

Hyperdimensional computing encoding for feature selection on the use case of epileptic seizure detection

Una Pale

Embedded Systems Laboratory (ESL)

Swiss Federal Institute of Technology Lausanne (EPFL)
Lausanne, Switzerland

UNA.PALE@EPFL.CH

Tomas Teijeiro

Embedded Systems Laboratory (ESL)

Swiss Federal Institute of Technology Lausanne (EPFL)
Lausanne, Switzerland

TOMAS.TEIEIRO@EPFL.CH

David Atienza

Embedded Systems Laboratory (ESL)

Swiss Federal Institute of Technology Lausanne (EPFL)
Lausanne, Switzerland

DAVID.ATIENZA@EPFL.CH

Abstract

The healthcare landscape is moving from the reactive interventions focused on symptoms treatment to a more proactive prevention, from one-size-fits-all to personalized medicine, and from centralized to distributed paradigms. Wearable IoT devices and novel algorithms for continuous monitoring are essential components of this transition. Hyperdimensional (HD) computing is an emerging ML paradigm inspired by neuroscience research with various aspects interesting for IoT devices and biomedical applications. Here we explore the not yet addressed topic of optimal encoding of spatio-temporal data, such as electroencephalogram (EEG) signals, and all information it entails to the HD vectors. Further, we demonstrate how the HD computing framework can be used to perform feature selection by choosing an adequate encoding. To the best of our knowledge, this is the first approach to perform feature selection using HD computing in the literature. As a result, we believe it can support the ML community to further foster the research in multiple directions related to feature and channels selection, as well as model interpretability.

1. Introduction

Hyperdimensional (HD) computing is an emerging ML paradigm inspired by neuroscience research, pointing towards the power of computation with massively distributed and redundant systems. Instead of representing data as conventional numbers, representing/projecting it to thousands (or millions) of neurons/numbers brings new perspectives. More specifically, HD computing or vector-symbolic architectures represent data in the shape of vectors with high dimensions, *hypervectors*, with usually more than 10000 values. This change in the data representation paradigm brings various advantages from a learning and hardware implementation perspective. From a learning perspective, it opens new paths for semi-supervised [Imani et al. \(2019a\)](#), distributed [Imani et al. \(2019b\)](#), continuous online learning [Moin et al. \(2021\)](#); [Benatti et al. \(2019\)](#), or multi-centroid learning [Pale et al. \(2022b\)](#). From the hardware

implementation perspective, possibilities for parallelization, open paths for designing efficient accelerators Imani et al. (2021) or in-memory computations Karunaratne et al. (2020, 2021). Its lower energy and memory requirements Burrello et al. (2021); Asgarinejad et al. (2020); Imani et al. (2019c) makes it an attractive solution for learning on less powerful devices, such as wearable and IoT systems.

Due to the advantages mentioned above, HD computing is drawing a lot of attention for biomedical applications, and has indeed been tested for electromyogram (EMG), gesture recognition Rahimi et al. (2016), EEG error-related potentials detection Rahimi et al. (2020), emotion recognition from GSR (galvanic-skin response) and electrocardiogram (ECG) and electroencephalogram (EEG) Chang et al. (2019), and epileptic seizure detection from EEG Burrello et al. (2019); Asgarinejad et al. (2020) among others.

EEG (iEEG) and EMG are spatio-temporal, noisy and non-stationary data, whose efficient encoding to HD vectors poses an essential part of the quality of HD learning. However, how to encode channel and feature information has not yet been systematically explored in the literature. In most of the existing literature utilizing EMG or EEG data, only raw data or *local-binary-patterns* (LBPs, Kaya et al. (2014), have been used as features encoded to vectors. Yet, similarly to standard ML approaches, the possibility to add more features can significantly improve the power of the models. In Pale et al. (2021), the authors showed that utilizing a wider variety of statistical, time, or frequency features with HD computing for epileptic seizure, the final detection score outperforms simple raw signal encoding. Thus, in this paper, we discuss possibilities on how to encode information about channels and various possible features one might need to extract from data, to capture all the relevant aspects.

As we shown later in this work, encoding can also enable analyzing performance, correlations, and learning capabilities of individual features. In Burrello et al. (2021) authors explored a similar idea using several individual HD classifiers for three different features. Analyzing various aspects of individual features is a crucial step for a) feature selection, as demonstrated in this paper, and b) for further interpretability of model decisions (which will not be discussed in the scope of this paper). Indeed, feature selection is crucial in designing wearable applications as it helps remove noisy and non-informative features while also directly leading to more lightweight models. To the best of our knowledge, a clear, straightforward methodology for feature selection using HD computing has not been so far presented in the literature.

In this paper, we assess our approaches in the context of epileptic seizure detection. Epileptic seizures are a chronic neurological disorder characterized by the unpredictable occurrence of seizures that affects a significant portion of the world population (0.6 to 0.8%) Mormann et al. (2007). Due to its high inter-patient variability, unpredictable nature and not yet completely understood origins, it still poses an open research question. Despite pharmacological treatments, one-third of patients still suffer from seizures Schmidt and Sillanpää (2012) and are being subject to serious health risks and many restrictions in daily life. There is no yet available wearable device for epilepsy prediction, detection, or continuous monitoring for outpatient environments. HD computing has been utilized for epilepsy detection due to its attractive properties, and only with various improvements to standard HD learning, it can reach the performance of state-of-the-art algorithms Pale et al.

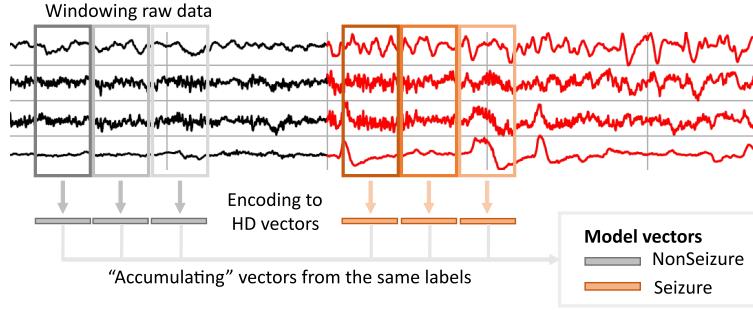


Figure 1: Simple illustration showing the HD computing workflow and learning process from windows of seizure and non-seizure data.

(2022a). Thus, further exploration on how to optimize HD computing and encoding for better performance is of high interest for epilepsy detection.

Speaking more broadly, the healthcare landscape is moving from reactive disease intervention to proactive prevention Waldman and Terzic (2019), as it is not only more cost-effective, but it usually leads to better quality of life Goetzel (2009). At the same time it is shifting from one-size-fits-all to personalized medicine Chan and Ginsburg (2011) and institution-centered to decentralized Puri et al. (2021). For all these aspects, novel algorithms design and optimization for lightweight and wearable IoT devices are essential Tricoli et al. (2017). Thus, even though we focus our work on epilepsy detection, proposed approaches and results are applicable for more general continuous monitoring, early detection of diseases, and preventive healthcare.

