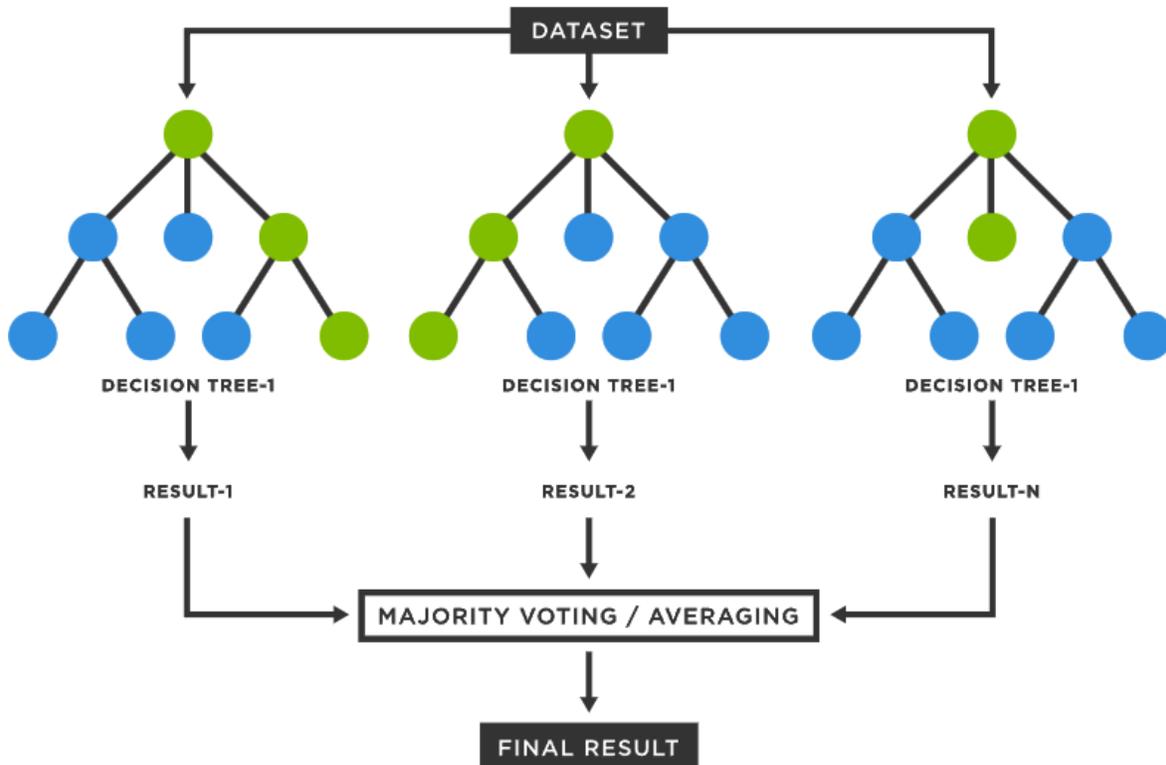
$$f(x) = \sum_{i=1}^{N_{sv}} a_i K(sv_t + x_t) + b$$

Where  $x_t$  is a vector of feature values,  $sv_t$  is one of the  $N_{sv}$  support vectors,  $K$  is the kernel function,  $a$  and  $b$  are modeling parameters. The kernel function allows for the dot product to be translated into a higher dimensional space. This enables classification even with non-linearly separable data.

The SVM complexity can vary depending on what kernel is being used [19]. Typical SVM-based classifiers in the literature [16, 20-24], for instance, utilize linear kernels as they are the simplest and most computationally efficient. The hardware complexity (which is generally measured as the number of multiplications and additions) for these linear kernels is just proportional to the number of features (the length of the vector  $x_t$ ). However, using such kernels may limit the overall accuracy that can be achieved if the features are not linearly separable. Alternatively, more complex non-linear SVM kernels can be employed to dramatically increase the maximum achievable accuracy. A popular kernel of this type is the Radial Basis Function (RBF) kernel, also known as a Gaussian SVM, which aims to translate the non-linearly separable data into a much higher dimensional space where they become linearly separable. However, while this can be a powerful option for SVM classification, it trades off computational complexity for accuracy. Specifically, using an RBF kernel would scale the hardware complexity proportional to the number of support vectors ( $N_{sv}$ ) multiplied by the number of features (the length of the vector  $x_t$ ). As a complex classification task may require hundreds of support vectors, utilizing complex kernels such as RBF might not be feasible for on-chip classification. Typical SVM-based seizure detection classifiers would end up having significant memory requirements to hold the support vectors needed for on-chip classification. At least 64 kB [16, 20-24] of memory is required leading to high on-chip area and power consumption. The work from [23] utilized a non-linear SVM and had an energy efficiency of 170  $\mu\text{J}/\text{classification}$  at 4.5 $\text{mm}^2$  area consumption. On the other hand, the work from [24] utilized a simpler 2<sup>nd</sup> order polynomial kernel for the SVM leading to an energy efficiency of 680  $\text{nJ}/\text{classification}$  and consuming 2.25 $\text{mm}^2$  of area.

### D. Decision-tree based classifiers

Decision-tree-based classifiers are among the most energy-efficient systems in the literature [13,25], achieving  $<50$  nJ/classification, which is an order of magnitude more efficient than the SVM-based designs cited earlier. They are also generally smaller, consuming only  $1\text{mm}^2$  [x-x] of area. Decision trees (also called Random Forests, shown in Fig. 11, when considering several decision trees running in parallel) achieve an energy-efficient design due to their simple comparator-based nature, as you only need to compare against thresholds (which are stored in memory), without the need for multiplications (contrary to neural networks and SVMs) [26]. The problem with these types of classifiers is that they tend to easily overfit the training data and can be unstable with small changes in the input. Instability refers to the fact that the model parameters (in this case, the thresholds, and the tree structures) will vary significantly when a little bit of noise is added to the inputs. This makes it difficult to get an interpretable and replicable result after training. Decision trees are hard to dynamically retrain since the process involves random sampling of data from a collection of training points (implying that all training points need to be stored in memory) [27]. It does not have a differentiable loss function that can be optimized for retraining purposes.



**Fig. 11.** Random forest classification is a collection of parallel independent decision trees. (<https://www.tibco.com/reference-center/what-is-a-random-forest>)

### *E. Online learning classifiers*

Another potential problem with conventional on-chip seizure detectors is that, while they usually incorporate on-chip feature calculations, training, and its associated computational complexity is usually completely offloaded to software. The datasets used to test such seizure detectors are usually too short such that long-term EEG signal drifts do not significantly affect the classifier accuracy. For a seizure detection system to remain accurate over long periods of time on patients with changing seizure patterns, regular signal post-processing, labeling, and retraining by an expert physician would be required. Such external intervention can also be costly and impractical. Recognizing this, a growing trend in seizure detection classifier design for the past three years is incorporating some kind of dynamic adaptation to the classifier.

The work in [23] implemented an SVM tuning based on the Alternating Direction Method of Multipliers (ADMM). ADMM is a technique that partitions a convex optimization problem into several smaller sub-problems so that the weight coefficients can be updated in a parallel manner [28]. This optimization algorithm is backed by the theory of computing gradients to optimize for the classifier's loss function [29] making it the mathematically correct way of retraining. The process involves matrix inversions where the matrix dimensions scale with the training data. The work optimized the tuning algorithm through feature selection and matrix rank approximation so that it can be implemented on the chip. The tuning algorithm improved the sensitivity by 1.2% and reduced the false alarm rate by 36% across 24 subjects. It ended up with a 170.9 uJ/classification energy efficiency. On the other hand, the work in [24] greatly simplified the SVM tuning by adding support vectors from a pre-trained set. It utilizes the input features that caused false positives and false negatives, normalizes them, and uses those values as additional support vectors for the classifier. The additional support vectors will change the decision boundary of the SVM. This approach is not an optimization algorithm based on a mathematical theory, compared to the previous work which did. Nevertheless, the work demonstrated a 1.8x improvement in classifier accuracy on a single test subject. Due to the simplicity of the training algorithm, the entire online learning classifier had an energy efficiency of 680 nJ/classification, which is 250x more efficient than the previous ADMM-based SVM retraining.

Table I summarizes all the work that has been cited so far so that the energy efficiency and classifier area can be compared. Neural networks tend to be the least efficient both in terms of area and energy efficiency among all the classifiers, which is expected given the explanations provided earlier in this chapter. Decision tree-based classifiers tend to be the best both in terms of area and energy efficiency. As explained previously, decision trees are hard to retrain online since all the parameters change with every new data. Online tuning was only featured in SVM-based classifiers, which were in the middle ground in terms of the metrics being used.

TABLE I  
COMPARISON OF ON-CHIP CLASSIFIERS FEATURING DIFFERENT CLASSIFIER TYPES

	<b>TBIOCAS 2022<sup>[18]</sup></b>	<b>JSSC 2020<sup>[23]</sup></b>	<b>JSSC 2022<sup>[24]</sup></b>	<b>JETCAS 2018<sup>[13]</sup></b>	<b>ISSCC 2020<sup>[25]</sup></b>
<b>Classifier Type</b>	CNN	SVM	SVM	Decision Trees	Decision Trees
<b>Energy Efficiency</b>	1.29 mJ/cls	170.9 uJ/cls	680 nJ/cls	41.2 nJ/cls	36 nJ/cls
<b>Classifier Area</b>	31.25 mm <sup>2</sup>	4.5 mm <sup>2</sup>	2.25 mm <sup>2</sup>	1 mm <sup>2</sup>	1 mm <sup>2</sup>
<b>Online Learning</b>	X	O	O	X	X

Among all the classifiers presented so far, an SVM-based classifier tends to show some promise, especially because online tuning methods are already being explored. However, only the work in [23] had the correct mathematical background of computing gradients to ensure that the SVM is being retrained to a better optimal point. The work in [24], while being significantly more energy efficient, had an online tuning scheme that is not entirely backed by theory and only proved the effectiveness of its dynamic adaptation on a single patient. A better approach, therefore, is to find a classifier type that can be retrained based on an appropriate mathematical/statistical theory (to ensure correct model optimization over time) while being computationally simple, so that energy efficiency and long-term high accuracy can be both achieved.

### III. ALGORITHM DESIGN

As highlighted in Chapter I, EEG signal power can drift over long periods of time. On-chip state-of-the-art classifiers do not usually see this problem as the datasets that they are working with tend to be short (several hours). For longer datasets lasting several days, these signal drifts can negatively affect the classifier accuracy. Chapter II has shown that there is now a growing interest in dynamically adapting classifiers so that they can track such long-term signal drifts to maintain high accuracies over time. Online retraining SVMs have been implemented on-chip, although only one has implemented a retraining method based on calculating the gradient [23]. This is an important consideration since this ensures that the classifier is always being retrained towards optimality over long periods of time [28]. The challenge now is to find a more energy-efficient approach to classification that can also be further optimized (i.e. retrained) through gradient calculation. After neural networks and SVMs, there is only one other classifier that can be retrained through a gradient descent method and is a good candidate for the target system. For this chapter, we will explore a binary classifier based on a generalized linear model, logistic regression.

#### *A. Logistic regression as a classifier*

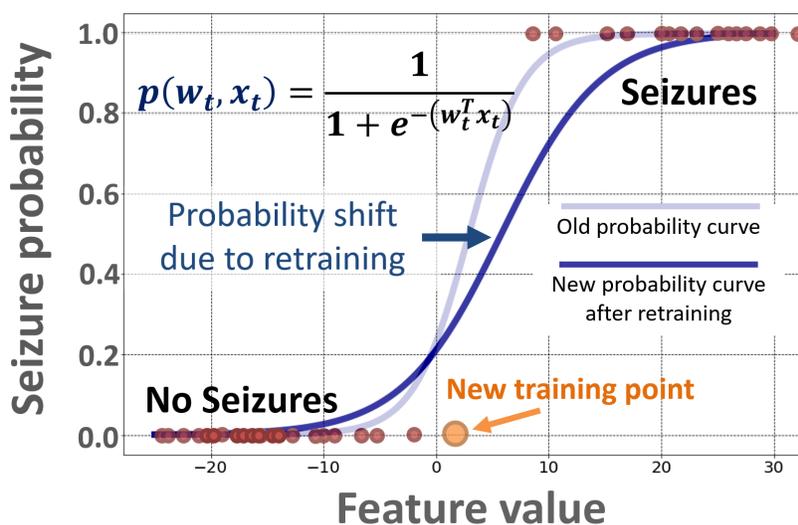
Logistic regression is a probabilistic model that utilizes the logistic function to map the weighted linear combination of input features to real values between 0 and 1, which can then be interpreted as probabilities. Thresholding the output to any value between 0 and 1 (typically 0.5), would result in binary classification [30]. The standard logistic function is shown in (1):

$$p(w_t, x_t) = \frac{1}{1 + e^{-w_t^T x_t}} \quad (1)$$

The  $w_t$  term refers to the vector of logistic regression feature weights corresponding to the vector of feature inputs  $x_t$ . The values for the weights in  $w_t$  are calculated through an iterative process to best fit the logistic function on the labeled set of feature inputs. That iterative process is how the logistic regression classifier is trained and is typically done in software as it uses the gradient descent algorithm for the optimization.

Since the output of logistic regression depends on the linear combination of weights and features, it performs very well on linearly separable data. As seizure and non-seizure events usually exhibit this property (especially using features that can detect the high-amplitude and high-frequency seizure signals, more on this later in Chapter IV), logistic regression can be used as a seizure event classifier. Prior work has compared logistic regression against other classifiers [31-33] for this application and has shown comparable performance. However, when the feature values between seizures and non-seizures vary over time, linear separability between the two classes cannot be maintained, leading to accuracy degradation [30]. SVMs, on the other hand, can utilize non-linear kernel functions to force class separability leading to better accuracy [19]. This is the reason why state-of-the-art seizure classifiers typically use SVMs.

The limitation of logistic regression on diminishing linear separability can be mitigated if logistic regression can track feature value changes over time. As the feature values  $x_t$  drift, the optimal feature weights  $w_t$  that was calculated during offline training might not hold true anymore. Thus, if a new set of weights can be calculated beyond the initial offline training period (i.e. online), the logistic function can shift dynamically, to maintain optimality. Fig. 12 illustrates this function shift on a one-dimensional feature example. As a new training point is introduced, the curve shifts to the right to ensure that the new point is properly classified.



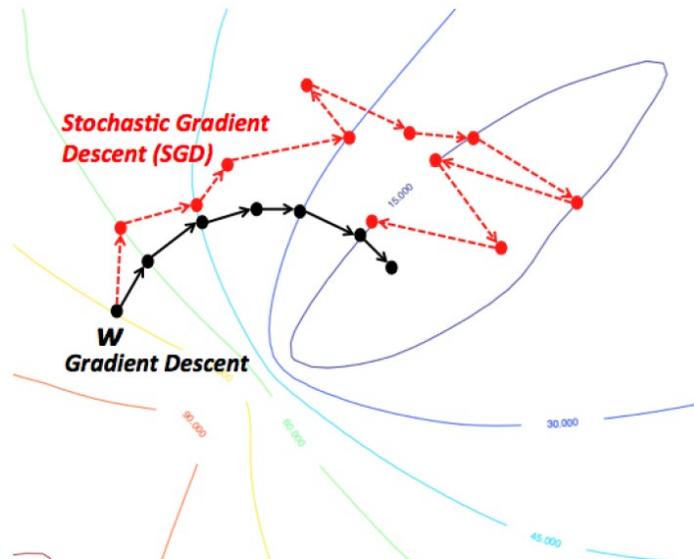
**Fig. 12.** Logistic function shift towards a new optimal curve due to updated feature weights after logistic regression retraining. The function shifts to the right (darker line) due to the introduction of a new training point (enlarged circle).

Note that Fig. 12 demonstrates the logistic regression probability shift on a single feature dimension (x-axis). However, for the classification tasks that will be implemented in the later chapters of this thesis, the total number of features needed for a single classification is more than 1. In this case, the logistic function is plotted on a  $n$ -dimensional space, where  $n$  is the total number of features. The logistic function shift corresponds to a calculation of a new vector of  $n$  feature weights  $w_t$  that will change the classifier's decision boundary in the  $n$ -dimensional space.

### *B. Stochastic gradient descent for logistic regression*

As logistic regression has a differentiable loss function, it means that it can be retrained through gradient descent, an iterative method to search for the minimum point of a function. If the loss function is minimized, the classifier achieves the highest classification accuracy given the points that it was trained on [30]. However, performing a conventional gradient descent requires the whole training data to be processed simultaneously, also called batch gradient descent. While this process will give the best accuracy improvement for the classifier, it requires that all training data must be saved in the memory and that all training data must be factored in the calculation. This increases both memory requirements and computational complexity, which is not practical. However, utilizing stochastic gradient descent instead can be further explored.

The stochastic gradient descent (SGD) algorithm is an iterative method of optimizing the classifier feature weights by approximating the calculation of the gradient descent using a new set of feature inputs [34,35]. This algorithm avoids the complex computation of the gradient on the whole training data. Fig. 13 illustrates how the stochastic gradient descent differs from the conventional gradient descent algorithm in terms of optimization. Gradient descent ensures that for every iteration, the direction of the step toward optimization is always correct. On the other hand, since SGD is just an approximation of the gradient descent, it only takes an approximate step toward where it thinks the correct direction toward optimality is. Indeed, SGD can make the classifier less optimal than it should be. However, if the learning rate (i.e. the step size) is tuned correctly, then it will not diverge too much away from optimality [35]. Gradient descent is faster in reaching the new optimal point but trades off the huge memory requirements and computational complexity. SGD only considers a single point as its training data on every iteration, making the updates feasible to be done in real-time, while being simple and efficient.



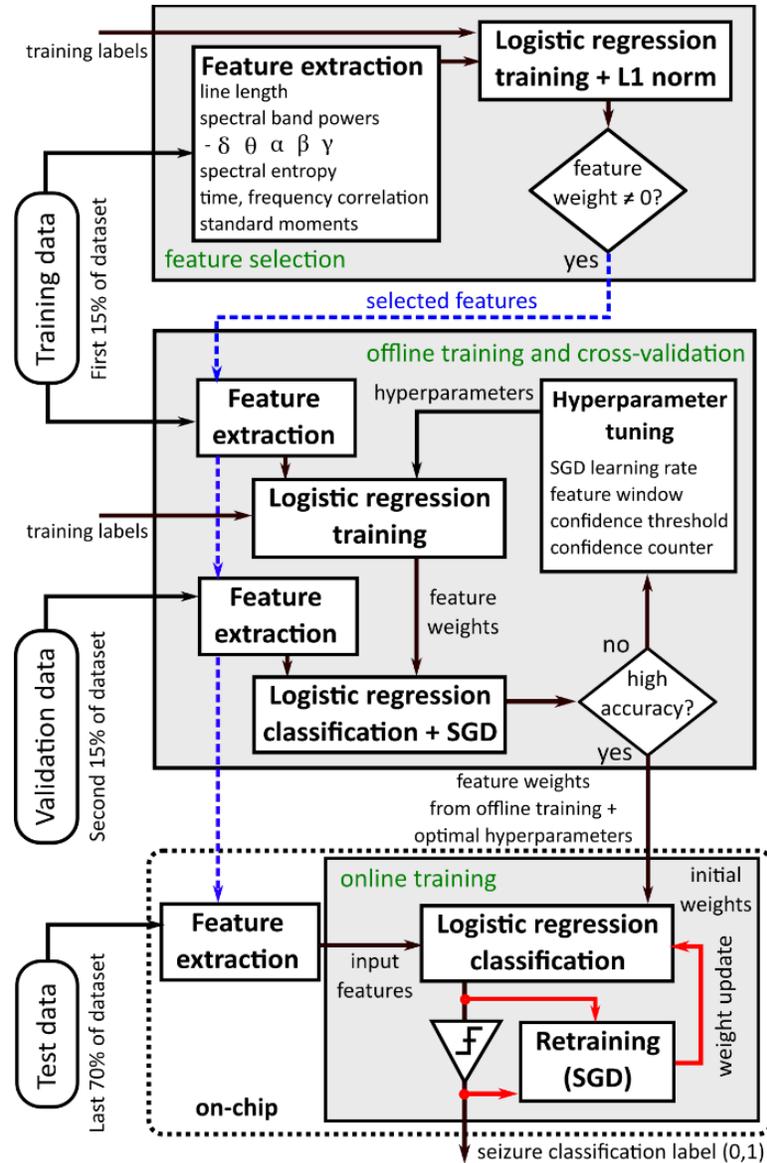
**Fig. 13.** Conventional gradient descent versus stochastic gradient descent on reaching optimality. (Image credit: <https://www.samvitjain.com/blog/gradient-descent/>)

The logistic regression weight update using SGD is computationally simple as shown in (2). The derivation for this weight update formula is available in the Appendix section.

$$w_{t+1} = w_t + \eta(y_t - p(w_t, x_t))x_t \quad (2)$$

The  $w_{t+1}$  term refers to the next set of feature weights after the update;  $\eta$  is the learning rate of the algorithm, which controls how much the feature weights will change based on new data; and  $y_t$  is the corresponding label for the current feature input. The SGD-based feature weight update can be done in a single iteration with minimal hardware. The update is also done in one epoch (i.e. one-shot retraining on the new data) as the classifier does not save the previous training points to minimize the memory requirements. The logistic function calculation can also be implemented using a look-up table to further reduce the computational complexity. Architectural optimizations will be covered further in Chapter IV.

