Smartwatch-Based Prediction of Transdermal Alcohol Levels Using Hyperdimensional Computing

Pere Vergés*, Manuel E. Segura*, Ramesh Arangott+, Angela Kristine Garcia+,
Laura Garcia Reynoso+, Alexandru Nicolau*, Tony Givargis* and Sergio Gago-Masague*
University of California Irvine*, Asahi Group Holdings Ltd.+
pvergesb@uci.edu, mesegur1@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*—Excessive alcohol consumption was responsible for 6% of global deaths in 2023. To encourage healthier drinking habits and enhance user awareness of their current condition, just-in-time interventions prove to be a suitable approach for informing users about their current state of intoxication. Current methods for determining blood alcohol content are intrusive and many also invasive, requiring users to use breathalizers or actively engage in urine or blood tests. In this study, we introduce an application utilizing Hyperdimensional Computing to predict if a user is under the influence of alcohol, achieving an accuracy of 93.5% on average. Furthermore, this application is designed to run on both smartphones and smartwatches, enabling full on device computation and online learning through a C implementation utilizing vectorial operations. The application has shown to be very efficient, having a training time per instance of 13.2 and 1.25ms on smartwatch and smartphone respectively and inference time of 6.8 and 1.1ms. Moreover the energy consumption of the running application is negligible compared to the energy usage of the idle device.**

***Index Terms*—Hyperdimensional Computing, Smartwatch, IoT**

I. Introduction

Alcohol abuse accounted for 6% of global deaths in 2023, with nearly 14% of fatalities occurring among individuals aged 20 to 39 years [1]. This alarming statistic underscores the significant impact of alcohol on human health, contributing to conditions like liver cirrhosis, cancer, and pancreatitis [1, 11]. Moreover, behavioral changes induced by alcohol, such as those leading to motor vehicle accidents and episodes of anger [32], further compound the detrimental effects.

While various methods, including blood, urine, and saliva tests and devices like breathalyzers, exist to detect current intoxication levels, these approaches share a common drawback—intrusiveness. Consequently, their applicability in everyday scenarios is limited, making them unsuitable for facilitating behavioral changes that could deter individuals from engaging in heavy drinking or undertaking hazardous activities, such as driving motorized vehicles.

This study endeavors to offer a solution for implementing just-in-time adaptive interventions (JITAI) when users surpass a blood alcohol concentration (BAC) of 0.08g/dL. JITAIs aims to provide personalized real-time interventions for users, typically with a mobile device. The end goal of our study is to mitigate instances of heavy drinking and alleviate associated consequences. Additionally, our application has the potential to address lower alcohol concentration levels, thereby enhancing individuals' awareness of their alcohol consumption habits.

While prior research has focused on estimating blood alcohol levels through mobile phone accelerometer data [16], we aim to present a practical application demonstrating the feasibility of conducting online learning and inference directly on the device. Our project extends beyond a mobile application to include a smartwatch application. Leveraging the additional sensor data available from smartwatches. Particularly the user's heart rate has been shown to correlate with blood alcohol levels [18], and this may allow us to achieve more reliable results.

In contrast to earlier studies employing machine learning algorithms like Support Vector Machines, Random Forests [16], or Convolutional Neural Networks [25], we adopt a Hyperdimensional Computing (HDC) [15] approach for this problem. HDC, also referred to as vector symbolic architectures (VSA) [9], offers a unique neuro-symbolic methodology. It translates the input information into distributed representations using high-dimensional randomized vectors [10], which function as holographic representations of information, wherein dimensions are independent and identically distributed (i.i.d). Each dimension carries an equal amount of information, leading to inherent robustness to noise [15].

