Efficient Privacy-Utility Optimization for Differentially Private Deep Learning: Application to Medical Diagnosis

Original Scientific Paper

Rafika Benladgham*
Abu Baker Belkaid University of Tlemcen, Faculty of Sciences, Department of Computer Sciences, LRIT
Tlemcen, Algeria
rafika.benledghem@univ-tlemcen.dz

Fethallah Hadjila
Abu Baker Belkaid University of Tlemcen, Faculty of Sciences, Department of Computer Sciences, LRIT
Tlemcen, Algeria
fethallah.hadjila@univ-tlemcen.dz

Adam Belloum
University of Amsterdam, Informatics Institute
Amsterdam, The Netherlands
a.s.z.belloum@uva.nl

*Corresponding author

Abstract *– The optimization of differentially private deep learning models in medical data analysis using efficient hyper-parameter tuning is still a challenging task. In this context, we address the fundamental issue of balancing privacy guarantees with model utility by simultaneously optimizing model parameters and privacy parameters across two primary medical datasets, with additional validation on PathMNIST. Our framework encompasses both tabular data (Wisconsin Breast Cancer dataset) and medical imaging (BreastMNIST and PathMNIST), implementing four distinct optimization approaches: Grid Search, Random Search, Bayesian Optimization, and Bat Algorithm. Through extensive experimentation, we demonstrate a promising performance: achieving 93.62% accuracy with strong privacy guarantees (ε = 0.5) for tabular data, and 74.91% accuracy for medical imaging, with the Bat Algorithm discovering an unprecedented privacy level (ε = 0.293). Further validation on PathMNIST histopathology images demonstrated the framework's scalability, achieving 44.71% accuracy with privacy guarantees (ε = 2.603). Our comparative analysis reveals that different medical data types require distinct optimization strategies, with Bayesian Optimization excelling in tabular data applications and Random Search providing efficient solutions for image processing. The experiments with PathMNIST histopathology images provided valuable insights into the framework's behavior with complex medical data, revealing configuration-dependent performance variations and computational trade-offs. Our framework incorporates Pareto analysis and visualization techniques to enable systematic exploration of privacy-utility trade-offs, while early stopping mechanisms optimize privacy budget utilization. This comprehensive approach, validated across diverse medical imaging complexities and data modalities, establishes practical guidelines for implementing privacy-preserving machine learning in healthcare settings while highlighting the importance of balanced optimization strategies and computational efficiency in secure and efficient medical data analysis.*

Keywords*: differential privacy, deep learning, medical data analysis, privacy-utility optimization, hyper-parameter tuning*

Received: November 20, 2024; Received in revised form: January 30, 2025; Accepted: March 7, 2025

1. INTRODUCTION

The rapid digital transformation of healthcare has led to an unprecedented accumulation of sensitive medical data, from structured tabular data to complex medical imaging [1]. While this data surge offers immense opportunities for advancing medical research and improving diagnostic accuracy through machine learning, it also introduces a critical challenge: balancing data utility and privacy protection [2]. This challenge is particularly acute in healthcare, where advancing research and safeguarding patient confidentiality must coexist. Consequently, there is an urgent need for robust privacy-preserving mechanisms that do not compromise the analytical capabilities of machine learning models.

Among privacy protection approaches, differential privacy (DP) stands out for its mathematically sound privacy assurances [3, 4]. Within deep learning applications, differentially private stochastic gradient descent (DP-SGD) has become the predominant DP implementation method [5]. This approach combines two key mechanisms: gradient clipping and noise addition to ensure privacy protection during model training [5]. The process involves first limiting individual gradients by clipping them to maintain a specific $\ell 2$ norm threshold, followed by incorporating Gaussian noise into the averaged gradient before model parameter updates [5, 6]. By implementing these modifications, DP-SGD achieves bounded sensitivity for each training update, thus establishing privacy guarantees through controlled noise introduction into the learning process [5, 7].

However, optimizing hyper-parameters in differentially private models is inherently more complex than in non-private settings. Private hyperparameter optimization requires tuning additional parameters, including the clipping norm and noise scale, which are highly sensitive and make the process intricate and demanding [8]. Earlier studies have focused on fine-tuning privacy parameters to match non-private model performance or achieving acceptable performance levels while ensuring privacy assurances [8]. Despite these efforts, significant gaps remain in optimizing hyperparameters for differentially private deep learning models.

Traditional HP.O methods, such as grid search (G.S) and random search (R.S), are non-adaptive, evaluating hyperparameters from fixed or randomly generated sets [9]. While simple to apply, they are computationally intensive and poorly suited for high-dimensional search spaces, especially when additional privacy parameters (e.g., noise multiplier, clipping norm) must be tuned [9]. While adaptive methods like Bayesian optimization (B.O) use a probabilistic model to link hyperparameters to performance metrics and have become the preferred choice over non-adaptive methods due to their superior performance and scalability [10, 11], they often struggle to dynamically adjust privacy parameters during training, which is critical for balancing utility and privacy in real-time applications [10, 11]. Furthermore, existing approaches lack the ability to effectively navigate the complex trade-offs between exploration and exploitation in private HPO, limiting their scalability and performance in privacy-sensitive domains like healthcare [9]. These limitations underscore the need for innovative optimization techniques that can handle the unique challenges of differentially private deep learning. Inspired by its adaptive exploration and exploitation capabilities, we propose the Bat Algorithm as a novel solution for dynamic parameter tuning, addressing these gaps and enabling more efficient and scalable privacy-preserving models.

Recent advancements have demonstrated the effectiveness of swarm intelligence algorithms, such as the Bat Algorithm, in navigating complex search spaces [12]. Inspired by bats' echolocation and social behaviors, the Bat Algorithm dynamically adjusts search patterns to identify optimal hyper-parameters. Its ability to balance local and global search capabilities and adaptive frequency tuning makes it particularly well-suited for fine-tuning hyper-parameters in complex scenarios [12]. In differentially private deep learning, where privacy parameters like noise multiplier and clipping norm are critical, the Bat Algorithm offers a promising approach for dynamic parameter tuning. This leads to a compelling research question: *Can the Bat Algorithm be integrated to dynamically adjust privacy parameters during training, further improving the efficiency of differentially private model optimization*?

