DARPA HyDREA: Hyperdimensional Computing: Robust, Efficient & Accurate

PI Tajana S Rosing, UCSD; tajana@ucsd.edu
Co-PIs T. Javidi & S. Dasgupta, UCSD; N. Zachariah, I. Doxas, NGMS

Hyper-Dimensional (HD) Data Enabled Neural Networks (HyDDENN) *Artificial Intelligence*

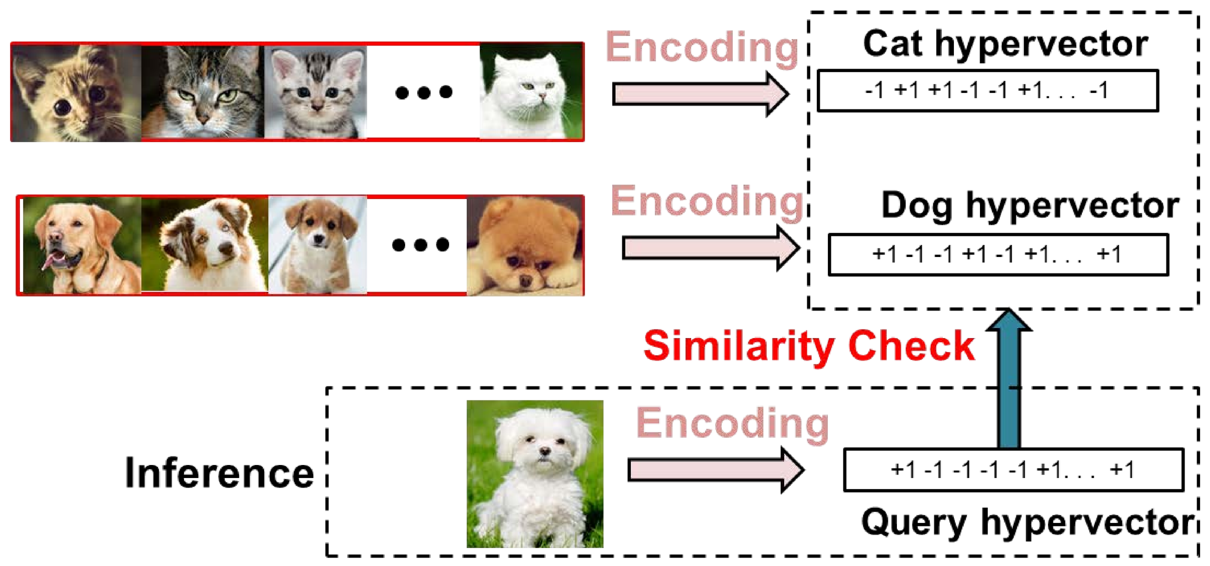
Why HD Computing?

Conventional machine learning (ML) approaches are not well suited to edge computing

- Model training is energy intensive, high-latency, requires complex optimization algorithms and is not robust to noise

Hyperdimensional (HD) computing uses high dimensional data representation, called hypervectors of ~10k bits, to address these issues:

- Inherent noise robustness due to large dimensionality
  - e.g. HD computing shows no change in classification accuracy even with 50% bits corrupted
- Supports real-time learning and reasoning; interpretable
  - Training can be done in just a single pass & can be done in distributed fashion on edge devices
- Simple operations (+, XOR, *, nearest search) that are easy to accelerate in HW due to high-parallelism



Project Goals and Metrics:

Design & demonstrate Hyperdimensional computing architecture that is as accurate but 100x more efficient than the state of the art

Technical Approach:

- Theoretical analysis of HD capabilities, definition of data encoding & decoding primitives
- Design HD computing architecture for classification and clustering
- Design the initial HW implementation of HD architecture
- Evaluate accuracy, robustness & efficiency vs. state of the art for contextual edge sensing applications & defense RF communications

Robust HD Encoding & Decoding

HD Encoding maps data $x \in \mathbb{R}^n$ to hypervector $\phi(x) \in \mathbb{Z}^{d \gg n}$

