Privacy-Preserving Neural Representation for Brain-Inspired Learning

Javier Roberto Rubalcava-Cortés[1], Alejandro Hernandez-Cano[2], Alejandra Citlalli Pacheco-Tovar[1], Farhad Imani[3]
Rosario Cammarota[4], Mohsen Imani[5⋆]

[1]Universidad Nacional Autonóma de México, [2]École Polytechnique Fédérale de Lausanne,
[3]University of Connecticut, [4]Intel Labs, [5]University of California Irvine

⋆Corresponding Author: m.imani@uci.edu

***Abstract*—In this paper, we propose BIPOD, a brain-inspired privacy-oriented machine learning. Our method rethinks privacy-preserving mechanisms by looking at how the human brain provides effective privacy with minimal cost. BIPOD exploits hyperdimensional computing (HDC) as a neurally-inspired computational model. HDC is motivated by the observation that the human brain operates on high-dimensional data representations. In HDC, objects are thereby encoded with high-dimensional vectors, called *hypervectors*, which have thousands of elements. BIPOD exploits this encoding as a holographic projection with both cryptographic and randomization-based features. BIPOD encoding is performed using a set of brain keys that are generated randomly. Therefore, attackers cannot get encoded data without accessing the encoding keys. In addition, revealing the encoding keys does not directly translate to information loss. We enhance BIPOD encoding method to mathematically create perturbation on encoded neural patterns to ensure a limited amount of information can be extracted from the encoded data. Since BIPOD encoding is a part of the learning process, thus can be optimized together to provide the best trade-off between accuracy, privacy, and efficiency. Our evaluation on a wide range of applications shows that BIPOD privacy-preserving techniques result in 11.3× higher information privacy with no loss in classification accuracy. In addition, at the same quality of learning, BIPOD provides significantly higher information privacy compared to state-of-the-art privacy-preserving techniques.**

I. Introduction

Advances in deep learning have led to breakthroughs in analyzing large-scale data produced in the Internet of Things (IoT). Many internet companies collect users' online activities to train learning and recommendation algorithms to predict their future interest [1], [2]. For example, collecting large-scale health data can be used to produce new diagnostic models. Similarly, collecting financial information, such as payment network, history, merchant data, and account holder information, can be used to train an accurate model for fraud detection. Although recent platforms provide higher computational efficiency to process deep learning and large-scale data, collecting and combining data from different sources remains very challenging. Particularly, privacy concerns prevent many users or organizations from sharing their data [3].

Privacy-preserving machine learning provides a promising solution by allowing different organizations to share their data [4]–[6]. Privacy-preserving learning protocols could be classified into two categories [6]: (1) secure multi-party computation that employs cryptographic tools to protect privacy among involved parties. This approach often brings huge extra computational overhead that cannot be supported by most practical systems or organizations. (2) perturbation and randomization-based approaches that sanitize samples prior to their release [7]. Although this approach provides limited privacy, it makes a controllable trade-off between accuracy and privacy. The drawback of this approach is that perturbation and learning algorithms are not well integrated. This results in a high-quality loss for small privacy preservation.

This paper aims to develop an ultra-lightweight privacy-preserving machine learning that can be used for many practical systems. Our method rethinks privacy-preserving mechanisms by looking at how the human brain provides effective privacy with minimal cost [8], [9]. Unlike today's computing systems, the brain does not store information in a human interpretable format. Our observed information from our sensors (vision, hearing, smell, touch, and taste) stores as a pattern of neural activity in the brain. Particularly, the Cerebellum cortex in the brain is responsible for our short/long-term memorization [10], [11]. In the cerebellum, the information is stored in holographic and high-dimensional space. More interestingly, these neural patterns are different from person to person [12]. Even when observing the same scene, the brain of two individuals will get entirely different neural representations.

In this paper, we propose BIPOD, a brain-inspired privacy-preserving mechanism. BIPOD exploits hyperdimensional computing (HDC) as a neurally-inspired computational model mimicking brain properties [10], [13]–[17]. There are few recent efforts tried to enhance HDC privacy. Work in [18] exploited MPC protocol to enable secure collaboration learning with the assumption that HDC encoding is a cryptography method. Work in [19], [20] used existing privacy-preserving mechanisms introduced for DNN to secure HDC data from a privacy perspective. However, since these methods are not well suited for HDC, a small privacy gain is obtained with a significant quality drop. Unlike the existing privacy-preserving method, BIPOD integrates learning and privacy by enabling computation over neural representation. Our solution modifies the HDC encoding to give inherent privacy-preserving features in high-dimensional while ensuring maximum quality of learning.

