Enhanced Detection of Transdermal Alcohol Levels Using Hyperdimensional Computing on Embedded Devices

Manuel E. Segura§*, Pere Vergés§*, Justin Tian Jin Chen*, Ramesh Arangott+, Angela Kristine Garcia+,
Laura Garcia Reynoso+, Alexandru Nicolau*, Tony Givargis* and Sergio Gago-Masague*
University of California Irvine*, Asahi Group Holdings Ltd.+
mesegur1@uci.edu, pvergesb@uci.edu, chenjt3@uci.edu, ramesh.arangott@asahigroup-holdings.com,
angela.kristinegarcia@asahigroup-holdings.com, laura.garcia@asahigroup-holdings.com,
nicolau@ics.uci.edu, givargis@uci.edu, sgagomas@uci.edu

Abstract—Alcohol consumption has a significant impact on individuals' health, with even more pronounced consequences when consumption becomes excessive. One approach to promoting healthier drinking habits is implementing just-in-time interventions, where timely notifications indicating intoxication are sent during heavy drinking episodes. However, the complexity or invasiveness of an intervention mechanism may deter an individual from using it in practice. Previous research tackled this challenge using collected motion data and conventional Machine Learning (ML) algorithms to classify heavy drinking episodes, but with impractical accuracy and computational efficiency for mobile devices. Consequently, we have elected to use Hyperdimensional Computing (HDC) to design a just-in-time intervention approach that is practical for smartphones, smart wearables, and IoT deployment. HDC is a framework that has proven results in processing real-time sensor data efficiently. This approach offers several advantages, including low latency, minimal power consumption, and high parallelism. We explore various HDC encoding designs and combine them with various HDC learning models to create an optimal and feasible approach for mobile devices. Our findings indicate an accuracy rate of 89%, which represents a substantial 12% improvement over the current state-of-the-art.

Index Terms—Hyperdimensional Computing, Classification, Machine Learning, Embedded Systems, Internet of Things, Health Monitoring, Time Series Analysis, Accelerometers, Feature Extraction

I. INTRODUCTION

Alcohol is a harmful substance that contributes to more than 200 diseases, injuries, and health conditions [1]. According to the World Health Organization, in 2022, 5.3% of global deaths were attributed to alcohol use, accounting for three million deaths annually. Furthermore, alcohol consumption is known to lead to various mental and behavioral disorders and injuries. The risks associated with alcohol significantly escalate when excessive consumption occurs within short time intervals [2].

Consequently, the timely and accurate detection of intoxication becomes imperative for mitigating these risks. Measurement methods for alcohol levels in the body can be categorized into six major categories: early Nicloux oxidation separation

methods, breath alcohol devices, bodily fluid testing, transdermal sensors, optical techniques, and intoxication estimation machine learning algorithms [3]. In particular, transdermal sensors are designed to gauge alcohol levels by detecting the gaseous phase of alcohol in the air surrounding the skin. These studies [4], [5] have revealed that TAC can accurately identify moderate alcohol consumption, reflected when TAC levels range between 0 and 0.02 g/dl. Nevertheless, they are not reliable indicators of lower drinking levels (i.e. whether an individual has ceased drinking), unless the number of drinks consumed is known [6].

Unlike other categories of body alcohol measurement methods, intoxication estimation machine learning algorithms do not directly measure ethanol, but instead use correlations between intoxication and other physiological measurements [3], making them less intrusive and suitable for timely intervention notifications. This study aims to establish a strong correlation between transdermal alcohol concentration (TAC) measurements and accelerometer data from users' smartphones, achieving accurate results without intruding on the user experience. This is particularly significant given the widespread use of smartphones and the inherent presence of accelerometers in these devices. Providing users with a tool to assess their alcohol levels can heighten awareness and foster responsible drinking habits.

While prior investigations have delved into this subject [7], [8], our objective is to enhance both accuracy and efficiency. Previous studies predominantly leaned on conventional machine learning algorithms, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), or Random Forest [7], [9], for classifying sobriety. Yet, the intricacy of this challenge lies in accurately determining sobriety across diverse scenarios, such as when the smartphone is in a pocket, on a table, or actively in use (e.g., gaming, calling, texting, or using various applications). Moreover, the data may be prone to noise, and signals may intermittently be lost, presenting hurdles in solving this issue. Our aspiration is to develop a less resource-intensive solution that can operate on or with smart wearables such as smartwatches or smart bracelets, enabling

§Both authors contributed equally to this research.

the seamless capture of information.

Hence, our approach involves leveraging Hyperdimensional Computing (HDC) for classification. HDC has demonstrated impressive results in handling noisy and real-time data across different applications. These applications span from emotion recognition [10], hand gesture recognition [11], [12], [13] to language recognition [14] and multiclass classification of electromyography (EMG), electroencephalography (EEG), and electrocorticography (ECoG) [15] biosignals. This technology provides several advantages, including low latency, low power consumption, and high parallelism, making it an excellent framework for addressing challenges within the Internet of Things (IoT) domain.