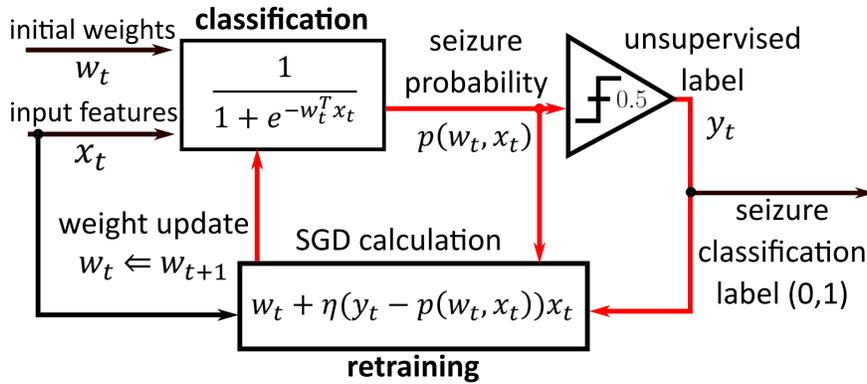
While SGD is an approximation, it can be used to dynamically update the feature weights online through a defined optimization algorithm. Fig. 14 describes the procedure. A set of feature weights are initially trained offline, using any software-based training algorithm available, to achieve the best possible accuracy from the training data. This process provides a good baseline for logistic regression classification. Cross-validation is performed by running the classification and SGD on validation data without external labels. This is where different hyperparameters are tuned to maximize accuracy during the unsupervised online learning phase. Then, upon classifier deployment, the classifier can utilize the test data to update the feature weights using SGD on the chip.



**Fig. 14.** Feature selection, offline training and cross-validation, and online (on-chip) retraining scheme. Feature selection reduces the features to be extracted through L1-norm regularization. Offline training phase generates the best possible set of starting feature weights and hyperparameter values for on-chip classification. Feature weights are dynamically updated on chip using SGD (highlighted in red).

### C. Enabling unsupervised learning

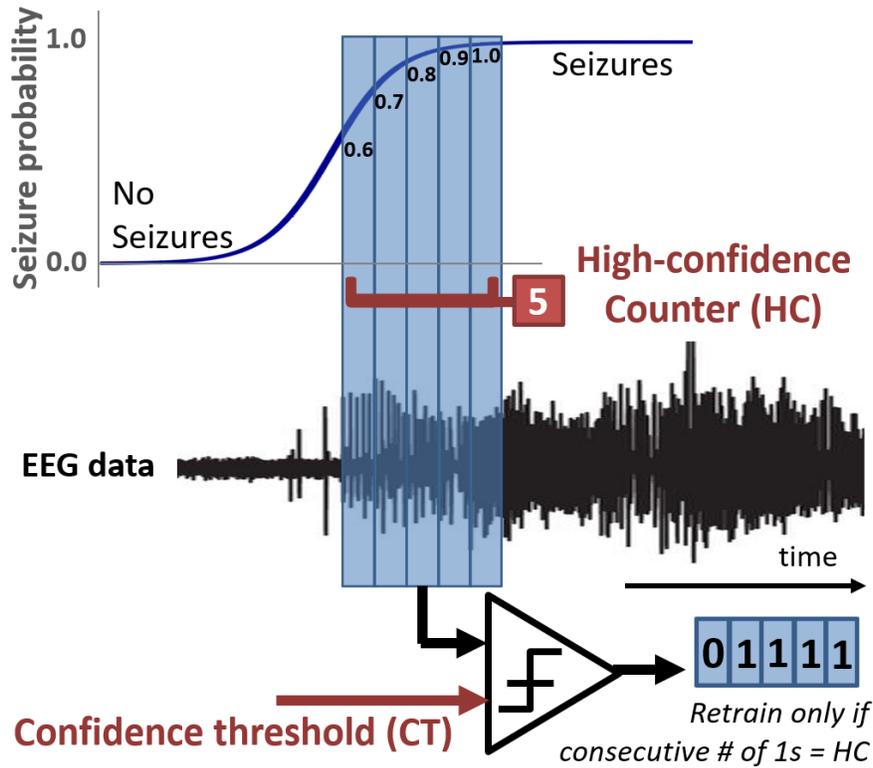
Traditionally, SGD is meant for supervised learning, where an external label is provided for every data input [34,35]. However, for an implantable system operating *in situ*, externally provided labels are not readily available. Thus, this approach places SGD within an unsupervised learning paradigm during the online classification phase. This is implemented through bootstrapping, which uses the classifier’s predicted probability output to update its own feature weights. The classifier’s output probability  $p(w_t, x_t)$  is rounded to either 0 or 1 and is then treated as a label  $y_t$  for SGD. This creates a positive feedback path between the classifier’s output and its input training label, highlighted in Fig. 15. Consequently, the cumulative accuracy over time is heavily dependent on the initial classifier accuracy after the offline training phase. It is critical that the initial logistic regression weights achieve a high classification accuracy during the training phase. The feature set used for this work, which will be described later in this section, adequately separates seizure and non-seizure events. Therefore, achieving high classification accuracy, at least during the training phase, is possible. The unsupervised online learning classifier will be tracking the long-term changes in these seizure and non-seizure patterns through feature weight updates *in situ*.



**Fig. 15.** Feedback loop when using the classifier’s own output probability (rounded off to 0 or 1) as the training label for SGD.

### D. Making the unsupervised online learning robust

While high classification accuracy is required for offline training, the classifier can still make occasional errors. Generally, any misclassification can degrade the accuracy due to positive feedback, as the classifier will retrain in the wrong direction. To avoid such an occurrence, the weights are only updated once a specified confidence threshold is reached by the logistic function output. Moreover, a series of high-confidence predictions are required to trigger the online feature weight update, shown in Fig. 16. Only the last set of features after the consecutive high-confidence predictions would then be used as new data for the SGD algorithm on the next weight update. This process ensures that short-term misclassifications and glitches will not negatively affect the update. The confidence threshold (CT) and high-confidence counter (HC) become additional hyperparameters during the offline training phase (Fig. 14) and are tuned on a patient-specific basis.

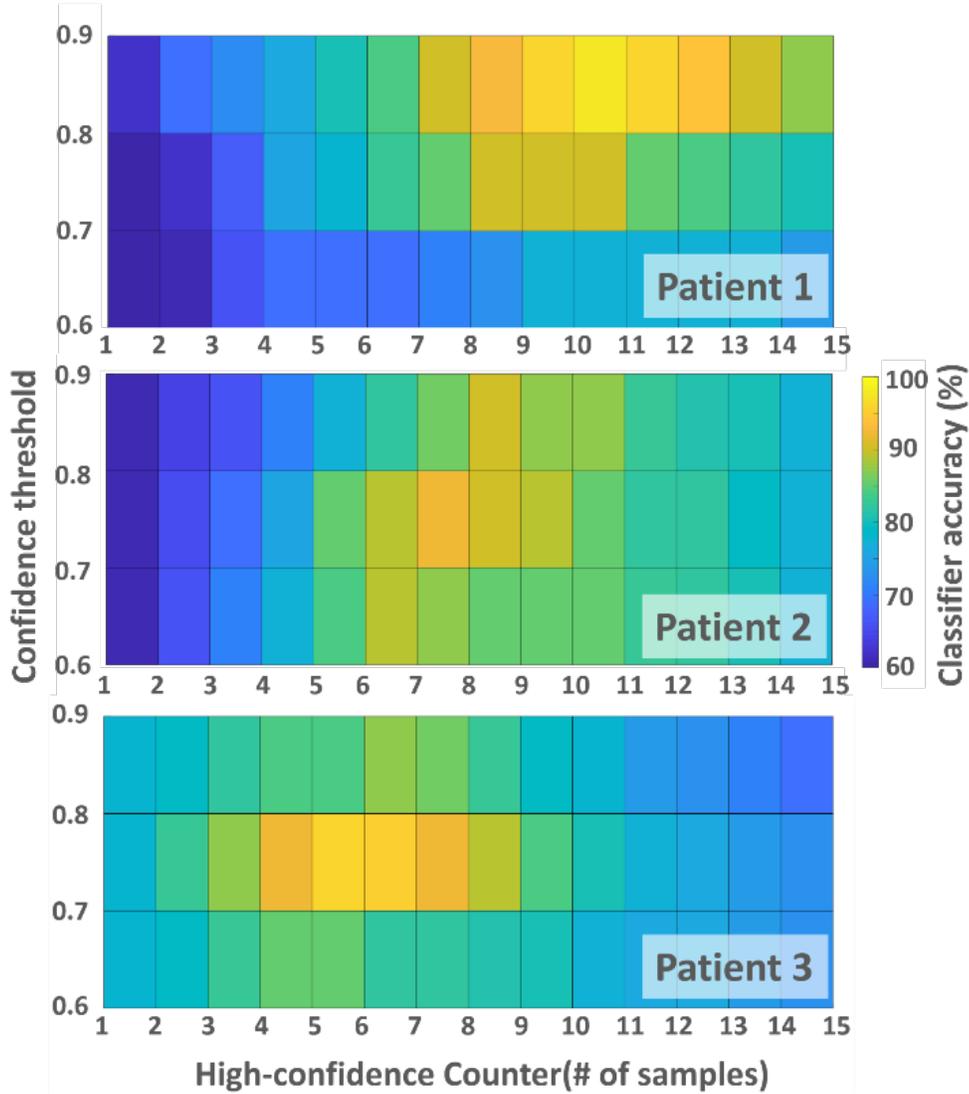


**Fig. 16.** Confidence thresholding technique implemented to only train the classifier once a series of high-confidence predictions are generated from the logistic regression output.

### E. Tuning the online learning hyperparameters

As stated in the previous section, the HC and CT hyperparameters are tuned on a patient-specific basis. The values of these hyperparameters depend on both the short-term and long-term variability of the EEG signals per patient. Noisy EEG signals require higher confidence thresholds. Long-term time-varying signals require shorter HC so that the classifier can track signal changes faster. Fig. 17 shows the achieved accuracies of the classifier during the hyperparameter tuning on three patients from the iEEG dataset [11]. A full description of the dataset will be provided in Chapter VI. HC, measured in terms of the number of samples (each sample corresponds to one complete feature window which is 1 ms), was swept from 1 to 15. CT, which thresholds the logistic function output, was swept from 0.6 to 0.9.

Figure 17 shows that for Patient 1, the optimal hyperparameter values are  $CT = 0.8$  and  $HC = 10$ . The high CT value implies that the EEG signal is relatively noisy. Thus, the threshold needs to be high to avoid misclassifications negatively affecting the online training process. The optimal HC is also high to further mitigate the noise. Patient 3, on the other hand, has low hyperparameter values ( $CT = 0.7$ ,  $HC = 5$ ). These imply that the EEG signal is less noisy (lower CT) and that the signal varies over the long term (lower HC to stay on track). Fig. 17 also shows that without patient-specific tuning, a common value for the hyperparameters ( $CT = 0.7$ ,  $HC = 7$ ) can be used instead for these three patients, albeit with maximum sensitivities only reaching about 90%.

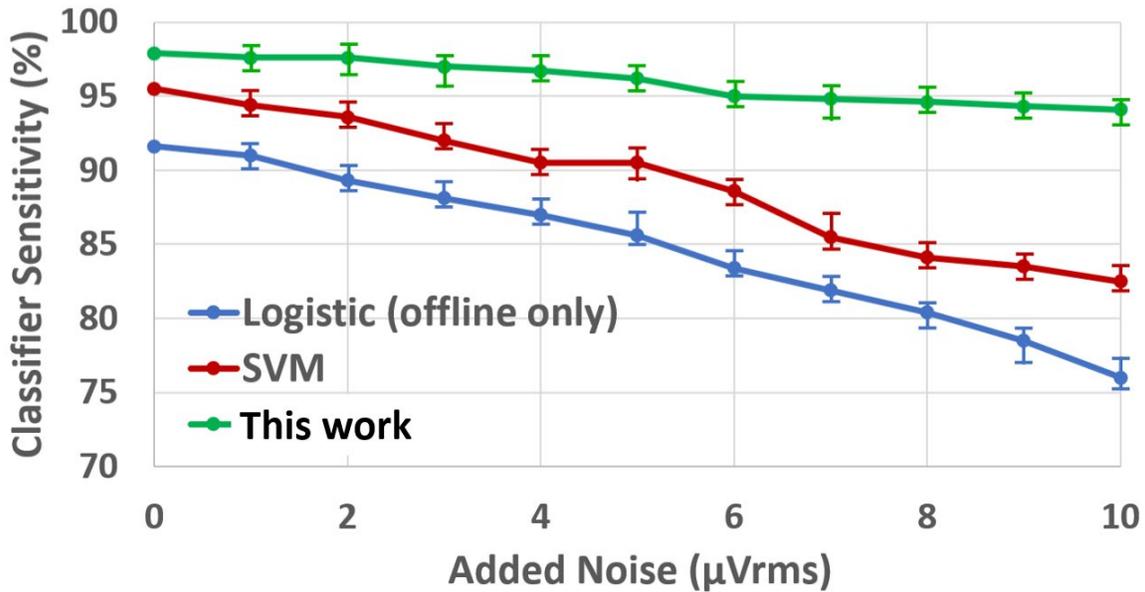


**Fig. 17.** The classifier accuracies (z-axis) during the tuning phase of the two hyperparameters: confidence threshold and high-confidence counter. Different patients have different optimal values.

### *F. Classifier stability*

The unsupervised online learning framework was also tested for classification stability, which measures how a machine learning algorithm performs when the dataset is perturbed by noise. The accuracy of a stable classifier does not significantly change after perturbation since it should be able to generalize and not overfit on a given dataset. Artificial white noise was added to the same dataset used in the previous section. The standard deviation of the noise was swept from 1 to 10  $\mu\text{Vrms}$  at 1  $\mu\text{Vrms}$  increment. At 10  $\mu\text{Vrms}$ , the added noise is comparable with the average biological noise measured during the non-seizure segments for each dataset. For each level, ten training and classification runs were performed to average out the effects of noise on the classification accuracy.

Fig. 18 shows the performance of the online learning framework when noise was added to the long-term EEG data. This is then compared against the other classifiers as a comparison point. The average sensitivity for the three patients was plotted. The sensitivity values that are plotted reflect the final values after the entire dataset has been run through the classifier. The sensitivity values vary within 1-2% from the mean at every noise level, shown as error bars in Fig. 18. It can be observed that while all classifier accuracies degrade as the noise level increases, the average sensitivity of logistic regression with the online learning enabled degrades much slower. At the maximum noise level, the sensitivity decreased by only  $\sim 4\%$  allowing it to achieve 11.6% better sensitivity than a representative SVM. This demonstrates both classification stability and the feasibility of the online learning scheme even with added noise on the dataset.

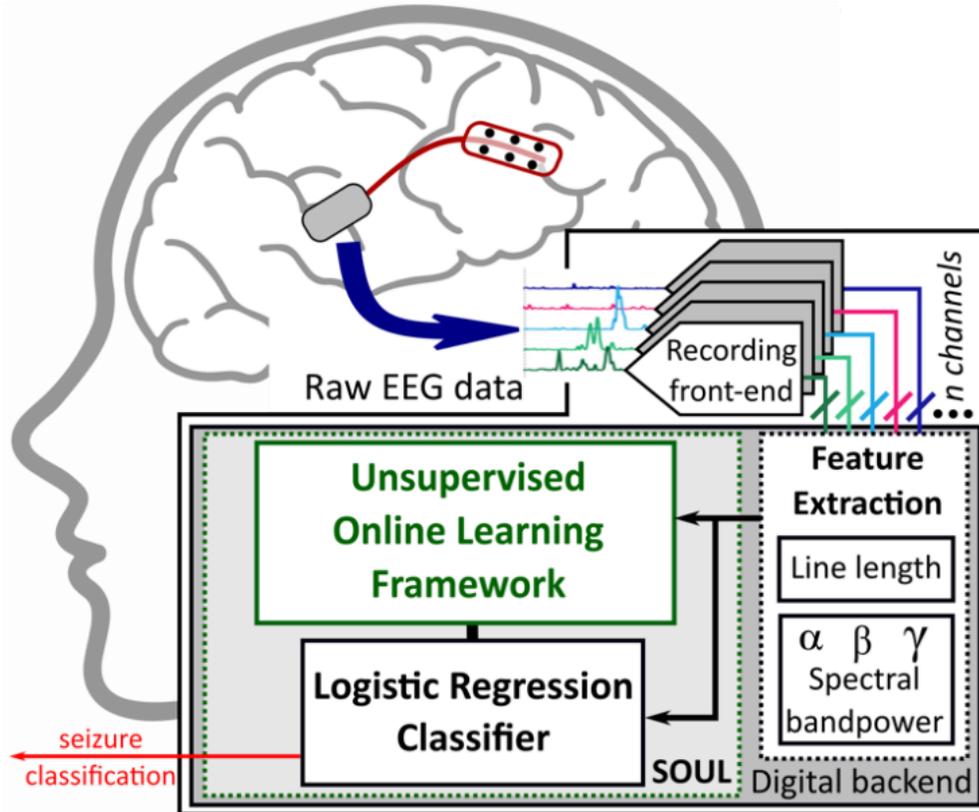


**Fig. 18.** Classifier sensitivities at increasing noise levels on the long-term EEG dataset, averaged per patient. Error bars represent the min and max values per run (ten runs per noise level).

It is worth noting that all classifiers started off with much higher accuracies during testing (not shown in the figure), capable of detecting seizures early on in the test set. However, as more test inputs are fed to the classifiers, the effect of the added noise coupled with shifting seizure patterns over time becomes much more prominent, blurring the difference between seizure and non-seizure events. Consequently, the accuracy degrades more as the noise level increases. However, the online learning framework enables self-correction. Since the classification started off with high accuracies, it was able to track the changing patterns in the EEG signal, and the effect of noise is averaged out as the retraining process is done over time. If the noise levels increase to the point that classifiers already struggle to maintain high accuracies at the start of the test, then the online learning framework will not be able to fix the problem since the retraining process will not give any useful information in improving the logistic regression classifier (especially since it is unsupervised). Therefore, it is paramount that logistic regression should be able to achieve high ( $\sim 90\%$ ) accuracies at the start of the test set so that it has a good enough starting point to be able to learn on its own.

## IV. ON-CHIP SEIZURE DETECTION

To verify the feasibility of the unsupervised online learning framework presented in the previous section, we demonstrate the use of this framework on an on-chip seizure detection system. This system would dynamically adapt to changes in neural signal patterns over time and maintain high detection accuracy without external intervention. This will be referred to as **SOUL (Stochastic-gradient-descent-based Online Unsupervised Logistic regression classifier)** [36,37], shown in Fig. 19. SOUL is initially trained offline and then feature weights are updated *in situ*. Moreover, due to the computationally simple algorithm and architectural optimizations used, SOUL is significantly more energy efficient than state-of-the-art on-chip seizure detectors.



**Fig. 19.** Proposed seizure detection system featuring a fully unsupervised online learning framework to maintain long-term high accuracy detection.

### A. Classifier features description

The feature extraction unit computes two main feature classes (Fig. 19): line length and spectral band powers for three frequency bands. These features are commonly used in seizure detection systems since they capture amplitude and frequency-dependent patterns usually attributed to seizure events. Other features were also considered, such as spectral entropy and time/frequency correlations, but were down-selected after running the initial training with L1-

norm regularization (Fig. 14), which zeroed out most of these features. The same L1-norm penalization was done to remove highly correlated channels which eventually lead to the current 8 channels supported by the classifier. After the channel and feature selection processes, the classification accuracies only decreased by <2% on average relative to the accuracies on the original set of features on all channels included in the datasets.

Line length [38] captures the high amplitude and high-frequency data characteristic of seizures, defined by the sum of the absolute value of differences between consecutive points, as shown in (3).

$$\sum_{t=1}^{N-1} |x_t - x_{t-1}| \quad (3)$$

Spectral band power captures frequency-dependent patterns, calculated by summing the spectral power over a specific frequency band. This feature has been shown to separate seizure and non-seizure events very well [13]. This can also be approximated by passing the signal through a bandpass filter on a specified frequency range and then performing a sum of squares, exploiting Parseval's theorem, as shown in (4). This approximation eliminates the need for dedicated Fast Fourier Transform (FFT) hardware in the system.

$$\sum_{t=0}^{N-1} x_t^2 = \frac{1}{N} \sum_{f=0}^{N-1} |X_f|^2 \quad (4)$$

The spectral band power is calculated for three EEG frequency bands:  $\alpha$ , 8-16 Hz;  $\beta$ , 16-32 Hz;  $\gamma$ , 32-96 Hz. Spectral power for the lower frequency bands were removed after the feature selection process described previously.

Both line length and spectral band power features require a specific sample window  $N$ . For this work, a 0.1-second window was used, which translates to a 100-sample sliding window for a 1 kHz input sampling rate. This feature window size was determined from the offline training phase (Fig. 14), as part of the hyperparameters that were optimized. This window controls how much input signal noise is smoothed out during the feature extraction process, which tends to dampen the feature value response due to averaging. However, the response to sudden signal transitions (which can be indicative of seizures) can also be delayed. To capture such changes, a 99% feature window overlap was chosen. That is, a new feature is calculated for every sample, for an effective classification rate of 1 kHz.

## B. Feature extraction hardware

The feature extraction hardware unit is shown in Fig. 20. Both line length and approximate spectral band power (using the sum of squares approximation) have a similar 100-sample register-based delay line, corresponding to the feature window, each connected to an accumulator to represent the summation of these 100 samples per feature. Each feature has a channel FIFO, controlled by the Channel ID signal that also controls the channel multiplexing state machine (timing diagram is shown in Fig. 21). Each channel FIFO is an 8-address register file that contains the current set of 100 samples for the corresponding active channel. The FIFO separates the feature data for each channel during the multiplexing phase. Channel multiplexing and serialization are employed to minimize the duplication of hardware. The feature extraction unit is reused for each channel, which forces the system to run at 8x the input sampling frequency.