In this work, we address the challenge of HPO in differentially private deep learning by focusing on four explicit hyperparameters (learning rate, batch size, privacy budget, and maximum gradient norm) and two implicit ones (noise multiplier, and training epochs via early stopping). Our contributions are fourfold:

1. **Novel Application of the Bat Algorithm:** We propose and evaluate the Bat Algorithm for HP.O in differentially private deep learning, marking its first application in this domain.
2. **Comprehensive Comparison:** We systematically compare the Bat Algorithm against baseline methods (G.S, R.S, and B.O), providing consistent and reproducible results.
3. **Real-World Validation:** We validate our approach on real-world medical datasets (Breast Cancer Wisconsin, BreastMnist), demonstrating its practical applicability in privacy-sensitive healthcare applications.
4. **Generalizability and Scalability:** To further demonstrate the generalizability, feasibility, and scalability of our framework, we extend our evaluation to the PathMNIST dataset, which is more complex in terms of both data structure (histopathology images) and model architecture (ResNet-50). This extension rigorously tests the applicability of our framework to larger and more complex datasets, further validating its potential for real-world deployment in privacy-sensitive medical applications.

The remainder of this paper is organized as follows: Section 2 outlines the methodology and experimental setup, Section 3 presents results and analysis, Section 4 discusses findings, and Section 5 concludes with implications and future research directions.

2. RELATED WORK

The intersection of differential privacy and deep learning has been an active area of research, particularly in optimizing the balance between privacy guarantees and model utility.

2.1. DIFFERENTIAL PRIVACY IN MACHINE LEARNING

Recent advancements in Differential privacy (DP) have significantly expanded theoretical foundations and

practical applications of privacy-preserving techniques. Kulynych et al. [13] introduced an attack-aware noise calibration framework that moves beyond traditional ε-based approaches, demonstrating improved model accuracy while maintaining strong privacy guarantees. Complementing this work, Lu [14] established crucial relationships between noise addition strategies in stochastic gradient descent (SGD) and their impact on the model performance. In the domain of privacy budget management, Thantharate et al. [15] developed a systematic approach for tracking cumulative privacy loss across iterative training processes, enabling more precise control over privacy budgets in multi-stage learning scenarios. Pan Ke et al. [16], systematically investigate differentially private deep learning, addressing privacy attacks and preservation with a novel taxonomy. Despite these efforts, optimizing the privacy-utility trade-off continues to pose substantial challenges.

2.2. PRIVACY-UTILITY TRADE-OFFS AND OPTIMIZATION

Transfer learning approaches have shown promising results in medical image diagnosis. Battula and Chandana. [17] demonstrated 99.68% accuracy for cervical cancer classification using an optimized SE-ResNet152 model, highlighting the potential of architecture optimization in healthcare applications. The growing need for privacy preservation, however, necessitates approaches that balance such high performance with robust privacy guarantees. The progress in privacy-preserving machine learning has significantly enhanced our understanding of the privacy-utility trade-off paradigm. Kumar et al. [18] introduced a novel geometric approach using kernel-based methods in Reproducing Kernel Hilbert Spaces (RKHS), effectively reducing accuracy loss while mitigating membership inference risks in sensitive applications. Based on this foundation, Ficiu et al. [19] developed PFairDP, employing Bayesian optimization to identify Pareto-optimal points balancing fairness, privacy, and utility. Significant contributions to federated learning frameworks have emerged, with with Avent et al. [20] presenting a Bayesian optimization methodology to efficiently characterize the privacy-utility trade-off of differentially private algorithms using empirical utility measurements, while Koskela et al. [21] propose a method to enhance differentially private machine learning by tuning hyperparameters on a random data subset and extrapolating optimal values, reducing both privacy leakage and computational cost. Arous et al. [22], demonstrated choice strategies of model parameters (e.g., activation functions) that can significantly impact the privacy utility balance without compromising either aspect.

2.3. HYPER-PARAMETER OPTIMIZATION IN DIFFERENTIALLY PRIVATE DEEP LEARNING (DPDL)

Numerous approaches have been proposed to address the challenge of hyper-parameter optimization in DP. Galli et al. [23] offer foundational insights by dynamically optimizing the clipping threshold in differentially private learning, showing that traditional grid search methods incur excessive privacy costs, while Wang et al. [24] developed DP-HyPO, an adaptive framework leveraging Gaussian process-based optimization.

Significant algorithmic contributions include evolutionary approaches for exploring hyperparameter spaces, Bayesian optimization for probabilistic performance modeling, and enhanced Particle Swarm Optimization (EPSO) as Gao et al. [25] demonstrated for optimizing learning rates while minimizing noise impact. Bu et al. [26] introduced a novel book-keeping technique that improves computational costs while maintaining accuracy, making private optimization comparable to standard training. Tobaben [27] provides foundational insights by analyzing hyperparameter and architectural impacts on the privacy-utility trade-off in DPDL, revealing grid search's inefficiencies with private data.

2.4. META-HEURISTIC APPROACHES AND PARETO OPTIMIZATION

The work by Ramalingam et al. [28] and Banerjee et al. [29] has demonstrated the effectiveness of genetic algorithms, particle swarm optimization, and ant colony optimization in navigating vast solution spaces. These approaches have shown a particular promise in healthcare applications, with Singh et al. [30] successfully applying them to enhance feature selection for disease diagnosis while maintaining privacy constraints. The Pareto optimization aspect of these methods, as explored by Harkare et al. [31],is crucial in balancing multiple competing objectives, such as model accuracy, privacy guarantees, and computational efficiency. Thakur et al. [32] further extended these concepts to resource-constrained environments, demonstrating significant improvements in operational efficiency while maintaining solution diversity.

Table 1. Overview of Key Related Work

Research Area	Key Contributions	Reference
Differential Privacy	Attack-aware noise calibration beyond ε-based approaches	[13]
	Noise addition strategies in SGD	[14]
Privacy-Utility Trade-offs and optimization	SE-ResNet152 optimization using DHO algorithm for medical image classification	[17]
	Subset-based hyperparameter tuning for privacy-utility optimization	[21]
Hyper-parameter Optimization in DPDL	DP-HyPO adaptive framework	[24]
	EPSO for learning rate optimization	[25]
	Systematic optimization strategy comparison	[26]
Meta-heuristic Approaches	Genetic and particle swarm optimization analysis	[28]
	Healthcare feature selection optimization	[29]

3. PROBLEM FORMULATION

Our problem formulation establishes a unified framework for optimizing the privacy-utility trade-off in differentially private deep learning models. We have defined an objective function that balances model accuracy and privacy guarantees and formulated the optimization problem for four distinct approaches: Grid Search, Random Search [33], Bayesian Optimization [34], and Bat Algorithm [35]. Each method navigates the hyperparameter space Θ in its unique way, aiming to find the optimal configuration $\theta*$ that maximizes our objective function $f(\varepsilon(\theta), A(\theta))$.

