AMS-HD: Acute Mountain Sickness Detection with Hyperdimensional Computing

Abu Kaisar Mohammad Masum[+], Reeti Pradhananga[+], Jonas I. Schmidt[+], Mehran Shoushtari Moghadam[✤],
M. Hassan Najafi[✤], Bige Unluturk[†], Ulkuhan Guler[*], Sercan Aygun[+]
[+]School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA
[✤]Electrical, Computer, and Systems Engineering Department, Case Western Reserve University, Cleveland, OH, USA
[†]Electrical and Computer Engineering & Biomedical Engineering, Michigan State University, East Lansing, MI, USA
[*]Electrical & Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA
{c00591145, c00563648, c00285531, sercan.aygun}@louisiana.edu, {moghadam, najafi}@case.edu, unluturk@msu.edu, uguler@wpi.edu

2025 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-5683-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCAS56072.2025.11043375

***Abstract*—Acute mountain sickness (AMS) is a potentially life-threatening condition that affects many individuals traveling to high altitudes. Early diagnosis is crucial, especially for travelers who may not have immediate access to medical resources. While traditional machine learning (ML) methods have been used to detect AMS using biomedical data (e.g., heart rate, blood oxygen saturation, respiration rate, blood pressure, and body temperature), hyperdimensional computing (HDC) has yet to be explored for this purpose using the few of biomedical data. Previous classification methods fall short of balancing accuracy with low hardware complexity, but HDC offers a promising solution. HDC provides a hardware-efficient alternative solution, making it well-suited for resource-constrained environments, such as wearable devices. Its lightweight architecture and efficient memory management make it ideal for embedded systems, enabling real-time AMS detection with accuracy comparable to traditional ML models. We introduce `AMS-HD`, a novel framework that leverages custom feature engineering and quasi-random hypervector encoding to further enhance the efficiency and accuracy of HDC for AMS detection. The proposed framework demonstrates the potential for seamless integration into wearable biomedical devices for on-the-go health monitoring.**

I. INTRODUCTION

Acute mountain sickness (AMS) is a common medical condition that can affect individuals who ascend rapidly to high altitudes, typically above $2,500$ meters, where the partial pressure of oxygen in the air decreases. Some common symptoms of AMS include headaches, nausea, tiredness, and vomiting, which can start at $6-12$ hours after reaching a high altitude. If AMS is left untreated, it can lead to life-threatening severe consequences such as high altitude cerebral edema [1]. Early detection and alerting patients to the symptoms is highly important. Leveraging lightweight wearable devices for continuous monitoring of vital signals can be a promising solution. If symptoms are analyzed promptly, early diagnosis could save many lives. Conventional machine learning (ML) methods have been explored for this task [2]. However, they are computationally intensive and unsuitable for regular and prompt diagnoses. For the first time, this work proposes an alternative solution for fast, accurate, and resource-efficient detection of AMS by leveraging an emerging model of computing, hyperdimensional computing (HDC).

HDC is an emerging paradigm that mimics certain key brain functions to create efficient and noise-resistant ML models [3], [4]. HDC has proven effective in various learning tasks, demonstrating strengths in both performance and reliability [5], [6]. The foundation of HDC is based on the concept that the essential aspects of human memory, perception, and thinking can be represented through mathematical properties of hyperdimensional spaces. High-dimensional vectors, termed hypervectors ($\mathcal{HV}$s), define the space in which HDC operates. HDC models consist of randomly arranged binary ('0's and '1's in logic systems) or bipolar values ('-1's and '$+1$'s in software platforms). This unconventional representation enables fast, reliable, and parallel processing of large amounts of data [7]. $\mathcal{HV}$s are one-dimensional vectors with D binary elements, where D ranges from hundreds to thousands. Achieving high-quality results requires orthogonality between $\mathcal{HV}$s, meaning the $\mathcal{HV}$s must be independent of one another. Traditional HDC systems generate pseudo-random vectors which are nearly orthogonal. Low-discrepancy (LD) sequences have been recently introduced to improve orthogonality between $\mathcal{HV}$s inspired by quasi-randomness suggested for high-quality bit-streams in stochastic computing [8], [9].