This framework proves highly suitable for several reasons: (1) Robustness to errors and noisy data, crucial in this application where intermittent data loss occurs and all computations and data gathering transpire on an embedded device [15]. (2) Low-latency and high parallelization [38, 36], facilitating efficiency in embedded devices. (3) HDC has demonstrated strong performance in classifying tasks involving signal data, such as Electromyography (EMG)[23], Electroencephalography (EEG), Electrocorticography (ECoG), and notably in estimating transdermal alcohol concentration (TAC)[29].

This study introduces a mobile and smartwatch Android application that employs HDC to classify whether a user is sober or intoxicated (when the BAC levels surpass 0.08 g/dL). Notably, the proposed application is specifically designed to

[1]Source: https://drugabusestatistics.org/alcohol-abuse-statistics/

conduct both online learning and inference directly on the targeted device, eliminating the need for external computation. This capability is made possible through an optimized C code implementation, leveraging loop optimization techniques and a vectorial implementation, resulting in notable efficiency gains.

II. Related Work

Methods for assessing alcohol content can be categorized into six major categories: early methods, breath alcohol devices, bodily fluid testing, transdermal sensors, optical techniques, and intoxication estimation machine learning algorithms [24]. Methods as early as 1918 used Nicloux oxidation separation, and required invasive blood sampling and tracking of alcohol consumption [2]. Bodily fluid testing, such as gas chromotography [22], also require invasive fluid sampling. Breath alcohol devices measure the ratio of of alveolar air to ethanol volume [40]. Transdermal sensors use electrical sensing or enzymatic reaction rate principles to measure alcohol intoxication through the skin [7]. Optical techniques, such as spectroscopic measurement of exhaled air and tissue [8], check the state of tissue rather than the emissions of ethanol.

Unlike the previously listed methods, intoxication estimation algorithms do not directly measure ethanol, but instead rely on correlations between intoxication and other physiological measurements, established with machine learning methods [24]. Previous work using photoplethysmography (PPG) and electrocardiogram (ECG) measurements with machine learning yielded promising accuracy in intoxication detection [39], and were also being considered for automobile steering systems [26]. Facial temperature measurements through thermal imaging cameras [19], and bioimpedances [6] through wearables have also been explored with promising results. Common smartphone and smartwearable measurements such as motion readings have also been explored with traditional machine learning models [16, 4]. With the advent of new sensors shipped with smartphone and smartwearable technology, the possibility of including these machine learning models into a non-invasive JITAI mobile system becomes more feasible [28].

However, the practicality of using machine learning implemented on smartwearables for JITAIs relies on the computational efficiency of the models. For example, in the realm of Human Activity Recognition (HAR), Shoaib et.al. applied various ML methods with smartphone sensor data [31, 30], but did not implement these on a smartphone platform. In a survey of HAR methods, Lima, et.al. compared "shallow" ML methods, which require manual feature engineering, with deep learning methods that will extract features and build the model at the same time [20]. Nevertheless, they expressed concern for power consumption for implementations on smartphones, and the tradeoffs between energy efficiency and classification model accuracy, citing that feature extraction and data frequency can negatively impact smartphone battery life [20].

On the other hand, HDC has demonstrated notable success in embedded systems and time-series data, with its low energy consumption and high accuracy across a diverse array of applications, as described previously. The HDC framework has incited interest in solving various machine learning problems, spanning from regression [14], reinforcement learning [5], and classification tasks [38, 37, 35].

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. 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. We will show that our HDC solution is strategically positioned to deliver JITAIs with effective performance on smartphones and smartwearables.

III. App Design and Implementation

In addressing the problem at hand, we employ a machine learning classifier based on HDC, implemented as an Android application. Using HDC for this task has shown great accuracy results [29], moreover this framework has shown great results in terms of energy, memory and efficiency results [38]. Our method aims to take advantage and improve on this features to create an application that is minimally intrusive to the user and achieves great accuracy and efficiency results.