- HDC is motivated by the observation that the human brain operates on high-dimensional data representations. In HDC, objects are thereby encoded with high-dimensional vectors, called *hypervectors*, which have thousands of elements [21]. BIPOD exploits this encoding as a holographic projection with certain privacy-preserving features.
- BIPOD encoding naturally has both cryptographic and randomization features. For cryptography, the encoding is performed using a set of brain keys that are generated randomly for different users. Thus, attackers cannot get encoded data

978-3-9819263-7-8/DATE23/© 2023 EDAA

without accessing the encoding keys. We develop a novel data recovery approach showing the vulnerability of prior methods to information attacks.
- Revealing the encoding keys does not directly translate to information loss. We enhance BIPOD encoding to mathematically create perturbation on encoded neural patterns. We introduce encoding variance and hyper-latent quantization as two effective techniques to enhance BIPOD encoding privacy. These methods limit the amount of information extracted from the encoded data, thus preserving privacy.
- Unlike existing privacy-preserving techniques that are not optimized for HDC, we enhance HDC encoding for highly accurate and efficient brain-inspired learning. Our techniques are designed to mathematically randomize data decoding from high-dimensional space while providing minimal overhead on learning accuracy. Our privacy-preserving method is a part of data encoding, thus having negligible cost on learning performance.

We evaluate BIPOD effectiveness on a wide range of applications. Our evaluation shows that BIPOD privacy-preserving techniques result in 11.3× higher information privacy with no loss in classification accuracy. In addition, at the same quality of learning, BIPOD provides significantly higher information privacy compared to state-of state-of-the-art privacy-preserving techniques.

II. Related work

Prior research applied the idea of hyperdimensional computing to diverse cognitive tasks, such as robotics, analogy-based reasoning, latent semantic analysis, language recognition, prediction from multimodal sensor fusion, and bio-signal processing [22]–[24]. Prior work also focused on the security and privacy of hyperdimensional computing. Work in [18], [25], [26] designed a framework for security collaborative learning using multi-party computation (MPC). The MPC assigns a personal key for each user, which cannot be accessed by other users of the cloud. However, this has the strong assumption that keys can stay secure. However, the keys can be revealed using a frequency attack or when a user collides with the cloud in practice. Work in [9], [27], develop a novel adversarial attack on an HDC-based classifier. Our solution is orthogonal to this paper, as our focus is on privacy-preserving rather than adversarial attacks. Work in [19] exploited conventional differential preserving mechanisms (i.e., noise injection) to ensure high-dimensional data is private. However, this method is not well developed for HDC; thus (1) results in a significant drop in classification accuracy, (2) it is only applicable to old linear encoding methods. (3) the privacy shown in the paper comes from the weak data recovery mechanism used.

In contrast, in this paper, we develop a novel approach that provides a theoretical trade-off between accuracy and privacy in hyperdimensional space. Our solution redesigns the state-of-the-art HDC encoding methods to mathematically ensure data privacy even when the encoding keys are revealed. Unlike prior work, our solution minimizes information leakage with no major quality loss.

III. Hyperdimensional computing

Brain-inspired Hyperdimensional Computing (HDC) is a neurally-inspired model of computation based on the observation that the human brain operates on high-dimensional and distributed representations of data [10]. The fundamental units of computation in HDC are high-dimensional data or "hypervectors" which are constructed from raw signals using an encoding procedure. HDC uses an encoding module to transform data into high-dimensional representation. The encoding leverages randomly generated hypervectors [21]. During the learning process, HDC can apply brain-like operations over encoded data. These operations include memorization, association, as well as similarity search. During memorization, HDC superimposes together the encodings of signal values to create a composite representation of a phenomenon of interest known as a "class hypervector". In inference, the associative search identifies an appropriate class for the encoded query hypervector.

A. Hyperdimensional Encoding

Encoding or transforming data into high-dimensional space is the first step of hyperdimensional computing. HDC encoding spreads the data over a very large hypervector, thus a substantial number of bits can be corrupted while preserving sufficient information. HDC encoding depends on the data type [21], [28].

We propose a novel encoding method inspired by the kernel trick to map data points into the high-dimensional space. The underlying idea of the kernel trick is that data, which is not linearly separable in original dimensions, might be linearly separable in higher dimensions. Let us consider certain functions $K(x, y)$ which are equivalent to the dot product in a different space, such that $K(x, y) = \Phi(x) \cdot \Phi(y)$, where $\Phi(\cdot)$ is often a function for high or infinite dimensional projection. The Radial Basis Function (RBF) is the most popular kernel, and previous HD computing encoders has been proposed to approximate it [29], giving a high dimensional projection $\varphi(\cdot)$ such that $\Phi(x) \cdot \Phi(y) \approx \varphi(x) \cdot \varphi(y)$. The high dimensional projection $\varphi(\cdot)$ to approximate the RBF kernel was written as a cosine function of a random affine transformation. In our case, φ is expressed as a $\tanh$ activation of a random linear map instead. This allows us to keep a non-linear transformation for a rich HDC representation while maintaining a monotonous and bijective relationship between the input and the hyperdimensional representation.

