2025 Design, Automation & Test in Europe Conference (DATE 2025)

Exploiting Boosting in Hyperdimensional Computing for Enhanced Reliability in Healthcare

SungHeon Jeong[1], Hamza Errahmouni Barkam[1], Sanggeon Yun[1],
Yeseong Kim[2], Shaahin Angizi[3] and Mohsen Imani[1]*
[1]University of California Irvine, [2]DGIST, [3]New Jersey Institute of Technology
*Corresponding Author: m.imani@uci.edu

***Abstract*—Hyperdimensional computing (HDC) enables efficient data encoding and processing in high-dimensional spaces, benefiting machine learning and data analysis. However, underutilization of these spaces can lead to overfitting and reduced model reliability, especially in data-limited systems—a critical issue in sectors like healthcare that demand robustness and consistent performance. We introduce BoostHD, an approach that applies boosting algorithms to partition the hyperdimensional space into subspaces, creating an ensemble of weak learners. By integrating boosting with HDC, BoostHD enhances performance and reliability beyond existing HDC methods. Our analysis highlights the importance of efficient utilization of hyperdimensional spaces for improved model performance. Experiments on healthcare datasets show that BoostHD outperforms state-of-the-art methods. On the WESAD dataset, it achieved an accuracy of $98.37\% \pm 0.32\%$, surpassing Random Forest, XGBoost, and OnlineHD. BoostHD also demonstrated superior inference efficiency and stability, maintaining high accuracy under data imbalance and noise. In person-specific evaluations, it achieved an average accuracy of 96.19%, outperforming other models. By addressing the limitations of both boosting and HDC, BoostHD expands the applicability of HDC in critical domains where reliability and precision are paramount.**

I. Introduction

Hyperdimensional computing (HDC) is a rapidly advancing field within artificial intelligence, offering numerous valuable qualities that apply to various areas [1]–[7]. HDC's ability to efficiently encode and process data in high-dimensional spaces has generated substantial interest, particularly in machine learning and data analysis [8]–[17]. As we contemplate the practical use of HDC-based models in real-world applications, we encounter crucial requirements beyond mere accuracy. These requirements encompass robustness, reliability, consistency, and resource efficiency [18]. Despite these requirements, analyses of HDC in high-dimensional spaces has not yet been fully explored which can lead to overfitting by selecting very high dimension above only performance [19]. Our research shows that the partitioning of high-dimensional spaces directly correlates with the utility of the space, fulfilling those requirements under certain condition.

Meeting the requirements of HDC is anticipated to significantly expand its applicability across various domains, particularly in fields where reliability and precision are paramount, such as healthcare [20], [21]. In addressing healthcare datasets, our evaluation focuses on critical issues such as robustness to noise, consistency in performance, and stability that are non-negotiable for the successful application of HDC in healthcare.

Ensemble methods have been recognized for their ability to prevent overfitting and achieve enhanced performance and stability in predictive modeling [22]. Among these methods, boosting stands out for its capability to aggregate the predictions of multiple weak learners into a robust and accurate ensemble model. The strengths of boosting encompass significant accuracy improvements, resistance to overfitting, versatility across a wide range of machine learning tasks. However, boosting‘s ability across sensitivity to noisy data, performance and stability is dependent on the ability of weak learner. The computational demands of training boosting ensembles can also present challenges in scenarios requiring real-time responses or operating under resource constraints. Furthermore, the overall performance of the ensemble model may be compromised if the performance of the weak learners is not assured, potentially leading to biases towards challenging examples in imbalanced datasets.

HDC offers promising attributes to effectively address the limitations associated with traditional boosting techniques, as highlighted in prior studies [23], [24]. However, a simplistic parallel ensemble of HDC models may inadvertently escalate the computational costs associated with training and may not guarantee robustness against noise for each weak learner. In our approach, we leverage the OnlineHD model [18] as a foundation, proposing a novel partitioning strategy where the model's hyperdimensional space (D) is divided among n weak learners, with each receiving a D/n dimensional segment. This segmentation approach prompts us to designate these segmented models as weak learners. Subsequently, we analyze the utilization efficiency of the hyperdimensional space by each weak learner Figure 5 and theoretically demonstrate the inherent limitations posed by high-dimensional spaces Equations 3, 2. Under conditions where the performance of weak learners is assured, BoostHD elevates the capabilities of OnlineHD, ensuring stability, and providing robustness against overfitting and noise.

II. Related Work

A. Boosting methods on Machine Learning

To overcome the limitations of individual predictive models, ensemble methods such as boosting have been developed to improve performance by combining multiple weak learners, such as shallow decision trees or simple neural networks. Boosting aggregates the predictions of these learners, trained