A. Application Design

The design of the application prioritizes minimal user intrusion and optimal efficiency, given its need to operate throughout the day. Moreover, this application can be easily modified to allow users to track specific times, such as periods when they are likely to consume drinks (e.g., during working hours) and try to improve its accuracy. The application aims for efficiency by conducting online learning without relying on cloud computation, utilizing the smartwatch's computational power exclusively. To achieve this, we implemented the code in C, utilizing the language's default libraries, ensuring the code is entirely self-contained. The implementation is tailored for the Android Operating System (OS), leveraging intrinsic vector operations from Android OS to accelerate computations. Figure 1 provides a general overview of the workflow, illustrating the generation of C code, the Android application's calls to the C code with accelerometer data for training or inference, and the C model returning the predicted data in the case of inference.

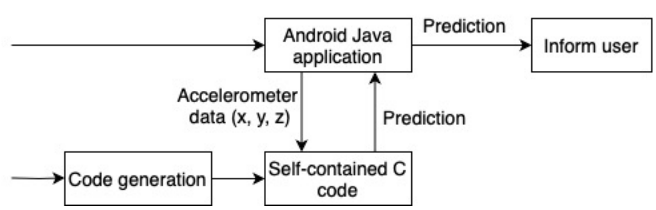


Fig. 1: Online learning smartwatch application workflow.

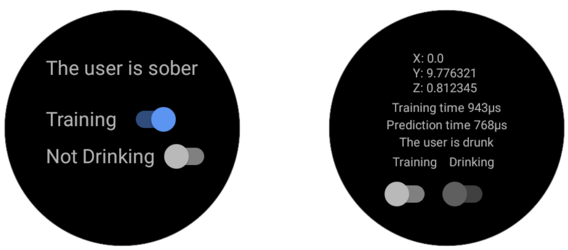
B. Classifier Design

The initial step in tackling this problem involves determining the data encoding strategy for the high-dimensional space.

Our goal is to devise an encoding method that minimizes computation time. To achieve this objective, we assessed two different encodings Level-ID [33] and the n-gram [17] paired with sinusoid [27] functions. Previous studies have indicated the efficacy of the n-gram encoding in achieving high accuracy. After evaluating both options using the Bar Crawl dataset [3], where both encodings yielded approximately 79% accuracy on average on the 6 selected subjects, we opted for the Level-ID encoding. Its superior efficiency in general and ease of implementation with vectorial operations influenced our decision. Additionally, we applied adaptive learning [13, 34] for the classifier, considering that most of the data consists of sober states, with only a small amount during intoxication. Adaptive learning helps prevent the model from saturating with sober values. A larger learning rate was chosen for the same reason, leading to a final accuracy of 93% on average. This prototyping phase utilized Torchhd [12].

1) *Application Implementation*

We developed using Android Studio, but to enhance efficiency, computations are executed using C code, which will be called from Android code. This choice is driven by C's inherent efficiency and flexibility to optimize the code for reduced computation time.



(a) Simple frontend. (b) Complex frontend.

Fig. 2: Online learning smartwatch application using HDC.

The application workflow begins with the user wearing the device (smartwatch or smartphone). Once data collection starts, the user can choose to either train the model or initiate data inference. Increased training and new data contribute to model improvement. During training, users can specify whether they are drinking or not. In inference mode, the model predicts the user's state, allowing to create different application and interventions regarding alcohol consumption. For example, one intervention could be alerting the user if they are intoxicated when they are about to drive a motor vehicle. An example of the application's frontend is depicted in Figure 2, in Figure 2b is shown the frontend with accelerometer data and execution times.