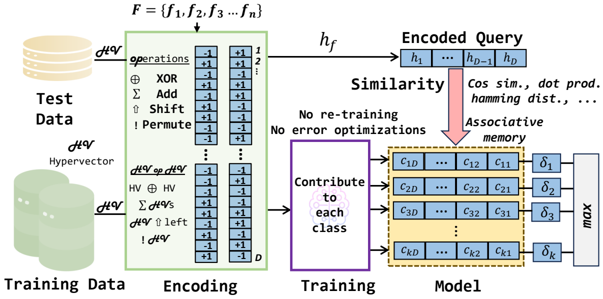

Fig. 1. Overview of an HDC framework: encoding, training, and similarity check.

Fig. 1 illustrates an overview of an HDC system. The system encodes the training and test data similarly based on the input features (f). The encoded vectors are processed using simple logical operations such as `XOR`, `Add`, `Shift`, and `Permute`. All encoded training samples contribute to the ML model. Each class of the contributing data outputs a *Class* $\mathcal{HV}$. During the inference, the test samples are encoded with similar encoding steps. A query $\mathcal{HV}$ is returned for each sample. In the classification phase, the system checks the similarity between the *Class* $\mathcal{HV}$*s* and the incoming query. The highest similarity indicates the closest match, the class of the test sample.

This work introduces **`AMS-HD`**, an HDC framework for fast and efficient AMS detection. Accurate *feature encoding* is essential for compelling AMS detection with HDC. The proposed framework enhances data encoding and feature engineering to efficiently process health-related information at high altitudes. **`AMS-HD`** enables real-time analysis of vital signs to facilitate proactive measures against AMS. The proposed solution highlights HDC's potential to transform health monitoring and predictive analytics, contributing to safer experiences at high altitudes. With its acceptable classification performance,

---

This work is supported in part by National Science Foundation (NSF) grants 2019511, 2339701, National Institute of Health (NIH) under Grant R01HL172293, and generous gift from Nvidia.

our solution can be integrated seamlessly into low-cost, low-power devices like pulse oximeters and smartwatches, making it ideal for real-time monitoring and early detection of AMS.

II. BACKGROUND

A. Conventional ML in AMS Prediction

ML algorithms can predict the severity of AMS, helping individuals take necessary precautions and descend from the high-altitude area if needed. Prior studies proposed ML algorithms such as regression and classification on physiological and environmental data for AMS detection [10]. Yang et al. [11] used ML to predict the susceptibility of severe AMS (sAMS) based on *genetic data*. However, real-time processing of genetic data is complicated and challenging. More physical and easy-to-handle data such as blood oxygen saturation SpO_2 and *Heart Rate* data are valuable to discovering the potential risk of AMS [12]. Information such as altitude, ambient temperature, atmospheric pressure, relative humidity, climbing speed, and heart rate variability can also be used for more accurate detection. Bagged trees, logistic regression (LR), linear support vector machines (SVM), and weighted k-nearest neighbor (kNN) are examples of prior approaches in the literature [2]. Moreover, there are connections between sleep quality and AMS, measured by intermittent hypoxia training, another research area on AMS. By generating pseudo-labels for AMS susceptibility and training a long short-term memory model to classify hypoxia tolerance, Wang et al. achieved acceptable results for AMS detection [13], [14]. While effective, all these prior solutions rely on powerful computers for real-time processing. However, when it comes to real-time processing with cost-effective and lightweight solutions with low power, run-time, and memory usage, an HDC-based solution can offer significant benefits.

B. An Emerging Computing Paradigm: HDC

HDC is a computational model each data point is represented by an $\mathcal{HV}$ consisting of hundreds to thousands of binary/bipolar components. $\mathcal{HV}$s are generated randomly to provide near/full orthogonality. HDC excels at representing symbolic information. The orthogonality helps a learning system easily distinguish between unique symbols. HDC is also robust against noise and errors. This stems from its unique data representation that does not depend on the significance of individual bits, such as the least significant bit (LSB), most significant bit (MSB), or sign bit.

HDC encoding involves several operations, such as *binding*, *bundling*, *shifting*, and *permutation* [15]. The information from HDC vectors is retrained in a new composite $\mathcal{HV}$ using *binding* operations. Element-wise multiplication on bipolar values (± 1) (`XOR` on 1-0 binary vectors) produces resultant $\mathcal{HV}$. Combining multiple $\mathcal{HV}$s into a single vector is known as *bundling*. This preserves the robustness of the data by aggregating holistic information with an invertible function. *Permutation* is another operation that preserves orthogonality by randomly rearranging the $\mathcal{HV}$ elements [16]. Conventional HDC uses pseudo-random methods for generating $\mathcal{HV}$s. However, this could lead to poor orthogonality and performance degradation [17]. Recent works use quasi-random sequences, like Sobol sequences [6], [18], for better orthogonality and higher quality and accuracy in HDC.