Let us assume a feature vector in original space $\mathbf{x} \in R^M$ is an input data. The encoding module maps this vector into high-dimensional vector, $\mathbf{h} \in \mathcal{R}^D$, where $D \gg M$. The encoding method that maps input vector into high-dimensional space is given by: $\mathbf{h} = \tanh(\mathbf{Bx})$. In other words, our encoding consists of a random projection of an input vector $\mathbf{x}$ with a projection matrix $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_D)^T \in \mathbb{R}^{D \times M}$, followed by a non-linear activation function.

In our encoding, $\mathbf{b}_i \in \mathbb{R}^M$ is randomly independent vector sampled from a normally distribution, $\mathbf{b}_i \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma^2}{\sqrt{M}}\mathbf{I}\right)$, known as the encoding basis or *keys*. Because of the random nature, the basis are nearly-orthogonal, $\delta(\mathbf{b}_j, \mathbf{b}_{j'}) \approx 0$, where δ denotes cosine similarity.

There are three factors that affect the quality of encoding:
- Dimensionality (D): increase results in a higher redundancy and robustness.
- Encoding Variance (σ^2): controls the distribution and variance of the representation.

- Activation Function ($\mathsf{tanh}(\cdot)$): sets the non-linearity of the representation in high-dimension.

In the rest of the paper, we discuss how these parameters can trade learning accuracy and information privacy.

B. Hyperdimensional Learning

In order to use HDC for classification problems, let us assume we have a data set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^M$ and $y_n \in \{1, \dots, K\}$. Our plan is to build a classifier to distinguish between K different classes. The first step needed is to map the original data to HD space, $\mathbf{h}_n = \mathsf{tanh}(\mathbf{B}\mathbf{x}) \in \mathbb{R}^D$, and introduce K different D-dimensional vectors, $\mathbf{w}_k \in \mathbb{R}^K$, known as the class hypervectors, initialized by adding hypervectors of the same class together:

$$\mathbf{w}_k = \sum_{n=1}^{N} \mathbf{h}_n 1\{k = y_n\}$$

where $1\{k = y_n\} = 1$ whenever $k = y_n$ and 0 otherwise. Using HDC terminology, this step corresponds to the "memorization" operator [10].

In order to make label inferences from a new data point $\mathbf{x}$, we compare the cosine similarity of its encoded hypervector, with all class hypervectors, that is, similarity search:

$$\hat{y} = \underset{1 \le k \le K}{\arg\max}\, \delta(\mathbf{w}_k, \mathsf{tanh}(\mathbf{B}\mathbf{x}))$$

Using the HDC iterative training, we further optimize the class hypervectors adaptively. This is done iterating batches of the data set, and updating the class hypervectors whenever the prediction $\hat{y}_n$ doesn't match the true label y_n.

$$\mathbf{w}_{y_n} \leftarrow \mathbf{w}_{y_n} + (1 - \alpha_{n,y_n})\mathbf{h}_n$$
$$\mathbf{w}_{\hat{y}_n} \leftarrow \mathbf{w}_{\hat{y}_n} - (1 - \alpha_{n,\hat{y}_n})\mathbf{h}_n$$

where $\alpha_{n,k} = \delta(\mathbf{w}_k, \mathbf{h}_n)$ is the cosine similarity of data point n with class label k.

IV. Privacy-Preserving Hyperdimensional Computing

In HDC, it is also possible to decode HDC representation back to the original space, given the encoding key. This data invertibility can be used to design interpretable HDC models. However, this can also result in privacy challenges, as the encoded information can be accessed by unauthorized users. In the following, we further explain the details of our HDC encoding module and how we can enhance it from a privacy-preserving perspective.

A. Data Recovery in Hyperspace

To obtain the information of encoded data, the attacker needs to decode the high-dimensional data back to the original space. This data decoding is very difficult when the attacker does not have knowledge about the encoding procedure and the encoding key [18]. Our goal is to make data recovery from potential attacks harder while providing minimal impact on the quality of learning.

We observe the encoded data in high-dimensional space $\mathbf{x}$ and also hold a copy of the transformation matrix $\mathbf{B}$. The problem can be stated as finding the closest vector $\mathbf{x}'$ in terms of L_2 norm:

$$\underset{\mathbf{x}' \in \mathbb{R}^M}{\arg\min} \|\mathbf{x} - \mathbf{x}'\|$$

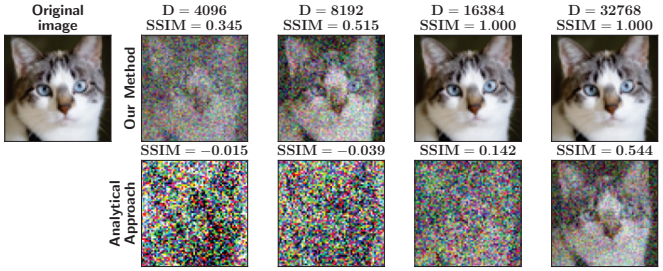

Fig. 1. Comparison of our data recovery method and previous work. On top: Data recovery of our proposed method of an image of a cat. Bottom: Previous method of data recovery for the same image.