Generalizable Insights about ML in the Context of Healthcare

This paper demonstrates how a novel HD computing approach can be used as an alternative to standard state-of-the-art machine learning (ML) approaches in the use case of epileptic seizure detection. We explore not yet addressed topic of optimal encoding of spatio-temporal data, such as EEG, and all information it entails to the HD vectors. Computational efficiency is critical for developing wearable devices for continuous monitoring of diseases such as epilepsy, making long battery lifetimes feasible and moving towards preventive healthcare. Further, we demonstrate an example of how we can utilize the HD computing framework to perform feature selection by choosing an adequate encoding. To the best of our knowledge, this is the first proposal of the feature selection using HD computing in the literature, and we trust it can foster research in the ML community for further development in this direction. From the clinical perspective, tools and models that can improve the interpretability of the models can lead not only to better service to the patients, but also to help as decision support for the doctors and caregivers.

2. Related Work

HD computing has been applied so far in several applications where spatio-temporal data such as EEG, iEEG Asgarinejad et al. (2020); Burrello et al. (2021) or EMG Rahimi et al. (2016); Benatti et al. (2019); Moin et al. (2019); Rahimi et al. (2020) was utilized. The

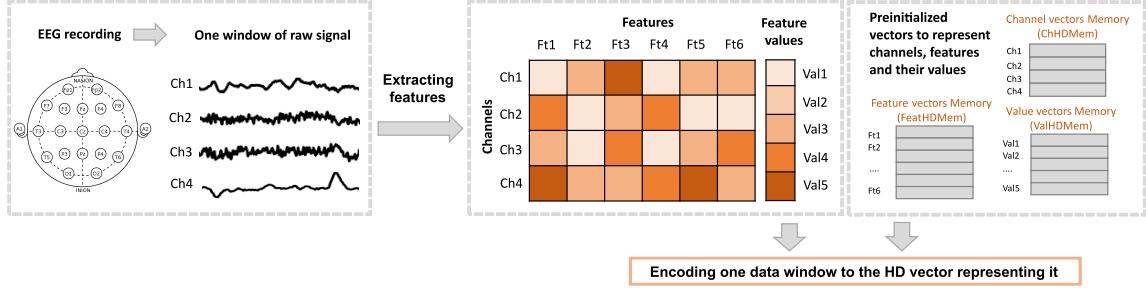


Figure 2: Illustration of encoding one data window to HD vector representing it.

majority of works used raw signal values or short local-binary-patterns (LBPs) to describe the changes of the signal in time and map it directly to HD vectors [Rahimi et al. \(2016\)](#); [Benatti et al. \(2019\)](#); [Moin et al. \(2019\)](#); [Burrello et al. \(2019\)](#); [Asgarinejad et al. \(2020\)](#); [Rahimi et al. \(2020\)](#). This enables combining channel and value (raw signal or LBP value) together and then further combining all samples belonging to the same class, similarly as illustrated on Fig. 1. Some of them have also encoded time information between neighboring samples by utilizing vector permutation. Furthermore, in [Karunaratne et al. \(2021\)](#) authors propose energy-efficient in-memory encoding for this way of encoding spatio-temporal signals. This is an interesting approach in its simplicity as it use directly raw signal values. However, it is limited to only mapping one type of information to HD vectors (i.e., raw signal amplitude or/and signal change trends).

In [Pale et al. \(2021\)](#) authors test, on a use-case of epilepsy seizure detection, how different feature types regularly used in non-HD ML classifiers for epilepsy detection perform in the HD computing framework. They show that, indeed, utilizing more different statistical, time, or frequency features outperforms simple raw signal encoding. Further, in recent [Burrello et al. \(2021\)](#), authors extend their previous work [Burrello et al. \(2019\)](#) by adding mean amplitude and line length features to LPB values. They resolved the problem of encoding more features to HD vectors by having three independent classifiers, each with its own model vectors. Predictions are merged using a simple linear layer that gives a final prediction based on distances from class models of each feature. This is a promising approach that enables the performance comparison for the different features.

In [Ge and Parhi \(2021\)](#), authors explored epileptic seizure detection using power-spectral features from iEEG data, and encountered similar problems as we point out here. They explored three different encoding approaches: 1) concatenating feature HD vectors to generate long HD vectors, 2) using multiple classifiers (one for each feature) and then integrating their predictions by majority voting, and 3) training one classifier using all features. In the last approach, vectors representing features and their values are combined in the first place, and then united with channel information. These approaches are highly interesting due to the broad spectrum of feature characteristics that are integrated. However, unfortunately, in [Ge and Parhi \(2021\)](#) they were not systematically compared from performance, memory or complexity aspects.

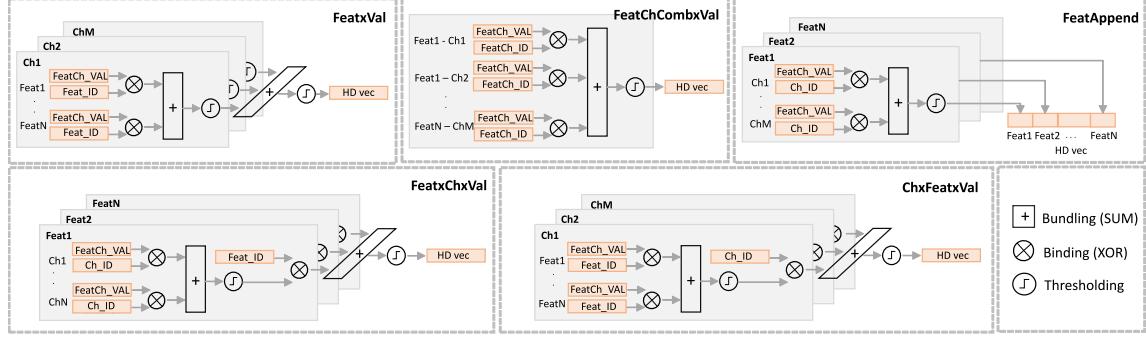


Figure 3: Schematic of different possibilities to encode information about channels, features and their values to HD vectors. Five different encoding approaches tested in the paper are illustrated.

3. Methods

3.1. Encoding spatio-temporal data for HD computing

Typical spatio-temporal data, such as EMG or EEG, is 2D data. It consists of several channels positioned in different physical locations (which can have specific relations between them) and recorded during time frames of various lengths. After preprocessing the signals, features are typically extracted in time, using time windows shifted by some time step. As this is repeated for each of the channels, this leads to 3D data containing information on the feature, channel, and time, as illustrated in Fig. 2. Within the HD computing workflow, this means that we need to define initial vectors that represent each of these entities: HD vectors representing each feature ($FeatID$ as in Fig. 2), and vectors representing each possible value of features ($FeatChVAL$), and channels ($ChID$). As feature values of various features might not have the same range and are not usually integer values, they need to be normalized and discretized to a specific number of bins.

The question that we address in this work is how to encode all this information into HD vectors. Moreover, we assess the different possible methods from several perspectives, which include classification performance, memory and computational complexity. The last two metrics are very relevant in the design of the next generation of smart wearable systems. Therefore, in Fig. 3 we illustrate different possibilities considered in this paper to encode the time window of data to an HD vector. The time window contains information on all features and their values calculated from each of the channels.

Although HD vectors can be binary (containing only 0 or 1), bipolar (containing -1 and 1), ternary (containing -1, 0 or 1), integer or even floating point, the most memory friendly and commonly used are binary HD vectors. Three basic arithmetic operations that are performed on the vectors are: 1) bundling or bitwise summation, 2) binding or bitwise XOR, and 3) thresholding to binarize vectors after summation. Bundling operation leads to a vector that is with high probability very similar to summed vectors, while on the other hand, binding leads to the vector that is orthogonal to the bound vectors. Thresholding is necessary to keep the final vector again in binary (bipolar or tertiary form) after summation.