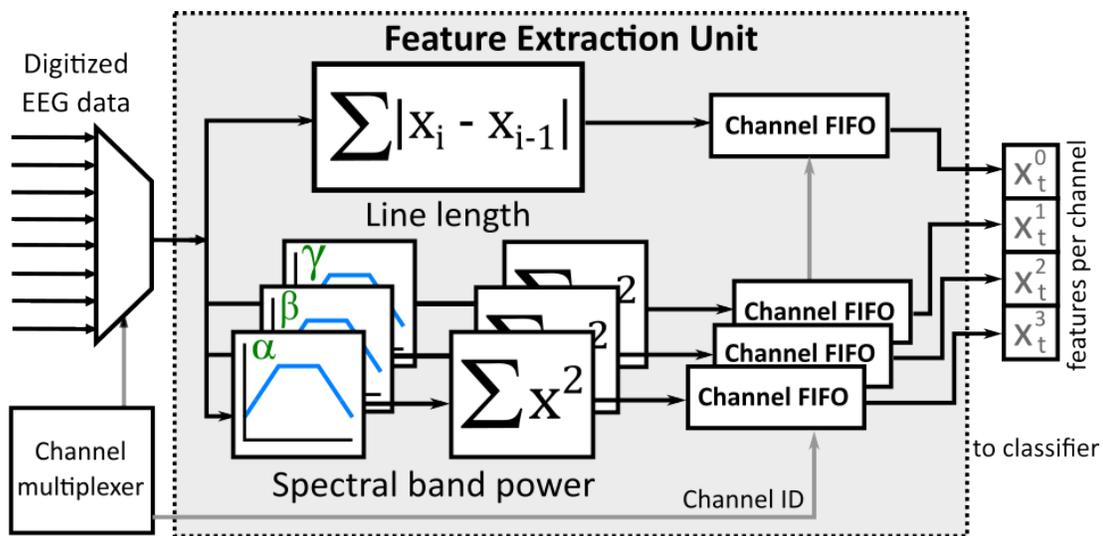


Fig. 20. Feature extraction hardware for SOUL.

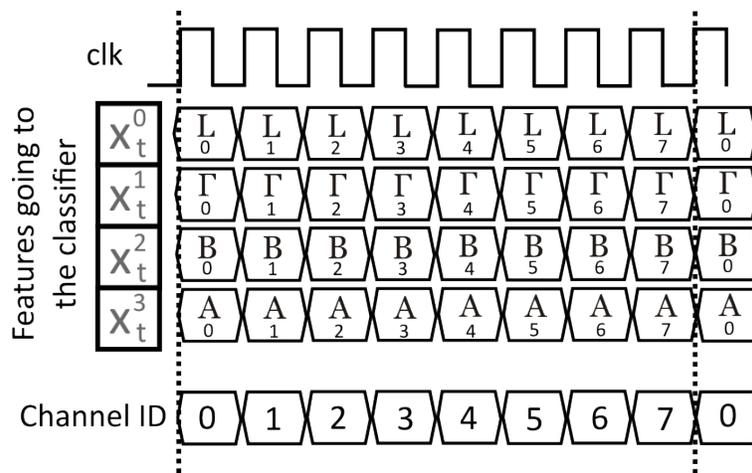
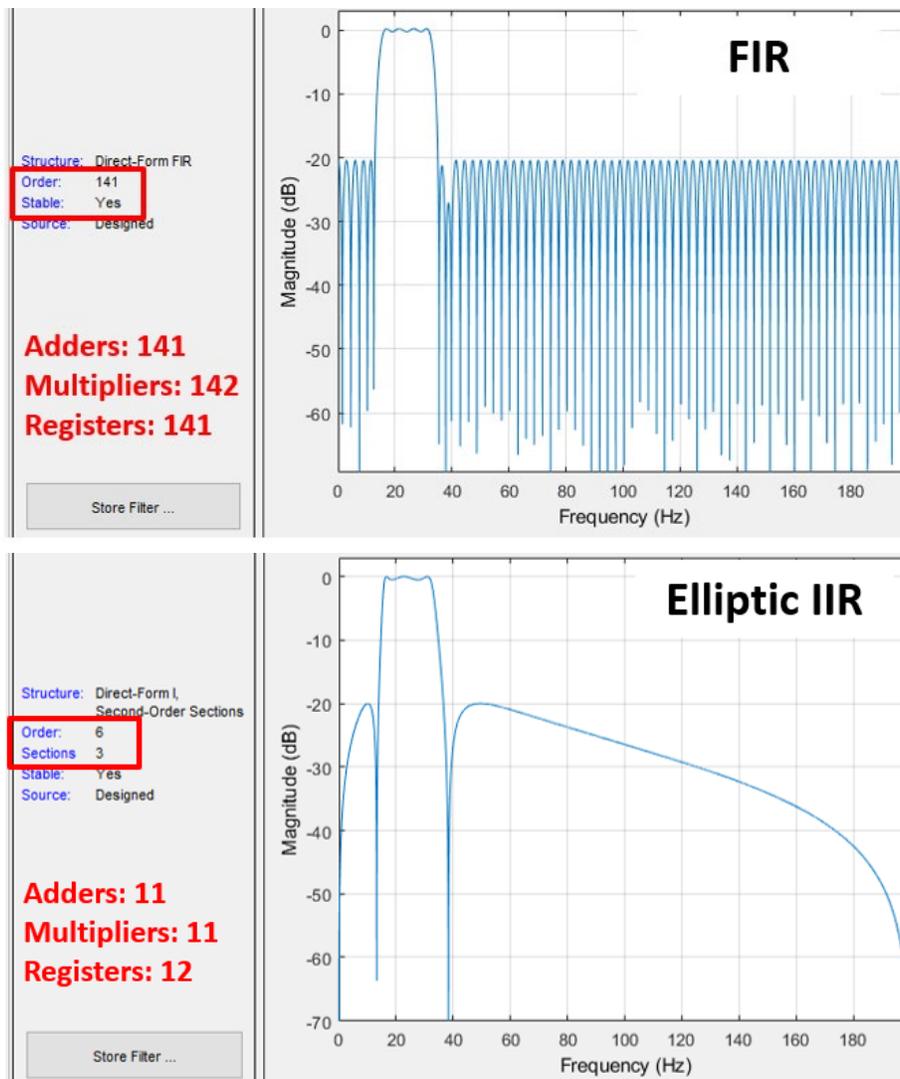


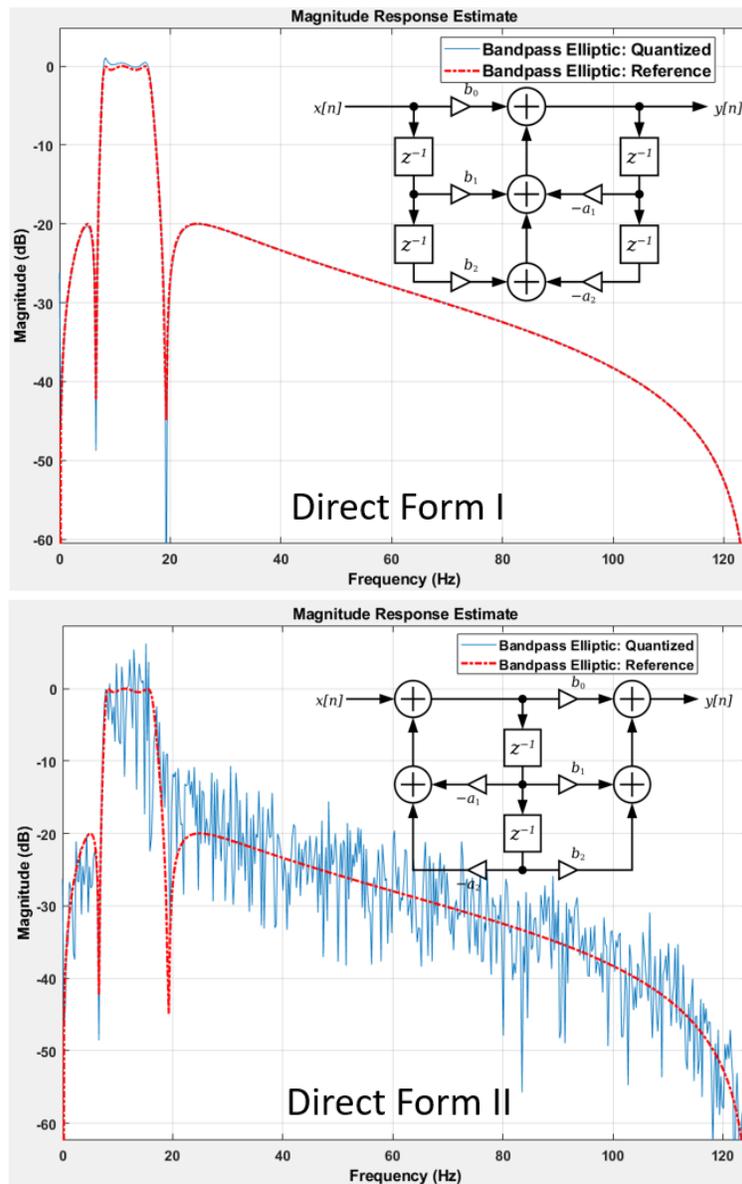
Fig. 21. Timing diagram for the SOUL feature extraction unit. It takes 8 clock cycles to go through every channel since the hardware for feature extraction is reused. The features are labeled accordingly (L: line length,  $\Gamma$ :  $\gamma$  band power, B:  $\beta$  band power, A:  $\alpha$  band power).

The spectral band power block uses IIR filters instead of the conventional FIR filter. During the feature extraction process, filters with at least 20 dB stopband were required for the spectral band power to work as a feature. If the filters do not meet those specifications, the approximated spectral band power using the sum of squares gets removed by the feature selection process leading to an accuracy degradation of >10%. Designing FIR filters in MATLAB, shown in Fig. 22, for a narrow passband, as an example, between 16-32 Hz, would require a minimum of 141 stages: each stage containing a register, an adder, and a multiplier. However, if elliptic IIR filters are used instead to achieve the same specification, it would only require three second-order sections: each section containing four registers, adders, and multipliers. Across all three spectral band power calculations, this filter choice translates to a 10x decrease in filter hardware requirements. When utilizing the elliptic filter architecture, the effects of frequency-dependent group delay on the classifier performance were ignored. It is assumed that this delay would be factored in during the offline training phase with minimal impact on detection latency.



**Fig. 22.** MATLAB-based implementations of a bandpass filter with similar specifications. The Elliptic IIR filter topology has more than 10x lower amount of hardware required compared to a conventional FIR filter topology.

The feature extraction unit computes in a 16-bit fixed-point format to avoid dedicated hardware for floating-point conversions. The Direct Form I IIR filter topology was used to avoid internal filter overflow. Fig. 23 demonstrates how the two IIR filter topologies (Direct Form I and Direct Form II) behave when the input bits are following a fixed-point (instead of floating-point) format. Given the 16-bit input to the system, 6 bits were set to be the integer part and the latter 10 were set to be the fractional part. This partitioning minimizes the round-off errors within the filter's internal states, which can cause instability. Through MATLAB filter design simulations, the number of bits can be reduced to 15 (with a 5-10 split between integer and fractional) given the datasets that were used for testing. However, 16 bits were retained as the hardware savings are marginal and an additional bit allows support for larger input signals to be processed.



**Fig. 23.** MATLAB-based implementations of an IIR bandpass filter using two different topologies. The topology diagram is superimposed on the corresponding plots. Direct Form I (top) is resilient to arithmetic overflow showing an almost similar quantized vs reference magnitude plots.

### C. Classification and online learning hardware

The SOUL hardware, shown in Fig. 24, merges the two modes of operation of the classifier: classification mode and retraining mode. During the classification mode, shown as the red path in the figure, the seizure probability is calculated using (1). The dot product for the logistic function is calculated in this mode. Since the four features from each channel are transferred one cycle at a time for every channel, the cumulative dot product is temporarily saved. Once all  $4 \times 8$  features are collected, then classification will proceed.

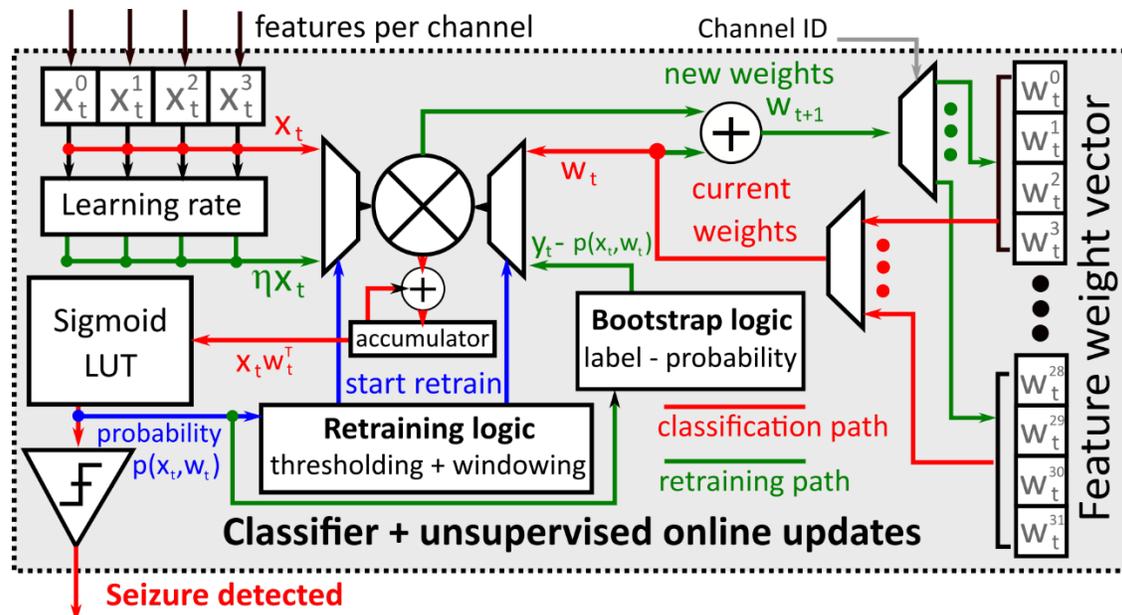
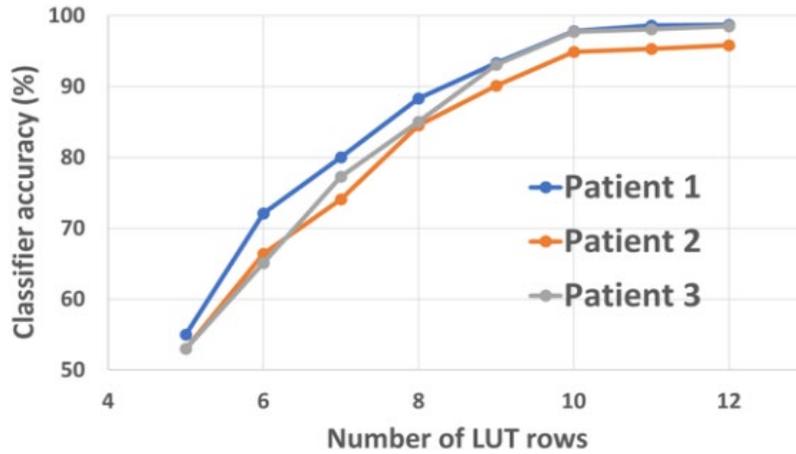


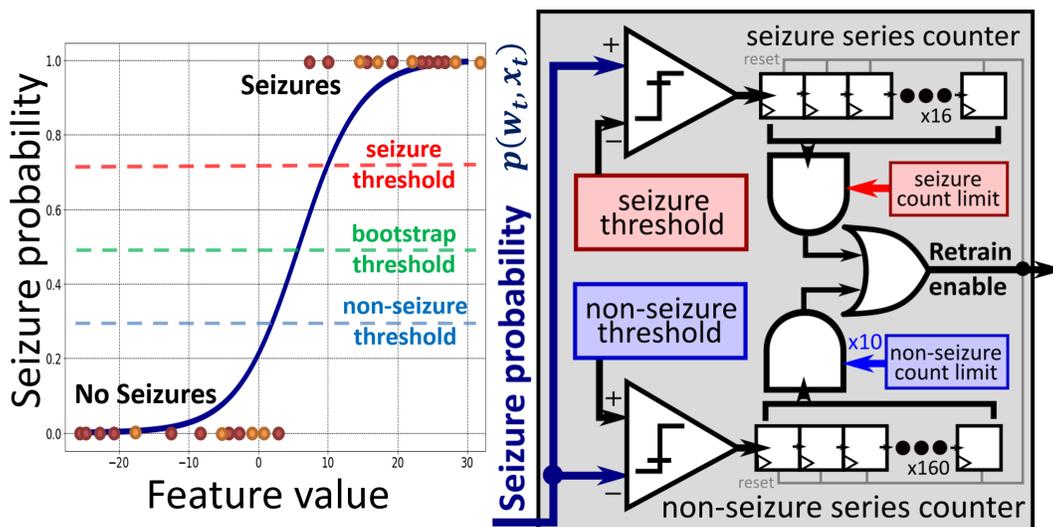
Fig. 24. Classification and online learning hardware for SOUL.

The logistic function is approximated using a look-up table (LUT) to minimize computation hardware. While the classifier output is rounded-off to determine whether a seizure is detected or not, the accuracy for the LUT will matter since the value of the logistic function is part of the SGD feature weight update formula, as shown in Fig. 15. For this system, a 10-entry LUT was found to be enough, shown in Fig. 25, as it impacts the classifier accuracy by  $<1\%$  compared to a classifier with full precision logistic function calculation.



**Fig. 25.** Logistic LUT approximation impact on classifier accuracy

The output of the LUT provides the input to two sets of comparators within the retraining logic shown in Fig. 16, which correspond to the high confidence thresholds as described previously in Chapter III. The hardware for the confidence thresholding and the high-confidence counters is shown in Fig. 26. Two separate confidence thresholds correspond to seizure and non-seizure. The seizure confidence threshold equals the value of  $CT$ , while the non-seizure confidence threshold equals  $1 - CT$ . The output of these comparators then goes to their corresponding series of shift registers representing the high-confidence counters. The HC value for non-seizures is set to 10x longer than the HC value for seizures to minimize the retraining frequency during the long non-seizure periods. This scaling balances the number of training points on the seizure and non-seizure events for an unbiased logistic model during the retraining period. Only when there is a series of high-confidence probability outputs and either one of the high-confidence counter limits is reached, the classifier goes into retraining mode. The retraining process can happen during either the seizure or non-seizure interval depending on which set of shift registers first reach the counter limit (corresponding to HC). The HC and CT parameters are programmable in hardware.



**Fig. 26.** Hardware implementation of the confidence thresholding and high-confidence counters for robust online learning.

Fig. 24 (green path) shows the retraining mode calculations following the SGD formula for logistic regression. The learning rate for the retraining was set to be approximately 0.015 (1/64) and is calculated with simple right shifts. The bootstrap register computes the difference between the generated label (thresholded against 0.5 for unsupervised learning) and the actual LUT-based logistic function approximation. The retraining mode finishes in 8 cycles, as the multiplier array is reused from the previous classification mode. During the update process, the old feature weight vectors are overwritten four at a time. Consequently, since the retraining mode consumes the same number of cycles as the classification, one input sample is ignored during the process. Once the retraining process is complete, the high-confidence counters (shown in Fig. 26) reset, and the classification mode begins for the next input sample. Accordingly, the collection of high-confidence detections starts again.

Fig. 27 shows the overall system architecture. The classifier receives 16-bit digitized neural data in 8 channels clocked at 1 kS/s. The implemented system supports 8 channels, but the algorithm is scalable to any number of channels. The classifier was fabricated in TSMC's 28 nm HPM process occupying 0.1 mm<sup>2</sup> in area, as shown in Fig. 28. The power consumption was measured to be 1.5  $\mu$ W by operating at 0.5V supply and 8 kHz clock frequency. This corresponds to an energy efficiency of 1.5 nJ/classification at a 1 kHz classification rate. The experimental results for SOUL in terms of long-term accuracy will be presented in Chapter VI of this document.

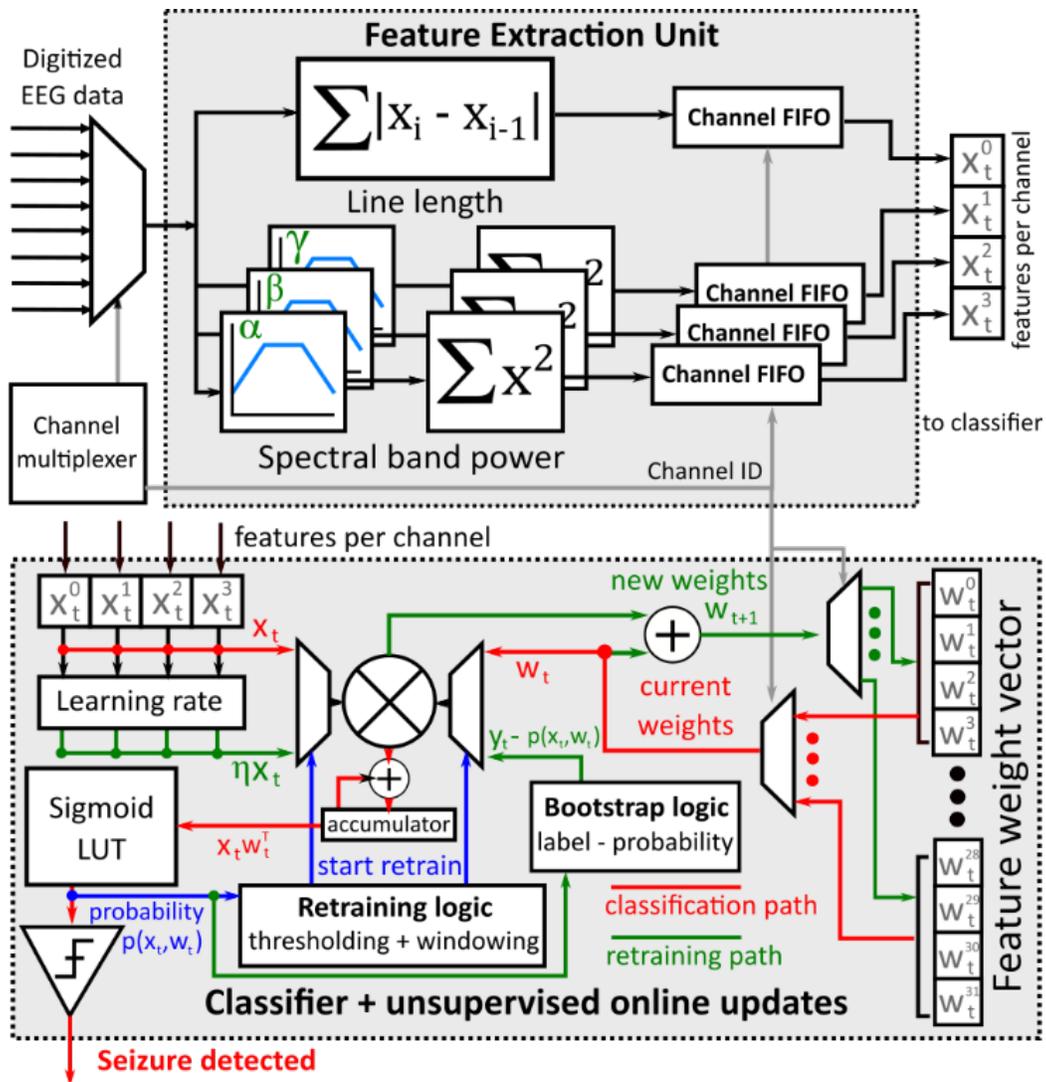


Fig. 27. System architecture (feature extraction unit + SOUL).