978-3-9826741-0-0/DATE25/© 2025 EDAA

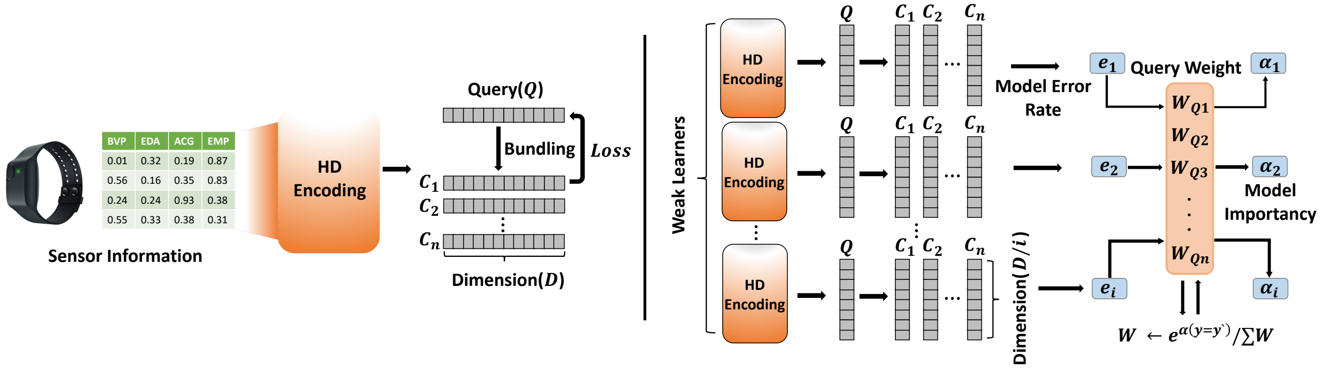

Fig. 1. Illustration of the BoostHD framework applied to hyperdimensional computing (HDC). Sensor information is encoded into a high-dimensional vector space (Dimension D). The query vector Q is bundled with multiple contextual vectors $C_1, C_2, \ldots, C_n$, forming weak learners with segments of the high-dimensional space. Each weak learner receives a partitioned subspace (D/n) of the original hyperdimensional space, optimizing the use of the entire dimensional space and minimizing overfitting. Query weights $W_{Q1}, W_{Q2}, \ldots, W_{Qn}$ and model importances are dynamically adjusted based on model error rates, with a boosting approach used to aggregate and adjust the ensemble performance, ensuring robustness and stability, particularly in noise-sensitive domains such as healthcare.

sequentially to correct each other's errors. A notable example is Adaptive Boosting (AdaBoost) [22], which assigns equal initial weights to data points, adjusts weights to focus on misclassified points, and uses confidence-weighted voting for final predictions. Gradient Boosting Machine (GBM) [25], in contrast, minimizes residual errors through gradient descent instead of reweighting data points. Extensions like LightGBM [26] and XGBoost [27] adopt this residual minimization strategy, offering improved performance over AdaBoost.

B. Healthcare with wearable device

The WESAD dataset [28] is a benchmark for wearable stress and affect detection, comprising physiological and motion data from wrist- and chest-worn devices collected from 15 subjects in a controlled lab setting. It includes diverse sensor modalities such as blood volume pulse, electrocardiogram, electrodermal activity, respiration, temperature, and three-axis acceleration, alongside three affective states—neutral, stress, and amusement. While convolutional neural networks (CNNs) dominate as the state-of-the-art solution [29], Hyperdimensional Computing (HDC) offers a resource-efficient alternative better suited for the constraints of wearable devices. HDC's lightweight, online learning approach effectively handles WESAD's multimodal data, making it a promising method for real-world applications in stress and affect detection. The combination of WESAD's rich modalities and HDC's computational efficiency underscores the dataset's potential as a testing ground for advanced wearable technologies.

C. Hyperdimensional Classification

HDC is inspired by neural processes in the brain, encoding data into a high-dimensional space. It learns universal patterns for each label by encoding data points as hypervectors $\mathbf{H}$ through matrix multiplication with Gaussian distribution values and trigonometric activation functions (**Binding**), such as sine and cosine. These hypervectors are combined (**Bundling**) to create class hypervectors $\mathbf{C}_l$, which represent each label in R^D space. During inference, a query hypervector $\mathbf{H}$ is encoded and compared to class hypervectors using a similarity function $\delta(\mathbf{H}, \mathbf{C}_l)$. This process is widely applicable, as demonstrated in [30]–[37].

Model updates now leverage similarity assessments to iteratively refine predictions [18], enhancing the model's ability to capture patterns and improve accuracy:

$$\delta(V_1, V_2) = \frac{V_1^{\dagger} V_2}{\|V_1\| \|V_2\|} \tag{1}$$

Key operations in HDC include: leftmargin=*

- **Bundling**: Combines hypervectors into a single vector ($\vec{R} = \vec{V_1} + \vec{V_2}$), enabling memorization in high-dimensional space.
- **Binding**: Associates orthogonal hypervectors ($\vec{R} = \vec{V_1} * \vec{V_2}$), creating a new hypervector orthogonal to inputs ($\delta(\vec{R}, \vec{V_1}) \simeq 0$).

III. Boosting Hyperdimensional Computing

In this section, we introduce the BoostHD framework, integrating boosting techniques into HDC with a focus on dimensionality (D). Instead of relying solely on a single robust learner with high D, this approach breaks down the dimension into numerous sub-dimensions, each represented by a weak learner. These weak learners are trained sequentially, with each one adaptively learning from and correcting the errors of its predecessor. This methodology effectively addresses the limitations of the strong learner, thereby enhancing its performance ceiling (as outlined in Algorithm 1). A notable feature of this method is its sequential training process, while parallelization becomes feasible during the inference phase.

