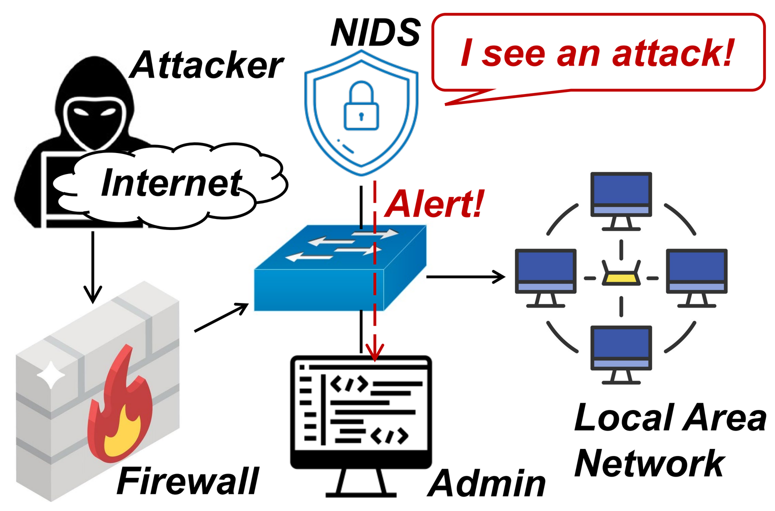
Late Breaking Results: Scalable and Efficient Hyperdimensional Computing for Network Intrusion Detection

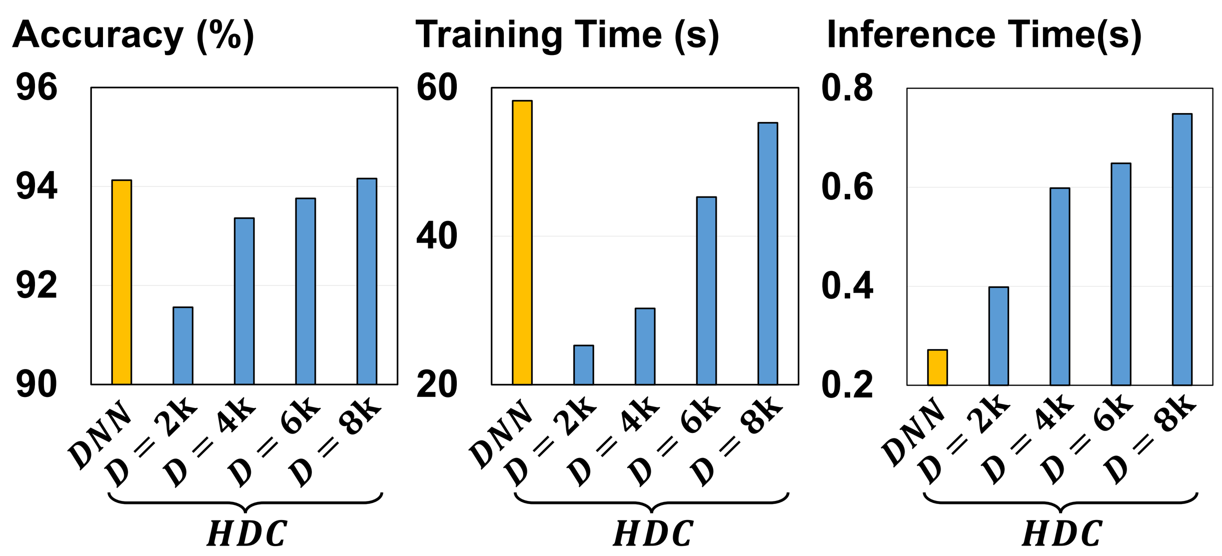
Junyao Wang, Hanning Chen, Mariam Issa, Sitao Huang, Mohsen Imani