A typical approach in the literature dealing with simpler, non-spatial data is to bind feature vectors with feature value vectors and then bundle (and threshold) them. Translating this approach to EEG data leads to two options: 1) $Feat \times Val$ and 2) $ChFeatComb \times Val$. The difference between them is, as illustrated in Fig. 3 and formulated by (1 and 2), that $Feat \times Val$ does not include information about channels but simply bundles (and normalizes) features ($Feat_{ID}$) and feature values from each channel ($FeatCh_{VAL}$). On the other side, $ChFeatComb \times Val$ approach treats each feature and channel combination as an individual feature and gives it an independent HD vector ($FeatCh_{ID}$). This approach distinguishes between channels as formulated in (2), but can lead to significantly large memory maps to store all initial HD vectors. This could happen in case data contains many channels (such as iEEG data) or when many features are extracted from each channel.

As this can be problematic from a memory perspective viewpoint in the context of wearable medical devices, we propose other more EEG-inspired approaches: 3) $Feat \times Ch \times Val$ and 4) $Ch \times Feat \times Val$. Both approaches first initialize vectors for each channel (Ch_{ID}), each feature ($Feat_{ID}$), and possible feature values ($FeatCh_{VAL}$). The difference is, in the order of bundling information. For $Feat \times Ch \times Val$, as formulated by (3), in the first step, channels (Ch_{ID}) and feature values on that channels ($FeatCh_{VAL}$) are bundled. Then, in the second step, the bundling with feature ID's ($Feat_{ID}$) follows. In $Ch \times Feat \times Val$ approach order of bundling is opposite, as formulated in (4).

$$Feat \times Val = \left[\sum_{i=1}^{numCh*numFeat} Feat_{IDi} \oplus FeatCh_{VALi} \right] \quad (1)$$

$$ChFeatComb \times Val = \left[\sum_{i=1}^{numCh*numFeat} FeatCh_{IDi} \oplus FeatCh_{VALi} \right] \quad (2)$$

$$Feat \times Ch \times Val = \left[\sum_{i=1}^{numFeat} Feat_{IDi} \oplus \left[\sum_{j=1}^{numCh} Ch_{IDj} \oplus FeatCh_{VALij} \right] \right] \quad (3)$$

$$Ch \times Feat \times Val = \left[\sum_{i=1}^{numCh} Ch_{IDi} \oplus \left[\sum_{j=1}^{numFeat} Feat_{IDj} \oplus FeatCh_{VALij} \right] \right] \quad (4)$$

The last approach, called *FeatAppend*, is designed with the goal to make it easier to interpret encoded vectors and determine which part of them comes from different features. In this approach, as illustrated in Fig. 3, channels and feature values are bound, bundled, and thresholded to get a vector representing the encoded sub-vector for each feature. Instead of binding it with other feature sub-vectors as in $Feat \times Ch \times Val$ these vectors are appended one after another, as formulated in (5). In order to get the same final vector dimension, initialized sub-vectors have, in this case, smaller dimensions. This encoding organization enables to analyze vectors for every single feature. For example, in this paper, we analyze what the separability of classes for each of the features is, as well as the predictions of individual features, and how confident they are with respect to other features. These measures are defined in next section and allow us to perform additional features selection steps.

$$FeatAppend = \left[\sum_{j=1}^{numCh} Ch_{IDj} \oplus Feat_1 Ch_{VALj} \right] \dots \left[\sum_{j=1}^{numCh} Ch_{IDj} \oplus Feat_{numFeat} Ch_{VALj} \right] \quad (5)$$

3.2. Feature selection

In the *FeatAppend* approach, it is known which part of the final encoded vector comes from each feature, which enables analysis per individual feature. In this paper, we define and measure several metrics for each feature:

- *Prediction*: Using only $d = D/numFeat$ bits corresponding to the feature of interest, we can determine the prediction for each sample using only that feature. A decision is made in the same way as when the whole vector is used; the label of the most similar class vector is given. As we use binary HD vectors, hamming distance is used as our similarity metric measure.
- *Feature certainty*: For each time moment, we can also quantify the certainty of each feature's label based on distances from both class vectors. The certainty is calculated as the difference between distances from two classes, divided by the average absolute distance for all features in the same time moment, as in formula (6).

$$C_f = \frac{|dist_{Sf} - dist_{Nsf}|}{\frac{1}{numFeat} \sum_{i=1}^{numFeat} (|dist_{Si} - dist_{NSi}|)} \quad (6)$$

- *Correlation*: Based on the predictions of each feature in time, it is possible to measure the correlation between predictions of different features and take it into account later for feature selection.
- *Class separability*: For binary HD vectors, separability (S_f) is measured as the relative hamming distance between class vectors (HD_S, HD_{Ns}) when using only the bits of the corresponding feature. This measurement is obtained using equation (7). Based on the separability measure it is also possible to estimate the usefulness of each feature.

$$S_f = \text{hamming} \left(HD_S \left[\frac{f * D}{numFeat} : \frac{(f + 1) * D}{numFeat} \right], HD_{Ns} \left[\frac{f * D}{numFeat} : \frac{(f + 1) * D}{numFeat} \right] \right) \quad (7)$$

Metrics measured per feature on a training set can then be used to perform feature selection, as shown in Fig. 4. In this paper, we demonstrate three ways to do it. Each approach starts by ordering features based on specific quality measures:

- *Feature performance*: Based on the predictions for each sample and the true labels, we can measure the performance of each feature. The exact performance metric of choice can depend on the application, and for epilepsy detection we define them later in Sec. 4.4.2.

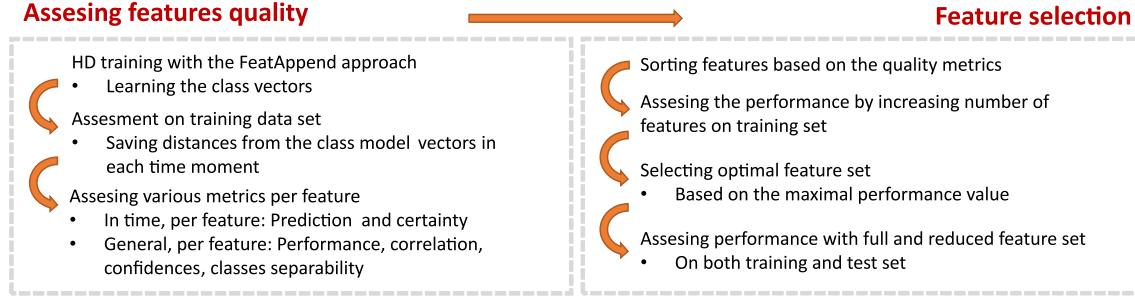


Figure 4: Feature selection workflow with detailed steps.