3.1. DIFFERENTIAL PRIVACY FRAMEWORK

A randomized algorithm $M : D \to R$ with domain D and range R is (ε, δ)- differentially private if for all $S \subseteq R$ and for all adjacent datasets $D, D' \in D$ [36]:

$$P[M(D) \in S] \leq exp(\varepsilon) \cdot P[M(D') \in S] + \delta \quad (1)$$

where: - ε is the privacy budget - δ is the failure probability.

3.2. DP-OPTIMIZATION COMPONENTS

3.2.1. Hyperparameter Space

Let $\theta = (lr, bs, nm, C)$ be the hyperparameter vector where: - lr: learning rate - bs: batch size - nm: noise multiplier - C: gradient clipping threshold The feasible space Θ is defined by:

$$\begin{gathered}\Theta = \{\theta \mid lr_{min} \leq lr \leq lr_{max}, \\ bs_{min} \leq bs \leq bs_{max}, \\ t\varepsilon_{min} \leq t\varepsilon \leq t\varepsilon_{max}, \\ C_{min} \leq C \leq C_{max}\end{gathered} \quad (2)$$

3.2.2. Privacy-Utility Metric

For any configuration θ: - $A(\theta)$: model accuracy - $\varepsilon(\theta)$: achieved privacy budget

3.2.3. Objective Function

The privacy-utility trade-off is quantified by:

$$f(\varepsilon, A) = \frac{\alpha e^{-\varepsilon}.\beta e^{(1-A)}}{\alpha e^{-\varepsilon}+\beta e^{(1-A)}}$$

where α, β are weighting parameters.

3.3. OPTIMIZATION PROBLEM

3.3.1. Primary Objective

Our goal is to find the optimal hyperparameter configuration $\theta*$ that maximizes $f(\varepsilon(\theta), A(\theta))$, the optimization problem can be formally stated as:

3.3.2. Pareto Optimality

To comprehensively analyze the trade-off between privacy and utility, we introduce the concept of Pareto optimality. The Pareto frontier P represents the set of non-dominated solutions where it's impossible to improve either privacy or utility without degrading the other. Formally, we define P as:

$$P=\{(\varepsilon, A) \in S \mid \nexists\ (\varepsilon', A') \in S: (\varepsilon' < \varepsilon \wedge A' \geq A) \vee (\varepsilon' \leq \varepsilon \wedge A' > A)\} \quad (4)$$

3.4. SOLUTION APPROACHES

To solve this optimization problem, and To find $\theta*$, we employ and compare four approaches distinct approaches:

1. Grid Search: Exhaustive search over a predefined hyper-parameter space,

$$\theta^*_{GS} = \arg\max_{\theta \in \Theta_{GS}} \mathrm{f}(\varepsilon(\theta), \mathrm{A}(\theta)) \quad (5)$$

2. Random Search: Randomly sampling configurations from the hyperparameter space [33]

$$\theta^*_{RS} \approx \arg\max_{\theta \in \Theta_{RS}} \mathrm{f}\big(\varepsilon(\theta), \mathrm{A}(\theta)\big) \quad (6)$$

3. Bayesian Optimization: Sequential model-based optimization using Gaussian Processes [34],

$$\theta^*_{BO} \approx \arg\max_{\theta \in \Theta_{BO}} \mathrm{f}\big(\varepsilon(\theta), \mathrm{A}(\theta)\big) \quad (7)$$

4. Bat Algorithm: A nature-inspired meta-heuristic optimization algorithm [35],

$$\theta^*_{BA} \approx \arg\max_{\theta \in \mathrm{BA}} \mathrm{f}\big(\varepsilon(\theta), \mathrm{A}(\theta)\big) \quad (8)$$

Where ΘX represents the search space explored by method X. Each method aims to efficiently navigate the hyperparameter space to find the configuration that maximizes our objective function, thus achieving the best privacy-utility trade-off for our differentially private deep learning model.

3.5. IMPLEMENTATION CONTEXT

The optimization is implemented using the Opacus privacy engine, which: 1. Computes per-sample gradients 2. Clips gradients to bound sensitivity 3. Adds calibrated Gaussian noise 4. Tracks privacy budget consumption The DP optimizer update step is:

$$\theta_{t+1} = \theta_t - \eta . m_t(\tilde{g}_t) \quad (9)$$

Where $\tilde{g}_t$ is the differentially private gradient:

$$\tilde{g}_t = \frac{1}{|B|} \sum_{i \in B} clip(\nabla_\theta L(\theta_t, x_i), C) + \mathcal{N}(0, \sigma_t^2 C^2 \mathbf{I}) \quad (10)$$

where $\tilde{g}_t$ is the privatized gradient, B is the batch size, C is the clipping threshold, and σ_t is the dynamically adjusted noise multiplier at step t.

4. METHODOLOGY

This section presents our comprehensive methodology for optimizing the privacy-utility trade-off in differentially private deep learning. Our framework encompasses model architectures, differential privacy

implementation, hyperparameter optimization techniques, and evaluation procedures. Beyond establishing the foundational approach, we extend our investigation to assess the framework's generalizability by incorporating the PathMNIST dataset — a complex collection of histopathology images that presents more challenging scenarios compared to the BreastMNIST and Breast Cancer Wisconsin datasets. This extension, implemented through a privacy-adapted ResNet-50 architecture with DP-Optimizer, enables us to evaluate our optimization framework's scalability and feasibility on larger, more complex models. Through this comprehensive approach, we aim to provide a clear, reproducible framework for comparing optimization strategies in privacy-preserving deep learning, while demonstrating its applicability across varying levels of task complexity.

4.1. EXPERIMENTAL FRAMEWORK OVERVIEW

Our experimental framework was implemented on the Google Colab Pro+ platform, leveraging TPU v2-8 accelerators that provide 8 cores with up to 180 teraflops of computation power and 64 GB of high-bandwidth memory (HBM). This infrastructure choice was crucial for handling the computational overhead. The summary of the setting parameters is shown in Table 2.