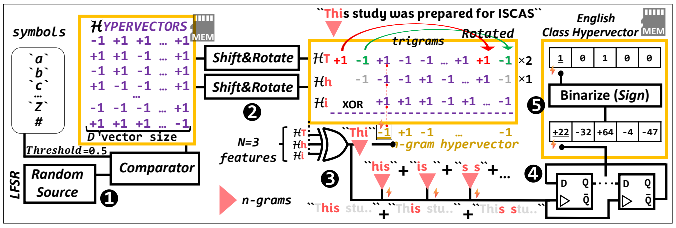

Fig. 2. An HDC classifier for language text processing. n-gram processing is exemplified. ❶: Random vector generation, ❷: Orthogonality preservation via shifting, ❸: Binding, ❹: Bundling, and ❺: Binarization via *subtract* from a threshold.

HDC has been used for many applications and ML tasks [19]. Fig. 2 illustrates the basic HDC operations over a language processing case study. Step ❶ includes vector generation. Text symbols (i.e., letters) are converted to D-sized $\mathcal{HV}$s using a random number generator (LFSR: Linear feedback shift register -*pseudo-random source*-) and a comparator. $\mathcal{HV}$s require orthogonality, so random numbers are compared with a 0.5 threshold value for 50%-50% $+1$ and -1 ratio in the vectors. The generated vectors are then utilized in encoding starting from Step ❷. The shifting operation keeps orthogonality before the multiplication (i.e., `XOR`), and the resultant vector is also orthogonal to the multiplicands. The language processing example in Fig. 2 utilizes n-gram processing, similar to 1-dimensional convolutions on the incoming text to be represented in the final $\mathcal{HV}$. Step ❸ applies 3-gram multiplications (*binding*) of each 3-element subset in the text, yielding an n-gram $\mathcal{HV}$ at each subset. In Step ❹, the resultant n-gram vectors are added column-wise (*bundling*) using a simple counter. Up to this step, any incoming text for the class-of-interest (here, the *English* language) contributes to the accumulation. When the training set is scanned for the corresponding class, a binarization is applied in Step ❺. This is the class $\mathcal{HV}$ that represents the *English* class in this example. Other languages are processed similarly, and other class $\mathcal{HV}$s are obtained for language classifications. Class $\mathcal{HV}$s are the model of the ML system ready to be deployed on any resource-constrained device such as wearable devices. Any incoming query follows the same encoding steps during the real-time processing, being compared by each class $\mathcal{HV}$. A simple dot product (or cosine similarity, hamming distance, etc.) is applied between each class $\mathcal{HV}$ and the query vector; the highest similarity is the classification decision.

III. PROPOSED **AMS-HD**

In this section, we explain the step-by-step process of our HDC classifier for classifying AMS based on two physiological parameters: *heartbeat rate (HR)* and *SpO_2 (Oxygen Saturation)*. SpO_2 measures the percentage of oxygen in the blood, while HR indicates the number of heartbeats per minute. As shown in Fig. 3, **AMS-HD** leverages HDC for binary and multiclass classifications of AMS. Unlike the prior art, this work focuses on feature engineering, data mining, and performance evaluation aspects of HDC in a *tabloid* biomedical data (i.e., patient samples on different altitudes recorded in a table with AMS scores) classification system. We discuss each component of the proposed system, from the data to the HDC classifier part.