- *Feature confidence:* Based on the certainty values and predictions per feature, as well as true labels per time moment, from the training set, features can be ordered based on the highest confidence. Confidence per feature is calculated as how much more certain is that feature during correct predictions ($C_f|TP$) versus wrong predictions ($C_f|FP$) and is calculated by the formula (8).

$$Conf_f = \frac{\overline{C_f|TP} - \overline{C_f|FP}}{\overline{C_f|FP}} \quad (8)$$

- *Feature performance and correlation:* If selection is made based only on the feature performance, it might lead to selecting features that are performing well but are highly correlated and thus maybe redundant. In this approach, we select features one by one by evaluating in each step how the performance changes when adding one of the not yet used features. This is a slower process (and computationally more complex) than sorting features based on different direct quality metrics, as in the previous two approaches. Nonetheless, it leads to a more elaborate order of features, which also includes the novelty each feature brings.

After features are ordered, the performance on the train and test set is assessed when increasing the number of features until all features are included. Prediction when using n features is given by summing up distances from seizure and non-seizure model vectors of individual features as in the formula (9). If the final distance is positive, the prediction (*Vote*) is seizure, otherwise non-seizure. From the performance curve of the training set, the optimal number of features is chosen as the number of features giving maximal performance (or the smallest number of features without loss in the performance when compared to using all features). In the end, as shown in Fig. 4, performance on the test set is measured for the chosen, reduced set of features. Then, we compare it with the case without feature selection.

$$Vote = sign \left[\sum_{i=1}^{numFeat} (dist_{NSi} - dist_{Si}) \right] \quad (9)$$

4. Experimental setup

4.1. Dataset

For the analysis, we use the CHB-MIT database. It consists of 24 subjects with medically-resistant seizures ranging in age from 1.5 to 22 years [Shoeb \(2009\)](#); [Goldberger Ary L. et al. \(2000\)](#). It is an EEG database collected by the Children’s Hospital of Boston and MIT, and contains 183 seizures overall, with an average of 7.6 ± 5.8 seizures per subject. We use the 18 channels (i.e., FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, T8-P8, P8-O2, FZ-CZ, CZ-PZ) that are common to all patients.

The original dataset contains more than 980 hours of recordings that are divided in approximately one hour-long files. Even if the common approach in literature is using balanced data preparation, it can lead to highly overestimated performance [Pale et al. \(2022b\)](#). Further, training on the full dataset using HD computing is not feasible due to its complexity. Thus, as proposed in [Pale et al. \(2022a\)](#), we use a data selection approach that contains all seizure segments and ten times more non-seizure data. Data is arranged in such a way that for each seizure file, seizure data is extracted and surrounded by non-seizure data, which was randomly selected from one of the files not containing any seizure. In this way, each file contains an equal ratio of seizure and non-seizure data, and can be easily used for leave-one-file-out cross-validation.

4.2. Feature Choices

In previous papers applying HD computing for epileptic seizure detection [Pale et al. \(2022b,a, 2021\)](#), 45 features were used for classification, including mean-amplitude, 37 entropy features and 8 relative frequency domain features. Based on a literature review by [Siddiqui et al. \(2020\)](#) discussing the importance of various features, frequency features, as well as line-length were reportedly the most useful. Thus, in this paper, we keep the mean-amplitude (*mean_ampl*) and use both relative and absolute values of power spectral density in the five common brain wave frequency bands: delta: [0.5-4] Hz (*p_delta* and *p_delta_rel*), theta: [4-8] Hz (*p_theta* and *p_theta_rel*), alpha: [8-12] Hz (*p_alpha* and *p_alpha_rel*), beta: [12-30] Hz (*p_beta* and *p_beta_rel*), gamma: [30-45] Hz (*p_gamma* and *p_gamma_rel*), and low-frequency components: [0-0.5] Hz (*p_dc* and *p_dc_rel*) and [0.1-0.5] Hz (*p_mov* and *p_mov_rel*). To calculate the relative values, we divide the absolute values by the total power (*p_tot*). In this case, we do not use entropy features, as a preliminary analysis and the literature showed a very small discriminative power. In the end, we included the line-length feature (*line_length*), as introduced in [Esteller et al. \(2001\)](#), and which showed a high discriminative power in the preliminary analysis. Thus, in total, we extract 19 features. Before extracting the features, the data is filtered with a zero-phase, 4th order Butterworth band-pass filter between [1, 20] Hz. Features are extracted from data segmented into 4-second windows with a 0.5-second step.

4.3. HD Learning Workflow

The standard HD computing workflow consists of a single-pass training, where HD vectors representing different data windows (in this case, four-second windows) from the same

label class are bundled all together to form a model HD vector representing that class (illustrated in Fig. 1). This approach is simple and fast. However, all data windows are equally important, which can, in highly imbalanced datasets such as epilepsy ones, lead to the domination of more common patterns in the final vectors. As shown in Pale et al. (2022a), this situation leads to an under-representation of less common patterns and ultimately lowers the performance. OnlineHD was proposed in Hernández-Cano et al. (2021) as an alternative, where each window vector is multiplied with a weight before being added to the model vector. The weight is defined by the similarity of the current vector to the current prototype vectors; the higher the similarity, the lower the weight, which helps in identifying more repeating patterns and lowers model saturation with them. In Pale et al. (2022a) standard HD and OnlineHD have been compared for the exact use case of epileptic seizure detection, and OnlineHD has shown to have higher performance. An essential step of OnlineHD is that, when a new sample vector is added, it is also subtracted from the opposite class if it is more similar. These steps help to make model vectors more separable and improve the performance.

4.4. Validation

4.4.1. FEATURE COMPARISON

A standard approach to compare features individually is the Kullback-Leibler or Jensen-Shannon divergences, which analyses the distance between the distributions of feature values for the different classes. As explained in Sec. 3.2 *FeatAppend* appending approach enables various other ways to compare features. For example, it is possible to compare predictions if the model only has individual features and their certainties in time. Moreover, the final performance per feature, confidence, correlation, and class separability can be assessed too.

4.4.2. PERFORMANCE EVALUATION

Training and evaluation are done independently for each individual due to the subject-specific nature of epileptic seizures. For each person, we perform leave-one-seizure out cross-validation and report the performance as the average across all cross-validation runs. In the end, we report performance as average over all subjects.

The performance of the classifier is evaluated with respect to seizure episode detection and duration-based (window-based) detection. There is an active discussion in the community on which performance measures to use to increase the interpretability of performance Ziyabari et al. (2019); Shah et al. (2020). Here we use the same metrics as in Pale et al. (2022b). More specifically, we measure sensitivity, predictivity, and F1 score on the level of episode detection, as well as on the level of episode duration. The performance at the episode level groups the signal into blocks of seizure and non-seizure. The performance at the duration level, on the other hand, cares about the correct prediction of each window, meaning that seizures need to be predicted correctly during their whole duration. Finally, to have one single final metric, we combine F1 scores for episodes ($F1E$) and duration ($F1D$) using the geometric mean as $F1DEgmean$.