The code is exclusively implemented in C, using only C standard libraries to ensure a self-contained codebase. Loop optimization techniques are applied to boost the efficiency of translating accelerometer data into high-dimensional space. Due to high dimensionality (which typically ranges from 1000 to 50000), optimizing loop execution is crucial.

```
f32 *e(f32 *a, f32 *b, f32* enc){
    for(int i = 0; i < INPUT_DIM; ++i){
        for(int j = 0; j < N; j++){
            enc[j] += a[(N * i) + j] * b[N + j];}
    return enc;
}
```

Listing 1: Optimized encoding (f32 means float32x4_t, and N is the number of vectorial blocks).

```
float* f(float *a, float* indices, float* enc){
  for(int i = 0; i < INPUT_DIM; ++i){
    for(int j = 0; j < DIM; j++){
      enc[(DIM*i)+j]=a[((int)indices[i]*DIM)+j];}
 return enc;
}
float *multibind(float *a, float *b){
  for(int i = 0; i < INPUT_DIM; ++i){
    for(int j = 0; j < DIM; j++){
      b[(DIM*i)+j]=a[(DIM*i)+j] * b[(DIM*i)+j];}}
  return b;
}
float *multiset(float *a){
  for(int i = 1; i < INPUT_DIM; i++){
    for(j = 0; j < DIMENSIONS; ++j){
      a[j] += a[(DIMENSIONS * i) + j];}}
  return a;
}
void encoding(...){
    float *enc=(float*)calloc(DIM*INPUT_DIM,4);
    enc = forward(SIGNALS, indices, enc);
    enc = multibind(CHANNELS, enc);
    enc = multiset(enc);
}
```

Listing 2: Unoptimized encoding.

We employ loop optimization techniques such as loop permutation and loop merging to bolster code efficiency. Combining these two techniques reduces the number of cache misses by 146 times during encoding. By merging the binding and bundling components of the ID-Level encoding and iterating first on the input dimension and then on the dimensions of the hypervector, we achieve a notable reduction in cache misses, in Listing 1, we show how the encoding is optimized in comparison to Listing 2 to reduce memory usage and number of loops by combining forward pass and bundling and binding operations and looping using vectorial block through the dimensions. Additionally, this approach lowers memory usage by accumulating the encoding result into a single hypervector. This avoids the need for a big matrix of hypervector size times input data size, which would be necessary to store all the data from the binding operations.

Furthermore, to leverage the high dimensionality, we've implemented a version of the code that incorporates Single Instruction Multiple Data (SIMD) vector operations for accelerated execution. This implementation is realized through the Android NDK, supporting ARM Advanced SIMD operations (Neon) that extend the ARMv7 and ARMv8 instruction set. Among the supported SIMD types, we opted for *float32x4_t*, as it computes the largest number of elements in a single operation. The use of floating-point values is necessary due to the adaptive learning algorithm, which employs weights to add hypervectors to the classifier.

The application's general workflow begins by reading accelerometer data from either a smartwatch or smartphone through Android Java code. The application will then call C

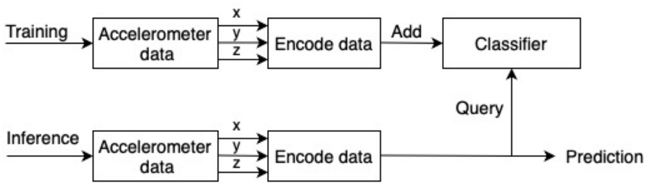


Fig. 3: Online learning TAC classification workflow.

code to either perform data training or inference. For both cases, the data will be encoded to hyperspace, as shown in Figure 3. Depending on whether the data is for training or inference, the sample will be added to the classifier. If the sample does not have new information, it will not be added to avoid saturation. In the case of inference, the encoded data will be queried to the classifier using the cosine similarity metric to determine if the user is sober or drunk. The C call will return the prediction, which will be displayed to the user through the application.

The application's general workflow commences with the Android Java code reading accelerometer data from the smartwatch or smartphone. Subsequently, the application initiates a call to the C code to perform either training or data inference. As illustrated in Figure 3, the data undergoes encoding into the hyperspace. Depending on whether it's training or inference, the sample is either added to the classifier (unless it lacks new information to prevent saturation) or the encoded data is queried to the classifier using the cosine similarity metric to ascertain whether the user is sober or intoxicated. The C call then returns the prediction, which is displayed to the user through the application.