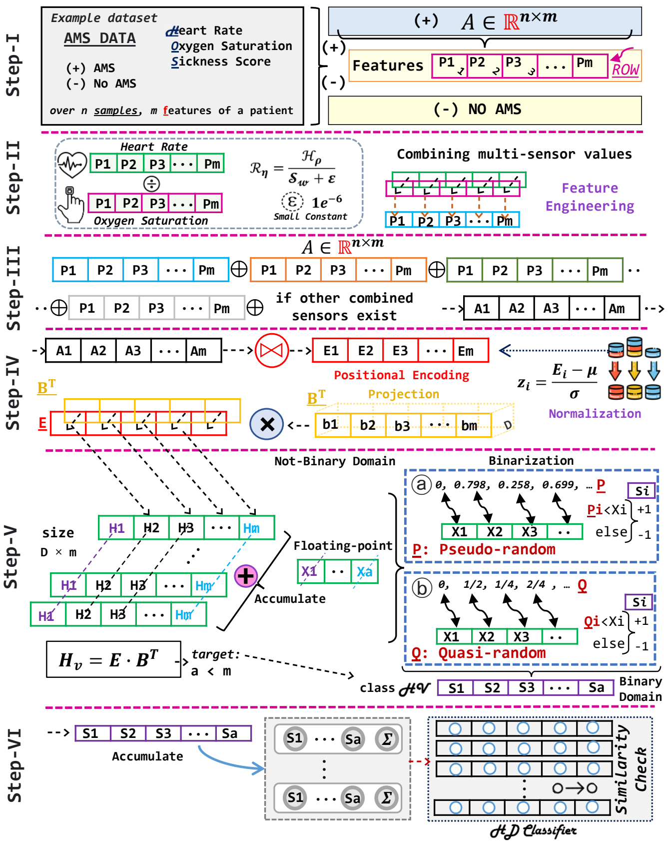

Fig. 3. **AMS-HD**: Proposed ML framework for classification task using Feature Engineering, HDC: Positional Encoding, Random $\mathcal{HV}$ Projection, and HD classifier (ⓐ Pseudo- vs. ⓑ Quasi-random).

A. Dataset Overview

In this study, we utilize a dataset that captures the body's response to stress in high-altitude environments, focusing on immune system changes through blood sample analysis. The original dataset included features such as SpO_2 (%), HR (bpm), carbon monoxide - CO (%), CO (ppm), and AMS Score [20]. From these, we selected two key features: SpO_2 and HR. Our tasks are divided into binary (as in Fig. 3 `Step-I`) and multiclass classifications. The AMS risk scoring system ranges from 1 to 12. For the binary case, an AMS score of 3 or higher is labeled as '*AMS*' with an alerting situation, while scores below 3 are labeled as '*NO AMS*.' For the multiclass classification, the AMS severity is categorized into three levels: *mild* (3–5), *moderate* (6–9), and *severe* (10–12), following the standardized classification of AMS [21].

B. Data Preparation and Feature Engineering

Selecting the right features from the dataset illustrated in `Step-I` of Fig. 3 is crucial in any ML task. In `Step-II`, we select features $P_1, P_2, \ldots, P_m$ from the dataset. We apply offline feature engineering to the dataset during training to enhance the quality and relevance of the input data. By tuning HR and SpO_2 features, we improve the model's predictive accuracy and ability to identify potential health risks at high altitudes. We propose a formulation: $R_n = \frac{H_p}{s_w+\epsilon}$, where H_p is the HR (or other sensor value) and s_w is the standard deviation of the corresponding feature with a negligibly small constant, ϵ (e.g.10^{-6}) that prevents division by zero. In Fig. 3, the P (blue) represents features: HR, SpO_2. They are transformed through feature engineering into this final vector, P. The division symbol in `Step-II` represents the heart rate divided by the oxygen saturation and yields the combined sensor package, P, in feature engineering. The proposed approach operates on the input data matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$. There are n samples and m scalar features in A, such as HR, SpO_2, and sickness score. `Step-III`, shows the features in case other combined sensors exist (multiple Ps) holding scalar numerical information (in floating point) accumulated.

C. Positional Encoding and Data Projection

The engineered features are encoded by positional encoding, denoted by E. These positions are projected afterward with matrix multiplication. Matrix B helps fix the dimensionality in `Step-IV`. In the HDC system, this resembles the *binding* (❸) operation. The positionally encoded vector E_i is an enhanced feature representation of the data by each position. These values are normalized using z-score normalization, where μ and σ represent the *mean* and *standard deviation* of features: $z_i = \frac{E_i-\mu}{\sigma}$.

D. $\mathcal{HV}$ Generation

To represent all features combined, non-binary scalars or floating-point $\mathcal{HV}$s must be created. In `Step-V`, random $\mathcal{HV}$s $H_i \in \mathbb{R}^{D\times m}$ are employed to produce feature vectors. In our HDC framework, H_is are projected to *accumulate* using a similar approach to *bundling* (❷). These projected $\mathcal{HV}$s have a higher dimensionality D than the original feature space. These $\mathcal{HV}$s are used to map the feature vectors into a high-dimensional space (depth, not feature-wise n), allowing for efficient encoding of patterns within the dataset. Following the *bundling* operation, the resulting scalars (Xs) are converted to the binary domain, creating the S class $\mathcal{HV}$. This process utilizes either ⓐ pseudo-random or ⓑ quasi-random sources following the recent trend in state-of-the-art HDC designs.