II. RELATED WORK

The proliferation of smartphones and other mobile technologies in the last 20 years has enabled research in the field of just-in-time adaptive interventions (JITAI). The purpose of JITAIs are to provide personalized real-time interventions for users, typically with a mobile device which also collects data [16]. These have proven highly effective at helping with issues such as weight management [17], managing stress [18], exercise [19], nicotine addiction [20], and alcohol consumption [21], [22]. In particular, the study performed in [21] demonstrated that personalized JITAIs deterred hazardous behavior during drinking episodes.

The application of mathematical models has led to less intrusive methods for predicting BAC or TAC, thereby opening the door for predictive algorithms to be used in JITAIs. Another study [23], employed a mathematical model that incorporates TAC data and the specific types of beverages an individual intends to consume to predict BAC. Furthermore, novel techniques have emerged that utilize machine learning and mobile data, such as accelerometer readings, to provide a cost-effective and practical way of predicting BAC or TAC. These techniques utilize classification algorithms like Random Forest [8], [7], CNNs (Convolutional Neural Networks) [24], [7], and SVMs (Support Vector Machines) [25], [7] to effectively process sensor data and accurately predict levels of intoxication. Similarly, techniques utilizing regression models have also been explored. Bayesian Regression Neural Networks (BRNN) have been used to perform regression analysis for BAC levels from accelerometer data [24], and furthermore, an ensemble of CNNs and bi-directional Long Short Term Memory (bi-LSTM) were also used for regression analysis that yielded promising results [26]. However, these neural networks are often computationally expensive to train and perform inference with, so they were not in the scope of our work for exploring practical methods for real-time IoT applications of JITAIs. Moreover, there was no code publicly available to reproduce the claimed results.

Remarkably, HDC, which has demonstrated notable success in embedded systems and time-series data, has not been previously employed in classification within this domain. HDC has demonstrated significant performance and energy consumption improvement over traditional methods for real-time and online learning, such as sensorimotor control [27] and light weight classifier performance through model quantization [28], model compression [29], and HDC algorithm optimization [30]. As a result, our HDC solution is strategically positioned to deliver JITAIs with effective performance on smartphones and smart wearables.

A. Contribution

Our approach offers a non-intrusive intoxication estimation machine learning algorithm that has a higher accuracy than the state-of-the-art while computationally efficient enough for real-time and online learning on smart wearables. Our solution fills the current research gap in JITAI systems for intoxication by introducing a high-performing model capable of running on embedded devices.

III. HYPERDIMENSIONAL COMPUTING

HDC is a brain-inspired framework that relies on high-dimensional vectors to represent information. These hypervectors have a dimensionality (d) on the order of thousands which are randomly sampled from hyperspace. Depending on the application's requirements, a range of hyperspaces can be selected [31], [32], [33], [34]. These spaces encompass a spectrum from binary to complex value representations. Among them, the Multiply Add Permute (MAP) [32] hyperspace is one of the most popular thanks to its efficiency and information representation capabilities. For that reason, we have selected MAP as the hyperdimensional framework for our application.

A. Distance Metric

Measuring the proximity of two hypervectors is crucial due to their high dimensionality and the importance of orthogonality. This metric determines whether two hypervectors are identical, similar, or vastly dissimilar. In our context, this metric aids in classifying which class a particular hypervector belongs to, facilitating the accurate classification of data points. Typically, the distance metric is computed using the dot product, although, depending on data characteristics and normalization, cosine similarity or hamming distance may also be applied.

B. Operations

HDC relies on three fundamental operations:

a) Superposition: This operation, also known as bundling, aggregates information from various hypervectors into a single hypervector. It can be represented as $\oplus : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{H}$. Notably, the resulting hypervector bears a high resemblance to each hypervector contributing to its formation. This operation is typically implemented as element-wise addition, rendering it high efficiency and parallelizability [31], [35].

b) Binding: This operation binds two variables together, similar to traditional computing's assignment of a value to a variable. It is represented as $\otimes : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{H}$. Unlike superposition, the resulting hypervector is entirely distinct from its components. In vector symbolic architectures, the

inverse operation of binding can be performed by binding the variable again with the same hypervector, effectively restoring it to its original state, therefore making the binding operation its self-inverse.

*c) **Permute***: This operation establishes order among hypervectors, this is vital for creating data structures and capturing data patterns in time series applications [12]. It can be represented as $p^i(\boldsymbol{h})$ where i denotes the number of times the operation is applied. Similar to binding, the resulting hypervector after the operation is dissimilar. The inverse permute operation, $p^{-i}(\boldsymbol{h})$, reverts the hypervector to its exact original state [36], [37].

C. Hyperdimensional Encoding and Classification

The HDC framework has incited interest in solving various machine learning problems, spanning from regression [38], reinforcement learning [39], and classification tasks [40] [41]. The fundamental workflow for hyperdimensional learning includes encoding training samples into the hyperspace, adding each sample to its respective class, and storing this information in associative memory per a chosen hyperdimensional learning model, as illustrated in Figure 1.
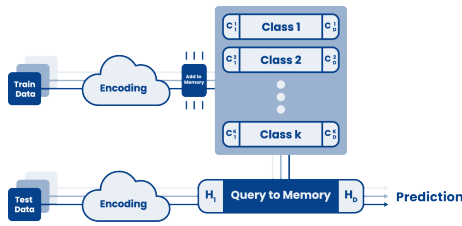

Fig. 1. Hyperdimensional classification workflow.