From a theoretical perspective, we can recover $\mathbf{x}$ without error under mild assumptions, using the fact that tanh is invertible, the problem is restated as a least-squares solution:

$$\underset{\mathbf{x}' \in \mathbb{R}^M}{\arg\min} \|\mathbf{z} - \mathbf{B}\mathbf{x}'\|^2 \qquad \text{where } \mathbf{z} = \mathsf{arctanh}(\mathbf{h}) \tag{1}$$

Because $\mathbf{z} = \mathbf{B} \cdot \mathbf{x}$, then $\mathbf{x}'$ will satisfy the lowest error $\|\mathbf{x} - \mathbf{x}'\|$. As long as the problem is well-posed, we can even guarantee the equality $\mathbf{x} = \mathbf{x}'$.

We contrast this method with state-of-the-art solutions that exploits analytical solution for data recovery [18]. In this work, the data recovery approach is proposed in a similar fashion, where the attacker possesses the encoded data $\mathbf{h}$ and encoding keys $\mathbf{B}$, but their encoding module was linear. Our method provides:

1) The lowest error recovery in terms of L_2 metric. Figure 1 compares the quality of data decoding of our approach with state-of-the-art analytical solution [18]. The results are reported when the hypervector dimensionality is changing from $D = 4k$ to $D = 32k$. Our evaluation shows that encoding to higher dimensions increases the chance of accurate data recovery. Figure 1 shows that our method can decode an image with a 100% recovery rate using hypervectors with over $D = 16k$ dimensions. However, in practice, the dimensionality required might be even lower depending on the dimensionality of the input data (M). Figure 1 also compares the quality of our data decoding with the analytical solution. Our results indicate that our method has a significantly higher quality of data recovery.
2) Flexibility to expand to other activation functions φ, where $\mathbf{h} = \varphi(\mathbf{B}\mathbf{x})$, as long as φ is invertible: $\mathbf{z} = \varphi^{-1}(\mathbf{h})$. Note that linear encoder is the special case when we have no activation function, and thus our method is suitable to use in the linear encoder too.

B. Computational Challenges in Decoding

Computers often work with floating-point representation with a fixed number of bits. Therefore, computing arctanh precisely is not always an option. If an input to the function, $|x|$, is large enough, $\mathsf{arctanh}(x)$ will yield noisy inverse values. The figure 2 plots 'tanh' and 'arctanh $\circ$ tanh' for different values of bits used in the representation of $\mathsf{tanh}(x)$. We observe that, with lower bits used in the representation, the function is less accurate.

In fact, if we take $|x|$ large enough, $\mathsf{tanh}(x)$ will be equal to ± 1, even though this should not be possible. This is because the precision is not enough to compute $\mathsf{tanh}(x)$. In this cases, $\mathsf{arctanh}(\pm 1)$, is not defined. As Figure 2 shows, this

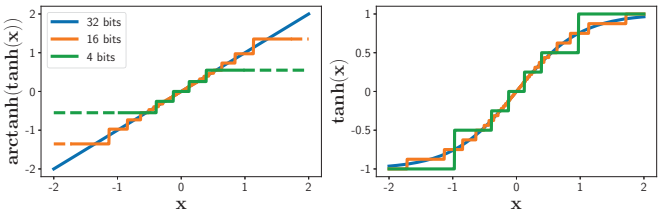

Fig. 2. Impact of quantization for tanh. Left: graph of arctanhtanh$x$, dashed lines show values where $\tanh(x) = \pm 1$ and its inverse goes to $\pm\infty$. Right: graph of $\tanh x$. Color: number of bits used to represent value x.

corresponds to x values where the line is dashed. Note that this effect happens regardless of the bits used, as long as it is fixed during computation. The difference lies in how far away from zero we can go. For instance, as show in Figure 2, with 32 bits we can invert correctly values for x up to 2.0, whereas with 4 bits making $|x| > 1$ will already yield incorrect values.

In order to provide more numerical stability in this cases, we provide a smart threshold correction procedure (STC), for $\varepsilon > 0$:

$$\mathrm{STC}_\varepsilon(x) = \begin{cases} x & -1 < x < 1 \\ 1 - \varepsilon & x \geq 1 \\ -1 + \varepsilon & x \leq -1 \end{cases}$$

And thus, $\mathbf{z}$ in Equation (1) is replaced to $\mathbf{z} = \mathsf{arctanh}(\mathrm{STC}_\varepsilon(\mathbf{h}))$.

C. Encoding Variance

Encoding variance (σ^2), or variance of generated base hypervectors ($\mathbf{B}$), directly affects the accuracy and privacy of the HDC encoding module. This variance determines a range of values that projected data would get, $\mathbf{B} \cdot \mathbf{x}$. This projected data is an input to our activation function, tanh. Therefore, its range determines the non-linearity that we can get from our encoding module.