1) ⓐ Pseudo-random $\mathcal{HV}$ Generation (HDC-R): In the first approach (Fig. 3 `Step-V` ⓐ), pseudo-random $\mathcal{HV}$s are created by using pseudo-random numbers from a hardware module such as an LFSR or a software platform (such as the `rand()` function in Python or MATLAB). The random numbers can also be generated based on predefined random sequences. These may provide hand-crafted randomness to maintain statistical properties that are beneficial for the classification task of HDC. Pseudo-orthogonal *class $\mathcal{HV}$s* are approximately well-distinguished from each other. However, multiple iterations may be required during the training phase to find the best random number candidates during the generation of $\mathcal{HV}$s.

2) ⓑ Quasi-random $\mathcal{HV}$ Generation (HDC-Q): Quasi-random $\mathcal{HV}$s (Fig. 3 `Step-V` ⓑ) are generated by using quasi-random sequences, such as Sobol [8]. These sequences are often called LD sequences [22], as they provide recurrent properties with more uniform distributions than pseudo-random sequences. The orthogonality properties are superior to the pseudo-random case, providing better vector representation. Quasi-random $\mathcal{HV}$s efficiently cover the space for high-dimensional features, resulting in a better performance.

`Step-V` is similar to Step ❶ of Fig. 2: *vector generation with comparison*. In Fig. 3, both in ⓐ and ⓑ, random numbers are compared with X_i; if the random number is less than

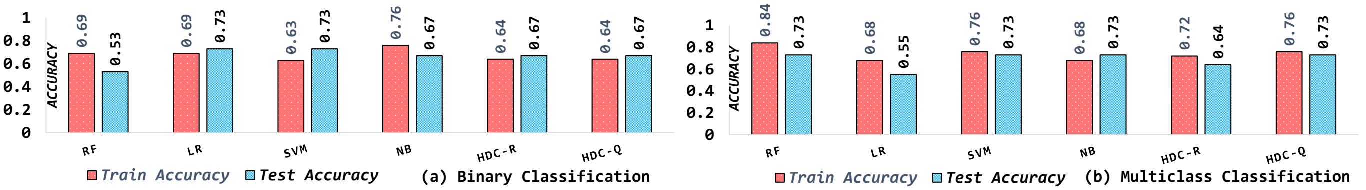

Fig. 4. Comparison of training and testing accuracy of different classifiers, (a) Binary classification, and (b) Multiclass classification (NB: Naive Bayes, SVM: Support Vector Machine, LR: Logistic Regression, RF: Random Forest).

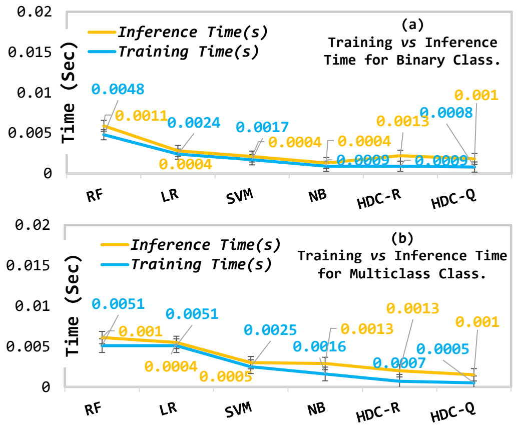

Fig. 5. Training & inference time (a) Binary (b) Multiclass.

X_i, S vector position is set to '+1', otherwise it is set to '−1'. In hardware, these values are set by logic-1 and logic-0, respectively.

E. $\mathcal{HD}$ Classifier

`Step-VI` in Fig. 3 demonstrates the HDC classifier. By combining the contributions from each class (S) using accumulation, we create the final deployable model. Each class represents the '*AMS*' versus '*NO AMS*' binary classifier or AMS severity screening for *mild*, *moderate*, and *severe* cases. The similarity check in the testing phase compares the input with those predefined class $\mathcal{HV}$s $S_1, S_2, \ldots, S_a$. Cosine similarity computes the similarity between a testing query $\mathcal{HV}$ and each class $\mathcal{HV}$ defined as $(\mathcal{HV}, S_i) = \frac{\mathcal{HV} \cdot S_i}{\|\mathcal{HV}\| \|S_i\|}$. The classifier calculates the similarity score for each class in the model versus the query, and the class with the highest score is selected as the predicted class for the input sample.