IV. Approach

To demonstrate the performance of the proposed app using HDC for BAC detection, we conducted evaluations using the Bar Crawl dataset for heavy drinking [3]. Six candidates with error-free data gathering were selected for each experiment, and performed 5 repetitions. We assessed the performance on three devices: Samsung Galaxy Watch4 (SG WATCH4): Equipped with an Exynos W920 Dual Core 1.18GHz processor, 1.5GB RAM, 16GB internal storage, and a 247mAh battery. Samsung Galaxy S20 (SG S20): Features an Octa-core processor ranging from 2GHz to 2.73GHz, 8GB RAM, 128GB internal storage, and a 4000mAh battery. Samsung A32 Lite (S A32): Comes with an Octa-core processor ranging from 1.8GHz to 2GHz, 4GB RAM, 64GB internal storage, and a 5000mAh battery.

V. Results

This section presents the evaluation of our model in terms of accuracy and efficiency to validate the proposed implementation.

A. Code validation

The initial experiment aimed to assess the correctness of our code. We prototyped the code using Torchhd and compared the results with our C implementation on both the smartwatch and smartphone, validating both scalar and vectorial versions. Results were consistent across all executions and devices, closely aligning with values obtained from the prototyped code using Torchhd. This experiment not only shows the correctness of our code but also underscores the potential and suitability of HDC for performing BAC classification.

In this specific task, we predicted the sobriety status of six subjects from a relevant study [3]. The dataset was split into 70% training and 30% testing. As depicted in Table I, both the prototyped versions and our final code exhibited virtually exact accuracy demonstrating the validity of our code implementation, averaging around 93.5%.

Subjects	BK	CC	MC	MJ	SA	SF
Torchhd	100%	100%	100%	62.76%	100%	100%
Our code	99.72%	100%	100%	62.76%	99.96%	100%

TABLE I: Code validation, comparing prototype version using Torchhd and our version executed in the smartwatch and smartphones in terms of accuracy.

B. Scalar and vectorial evaluation

In this experiment we evaluate the code's efficiency in processing accelerometer data for both training and inference. Our goal is to measure the speedup achieved by using vectorial operations compared to scalar operations in two versions of the code. The evaluation is split into training and inference, as detailed in Section III. Results for training are presented in Table II, while Table III provides results for inference. These results show that HDC is very efficient at the task in hand and can perform online learning on device. Moreover we see a speedup between 1.2x to 2.1x when performing vectorial operations, this is the expected results since only a part of the application is improved with vectorial operations and the limited general resources of the embedded device.

Training Time	SG WATCH4	SG S20	S A32
Scalar	19.094ms	1.563ms	2.088ms
Vectorial	13.251ms	1.267ms	1.386ms
Speedup	1.4409ms	1.233ms	1.506ms

TABLE II: Training efficiency results of scalar and vectorial.

Inference Time	SG WATCH4	SG S20	S A32
Scalar	14.626ms	0.853ms	2.210ms
Vectorial	6.818ms	0.575ms	1.627ms
Speedup	2.145ms	1.483ms	1.358ms

TABLE III: Efficiency results evaluating inference for scalar and vectorial versions.

Furthermore, we assessed the memory usage of the application using the Android Studio profiler, which provides insights into the quantity and distribution of memory consumption. Table IV presents the results obtained for both the smartwatch and smartphones, and these show that the memory usage in the smartwatch is 1.18% of the total capacity and for the smartphones is of 0.156%, which allows the devices to perform our task in background without disturbing the general usage of the hardware.

Inference	SG WATCH4	SG S20	S A32
Native (C)	130.6MB	22.7MB	22.3MB
Total memory	186.6MB	106.1MB	75.6MB

TABLE IV: Application memory usage both C code only and complete application.