As σ^2 goes to ∞, our encoding converges to a sign function:

$$\tanh(\mathbf{Bx}) \sim \mathrm{sign}(\mathbf{Bx})$$

This means that the learning capabilities of the HDC module will be weakened. This occurs because the encoded vector will be condensed in the discrete $\{-1, 1\}^D$ space.

On the other hand, for small values of variance ($\sigma^2 \ll 1$), the projected data will get smaller values ($|\mathbf{B} \cdot \mathbf{x}| \ll 1$). Therefore, the non-linear effect of tanh will not contribute to our encoding process. In practice, when the input data $\mathbf{x}$ is normalized, the variance equal to $\sigma^2 = 1$, generally achieves the best performance.

From a privacy perspective, larger values of σ^2 will end up causing more problems to invert because of numerical limitations, as explained in section IV-B. This means that increasing the encoding variance gives a trade-off between the accuracy obtainable by the learning system and the capability of the attacker to perform data recovery successfully, which is the privacy concern in this work.

D. Hyper-latent Quantization

We exploit the numerical issues discussed in section IV-B to introduce a new hyparameter $q \in \mathbb{N}^+$, the "*hyper-latent quantization factor*". We use this number to quantize the hypervector elements after the activation function. That is, the new hyperdimensional encoding $\mathbf{h}'$ of $\mathbf{h}$ will be:

$$\mathbf{h}' = \mathrm{quantize}(\mathbf{h}, q)$$

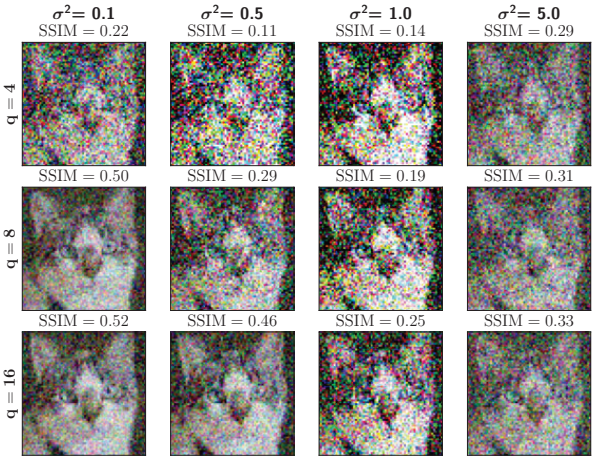

Fig. 3. Impact of quantization (q) and encoding variance (σ^2) in data recovery.

Note that the original hypervector before applying quantization will never be released, and the attacker will not be able to access that information. So the HDC learning module will also work with less precise data. This creates another trade-off between data privacy and accuracy.

Hyper-latent quantization and tuning the encoding variance can be jointly optimized to achieve the best possible accuracy values while maintaining the data as private as possible. Figure 3 shows an example of the effects of the number of bits used in the representation (quantization, q) and σ^2 during data recovery. The maximum recovery is provided for smaller values of σ^2, and the quality of the decoding decreases as σ^2 grows larger. Increasing the precision q makes the data recovery process more reliable regardless of the value of σ^2.

E. Privacy Metrics

Evaluating the success of our approach requires assessing whether the recovered data exposes private information. We evaluated the privacy risk quantitatively with multiple metrics to gather evidence of how well BIPOD protects data from multiple sources.

- **Mean Squared Error (MSE):** Error-based metrics quantify the error an attacker makes in creating his estimate. MSE is commonly used to evaluate regression problems. In statistical parameter estimations, a common goal is to minimize the mean squared error. As a privacy metric, the MSE describes the error between reconstructions $\hat{y}$ by the attacker and the true data y.

$$priv_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- **Structural Similarity Index Measure (SSIM)** MSE approach estimates absolute errors; that is, it does not represent how brains perceive similarity, so we also employed the perception-based Structural Similarity Index Measure to evaluate quality.

$$priv_{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_\mu)(2\sigma_{xy} + c_\sigma)}{(\mu_x^2 + \mu_y^2 + c_\mu)(\sigma_x^2 + \sigma_y^2 + c_\sigma)}$$

where x and y are two representations being compared, μ_* and σ_* represents the mean value and standard deviation, respectively, $c_\mu = (k_\mu L)^2$ and $c_\sigma = (k_\sigma L)^2$ are constants to control instability and L a dynamic range.

TABLE I
LIST OF DATASETS USED (N: TOTAL NUMBER OF DATA POINTS, M: NUMBER OF FEATURES, K: NUMBER OF CLASSES).

