Detection of Epileptic Seizures from Surface EEG Using Hyperdimensional Computing

Fatemeh Asgarinejad, Anthony Thomas, and Tajana Rosing
University of California San Diego, La Jolla, CA 92093, USA
{ahthomas, tajana}@ucsd.edu

Abstract— **Recent years have seen a growing interest in the development of non-invasive devices capable of detecting seizures which can be worn in everyday life. Such devices must be lightweight and unobtrusive which severely limit their on-board computing power and battery life. In this paper, we propose a novel technique based on hyperdimensional (HD) computing to detect epileptic seizures from 2-channel surface EEG recordings. The proposed technique eliminates the need for complicated feature extraction techniques required in conventional ML algorithms. The HD algorithm is also simple to implement and does not require expert knowledge for architectural optimizations needed for approaches based on neural networks. In addition, our proposed technique is light-weight and meets the computation and memory constraints of ultra-small devices. Experimental results on a publicly available dataset indicates our approach improves the accuracy compared to state-of-the-art techniques while consuming smaller or comparable power.**

I. INTRODUCTION

Epilepsy is a serious neurological condition affecting over 65 million people worldwide [1]. Epilepsy is marked by persistent seizures which vary substantially in severity and length: mild seizures may not even be noticeable while severe seizures can be accompanied by memory loss or convulsions which can cause injury [1]. Because the symptoms of epilepsy vary considerably between patients, doctors typically conduct a range of tests to characterize symptoms prior to deciding on a course of treatment. Measuring the frequency and duration of seizure events is an important piece of information both in deciding on a course of treatment and on evaluating treatment effectiveness [2]. Traditionally, these measurements have been gathered in hospital settings through use of an *electroencephelogram* (EEG) which records the electrical output of the brain and can be used to detect patterns of abnormal electrical activity characteristic of epileptic seizures. While such methods are accurate, they are costly, requiring extended hospital stays by patients and time-consuming analysis of EEG recordings by neurologists. Accordingly, recent years have seen growing interest in the development of non-invasive devices capable of detecting seizures which can be worn in everyday life [3].

The development of such devices poses a significant challenge both from a system design and signal processing perspective. In order to be aesthetically appealing and practical, devices must be lightweight and unobtrusive which limits their on-board computing power and battery life. In this setting, many state-of-the-art detection algorithms (for instance [4]) may be infeasible due to excessive memory and/or power requirements. Another option is to stream data collected on the wearable device to a mobile computing platform like a Raspberry-Pi which is capable of more sophisticated computations [5]. Nonetheless, these devices still face limitations in terms of battery life and computing power and so optimizing the signal processing algorithm is still desirable. In this work, we focus on the latter setting.

In addition, EEG readings collected in clinical settings are typically gathered from electrodes distributed over the entire surface of the skull. By contrast, a wearable device may only be able to gather reading from one or two electrodes at specific locations which may limit the ability of the system to detect seizures characterized by activity in other areas of the brain [3].

Hyperdimensional (HD) computing is a promising new technique for addressing machine learning problems on resource constrained devices. HD computing emerged from models of human memory developed in the computational neuroscience community [6] and embeds raw signal values into a (very) high dimensional vector space under a random mapping. The high dimensional representations can be combined using simple arithmetic operations to generate composite representations for related samples of a signal (e.g. classes) which can be used in learning algorithms. HD computing is appealing from an algorithmic perspective as it requires only primitive element-wise arithmetic operations over vectors and can be efficiently trained using a simple perceptron style algorithm [7]. Additionally, in contrast to classical signal-processing approaches that need hand-designed features to be extracted from the signal (which may fail to capture relevant information), HD can be applied on raw signals. HD computing has gained significant popularity in the hardware community in recent years as an energy efficient alternative to classical machine learning algorithms [7], [8], [9].

Recent work in [10], [11] applied HD computing to seizure detection on full-channel *intracranial* EEG (iEEG) recordings and found that HD based methods were able to deliver comparable or superior accuracy to approaches based on state-of-the-art machine learning algorithms - support vector machines and neural networks - while requiring less memory and fewer training examples. However, the *intracranial* EEG recordings used in these works require surgery to place electrodes directly on the brain and provide access to [xx] channels.