In terms of energy use we have evaluated the mobile application by tracing the device during 10 minutes while using the application and when the device is not using the application to evaluate and compare the energy usage difference. In Table V, we show the average energy consumption of the devices when idle and when using the application. The results show that the energy consumption is minimal and negligible compared to other tasks [21].

Average energy usage	SG S20	S A32
Using our app (C)	780µAh	940µAh
Idle	774µAh	936µAh

TABLE V: Application energy usage on smartphones.

C. Dimensionality

In this set of experiments, we ran the application using different dimensions to demonstrate the impact on execution time when reducing the number of dimensions. While a reduction in dimensions may affect the accuracy of the algorithm, for this specific task utilizing only accelerometer data, we observed no degradation until reaching 512 dimensions. This experiment highlights that utilizing only 1024 dimensions can yield the same accuracy as using 10240 dimensions while significantly reducing the execution time. The maximal execution time per sample was 1.868 seconds for the SG WATCH4, and the minimal execution time was 0.204 seconds for the SG S20.

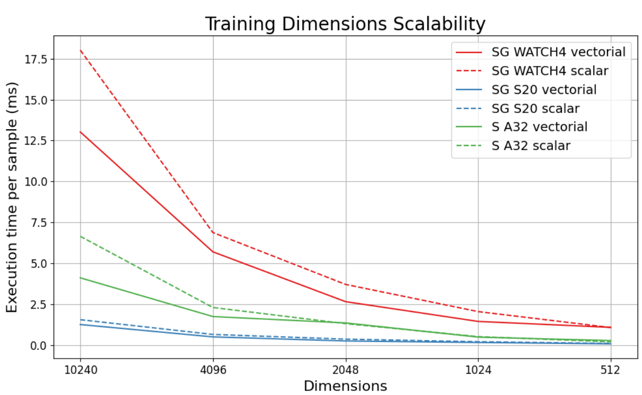


Fig. 4: Dimension scalability for training on all devices.

VI. Discussion

This research aimed to demonstrate the viability of detecting alcohol intoxication in individuals using smartphone or smartwatch data, with a focus on on-device computation and minimal user intrusiveness. In the initial experiment, we validated our code implementation and showcased the performance of transdermal alcohol detection through HDC, achieving a 93.5% accuracy. However, further exploration is essential given

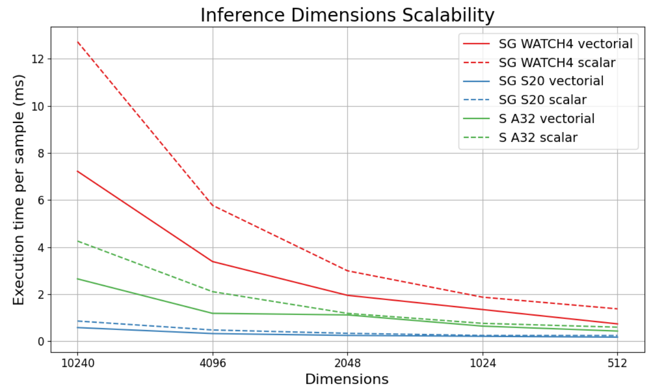


Fig. 5: Dimension scalability for inference on all devices.

the limited dataset with data for a small number of subjects and a short duration. To gain a more comprehensive understanding and develop a robust solution, creating a larger dataset with more subjects, extensive data collection, additional validation data (e.g., blood alcohol content, when the user is actively drinking), and incorporation of diverse data values (e.g., heart rate, gyroscope) is necessary. In the second experiment, we demonstrated the efficiency of our C code implementation for heavy computations, comparing the vectorial and scalar versions. The vectorial version exhibited a speedup ranging from 1.2x to 1.5x for training and 1.3x to 2.1x for inference. The execution time per sample on the smartwatch was 13.2ms for training and 6.818ms for inference, highlighting the fast execution time suitable for embedded devices. Memory usage evaluation indicated that the smartwatch's 16GB memory was more than sufficient, consuming less than 0.1% of the total space. In terms of energy usage (Table V), the application showed negligible energy consumption compared to the device's idle mode.