	N	M	K	Description	Reference
PHONE	10929	561	6	Activity recognition (Smartphone)	[31]
ISOLET	7797	617	26	Voice Recognition	[32]
MNIST	70000	784	10	Handwritten digits	[33]
HAR	10299	561	6	Activity recognition(Mobile)	[34]
SOFTWARE	1109	21	2	PC1 Software defect prediction	[35]
MICRO-MASS	360	1300	10	Microorganisms identification.	[36]
OPTDIGITS	5620	106	10	Optical Recognition of Handwritten Digits	[37]

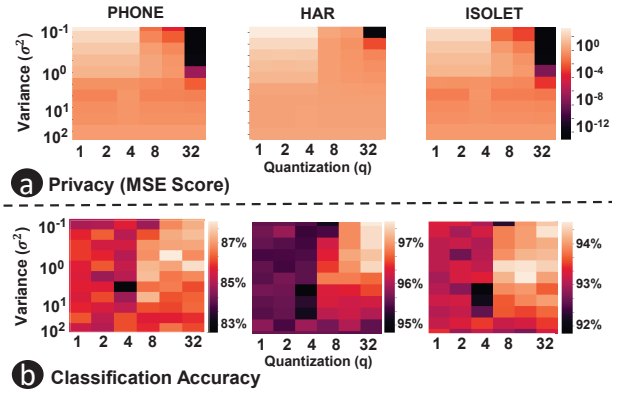

Fig. 4. Trade-off between accuracy and privacy for different values of encoding variance, and hyper-latent quantization.

V. EVALUATIONS

We verified BIPOD functionality using Python implementation. Our implementation is built from the open-source library Pytorch [30], which allows to run in both CPU and GPU, achieving peak performance on both platforms. We evaluate the effectiveness of BIPOD on a wide range of classification applications, listed in Table I.

A. Accuracy and Privacy Trade-off

Figure 4 shows the trade-off between privacy and classification accuracy using different encoding variances and hyper-latent quantization. The results are reported for three applications. First, we observe that increasing encoding variance is an effective mechanism to prevent potential data attacks. However, this comes at the cost of lower quality of classification as tanh non-linearity in BIPOD encoder is not well utilized. The trade-off in the encoding variance makes the data recovery significantly harder. However, if the data is linearly separable, the variance would have almost no impact on the prediction quality.

We also observe a similar trade-off for hyper-latent quantization, where quantization defines as the number of bits used in the numeric representation. For small q values, the data reconstruction is less effective as the data is protected. This comes with a minor overhead on the classification accuracy. However, as q increases, the data recovery rate will succeed even if that does not significantly boost the classification accuracy. Our evaluation shows that BIPOD can provide the best trade-off between accuracy and privacy using the encoding variance equal to 10 and $q = 8$ quantization.

B. Impact of Encoding

Figure 5 compares BIPOD quality of data recovery with state-of-the-art HDC encoding methods: a projection-based encoding (Uniform) [18], a bipolar seed-based encoding (Bipolar) [28]. The "Uniform" encoder uses random uniform bases

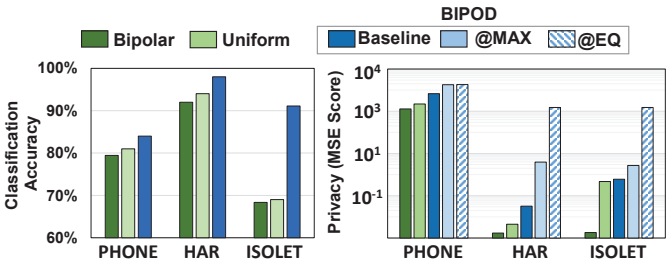

Fig. 5. Comparison of BIPOD with state-of-the-art HDC encoding methods in terms of accuracy and privacy.

and no activation function, and the "bipolar" encoder uses discrete random bases in $\{-1, +1\}^M$ to represent objects and positions. For BIPOD, the results are reported for three configurations: (1) **BIPOD-BASELINE:** without our optimizations, (2) **BIPOD-MAX:** optimized BIPOD that provides maximum accuracy to our baseline method, and (3) **BIPOD-EQ:** optimized BIPOD providing the same accuracy as the best prior encoding method. Our results show that BIPOD with no optimization can still provide higher classification accuracy and privacy level than the existing encoding method. We observe that our BIPOD provides maximum classification accuracy and privacy using tanh encoder. The tanh non-linearity provides more opportunities to separate non-linear data while at the same time serving as an initial barrier to making data recovery harder. BIPOD-MAX configuration can further enhance the privacy level by 11.3× with no quality loss. Finally, BIPOD-EQ improves the privacy level by 340.1× while ensuring the accuracy is the same as state-of-the-art HDC methods. This is achieved by setting quantization and variance values to ensure maximum privacy enhancement for a small loss in classification accuracy.