Table 2. Experimental Framework Specifications

Component	Specification
Platform	Google Colab Pro+
Hardware	TPU v2-8 (8 cores, 180 teraflops)
Memory	64 GB HBM
Framework	Python 3 + PyTorch
Cross-Validation	3-fold
Early Stopping	with patience monitoring
Optimizer	ADAM
Loss function	CrossEntropy

4.2. DATASETS AND PREPROCESSING

4.2.1. Breast Cancer Wisconsin Dataset

The Wisconsin Breast Cancer Dataset (UCI Repository) contains 569 samples with 30 features and binary classification (malignant/benign). Data was preprocessed and split into training (301), validation (85), and test (183) sets, then converted to PyTorch tensors for model training.

4.3.1. Model architecture

Multi-Layer Perceptron architecture comprises Input(30) → Linear(20) → Linear(10) → Linear(10) → Linear(10) → Linear(5) → Output(2) with ReLU activations between layers. The architecture employs gradual dimension reduction to prevent overfitting on the breast cancer Wisconsin classification task.

4.3.2. Medical Image Datasets

The BreastMNIST and PathMNIST datasets (MedMNIST v2.2.3) represent distinct medical imaging modalities while sharing standardized preprocessing requirements. Both datasets undergo similar technical preprocessing steps, with images normalized (μ=0.5, σ=0.5) and formatted to 28×28 pixel RGB resolution for deep learning compatibility. However, they differ significantly in their modalities and clinical applications. BreastMNIST focuses on breast imaging diagnostics, containing 780 medical images distributed across training (546), validation (78), and test (156) sets for binary classification tasks. In contrast, PathMNIST encompasses histopathological imaging, presenting a larger collection of 107,180 microscopic tissue images from colon pathology. These are divided into training (89,996), validation (10,004), and test (7,180) sets, supporting a more complex nine-class classification challenge that reflects the diverse cellular patterns and tissue characteristics encountered in pathological analysis.

4.3.3. ResNet Architectures

Our implementation utilizes modified ResNet architectures (ResNet-18 and ResNet-50) with specific privacy-focused adaptations for medical image classification. Both models share fundamental privacy-preserving modifications, replacing BatchNorm layers with GroupNorm (32 groups) to comply with Opacus's privacy requirements, as BatchNorm operations can leak private information across training examples. The architectures maintain pre-trained backbones in frozen evaluation mode while incorporating trainable classification heads. The key distinction lies in their complexity and target tasks: ResNet-18 is configured for binary classification of breast images, and ResNet-50, being deeper and more complex, handles the nine-class histopathology classification task for PathMNIST. Both architectures preserve privacy guarantees through GroupNorm's channel-based normalization approach and ensure efficient feature extraction through their frozen pre trained backbones, demonstrating adaptability to different medical imaging modalities while maintaining privacy-preserving characteristics.

4.3.4. Training Configuration

Our training framework implements model-specific configurations to ensure optimal convergence while managing computational resources effectively. The MLP architecture employs a maximum of 500 epochs with a patience value of 250, while the ResNet-18 training is configured with 20 maximum epochs and a patience threshold of 12. For the PathMNIST experiments using the Bat Algorithm, we established two distinct configurations as detailed in Table 3, where both maintain a population size of 2 bats but differ in their convergence parameters: Configuration 1 uses 14 epochs with early stopping at 10, while Configuration 2 employs 15 epochs with early stopping at 13.

Across all models, we implemented early stopping monitoring validation accuracy to prevent overfitting and ensure optimal convergence. These parameter adjustments, particularly for the PathMNIST experiments, were essential to complete the optimization process while managing computational constraints.

Table 3. Bat Algorithm Optimization Parameters: Two Configurations for PathMNIST classification

	1st Configuration	2nd Configuration
Optimization Parameters		
Population Size	2 bats	2 bats
Max-iterations	14 (convergence at 11)	4 (convergence at 2)
Epochs	15	15
Early Stopping Patience	10	13

4.4. DIFFERENTIAL PRIVACY IMPLEMENTATION

Table 4. Differential Privacy Components

Component	Implementation
Library	Opacus
Engine	PrivacyEngine
Delta (δ)	Fixed at 10^{-4}
Privacy Mechanism	Gradient clipping + Gaussian noise

Differential privacy is implemented via Opacus Privacy Engine with two key components:

1. Privacy Engine Operation:
- Per-sample gradient computation
- Gradient clipping for sensitivity bounds
- Calibrated Gaussian noise addition
- Privacy budget tracking
2. Dynamic Privacy Management:
- Target epsilon (εtarget) specification
- Dynamic noise multiplier (σ) adjustment
- Automated noise calibration
- Privacy parameter reporting

The engine adjusts the noise multiplier during training to balance target epsilon and model utility, following equation (10).

4.5. HYPERPARAMETER SPACE

Our hyperparameter optimization space was carefully defined to accommodate both discrete and continuous optimization methods as shown in Table 5:

4.5.1. Parameter Adjustment Mechanisms

Parameter adjustment mechanisms for continuous optimization include reflection methods that handle out-of-bounds values through boundary reflection, and value adjustment processes that discretize batch sizes, enforce integer constraints, clip boundary values, and prevent negative values.

Table 5. Hyperparameter Space Definition

Parameter	Discrete Values	Continuous Range	Notes
Learning Rate	{0.001, 0.01, 0.1}	[0.001, 0.1]	Three orders of magnitude
Batch Size	{16, 32, 64, 128, 512}	[16, 512]	Powers of 2
Max Gradient Norm	{1.2, 5.6}	[1.2, 5.6]	Conservative range
Privacy Budget (ε)	{0.5, 1.0, 8.0}	[0.2, 8.0]	Strict to relaxed privacy
Privacy Delta (δ)	Fixed at 10^{-4}		Standard failure probability

4.6. OPTIMIZATION FRAMEWORK

The optimization framework is designed to systematically achieve an optimal privacy-accuracy balance in Differentially Private Deep Learning (DPDL) models. It comprises three key phases: initialization, optimization, and Pareto analysis. Fig.1 illustrates the comprehensive workflow, while Fig.2 provides a detailed flowchart of the optimization methods. This framework establishes a robust foundation for achieving optimal privacy-utility trade-offs through systematic parameter tuning and multi-objective optimization.