junyaow4@uci.edu, University of California, Irvine, United States

Security on Edge Devices



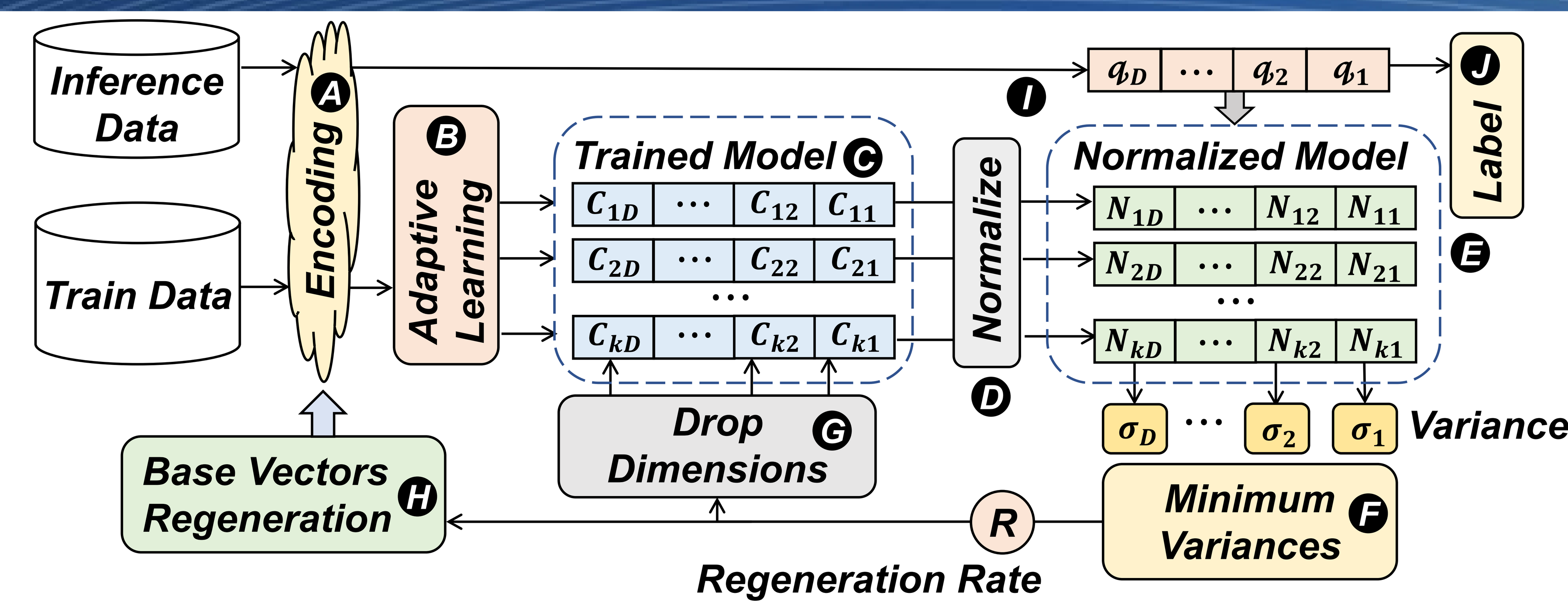
Network Intrusion Detection (NIDS)



Challenge 1:
Increasingly sophisticated deep learning models are developed for security attack detection
⇒ So Expensive!
❖ **Security on edge devices?**
❖ **Real-time attack detection?**
Soution1:
Hyperdimensional Computing (HDC)!