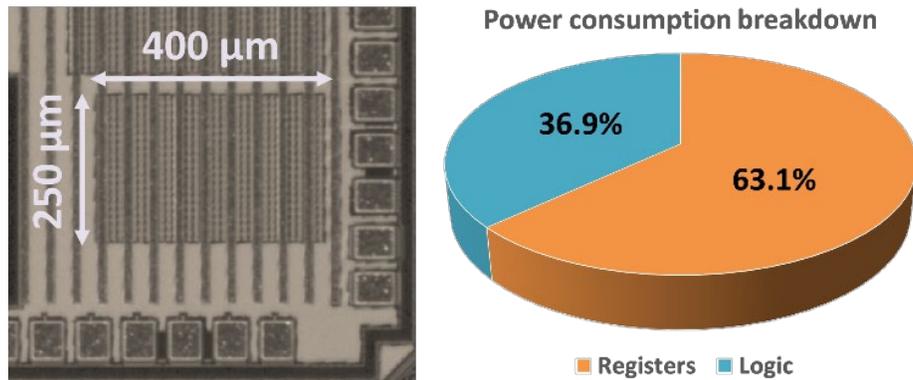


Fig. 28. Chip micrograph of SOUL in TSMC's 28 nm process and the power consumption breakdown (post-layout estimate).

## V. ON-CHIP SEIZURE PREDICTION

The same unsupervised online learning framework used for SOUL can be used for a predictor-type of classifier. Seizure predictors are algorithms or devices designed to forecast the occurrence of a seizure minutes before its actual onset. Utilizing the same EEG signals used in seizure detection, these predictors analyze patterns and anomalies that precede a seizure event. Fig. 29 illustrates the main difference between what a seizure predictor and a seizure detector classify. Seizure detectors, such as SOUL, differentiate between seizure states (called ictal EEG segments) and non-seizure states (called interictal EEG segments). They focus on the immediate identification of a seizure event by recognizing the specific characteristics and patterns that define the ictal phase. The transition from interictal to ictal is the critical point for seizure detectors, and their function is to provide real-time detection at the onset of this transition. In contrast, seizure predictors are designed to classify between interictal and a new region (within the interictal segment) leading up to, but not exactly, the seizure event (which will now be defined as preictal EEG segments). The preictal phase, which can also be thought of as the seizure prediction time window, represents the period during which certain physiological changes occur that are indicative of an impending seizure. The duration of the preictal phase varies among individuals. However, there is no standard length on what classifies as a preictal phase or an interictal phase. In some studies, this window has been identified to be as long as 30 minutes to an hour before the onset of a seizure [39]. Seizure predictors analyze these preictal phases to forecast the likelihood of a seizure minutes before it actually happens. This predictive capability allows for early intervention and management, providing a window of opportunity that is not available with seizure detectors alone.

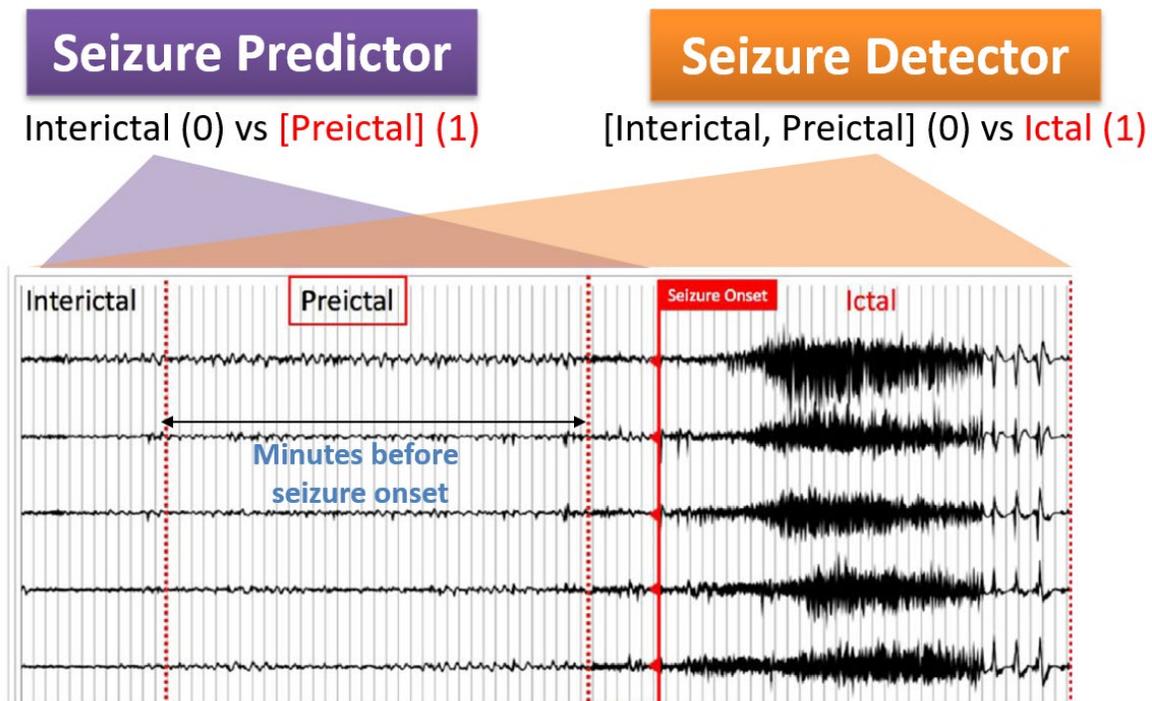
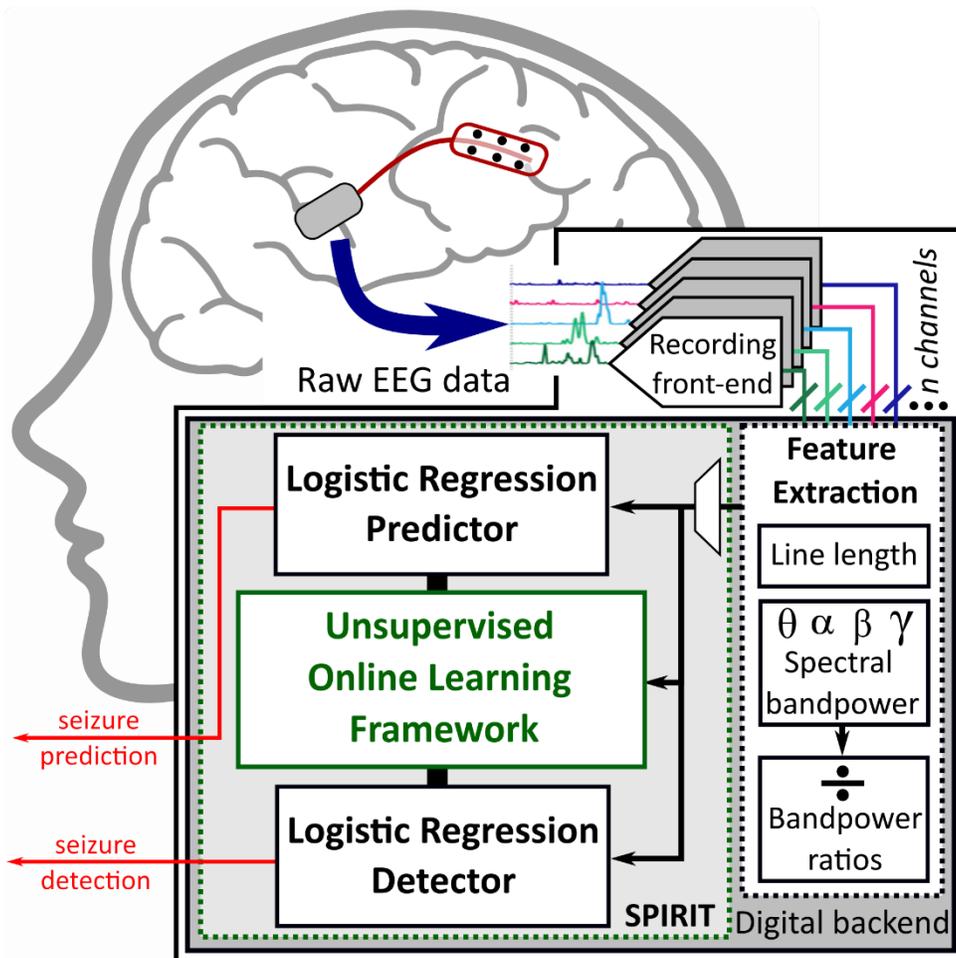


Fig. 29. Seizure predictor vs seizure detector showing the regions in the EEG that they classify.

This chapter covers how a seizure predictor can be realized. The first step is to be able to extract relevant features that can help differentiate interictal and preictal phases. As shown in Fig. 29 earlier, the time domain differences between the interictal and preictal segments are not directly obvious, which makes classification between the two very challenging. This would require an introduction to new features to be extracted that can better differentiate the two. Afterward, we will develop a classifier that would process these features to perform seizure prediction. This can be any classifier that is available. However, given that we have already developed SOUL and its online learning framework, we can leverage these to also improve the predictor accuracy over time. An accurate detector, such as SOUL, can help verify if the prediction was correct within the seizure prediction window. Predictor retraining, following the same online learning framework, will then be performed whenever the predictor output and the detector output do not match. This predictor that will leverage a SOUL to help retrain it to a higher prediction accuracy will be referred to as **SPIRIT (SOUL-based Predictor with Integrated detector for Retraining and In situ accuracy Tuning)**, illustrated in Fig. 30. By maximizing hardware reuse from the same detector hardware (SOUL and SPIRIT will both be based on logistic regression), SPIRIT is significantly energy efficient, even compared to state-of-the-art on-chip seizure detectors.

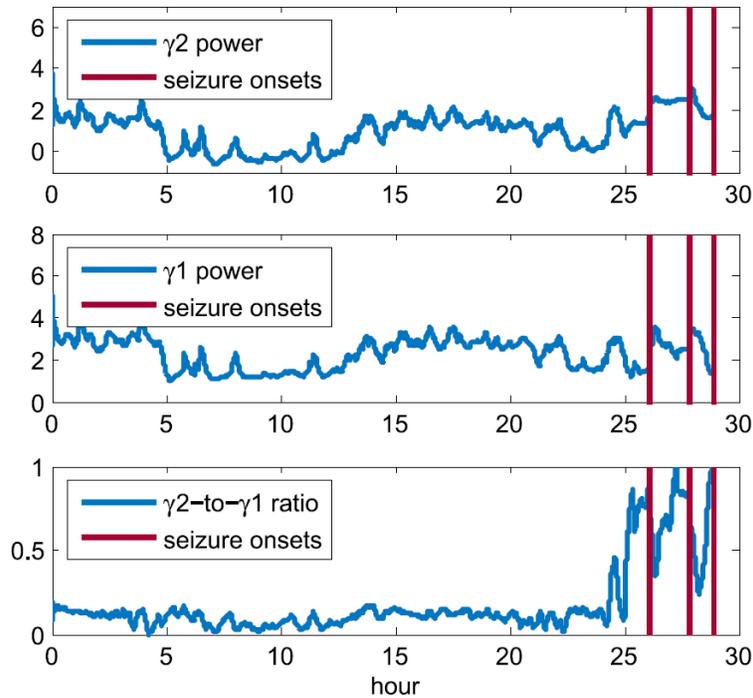


**Fig. 30.** Proposed seizure prediction system with an integrated unsupervised online learning detector to dynamically retrain the predictor.

### *A. Classifier features description*

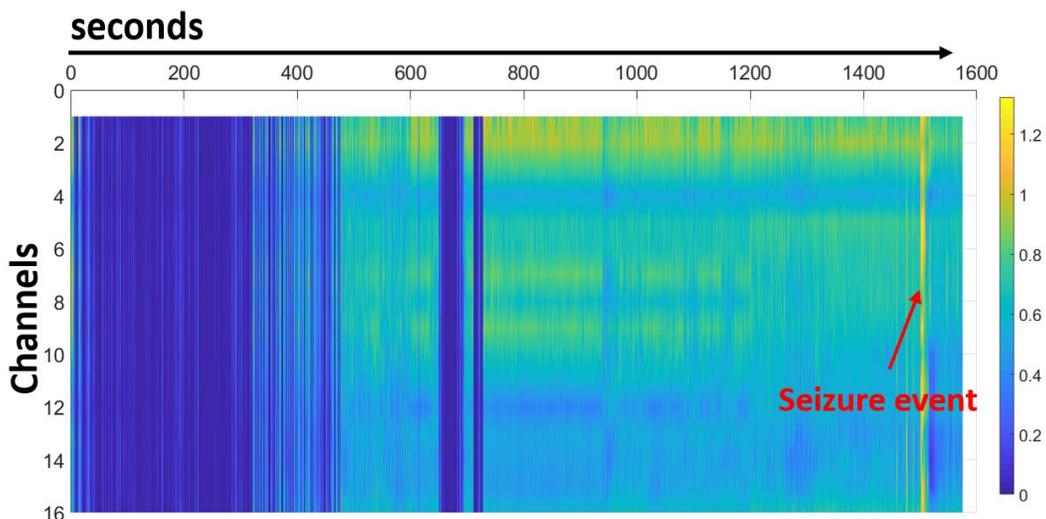
Differentiating the interictal and preictal events is challenging due to how similar the events are, both in the time and frequency domain. There has been research, primarily using software-based approaches, on seizure predictors and what features can be relevant to better classify between interictal and preictal phases. A crowd-sourced seizure prediction challenge, using a human intracranial (iEEG) dataset, has been offered in the past [12] (using the same dataset that was used in this work [11]) to analyze what the most common feature sets are to accurately perform prediction. From that research, spectral power in conjunction with statistics-based features stood out as the most common combination of features that can be utilized for an accurate seizure prediction. Spectral power-based feature extraction has already been developed for SOUL's seizure detection classification before, therefore it can be reused for SPIRIT's seizure prediction. However, statistics-based features such as variance, correlation, and standard moments would be challenging to implement in hardware as they require significant amounts of memory to hold all the data points before being able to calculate a feature value. A sequentially processed dataset requires queueing the data points as new data comes in and the oldest data is discarded. This scales with the window size where these statistics will be measured from.

There has been research on low-complexity features that can be used for seizure prediction [42,43]. It has been observed that ratios of spectral powers can provide a distinctive signature that can be associated with an impending seizure, as shown in Fig. 31. While individual spectral band powers might not help predict the onset of seizures, the ratios of two different spectral band powers showed some promise. The work in [42] has shown that seizures from some patients can be perfectly predicted, the false alarm rate is only around 0.1 per hour, between 3 to 75 minutes before the seizure onset using only just a single feature of the spectral power ratio. The advantage of using these features is that they can be easily calculated from the current feature set SOUL had, as it already has parallel spectral band power calculations.



**Fig. 31.** Spectral power of an individual frequency band (top and middle panel) does not seem to indicate any incoming seizure events. However, calculating for their ratios would give a significant increase in values as the seizure onset approaches. Image credit [x]

Testing this new feature calculation out on one of the long-term datasets show some promise, as seen in Fig. 32. For several minutes before the actual seizure event (i.e. the preictal segment), an increased activity in the spectral power ratios can be observed, contrasting the lower feature values towards the interictal (left side) region. Therefore, such features can be integrated into the feature extraction process which can then be used by the seizure predictor classifier.

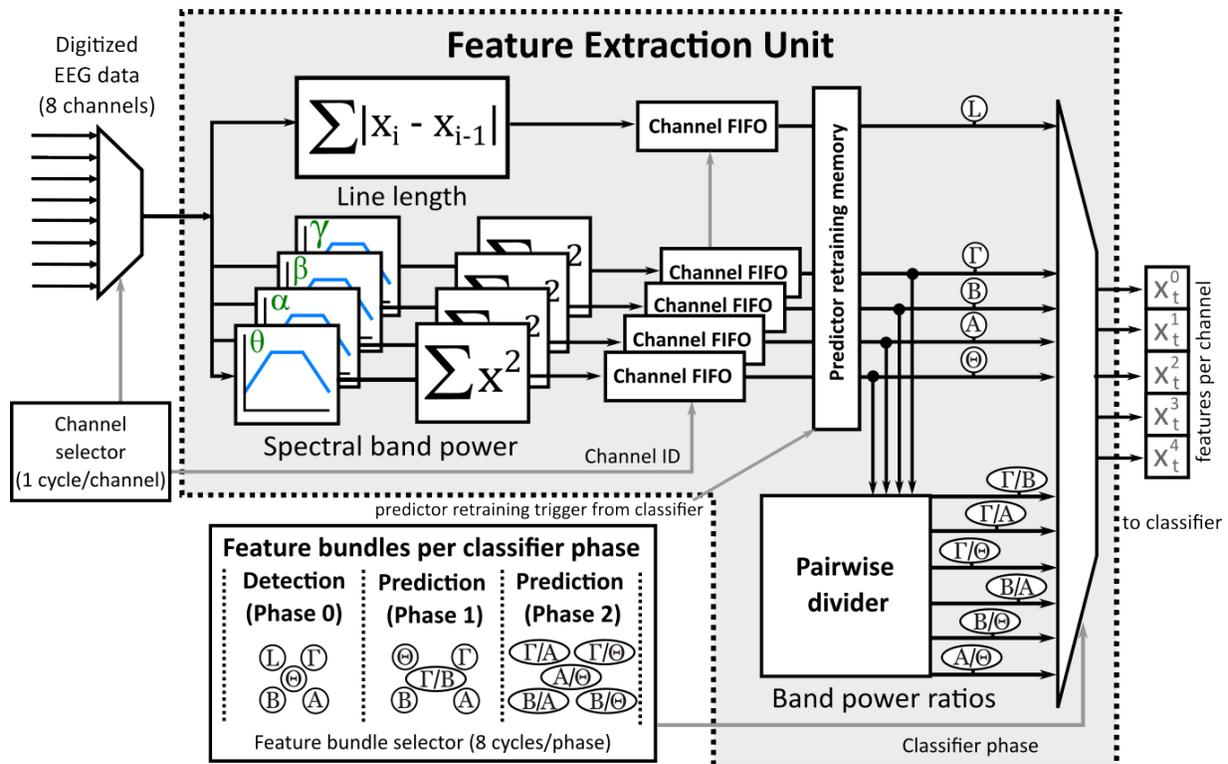


**Fig. 32.** Minutes before the actual seizure event, happening at around 1,500 second mark, an increase in spectral power ratio is observed.

Seizure prediction for SPIRIT would require 10 total features. Four spectral band powers are calculated from four EEG frequency bands:  $\theta$ , 4-8 Hz;  $\alpha$ , 8-16 Hz;  $\beta$ , 16-32 Hz;  $\gamma$ , 32-96 Hz. It is noted that these are the same set of frequency bands used by the original SOUL implementation with the additional  $\theta$  band to capture the lower frequencies more. Preictal events are much quieter in terms of electrical activity, so capturing an additional spectral band power towards the lower frequencies is beneficial. Moreover, six spectral band power ratios are also calculated from these four spectral band powers:  $\gamma/\beta$ ,  $\gamma/\alpha$ ,  $\gamma/\theta$ ,  $\beta/\alpha$ ,  $\beta/\theta$ , and  $\alpha/\theta$ . These specific pairs were all the possible combinations of ratios such that the numerator is always the higher frequency band. Consequently, as the electrical activity gradually increases as the seizure onset approaches, these ratios would have increasing values which can then be used to signal the preictal event. All 10 features for SPIRIT follow the same 100-sample sliding window used in SOUL.

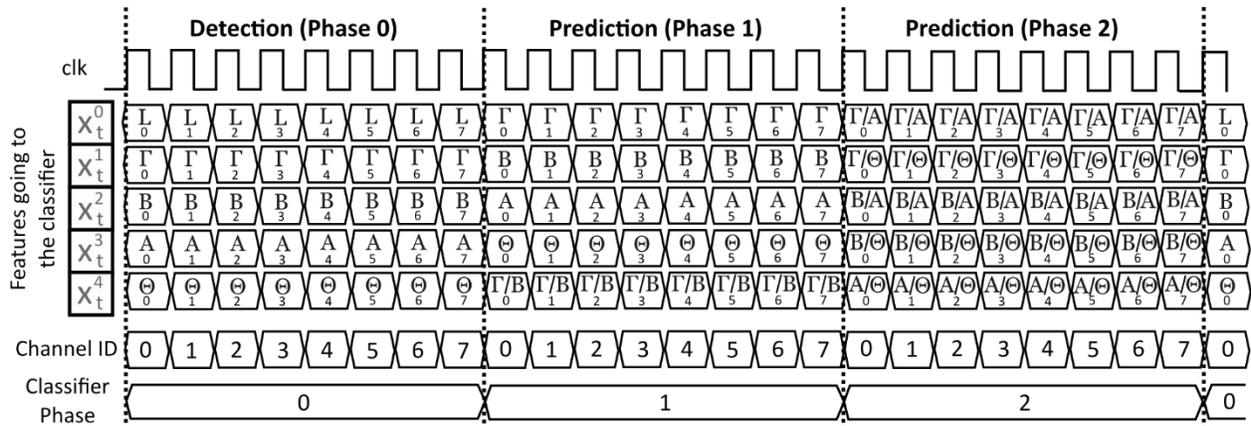
### B. Feature extraction hardware

The feature extraction hardware unit that supports both seizure detection and prediction is shown in Fig. 33. To save on additional hardware due to the increased number of features, only five features would be issued by the feature extraction block at any given clock cycle. Five features perfectly divide the 10 features needed by SPIRIT for seizure prediction into two. Furthermore, the seizure detection aspect would then be allowed to include the  $\theta$  spectral band power to be part of its features in addition to the four original features used by SOUL. This enables five features for seizure detection which matches the five-feature issue width of the feature extraction block.