In this work, we present an algorithm for detection of epileptic seizures from 2-channel *surface* EEG recordings on lightweight embedded devices based on HD computing. To the best of our knowledge we are the first to apply HD computing to the problem of seizure detection using non-invasive *surface* EEG recordings. We present an extensive comparison of our approach against other methods, in particular a "state-of-the-art" algorithm based on a Convolutional Neural Network (CNN) proposed in [12], and a recently proposed algorithm for detection on a lightweight device proposed in [3]. We find that our proposed method achieves an average accuracy of 82.99%, which outperforms the CNN by 6.43%, and random forest by 11.84%.

II. MATERIALS AND METHODS

A. EEG dataset and preprocessing

In this work, we follow the setup of [3] who recently proposed a lightweight wearable device for seizure monitoring. Their device consists of a pair of eyeglasses with EEG electrodes sitting at either temple. We emulate data gathered from such a device using a publicly available dataset collected by the Children's Hospital of Boston and MIT [13]. The dataset consists of 22 subjects with medically resistant seizures ranging in age from 1.5 to 22 years. As in [3], we use data on the first ten patients for which data was gathered in compliance with the acquisition specification outlined in [14]. To be consistent with the device proposed in [3] we consider only data from the F_7T_7 and F_8T_8 electrodes. Overall, we observe a total of 55 seizures in the dataset. As is common in biosignal processing, we partition the EEG signal into non-overlapping windows of length w and treat each window as a single "observation" when training algorithms.

B. Hyperdimensional Computing

Overview. HD computing is based on evidence from the neuroscience community which suggests that the human brain computes on high-dimensional randomized representations of data rather than scalar numerical values [6]. In this literature, the high dimensional space is typically referred to a *hyperspace* - to emphasize its unusually high dimensionality - and points in the hyperspace are referred to as *hypervectors*. The dimension of the hyperspace is denoted H and is typically over $5,000$. The elements of hypervectors are typically either bits (e.g. `0,1`) or integers. Real and complex numbers are not inherently unsupported but generally avoided for computational reasons.

Because of their high-dimensionality, any randomly chosen pair of hypervectors will be nearly orthogonal with high-probability. A useful consequence of this fact is that, for any some collection of hypervectors p, q, v, the vector represented by their superposition $S = p + q + v$ is, with high-probability, closer to p, q and v than any other randomly chosen vector in the space. Thus, sets (e.g. examples corresponding to a particular class) can be represented simply by computing an element-wise sum over the individual hypervectors.

Given a vector z we can test if z, or a similar pattern, is contained in S simply by computing a suitable similarity metric $\alpha = \delta(\boldsymbol{z}, \boldsymbol{S})$ and comparing α to some critical value. The similarity metric is typically the dot-product or cosine distance. It is straightforward to extend this general approach to encode sequences of observations (e.g. a time-series) [15], [6].

Development of an HD model consists of three stages: *encoding*, *training*, and *querying*. Encoding refers to the process of obtaining a high-dimensional representation from a raw signal. Training refers to the process of agglomerating training data into some predictive model, and querying refers to the process of predicting a label for some new piece of unlabeled data. Throughout this description we assume the training data consists of a set of N labeled examples $\{\boldsymbol{x}_i, y_i\}_{i=0}^{N-1}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{0, 1\}$, where D denotes the dimensionality of the input data to the learning algorithm

Encoding. The encoding step maps a window of the raw signal in low-dimensional space to its high-dimensional representation. In this work we use the general purpose encoding method proposed in [7]. As this material is not yet standard fare, we here briefly review the essentials.

Let $\boldsymbol{x} = \{x_0, x_1, ..., x_D\} \in \mathbb{R}^D$ be a d-dimensional input vector. In general, the input vector may be raw signal values or features extracted from the signal. Our goal is to represent this data as a vector $\boldsymbol{h} \in \mathbb{Z}^H$ where $H \gg D$ and $\mathbb{Z}$ denotes the integers.