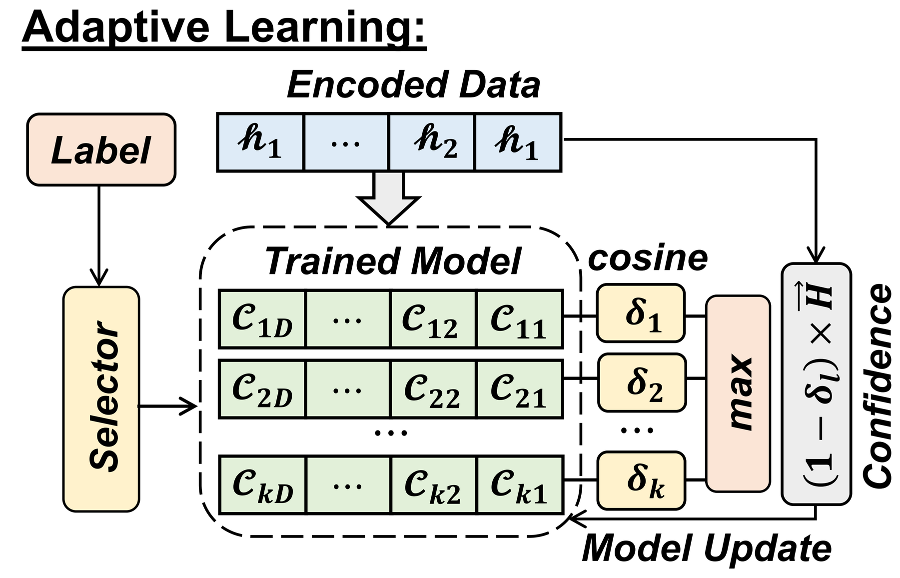
Challenge 2:
Existing HDC use static encoders
⇒ *Requires Extremely high dimensionality to achieve reasonable accuracy*
❖ ***Intensive memory & Computation***
❖ ***Huge latency in attack detection!***
Solution 2:
Dynamic Encoding!

Our Methodology



An Overview of Our Proposed HDC Framework for Network Intrusion Detection

Adaptive Learning:

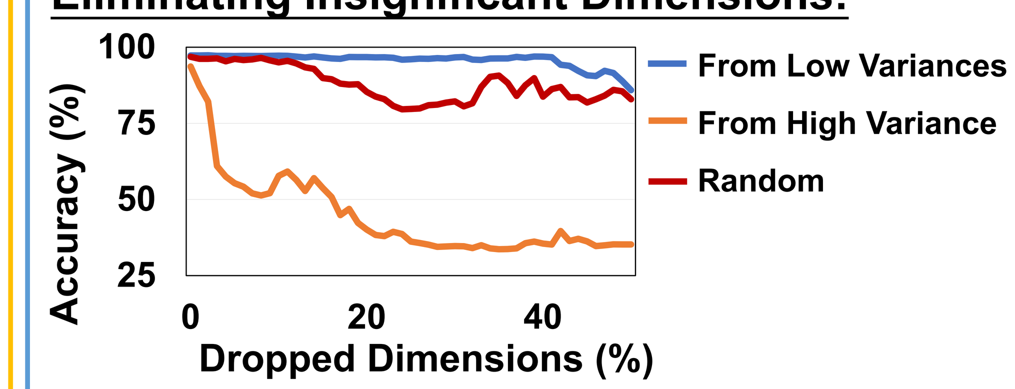


If $\mathcal{H}$ has maximum similarity with class ℓ' while its true label is ℓ:

$$\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell + \eta(1-\delta_l) \times \mathcal{H}$$
$$\mathcal{C}_{\ell'} \leftarrow \mathcal{C}_{\ell'} + \eta(1-\delta_{l'}) \times \mathcal{H}$$

✓ *Eliminate model saturation!*
✓ *Better capture uncommon data samples!*

Eliminating Insignificant Dimensions:



Dimensions with similar values over all classes store common info and thereby playing a minor role in classification! We can drop them!

Dimension Regeneration:

$\mathcal{F} = \{f_1, f_2, \dots, f_n\}\ (f_i \in \mathbb{R})$

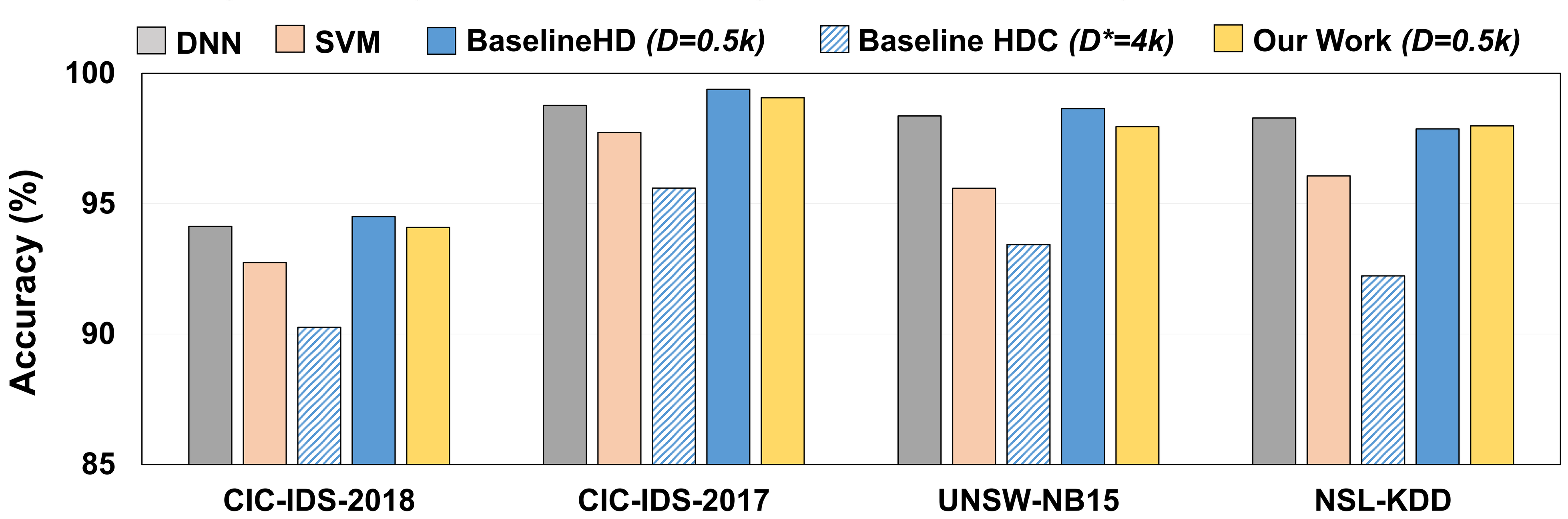
$\Rightarrow \mathcal{H} = \{h_1, h_2, \dots, h_D\}(0 \le h_i \le 1, h_i \in \mathbb{R})$.

$h_i = \cos(\mathcal{B}_i \cdot \mathcal{F} + c) \times \sin(\mathcal{B}_i \cdot \mathcal{F})$.

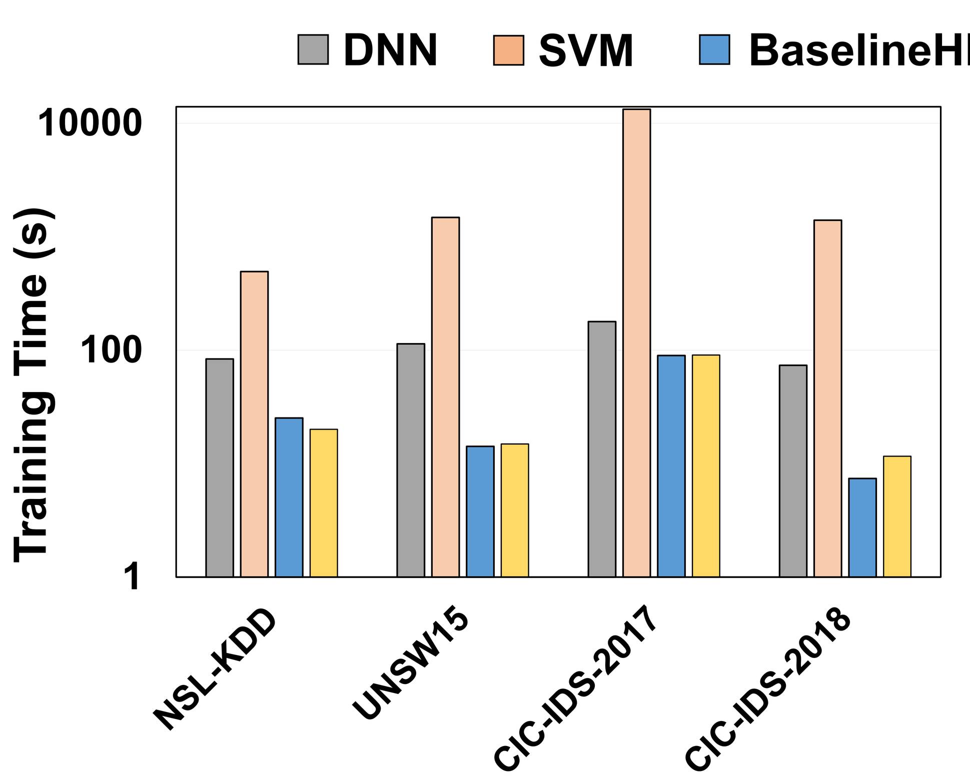
$\mathcal{B}_i = \{b_1, b_2, \dots, b_n\},\ b_i \sim \text{Gaussian}(0,1),\ c \sim \text{Uniform}[0, 2\pi]$