In epilepsy detection, raw label predictions often lead to an unrealistic behavior of seizure dynamics (e.g., seizures lasting only a few seconds, or seizures that are only a few seconds apart). Thus, label post-processing is an integral part of the whole pipeline. Here we,

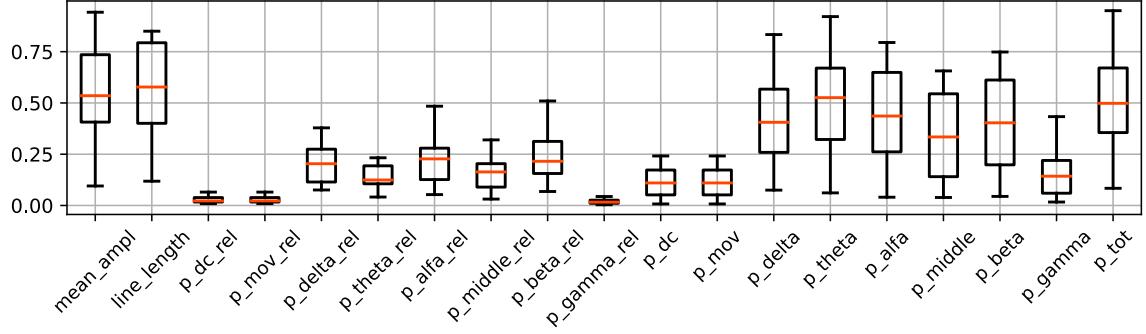
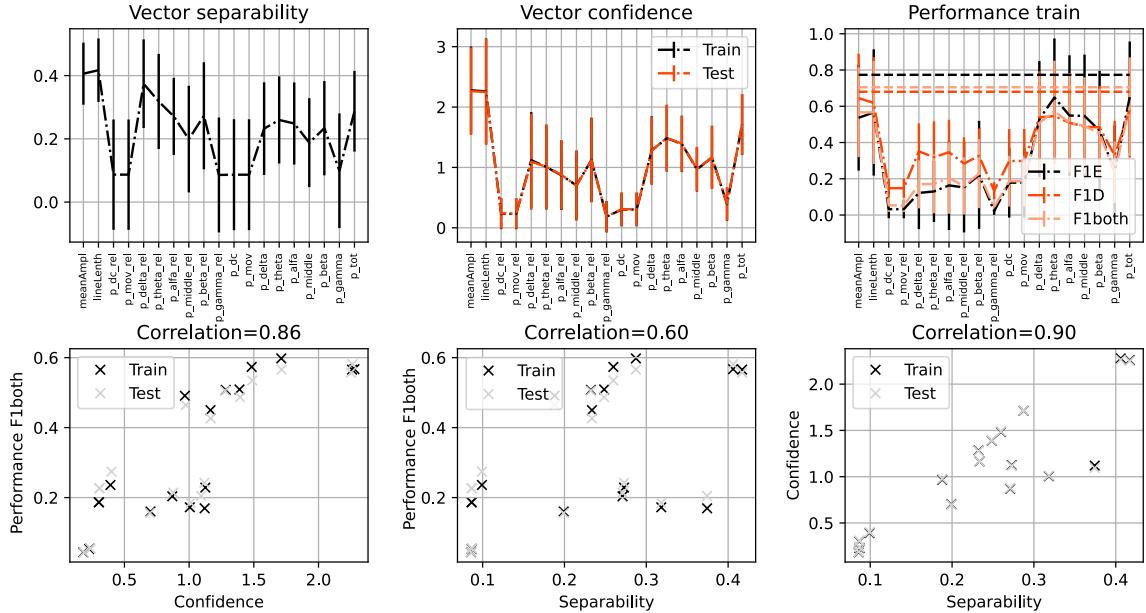


Figure 5: Jensen-Shannon divergence of features.

Figure 6: Comparison of features based on *FeatAppend* approach. The averaged values over all subjects are shown.

post-process raw label predictions by performing a moving average with majority voting, using a window size of 5s.

5. Results

5.1. Feature comparison

Fig. 5 shows Jensen-Shannon divergence of 19 features we used and their distribution over all channels for all subjects. There is a clear difference between features, where mean amplitude, line length, total energy, and absolute spectral powers of the delta, theta, alpha, beta, and middle-range are quite discriminative. Relative powers seem to be less discriminative than absolute values.

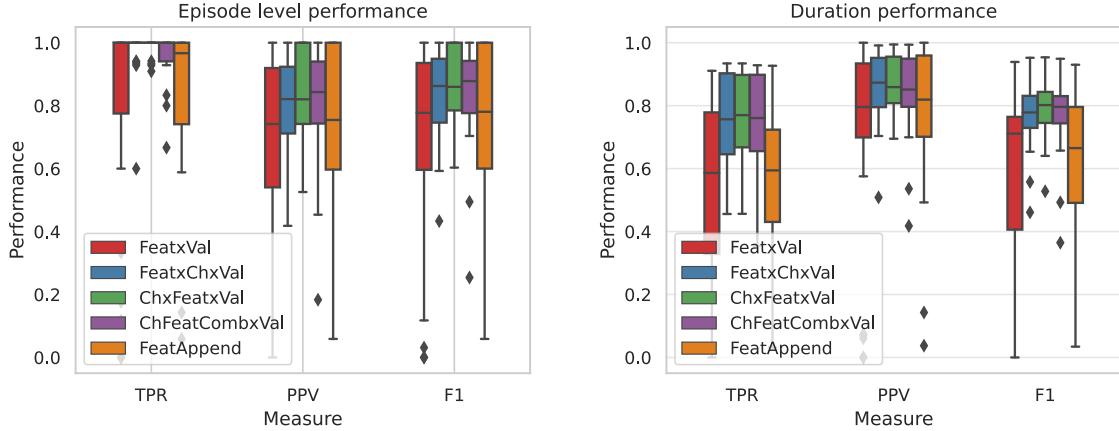


Figure 7: Performance of different EEG encoding approaches.

Fig. 6 shows comparison of the features based on measures extracted using *FeatAppend* approach. More specifically, the separability of vectors for the two classes are shown for each feature, average confidence of each feature, and the performance (F1 score for episodes, duration, and their gmean) when using only individual features. In the subplot of performance, the horizontal line shows the performance achieved when using all the features. It is visible that no single feature reaches the performance of all features, but some of them get quite close to it (i.e., mean amplitude, line length, total power, and power of delta and theta). In the bottom row, the correlation between confidence and performance, separability and performance, and separability and confidence are plotted, showing high correlation values. This confirms that the *FeatAppend* approach can indeed be used to investigate different properties of individual features.

5.2. Encoding comparison

Next, as *FeatAppend* is one of the possible approaches to encode all EEG data aspects into the vectors to use them for HD learning, we compare it with other possible encoding approaches. Fig. 7 shows the average performance for all subjects without any post-processing. Performances on the level of seizure episodes and seizure duration are shown. TPR or true positive ratio is the measure of sensitivity, PPV or positive predictive value is the measure of precision, and F1 score is gmean between TPR and PPV.

Feat × Val encoding, which does not takes channel information into account, leads to lower performance encoding types that take channel information into account. There is no significant difference in performance between the three approaches that include channel information: *Feat × Ch × Val*, *Ch × Feat × Val* and *ChFeatComb × Val*. The *FeatAppend* approach, even if including channel information, yields a smaller performance than the three approaches that utilize channel information, probably due to the smaller number of dimensions per feature. Yet, performance is not worse than the *Feat × Val* approach.