This research underscores the feasibility and suitability of HDC for alcohol detection, showcasing high accuracy, time and energy efficiency, and minimal memory usage. The model's ability for online learning and on-device computation further emphasizes its appropriateness for the task at hand. However, future work should involve model verification and improvement by incorporating additional features such as heart rate, gyroscope data, and location data. Creating a larger dataset with data from both smartwatches and smartphones, encompassing a broader participant pool, is essential for validating and enhancing this work.

VII. Conclusion

Our team has developed a smartphone application that works with a smartwatch to detect alcohol intoxication in users. The application is designed to be minimally invasive and provides users with reminders of their current state. By using hyperdimensional computing, the application can predict whether a user is intoxicated or not. This approach enables the application to learn online, using only the device's computational power for both training and testing.

Our study shows that hyperdimensional computing is highly effective and achieves impressive accuracy results with minimal energy consumption. Our approach efficiently handles both training and inference tasks on embedded devices, thanks to the hyperdimensional computing nature and our loop optimizations and vectorial programming. To enhance our work further, it's essential to expand the dataset to include data from smartwatch and smartphone sensors such as accelerometers, gyroscopes, and heart rate monitors.

References

[1] A. H. M. et al. *Global burden of diseases, injuries, and risk factors for young people's health during 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013*. In: The Lancet 387.10036 (2016), pp. 2383–2401.

[2] R. Andréasson and A. Jones. *Erik M.P. Widmark (1889–1945): Swedish pioneer in forensic alcohol toxicology*. In: Forensic Science International 72.1 (1995), pp. 1–14.

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

[4] 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.

[5] H. Chen et al. *DARL: Distributed Reconfigurable Accelerator for Hyperdimensional Reinforcement Learning*. In: Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design. 2022.

[6] M. Czaplik et al. *Evaluation of a new non-invasive measurement technique based on bioimpedance spectroscopy to estimate blood alcohol content: A pilot study*. In: Biomedizinische Technik 64 (3 2019), pp. 365–371.

[7] R. E. Davis-Martin, S. M. Alessi, and E. D. Boudreaux. *Alcohol Use Disorder in the Age of Technology: A Review of Wearable Biosensors in Alcohol Use Disorder Treatment*. In: Frontiers in Psychiatry 12 (2021).

[8] C. M. Gardner. *Transmission versus reflectance spectroscopy for quantitation*. In: Journal of Biomedical Optics 23.1 (2018), p. 018001.

[9] R. W. Gayler. *Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience*. In: Joint International Conference on Cognitive Science (ICCS/ASCS). 2003, pp. 133–138.

[10] T. van Gelder. *The dynamical hypothesis in cognitive science*. In: Behavioral and Brain Sciences 21.5 (1998), 615–628.

[11] M. e. a. Griswold. *Alcohol use and burden for 195 countries and territories, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016*. In: LANCET 392 (2018).

[12] M. Heddes et al. *Torchhd: An Open Source Python Library to Support Research on Hyperdimensional Computing and Vector Symbolic Architectures*. In: Journal of Machine Learning Research 24.255 (2023), pp. 1–10.

[13] A. Hernández-Cano et al. *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.

[14] A. Hernández-Cano et al. *RegHD: Robust and Efficient Regression in Hyper-Dimensional Learning System*. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). 2021, pp. 7–12.

[15] P. Kanerva. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. In: Cognitive Computation 1 (2009), pp. 139–159.

[16] J. Killian et al. *Learning to Detect Heavy Drinking Episodes Using Smartphone Accelerometer Data*. In: KHD@IJCAI. 2019.

[17] 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.

[18] P. Koskinen, J. Virolainen, and M. Kupari. *Acute Alcohol Intake Decreases Short-Term Heart Rate Variability in Healthy Subjects*. In: Clinical Science 87.2 (1994), pp. 225–230.