4.6.1. Objective Function

For the objective function defined in equation (3), we set $\alpha = \beta = 0.5$ to ensure equal importance between privacy preservation (ε) and model accuracy (A), thus achieving a balanced privacy-utility optimization without favoring either aspect.

4.6.2. Hyperparameter Tuning Methods

To identify the optimal hyperparameter configurations, we employ four distinct methods G.S, R.S, B.O, and B.A. Each method is described below:

a) Grid Search (G.S): Deterministic Exploration

G.S operates as an exhaustive search method, systematically evaluating every possible combination of hyperparameters within a predefined discrete space. By exploring the entire search space, G.S identifies the optimal configuration that effectively balances the privacy-utility trade-off. This method ensures a thorough and methodical approach to hyperparameter tuning, albeit at a higher computational cost.

b) Random Search (R.S): Stochastic Exploration

R.S samples hyperparameter configurations from a uniform distribution $U(\theta_{RS})$ over a predefined discrete search space Θ_{RS}. It evaluates a fixed number of configurations (N), typically covering approximately 25% of the parameter space. By focusing on a subset of the search space, R.S efficiently approximates the optimal solution while significantly reducing computational overhead compared to exhaustive methods like G.S.

c) Bayesian Optimization (B.O): Sequential Model-Based Optimization

Bayesian Optimization employs a Gaussian Process (GP) as a surrogate model to approximate the objective function $f(\varepsilon(\theta), A(\theta))$. The process begins with 10 warm-up points, randomly sampled to initialize the GP model. At each iteration, the next hyperparameter configuration is selected by maximizing the Expected Improvement (EI):

$$\theta_{t+1} = \arg\max_{\theta \in \Theta} \mathrm{EI}(\theta|D) \tag{11}$$

Where $D=\{(\theta_i, f_i)\}_{i=1}^{t}$ represents the set of observations up to iteration t. This approach sequentially refines the GP model, guiding the search toward regions of the hyperparameter space that are most likely to improve performance.

Through this iterative process, equation (11) guides us toward the optimal solution represented in equation (7). At each step t:

- The GP model is updated using all previous observations D
- $EI(\theta|D)$ estimates where the next evaluation might most improve upon our current best solution
- This sequential refinement helps us approximate the optimal hyper-parameters ${\theta^*}_{BO}^*$ that maximize our objective function $f(\varepsilon(\theta), A(\theta))$

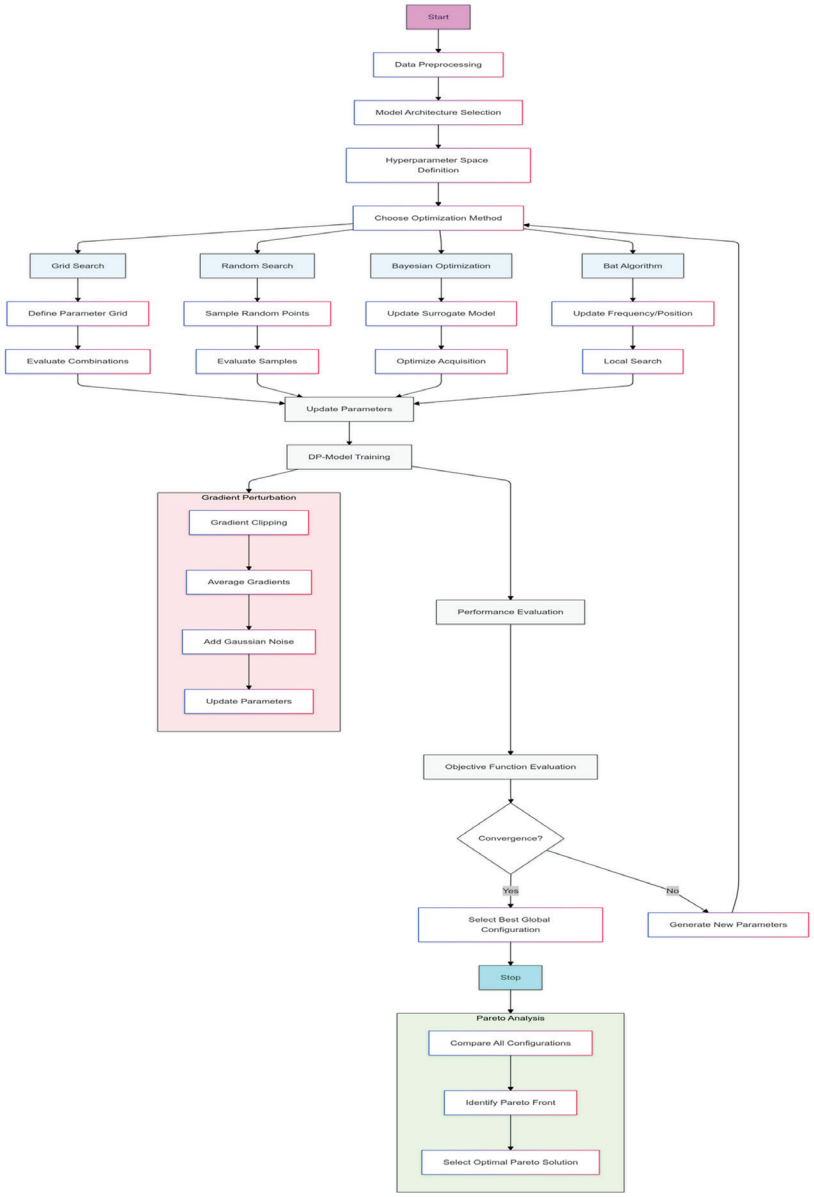

Fig. 1. Flowchart of Hyperparameter Optimization with Gradient Perturbation for Privacy-Accuracy Balance in DPDL