C. Dimensionality & Privacy-Accuracy

Figure 6 compares the effectiveness of BIPOD quality of classification and data privacy using different dimension sizes (D). The graph reports normalized privacy values. Our result indicates that dimensionality affects both classification accuracy and data decoding rate. Using the default value of $\sigma^2 = 1$ and no quantization, we can reconstruct original data from the encoded hypervector accurately. In other words, with no optimization, our solution lacks privacy. Using low dimensional hypervectors, our solution gets higher privacy at the penalty of low classification accuracy. However, dimensionality has a more significant impact on improving data privacy. Hypervectors in lower dimensions still have enough redundancy to enable learning and memorization. However, the data decoding only depends on a single hypervector; thus, reducing dimensionality directly affects the reconstruction quality. In parallel to dimension, optimizing the variance and quantization yields a complex representation to decode but has minimal cost on classification accuracy. Our evaluation shows that reducing dimensionality from $D = 10k$ to $D = 1k$ results in a 2.9× improvement on data privacy while only having about 3% effect on the classification accuracy.

D. BIPOD & Data Recovery

Table II compares BIPOD data recovery method with the state-of-the-art analytical approach [18], [19]. As discussed

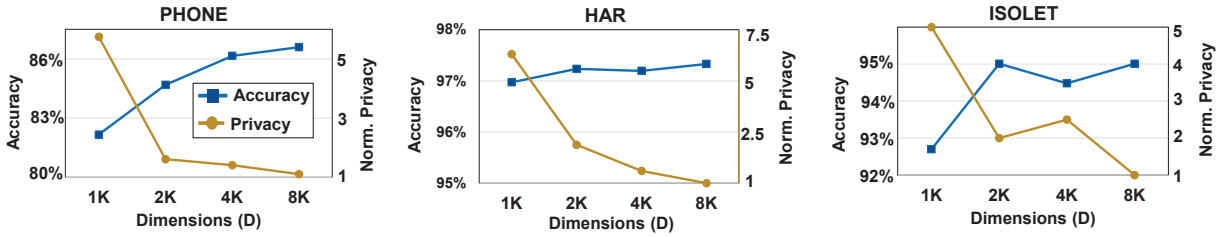

Fig. 6. Impact of dimensionality on the efficiency of BIPOD classification and data recovery for different data sets.

TABLE II
DATA RECOVERY USING ANALYTICAL AND PROPOSED BIPOD.

Dimensions	PHONE 1k	PHONE 2k	PHONE 4k	PHONE 8k	HAR 1k	HAR 2k	HAR 4k	HAR 8k
Analytical [19]	23%	28%	33%	41%	36%	42%	57%	60%
BIPOD	79%	86%	98%	100%	91%	96%	99%	100%

TABLE III
PRIVACY SCORE (MSE) AT DIFFERENT LEVELS OF ACCURACY FOR BIPOD AND PRIVE-HD [19].

Accuracy	85%	87%	89%	90%	91%
Prive-HD [19]	2.19	2.03	2.02	unreached	unreached
BIPOD	1506	1506	52.21	2.51	1.26

in section IV-A, our least-squares formulation provides more precise data reconstruction. The difference is intensified when the data dimensionality is low because the analytical approach relies on the orthogonality of random vectors that go near zero in high dimensional. However, BIPOD does not make such simplifications, and instead, the challenges in recovery come from having precise enough numerical representation. Thus, even for small dimensions, our approach will be more realistic simulation of potential attackers in our privacy-oriented HDC system. For example, our evaluation shows that our data recovery can provide 100% data recovery using $D = 8k$ hypervector size, while the reconstruction rate of the analytical approach does not exceed 60%.

E. Comparison with State-of-the-art

We also compare the effectiveness of our data privacy techniques with state-of-the-art privacy techniques used for HDC. Work in [19] exploited a differential privacy technique, defined for neural network, to secure the HDC model from a privacy perspective. In particular, it used noise injection to encode data as a solution to make data recovery less effective while having minimal impact on accuracy. Table III shows the privacy (MSE score) of BIPOD and the state-of-the-art approach provided when ensuring the same level of classification accuracy. Our evaluation shows that for a fixed level of accuracy, BIPOD provides significantly higher privacy compared to the state-of-the-art method. For example, as Table III indicates, existing approaches are not capable of providing a high quality of learning while ensuring privacy.

VI. CONCLUSION

In this paper, we propose brain-inspired privacy-oriented machine learning. Our method rethinks privacy-preserving mechanisms by looking at how the human brain provides effective privacy with minimal cost. BIPOD exploits hyperdimensional computing (HDC) as a neurally-inspired computational model. BIPOD exploits this encoding as a holographic projection with both cryptographic and randomization-based features. BIPOD encoding is performed using a set of brain keys that are generated randomly. Therefore, attackers cannot get encoded data without accessing the encoding keys.

ACKNOWLEDGEMENTS

This work was supported in part by National Science Foundation #2127780, Semiconductor Research Corporation (SRC) AI Hardware and Hardware Security, Department of the Navy, Office of Naval Research, grants #N00014-21-1-2225 and #N00014-22-1-2067, the Air Force Office of Scientific Research under award #FA9550-22-1-0253, and a generous gift from Cisco.