The encoding phase within the hyperdimensional learning workflow is computationally intensive and significantly impacts classification outcomes. Encodings can be either unstructured or structured. Unstructured encodings are typically created through vector projections, including random projection [42], sinusoid projection [43], or fractional power encoding [44], [45]. Structured encodings employ different types of basis hypervectors, such as random hypervectors, level hypervectors with linear correlations, and circular hypervectors suitable for circular data representation, such as seasonal variations. These basis hypervectors are utilized to create various structures, such as the key-value encoding [46], [47], [42], n-gram encoding [48] and the generic encoding [49], which combines elements of both n-gram and key-value encoding. Moreover, one can create more specific encodings that better tailor their application by combining the aforementioned encodings using hyperdimensional operations.

Once the training phase concludes, inference begins, encoding test samples to the hyperspace, where it's important to note that the encoding process remains consistent for both training and inference. For each encoded sample, a similarity metric is employed to determine the class most similar to the given instance, ultimately facilitating classification. While this serves as the fundamental workflow, several enhancements to this approach have been explored. These enhancements range from improved single-pass learning using adaptive learning [50], [51], to iterative training [51], and the use of dimension regeneration [43], [52], for dimensions carrying minimal to no information.

IV. Methods

The initial stage of our method focused on data treatment, which is a critical step for this application. This was particularly significant because the problem is meant to be addressed in real-time. It was also essential to have a streamlined and efficient method for processing input data, along with the removal of noise. Providing accurate parameters to the HDC classification model is extremely important for obtaining high accuracy results.

A. Data Selection and Cleaning

We utilized the "Bar Crawl: Detecting Heavy Drinking" dataset from a previous study [7], [53]. This dataset, gathered from 11 iOS and 2 Android smartphones and TAC measurements from a SCRAM ankle bracelet, consists of a real-time series with a substantial 14 million accelerometer readings and 715 TAC (Transdermal Alcohol Concentration) readings. The usable data for the analysis consists of the reading of 13 participants [7]. The raw accelerometer data was collected at a sampling rate of 40Hz [7]. The TAC data, obtained at 30-minute intervals using SCRAM ankle bracelets, underwent two processing steps: (1) a zero-phase low-pass filter to reduce noise without shifting phase, and (2) a backward shift of 45 minutes to align labels more closely with the true intoxication of the participant, considering the 45-minute duration for alcohol to exit through the skin [7]. Subsequently, the raw accelerometer data was windowed into 10-second overlapping windows (400 samples per window), with TAC data interpolated for those 10-second intervals [7].

Minimizing the amount of data and treatment done to the data is fundamental to reducing energy and execution time in a real-time application, where fast and efficient responses are necessary to be able to execute the application in a real scenario. We considered the same engineered feature types as used in a previous study [7], which resulted in a total of 600 features. Similar to the previous study, we used Autogluon [54] to identify the top 20% (120) of features with the highest importance scores, which measure how much each feature influenced the resulting prediction. We then used these 120 features in our approach. However, upon further examination, only 18 individual features had Gini importance higher than 0.0001, making clear that most of the 600 features almost did not contribute to the right prediction. The feature types we considered are listed in Table I. The list of types corroborated the importance of Mel Frequency Cepstral Coefficients (MFCC) for this problem from the previous study [7].

B. HDC Approach

With our HDC approach, we first determined how data will be encoded. In the literature [41] we observed that there is

Feature Type	Description
Mean	Average of each x, y, z signal
Standard Deviation	Standard deviation of each x, y, z signal
Median	Median of each x, y, z signal
Spectral Entropy	Entropy in frequency and time domain (2 total metrics) for each axis
Spectral Centroid	Weighted mean of frequencies for each axis
Spectral Spread	Measure of variance about the centroid for each axis
RMS	Root-mean-square of accelerations for each axis
MFCC Covariance	MFCC covariance entries for each of 6 axis combinations

a certain set of general encodings that have proved to work well in similar applications. The four encodings mentioned in Section III-C that we used for raw accelerometer time-series data were the following: *Key-Value (KV) [46], [47], [42], N-gram (N) [48], Generic [49], Sinusoid projection (S. Proj) [43]*.

We used the *Density* encoding [55] for the engineered feature data. This encoding is an alternate implementation of level hypervectors to encode information. Our final ensemble encoding method was a combination of the Density encoding and one of the four mentioned previously.