- Position ID Encoding
  - Quantizes x into m bins and encodes as a sequence
  - Preserves L1 distances up to additive distortion
  - Increases distances between clusters: robust to noise!
- Random Projection Encoding
  - Project data onto d *random* directions in $\mathbb{R}^n$ & quantize
  - Preserves Euclidean distances up to additive distortion
  - Encodings are sparse - only $k \ll d$ bits matter

HD decoding

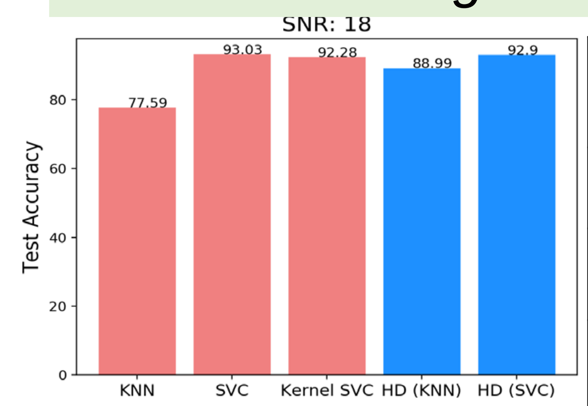
- Returns original data when hypervector dimensionality *O(N logM)*, where data consists of *N* symbols drawn from alphabet of size *M*

Applications of HD Computing

HD-based RF Signal Classification

- Goal: classify the modulation used by nearby transmitters
- Datasets created with dynamic GNU radio channel model using AWGN, random walk process to simulate carrier frequency drift & sample rate offset drive, Rayleigh fading

HD classification has comparable accuracy vs. conventional machine learning but is more computationally efficient

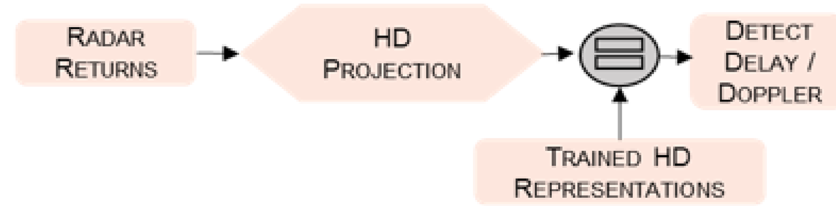


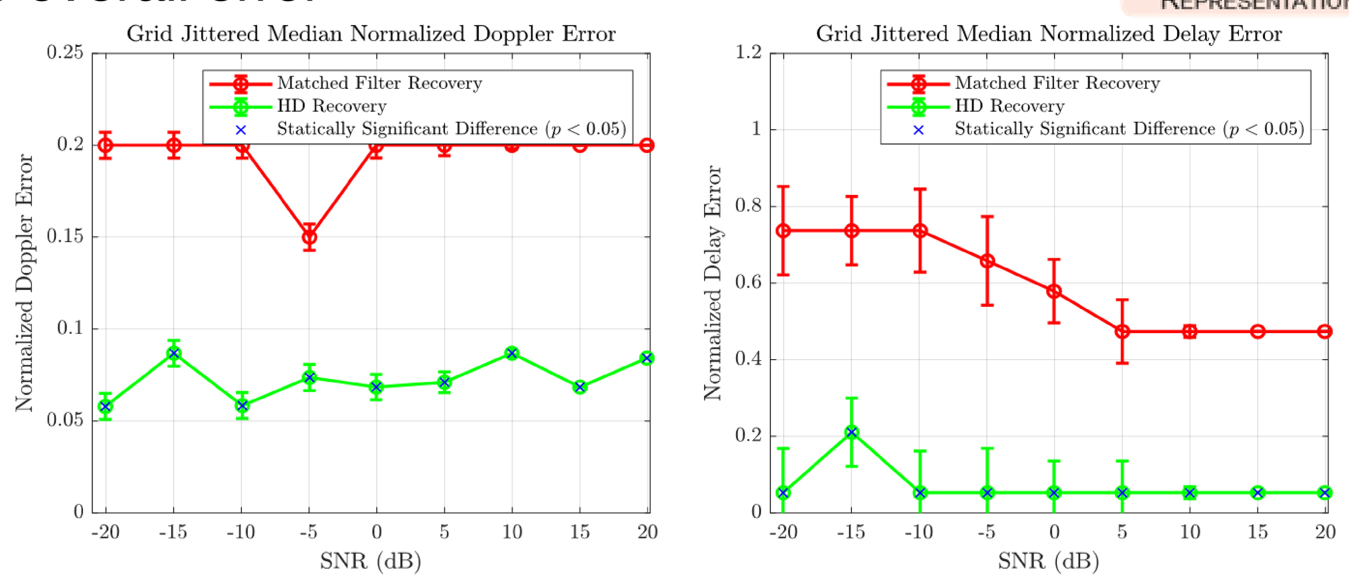
	KNN	Linear SVC	Quant + LinSVC	HD + KNN	HD + SVC	HD Percept.
AWGN	84	98	98	95	98	97
Dynamic SNR 6dB	65	90	90	81	90	90
Dynamic SNR 18dB	78	93	92	89	93	92