Fig. 8 shows the memory needed to store all HD vectors (for channels, features, and values) for each of the approaches is shown. Relative ratios are shown as memory directly depends on the chosen dimension D (in this case 19000) and whether HD vectors are

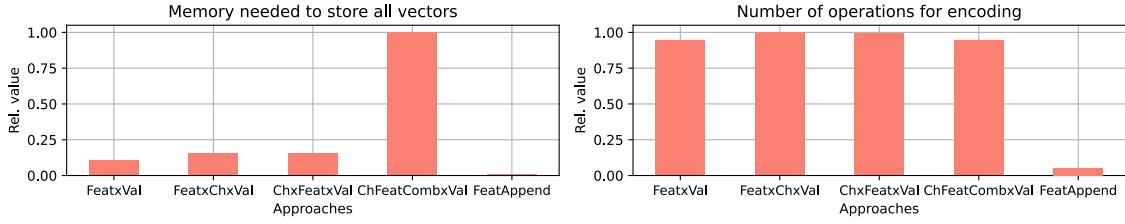


Figure 8: Comparison of encoding approaches in terms of memory needed and number of binding and bundling operations needed to encode one data window.

binary or not. Due to the large number of combinations of features and channels, the $ChFeatComb \times Val$ approach is the most memory-demanding one, and scales the worst. $FeatAppend$ approach is the approach that requires the least amount of memory as base vectors have lower $D/numFeat$ dimensions.

Further, the relative number of bundling and binding operations needed to encode one window of data to the HD vector representing it is shown as well in Fig. 8. The $FeatAppend$ approach requires significantly less operations due to the effectively smaller number of dimensions per vector for each feature. From other approaches, $ChFeatComb \times Val$ and $Feat \times Val$ require slightly less operations than $Feat \times Ch \times Val$ and $Ch \times Feat \times Val$.

5.3. Feature selection

Fig. 9 shows the performance when adding features one-by-one, for each of the three methods described in Sec. 9. The difference between approaches relies in how features are ordered. In the first row of Fig. 9 features are ordered based on their performance metric (in this case, their F1DE performance) and in the second row based on average feature confidences given by equation (8). Finally, in the third row, features are added one by one choosing the best feature to increase performance when added to previously chosen features. This approach can be referred to as more optimal order and choice of features, as it takes into account correlation between features. In this case, performance increases and reaches performance higher than when using the first two approaches.

The last column shows the boxplots of feature orders for each feature selection method, for all subjects. The smaller the ranking number, the sooner that feature was chosen in the incremental feature selection. What can be noticed is that the ranking of features in the first two approaches when using solely feature performance or feature confidence is similar to the results given by the discriminative power analysis shown in Fig. 5. In the last case of more optimal feature selection, feature order is slightly different due to feature correlations that were taken into account.

Fig. 10 shows the results of optimal feature selection for every subject. The first graph shows the chosen number of features per subject and average for all subjects, which is shown with the horizontal line. The following two graphs show the performance improvement (or decrease), i.e., F1 for episodes and gmean of F1 for episodes and duration, for the training and testing set. In the title of the figures, we provide the average performance for all subjects on the test set. A significant variability exists between the number of features chosen per subject, ranging from 1 to 10 features, with an average of 5.8 or 30% of features.

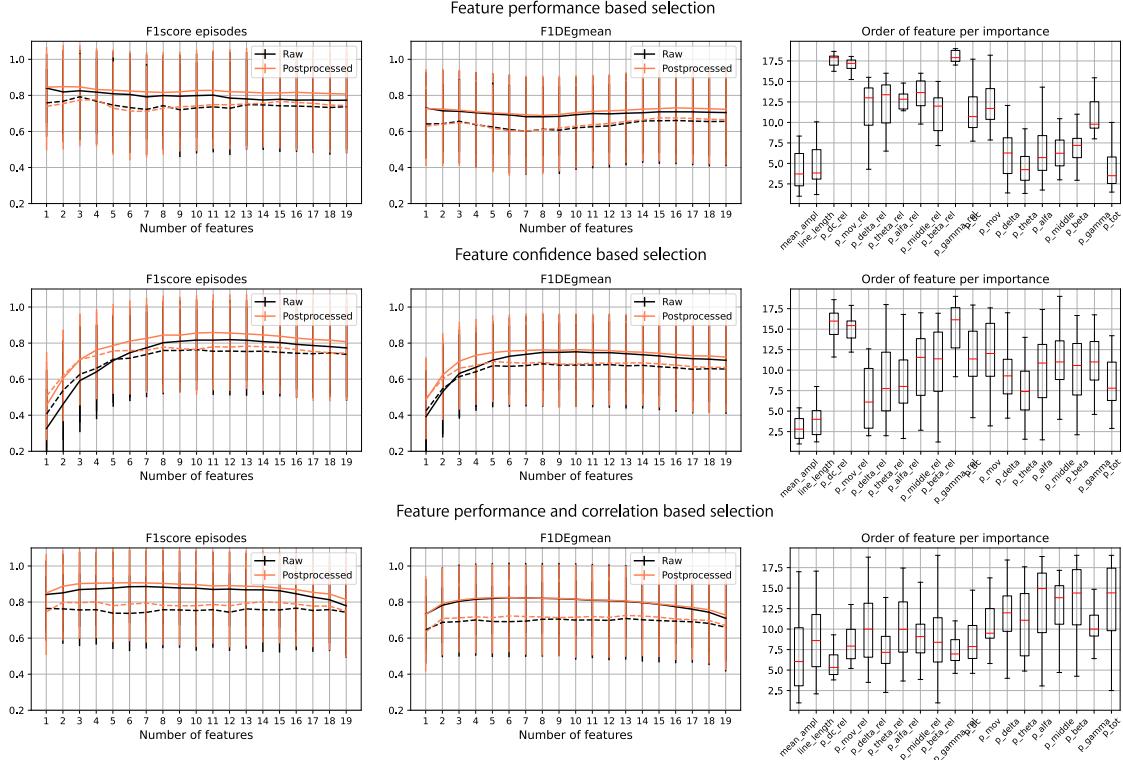


Figure 9: Performance evolution by incrementally adding one-by-one new feature for three approaches: based on feature performance, feature confidence and both feature performance and correlation.

In Table 1 shows more detailed results. More precisely, results for all three feature selection methods are given. For each of the methods, we show results when we optimized the F1 score only for episodes (F1E), or when the F1 score for seizure duration is also taken into account (F1DE). Further, in the table, average F1 and F1DE performances for both train and test are shown before and after label post-processing. What can be noticed is that when optimizing only F1E, fewer features are needed, but it usually leads to a smaller performance increase for F1DE. When F1DE is optimized, this leads to a significant performance increase both for F1DE and F1E but at the price of a slightly higher number of features chosen. In general, the performance increase is smaller on the test set than on the train set, which is reasonable as the optimal number of features was chosen based on the train set without knowledge about the test set. Yet, the performance increase is not negligible, ranging up to 7% for the test set.

When comparing the three different feature selection approaches, there are slight differences in the optimal number of features chosen and the performance gained. Feature selection based only on the performance or confidence leads to a slightly higher number of features but not a higher performance increase. This is due to information about feature correlations that are missing.