In addition to developing an encoding method, we needed to determine which learning model would work best for our classification problem. Several works have proposed different classification learning models. These models differ in accuracy and training time. However, inference time is very similar for all of them [41]. We considered 6 different models:

- *VanillaHD (V)*, proposes an approach where all samples are added to the memory without modification. This is a single-pass algorithm.
- *AdaptHD (A) [50]*, This model performs adaptive learning, this is done by adding a sample to the memory if it is misclassified. Can be single-pass or perform iterative learning.
- *OnlineHD (O) [51]*, This model builds upon *AdaptHD* by doing a weighted addition of the samples. It can be single-pass or perform iterative learning.
- *RefineHD (R) [56]*, this model improves upon *OnlineHD* and *AdaptHD* by adding samples that are correctly classified but differ largely from the class. Moreover, it proposes the use of the flocet encoding in the key-value encoding. It can be single-pass or perform iterative learning.
- *NeuralHD (N) [43]*, this model uses dimensions regeneration iterative training, and adaptive learning to improve the model's learning.
- *DistHD (D) [52]*, this model combines dimension regeneration, top two classifications, adaptive learning, and iterative training to achieve better accuracy

From these models, the first four can be performed using single-pass learning which is extremely useful for fast and low-resource training and inference. If the training can be offloaded and performed asynchronously, then performing iterative training can help achieve better results.

The HDC learning models offer a key advantage by allowing the capture of temporal information while training the model. We leveraged this advantage by using the 400-sample raw time-series data windows in our HDC approach, combining it with our engineered features for classification. This is reflected in our HDC encoding selection, where we explored HDC encodings used to encode time-series data in order to develop our own unique encoder. For this study, we used both raw data and feature-engineered data. We used two approaches for this problem, one where we shuffled the windows of all 13 participant and one where we used the windows sorted in chronological order, this second approach was thought to work better thanks to the inherent time dependence that this problem carries. For example, in the HDC implementation of the time-series classification algorithm MiniROCKET named HDC-MiniROCKET, timestamps were able to be encoded in addition to the feature data provided by the convolutions, which allowed for significant accuracy improvements [57]. Similarly, we aimed to consider longer patterns occurring during the bar crawl data collection period to enhance overall accuracy.

C. Outline of Experiments

The outline of our experiments [§] was split into 3 phases. The implementation of each phase of the experiments was done using the Torchhd [58] a python library. The first phase consisted of comparing combinations of the *Density* encoding with the 4 other encodings used for the raw time-series windows. We used *RefineHD* as a default for testing in this phase. This provided us with an optimal encoding method for the second phase, learning model selection. In the second phase we used our finalized ensemble encoding method with our 6 potential learning models to determine the best encoding and learning method combination. Our third and final phase of experiments consisted of refining our final encoding and learning method combination with hyperparameter refinement.

For the execution of our experiments, we employed a Raspberry Pi 4 Model B with a Quad core Cortex-A72 (ARM v8) and 4GB of LPDDR4-3200 SDRAM memory [59]. We chose this platform as it's computational specifications are similar in specifications to a low-end smartphone.

We divided the data into two sets, with a 70/30 ratio. When the data was randomly shuffled, we divided it randomly. For chronologically ordered data, we reserved the last 30%.

V. RESULTS AND DISCUSSION

Our results are split into the three major phases of our experiments: encoding selection, learning model selection, and hyperparameter tuning. We also performed a baseline comparison with the results of the previous study, and explore execution times of the different encodings on the Raspberry Pi hardware. Encodings and methods are referred to by their

[§]Project link: https://github.com/mesegur1/bar_crawl_data

abbreviations listed in Section IV-B when referenced in tables and figures.

A. First Phase Results - Encoding Selection

Our findings for the encoding selection, presented in Table II and Table III, shed light on the encoding selection process. For each encoding that is paired with the *Density* encoding, we recorded the accuracy and F1-score and compare results. The F1-scores were expected lower with the chronologically ordered data, due to TAC levels being lower in the chronologically last 30% of the data (mostly the negative "sober" label). The *Generic* encoding and *Density* encoding combination attains the one of the highest accuracy levels for both shuffled and ordered data. The breakdown of our final ensemble encoder is shown in Algorithm 1.

Algorithm 1: Ensemble HDC Encoder

Input : A window W, composed of raw accelerometer data A, and engineered features F.

Output: A hypervector.

1 // Apply Generic encoding to x, y, and z axis to encode.
2 $H_1 \leftarrow \text{GenericEncode}(A.time, A.x, A.y, A.z)$
3 // Apply Density encoding to 120 engineered features.
4 $H_2 \leftarrow \text{DensityEncode}(F)$
5 // Bind and bundle H_1 and H_2.
6 $H_f \leftarrow H_1 \oplus H_2 \oplus (H_1 \otimes H_2)$
7 **return** H_f

TABLE II
PERFORMANCE EVALUATION OF THE HDC ENCODINGS, WITH RANDOMLY SHUFFLED DATA.

Metric	KV (RL)	S. Proj	KV (SN)	Generic
Accuracy	79.069	78.78	**82.857**	82.41
F1-Score	0.571	0.572	0.67	**0.692**

TABLE III
PERFORMANCE EVALUATION OF THE HDC ENCODINGS, WITH CHRONOLOGICALLY ORDERED DATA.