[19] J. Kubicek et al. *Prediction Model of Alcohol Intoxication from Facial Temperature Dynamics Based on K-Means Clustering Driven by Evolutionary Computing*. In: Symmetry 11.8 (2019).

[20] W. S. Lima et al. *Human Activity Recognition Using Inertial Sensors in a Smartphone: An Overview*. In: Sensors (Basel, Switzerland) 19 (2019).

[21] X. Liu et al. *Characterizing Smartwatch Usage in the Wild*. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. 2017, 385–398.

[22] L. D. Mihretu et al. *Determination of ethanol in blood using headspace gas chromatography with flameionization detector (HS-GC-FID): Validation of a method*. In: Cogent Chemistry 6.1 (2020), p. 1760187.

[23] A. Moin et al. *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.

[24] S. Paprocki, M. Qassem, and P. A. Kyriacou. *Review of Ethanol Intoxication Sensing Technologies and Techniques*. In: Sensors 22 (18 2022).

[25] S. Pouyanfar et al. *A Survey on Deep Learning: Algorithms, Techniques, and Applications*. In: ACM Comput. Surv. 51.5 (2018).

[26] L. Rachakonda et al. *Smart-Steering: An IoMT-Device to Monitor Blood Alcohol Concentration using Physiological Signals*. In: 2020, pp. 1–6.

[27] A. Rahimi and B. Recht. *Random Features for Large-Scale Kernel Machines*. In: Neural Information Processing Systems. 2007.

[28] M. Reichert et al. *Studying the impact of built environments on human mental health in everyday life: methodological developments, state-of-the-art and technological frontiers*. en. In: Curr. Opin. Psychol. 32 (2020), pp. 158–164.

[29] M. E. Segura et al. *Enhanced Detection of Transdermal Alcohol Levels Using Hyperdimensional Computing on Embedded Devices*. 2024.

[30] M. Shoaib et al. *Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors*. In: Sensors (Basel, Switzerland) 16 (2016).

[31] M. Shoaib et al. *Fusion of Smartphone Motion Sensors for Physical Activity Recognition*. In: Sensors (Basel, Switzerland) 14 (2014), pp. 10146 –10176.

[32] C. Steele and R. Josephs. *Alcohol Myopia. Its Prized and Dangerous Effects*. In: The American psychologist 45 (1990), pp. 921–33.

[33] A. Thomas, S. Dasgupta, and T. Rosing. *A Theoretical Perspective on Hyperdimensional Computing*. In: Journal of Artificial Intelligence Research 72 (2021), pp. 215–249.

[34] P. Verges, T. Givargis, and A. Nicolau. *RefineHD: Accurate and Efficient Single-Pass Adaptive Learning Using Hyperdimensional Computing*. In: 2023 IEEE International Conference on Rebooting Computing (ICRC). 2023, pp. 1–8.

[35] P. Verges et al. *Molecular Classification Using Hyperdimensional Graph Classification*. 2024.

[36] P. Vergés et al. *Accelerating Permute and N-Gram Operations for Hyperdimensional Learning in Embedded Systems*. In: 2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). 2023, pp. 253–260.

[37] P. Vergés et al. *Classification using Hyperdimensional Computing: A Review with Comparative Analysis*. In.

[38] P. Vergés et al. *HDCC: A Hyperdimensional Computing compiler for classification on embedded systems and high-performance computing*.

[39] W. F. Wang, C. Y. Yang, and Y. F. Wu. *SVM-based classification method to identify alcohol consumption using ECG and PPG monitoring*. In: Personal and Ubiquitous Computing 22 (2 2018), pp. 275–287.

[40] B. M. Wright, T. P. Jones, and A. W. Jones. *Breath Alcohol Analysis and the Blood: Breath Ratio*. In: Medicine, Science and the Law 15.3 (1975). PMID: 1160565, pp. 205–210.