**Fig 33.** Feature extraction unit for SPIRIT featuring a five-feature issue width, enabling detection (five features) to be completed in a single phase (8 cycles) and prediction (10 features) to be completed in the next two phases (16 cycles).

Supporting both seizure detection and prediction would require the five-feature issue to select specific combinations of features at any given cycle, depending on whether the classifier is performing detection or prediction. Fig. 34 shows a timing diagram demonstrating how the five-feature issue width is utilized to support both detection and prediction in three different phases. The first phase is meant for detection, supplying the classifier with the five seizure detection features: line length,  $\Gamma$ , B, A,  $\Theta$ . The last four represent the spectral band powers from the  $\gamma$ ,  $\beta$ ,  $\alpha$ , and  $\theta$  frequency bands respectively. Afterward, the next two phases are meant for prediction, supplying the classifier with the 10 seizure prediction features:  $\Gamma$ , B, A,  $\Theta$ ,  $\Gamma/B$ ,  $\Gamma/A$ ,  $\Gamma/\Theta$ , B/A, B/ $\Theta$ , and A/ $\Theta$ . The first four are the same spectral band powers for detection earlier, while the next six are the spectral band power ratios coming out of the pairwise divider logic shown in Fig. 33. As the feature extraction block remains time multiplexed across all 8 channels supported by the classifier, each phase lasts for 8 clock cycles each. As there are three phases in total to support both seizure detection and prediction, there are a total of 24 clock cycles to process both classifications from a single input sample. Consequently, this requires that the feature extraction block (and therefore, the classifier as well, to be shown later) operate at 24 kHz, to match the 1 kHz sampling speed at the input and maintain the 1 kHz classification rate as with SOUL.



**Fig 34.** The timing diagram demonstrating how the five features needed for detection are issued in the first phase, and the 10 features needed for prediction are issued in the next two phases. Each phase lasts for 8 clock cycles reflecting the channel multiplexing.

Supporting the 100-sample sliding window for SPIRIT does not necessarily translate to adding FIFO memory for the spectral band power ratios. As can be seen in Fig 33, the outputs of the pairwise divider, which computes the spectral band power ratios, do not have the memory at its output stage. Only the spectral band powers only require such memory as the ratios can always be calculated in real time by the feature extraction block. That is, there is no need to save the values of spectral band power ratios into memory as they can always be made available through the computation of the always-active divider. This trades off increased dynamic power, as the divider is always active every clock cycle, to save on feature extraction area and minimize leakage power for the additional memory requirement if ever.

### C. Classification and online learning hardware

The SPIRIT hardware, shown in Fig. 35, mainly utilizes the same architecture used in SOUL before, albeit with additional hardware to support the seizure prediction phase. Since the detection and prediction classification would require different sets of feature weights, there is an increased memory requirement to contain all of the feature weights. Specifically, seizure detection in SPIRIT requires five features to be computed across all 8 channels, translating to 40 feature weights, while seizure prediction in SPIRIT requires ten features to be computed across the same set of channels, translating to 80 feature weights. Combined, there are 120 unique feature weights that need to be loaded into the classifier for it to perform both seizure detection and prediction. As a reference, the original implementation for SOUL would only require 32 unique feature weights to be stored. The 120 feature weights are issued 5 weights at a time to match the issue width of the feature extraction block described earlier. It would then take 24 clock cycles, consistent with the timing diagram shown in Fig. 34, to complete both seizure detection and prediction from a single sample.

As shown in Fig. 35, throughout all 24 cycles, the same set of array multipliers, used in calculating the dot product for classification and the SGD calculation during online retraining, are used to compute five parallel multiplications at any given time. During classification, it would still pass through the same 10-entry sigmoid LUT used by SOUL. As the classifier supports both detection and prediction, the rounded-up output of the LUT (the output label) is processed differently depending on whether the classifier is in the detection or prediction phase.

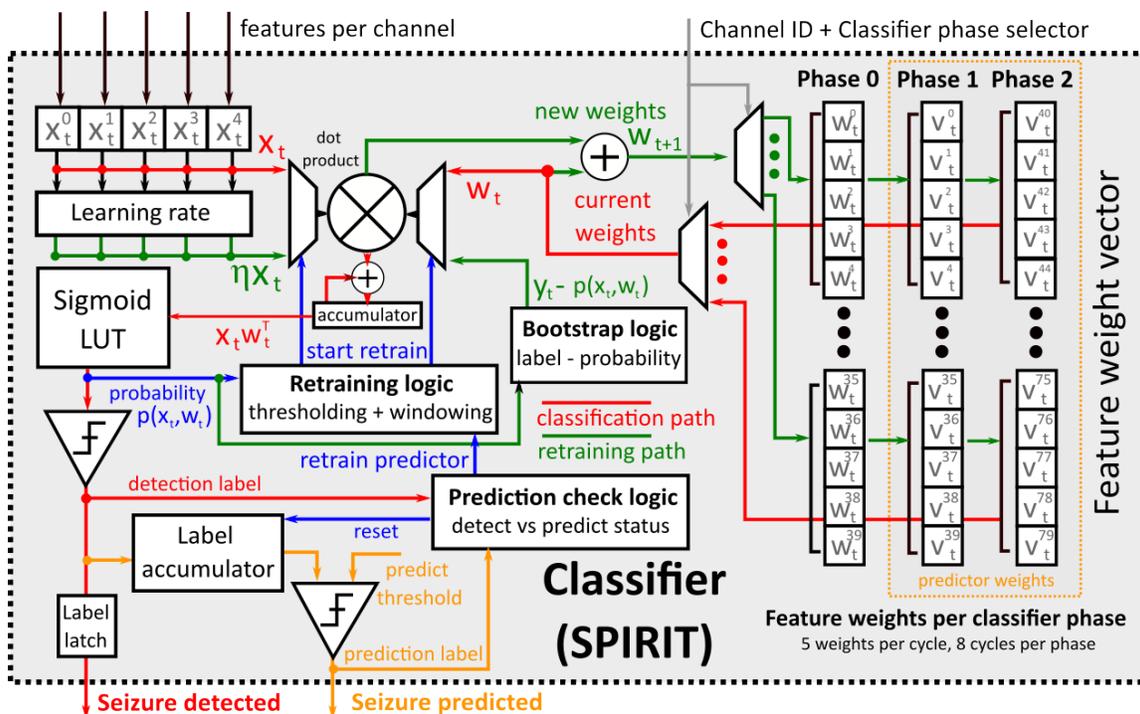
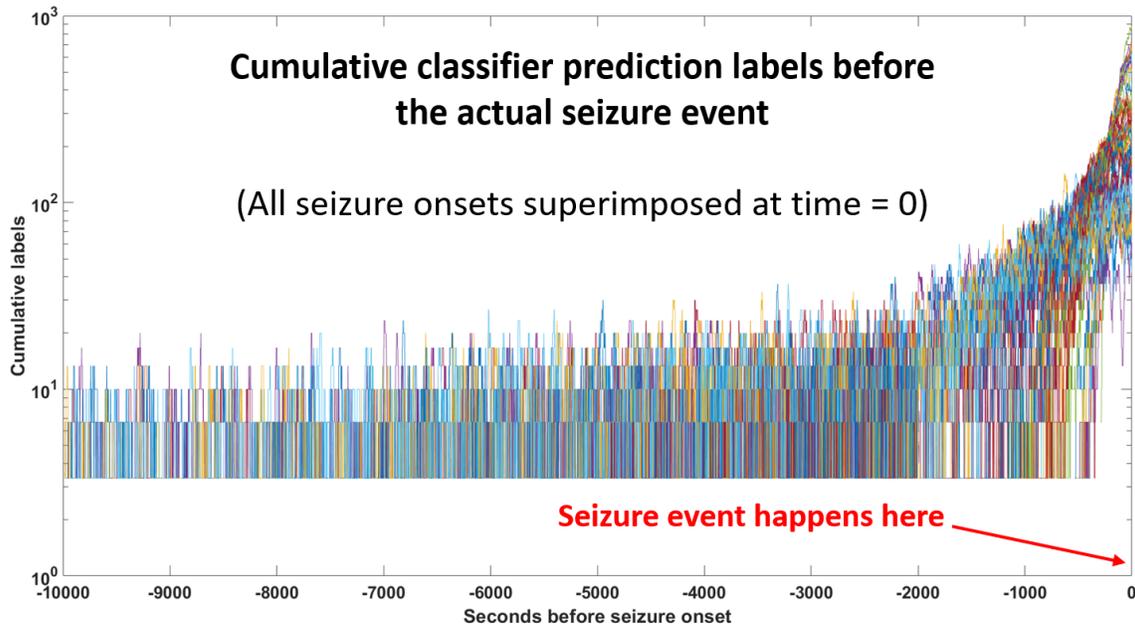


Fig. 35. Classification and online learning hardware for SPIRIT.

During the prediction phase of SPIRIT, the output label goes through a label accumulator which accumulates the prediction labels (+1 if the label is 1, and -1 if the label is 0). This

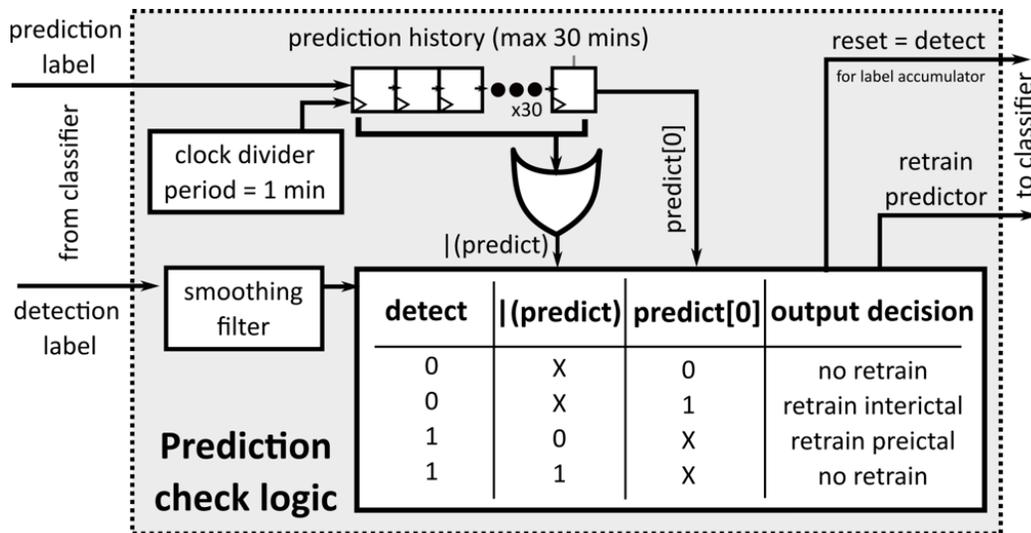
accumulation process is done to smooth out the relatively noisy prediction labels, a consequence of hard-to-differentiate interictal and preictal events. The accumulator value increases as the number of positive predictions outnumbers the negative predictions. Fig. 36 shows how the accumulator value increases as the seizure onset approaches at time = 0 (right). Therefore, a prediction threshold can be implemented which will then correspond to the final prediction output of the classifier, shown in Fig. 35.



**Fig. 36.** The accumulator value increases as the seizure event approaches, which can then be compared against a threshold to output the prediction.

The prediction check logic, also shown in Fig. 35, determines whether the predictor needs to be retrained. It compares the output of the detector with the predictor output history within a maximum window of 30 minutes. A 30-minute seizure prediction window was chosen as it was the average pre-ictal window size when training seizure predictors [39] as well as the average seizure prediction times on several seizure prediction studies [39-41]. As the detector and predictor have their own independent outputs, there are four possible cases that can arise. Conveniently, this matches the statistical confusion matrix in determining actual versus predicted outputs in terms of classifier accuracy. Fig. 37 shows the truth table with all the possible combinations of the predictor and detector outputs and how these different cases translate into the confusion matrix (whether it is a true positive/negative or a false positive/negative). If the predictor and detector outputs match, then the predictor was correct so there is no need to retrain the predictor. However, if the predictor and detector outputs were different, then it would be assumed that the detector is correct (the detector will always be treated as the ground truth for SPIRIT) and that the predictor needs to be retrained.

Predictor output	Detector output after some time*	What happened	Detector action for predictor
No seizure (0)	No seizure (0)	True negative	Do nothing
No seizure (0)	Seizure (1)	False negative	Preictal (1) retrain
Seizure coming (1)	No seizure (0)	False positive	Interictal (0) retrain
Seizure coming (1)	Seizure (1)	True positive	Do nothing

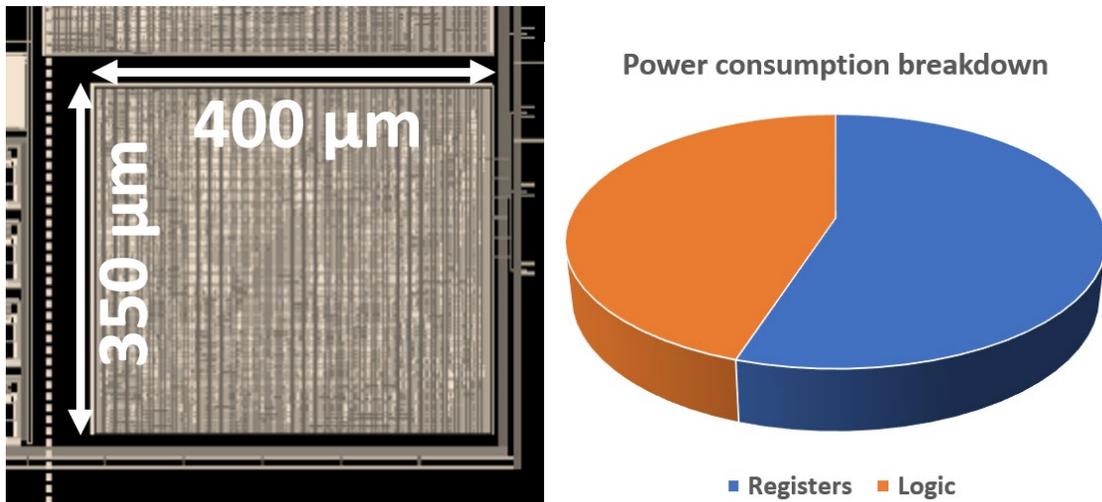


**Fig 37.** Predictor retraining is determined whether the predictor and detector outputs match or not. The hardware equivalent of these conditional checks is also shown.

Fig. 37 also shows the corresponding hardware for the prediction checker. The checker maintains a history of the past prediction outputs for the last 30 minutes and uses that information to determine whether there was a correct prediction or not. Whenever the detector output is 0 (no seizure is detected), then the logic checks what the prediction 30 minutes ago was (checking for predict[0] as shown in the diagram, the last bit of the prediction window shift register). If the prediction output is 1, then there is a mismatch and the predictor needs to be retrained using an interictal sample. An interictal sample is needed for the retraining since the mismatch was due to a false alarm, implying that the predictor is too sensitive. Training on an interictal sample will bias it towards less sensitivity. On the other hand, if the detector output is 1 (there is a seizure right now), then the logic checks whether that seizure event has been predicted anywhere within the 30-minute window (checking for the OR-reduced predict value as shown in the diagram). If there was no positive seizure prediction that happened within the last 30 minutes, then there is a mismatch and the predictor needs to be retrained using a pre-ictal sample. This time, a preictal sample is needed since the mismatch was due to the predictor not being sensitive enough (which made it unable to properly predict the seizure). Training on a preictal sample will bias it towards increased sensitivity. This method of retraining the predictor ensures that the retraining is always done due to a mismatch between detector and predictor outputs and that the retraining is always done to correctly fix the mismatch (deciding whether to use an interictal or a preictal sample for retraining biases the model differently).

As a final note on the diagram shown in Fig. 37, the detection label also goes through a smoothing filter so that the prediction check logic does not trigger due to a sudden misclassification of the detector. The method is similar to how the confidence thresholding and high-confidence counters were done for SOUL (Fig. 26). Only a series of positive detections will output a 1. For SPIRIT, a consecutive stream of 5 positive detections will output a 1. This ensures that a misclassification due to glitches can be safely ignored and will not cause the predictor to retrain toward the wrong direction. This is an important step since the detector output will always be held as ground truth by SPIRIT. While the detector can still make mistakes (it does not have 100% accuracy) and the predictor mistakenly retrains itself, it is assumed that the detector can correct the error later as correct detections will outnumber the number of wrong detections. It is important to note that only the detector can influence the retraining of the predictor and not the other way around. This ensures that even if the predictor is retrained wrongly due to the detector misclassification, the detector will not be trained wrongly as well. The detector will remain independent of the predictor.

SPIRIT was also fabricated in TSMC’s 28 nm HPM process, similar to SOUL, occupying 0.14 mm<sup>2</sup> in area. The chip micrograph is shown in Fig. 38. As SPIRIT is operating at 3x the clock frequency than SOUL (since it considers three classification phases), the supply voltage was only able to go down to 0.65V. The power consumption was measured to be 17.2 μW (power breakdown between registers and logic is shown in Fig. 38), which corresponds to an energy efficiency of 17.2 nJ/classification at a 1 kHz classification rate. The experimental results for SPIRIT in terms of prediction time will be presented in the next section of this document.



**Fig. 38.** Chip micrograph for SPIRIT and the power consumption breakdown.

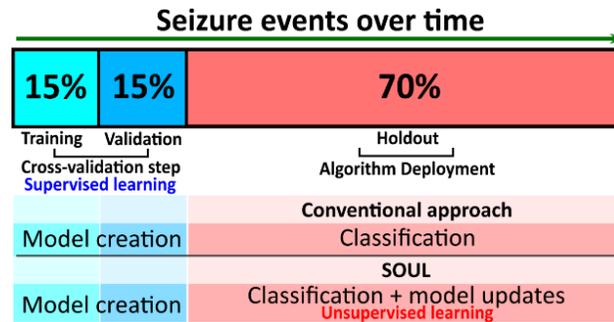
The next chapter focuses on the classification results of SOUL and SPIRIT. It is worth mentioning that SPIRIT can be configured in detection-only mode. In this configuration, SPIRIT acts SOUL, performing only seizure detection using the exact same features in SOUL. The weights for the additional  $\theta$  band power in SPIRIT are zeroed out. SPIRIT (in detection-only mode) and SOUL have exactly the same performance in terms of accuracy and are interchangeable when focusing only on seizure detection.

## VI. CLASSIFIER PERFORMANCE

### A. Dataset description

The performance of SOUL and SPIRIT is tested using the intracranial EEG (iEEG) dataset [11] and the CHB-MIT scalp EEG dataset [44]. The former features >100-hour recordings on three patients to demonstrate how online learning performs over a long period of time. The latter is a collection of relatively short recordings of 24 patients for performance comparisons on a wider population. The CHB-MIT dataset also allows for state-of-the-art comparisons as it is a commonly used dataset to test seizure classifiers. Moreover, using these two datasets also measures how the classifiers perform on datasets having different recording processes (iEEG versus scalp EEG).

The iEEG dataset was divided into 15% training, 15% validation, and 70% testing sets, as illustrated in Fig. 39. Contrary to random sampling during offline training, which is typically done in conventional machine learning approaches, time-series causality is maintained by considering only the first 30% (training + validation) of the data. Due to the limited seizure data for some patients in the CHB-MIT dataset, at least 2 seizure events were used for training and validation. However, if applicable, an approximate 15-15-70 split is still applied. For both datasets, the non-seizure samples were trimmed to balance the training data (equal number of seizure and non-seizure training points). Non-seizure samples closest to the start and end of the seizure events were retained to improve classification accuracy. The duration of the training, validation, and test sets, as well as the division between seizure and non-seizure samples, are shown in Table II.



**Fig 39.** Dataset partitioning maintaining the causality of the time series data.

When training SPIRIT for seizure prediction, the dataset was labeled differently. Since SPIRIT has a 30-minute seizure prediction window, 30 minutes of the interictal dataset leading up to every seizure event is labeled as ‘1’ which then becomes the preictal dataset. The actual ictal/seizure events were removed during the training process. Since we only want to focus on classifying between interictal and preictal events for prediction, the high activity during ictal events can degrade the training quality. During the tests, the SPIRIT predictor will inherently ignore the ictal events when it sees that the detector output is ‘1’, corresponding to a detected seizure event. This prevents any possible predictor misclassifications when the actual seizure event is already occurring. For both seizure detection and prediction, classifier accuracy is reported by calculating the sensitivity (true positive rate) and specificity (true negative rate), as shown below:

$$\text{Sensitivity} = \frac{\text{detected seizures (label 1)}}{\text{number of seizures}} \quad \text{Specificity} = \frac{\text{negative detections (label 0)}}{\text{number of non-seizures}}$$

Both sensitivity and specificity are calculated on a sample-per-sample basis. That is, the corresponding classifier output for every sample is checked against the true label that was provided with the dataset. This method increases the granularity of the reported sensitivity and specificity, which is beneficial for datasets that contain very few seizures such as CHB-MIT.