Metric	KV (RL)	S. Proj	KV (SN)	Generic
Accuracy	89.274	89.476	84.085	**89.47**
F1-Score	0.239	0.242	**0.287**	0.241

We tested the ensemble *Generic* encoding and found it to be effective. However, we also aimed to evaluate its performance using different n-gram sizes as this can greatly impact the algorithm's effectiveness. To achieve this, we experimented with sizes ranging from 2 to 7, as studies have shown that this range tends to yield optimal results [41]. According to Figure 2, we can conclude that the optimal n-gram size achieved is 6, particularly when the input data is shuffled. However, if the data is ordered, it is preferable to use an n-gram of size 2, since it provides the same efficiency while

being more efficient and will result in faster training and inference. For the purposes of the next experiments, we use n-gram size 6.
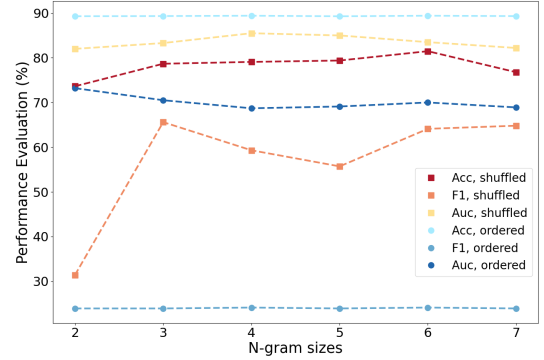

Fig. 2. Performance evaluation of the generic encoding with different n-gram sizes, with randomly shuffled and chronologically ordered data.

B. Second Phase Results - Learning Model Selection

Combining the ensemble *Generic* encoding with the learning models described in Section IV-B, we again experimented with both randomly shuffled and chronologically ordered data, and collected accuracy and F1-metrics as shown in Table IV and Table V. Using the *RefineHD* model had one of the highest accuracy scores among the different models, for both shuffled and ordered data. Similar to the previous tests, the F1-Score was lower for the ordered data due to the negative/positive label ratio. We decided to use this combination for our experimentation in the upcoming section.

TABLE IV
PERFORMANCE EVALUATION OF THE HDC CLASSIFICATION MODELS, WITH RANDOMLY SHUFFLED DATA.

Metric	V	A	O	R	N	D
Accuracy	66.63	78.62	82.62	**84.00**	82.34	76.132
F1-Score	0.51	0.69	0.56	**0.70**	0.70	0.66

TABLE V
PERFORMANCE EVALUATION OF THE HDC CLASSIFICATION MODELS, WITH CHRONOLOGICALLY ORDERED DATA.

Metric	V	A	O	R	N	D
Accuracy	47.18	**89.73**	78.75	89.47	89.69	62.94
F1-Score	0.139	**0.24**	0.23	0.24	**0.24**	0.14

C. Third Phase Results - Hyperparameter Tuning

After deciding on an ensemble *Generic* encoding with an n-gram size of 6 and choosing the HDC learning model (*RefineHD*), the next important step was to determine the optimal learning rate for the iterative version. We experimented with learning rates ranging from 1 to 5 for both shuffled and ordered data to find the best option. The learning rates for HDC learning models are typically larger than those used for conventional ML models.

Metric	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Accuracy	76.64	81.85	**82.41**	81.30	78.63
F1-Score	0.659	0.637	**0.692**	0.684	0.628

TABLE VII
PERFORMANCE EVALUATION OF THE ENSEMBLE GENERIC ENCODING
AND REFINEHD MODEL USING DIFFERENT LEARNING RATES, WITH
CHRONOLOGICALLY ORDERED DATA.

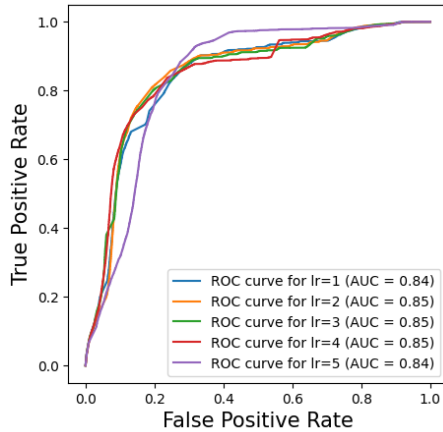| Metric	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Accuracy	89.43	89.33	**89.47**	89.38	89.4
F1-Score	0.241	0.239	**0.241**	0.24	0.24

Fig. 3. ROC curve evaluation of the ensemble Generic encoding and RefineHD model with different learning rates, with randomly shuffled data.

TABLE VIII
COMPARISON OF PRECISION, RECALL, AND SOBER/DRUNK ACCURACY
BREAKDOWN WITH LEARNING RATE $\alpha = 3$, WITH RANDOMLY SHUFFLED
AND CHRONOLOGICALLY ORDERED DATA.

Metric	Randomly Shuffled	Chronologically Ordered
Accuracy	82.41	89.47
Sober Accuracy	86.12	93.33
Drunk Accuracy	72.63	28.51
Precision	0.661	0.209
Recall	0.726	0.285

Upon examining the results presented in Table VI, Table VII, and the receiver operating characteristic (ROC) curves in Figure 3, it becomes evident that setting the learning rate to 3 yields the highest levels of accuracy with both randomly shuffled data and chronologically ordered data. Looking further into the performance with learning rate $\alpha = 3$ in Table VIII, we consider accuracy, including sober and drunk accuracy, precision, and recall metrics. Although the precision and recall were lower in the chronologically ordered case, due to the negative/positive label ratio, We can see that the classifier excels at the classification of the negative case (sober).