IV. EXPERIMENTAL RESULTS

Fig. 4 compares the training and testing accuracy of various ML classifiers with the HDC method for binary (Fig. 4(a)) and multiclass (Fig. 4(b)) AMS classification. For binary classification with conventional ML, the highest test accuracy was achieved by *LR* and *SVM* with 73%, while the lowest was from the random forest (*RF*) with 53% success. However, in multiclass classification, one of the best performance was achieved by quasi-random HDC similar to *RF*, *SVM*, and (naive Bayes) *NB* with a test accuracy of 73%. Regarding timing, HDC exhibits the shortest training times, with the quasi-random method achieving the lowest inference time in Fig. 5.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS REGARDING MEMORY USAGE, POWER, AND ENERGY.

Model	Mem (MB)	Min P (W)	Max P (W)	Avg E (J)
HDC-Q	0.0002	2.6	3.0	0.02120
HDC-R	0.0002	2.6	3.3	0.02655
NB	0.0003 ↑	2.7 ↑	3.6 ↑	0.00285 ↓
SVM	0.0010 ↑	2.7 ↑	3.6 ↑	0.00533 ↓
LR	0.0003 ↑	2.7 ↑	3.6 ↑	0.00753 ↓
RF	0.0003 ↑	2.7 ↑	3.6 ↑	0.01512 ↓

Arrows indicate increases (↑) or decreases (↓) compared to HDC.

TABLE II
CLASSIFICATION PERFORMANCE OF VARIOUS MODELS

Binary	RF	LR	SVM	NB	HDC-R	HDC-Q
Accuracy	0.79	0.79	0.79	0.79	0.70	0.70
Recall	0.69	0.69	0.69	0.69	0.68	0.68
F1-score	0.64	0.64	0.64	0.64	0.66	0.66
Multiclass	**RF**	**LR**	**SVM**	**NB**	**HDC-R**	**HDC-Q**
Accuracy	0.78	0.53	0.86	0.86	0.72	0.72
Recall	0.75	0.56	0.72	0.72	0.64	0.72
F1-score	0.73	0.54	0.73	0.73	0.64	0.72

Table I shows the performance comparison for the binary and multiclass classifiers. Quasi-random and pseudo-random have the lowest memory (0.0002 MB) and power consumption (min: 2.6W, max: 3.3W), highlighting better memory and power usage in HDC methods. *NB* uses more memory but is the most energy-efficient model. Conventional models consume more memory and power (P), which makes them unsuitable for power-constrained designs.

Performance results for different models for binary and multiclass are summarized in Table II. The precision score observed in all models was relatively consistent, with all conventional methods leading to 79% accuracy for binary classification. The HDC models (HDC-R, HDC-Q) show slightly lower accuracy around 70%. On the other hand, their Recall and F1-score are consistent across all the compared models, and the competitive F1-score is around 66% for the HDC models. For multiclass classification, *SVM* excels with the highest accuracy of 86% and F1-score of 73%. The HDC-Q model maintains consistent performance with accuracy, recall, and F1-score of 72%. The overall performance of HDC demonstrates competitive results, especially in multiclass tasks, providing a good balance between F1-score and recall.

V. CONCLUSION

HDC offers a powerful solution for various ML tasks, especially in biomedical applications. This work introduced **`AMS-HD`**, an HDC framework for real-time, accurate, and resource-efficient detection of acute mountain sickness (AMS). The proposed framework is highly suitable for real-time health monitoring and AMS diagnosis at altitude. **`AMS-HD`** can be integrated with wearable and IoT devices for enhanced health monitoring and timely interventions, saving many lives.

REFERENCES

[1] E. Hohenhaus, A. Paul, R. McCullough, H. Kucherer, and P. Bartsch, "Ventilatory and pulmonary vascular response to hypoxia and susceptibility to high altitude pulmonary oedema," *European Respiratory Journal*, vol. 8, no. 11, pp. 1825–1833, 1995.

[2] C.-Y. Wei, P.-N. Chen, S.-S. Lin, T.-W. Huang, L.-C. Sun, C.-W. Tseng, and K.-F. Lin, "Using machine learning to determine the correlation between physiological and environmental parameters and the induction of acute mountain sickness," *BMC bioinformatics*, vol. 22, no. Suppl 5, p. 628, 2021.