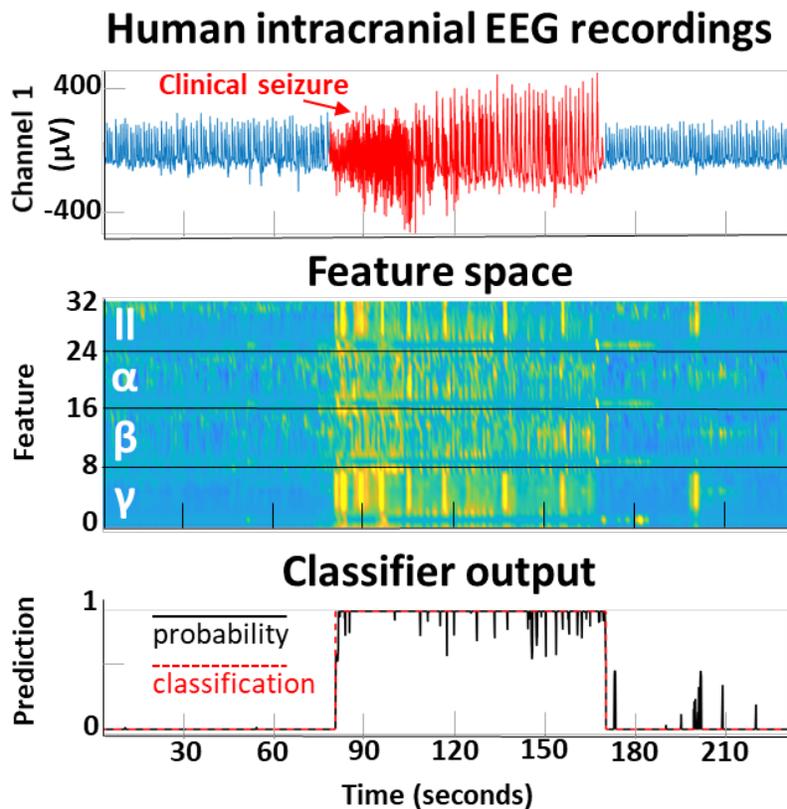
TABLE II  
DATASET PARTITIONING BETWEEN TRAINING, VALIDATION AND TEST

	Training set		Validation set			Test set		
	Seizures (#, sec)		Seizures (#, sec)		NS (hrs)	Seizures (#, sec)		NS (hrs)
<b>Patient 1*</b>	51	2479	51	2295	21.4	239	12667	98.2
<b>Patient 2*</b>	31	1953	31	2046	19.3	142	9514	190.1
<b>Patient 3*</b>	71	2982	71	3834	25.3	332	20252	246.9
<b>Patient 1</b>	1	42	1	29	0.57	5	385	37.1
<b>Patient 2</b>	1	84	1	83	1.06	1	11	19.2
<b>Patient 3</b>	1	54	1	67	1.11	5	295	36.8
<b>Patient 4</b>	1	51	1	113	8.37	2	222	129
<b>Patient 5</b>	1	117	1	112	7.18	3	339	26.7
<b>Patient 6</b>	2	33	2	39	10.0	6	101	54.6
<b>Patient 7</b>	1	88	1	98	3.55	1	145	23.5
<b>Patient 8</b>	1	173	1	192	3.06	3	564	16.2
<b>Patient 9</b>	1	66	1	81	5.43	2	137	40.8
<b>Patient 10</b>	1	37	1	72	16.2	5	352	16.1
<b>Patient 11</b>	1	24	1	34	1.67	1	754	1.04
<b>Patient 12</b>	6	175	6	258	4.68	28	1042	16.0
<b>Patient 13</b>	2	118	2	99	7.89	8	342	7.83
<b>Patient 14</b>	1	16	1	22	0.83	6	147	22.6
<b>Patient 15</b>	3	286	3	244	4.78	14	1412	20.7
<b>Patient 16</b>	2	22	2	24	5.03	6	58	8.69
<b>Patient 17</b>	1	92	1	117	1.21	1	90	19.1
<b>Patient 18</b>	1	52	1	32	0.18	4	245	6.47
<b>Patient 19</b>	1	80	1	79	1.74	1	83	1.08
<b>Patient 20</b>	1	31	1	32	1.37	6	247	17.2
<b>Patient 21</b>	1	58	1	52	1.37	2	97	13.1
<b>Patient 22</b>	1	60	1	76	4.94	1	74	9.11
<b>Patient 23</b>	1	115	1	22	1.75	5	301	23.7
<b>Patient 24</b>	2	54	2	65	2.62	11	397	18.0

\* = iEEG dataset, S = Seizures, NS = Non-seizures

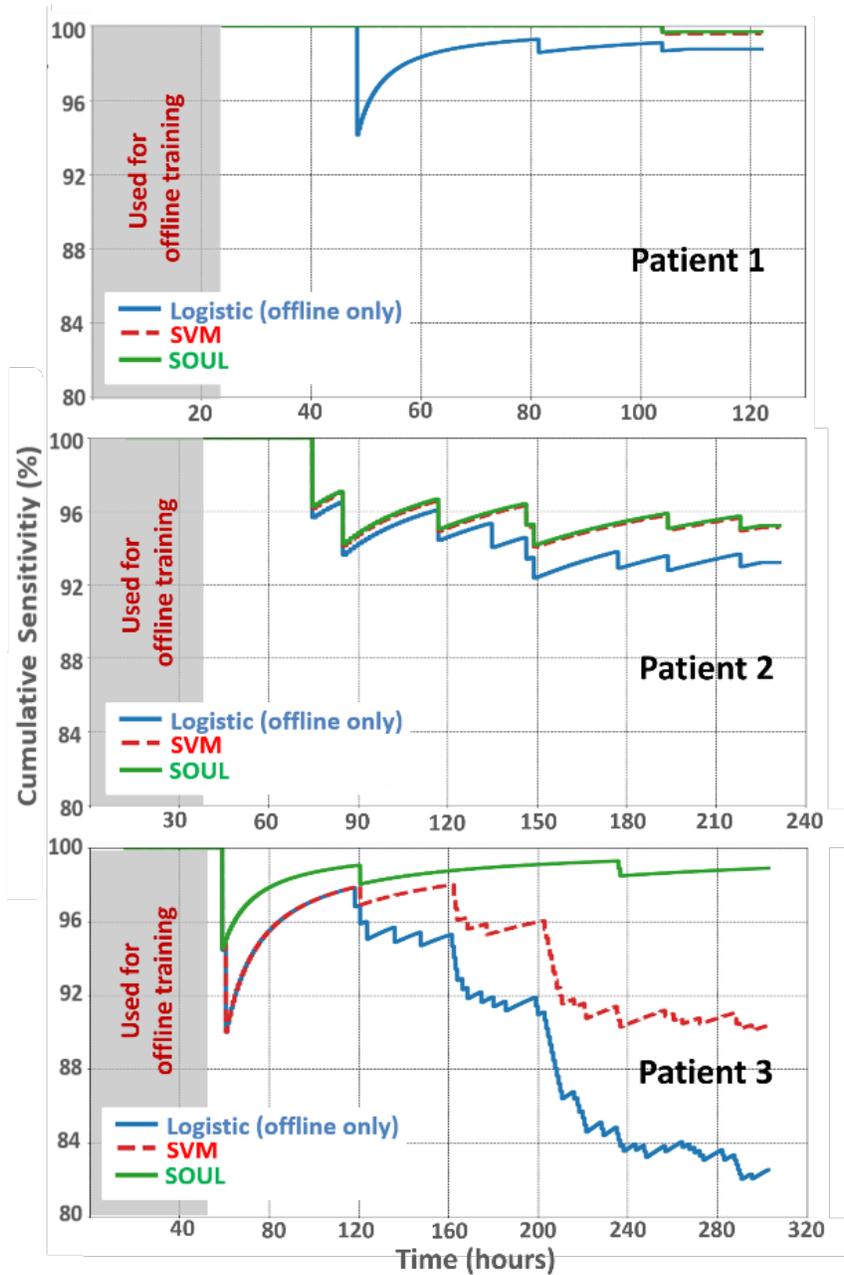
## B. Seizure detection performance on long-term iEEG data

The iEEG data is comprised of recordings from three human patients that had the lowest seizure prediction performances out of the ten patients in a clinical trial done in [11]. A sample real-time SOUL classification is shown in Fig. 40. Fig. 41 shows the classification performance over time of three classifiers: SPIRIT (in detection-only mode), logistic regression, and a representative SVM. The latter two are only trained offline. As stated in the previous chapter, SPIRIT (in detection-only mode) acts exactly the same as SOUL and is therefore interchangeable. Whenever seizure detection is performed, SOUL will be mentioned instead for brevity. Fig. 41 shows that incorporating online learning results in an average sensitivity and specificity of 97.9% and 98.2% for the three patients. For the three patients, the average sensitivity improvement is 6.5% with <1% specificity degradation. This degradation is a consequence of utilizing a linear classifier, such as the logistic regression used in SOUL. As a new seizure training point is introduced during retraining, the classifier tends to bias towards increased sensitivity (so that succeeding seizures can be better detected) while sacrificing specificity (as higher sensitivity leads to increased false alarms). This effect can be mitigated by also training on the non-seizure segments. The overall specificity degradation of <1% is considered an acceptable trade-off for a more significant improvement in sensitivity.

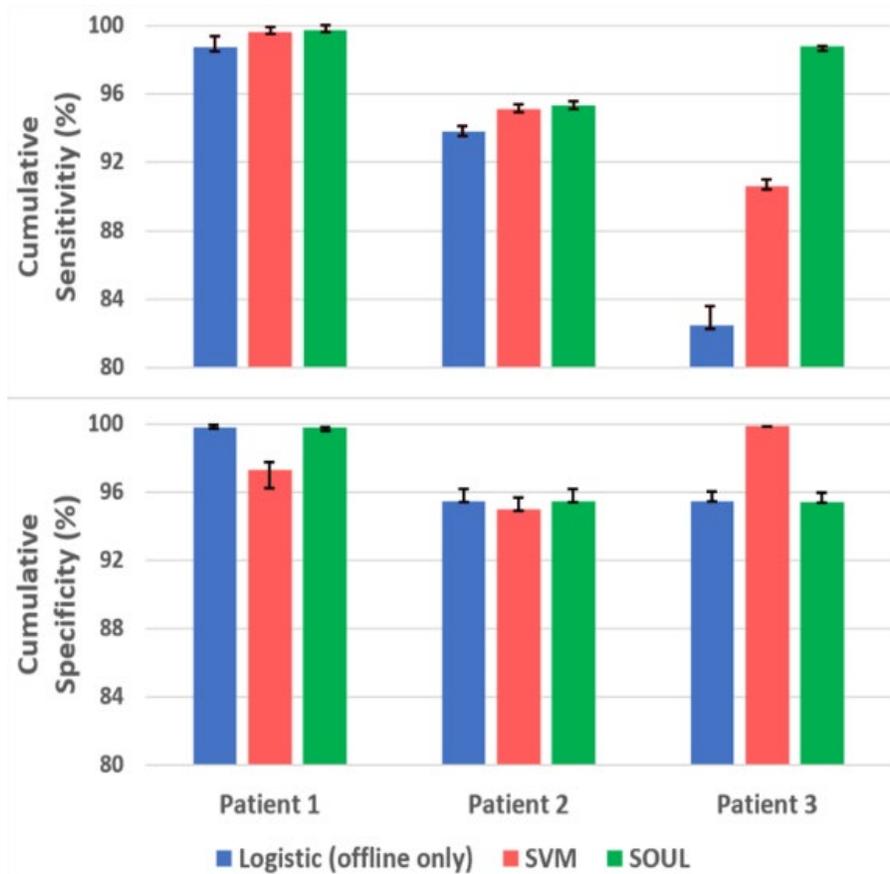


**Fig. 40.** SOUL classification in real-time showing the corresponding feature space and logistic output probability for the EEG recording.

While the performance of SOUL and the SVM is the same for Patients 1 and 2, a significant performance difference is observed for Patient 3. The decreasing sensitivities of the conventional offline-only-trained classifiers demonstrate that seizure patterns change over time, which leads to missed detections. As SOUL tunes the feature weights during classification, it effectively tracks the iEEG signal variability, allowing sensitivity to be maintained over time. Fig. 42 shows the summary of final sensitivity and specificity values across all three patients after running the test dataset. In this work,  $<1.2$  false alarms per day ( $>95\%$  specificity) are maintained for all patients in all algorithms, equivalent to false alarm rates of commercial devices [5,6].



**Fig. 41.** Comparison of cumulative sensitivity over time for different classifiers versus SOUL for all three iEEG recordings. For Patients 1 and 2, SOUL and SVM sensitivity performance is equivalent.

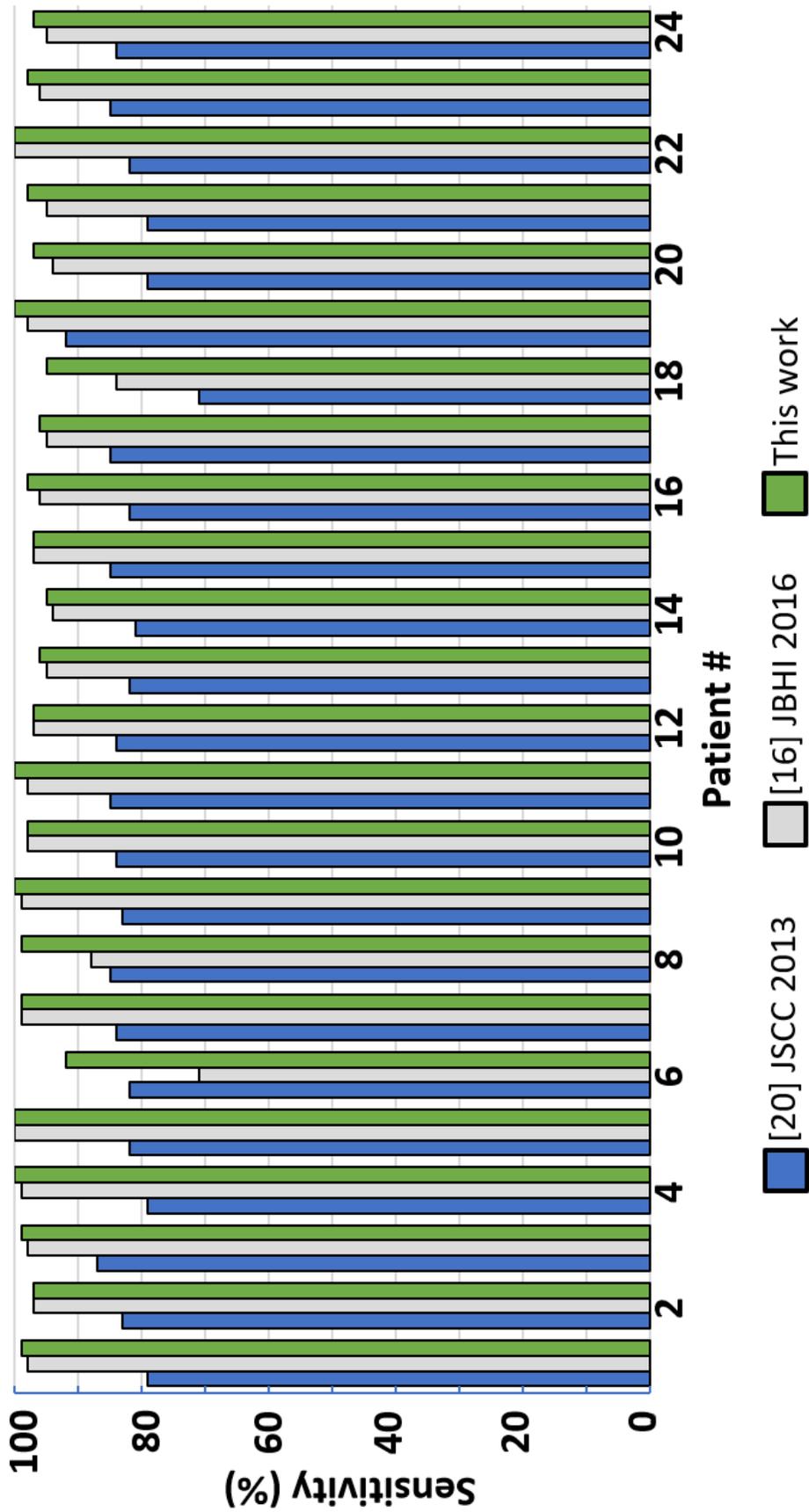


**Fig. 42.** Final sensitivity and specificity values at the end of the iEEG testing period; error bars indicate max and min values within the last 24 hours. All classifiers are trained so that specificity remains >95%.

### C. Seizure detection performance on scalp EEG data

The CHB-MIT dataset consists of scalp EEG recordings from 24 pediatric subjects with intractable seizures [44]. Across all subjects, the mean recording time was 41 hours and the mean number of recorded seizure events per subject was 7.6. Compared to the iEEG dataset, this is significantly shorter in terms of recording time and the number of seizures. However, this dataset is used for comparison since most seizure detection systems refer to this dataset.

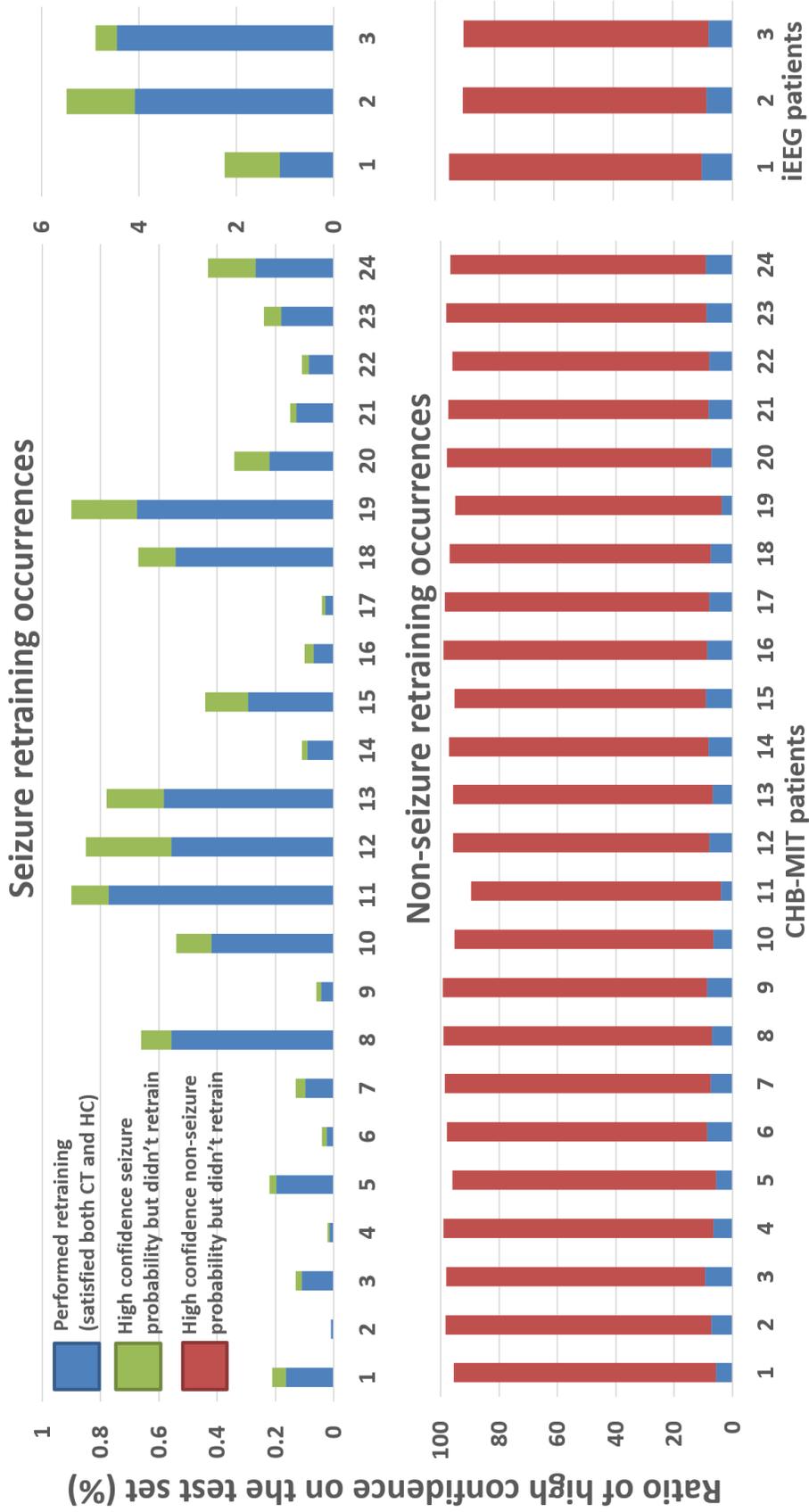
Fig. 43 shows the comparison to other works which presented their results on a per-patient basis across all 24 subjects [16,20]. For some select subjects (subjects 6, 8, 18), greater than 12% improvement in sensitivity was observed. For the rest of the subjects, there was a 1-3% improvement. Across all 24 subjects, the average sensitivity improved by 14.8% compared to [20], and 1.8% compared to [16]. The average specificity for all subjects was 98.2%, which is 2.7% better than [20] and 0.2% better than [16]. Since SOUL was able to output the correct label within the seizure window for all seizure events, the event-based sensitivity for all patients is 100%.



**Fig. 43.** Sensitivity comparison against state of the art on the CHB-MIT dataset on select patients where significant improvement was observed. For the remainder of the patients, an average of 1-3% improvement was observed.

The classification performance of SOUL on the two datasets shows that the proposed unsupervised online learning scheme works for both iEEG and scalp EEG. This demonstrates the flexibility of the algorithm on different EEG recording methods, as well as on different recording lengths. Compared to the other classifiers, SOUL maintains equal or higher sensitivities over the entire classification period. As classification goes on for longer, the sensitivity improvement from SOUL increases, as seen in Fig. 41 (Patient 3).

Fig. 44 shows the percentage of the test data translating to high-confidence classifications. The high confidence percentage for seizure classification is very low ( $\sim 5\%$  for iEEG and  $< 1\%$  for CHB-MIT) with respect to the overall test data. This is directly correlated to the rarity of seizure events. Accordingly, the high confidence percentage for non-seizure classification is very high ( $\sim 90\%$ ) as these comprise the bulk of the EEG recordings. The amount of retraining enabled from the consecutive high-confidence classifications is also shown in the figure. It is more likely for SOUL to retrain on a seizure event ( $> 50\%$ ) than on non-seizure data ( $< 10\%$ ). This is due to how the HC hyperparameter for non-seizures is set up. As shown in Chapter IV.C, HC for non-seizures is 10x the HC value for seizures. This is to minimize the retraining frequency of SOUL over the long non-seizure periods. Effectively, SOUL biases towards higher sensitivity by retraining more frequently on rare seizure events.