Algorithm 1 Pseudo code for BoostHD

```
Input:
    $X$: Set of data points
    $y$: Labels
    $x$: Test data point
Parameters:
    $d$: Dimensionality
    $n$: Number of learners
    $W_s$: Sample weight
Output:
    $f_{\theta_i}$: Trained learners parameterized by $\theta$
    $\alpha_i$: Weight of learner
    $\hat{y}$: Prediction

procedure TRAINING($X$, $y$)
    Initialize learners $f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_n}$
    for $i \in \{1, 2, \ldots, n\}$ do
        Train $f_{\theta_i}$ with $X$ and $y$
        $\hat{y}_i = f_{\theta_i}(X)$
        $e_{\theta_i} \leftarrow$ Error rate of $f_{\theta_i}$
        $\alpha_i = W_s \cdot e_{\theta_i}$
        $W_s \leftarrow e^{\alpha_i \cdot (y \neq \hat{y})} / \sum W_s$
    end for
end procedure
procedure INFERENCE($x$)
    $\hat{ys} = f_\theta(x)$
    $\hat{y} = \text{argmax}\left(\sum^n \hat{ys} \cdot \alpha\right)$
end procedure
```

Performance, assessed based on the parameters D and the number of weak learners (N_L), shows a direct relationship, ensuring stable and improved performance with substantial values for both D and N_L, as depicted in Figure 3. However, it's crucial to note that elevated values of D and N_L introduce increased computational costs, establishing an inherent trade-off between computational cost and performance. To maintain the effectiveness of weak learners, preserving a baseline dimensionality is imperative.Failure to do so, as exemplified in the case where $N_L = 100$ and $D_{total} = 1$K, can lead to a substantial degradation in performance, as demonstrated in Figure 3(b).

$$\mu_\lambda = \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} f(\lambda)\lambda \, d\lambda \sim \frac{1}{3\pi q}(\lambda_{\text{max}} - \lambda_{\text{min}})^{3/2} \tag{2}$$

$$\sigma_\lambda^2 \sim \frac{1}{2\pi\sigma^2 q}\left(\frac{1}{2}(\lambda_{\text{max}}^2 - \lambda_{\text{min}}^2) - 2\mu(\lambda_{\text{max}} - \lambda_{\text{min}}) + \mu^2(\ln|\lambda_{\text{max}}| - \ln|\lambda_{\text{min}}|)\right) \tag{3}$$

In the context of HDC, partitioning dimensions can be viewed as a transformation in the geometric shape of the kernel (k). HDC frameworks often employ the Gaussian Kernel as their foundational element, represented as $k_{i,j} \sim \mathcal{N}(0, 1)$. When this kernel transforms a hyperdimensional space, the geometric characteristics of k can be described by leveraging the theory of the Marchenko-Pastur distribution of random matrices ([38]). Equations 2 and 3 provide insights into this process. Here, q is defined as the ratio of the number of columns (N_c) to the number of rows (N_r), σ represents the standard deviation of the Gaussian distribution (i.e., 1), λ denotes singular values, and λ_{max} and λ_{min} signify the upper and lower bounds of λ. The mean and variance of λ are expressed in Equations 2 and 3, respectively. Considering that N_c is determined by the dataset, and N_r is a parameter represented by D, under the assumption of a fixed input shape, it becomes evident that q exhibits an inverse relationship with D, while μ_λ demonstrates a direct proportionality with D. Conversely, Equation 3 formulates σ_λ^2 as a function of three distinct terms: T_1 (Equation 4), T_2 (Equation 5), and T_3 (Equation 6). Each term converges to a specific value and experiences minimal fluctuations after that. Consequently, μ_λ increases directly in proportion to D, while σ_λ^2 remains constant as D increases.

This proportionality implies that irrespective of the values of μ_λ and σ_λ^2, λ maintains a consistent absolute interval value due to the stability of σ_λ^2 (Equation 7). As D increases, the values of λ escalate, yet the interval remains steady. This phenomenon results in the ratio of the minor axis (A_S) to the major axis (A_L) asymptotically approaching unity (i.e., 1), leading to a circular shape. This observation underscores the pivotal role of D in shaping the kernel, wherein an elliptical kernel is transformed into a broadly distributed circular shape beyond the constrained space formed by input biases, as illustrated in Figure 4.

$$\lim_{q\to\infty} \frac{1}{q}(\lambda_{\text{max}}^2 - \lambda_{\text{min}}^2) = \lim_{q\to\infty}\left((1+\sqrt{q})^4 - (q-\sqrt{q})^4\right) = k \tag{4}$$

$$\lim_{q\to\infty} \frac{1}{q}\left(-2\mu(\lambda_{\text{max}} - \lambda_{\text{min}})\right) = 0 \tag{5}$$

$$\lim_{q\to\infty} \frac{1}{q}\left(\mu^2(\ln|\lambda_{\text{max}}| - ln|\lambda_{\text{min}}|)\right) = 0 \tag{6}$$