D. Baseline Comparison with Original Study

As a preliminary step, we sought to replicate the previous study and make a comparative assessment of our approach. In this specific experiment, we compared the results of our HDC approach using randomly shuffled data and chronologically ordered data with results of the methods of the previous study, including their results for their SVM, CNN, and random forest approaches [7]. Table IX provides a comprehensive view of the results achieved by both methods. Our approach has higher accuracy scores than previous used models, and also higher recall when the data is randomly shuffled. When the data is chronologically ordered, precision and recall drop as before because of the negative/positive label ratio, but the sober accuracy (negative label) is much higher. Notably, our approach demonstrates a significant improvement when the data is chronologically ordered, boasting a 12% increase in general accuracy and a 12% enhancement in sober accuracy compared to the results of the best performing random forest model from the previous study.

TABLE IX
PERFORMANCE EVALUATION OF THE OUR HDC APPROACH (RANDOMLY
SHUFFLED DATA (S) AND CHRONOLOGICALLY ORDERED DATA (O)) VS
RESULTS OF PREVIOUS STUDY.

Metric	SVM [7]	CNN [7]	RF [7]	HDC (S)	HDC (O)
Accuracy	74.67	74.27	77.48	82.39	**89.47**
Sober Acc	0.812	0.789	0.815	0.860	**0.933**
Precision	0.636	0.636	**0.665**	0.661	0.209
Recall	0.622	0.656	0.698	**0.725**	0.285

Finally with the final combination of our ensemble *Generic* encoder with n-gram size 6 and a learning rate 3 with the *RefineHD* method we obtain an 89% accuracy when the data is ordered chronologically which surpasses the previous work by **12%**.

E. Hardware Execution Times per Window

To verify the feasibility of our approach for classifying live data in real-time, we measured the average execution time of classifying 1 10 second window with combinations of the four encodings and the *RefineHD* model on the Raspberri Pi. In order to be feasible, classification for a 10 window needs to be significantly less than 10 seconds. We found that our chosen ensemble *Generic* encoding approach with *RefineHD* took on average 0.34 seconds on the Raspberry Pi, as shown in Table X.

TABLE X
ENCODINGS EXECUTION TIME PER SAMPLE.

Metric	KV (RL)	S. Proj	KV (SN)	Generic
Time (s)	0.3	**0.1**	0.71	0.34

Our experiments on the Raspberry Pi 4 indicate the potential for implementing our HDC approach on a smartphone, or even a smartphone-smartwatch duo. This paves the way for immediate assessments and JITAIs, allowing smartphones to

process sensor data from smartwatches and conduct classification without relying on more computing-intensive resources such as cloud servers.

VI. FUTURE WORK

To address the shortcomings of using only accelerometer data collected from smartphones, we are expanding our consideration to additional sensor measurements besides accelerometer data, collected by smart watches or other smart wearable devices. These will include heart-rate, gyroscope, temperature, and blood oxygenation level measurements, paired with TAC measurements from a TAC measuring bracelet for correlation. These new measurements have been found to have some correlation to alcohol levels and intoxication [60]. We are developing a mobile application for data collection of these measurements from these smart wearables as part of a larger future study.

Once we have a new dataset collected from smart wearables, our next steps would be running our HDC approach on smart wearable devices with the new dataset to classify TAC measurements, and further refining our approach with the results. Then, we can modify our smart wearable application for reading live data. Our end goal is a complete JITAI solution that can be deployed to consumers, and is both affordable and accessible to the general public, ultimately leading to a more significant impact on promoting healthier drinking habits.

VII. CONCLUSIONS

Alcohol consumption significantly impacts an individual's well-being, with more pronounced consequences when consumed excessively. Various strategies aim to promote responsible drinking habits; among these, JITAIs stand out as a promising approach. To address this issue, we proposed the use of HDC to test the feasibility of IoT-based JITAIs that could address a wide range of alcohol consumption issues. Our research demonstrated that hyperdimensional learning outperforms the traditional random forest model in this context. After exploring various encoding methods and hyperdimensional learning techniques, we found that the optimal configuration involves using a generic encoding with an n-gram size of 6, especially when combined with the RefineHD hyperdimensional learning method.

Our study shows significant improvement with an accuracy rate of 89%, which is 12% higher than previous results. Additionally, our work corroborates that organizing samples temporally leads to an additional +7% accuracy, highlighting the advantageous leverage that HDC gains from time series data. In summary, this enables an accurate JITAI solution that would be widely available to the general public, and would be less invasive and cumbersome and provide less stigma when used in practice.

REFERENCES

[1] J. M. Wang HJ, Zakhari S, "Alcohol, inflammation, and gut-liver-brain interactions in tissue damage and disease development," in *World J Gastroenterol*, 2010.

[2] P. Mathurin and P. Deltenre, "Effect of binge drinking on the liver: An alarming public health issue?" *Gut*, 2009.

[3] S. Paprocki, M. Qassem, and P. A. Kyriacou, "Review of ethanol intoxication sensing technologies and techniques," *Sensors*, vol. 22, 9 2022.