REFERENCES

[1] J. Yuan *et al.*, "Solving cold-start problem in large-scale recommendation engines: A deep learning approach," in *IEEE Big Data*, pp. 1901–1910, IEEE, 2016.
[2] Y. Zheng *et al.*, "Deep cnn-assisted personalized recommendation over big data for mobile wireless networks," *Wireless Communications and Mobile Computing*, 2019.
[3] A.-R. Sadeghi *et al.*, "Security and privacy challenges in industrial internet of things," in *DAC*, IEEE, 2015.
[4] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.
[5] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, "Privacy-preserving machine learning as a service.," *Proc. Priv. Enhancing Technol.*, vol. 2018, no. 3, pp. 123–142, 2018.
[6] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for big data systems," in *2015 IEEE 35th international conference on distributed computing systems*, pp. 318–327, IEEE, 2015.
[7] M. U. Hassan *et al.*, "Differential privacy techniques for cyber physical systems: a survey," *IEEE Communications Surveys & Tutorials*, pp. 746–789, 2019.
[8] A. Agarwal *et al.*, "Protecting privacy of users in brain-computer interface applications," *TNSRE*, 2019.
[9] F. Yang and S. Ren, "Adversarial attacks on brain-inspired hyperdimensional computing-based classifiers," *arXiv preprint arXiv:2006.05594*, 2020.
[10] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, no. 2, pp. 139–159, 2009.
[11] Z. Zou *et al.*, "Biohd: an efficient genome sequence search platform using hyperdimensional memorization," in *ISCA*, pp. 656–669, 2022.
[12] D. D. Wagner *et al.*, "Decoding the neural representation of self and person knowledge with multivariate pattern analysis and data-driven approaches," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 10, no. 1, p. e1482, 2019.
[13] Z. Zou *et al.*, "Eventhd: Robust and efficient hyperdimensional learning with neuromorphic sensor," *Frontiers in Neuroscience*, vol. 16, 2022.
[14] P. Poduval *et al.*, "Graphd: Graph-based hyperdimensional memorization for brain-like cognitive learning," *Frontiers in Neuroscience*, p. 5, 2022.
[15] Z. Zou *et al.*, "Memory-inspired spiking hyperdimensional network for robust online learning," *Scientific reports*, vol. 12, no. 1, pp. 1–13, 2022.
[16] P. Poduval *et al.*, "Cognitive correlative encoding for genome sequence matching in hyperdimensional system," in *ACM/IEEE DAC*, pp. 781–786, IEEE, 2021.
[17] 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)*, pp. 1–8, IEEE, 2017.
[18] M. Imani *et al.*, "A framework for collaborative learning in secure high-dimensional space," in *CLOUD*, pp. 435–446, IEEE, 2019.
[19] B. Khaleghi *et al.*, "Prive-hd: Privacy-preserved hyperdimensional computing," *arXiv preprint arXiv:2005.06716*, 2020.
[20] A. Hernández-Cano *et al.*, "Prid: Model inversion privacy attacks in hyperdimensional learning systems," in *ACM/IEEE DAC*, pp. 553–558, IEEE, 2021.
[21] A. Rahimi *et al.*, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *ISLPED*, pp. 64–69, ACM, 2016.
[22] A. Burrello *et al.*, "One-shot learning for ieeg seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing," in *BioCAS*, IEEE, 2018.
[23] A. Mitrokhin *et al.*, "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception," *Science Robotics*, vol. 4, no. 30, 2019.
[24] D. Kleyko *et al.*, "Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics," *TNNLS*, 2018.
[25] Z. Zou *et al.*, "Scalable edge-based hyperdimensional learning system with brain-like neural adaptation," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, 2021.
[26] A. Hernandez-Cane *et al.*, "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system," in *DATE*, pp. 56–61, IEEE, 2021.
[27] D. Ma *et al.*, "Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing," *arXiv preprint arXiv:2103.08668*, 2021.
[28] A. Moin *et al.*, "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," *Nature Electronics*, vol. 4, no. 1, pp. 54–63, 2021.
[29] M. Imani *et al.*, "DUAL: Acceleration of clustering algorithms using digital-based processing in-memory," in *MICRO*, IEEE, Oct. 2020.
[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NIPS*, vol. 32, pp. 8026–8037, 2019.
[31] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.
[32] "Uci machine learning repository." http://archive.ics.uci.edu/ml/datasets/ISOLET.
[33] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.
[34] D. Anguita *et al.*, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *AAL*, pp. 216–223, Springer, 2012.
[35] J. Sayyad Shirabad and T. Menzies, "The PROMISE Repository of Software Engineering Databases.." School of Information Technology and Engineering, University of Ottawa, Canada, 2005.
[36] J.-B. V. Pierre Mahé, "Micro-mass data set." https://www.openml.org/d/1514.
[37] C. K. E. Alpaydin, "Optdigits data set." https://www.openml.org/d/28.