$$\lim_{x\to\infty} \sigma_\lambda^2 = k \tag{7}$$

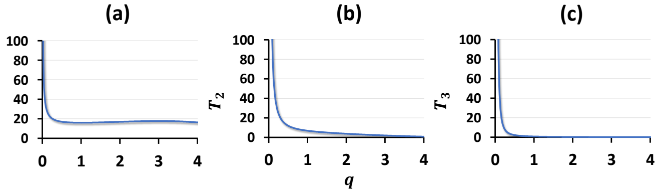

Fig. 2. Extreme distribution of terms, Eq. 4, 5, 6 in σ_λ^2

Within the framework of HDC, the theoretical definition of utilizing the subspace formed by classifiers can be expressed as rank$(K)/D$, where rank(K) represents the rank of the matrix formed by the classifier, and D signifies the dimensionality of the space. However, in practical applications, the effective span of the subspace experiences attenuation due to various factors denoted by Π. These factors are the product sums of cosine similarity values between class hypervectors, which

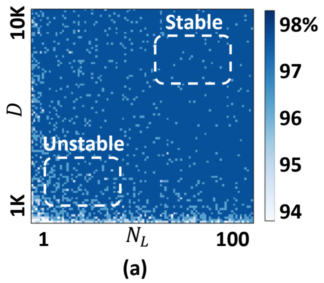

Fig. 3. Accuracy heatmap based on N_L and their respective D. In (a) and (b), N_L takes values from 1 to 100 and 10 to 100 with each step 1, 10. For (a), the accuracy is presented for each specified dimension. For (b), the total dimension(D_{total}) is divided among the N_L, where each learner possesses a dimension size of D_{total}/N_L.

embody constraints or characteristics inherent to the system. Consequently, the actual Span Utilization (SP) is defined as the quotient of rank$(K)/D$ divided by the product of $\pi_1, \pi_2, ..., \pi_n$. This representation encapsulates the reduced space that is practically attainable, considering the influence of the aforementioned factors. In the realm of HDC, where cosine similarity serves as the metric of choice, maximizing the utilization of this subspace directly impacts the system's performance. With a focus on optimizing SP (as depicted in Figure 5), the BoostHD approach is superior to traditional HDC methodologies. By augmenting the subspace's utilization, BoostHD optimizes computational resources and significantly enhances the accuracy and efficiency of high-dimensional data processing. This observation underscores the practical relevance of subspaces in HDC and highlights the pivotal role played by BoostHD in maximizing their utility.

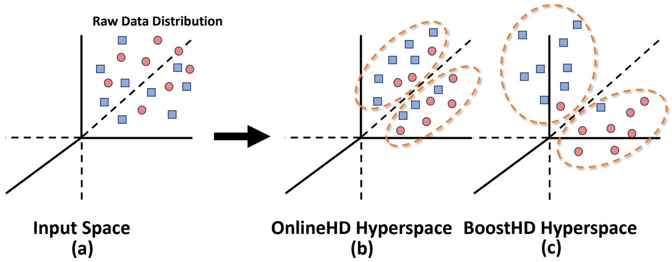

Fig. 4. In the process of kernel transformation. Data is mapped into a hyperdimensional space. (a) illustrates the distribution of the raw data has a biased distribution. (b) represents a scenario where $N_c = 4000$, while (c) corresponds to $N_c = 400$. From the perspective of span utilization, the mapping illustrated in (c) demonstrates superior efficiency compared to that in (b).

IV. Evaluation

In our experimental setup, we used a GeForce RTX 4070 GPU and a 12th generation Intel(R) Core(TM) i7-12700K CPU. Each experiment was conducted over 10 runs, employing datasets including WESAD [28], the Nurse Stress Dataset [39], and the Stress-Predict Dataset [40] to evaluate accuracy and time efficiency. For other aspects of the experiments, only the WESAD dataset was used. The WESAD dataset, a benchmark for stress detection research, features multimodal data from wearable devices like Empatica E4 and RespiBAN collected from 16 subjects. It includes physiological and motion indicators such as EDA, ECG, EMG, and BVP. The Nurse Stress Dataset [39] and Stress-Predict Dataset [40], collected via the E4 sensor, also include EDA, ECG, EMG, and BVP data from 37 and 15 subjects, respectively. Stress level classification was reduced to three labels: good, common, and stress. Test data was organized by subject units, and all results were based on the test dataset.

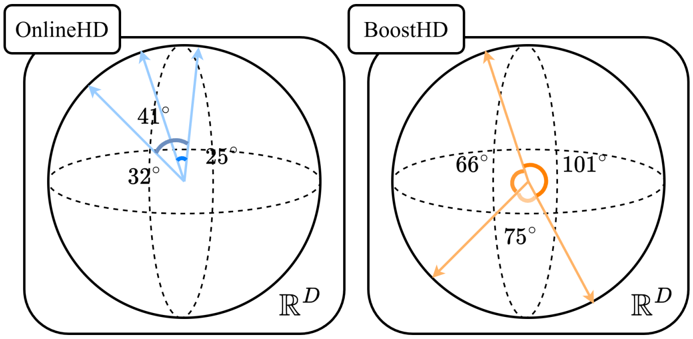

Fig. 5. Example of BoostHD and OnlineHD SP. The orange one is BoostHD's class hypervectors, and the blue one is OnlineHD's class hypervector. BoostHD uses much more space at hyperdimensional space by composing high cosine similarity across class hypervectors.