[4] R. J. et al, *Using Transdermal Alcohol Monitoring to Detect Low-Level Drinking*, 2015.

[5] ——, *Processing transdermal alcohol concentration (TAC) data to detect low-level drinking*, 2018.

[6] J. D. Roache, T. E. Karns-Wright, M. Goros, N. Hill-Kapturczak, C. W. Mathias, and D. M. Dougherty, "Processing transdermal alcohol concentration (tac) data to detect low-level drinking," *Alcohol*, vol. 81, pp. 101–110, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0741832918301101

[7] J. Killian, K. M. Passino, A. Nandi, D. R. Madden, and J. D. Clapp, "Learning to detect heavy drinking episodes using smartphone accelerometer data," in *KHD@IJCAI*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201666442

[8] S. S. Chawathe, "Using accelerometers in mobile phones to estimate blood alcohol levels," in *2020 IEEE International Smart Cities Conference (ISC2)*, 2020, pp. 1–8.

[9] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, sep 2018. [Online]. Available: https://doi.org/10.1145/3234150

[10] E.-J. Chang, A. Rahimi, L. Benini, and A.-Y. A. Wu, "Hyperdimensional computing-based multimodality emotion recognition with physiological signals," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019, pp. 137–141.

[11] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, F. L. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey, "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," *Nature Electronics*, vol. 4, pp. 54–63, 2020.

[12] A. Moin, A. Zhou, S. Benatti, A. Rahimi, L. Benini, and J. M. Rabaey, "Analysis of contraction effort level in emg-based gesture recognition using hyperdimensional computing," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2019, pp. 1–4.

[13] 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 *2016 IEEE International Conference on Rebooting Computing (ICRC)*, 2016, pp. 1–8.

[14] A. e. a. Rahimi, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ser. ISLPED '16. New York, NY, USA: Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/2934583.2934624

[15] R. A. et al, "Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of exg signals," *Proceedings of the IEEE*, 2019.

[16] M. Reichert, U. Braun, S. Lautenbach, A. Zipf, U. Ebner-Priemer, H. Tost, and A. Meyer-Lindenberg, "Studying the impact of built environments on human mental health in everyday life: methodological developments, state-of-the-art and technological frontiers," *Curr. Opin. Psychol.*, vol. 32, pp. 158–164, Apr. 2020.

[17] K. Patrick, F. Raab, M. A. Adams, L. Dillon, M. Zabinski, C. L. Rock, W. G. Griswold, and G. J. Norman, "A text message-based intervention for weight loss: randomized controlled trial," *J. Med. Internet Res.*, vol. 11, no. 1, p. e1, Jan. 2009.

[18] Smyth and Heron, "Is providing mobile interventions "just-in-time" helpful? an experimental proof of concept study of just-in-time intervention for stress management," in *2016 IEEE Wireless Health (WH)*, 2016, pp. 1–7.

[19] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay, "Activity sensing in the wild: a field trial of ubifit garden," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1797–1806. [Online]. Available: https://doi.org/10.1145/1357054.1357335

[20] W. Riley, J. Obermayer, and J. Jean-Mary, "Internet and mobile phone text messaging intervention for college smokers," *J Am. Coll. Health*, vol. 57, no. 2, pp. 245–248, Sep. 2008.

[21] D. H. Gustafson, F. M. McTavish, M.-Y. Chih, A. K. Atwood, R. A. Johnson, M. G. Boyle, M. S. Levy, H. Driscoll, S. M. Chisholm, L. Dillenburg, A. Isham, and D. Shah, "A smartphone application to support recovery from alcoholism: a randomized clinical trial," *JAMA Psychiatry*, vol. 71, no. 5, pp. 566–572, May 2014.

[22] I. Nahum-Shani, S. N. Smith, B. J. Spring, L. M. Collins, K. Witkiewitz, A. Tewari, and S. A. Murphy, "Just-in-time adaptive interventions (JITAIs) in mobile health: Key components and design principles for ongoing health behavior support," *Ann. Behav. Med.*, vol. 52, no. 6, pp. 446–462, May 2018.

[23] J. Baldwin, *Wearable blood alcohol measuring device, US Patent App.*, 2014.

[24] P. Gharani, B. Suffoletto, T. Chung, and H. A. Karimi, "An artificial neural network for movement pattern analysis to estimate blood alcohol content level," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/12/2897

[25] G. et al, "Real-time prediction of blood alcohol content using smartwatch sensor data." *Smart Health*, 2016.

[26] R. Li, G. P. Balakrishnan, J. Nie, Y. Li, E. Agu, K. Grimone, D. Herman, A. M. Abrantes, and M. D. Stein, "Estimation of blood alcohol concentration from smartphone gait data using neural networks," *IEEE Access*, vol. 9, pp. 61 237–61 255, 2021.

[27] A. Mitrokhin, P. Sutor, C. Fermüller, and Y. Aloimonos, "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception," *Science Robotics*, vol. 4, p. eaaw6736, 05 2019.

[28] M. Imani, S. Bosch, S. Datta, S. Ramakrishna, S. Salamat, J. M. Rabaey, and T. Rosing, "Quanthd: A quantization framework for hyperdimensional computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2268–2278, 2020.