***HD has three orders of magnitude better margin** as the information is spread across larger dimensionality*

HD Application to Matched Filtering (by NGMS)

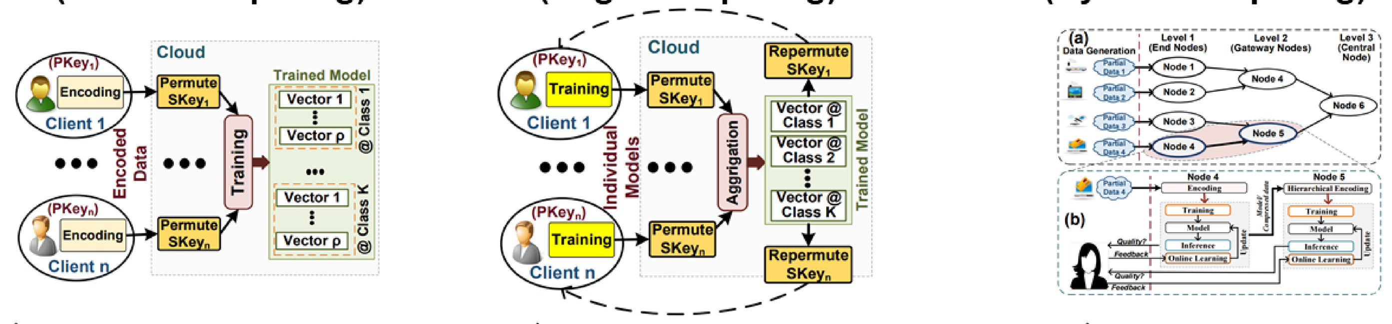
Goal: Enhance matched filtering with HD to reduce the overall error





HD reduces error by 80% for Delay & Doppler Estimates as compared to the Matched Filter

Secure Distributed HD Learning



- ✔ Send HVs during training
- ✔ High speed learning ☺
- ✔ Send pretrained HVs per single data type
- ✔ Drastically lower bandwidth ☺
- ✔ Distributed learning with varying data types
- ✔ Combine hypervectors to aggregate information

*SecureHD **encoding is 146× faster** and **decoding is 6.8× faster** vs homomorphic encryption [Microsoft SEAL]*

Results: HD Classification & Clustering

- Datasets:
  - UCI ML Repository: Iris, Isolet, UCIchar (activity recognition)
  - Cardio (medical), EMG (gesture recognition), Face (face recognition)
  - Fundamental Clustering Prob Suite: Hepta, Tetra, TwoDiamonds, Wingnut
- Measurements done on:
  - CPU: Intel i7-8700K with 16GB RAM
  - GPU: Nvidia GTX 1080 Ti with 11GB VRAM
  - FPGA: Kintex-7 (KC705)
  - PIM: Processing-In-Memory, only simulated
    - Simulations on 45 nm technology node in Cadence Virtuoso
    - VTEAM ReRAM model: Ron/Roff = 10k/10MΩ, SET/RESET = 1.1ns