We first quantize the support of each x_i into Q bins. We then generate "level-hypervectors" representing each quantizer bin as follows. The level hypervector for the first quantization bin $\boldsymbol{l}_0 \in \{\pm 1\}^H$ is generated by randomly sampling from the uniform distribution over $\{\pm 1\}^H$. The level hypervector for the second bin $\boldsymbol{l}_1$ is generated by randomly flipping $\lfloor \frac{H}{2Q} \rfloor$ randomly chosen coordinates. For example, if $H = 10,000$ and $Q = 100$, then $\boldsymbol{l}_0$ and $\boldsymbol{l}_1$ will differ (in expectation) on 200 coordinates. Similarly, $\boldsymbol{l}_2$ is generated by flipping $\lfloor \frac{H}{2Q} \rfloor$ random coordinates of $\boldsymbol{l}_1$. Thus, the expected similarity (dot-product) between $\boldsymbol{l}_0$ and $\boldsymbol{l}_j$ gradually decays to zero as $j \to Q$. Intuitively, this ensures that if two different data points are similar in the input space, they are also similar (as measured by dot product) in the high-dimensional space. We denote the set of all level hypervectors $\mathcal{L} = \{\boldsymbol{l}_j\}_{j=0}^{Q-1}$.

In the case of EEG analysis, $\boldsymbol{x}$ represents a sequence of EEG readings (e.g. a time-series), hence the sequential ordering of the $x_i \in \boldsymbol{x}$ must be taken to ensure information about the ordering of the x_i is preserved in the high-dimensional encoding. To do so, we use a permutation operation that differentiates the temporal positions of features by assigning a unique permutation for each index. Let $q(x_i)$ denote a function which takes a coordinate $x_i \in \boldsymbol{x}$ and returns the level hypervector $\boldsymbol{l}_j$ corresponding to the appropriate quantization bin. Then encoding of a window $\boldsymbol{x}$ proceeds as follows:

$$\boldsymbol{h} = \sum_{i=0}^{D-1} q(x_i) * \rho^i \tag{1}$$

where $*\rho^i$ denotes left-rotation of $q(x_i)$ by i indices. That is, for all $x_i \in \boldsymbol{x}$, the corresponding level hypervector is decided, rotated left by i, and accumulated to realize $\boldsymbol{h}$.

Training. Given encoded hypervectors, training an HD model is extremely simple. As noted above, to generate a single hypervector representing a class k – which we denote $\boldsymbol{c}_k$ – we need only sum the encoded windows corresponding to that class. More formally, the class hypervector for class k is obtained as follows:

$$\boldsymbol{c}_k = \sum_{\boldsymbol{x}_i \text{ s.t. } y_i = k} enc(\boldsymbol{x}_i) \tag{2}$$

Where "enc" is the encoding function described above. We remark that HD training is advantageous as it does not require complex optimization algorithms as in other conventional machine learning techniques like SVMs or neural networks.

Querying. Given a trained HD model, the ultimate goal is typically to use the model to infer the label of a new data point. Suppose $\boldsymbol{x}_i$ is a new data point for which we do not know the label. To predict a label, we simply encode it to an HD representation using exactly the same procedure as during training and compute the cosine similarity of the resulting vector with each of the "class hypervectors" learned during training and return the class which maximizes this similarity. More formally, the class label for a query example $\boldsymbol{x}_q$ is computed as:

$$\underset{k \in 1:K}{\text{argmax}} \frac{\langle enc(\boldsymbol{x}_q), \boldsymbol{c}_k \rangle}{||enc(\boldsymbol{x}_i)|| ||\boldsymbol{c}_k||} \tag{3}$$

where $\langle \star, \star \rangle$ denotes the inner product.

III. EXPERIMENTAL RESULTS

In the following section we present an extensive series of tests to evaluate the effectiveness of HD compute for seizure detection on surface EEG. We focus primarily on comparing the accuracy and energy efficiency of HD computing to other widely used machine learning techniques.

A. Baseline Methods

While a multitude of algorithms for seizure detection have been proposed in the literature, we here focus on a handful of the most popular techniques, namely: K-nearest neighbors (KNN), support vector machines (SVM), logistic regression, random forests (RF) and convolutional neural networks (CNNs). For a thorough overview of existing approaches the interested reader is referred to a recent survey in [16]. With the exception of CNNs and HD, these algorithms all typically require that features to be extracted from the signal prior to training. As noted above, feature extraction for seizure detection is itself a major topic of research and we here limit ourselves to a handful of features commonly cited in literature [16].

When considering methods which require feature extraction, we follow the analysis method outlined in [3] and preprocess the signal with a wavelet decomposition and then extract several different entropy features including sample, permutation, Renyi, Shannon, and Tsallis entropies. We remark that our results provide evidence that HD is more robust to noise and eliminates the need for preprocessing on the raw signal values.