**Fig. 44.** Seizure and non-seizure retraining occurrences based on high confidence classifier outputs for the two datasets. SOUL retraining more during the rare seizure segments to bias towards increasing sensitivity (increasing seizure detection rate).

#### D. Seizure prediction performance

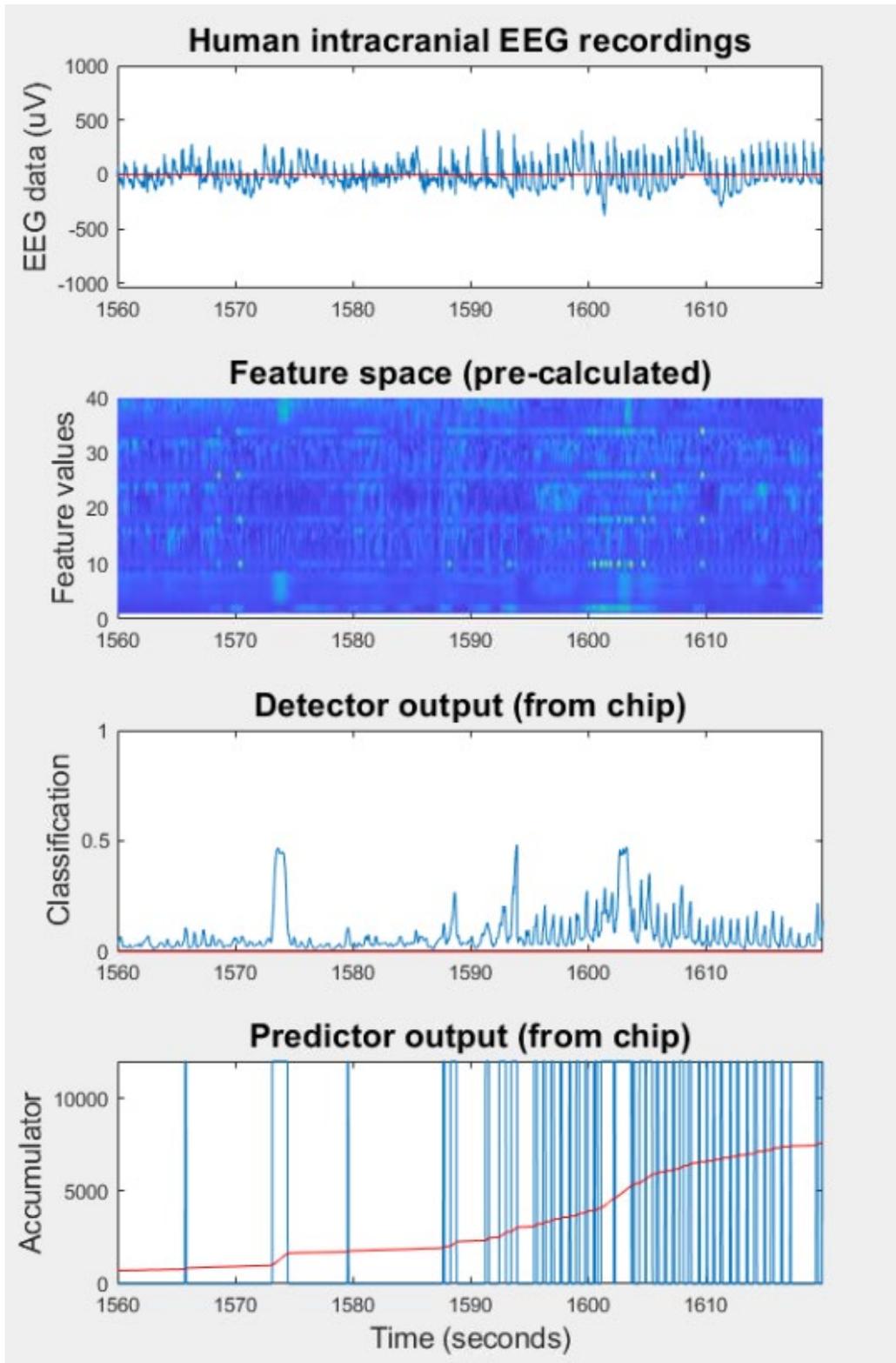
The previous sections focusing on seizure detection only utilized SPIRIT in detection-only mode. It has been stated in the previous chapter that in this mode, SPIRIT and SOUL function exactly the same. SOUL was the term used in the previous sections to indicate that we were only testing seizure detection. For this section, however, the full functionality of SPIRIT will be enabled (detection + prediction). The seizure prediction performance of SPIRIT was tested using the same datasets and using the exact same dataset splits as explained in Section A of this chapter, shown in Fig. 39. Fig. 45 shows a real-time SPIRIT classification showing both detection and prediction (through the prediction label accumulator) outputs. As also explained in that section, 30 minutes (since SPIRIT has a 30-minute prediction window) of the interictal dataset leading up to every seizure event is labeled as ‘1’ which then becomes the preictal dataset.

Table III shows a summary of the accuracy values for both the detection and prediction done by SPIRIT. Since SPIRIT utilizes the same seizure detection hardware as SOUL, the detection sensitivity and specificity remained the same. The prediction accuracy, given the 30-minute prediction window used by SPIRIT, was found to be 97.5% and 96.2%, in terms of sensitivity and specificity, for the CHB-MIT dataset, and 96.6% and 94.8% on the same metrics for the long-term iEEG dataset. Note that since the sensitivity of both the detector and predictor is the same for the CHB-MIT dataset, it means that the predictor was able to predict all of the seizures that were detected. Consequently, the 96.6% sensitivity of the predictor in the iEEG dataset implies that only 98.7% of all the seizures detected were correctly predicted (98.7% or the 97.9% detector sensitivity rate equals 96.6%). Finally, it is worth noting, that these results also rely on how much of a prediction window is actually being used, which was why there is a need to highlight that SPIRIT uses a 30-minute window. If the prediction window is large enough, all seizures will always be correctly predicted. However, setting a defined prediction time window allows for the identification of false predictions which makes the accuracy measurement more practical.

TABLE III  
SPIRIT SEIZURE DETECTION AND PREDICTION RESULTS FOR THE TWO DATASETS

CHB-MIT				iEEG		
Averages	Sensitivity	Specificity	Prediction time	Sensitivity	Specificity	Prediction time
<b>Detection</b>	97.5%	98.2%		97.9%	98.2%	
<b>Prediction</b>	97.5%	96.2%	8.4 minutes	96.6%	94.8%	7 minutes

As SPIRIT is a seizure predictor, another metric that can be presented is the prediction time, that is, how far ahead into the future can it correctly predict an event. Table III showed the average prediction time for the two datasets, 8.4 minutes and 7 minutes for the CHB-MIT and iEEG datasets respectively. Fig. 46 shows the maximum, minimum, and average values of these prediction times on a patient-per-patient basis, showcasing how the prediction times vary greatly from patient to patient. It is also worth noting that the prediction time maxes out at 30 minutes, as SPIRIT only considers this maximum window size. If there was supposed to be a correct prediction 31 minutes away, for example, this is treated as a false alarm instead as it is greater than the prediction window of SPIRIT.



**Fig. 45.** Real-time chip testing capture of SPIRIT. Logistic output probability for detection and label accumulator output for prediction is shown.

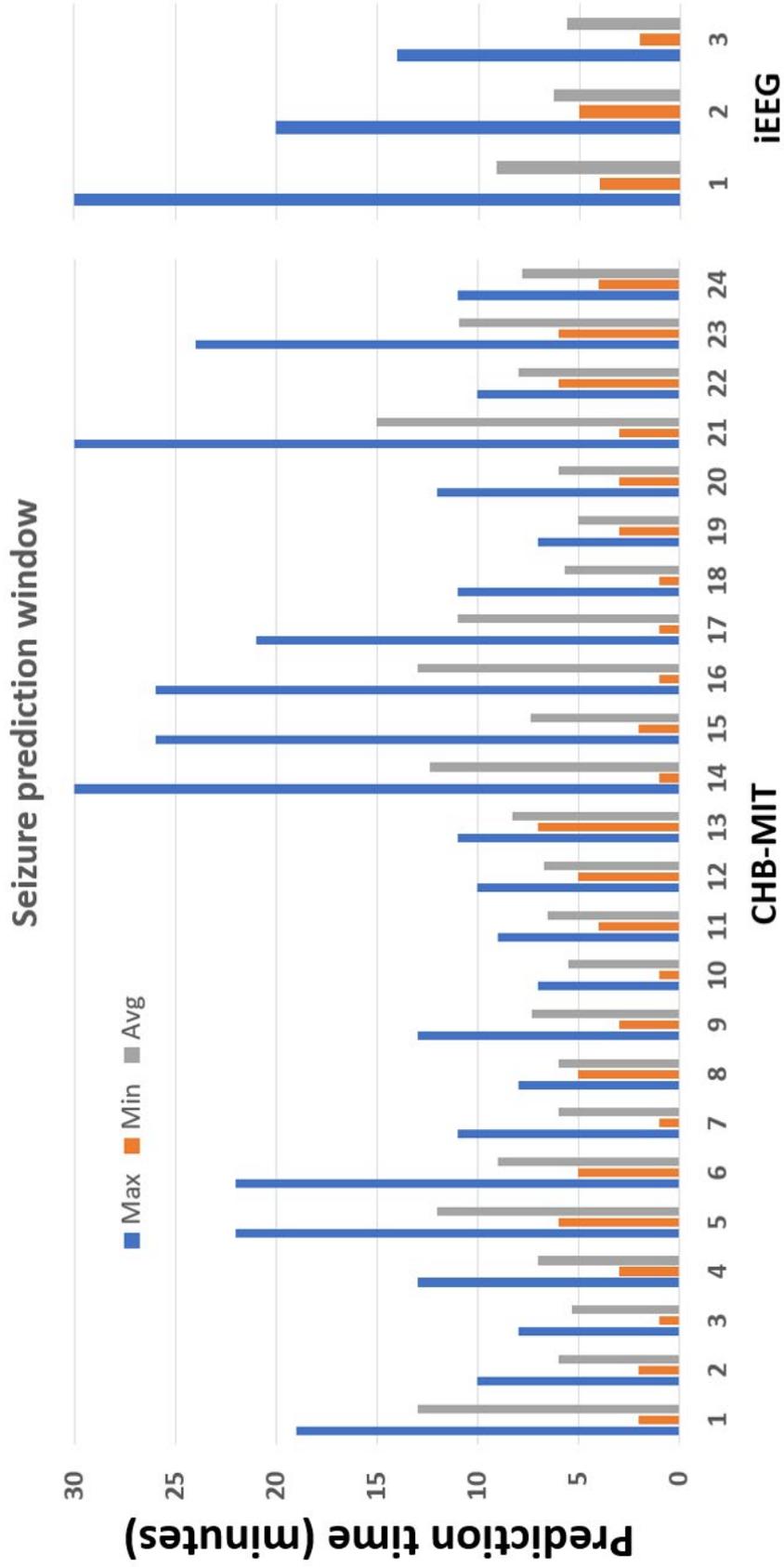


Fig. 46. SPIRIT patient-per-patient seizure prediction time statistics.

## VII. RESULTS AND DISCUSSIONS

### *A. Discussion on SOUL and SPIRIT chip results*

As discussed in Chapters IV and V, SOUL and SPIRIT were fabricated in TSMC’s 28 nm HPM process occupying 0.1 mm<sup>2</sup> and 0.14 mm<sup>2</sup> in area, respectively. For SOUL, the power consumption was measured to be 1.5  $\mu$ W. It managed to operate down to 0.5 V at 8 kHz clock frequency. This corresponds to an energy efficiency of 1.5 nJ/classification at a 1 kHz classification rate. With the additional hardware for seizure prediction and increased clock frequency, SPIRIT had an order of magnitude larger power consumption at 17.2  $\mu$ W. It only managed to operate down to 0.65 V mainly caused by the increased clock frequency of 24 kHz (3x higher than SOUL). This translates to an energy efficiency of 17.2 nJ/classification at the same 1 kHz classification rate. Since the power consumption is leakage-dominated in this regime, the described hardware reduction and reuse techniques presented in Chapters IV and V significantly impact the total power. Further power reduction could be achieved by implementing the classifier in a low-power (LP) process, instead of the high-performance mobile (HPM) variant that was used in this implementation. Digital logic and memory requirements were significantly reduced due to the relative computational simplicity of logistic regression coupled with architectural optimizations implemented to support online learning. The classifier-relevant memory is only 200 bytes for SOUL and about 2 kbytes for SPIRIT, mainly dominated by the feature weight values. Moreover, as both online learning classifiers use SGD and retrain in an unsupervised manner once deployed, only the current input features are used and stored at any given time. Consequently, there is no need to store a large amount of neural data for offline processing or training. When compared to the memory used for SVM-based systems [16, 20-24], typically used to store signal data and classifier parameters, SOUL and SPIRIT require 300x and 30x smaller memory respectively. A register-based memory implementation (instead of SRAM-based) was used for both online-learning classifiers, due to the very low memory requirements. Nevertheless, the classifier-relevant memory is still very negligible when compared to the pipeline registers required for the filters and feature extraction logic, which dominate the power consumption at about 70% for both implementations.

A comparison of this work with the state-of-the-art is shown in Table IV. For SOUL, the architectural optimizations lead to at least 10x lower area and 24x better energy efficiency compared to other on-chip state-of-the-art classifiers [13,25]. SPIRIT, on the other hand, still has about 2x energy efficiency compared to the most energy-efficient seizure detector implementation in the literature [25] (aside from SOUL). A work [45] has been published to claim the first on-chip implementation of a seizure prediction classifier that also has the capability to tune itself online, albeit supervised (i.e. requiring external labels). Yet, SPIRIT still is 5.6x more energy efficient and is 28x smaller in terms of area compared to it. This is attributed to both the algorithm choice and the hardware optimizations highlighted earlier. Thus, SPIRIT would be the first on-chip unsupervised online learning seizure predictor that is also the most energy and area efficient.

TABLE IV  
COMPARISON TABLE VS. RECENT STATE-OF-THE-ART ON-CHIP SEIZURE EVENT CLASSIFIERS

Dataset (# of patients)	JETCAS 2018 <sup>[13]</sup>	ISSCC 2020 <sup>[21]</sup>	JSSC 2020 <sup>[23]</sup>		ISSCC 2021 <sup>[46]</sup>	JSSC 2022 <sup>[24]</sup>		JSSC 2022 <sup>[26]</sup>		JSSC 2022 <sup>[36]</sup> SOUL		SPIRIT		
	iEEG (26)	EPILEPSIAE (-) <sup>(d)</sup>	MIT (24)	Local (2)	Bonn University (5)	MIT (24)	Local (1)	MIT (24)	iEEG (6)	iEEG (3)	MIT (24)	iEEG (3)	MIT (24)	
Channels	32	8	16	16	Programmable	16	16	256	8	8	8	8	8	
Classifier	Decision Trees	EDM-Brain Forest	Non-linear SVM		Reconfigurable Neural network	GTCA-SVM		NeuralTree (multi-class)		Logistic regression + SGD				
Online Training Algorithm (Labeling method)	X	X	ADMM (Supervised)		ALE (Supervised)	GTCA (Supervised)	X		Stochastic Gradient Descent (Unsupervised)					
Classification type (detect/predict)	detect	detect	detect	detect	detect	detect	detect	detect	detect	detect	detect	detect	predict	
Sensitivity (%)	83.7	96.7	96.6	96.1	99.84	97.8	71.9	95.6	94.0	98.6	92.0	97.9	96.6	97.5
Specificity (%)	88.1	- <sup>(e)</sup>	99.5	99.7	- <sup>(d)</sup>	99.5	96.4	96.8	96.9	99.7	99.1	98.2	94.8	96.2
Latency (s)	1.79	- <sup>(d)</sup>	0.71		- <sup>(d)</sup>	<1	<1	<1		- <sup>(d)</sup>	- <sup>(d)</sup>	2.6	1.6	-420 <sup>(f)</sup>
Technology (nm)	65	65	40	40	65	40	40	65	65	40	40	28	28	28
Supply voltage (V)		1.2	0.58		0.75	0.7		1.2		0.49		0.5		0.65
Clock frequency (kHz)		1,000	130, 65		2,500	1,000		128		6,050		8		24
Power ( $\mu$ W)	- <sup>(d)</sup>	9.6	1,900		32.1	- <sup>(d)</sup>		271 <sup>(c)</sup>		2,310		1.5		17.2
Classification rate (cls/s)		- <sup>(d)</sup>	- <sup>(d)</sup>		15.6	32		2,000		0.2		1,000		
Energy Efficiency (nJ/cls)	41.2	36	170,900		2,060	680 <sup>(c)</sup>		136 <sup>(c)</sup>		96.2 <sup>(e)</sup>		1.5		17.2
Classifier Area (mm <sup>2</sup> )	1	1 <sup>(b)</sup>	4.5		1.74	2.25 <sup>(b)</sup>		1.23 <sup>(b)</sup>		4		0.1		0.14

a: Reported 0.80 false alarms per hour  
b: Estimated from chip photo

c: DBE only, estimated from power breakdown  
d: Not reported

e: Not real-time (does not include feature extraction). Estimated 11.5 J/s considering 0.2 cls/s.  
f: Since this is predicting seizures, the effective latency (measured from start of seizure event) is negative.

## *B. Discussion on the use of logistic regression*

Among the cited works in Table IV, SOUL and SPIRIT were the only ones implemented using logistic regression. Neural networks, SVMs, and decision trees are well represented. Chapter II has covered their advantages and disadvantages. The logistic regression classifier used in SOUL and SPIRIT is the only generalized linear model type of classifier in recent literature. Linear models trade off simplicity with accuracy. Consequently, they will have difficulty with non-linearly separable data. More complex classifiers, such as the ones stated earlier, can achieve higher accuracies for such classification tasks. However, for recordings that do not have a very good linear separability between seizures and non-seizures, the online learning scheme presented in this thesis compensates. It exploits the fact that we tend to get very high accuracies on short data and thus we can use those data points to dynamically update the model as it runs over time. This online adaptation, which is based on stochastic gradient descent, enables the classifier to update the model towards optimality. Compare this to other simplified online tuning approaches [24] that might not correctly work over time since it is not based on mathematical optimization techniques. A gradient descent-based adaptation scheme has been implemented on an SVM [23] but is computationally complex. SOUL and SPIRIT leverage the fact that logistic regression gradient descent leads to a very simple formula. Thus, with a gradient descent-based update of logistic regression, the work presented here maintains high accuracy over time through online adaptation while still being very energy efficient.

It is also worth noting that the unsupervised approach in online learning is relatively robust. In this case, samples that are hard to classify would typically have logistic function output probabilities very near 0.5, which translates to low confidence. As shown in Chapter III. E, CT values are generally in the 0.7 to 0.8 range to represent high confidence. If the output probabilities are close to 0.5, then the retraining process cannot begin as a series of high confidence is not observed. Consequently, a non-linear classification task will not retrain SOUL in the wrong direction. Chapter III. F has also demonstrated that the unsupervised online learning approach is noise tolerant (and thus, the machine learning model is stable), mainly because the retraining process averages out the noise over time. While that depends on the initial accuracy achieved by the logistic regression after the initial offline training, it has been shown here that, at least for the datasets used in this work, the unsupervised online learning framework is feasible.

## *C. Discussion on seizure detection performance*

The classifier performance in terms of accuracy has already been reported in Chapter VI. It has been shown that SOUL (or SPIRIT – detection mode) performed well for seizure detection on both short-term (CHB-MIT) and long-term (iEEG) datasets with differing recording modalities and qualities (scalp and intracranial EEG respectively). Table IV also shows the accuracy results together with the most recent on-chip state-of-the-art seizure detection systems. Some cited works used different datasets that introduced partiality since the EEG signal recording quality can be different.

The classifier performance on long-term data (recording times ranging from several days to weeks) was not explicitly addressed in the other works. Many algorithms that work well on

short EEG recordings (i.e. within a day) may fail to work on longer recordings (i.e. several days). As SOUL has demonstrated that maintaining high accuracies over long periods of time is possible through SGD-based online learning, it would be interesting to see how the different online learning techniques from other implementations [23,24,45,46] would compare on the same long-term data.

The reported seizure detection latency for SOUL was 1.6-2.6 seconds which is relatively high when compared to the state-of-the-art. This can be attributed to the frequency-dependent group delay introduced by IIR filters on the feature extraction unit, which varies the spectral power feature values when it arrives at the classifier. This group delay can be compensated by cascading a corresponding phase equalizer after every IIR filter, which increases the filter hardware requirements by approximately 2x. The relatively high detection latency might also be a consequence of the limited feature set that was used since the feature selection process only selected features based on accuracy and not latency. Nevertheless, it has been shown [5,6] that latencies less than 5 seconds have demonstrated clinical efficacy in detection-triggered stimulation devices.

#### *D. Discussion on seizure prediction performance*

Table IV compared SPIRIT to the only other on-chip seizure prediction system in the literature [45]. It also has an online tuning capability that is done following an ADMM-based optimization for an SVM-based classifier. Chapter II. C has described that this type of optimization is indeed backed by mathematical theory, albeit computationally complex. The approach to online tuning is sound and would indeed push the SVM classifier toward optimality for every retraining that is performed. However, the tradeoff to support this approach is evident, as it consumes 2.31 mW of power compared to SPIRIT's 17.2  $\mu$ W (a 134x difference). In terms of area, the SVM-based predictor consumes 4 mm<sup>2</sup>, while SPIRIT only consumes 0.14 mm<sup>2</sup> (a 28x difference). The same work highlighted a 96.2 nJ/classification of energy efficiency which is, when compared to SPIRIT's 17.2 nJ/classification, is about 5.6x more.