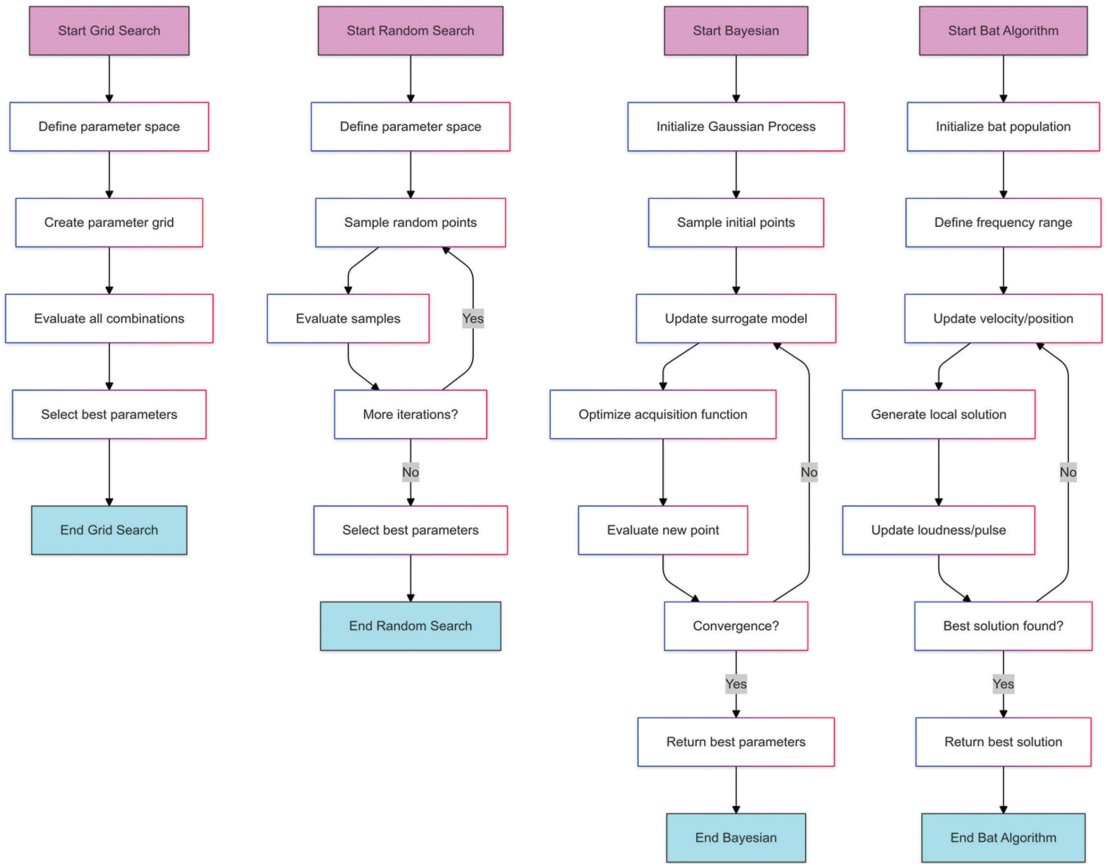

Fig. 2. Flowchart of each Optimization Method

B.O. effectively balances exploration and exploitation, making it highly efficient for high-dimensional spaces.

- Nature-inspired metaheuristic for hyperparameter tuning: The Bat Algorithm is a swarm intelligence-based optimization method that mimics the echolocation behavior of bats to navigate the hyperparameter space. It combines global exploration and local exploitation by iteratively updating the position x_i and velocity v_i of each bat i. The algorithm is guided by frequency f_i, loudness A_i, and pulse emission rate r_i, which are dynamically adjusted to balance exploration and exploitation. Algorithm 1 provides the complete pseudocode for the Bat Algorithm, illustrating its iterative parameter updates for balanced local and global search space exploration.

Algorithm 1
Bat Algorithm for Hyperparameter Tuning

1: **Initialize Parameters**:
2: Population size: $N = 10$ bats
3: Dimensions: $D = 4$ parameters
4: Frequency range: $[f_{min}, f_{max}] = [0, 10]$
5: Loudness: $\boldsymbol{A_i} = 1.0$ (initial)
6: Pulse emission rate: $\boldsymbol{r_i = r_i^0}$ (initial)
7: Alpha ($\boldsymbol{\alpha}$): 0.9
8: Gamma ($\boldsymbol{\gamma}$): 0.9
9: Maximum iterations: $\boldsymbol{T_{max}}$ (dataset-specific)
10: **Initialize Population:**
11: Randomly initialize positions $\boldsymbol{x_i}$ and velocities $\boldsymbol{v_i}$ for each bat i.
12: **Evaluate Initial Fitness:**
13: Compute fitness $\boldsymbol{f(x_i)}$ for each bat i.
14: Identify the global best solution $\boldsymbol{x_{best}}$.
15: Main Loop (**for** $\boldsymbol{t}$=**1 to** $\boldsymbol{T_{max}}$):
16: **for each** bat i **do**
17: Generate frequency $\boldsymbol{fi}$:
18: $\boldsymbol{fi}=\boldsymbol{f_{min}}+(\boldsymbol{f_{max}}-\boldsymbol{f_{min}})\cdot\boldsymbol{\beta}$, $\boldsymbol{\beta}\in[0,1]$
19: Update velocity $\boldsymbol{vi}$:
20: $\boldsymbol{v_i^t}+1=\boldsymbol{v_i^t}+(\boldsymbol{x_i^t}-\boldsymbol{x_{best}})\cdot\boldsymbol{fi}$
21: Update position $\boldsymbol{x_i}$:
22: $\boldsymbol{x_i^{t+1}}=\boldsymbol{x_i^t}+\boldsymbol{v_i^{t+1}}$
23: **if** $\boldsymbol{rand}>\boldsymbol{r_i}$ **then**
24: Perform local search:
25: $\boldsymbol{x_{new}}=\boldsymbol{x_{old}}+\boldsymbol{\varepsilon}\cdot\boldsymbol{A^t}$, $\varepsilon\in[-1,1]$
26: **end if**
27: Evaluate fitness $\boldsymbol{f(x_i^{t+1})}$.
28: **if** $\boldsymbol{rand} < \boldsymbol{A_i}$ and $\boldsymbol{f(x_i^{t+1})} < \boldsymbol{f\,x_{best}}$) **then**
29: Update $\boldsymbol{x_{best}} = \boldsymbol{x_i^{t+1}}$.
30: **end if**
31: Update loudness $\boldsymbol{A_i}$ and pulse emission rate $\boldsymbol{r_i}$:
32: $\boldsymbol{A_i^{t+1}}=\boldsymbol{\alpha}\cdot\boldsymbol{A_i^t}$
33: $\boldsymbol{r_i^{t+1}}=\boldsymbol{r_i^0}\cdot[1-\exp^{(-\gamma\cdot t)}]$
34: **end for**
35: **Return Optimal Solution:**
36: Output the global best solution $\boldsymbol{x_{best}}$.