In addition to the entropy features described above, we also extract several features from the frequency domain representation of the signal. For each signal window, we compute the power spectral density using Welch's method [17] and extract the relative power in the five brain wave frequency bins ($\delta : 0.5-4$Hz, $\theta : 4-8$Hz, $\alpha : 8-12$Hz, $\beta : 12-30$Hz and $\gamma : 30-45Hz$) as well as a low frequency bin from $0-0.5Hz$. These features are commonly held to be medically relevant for detecting seizures [18].

Finally, as a baseline for the CNN, we consider the method proposed in [12] which consists of a single convolution layer with six 5×5 filters and ReLU activation. The conv layers are followed by a 2×2 pooling layer and a fully connected layer. The input of the CNN has a square ($k\times k\times 2$) array, which we obtain by reshaping the recorded EEG signals of the channel in certain window lengths. We examined an exhaustive grid of hyper parameters including different optimizers (Adadelta, SGD, Adam), learning rates (e-1, e-3, e-5, e-7), batch sizes (5, 10, 20, 32, 50), epochs (10, 25, 50, 75, 100) and window sizes (e.g. k) (24×24, 32×32, 46×46, 64×64, representing 1, 2, 4, and 8 seconds). Similarly for the other ML techniques, hyperparameters were set using grid search over a validation set. Reported results are obtained from the model yielding the best performance on the validation data.

We split the training and test datasets by 75%-25% ratios of total continuous windows. We further reserve 25% of the training data as a validation set for hyperparameter tuning. We make sure adjacent windows are not distributed into both training and test to avoid bias in our test set. We repeat all experiments 100 times, re-sampling the train, test, and validation sets on each iteration and report the median, 25th and 75th percentiles of results on all trials.

B. Performance Evaluation

Sensitivity and *specificity* are standard metrics to evaluate the performance of seizure detection algorithms [12], [3]. Sensitivity indicates the ratio of correctly detected positive (ictal) labels to total positive labels, i.e., sensitivity $= \frac{\text{tp}}{\text{tp+fn}}$. On the other hand, specificity denotes the detection performance of false (interictal) labels, i.e., specificity $= \frac{\text{tn}}{\text{tn+fp}}$. We define the overall performance of the model as the geometric mean of sensitivity and specificity. The geometric mean penalizes the performance if either sensitivity or specificity is skewed: performance $=$ (sensitivity $\times$ specificity)$^{\frac{1}{2}}$.

Figure 1 compares the accuracy of our proposed HD-based seizure detection with other ML techniques, particularly random forest of [3] and CNN [12] using 1-second

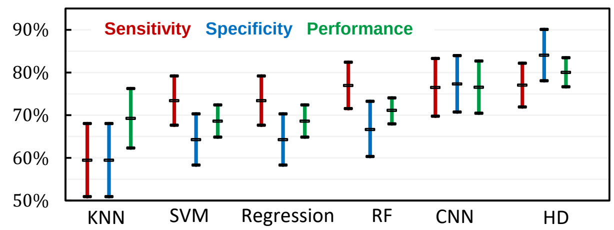

Fig. 1. Comparing the sensitivity, specificity, and overall performance of HD and baseline techniques on 1-second windows. Each bar represents 25^{th} to 75^{th} percentile as well as the median (of 100 train/test experiments).

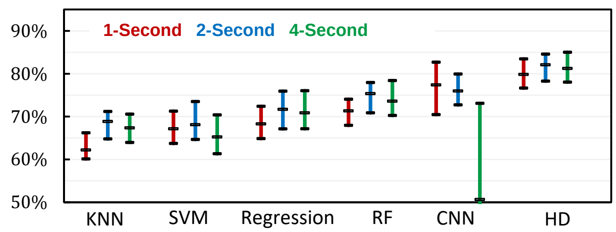

Fig. 2. Impact of EEG window size on overall performance of the models. Each bar shows the 25^{th} to 75^{th} percentile range.

EEG windows. The overall performance of HD surpasses these methods, achieving an average performance of 81.69%, which is by 5.13% and 10.54% better than that of CNN [12] and random forest (RF) in the recent work [3], respectively.