It is important to note, however, that the 96.2 nJ/classification number that was reported is calculated by only considering the SVM latency when performing a classification and does not include the feature extraction latency. Therefore, it does not cover the entire classification process (as it also needs to include the feature extraction latency) and the reported number is not the real-time energy efficiency for the classifier. The work stated that the feature extraction latency is 5 seconds, which would imply a true classification rate of 0.2 classifications/sec. Consequently, this translates to a true energy efficiency of 11.5 J/classification for that work. That is about 6 orders of magnitude compared to SPIRIT's energy efficiency. This significant advantage of SPIRIT can be attributed to a number of considerations already reported in this thesis:

- 1) Algorithm choice of using logistic regression as the base classifier for SPIRIT, making the online retraining process computationally simple (Chapter III. B).
- 2) Feature extraction that is computed in a sliding window that matches the input sampling rate, leading to a very high classification rate (Chapter IV. B. Chapter V. B).
- 3) Architectural optimizations in SPIRIT, such as choosing simple features and significant hardware reuse, lead to very low power consumption (Chapters V. B and C).

## VIII. FUTURE WORK

It can be argued that reducing the digital backend power consumption might not offer a significant benefit in terms of overall system power when the analog front ends are included. However, given that the current implementation is significantly more energy-efficient than the state-of-the-art, this gives more room for more complex feature extraction units to be incorporated with the classification hardware. This can further improve the classifier performance, especially on non-linearly separable data, as well as the long-term performance of the unsupervised online learning scheme.

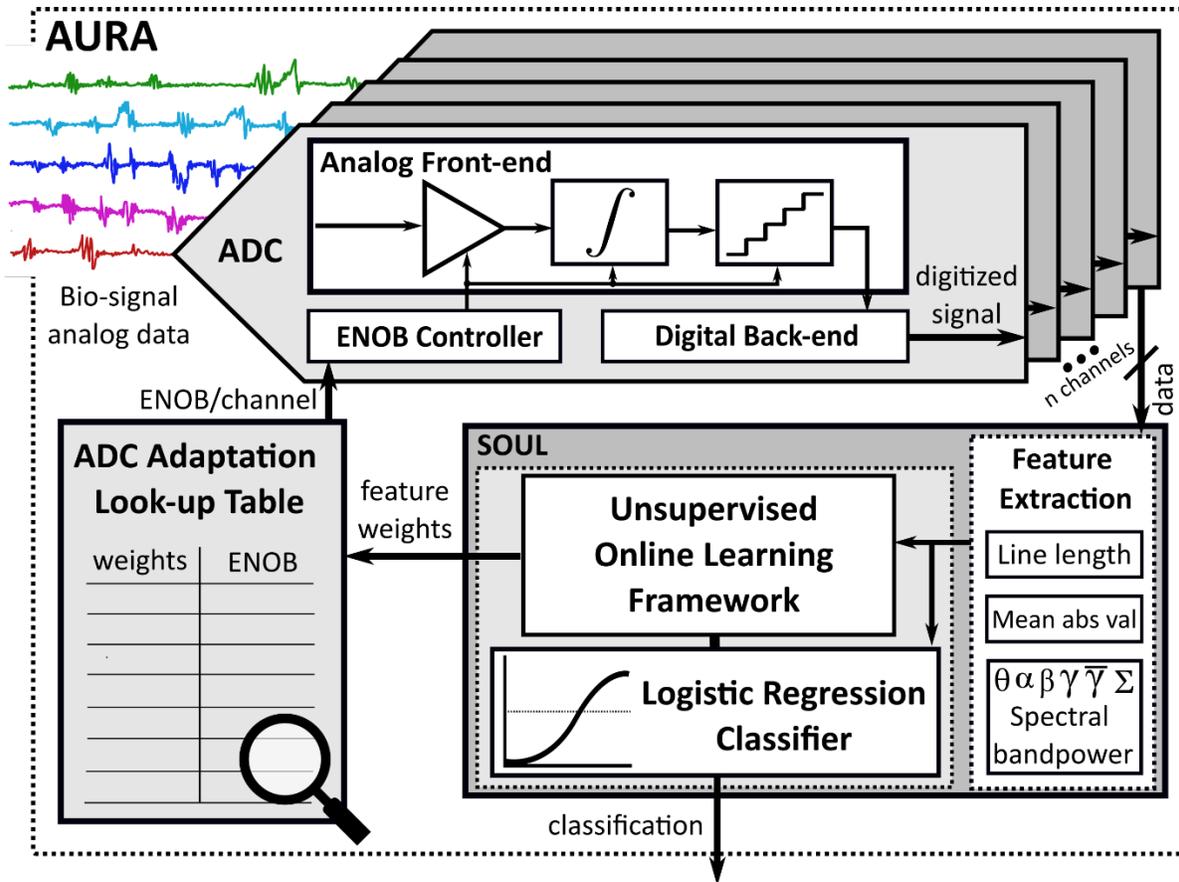
There is another benefit of using logistic regression as the main classifier that has not been highlighted that much in this thesis – the feature weights are, in fact, interpretable. The feature weights that are being dynamically updated throughout the online learning scheme of SOUL and SPIRIT inherently refer to the importance of different features in a given classification model. This is a consequence of logistic regression being a linear model. There is a relationship between the feature weights and how the model is supposed to be interpreted. The larger the feature weights are (in terms of magnitude), the more important those features become. Feature weights that tend to approach 0 imply that those features are not relevant at all (as the dot product of values will make the corresponding contribution of that specific feature to a model to be almost insignificant). Therefore, if these feature weights can be used to dynamically tune the analog front-ends, that is, the analog-to-digital converters (ADCs), then a more energy-efficient design can be achieved:

1) If the feature weights corresponding to a channel are very large, then that channel is important and should have the best ADC resolution possible.

2) If the feature weights corresponding to a channel are very small, then that channel is not that important. Therefore, the ADC resolution can be reduced further, saving power.

3) If the feature weights corresponding to a channel are (or near) 0, then that channel can be safely turned off since it does not contribute to the classification model anymore, significantly saving power.

Fig. 47 illustrates the concept. The feature weights can be mapped to a look-up table of ADC effective number of bits (ENOB). This ENOB mapping can be used by the ADC to change the corresponding resolution on a per-channel basis. As the feature weights are dynamically changing since SOUL is online learning, the ADC resolution can also dynamically change with the retrained classifier. This would enable the entire system, both analog and digital, to be more energy-efficient and application specific.



**Fig. 47.** Leveraging the interpretability of logistic regression classifier weights to dynamically tune the analog front-end (i.e. ADC resolution) such that the power consumption can be further reduced.

One of the limitations of this work was the use of pre-recorded datasets. Another possible future work would be to deploy the chip on an actual subject (say, an animal). This opens up questions on whether the algorithm is robust enough given the motion artifacts that may corrupt the EEG signals. This also enables much longer experiments and clinical trials to fully verify the capabilities of the chip described in this work. Another path to explore is using the classifier tuned for a different task, not only seizures. This would require a different set of features, more relevant to whatever the target application is, to be included with the unsupervised online learning framework. This would also show if the framework would only work for seizure detection or prediction, or can be used with other biological sensing as well (such as drowsiness detection, gesture recognition, etc.).

## IX. CONCLUSIONS

This thesis has demonstrated the capability of an unsupervised, online learning framework based on logistic regression and stochastic gradient descent to advance the state-of-the-art in long-term, high-accuracy, energy-efficient classification for seizure detection and prediction. Both of the on-chip implementations described in this work achieved the most energy-efficient classifier design in the literature to date to our knowledge. While logistic regression, on its own, is not the best-performing classifier for this task (which explains why it has not been used in the recent state-of-the-art), augmenting it with an optimization algorithm and energy-efficient design achieves high detection and prediction accuracies with the low area and energy consumption. While there has been a growing interest in online tuning algorithms in the literature, the two classifiers presented here are the only on-chip implementations to our knowledge that can do it in an unsupervised manner. Architectural optimizations also reduced the hardware requirements, leading to significantly less overall area and leakage power. The two classifiers presented here also have the smallest footprint for seizure detection and prediction on a chip.

SOUL is a logistic regression-based classifier, designed for seizure detection, that dynamically retrain itself using SGD without any external intervention. SOUL's performance has been evaluated on two datasets, for a total of 27 human subjects. For the long-term iEEG dataset, incorporating online learning results in an average sensitivity and specificity of 97.9% and 98.2% respectively, improving sensitivity by 6.5% on average with <1% specificity degradation over three patients. For the scalp EEG dataset, the classifier achieves 97.5% and 98.2% average sensitivity and specificity over 24 subjects. The sensitivity for the subjects either stayed the same (6/24) or improved (15/24) by 1-3%. Moreover, an improvement of >12% was observed on three subjects when compared against other state-of-the-art presenting a per-subject sensitivity breakdown.

Leveraging the accurate seizure detector, SPIRIT is also a logistic regression-based classifier, now designed for seizure prediction, that utilizes SOUL to train itself also without external intervention. The seizure predictor was also evaluated on the same datasets and achieved 96.6% average sensitivity and 94.8% average specificity for the long-term iEEG dataset, being able to predict incoming seizures up to 7 minutes (on average) before they begin. For the scalp EEG dataset, the predictor achieved a 97.5% average sensitivity and 96.2% average specificity, being able to predict an incoming seizure event 8.4 minutes (on average) in advance.

The significant benefit of the online learning approach is that the reported high accuracies were achieved on energy-efficient hardware. For SOUL, the combination of the proposed algorithmic approach and circuit-level optimizations resulted in an energy efficiency of 1.5 nJ/classification, which is at least 24x better than the state-of-the-art. It also consumes 0.1 mm<sup>2</sup> of area making it the smallest seizure detector classifier in the literature (by a factor of 10x). For SPIRIT, using the same architectural optimizations that made an energy-efficient SOUL, the energy efficiency for prediction was 17.2 nJ/classification, which is at least 5.6x better than the only other on-chip seizure predictor in the literature. Furthermore, SPIRIT's power consumption is at about 134x smaller at 17.2  $\mu$ W, while also being 28x smaller at 0.14 mm<sup>2</sup>. The energy efficiency of SPIRIT, combined with its unsupervised online learning capability, enables longer-lasting neuromodulation devices that require little to no maintenance from the user.

## REFERENCES

- [1] "Epilepsy," World Health Organization, Sep. 13, 2021. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/epilepsy>
- [2] Fisher, R. S., et al. (2005). "Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)." *Epilepsia* 46(4): 470-472.
- [3] Niedermeyer, E., & da Silva, F. L. (2004). "Electroencephalography: Basic Principles, Clinical Applications, and Related Fields." *Lippincott Williams & Wilkins*.
- [4] Echauz, J. et al., "Long-term validation of detection algorithms suitable for an implantable device", *Epilepsia*. pp. 35-36, Jan 2001.
- [5] "NeuroPace RNS System Clinical Summary", NeuroPace, June 2020. [Online]. Available: <https://www.neuropace.com/wp-content/uploads/2021/02/neuropace-rns-system-clinical-summary.pdf>
- [6] "Summary of safety and effectiveness data: NeuroPace RNS System." U.S. Food & Drug Administration, Nov. 14, 2013. [Online]. Available: [https://www.accessdata.fda.gov/cdrh\\_docs/pdf10/p100026b.pdf](https://www.accessdata.fda.gov/cdrh_docs/pdf10/p100026b.pdf).
- [7] "Premarket Approval (PMA): NeuroPace RNS System", U.S. Food & Drug Administration, Sep. 13, 2021. [Online]. Available: <https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfPMA/pma.cfm?id=P100026>
- [8] "Premarket Approval (PMA): Medtronic DBS Therapy for Epilepsy", U.S. Food & Drug Administration, Sep. 13, 2021. [Online]. Available: <https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfpma/pma.cfm?id=P960009S219>
- [9] "Clinical Outcomes: Epilepsy Deep Brain Stimulation", Medtronic, 2018. [Online]. Available: <https://www.medtronic.com/us-en/healthcare-professionals/therapies-procedures/neurological/deep-brain-stimulation/indications/epilepsy/clinical-outcomes.html>
- [10] P. J. Karoly, et al., "The circadian profile of epilepsy improves seizure forecasting", *Brain*, Vol. 140,8, Aug 2017, pp. 2169–2182.
- [11] M. Cook, "Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: a first in-man study.," *Lancet Neurology*, vol. 12, pp. 563–71, June 2013.
- [12] L. Kuhlmann et al., "Epilepsyecosystem.org: crowd-sourcing reproducible seizure prediction with long-term human intracranial EEG," *Brain*, vol. 141, pp. 2619–2630, August 2018.

- [13] M. Shoaran, et al., "Energy-Efficient Classification for Resource-Constrained Biomedical Applications," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 693-707, Dec. 2018.
- [14] Rashid M, Sulaiman N, P P Abdul Majeed A, et al. "Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review." *Front Neurobot.* 2020;14:25. Published 2020 Jun 3.
- [15] M. A. B. Altaf, J. Tillak, Y. Kifle and J. Yoo, "A 1.83 $\mu$ J/classification nonlinear support-vector-machine-based patient-specific seizure classification SoC," *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, San Francisco, CA, USA, 2013, pp. 100-101.
- [16] C. Zhang, et al., "Design and Implementation of an On-Chip Patient-Specific Closed-Loop Seizure Onset and Termination Detection System," in *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 4, pp. 996-1007, July 2016.
- [17] Ripley B. "Pattern Recognition and Neural Networks." *Cambridge University Press*; 1996.
- [18] C. Li, C. Lammie, X. Dong, A. Amirsoleimani, M. R. Azghadi and R. Genov, "Seizure Detection and Prediction by Parallel Memristive Convolutional Neural Networks," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 4, pp. 609-625, Aug. 2022.
- [19] Evgeniou T, Pontil M. "Support Vector Machines: Theory and Applications. In: Lecture Notes in Computer Science." Vol 1933. *Springer*; 2001:249-257.
- [20] J. Yoo, et al., "An 8-Channel Scalable EEG Acquisition SoC With Patient-Specific Seizure Classification and Recording Processor," in *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 214-228, Jan. 2013.
- [21] G. O'Leary et al., "A recursive-memory brain-state classifier with 32-channel track-and-zoom  $\Delta\Sigma$  ADCs and Charge-Balanced Programmable Waveform Neurostimulators," *2018 IEEE International Solid - State Circuits Conference - (ISSCC), 2018*, pp. 296-298.
- [22] Y. Wang, et al., "A Closed-Loop Neuromodulation Chipset with 2-Level Classification Achieving 1.5Vpp CM Interference Tolerance, 35dB Stimulation Artifact Rejection in 0.5ms and 97.8% Sensitivity Seizure Detection," *2020 IEEE International Solid- State Circuits Conference - (ISSCC), 2020*, pp. 406-408.
- [23] S. Huang, et al., "A 1.9-mW SVM Processor With On-Chip Active Learning for Epileptic Seizure Control," in *IEEE Journal of Solid-State Circuits*, vol. 55, no. 2, pp. 452-464, Feb. 2020.
- [24] M. Zhang, L. Zhang, C. -W. Tsai and J. Yoo, "A Patient-Specific Closed-Loop Epilepsy Management SoC With One-Shot Learning and Online Tuning," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1049-1060, April 2022.

- [25] G. O'Leary et al., "A Neuromorphic Multiplier-Less Bit-Serial Weight-Memory-Optimized 1024-Tree Brain-State Classifier and Neuromodulation SoC with an 8-Channel Noise-Shaping SAR ADC Array," *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2020, pp. 402-404.
- [26] U. Shin et al., "NeuralTree: A 256-Channel 0.227- $\mu$ J/Class Versatile Neural Activity Classification and Closed-Loop Neuromodulation SoC," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3243-3257, Nov. 2022.
- [27] Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3–29.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
- [29] S.-A. Huang and C.-H. Yang, "A hardware-efficient ADMM-based SVM training algorithm for edge computing," Jul. 2019, arXiv:1907.09916. [Online]. Available: <https://arxiv.org/abs/1907.09916>.
- [30] Kleinbaum D, Klein M. "Logistic Regression. A Self-Learning Text." *Springer*; 1994.
- [31] A. Page, et al., "A Flexible Multichannel EEG Feature Extractor and Classifier for Seizure Detection," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 2, pp. 109-113, Feb. 2015.
- [32] S. Chen, et al., "Automatic Diagnosis of Epileptic Seizure in Electroencephalography Signals Using Nonlinear Dynamics Features," in *IEEE Access*, vol. 7, pp. 61046-61056, 2019.
- [33] Y. Li, et al., "Epileptic Seizure Classification of EEGs Using Time-Frequency Analysis Based Multiscale Radial Basis Functions," in *IEEE Journal of Biomedical and Health Informatics*, pp. 386-397, March 2018.
- [34] Kiefer, J., and J. Wolfowitz. "Stochastic Estimation of the Maximum of a Regression Function." *The Annals of Mathematical Statistics*, vol. 23, no. 3, 1952, pp. 462–466.
- [35] Bottou L. "Large-Scale Machine Learning with Stochastic Gradient Descent." *In: Proceedings of COMPSTAT'2010*. Springer; 2010:177-186.
- [36] A. Chua, et al., "Unsupervised Online Learning for Long-Term High Sensitivity Seizure Detection," *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 528-531.
- [37] A. Chua, et al., "A 1.5nJ/cls Unsupervised Online Learning Classifier for Seizure Detection," *2021 Symposium on VLSI Circuits, 2021*, pp. 1-2.

- [38] L. Logesparan, et al., "Optimal features for online seizure detection," *Medical & Biological Engineering & Computing*, pp. 659–669, Jul 2012.
- [39] Mormann F, Andrzejak RG, Elger CE, Lehnertz K. "Seizure prediction: the long and winding road." *Brain*. 2007;130(Pt 2):314-333.
- [40] Alotaiby, T.N., Alshebeili, S.A., Alshawi, T. et al. "EEG seizure detection and prediction algorithms: a survey." *EURASIP J. Adv. Signal Process.* 2014, 183 (2014).
- [41] Kiral-Kornek I, Roy S, Nurse E, et al. "Epileptic Seizure Prediction Using Big Data and Deep Learning: Toward a Mobile System." *EBioMedicine*. 2018;27:103-111.
- [42] Z. Zhang and K. K. Parhi, "Low-Complexity Seizure Prediction From iEEG/sEEG Using Spectral Power and Ratios of Spectral Power," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 693-706, June 2016.
- [43] K. K. Parhi and Z. Zhang, "Discriminative Ratio of Spectral Power and Relative Power Features Derived via Frequency-Domain Model Ratio With Application to Seizure Prediction," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 4, pp. 645-657, Aug. 2019.
- [44] Goldberger, A., et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.," *Circulation [Online]*, pp. e215–e220.
- [45] Y. -Y. Hsieh, Y. -C. Lin and C. -H. Yang, "A 96.2-nJ/class Neural Signal Processor With Adaptable Intelligence for Seizure Prediction," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 1, pp. 167-176, Jan. 2023
- [46] J. Liu et al., "4.5 BioAIP: A Reconfigurable Biomedical AI Processor with Adaptive Learning for Versatile Intelligent Health Monitoring," *2021 IEEE International Solid- State Circuits Conference (ISSCC), 2021*, pp. 62-64.

## APPENDIX

This is the mathematical derivation on how the stochastic gradient descent can be applied to logistic regression leading to the simple weight update formula presented in this work.

$$p_i = s(\mathbf{w}^\top \mathbf{x}_i) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}_i}}$$

Now we can estimate the parameters  $\mathbf{w}$  via maximum likelihood. We have the problem

$$\begin{aligned} \hat{\mathbf{w}}_{\text{LR}} &= \arg \max_{\mathbf{w}} P(\hat{Y}_1 = y_1, \dots, \hat{Y}_n = y_n \mid \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{w}) \\ &= \arg \max_{\mathbf{w}} \prod_{i=1}^n P(\hat{Y}_i = y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \arg \max_{\mathbf{w}} \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{(1-y_i)} \\ &= \arg \max_{\mathbf{w}} \ln \left[ \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{(1-y_i)} \right] \\ &= \arg \max_{\mathbf{w}} \sum_{i=1}^n y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \\ &= \arg \min_{\mathbf{w}} - \sum_{i=1}^n y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \\ \nabla_{\mathbf{w}} L(\mathbf{w}) &= \nabla_{\mathbf{w}} \left( - \sum_{i=1}^n y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \right) \\ &= - \sum_{i=1}^n y_i \nabla_{\mathbf{w}} \ln p_i + (1 - y_i) \nabla_{\mathbf{w}} \ln(1 - p_i) \\ &= - \sum_{i=1}^n \frac{y_i}{p_i} \nabla_{\mathbf{w}} p_i - \frac{1 - y_i}{1 - p_i} \nabla_{\mathbf{w}} p_i \\ &= - \sum_{i=1}^n \left( \frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i} \right) \nabla_{\mathbf{w}} p_i \end{aligned}$$

Note that  $\nabla_{\mathbf{z}} s(\mathbf{z}) = s(\mathbf{z})(1 - s(\mathbf{z}))$ , and from the chain rule we have that

$$\nabla_{\mathbf{w}} p_i = \nabla_{\mathbf{w}} s(\mathbf{w}^\top \mathbf{x}_i) = s(\mathbf{w}^\top \mathbf{x}_i)(1 - s(\mathbf{w}^\top \mathbf{x}_i)) \mathbf{x}_i = p_i(1 - p_i) \mathbf{x}_i$$

Plugging in this gradient value, we have

$$\begin{aligned}\nabla_{\mathbf{w}}L(\mathbf{w}) &= -\sum_{i=1}^n \left( \frac{y_i}{p_i} - \frac{1-y_i}{1-p_i} \right) \nabla_{\mathbf{w}}p_i \\ &= -\sum_{i=1}^n \left( \frac{y_i}{p_i} - \frac{1-y_i}{1-p_i} \right) p_i(1-p_i)\mathbf{x}_i \\ &= -\sum_{i=1}^n (y_i(1-p_i) - (1-y_i)(p_i)) \mathbf{x}_i \\ &= -\sum_{i=1}^n (y_i - p_i) \mathbf{x}_i\end{aligned}$$

The gradient descent update is thus

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \epsilon \sum_{i=1}^n (y_i - p_i) \mathbf{x}_i$$

The final result would now resemble the SGD formula as presented in the text.

$$w_{t+1} = w_t + \eta(y_t - p(w_t, x_t))x_t$$