[3] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, pp. 139–159, 2009.

[4] 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 *IEEE CLOUD*. IEEE, 2019, pp. 435–446.

[5] A. Hernández-Cano, C. Zhuo, X. Yin, and M. Imani, "Real-time and robust hyperdimensional classification," in *GLSVLSI*, 2021, pp. 397–402.

[6] S. Aygun, M. H. Najafi, and M. Imani, "A linear-time, optimization-free, and edge device-compatible hypervector encoding," in *DATE'23*.

[7] O. Räsänen and S. Kakouros, "Modeling dependencies in multiple parallel data streams with hyperdimensional computing," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 899–903, 2014.

[8] S. Liu and J. Han, "Toward energy-efficient stochastic circuits using parallel sobol sequences," *IEEE TVLSI*, vol. 26, no. 7, pp. 1326–1339, 2018.

[9] S. Aygun and M. H. Najafi, "Sobol sequence optimization for hardware-efficient vector symbolic architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2024.

[10] Y. Wu, P. Li, Z. Zhong, J. Xie, S. Zhou, Y. Gao, and J. Chen, "Assessment of acute mountain sickness: How to integrate the advantages of the lake louise score and the chinese ams score," 2020.

[11] M. Yang, Y. Wu, X.-b. Yang, T. Liu, Y. Zhang, Y. Zhuo, Y. Luo, and N. Zhang, "Establishing a prediction model of severe acute mountain sickness using machine learning of support vector machine recursive feature elimination," *Scientific Reports*, vol. 13, no. 1, p. 4633, 2023.

[12] T. Dünnwald, R. Kienast, D. Niederseer, and M. Burtscher, "The use of pulse oximetry in the assessment of acclimatization to high altitude," *Sensors*, vol. 21, no. 4, p. 1263, 2021.

[13] L. Wang, R. Xiao, J. Chen, L. Zhu, D. Shi, and J. Wang, "A slow feature based lstm network for susceptibility assessment of acute mountain sickness with heterogeneous data," *Biomedical Signal Processing and Control*, vol. 80, p. 104355, 2023.

[14] L. Wang, D. Shi, L. Zhu, and J. Wang, "Event-triggered pseudo supervised meta learning for susceptibility assessment of acute mountain sickness," in *2024 39th YAC*.

[15] A. Kazemi, F. Müller, M. M. Sharifi, H. Errahmouni, G. Gerlach, T. Kämpfe, M. Imani, X. S. Hu, and M. Niemier, "Achieving software-equivalent accuracy for hyperdimensional computing with ferroelectric-based in-memory computing," *Scientific Reports*, vol. 12, no. 1, p. 19201, Nov 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-23116-w

[16] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *ISLPED*, 2016, pp. 64–69.

[17] M. Moghadam, S. Aygun, F. S. Banitaba, and M. H. Najafi, "All you need is unary: End-to-end unary bit-stream processing in hyperdimensional computing," ser. ISLPED '24, 2024, p. 1–6. [Online]. Available: https://doi.org/10.1145/3665314.3670834

[18] M. S. Moghadam, S. Aygun, and M. H. Najafi, "No-multiplication deterministic hyperdimensional encoding for resource-constrained devices," *IEEE Embedded Systems Letters*, vol. 15, no. 4, pp. 210–213, 2023.

[19] S. Aygun, M. S. Moghadam, M. H. Najafi, and M. Imani, "Learning from hypervectors: A survey on hypervector encoding," 2023. [Online]. Available: https://arxiv.org/abs/2308.00685

[20] K. Pham, S. Frost, K. Parikh, N. Puvvula, B. Oeung, and E. C. Heinrich, "Inflammatory gene expression during acute high-altitude exposure," *The Journal of physiology*, vol. 600, no. 18, pp. 4169–4186, 2022.

[21] R. C. Roach, P. H. Hackett, O. Oelz, P. Bärtsch, A. M. Luks, M. J. MacInnis, J. K. Baillie, and L. L. A. S. C. Committee, "The 2018 lake louise acute mountain sickness score," *High altitude medicine & biology*, vol. 19, no. 1, pp. 4–6, 2018.

[22] M. S. Moghadam, S. Aygun, M. R. Alam, and M. H. Najafi, "P2lsg: Powers-of-2 low-discrepancy sequence generator for stochastic computing," in *2024 29th ASP-DAC*.