Datasets were preprocessed using a moving average filter with a window size of 30, extracting statistical features such as minimum, maximum, mean, and standard deviation. To address varying ranges, normalization was applied to ensure consistent scaling. The experimental models included AdaBoost (learning rate = 1.0, 10 estimators), Random Forest (bootstrap enabled, 10 estimators), XGBoost (10 estimators), SVM (linear kernel), and a DNN comprising convolutional layers with a learning rate of 0.001, four linear layers [2048, 1024, 512, classes], ReLU activation, and dropout. Additionally, OnlineHD was configured with dimensional adjustment, bootstrap enabled, a learning rate of 0.035, and Gaussian distribution $N(0, 1)$. For ensemble models, $N_L = 10$. The HDC model was evaluated with D_{total} values ranging from 10 to 10,000, while for the BoostHD model, the weak learner dimensionality (D_{wl}) was set to D_{total}/N_L.

A. Performance

In our comprehensive experimentation, we subjected the BoostHD algorithm to rigorous evaluation across three distinct datasets: WESAD [28], Nurse Stress Dataset [39], and Stress-Predict Dataset [40]. The focal points of our assessment encompassed key metrics such as accuracy, inference time, and training time. Our findings unveiled the remarkable prowess of BoostHD, consistently attaining the highest accuracy levels across all three datasets, as presented in Table I. Notably, when compared to OnlineHD on the WESAD [28] dataset, the accuracy achieved by BoostHD remained outside the range of two standard deviations. This distinction was further underscored by OnlineHD's inability to reach a 98% accuracy

TABLE I
ACCURACY (%) PERFORMANCE: COMPARES BOOSTHD‘S ACCURACY WITH VARIOUS BASELINES.

Dataset	Adaboost	RF	XGBoost	SVM	DNN	OnlineHD	BoostHD
WESAD	93.13 ± 0.01	97.13 ± 0.06	96.88 ± 0.01	93.12 ± 0.01	96.75 ± 1.05	96.37 ± 0.40	**98.37** ± 0.32
Nurse Stress Dataset	55.28 ±0.01	59.35 ±0.78	61.01 ±0.01	60.99 ±0.01	60.04 ±0.06	61.37 ±0.10	**61.52** ±0.07
Stress-Predict Dataset	67.54 ± 0.01	67.76 ± 0.12	65.76 ± 0.01	67.30 ± 0.01	67.29 ± 0.07	65.79 ± 0.17	**68.10** ± 0.09

TABLE II
INFERENCE EFFICIENCY: COMPARES BOOSTHD‘S INFERENCE TIME (10^{-5} SECONDS) WITH VARIOUS BASELINES

Dataset	Adaboost	RF	XGBoost	SVM	DNN	OnlineHD	BoostHD
WESAD	46.3	38.5	47.6	108.3	37.0	**7.57**	11.0
Nurse Stress Dataset	145.2	179.7	131.1	188.4	38.6	14.5	**12.1**
Stress-Predict Dataset	63.4	91.5	58.7	265	43.7	13.2	**12.0**

threshold in any of the ten independent trials, while BoostHD consistently surpassed this benchmark.

BoostHD strategically combines in-memory learning, exemplified by HDC, with adaptive learning mechanisms within an ensemble framework, resulting in a significantly accelerated learning process, even when training is serialized. Empirical evaluations in a GPU environment demonstrated BoostHD's clear superiority over DNNs, achieving substantially faster processing speeds across the three benchmark datasets. In a CPU environment, BoostHD achieved remarkable computational efficiency, processing data 26 times faster than DNN counterparts. Additionally, optimization efforts tailored BoostHD for parallel processing, greatly improving inference efficiency Table II. These enhancements were particularly effective when processing the relatively large input vectors in the Nurse Stress Dataset [39] and Stress-Predict Dataset [40], resulting in a significant reduction in inference time. This firmly establishes BoostHD as the most time-efficient model across all evaluated datasets, particularly for the Nurse Stress Dataset [39] and Stress-Predict Dataset [40].

B. Stability

BoostHD demonstrates strong and consistent performance and exhibits stable convergence even in constrained environments. As illustrated in Figure 3, the accuracy of the weak learner experiences a steep decline when the minimum dimensionality requirement is unmet. By ensuring this minimum dimensionality and progressively increasing the value of D, we observe a clear separation between the two models within a given one standard deviation (σ), as shown in Figure 6(a). Specifically, the value of μ_σ for BoostHD stands at 0.0046, while for OnlineHD, it registers at 0.0127, signifying a roughly threefold disparity. This lower σ value for BoostHD underscores its superior stability. In the context of BoostHD, it's notable that σ scales proportionally with $1/N_L$ and inversely with D provided that the baseline condition is maintained (as depicted in Figure 3(a), upper right). Moreover, when the value of N_L surpasses a specific threshold (k), such as 50 as illustrated in Figure 3(a) (i.e., $N_L > k$), the σ value remains consistently preserved.