4.7. PARETO ANALYSIS

Our Pareto efficiency analysis identifies the optimal trade-off between privacy preservation (ε) and model accuracy (A) across all optimization methods. The analysis involves two key steps:

1. Pareto Front Identification

- We use a non-dominated sorting algorithm to identify the Pareto front, comprising solutions that are not dominated by any other configuration in terms of ε and A.
- Optimization occurs in a two-dimensional space, where each point represents a unique (ε, A) combination.
- Pair-wise dominance comparisons determine dominance: a solution (ε_1, A_1) dominates (ε_2, A_2) if $\varepsilon_1 \leq \varepsilon_2$ and $A_1 \geq A_2$, with at least one strict inequality.

2. Optimal Point Selection

- The optimal Pareto point is selected from the set of non-dominated solutions (P), satisfying the dominance relation in Equation (4).
- If the global best solution matches the Pareto point for a method, it indicates the method's superiority in achieving the best non-dominated trade-off. This highlights the method's robustness, efficiency, and practical applicability for privacy-preserving machine learning tasks while providing a basis for method comparison and future research.

5. COMPREHENSIVE ANALYSIS OF RESULTS

Based on our experimental evaluation across multiple datasets and optimization methods, we present a detailed analysis of the performance metrics, hyperparameter configurations, computational resources utilized, and comparative assessment of optimization strategies in our framework. Our analysis includes a comprehensive performance evaluation across key metrics, a detailed examination of how hyperparameters influence outcomes, a specific analysis of the PathMNIST dataset's performance, an assessment of resource utilization patterns, and a comparative investigation of the strengths and limitations of different methods across various datasets and optimization objectives.

5.1. PERFORMANCE METRICS ANALYSIS

We begin our analysis by examining the performance metrics detailed in Table 6, focusing on accuracy, privacy preservation, and computational efficiency across all optimization methods and datasets. This analysis provides insights into how each method balances these critical performance dimensions.

The Grid Search method demonstrated consistent performance across datasets, achieving an accuracy of 93.40% on the Breast Cancer Wisconsin dataset and 74.18% on BreastMNIST. Maintaining a privacy budget (ε) of 0.500 required substantial computational resources, particularly evident in the 4,000.33 seconds processing time for the Wisconsin dataset.

Random Search exhibited comparable accuracy metrics, reaching 92.53% on the Wisconsin dataset and 74.91% on BreastMNIST. Notably, it achieved these results with varying privacy budgets - 0.500 for Wisconsin and 1.000 for BreastMNIST. The method showed improved time efficiency compared to Grid Search, completing the Wisconsin dataset analysis in 2,749.65 seconds.

Bayesian Optimization achieved the highest accuracy on the Wisconsin dataset at 93.62%, while maintaining a privacy budget of 0.501. The method demonstrated consistent performance on BreastMNIST with 73.99% accuracy. Its computational requirements were significant, requiring 17,828.72 seconds for the Wisconsin dataset, though with reduced memory usage of 787.75 MB.

The Bat Algorithm showed distinct characteristics, achieving Pareto optimality on both datasets. While its accuracy was slightly lower (93.18% for Wisconsin, 73.08% for BreastMNIST), it demonstrated efficient privacy preservation with ε values of 2.258 and 0.293 respectively.

Table 6. Performance Metrics Across Datasets and Optimization Methods

Dataset	Method	Privacy (ε)	Accuracy (%)	Fitness (f)	Time (s)	Memory (MB)	Pareto Optimal?
Breast cancer Wisconsin	Grid Search	0.500	93.40	0.1840	4,000.33	1,064.83	Yes
	Random Search	0.500	92.53	0.1834	2,749.65	845.08	No
	Bayesian Opt.	0.501	93.62	0.1840	17,828.72	787.75	No
	Bat Algorithm	2.258	93.18	0.0470	14,562.20	1,103.16	Yes
BreastMNIST	Grid Search	0.500	74.18	0.1699	12,931.66	1,796.29	Yes
	Random Search	1.000	74.91	0.1697	6,499.74	1,555.48	Yes
	Bayesian Opt.	0.500	73.99	0.1697	38,076.04	1,322.85	Yes
	Bat Algorithm	0.293	73.08	0.1887	10,132.33	4,348.25	Yes

5.2. HYPERPARAMETER CONFIGURATION ANALYSIS

To understand the factors driving performance differences, we examine the optimal hyper-parameter configurations identified by each method across datasets, as presented in Table 7. This analysis reveals key patterns in parameter selection and their impact on optimization outcomes.

The optimal learning rates varied significantly across methods. Grid Search performed best with smaller learning rates (0.01), while the Bat Algorithm required higher rates (0.0965 for Wisconsin). Batch sizes showed a clear pattern, with most optimal configurations favoring larger batches (512) for the Wisconsin dataset and varying sizes for BreastMNIST. Privacy budgets demonstrated method-specific patterns. Random Search and Grid Search maintained consistent budgets (0.500), while the Bat Algorithm and Bayesian Optimization showed more variation. The max gradient norm values remained relatively stable across methods for BreastMNIST but showed greater variation in the Wisconsin dataset. Training epochs exhibited method-specific patterns, with Grid Search requiring 273 epochs for optimal performance on the Wisconsin dataset, while BreastMNIST achieved optimal results with significantly fewer epochs (12-14) across all methods.

Table 7. Best Hyperparameter Configurations for Each Method.

Dataset	Method	Learning Rate (lr)	Batch Size(bs)	Privacy Budget (ε_{terget})	Max Grad Norm (C)	Noise Multiplier (σ)	Training Epochs	Final Accuracy(%)
Breast cancer Wisconsin	Grid Search	0.01	32	0.500	1.2	42.5	273	93.40
	Random Search	0.1	512	0.500	5.6	135.0	252	92.53
	Bayesian Opt.	0.018	128	0.501	3.118	77.5	265	93.62
	Bat Algorithm	0.0965	512	2.258	3.388	95.5	248	93.18
BreastMNIST	Grid Search	0.1	32	0.500	1.2	6.25	12	74.18
	Random Search	0.1	64	1.000	1.2	8.75	12	74.91
	Bayesian Opt.	0.1	128	0.500	4.906	12.19	15	73.99
	Bat Algorithm	0.0429	512	0.293	3.114	18.5	14	73.08

5.3. PATHMNIST-SPECIFIC PERFORMANCE ANALYSIS

As detailed in Table 8, our focused analysis of the PathMNIST dataset provides additional insights into the optimization framework's capability to handle complex medical imaging data and adapt configurations for improved performance. The results demonstrate significant variations between configurations and their impact on multiple performance dimensions.