[29] J. Morris, M. Imani, S. Bosch, A. Thomas, H. Shu, and T. Rosing, "Comphd: Efficient hyperdimensional computing using model compression," 07 2019.

[30] B. Khaleghi, H. Xu, J. Morris, and T. Rosing, "tiny-hd: Ultra-efficient hyperdimensional computing engine for iot applications," in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2021, pp. 408–413.

[31] T. A. Plate, *Holographic Reduced Representation: Distributed representation for cognitive structures*, 2003, vol. 150.

[32] R. W. Gayler, "Multiplicative binding, representation operators & analogy (workshop poster)," 1998.

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

[34] P. Kanerva *et al.*, "Fully distributed representation," *PAT*, vol. 1, no. 5, p. 10000, 1997.

[35] P. Kanerva, *Sparse distributed memory*, 1988.

[36] T. Cohen and D. Widdows, "Bringing order to neural word embeddings with embeddings augmented by random permutations (EARP)," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 465–475. [Online]. Available: https://aclanthology.org/K18-1045

[37] P. Vergés, I. Nunes, M. Heddes, T. Givargis, and A. Nicolau, "Accelerating the permute and n-gram operations for hyperdimensional learning in embedded systems," EasyChair Preprint no. 11190, EasyChair, 2023.

[38] A. Hernández-Cano, C. Zhuo, X. Yin, and M. Imani, "Reghd: Robust and efficient regression in hyper-dimensional learning system," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 7–12.

[39] H. Chen, M. Issa, Y. Ni, and M. Imani, "Darl: Distributed reconfigurable accelerator for hyperdimensional reinforcement learning," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3508352.3549437

[40] P. Vergés, M. Heddes, I. Nunes, T. Givargis, and A. Nicolau, "Hdcc: A hyperdimensional computing compiler for classification on embedded systems and high-performance computing," 2023.

[41] ——, "Classification using hyperdimensional computing: A review with comparative analysis," *Computing in Science & Engineering*.

[42] A. Thomas, S. Dasgupta, and T. Rosing, "A theoretical perspective on hyperdimensional computing," *Journal of Artificial Intelligence Research*, vol. 72, pp. 215–249, oct 2021.

[43] Z. Zou, Y. Kim, F. Imani, H. Alimohamadi, R. Cammarota, and M. Imani, "Scalable edge-based hyperdimensional learning system with brain-like neural adaptation," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15.

[44] E. P. Frady, D. Kleyko, C. J. Kymn, B. A. Olshausen, and F. T. Sommer, "Computing on functions using randomized vector representations," 2021.

[45] T. Plate, "Distributed representations and nested compositional structure," 08 2000.

[46] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[47] J. Morris, R. Fernando, Y. Hao, M. Imani, B. Aksanli, and T. Rosing, "Locality-based encoder and model quantization for efficient hyperdimensional computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 4, pp. 897–907, 2022.

[48] D. Kleyko and E. Osipov, "Brain-like classifier of temporal patterns," in *2014 International Conference on Computer and Information Sciences (ICCOINS)*, 2014, pp. 1–6.

[49] B. Khaleghi, J. Kang, H. Xu, J. Morris, and T. Rosing, "Generic: Highly efficient learning engine on edge using hyperdimensional computing," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1117–1122. [Online]. Available: https://doi.org/10.1145/3489517.3530669

[50] M. Imani, J. Morris, S. Bosch, H. Shu, G. D. Micheli, and T. Rosing, "Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2019, pp. 1–4.

[51] A. Hernández-Cano, N. Matsumoto, E. Ping, and M. Imani, "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 56–61.

[52] J. Wang, S. Huang, and M. Imani, "Disthd: A learner-aware dynamic encoding method for hyperdimensional classification," 2023.

[53] "Bar Crawl: Detecting Heavy Drinking," UCI Machine Learning Repository, 2020, DOI: https://doi.org/10.24432/C5TK6G.

[54] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," 2020.

[55] D. Kleyko, M. Kheffache, E. P. Frady, U. Wiklund, and E. Osipov, "Density encoding enables resource-efficient randomly connected neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3777–3783, 2021.

[56] P. Vergés, T. Givargis, and A. Nicolau, "Refinehd: Accurate and efficient single-pass adaptive learning using hyperdimensional computing," EasyChair Preprint no. 11191, EasyChair, 2023.

[57] K. Schlegel, P. Neubert, and P. Protzel, "Hdc-minirocket: Explicit time encoding in time series classification with hyperdimensional computing," 2022.

[58] M. Heddes, I. Nunes, P. Vergés, D. Kleyko, D. Abraham, T. Givargis, A. Nicolau, and A. Veidenbaum, "Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures," *Journal of Machine Learning Research*, vol. 24, no. 255, pp. 1–10, 2023. [Online]. Available: http://jmlr.org/papers/v24/23-0300.html

[59] RaspberryPi. (2023) Raspery pi 4 model b specs. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[60] S. Paprocki, M. Qassem, and P. A. Kyriacou, "Review of ethanol intoxication sensing technologies and techniques," *Sensors*, vol. 22, no. 18, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/18/6819