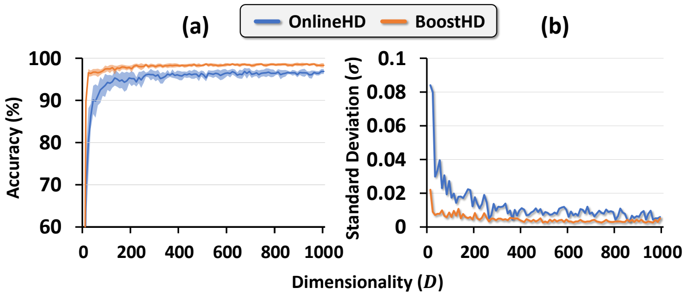

Fig. 6. Analysis of the impact of D on the stability of BoostHD and OnlineHD. (a), Accuracy of BoostHD and OnlineHD as a function of D, with error bars representing σ. (b), Variation of the σ in (a) with respect to D.

C. Overfitting

BoostHD presents a noteworthy advantage over traditional HDC methods in its resilience to overfitting. To empirically assess this characteristic, we intentionally induce overfitting by crafting an imbalanced dataset ($\mathbf{D}$) through the generation of data subsets. The extent of overfitting, denoted as r, is quantified by creating subset data for all classes except the target class (C_{target}). This process yields the final dataset $\mathbf{D}$ as described in Equation 8. We employ Macro accuracy as a more suitable metric for imbalanced datasets to ensure a fair performance evaluation that the varying sample counts per class do not skew. As the value of r increases, OnlineHD experiences a noticeable decline in performance. In stark contrast, BoostHD consistently maintains its performance levels, demonstrating a robust ability to preserve stability, as depicted in Figure 7.

$$\mathbf{D} = \begin{cases} \text{x}, & \text{if } y = C_{\text{target}} \\ \text{x} \times r, & \text{if } y \neq C_{\text{target}} \end{cases} \tag{8}$$

D. Robustness

We assess the robustness of models against bitflip noise to explore in wearable device originating from hardware components according to its probability of bit flip error p_b. In Figure 8, our experiments spanned two distinct ranges of p_b, given the substantial impact of this parameter on

TABLE III
PERSON SPECIFIC PERFORMANCE IN ACCURACY (%)

	Left hands	Female	Age $\leq$ 25	Age $\geq$ 30	Height $\leq$ 170	Height $\geq$ 185	AVERAGE
Adaboost	95.63	96.86	96.05	89.24	90.74	91.27	93.30
RF	97.15	97.7	98.03	88.29	95.06	94.92	95.19
XGBoost	98.73	98.12	**98.68**	91.41	95.06	94.24	96.04
SVM	97.47	95.4	94.74	92.41	90.12	**95.24**	94.23
DNN	98.42	95.4	94.41	88.92	92.59	93.57	93.89
OnlineHD	98.73	97.8	94.08	92.41	92.59	94.6	95.04
BoostHD	**99.05**	**98.33**	96.38	**93.35**	**96.3**	93.73	**96.19**

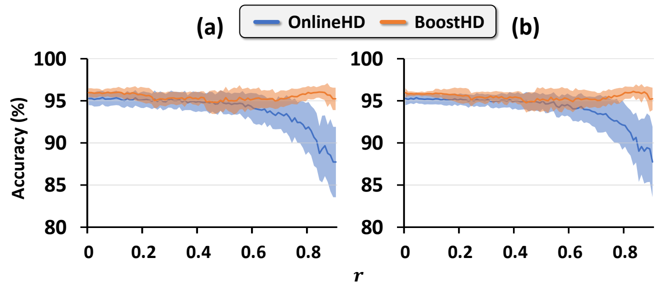

Fig. 7. Analysis of the impact of r on overfitting BoostHD and OnlineHD. (a), (b) are the error variance in the overfitting with D_{total} = 1000 and N_L = 10. (b), D_{total} = 4000 and N_L = 10.

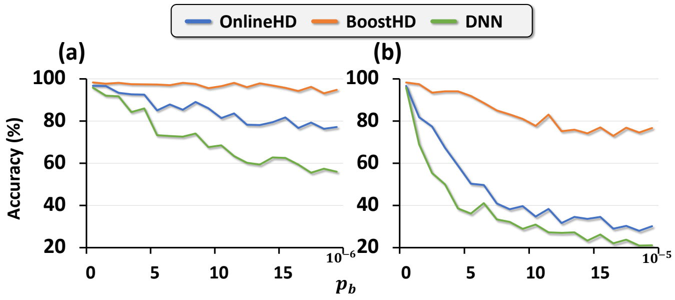

Fig. 8. Robustness Analysis of BoostHD at different levels of p_b. At point (a), BoostHD experiences a mere 3% decrement in performance, which is 6 and 13 times less pronounced than OnlineHD and DNN. Furthermore, at point (b), BoostHD manifests a 21% reduction in performance, a figure that is 3 and 3.5 times less than the corresponding performance drops in OnlineHD and DNN.