HD Classification PIM it **167,000x faster** for inference vs. perceptron, which was the fastest of the state of the art

Speed vs HD	SVM	Precept	MLP
Training	2.9	1.9	7.3
Testing	13.8	1.7	4.8

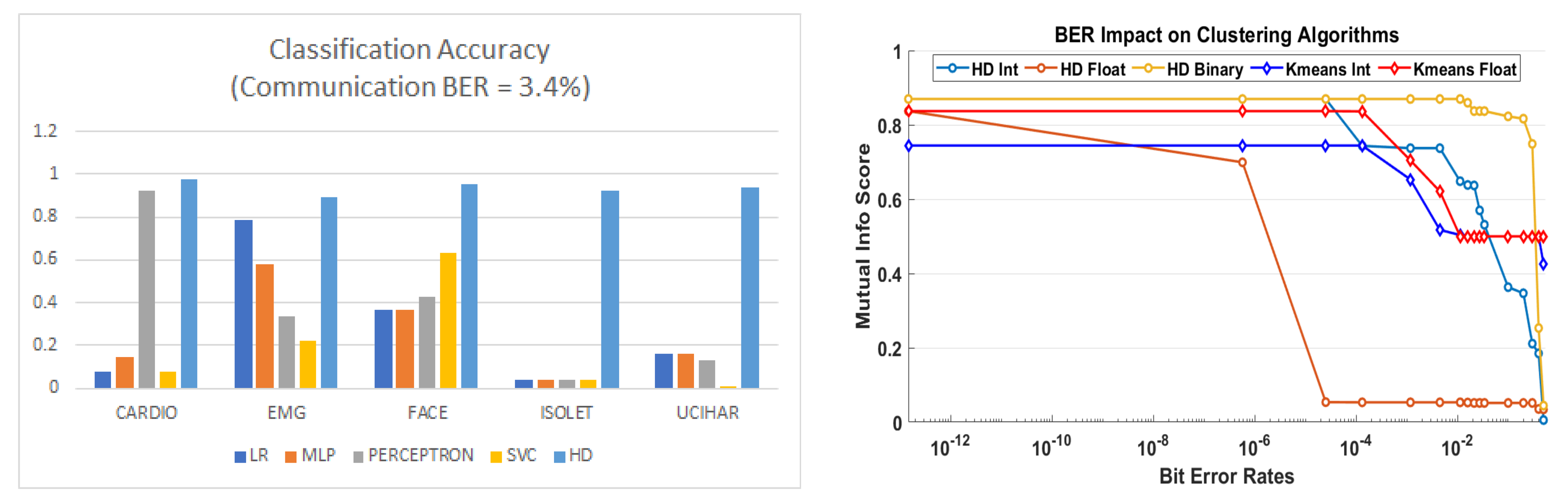
All comparisons are to HD CPU implementation

	Encoding		Training		Inference	
	Speedup	Energy Efficiency	Speedup	Energy Efficiency	Speedup	Energy Efficiency
GPU	194x	34x	33x	14x	117x	21x
FPGA	1,930x	68,000x	580x	1,070x	280x	922x
PIM	1,007,990x	594,523x	88,065x*	164,718x*	98,778x*	82,200,185x*

HD Clustering PIM is **855,000x faster & 2,700x more efficient**

HD Clustering	Encoding		Clustering		Encoding + Clustering	
	Speedup	Energy Efficiency	Speedup	Energy Efficiency	Speedup	Energy Efficiency
GPU	95x	15x	945x	226x	127x	21x
FPGA	141,800x	950,000x	55x	203x	196x	731x
ASIC-PIM	2,177,000x	69,000x	332,000x	763x	855,000x	2,700x

Accuracy of HD versus other ML Techniques with BER



HD Computing has applications to many areas in edge sensing and learning – from supervised classification and clustering that have large noise resilience, to applications in defense RF and secure edge computing.

Next steps are to manufacture HD PIM accelerator HW, design tools needed to map code onto the HD PIM and integrate it & test with the rest of the system.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).