Figure 2 shows the impact of increasing the EEG window length on the overall performance of the models. Except the CNN [12], the other techniques generally benefit from increasing the window size as they can absorb more information. Despite considerable hyperparameter tuning of the CNN (see Section III-A), we find it outputs a constant 0 (or 1) when large windows are used, which results in poor overall performance. Using 4-second windows, the 25th, average, and 75th percentiles of HD's overall performance increases to 78.02%, 82.57%, and 85.00%. Among the baseline models, CNN using 1-second windows still provides the highest average performance of 76.56% (even compared to 4-second windows of the other models). Thus, compared with this 'best' baseline, HD improves the average accuracy by 6.01%.

We also observed that HD's accuracy improves up to window sizes of six seconds (not shown in Figure 2) and then degrades. This is because the amount of information that a hypervector can retain saturates. Note that a window size of six seconds has $D = 6 \times f_s \times 2$ channels $= 3,072$ input vector dimension which is significant compared to the length of hypervectors $H = 5,000$ we considered in our experiments (as mentioned in Section II-B, H needs to be much larger than D). Using 6-second windows, the average accuracy of HD reaches 82.99% and surpasses the state-of-the-art CNN [12] by 6.43%, and the random forest [3] by 11.84%. We attribute the difference in performance between our work and [3] to a different methodology for partitioning the data into train and test set and note that on some runs we observe comparable (or superior) performance to their results.

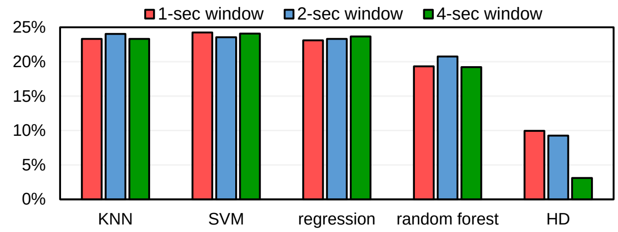

Fig. 3. Power overhead of different techniques (with different window lengths) compared to CNN.

Lastly, we find the performance of HD is comparable with and without pre-processing of the signal to remove noise. Remember that in our experiments we used raw signals without any pre-processing such as signal denoising or bandpass filter. To examine the effectiveness of HD against noisy signals, we repeated our experiments by passing all windows through a Butterworth bandpass filter in the range [0:5−45Hz] before further processing. Using signal windows of one to eight seconds, we observed that exploiting a bandpass filter only improves the HD performance by 0.29%, on average, indicating even in the presence of noise HD could deliver the maximum accuracy. This suggests that HD computing may help improve tolerance to underlying noise in data.

C. Power Consumption and Execution Time

We implemented our proposed technique and the baseline techniques using Python on Raspberry-Pi 3. We used Hioki 3334 power meter to measure the power consumption. We realized the baseline light-weight CNN [12] consumes the least power as it requires smallest amount of computation. We thus normalized the power consumption of the other techniques to CNN power. Figure 3 compares the power consumption of different techniques for three different window sizes. Techniques based on feature extraction, i.e., KNN, SVM, linear regression and random forest almost consume the same amount of power (correlates with computations) due to the complicated feature extraction step. The power overhead of HD reduces as the window size increases. Note that for larger window size, HD requires more computation. However, the time between arrival of windows also increases and on overall the device benefits more ratio of idle time. With 4-second windows, HD only consumes 3.1% higher power than CNN, and further decreases if 6-second windows are used for HD (which we also showed that achieves the highest accuracy).

As mentioned above, for the baseline techniques other than CNN (i.e., KNN, SVM, regression and random forest) the Raspberry-Pi CPU is fully ($> 96\%$) occupied in the entire duration of processing an EEG window. In this sense, detecting a w-second window takes $\sim 2w$ seconds: w seconds for receiving/recording the window, and $\sim w$ seconds for processing. Thus, although these techniques can benefit accuracy improvement by increasing the window length, the detection time will increase accordingly. For HD, the detection time also depends on the number of input dimensions. However, we observed that processing a w-second window takes less than $0.1w$. The system can therefore detect a seizure after 0.4 second, if 4-second windows are used. It also implies that the HD implementation can be ported to ultra-light devices with $\sim 0.1\times$ computation capability of Raspberry-Pi. Moreover, recent studies have shown that HD, thanks to its bit-level and highly parallelizable operations, can significantly benefit from Processing-in-Memory (PIM) and FPGA hardware platforms. The PIM [19] and FPGA [20] implementations of HD have demonstrated three orders of magnitude speed-up and energy reduction compared to CPU implementation, which indicates HD as an ultra low-power yet high-speed approach for seizure detection.