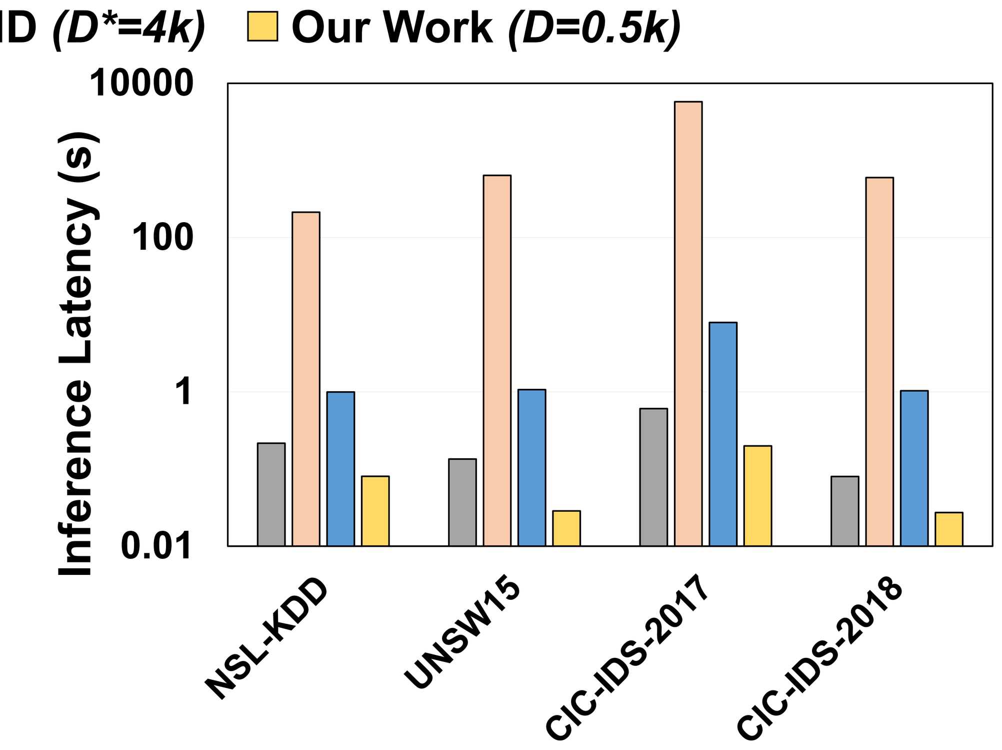
Evaluations

- **Accuracy**
  - ✓ Comparable accuracy to SOTA DNNs, 1.63% higher accuracy than SVMs.
  - ✓ Comparable accuracy to SOTA HDC but use 8.0× lower dimensionality.
  - ✓ 4.28% higher accuracy than SOTA HDC using the same dimensionality.



- **Efficiency**
  - ✓ Training: 2.47 × faster than SOTA DNNs, 1.85 × faster than SOTA HDC.
  - ✓ Inference: 15.29 × faster than SOTA HDC.