performance. Our experimental scope was deliberately limited to a narrow range of p_b. Despite this inherent constraint and our meticulous approach of conducting 100 independent trials to ensure statistical validity, we observed occasional peaks in the graph. Nevertheless, a discernible trend as a function of p_b persisted. Within the range where $p_b = 10^{-5}$, as illustrated in Figure 8(a), BoostHD incurred an accuracy loss of no more than 5.7%. This represents approximately one-fourth of the loss observed in OnlineHD and roughly one-seventh of that observed in DNN. To statistically validate the accuracies concerning varying p_b, we employed the Median Absolute Deviation (MAD) as a measure of robustness (defined as MAD $=$ median($|X_i-$median$(X)|$)). In Figure 8(a), the MAD for BoostHD was quantified at 0.024, which is six times lower than that of OnlineHD (0.1454) and four times lower than that of DNN (0.083). In Figure 8(b), BoostHD exhibited a MAD of 0.005, which is three times lower than OnlineHD (0.015) and a substantial 30 times lower than DNN (0.1509). These results are compelling evidence of BoostHD's superior robustness compared to the other methods.

E. Bias and reliability for person-specific groups

We carefully segmented subjects based on various attributes in WESAD [28], including hand preference, gender, age, and height. This stratification resulted in subsets with specific subject characteristics, such as left-hand preference, female gender, age groups, and height categories. Our primary objective was to assess how well models performed across individuals with diverse characteristics, a crucial consideration for healthcare applications to ensure fairness and accuracy. Notably, our BoostHD model consistently outperformed other models in all categories III, except for two cases where it ranked second. This underscores the potential of combining boosting methods and HDC, especially in healthcare, where equitable performance is essential.

V. Conclusion

We introduced BoostHD, a novel method that integrates hyperdimensional computing with boosting to create a robust ensemble model. BoostHD effectively overcomes limitations of traditional HDC by efficiently utilizing high-dimensional spaces and preventing overfitting. Our experiments on healthcare datasets demonstrated that BoostHD consistently outperforms existing methods in accuracy, efficiency, stability, and robustness. These results confirm BoostHD's potential in critical domains where reliability and precision are essential.

Acknowledgements

This work was supported in part by the DARPA Young Faculty Award, the National Science Foundation (NSF) under Grants #2127780, #2319198, #2321840, #2312517, and #2235472, the Semiconductor Research Corporation (SRC), the Office of Naval Research through the Young Investigator Program Award, and Grants #N00014-21-1-2225 and #N00014-22-1-2067, Army Research Office Grant #W911NF2410360. Additionally, support was provided by the Air Force Office of Scientific Research under Award #FA9550-22-1-0253, along with generous gifts from Xilinx and Cisco.

References

[1] Y. Ni, Z. Zou, W. Huang, H. Chen, W. Y. Chung, S. Cho, R. Krishnan, P. Mercati, and M. Imani, “Heal: Brain-inspired hyperdimensional efficient active learning,” *arXiv preprint arXiv:2402.11223*, 2024.

[2] M. Imani, Y. Kim, S. Riazi, J. Messerly, P. Liu, F. Koushanfar, and T. Rosing, “A framework for collaborative learning in secure high-dimensional space,” in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. IEEE, 2019, pp. 435–446.

[3] Z. Zou, H. Chen, P. Poduval, Y. Kim, M. Imani, E. Sadredini, R. Cammarota, and M. Imani, “Biohd: an efficient genome sequence search platform using hyperdimensional memorization,” in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 656–669.

[4] Y. Ni, M. Issa, D. Abraham, M. Imani, X. Yin, and M. Imani, “Hdpg: Hyperdimensional policy-based reinforcement learning for continuous control,” in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 1141–1146.

[5] Y. Ni, Y. Kim, T. Rosing, and M. Imani, “Algorithm-hardware co-design for efficient brain-inspired hyperdimensional learning on edge. in 2022 design, automation & test in europe conference & exhibition (date),” *IEEE, 292ś297*, 2022.

[6] H. Chen, Y. Ni, A. Zakeri, Z. Zou, S. Yun, F. Wen, B. Khaleghi, N. Srinivasa, H. Latapie, and M. Imani, “Hdreason: Algorithm-hardware codesign for hyperdimensional knowledge graph reasoning,” *arXiv preprint arXiv:2403.05763*, 2024.

[7] H. Chen, Y. Ni, W. Huang, and M. Imani, “Scalable and interpretable brain-inspired hyper-dimensional computing intelligence with hardware-software co-design,” in *2024 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2024, pp. 1–8.

[8] D. Kleyko, D. Rachkovskij, E. Osipov, and A. Rahimi, “A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges,” *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–52, 2023.

[9] S. Yun, H. E. Barkam, P. R. Genssler, H. Latapie, H. Amrouch, and M. Imani, “Hyperdimensional computing for robust and efficient unsupervised learning,” in *2023 57th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2023, pp. 281–288.

[10] S. Yun, R. Masukawa, S. Jeong, and M. Imani, “Spatial-aware image retrieval: A hyperdimensional computing approach for efficient similarity hashing,” *arXiv preprint arXiv:2404.11025*, 2024.

[11] Y. Ni, N. Lesica, F.-G. Zeng, and M. Imani, “Neurally-inspired hyperdimensional classification for efficient and robust biosignal processing,” in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022, pp. 1–9.

[12] 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*, 2022, pp. 1–9.