IV. Conclusion

In this paper, we applied hyperdimensional (HD) computing to detect seizures from 2-channel surface EEG recordings. The proposed technique can be readily applied on raw EEG signals without any feature extraction or domain expertise. Our experiments using a publicly available dataset collected by the Children's Hospital of Boston and MIT revealed HD achieves an average performance up to 82.99%, which outperforms the recent work based on CNN by 6.43%, and random forest by 11.84%. The accuracy can be further improved by increasing the length of hypervectors, which also allows to use larger signal window size to extract more information. The proposed technique uses a small fraction ($\sim$10%) of the recording time for processing on Raspberry-Pi, which indicates it can be easily ported to ultra-light devices.

Acknowledgements

This work was supported in part by CRISP, one of six centers in JUMP, an SRC program sponsored by DARPA, in part by SRC-Global Research Collaboration grant, and also NSF grants #1527034, #1730158, #1826967, and #1911095.

References

[1] F. Tang, A. Hartz, and B. Bauer, "Drug-resistant epilepsy: multiple hypotheses, few answers," *Frontiers in neurology*, vol. 8, p. 301, 2017.

[2] C. E. Stafstrom and L. Carmant, "Seizures and epilepsy: an overview for neuroscientists," *Cold Spring Harbor perspectives in medicine*, vol. 5, no. 6, p. a022426, 2015.

[3] D. Sopic, A. Aminifar, and D. Atienza, "e-glass: A wearable system for real-time detection of epileptic seizures," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2018.

[4] A. Emami, N. Kunii, T. Matsuo, T. Shinozaki, K. Kawai, and H. Takahashi, "Seizure detection by convolutional neural network-based analysis of scalp electroencephalography plot images," *NeuroImage: Clinical*, vol. 22, p. 101684, 2019.

[5] T. R. P. Foundation, "The raspberry pi." `https://www.raspberrypi.org/`, 2020.

[6] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, no. 2, pp. 139–159, 2009.

[7] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "Voicehd: Hyperdimensional computing for efficient speech recognition," in *International Conference on Rebooting Computing (ICRC)*, pp. 1–8, IEEE, 2017.

[8] M. Imani, C. Huang, D. Kong, and T. Rosing, "Hierarchical hyperdimensional computing for energy efficient classification," in *IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2018.

[9] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, pp. 64–69, ACM, 2016.

[10] A. Burrello, K. Schindler, L. Benini, and A. Rahimi, "One-shot learning for ieeg seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing," in *Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4, IEEE, 2018.

[11] A. Burrello, K. A. Schindler, L. Benini, and A. Rahimi, "Hyperdimensional computing with local binary patterns: One-shot learning for seizure onset detection and identification of ictogenic brain regions from short-time ieeg recordings," *IEEE Transactions on Biomedical Engineering*, 2019.

[12] M. Zhou, C. Tian, R. Cao, B. Wang, Y. Niu, T. Hu, H. Guo, and J. Xiang, "Epileptic seizure detection based on eeg signals and cnn," *Frontiers in neuroinformatics*, vol. 12, p. 95, 2018.

[13] A. L. Goldberger, L. A. Amaral, L. Glass, *et al.*, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[14] DIY tDCS, "10/20 system electrode distances." `https://www.diytdcs.com/2012/07/1020-system-electrode-distances/`, 2020.

[15] T. A. Plate, "Holographic reduced representations," *IEEE Transactions on Neural networks*, vol. 6, no. 3, pp. 623–641, 1995.

[16] U. R. Acharya, H. Fujita, *et al.*, "Application of entropies for automated diagnosis of epilepsy using eeg signals: A review," *Knowledge-Based Systems*, vol. 88, pp. 85–96, 2015.

[17] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[18] M. Teplan *et al.*, "Fundamentals of eeg measurement," *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.

[19] S. Gupta, M. Imani, and T. Rosing, "Felix: Fast and energy-efficient logic in memory," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7, IEEE, 2018.

[20] S. Salamat, M. Imani, B. Khaleghi, and T. Rosing, "F5-hd: Fast flexible fpga-based framework for refreshing hyperdimensional computing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 53–62, 2019.