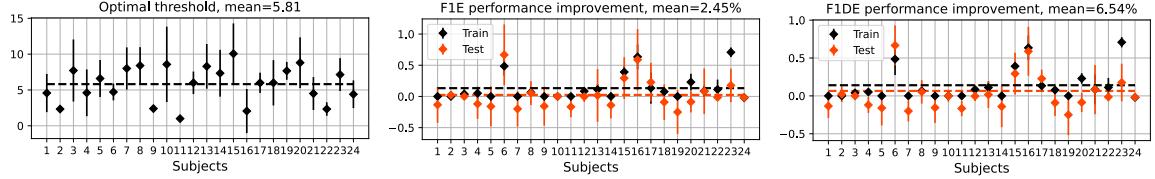


Figure 10: Optimal number of features and performance after feature selection. Example is showed for feature selection using both feature performances and correlations. The horizontal lines represent the average for all subjects.

Table 1: Optimal number of features and performance change for different feature selection approaches.

Feat. selection approach	Perf. used	Nr. Feat.	Train Raw		Train Post		Test Raw		Test Post	
			F1E	F1DE	F1E	F1DE	F1E	F1DE	F1E	F1DE
Feature performance	F1E	2.84	12.29	4.63	6.75	1.43	4.78	-1.40	-0.27	-4.53
	F1DE	6.99	9.41	7.49	6.60	5.39	4.94	3.69	3.64	1.90
Feature confidence	F1E	5.98	10.92	8.55	6.59	5.56	4.00	3.75	2.90	2.28
	F1DE	6.62	9.44	10.26	6.60	7.81	3.56	6.03	3.43	4.82
Feat perf and corr	F1E	2.65	14.61	8.91	9.35	5.45	5.84	1.88	1.79	-1.06
	F1DE	5.81	13.29	14.07	9.44	10.84	2.45	6.54	5.27	6.96

Finally, the code and data required to reproduce the presented results are available online as open-source (link will be available after the double-blind peer-review process).

6. Discussion

This paper draws attention to not yet discussed and properly explored topics of mapping and encoding spatio-temporal data such as EEG or EMG to HD vectors. Although HD computing has been utilized and has shown promising results for various biomedical applications (in particular utilizing EEG and EMG), most works in the literature only use raw data (or LBP values), namely, only one feature per channel. Thus, the optimal encoding when more features per channel or more available data modalities are used remained unclear.

In this work, we propose five alternatives to encode feature values for all channels of one data window into an HD vector and test it on epileptic seizure detection. Our results show that including channel information is beneficial for epileptic detection performance, but that the order in which features and channels are mapped to corresponding values is not relevant. Further, we show that the *ChFeatComb* \times *Val* approach is the most memory demanding and that *Feat* \times *Ch* \times *Val*, *ChxFeat* \times *Val* and *FeatAppend* are comparable and more appropriate. *FeatAppend* requires the least amount of memory and operations to encode vectors due to the effectively smaller number of dimensions per feature vector. By creating individual vectors for each feature and appending them next to each other, in *FeatAppend* performance is slightly reduced. This situation is probably due to fewer HD dimensions used per feature. Thus, it could be improved by increasing the total dimension of vectors, but this also leads to higher memory requirements, which could make it less friendly for wearable devices.

By utilizing the *FeatAppend* approach, we present, to the best of our knowledge, for the first time in the literature, a way to perform feature selection using HD computing. An incremental feature approach was tested with three different methods to determine the order of features to be added. All approaches led to a significant reduction of features while keeping or even significantly improving the performance compared to using all features. In the future, approaches using feature elimination could be similarly tested.

FeatAppend is interesting not only from a feature selection perspective but also from the clinical perspective of a deeper understanding of various features and their properties. For example, we investigated several measures per feature: performance, probabilities of decisions, confidences, correlation, and separability of classes, which can all lead to knowledge discovery related to the usefulness of features. This approach can be adapted to *ChannelAppend* and be used in an analogous way for channel comparison, channel selection, and potentially seizure localization.

Thus we hope that in the future, this work will serve as inspiration for further research and novel ideas in the direction of feature exploration, feature and prediction interpretability, and channel selection. More specifically, for epileptic seizure detection, seizure localization and more detailed feature quality assessment for different seizure-type classifications could be research venues.

Limitations As the majority of works dealing with large and unbalanced databases, such as epilepsy ones, this work was done only on a sub-selection of the CHB-MIT database due to computational limitations. Namely, training on the whole database takes an extensive amount of time. Thus, by using a random selection of non-seizure data and 10 times more non-seizure than seizure data, we tried to minimize the negative effects of not using the whole database. Still, in the future, it would be interesting to optimize and adapt the code for GPU processors and reproduce the analysis with the whole database.

The encoding approach based on feature appending is simple and enables a detailed comparison of each feature, but it has the drawback that the dimension of the HD vectors scales with the number of features, potentially making it more memory and time-consuming. At the same time, reducing the number of dimensions per feature lowers the information capacity that can be stored and ultimately lowers the performance. Thus, different approaches that enable the comparison of a high number of features without limiting the number of dimensions per feature should be designed in the future.

7. Conclusion

In this paper, we have demonstrated how a novel approach of hyperdimensional computing (HD) can be used as an alternative to standard state-of-the-art ML approaches in the use case of epileptic seizure detection. We explored the not yet addressed topic of optimal encoding of spatio-temporal data, such as EEG, and all the information it entails, into the HD vectors. We compared different approaches with respect to their memory and computational complexity, as these metrics are of high interest for wearable devices for continuous monitoring of diseases, such as epilepsy or stress conditions. Hence, those approaches with lower complexity could make longer battery lifetimes feasible and make a step towards a preventive healthcare.

In addition, we have demonstrated how HD computing framework can be utilized to perform feature selection by choosing an adequate encoding. Three approaches were tested and led to a significant reduction of features, while keeping or even significantly improving the performance compared to using all the features. Overall, we expect that this work can serve as inspiration for further research and novel ideas in the direction of feature exploration, feature and prediction interpretability, and channel selection.

References

- Fatemeh Asgarinejad, Anthony Thomas, and Tajana Rosing. Detection of Epileptic Seizures from Surface EEG Using Hyperdimensional Computing. In *IEEE EMBC*, 2020. doi: 10.1109/EMBC44109.2020.9175328.
- Simone Benatti, Fabio Montagna, Victor Kartsch, Abbas Rahimi, Davide Rossi, and Luca Benini. Online Learning and Classification of EMG-Based Gestures on a Parallel Ultra-Low Power Platform Using Hyperdimensional Computing. *IEEE Transactions on Biomedical Circuits and Systems*, 13(3), 2019. doi: 10.1109/TBCAS.2019.2914476.
- Alessio Burrello, L. Cavigelli, K. Schindler, L. Benini, and Abbas Rahimi. Laelaps: An Energy-Efficient Seizure Detection Algorithm from Long-term Human iEEG Recordings without False Alarms. *DATE*, 2019. doi: 10.23919/DAT.2019.8715186.
- Alessio Burrello, Simone Benatti, Kaspar Schindler, Luca Benini, and Abbas Rahimi. An Ensemble of Hyperdimensional Classifiers: Hardware-Friendly Short-Latency Seizure Detection With Automatic iEEG Electrode Selection. *IEEE Journal of Biomedical and Health Informatics*, 25(4):935–946, 2021. doi: 10.1109/JBHI.2020.3022211.
- Isaac S. Chan and Geoffrey S. Ginsburg. Personalized medicine: progress and promise. *Annual Review of Genomics and Human Genetics*, 12:217–244, 2011. doi: 10.1146/annurev-genom-082410-101446.
- En-Jui Chang, Abbas Rahimi, Luca Benini, and An-Yeu Andy Wu. Hyperdimensional Computing-based Multimodality Emotion Recognition with Physiological Signals. In *2019 IEEE AICAS*, 2019. doi: 10.1109/AICAS.2019.8771622.
- R. Esteller, J. Echauz, T. Tcheng, B. Litt, and B. Pless. Line length: an efficient feature for seizure onset detection. In *IEEE EMBC*, 2001. doi: 10.1109/IEMBS.2001.1020545.
- Lulu Ge and Keshab K. Parhi. Seizure Detection Using Power Spectral Density via Hyperdimensional Computing. In *IEEE ICASSP*, 2021. doi: 10.1109/ICASSP39728.2021.9414083.
- Ron Z. Goetzel. Do prevention or treatment services save money? The wrong debate. *Health Affairs (Project Hope)*, 28(1):37–41, 2009. doi: 10.1377/hlthaff.28.1.37.
- Goldberger Ary L., Amaral Luis A. N., Glass Leon, Hausdorff Jeffrey M., Ivanov Plamen Ch., Mark Roger G., Mietus Joseph E., Moody George B., Peng Chung-Kang, and Stanley H. Eugene. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215.