[13] R. Chen, M. Sodhi, M. Imani, M. Khanzadeh, A. Yadollahi, and F. Imani, “Brain-inspired computing for in-process melt pool characterization in additive manufacturing,” *CIRP Journal of Manufacturing Science and Technology*, vol. 41, pp. 380–390, 2023.

[14] A. Hernández-Cano, C. Zhuo, X. Yin, and M. Imani, “Real-time and robust hyperdimensional classification,” in *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, 2021, pp. 397–402.

[15] Y. Ni, W. Y. Chung, S. Cho, Z. Zou, and M. Imani, “Efficient exploration in edge-friendly hyperdimensional reinforcement learning,” in *Proceedings of the Great Lakes Symposium on VLSI 2024*, 2024, pp. 111–118.

[16] H. Chen and M. Imani, “Density-aware parallel hyperdimensional genome sequence matching,” in *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2022, pp. 1–4.

[17] Y. Ni, H. Chen, P. Poduval, Z. Zou, P. Mercati, and M. Imani, “Brain-inspired trustworthy hyperdimensional computing with efficient uncertainty quantification,” in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 01–09.

[18] A. Hernandez-Cane *et al.*, “Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system,” in *DATE*, 2021.

[19] X. Ying, “An overview of overfitting and its solutions,” in *Journal of physics: Conference series*, vol. 1168. IOP Publishing, 2019, p. 022022.

[20] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[21] Y. Kumar, A. Koul, R. Singla, and M. F. Ijaz, “Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda,” *Journal of ambient intelligence and humanized computing*, pp. 1–28, 2022.

[22] R. E. Schapire, “Explaining adaboost,” in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*. Springer, 2013, pp. 37–52.

[23] D. Kleyko, D. Rachkovskij, E. Osipov, and A. Rahimi, “A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges,” *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–52, 2023.

[24] 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. [Online]. Available: https://doi.org/10.1613%2Fjair.1.12664

[25] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.

[26] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in neural information processing systems*, vol. 30, 2017.

[27] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou *et al.*, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[28] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, “Introducing wesad, a multimodal dataset for wearable stress and affect detection,” in *Proceedings of the 20th ACM international conference on multimodal interaction*, 2018, pp. 400–408.

[29] A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, “Deep PPG: Large-Scale heart rate estimation with convolutional neural networks,” *Sensors (Basel)*, vol. 19, no. 14, Jul. 2019.

[30] H. E. Barkam, S. E. Jeon, S. Yun, C. Yeung, Z. Zou, X. Jiao, N. Srinivasa, and M. Imani, “Hyperdimensional computing for resilient edge learning,” in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–8.

[31] H. E. Barkam, S. Yun, H. Chen, P. Gensler, A. Mema, A. Ding, G. Michelogiannakis, H. Amrouch, and M. Imani, “Reliable hyperdimensional reasoning on unreliable emerging technologies,” in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–9.

[32] H. E. Barkam, S. E. Jeon, C. Yeung, Z. Zou, X. Jiao, and M. Imani, “Comprehensive analysis of hyperdimensional computing against gradient based attacks,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–2.

[33] H. Chen, A. Zakeri, F. Wen, H. E. Barkam, and M. Imani, “Hypergraf: Hyperdimensional graph-based reasoning acceleration on fpga,” in *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE, 2023, pp. 34–41.

[34] M. Imani, D. Kong, A. Rahimi, and T. Rosing, “Voicehd: Hyperdimensional computing for efficient speech recognition,” in *2017 IEEE international conference on rebooting computing (ICRC)*. IEEE, 2017, pp. 1–8.

[35] M. Issa, S. Shahhosseini, Y. Ni, T. Hu, D. Abraham, A. M. Rahmani, N. Dutt, and M. Imani, “Hyperdimensional hybrid learning on end-edge-cloud networks,” in *2022 IEEE 40th International Conference on Computer Design (ICCD)*. IEEE, 2022, pp. 652–655.

[36] P. Poduval, M. Issa, F. Imani, C. Zhuo, X. Yin, H. Najafi, and M. Imani, “Robust in-memory computing with hyperdimensional stochastic representation. in 2021 ieee,” in *ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 1–6.

[37] P. Poduval, Z. Zou, H. Najafi, H. Homayoun, and M. Imani, “Stochd: Stochastic hyperdimensional system for efficient and robust learning from raw data,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1195–1200.

[38] F. Götze and A. Tikhomirov, “Rate of convergence in probability to the marchenko-pastur law,” *Bernoulli*, vol. 10, no. 3, pp. 503–548, 2004.

[39] S. Hosseini, R. Gottumukkala, S. Katragadda, R. T. Bhupatiraju, Z. Ashkar, C. W. Borst, and K. Cochran, “A multimodal sensor dataset for continuous stress detection of nurses in a hospital,” *Scientific Data*, vol. 9, no. 1, p. 255, 2022.

[40] T. Iqbal, A. J. Simpkin, D. Roshan, N. Glynn, J. Killilea, J. Walsh, G. Molloy, S. Ganly, H. Ryman, E. Coen *et al.*, “Stress monitoring using wearable sensors: A pilot study and stress-predict dataset,” *Sensors*, vol. 22, no. 21, p. 8135, 2022.