- **Robustness Against Hardware Failures**
  - ✓ 12.9 × higher robustness than SOTA DNNs
  - ✓ Maximized robustness at 1-bit precision

Robustness Against Hardware Noise

	Hardware Error	*1.0%*	*2.0%*	*5.0%*	*10.0%*	*15.0%*
	DNN	3.9%	10.7%	17.8%	32.1%	41.2%
Our Work	*1 bit*	0.0%	0.0%	1.0%	3.1%	4.1%
Our Work	*2 bits*	1.9%	2.3%	4.5%	7.9%	10.4%
Our Work	*4 bits*	2.3%	4.7%	8.4%	13.1%	17.3%
Our Work	*8 bits*	3.6%	7.9%	13.7%	18.3%	22.9%

- **Cross Platform Evaluation**
  - ✓ CPUs demonstrate more strength for high bitwidth data
  - ✓ FPGA shows excellent energy efficient improvement compared to CPU

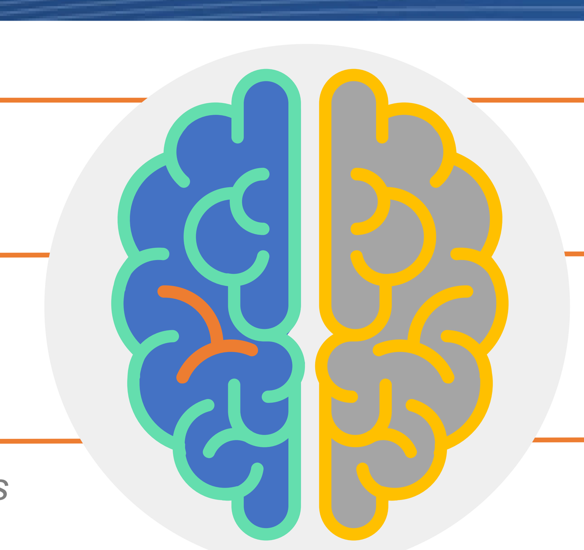
	*D**	*CPU*	*FPGA*
32 bits	1.2k	6.6×	16×
16 bits	2.1k	4.0×	24×
8 bits	3.6k	2.4×	34×
4 bits	5.6k	1.5×	31×
2 bits	7.5k	1.2×	28×
1 bit	8.8k	**1.0×**	28×

HDC Introduction

Cerebellum
Sparse high dimensional representations

Robustness against noise
Works well with multiple noisy input and computation

Efficient
The brain works at as low as 20W of energy



High-dimensional
Our basic elements are hypervectors (~thousands)

Holographic encoding
Info of every feature is on all the dimensions of hypervectors

HDC Algebra
Simple and fast, very efficient computation.

- **Binding (+):** Element-wise addition, i.e., $\mathcal{H}_{bundle} = \mathcal{H}_1 + \mathcal{H}_2$, $\delta(\mathcal{H}_{bundle}, \mathcal{H}_1) \gg 0, \delta(\mathcal{H}_{bundle}, \mathcal{H}_3) \approx 0$
- **Bundling:** Element-wise multiplication, i.e., $\mathcal{H}_{bind} = \mathcal{H}_1 * \mathcal{H}_2, \delta(\mathcal{H}_{bind}, \mathcal{H}_1) \approx 0, \delta(\mathcal{H}_{bind}, \mathcal{H}_2) \approx 0$
- **Reasoning:** measuring the similarity of hypervectors, e.g., cosine similarity $\delta(\mathcal{H}_1, \mathcal{H}_2) = \frac{\mathcal{H}_1 \cdot \mathcal{H}_2}{\|\mathcal{H}_1\| \cdot \|\mathcal{H}_2\|}$

Conclusion

This work:
- ✓ The first time for the dynamic HDC learning technique being applied in cyber security
- ✓ Identify insignificant dimensions to reduce unnecessary high-dimensional computations
- ✓ 2.47 × faster training and 15.29 × faster inference than SOTA learning methods

Future work:
❖ ***Challenge 1: The Accuracy of HDC***
- Currently still lower than deep learning methodologies in many cases

❖ **Challenge 2: Explanability of HDC**
- Understanding HDC from the theoretical perspective is currently very limited

Selected Reference

1. Junyao Wang, et. al. DistHD: A Learner-Aware Encoding Method for Hyperdimensional Classification, the 60th Annual Design Automation Conference 2023.
2. Junyao Wang, et. al. Late Breaking Result: Scalable and Efficient Hyperdimensional Computing for Network Intrusion Detection, the 60th Annual Design Automation Conference 2023.