Alejandro Hernández-Cano, Yeseong Kim, and Mohsen Imani. A Framework for Efficient and Binary Clustering in High-Dimensional Space. In *DATE*, 2021. doi: 10.23919/DATE51398.2021.9474008.

Mohsen Imani, Samuel Bosch, Mojān Javaheripi, Bita Rouhani, Xinyu Wu, Farinaz Koushanfar, and Tajana Rosing. SemiHD: Semi-Supervised Learning Using Hyperdimensional Computing. In *IEEE ICCAD*, 2019a. doi: 10.1109/ICCAD45719.2019.8942165.

Mohsen Imani, Yeseong Kim, Sadegh Riazi, John Messerly, Patric Liu, Farinaz Koushanfar, and Tajana Rosing. A Framework for Collaborative Learning in Secure High-Dimensional Space. In *IEEE CLOUD*, 2019b. doi: 10.1109/CLOUD.2019.00076.

Mohsen Imani, Sahand Salamat, Behnam Khaleghi, Mohammad Samragh, Farinaz Koushanfar, and Tajana Rosing. SparseHD: Algorithm-Hardware Co-optimization for Efficient High-Dimensional Computing. In *IEE FCCM*, 2019c. doi: 10.1109/FCCM.2019.00034.

Mohsen Imani, Zhiwen Zou, Samuel Bosch, Sanjay Anantha Rao, Sahand Salamat, Venkatesh Kumar, Yeseong Kim, and Tajana Rosing. Revisiting HyperDimensional Learning for FPGA and Low-Power Architectures. In *IEEE HPCA*, 2021. doi: 10.1109/HPCA51647.2021.00028.

Geethan Karunaratne, Manuel Le Gallo, Giovanni Cherubini, Luca Benini, Abbas Rahimi, and Abu Sebastian. In-memory hyperdimensional computing. *Nature Electronics*, 3(6): 327–337, 2020. ISSN 2520-1131. doi: 10.1038/s41928-020-0410-3.

Geethan Karunaratne, Manuel Le Gallo, Michael Hersche, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Energy Efficient In-Memory Hyperdimensional Encoding for Spatio-Temporal Signal Processing. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(5), 2021. doi: 10.1109/TCSII.2021.3068126.

Yılmaz Kaya, Murat Uyar, Ramazan Tekin, and Selçuk Yıldırım. 1D-local binary pattern based feature extraction for classification of epileptic EEG signals. *Applied Mathematics and Computation*, 243:209–219, 2014. doi: 10.1016/j.amc.2014.05.128.

Ali Moin, Andy Zhou, S. Benatti, A. Rahimi, G. Alexandrov, Alisha Menon, Senam Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, A. C. Arias, L. Benini, and J. Rabaey. Adaptive EMG-based hand gesture recognition using hyperdimensional computing. *ArXiv*, 2019.

Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, Fred Burghardt, Luca Benini, Ana C. Arias, and Jan M. Rabaey. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1), 2021. doi: 10.1038/s41928-020-00510-8.

Florian Mormann, Ralph G. Andrzejak, Christian E. Elger, and Klaus Lehnertz. Seizure prediction: the long and winding road. *Brain: A Journal of Neurology*, 130:314–333, 2007. doi: 10.1093/brain/awl241.

- Una Pale, T. Teijeiro, and David Atienza Alonso. Systematic Assessment of Hyperdimensional Computing for Epileptic Seizure Detection. *IEEE EMBC*, 2021. doi: 10.1109/EMBC46164.2021.9629648.
- Una Pale, Tomas Teijeiro, and David Atienza. Exploration of Hyperdimensional Computing Strategies for Enhanced Learning on Epileptic Seizure Detection. *arXiv:2201.09759*, 2022a.
- Una Pale, Tomas Teijeiro, and David Atienza. Multi-Centroid Hyperdimensional Computing Approach for Epileptic Seizure Detection. *Frontiers in Neurology*, 13, 2022b.
- Vikram Puri, Aman Kataria, and Vishal Sharma. Artificial intelligence-powered decentralized framework for Internet of Things in Healthcare 4.0. *Transactions on Emerging Telecommunications Technologies*, page e4245, 2021. doi: 10.1002/ett.4245.
- A. Rahimi, S. Benatti, P. Kanerva, L. Benini, and J. M. Rabaey. Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition. In *IEEE ICRC*, 2016. doi: 10.1109/ICRC.2016.7738683.
- Abbas Rahimi, Artiom Tchouprina, Pentti Kanerva, José del R. Millán, and Jan M. Rabaey. Hyperdimensional Computing for Blind and One-Shot Classification of EEG Error-Related Potentials. *Mobile Networks and Applications*, 25(5):1958–1969, 2020. doi: 10.1007/s11036-017-0942-6.
- Dieter Schmidt and Matti Sillanpää. Evidence-based review on the natural history of the epilepsies. *Current Opinion in Neurology*, 25(2):159–163, 2012. doi: 10.1097/WCO.0b013e3283507e73.
- V. Shah, I. Obeid, J. Picone, G. Ekladious, R. Iskander, and Y. Roy. Validation of Temporal Scoring Metrics for Automatic Seizure Detection. In *IEEE SPMB*, 2020. doi: 10.1109/SPMB50085.2020.9353631.
- Ali Hossam Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment*. Thesis, Massachusetts Institute of Technology, 2009.
- Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Xiaodi Huang, and Nasir Hussain. A review of epileptic seizure detection using machine learning classifiers. *Brain Informatics*, 7(1):5, 2020. doi: 10.1186/s40708-020-00105-1.
- Antonio Tricoli, Noushin Nasiri, and Sayan De. Wearable and Miniaturized Sensor Technologies for Personalized and Preventive Medicine. *Advanced Functional Materials*, 27(15):1605271, 2017. doi: 10.1002/adfm.201605271.
- Scott A Waldman and Andre Terzic. Healthcare Evolves From Reactive to Proactive. *Clinical pharmacology and therapeutics*, 105(1):10–13, 2019. doi: 10.1002/cpt.1295.
- Saeedeh Ziyabari, Vinit Shah, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. Objective evaluation metrics for automatic classification of EEG events. *arXiv:1712.10107*, 2019.