Table 8. Results on the PathMNIST dataset

	1st Configuration	2nd Configuration
Global Best Solution		
Learning Rate (lr)	0.03106	0.01760
Batch Size	128	512
Max-Grad-Norm (C)	2.179	1.732
TargetEpsilon(ε)	0.508	2.603
Performance Metrics		
Objective Function	0.1435	0.0328
GlobalBest Accuracy	39.97%	44.71%
Global Best Epsilon	0.508	2.603
Time (S)	18,238.194	13,678.6455
Memory (MB)	8,479.0875	7,522.4725
Pareto Optimal?	No	Yes

The PathMNIST dataset results revealed significant improvements between configurations. The global best accuracy increased from 39.97% to 44.71%, accompanied by changes in the learning rate from 0.03106 to 0.01760. The second configuration achieved Pareto optimality while reducing memory requirements from 8,479.0875 MB to 7,522.4725 MB.

The objective function improved from 0.1435 to 0.0328, indicating enhanced optimization performance. The global best epsilon value increased from 0.508 to 2.603, suggesting a different privacy-utility trade-off in the optimal configuration. Execution time increased from 8,238.194 seconds to 13,678.6455 seconds, demonstrating the computational cost of achieving improved performance metrics.

5.4. RESOURCE UTILIZATION ASSESSMENT

Understanding the computational demands of each optimization method is crucial for practical implementation. Our analysis of resource utilization reveals significant variations in memory and time requirements across methods and datasets.

Memory requirements varied significantly across methods and datasets. For the Wisconsin dataset, Grid

Search utilized 1,064.83 MB, while the Bat Algorithm required 1,103.16 MB. BreastMNIST showed higher memory requirements overall, with the Bat Algorithm consuming 4,348.25 MB.

Execution times demonstrated substantial variation, ranging from 2,749.65 seconds for Random Search to 17,828.72 seconds for Bayesian Optimization on the Wisconsin dataset. BreastMNIST generally required longer processing times, with Grid Search taking 12,931.66 seconds and Bayesian Optimization requiring 38,076.04 seconds.

The fitness values across methods remained relatively consistent within each dataset, suggesting that different optimization approaches converged to similarly optimal solutions despite varying computational requirements and privacy-utility trade-offs.

5.5. COMPARATIVE ANALYSIS

To synthesize our findings, we examine the relative strengths and limitations of each optimization method across datasets, highlighting key trade-offs and operational considerations.

Our comparative analysis reveals distinctive patterns across optimization methods and datasets, highlighting the inherent trade-offs between privacy, accuracy, and computational efficiency. On the Wisconsin dataset, Grid Search and Bayesian Optimization achieved comparable accuracy levels (93.40% and 93.62% respectively) while maintaining similar privacy budgets (0.500 and 0.501). However, Bayesian Optimization required approximately 4.5 times more computational time, suggesting a significant efficiency trade-off for marginal accuracy improvement.

The BreastMNIST dataset results demonstrate different optimization dynamics. Random Search emerged as the top performer with 74.91% accuracy, albeit requiring a higher privacy budget (ε = 1.000) compared to other methods. This illustrates the inherent tension between privacy preservation and model performance. Grid Search achieved comparable accuracy (74.18%) with half the privacy budget (ε = 0.500), representing a potentially more balanced solution for privacy-sensitive applications.

The Bat Algorithm's performance presents an interesting case study in multi-objective optimization. Despite achieving lower accuracy scores on both datasets, it consistently achieved Pareto optimality, suggesting superior performance in balancing multiple competing objectives. Its moderate memory requirements (1,103.16 MB for Wisconsin, 4,348.25 MB for BreastMNIST) and execution times position it as a practical choice for resource-constrained environments.

A cross-dataset comparison reveals that optimization methods demonstrate dataset-specific strengths. While Grid Search exhibited stable performance across both datasets, Random Search showed higher variability, performing notably better on the BreastMNIST dataset. This suggests that dataset characteristics significantly influence the effectiveness of different optimization strategies.

5.6. EXTENDED ANALYSIS WITH VISUALIZATION RESULTS

This analysis presents a comprehensive visualization-based examination of our optimization results across three distinct datasets: Wisconsin Breast Cancer, BreastMNIST, and PathMNIST. Our visualization framework employs two key components: privacy-accuracy trade-off plots with fitness value indicators, and convergence plots showing the evolution of fitness values over iterations. This dual visualization approach enables us to understand both the final solution space and the optimization trajectory for each method.

5.6.1. Breast Cancer Wisconsin Dataset

The Breast Cancer Wisconsin Dataset optimization results reveal distinct patterns across the four optimization methods, showcasing various approaches to balancing privacy and accuracy.

Grid Search and Random Search (Fig. 3a, 3b) demonstrate similar exploration patterns, characterized by discrete, well-defined sampling points. The privacy-accuracy trade-off plots show concentrated exploration in specific regions, with Grid Search providing more systematic coverage while Random Search offers more scattered distribution. Both methods achieve relatively quick convergence as shown in their fitness evolution plots, reaching stable fitness values early in the optimization process.

Bayesian Optimization (Fig. 3c) exhibits a more sophisticated exploration strategy, with dense sampling in promising regions of the solution space. The privacy-accuracy plot reveals a continuous distribution of points, suggesting a more thorough exploration of the trade-off space. The convergence plot shows rapid initial improvement followed by consistent refinement, indicating efficient optimization behavior.

The Bat Algorithm (Fig. 3d) demonstrates a unique exploration pattern, with initial broad coverage followed by concentrated sampling in high-performing regions. The privacy-accuracy plot shows clusters of solutions, particularly in areas of favorable trade-offs. The fitness evolution plot reveals a distinctive stepped pattern, suggesting periodic improvements in solution quality as the algorithm explores the search space.

Regarding optimal solutions, all methods successfully identified configurations that balance privacy and accuracy, with the Bayesian Optimization and Bat Algorithm showing a particularly effective exploration of the solution space near the Pareto frontier. The convergence behavior suggests that while Grid Search and Random Search reach stable solutions quickly, Bayesian Optimization and the Bat Algorithm continue to refine their solutions throughout the optimization process, potentially discovering more nuanced trade-offs.

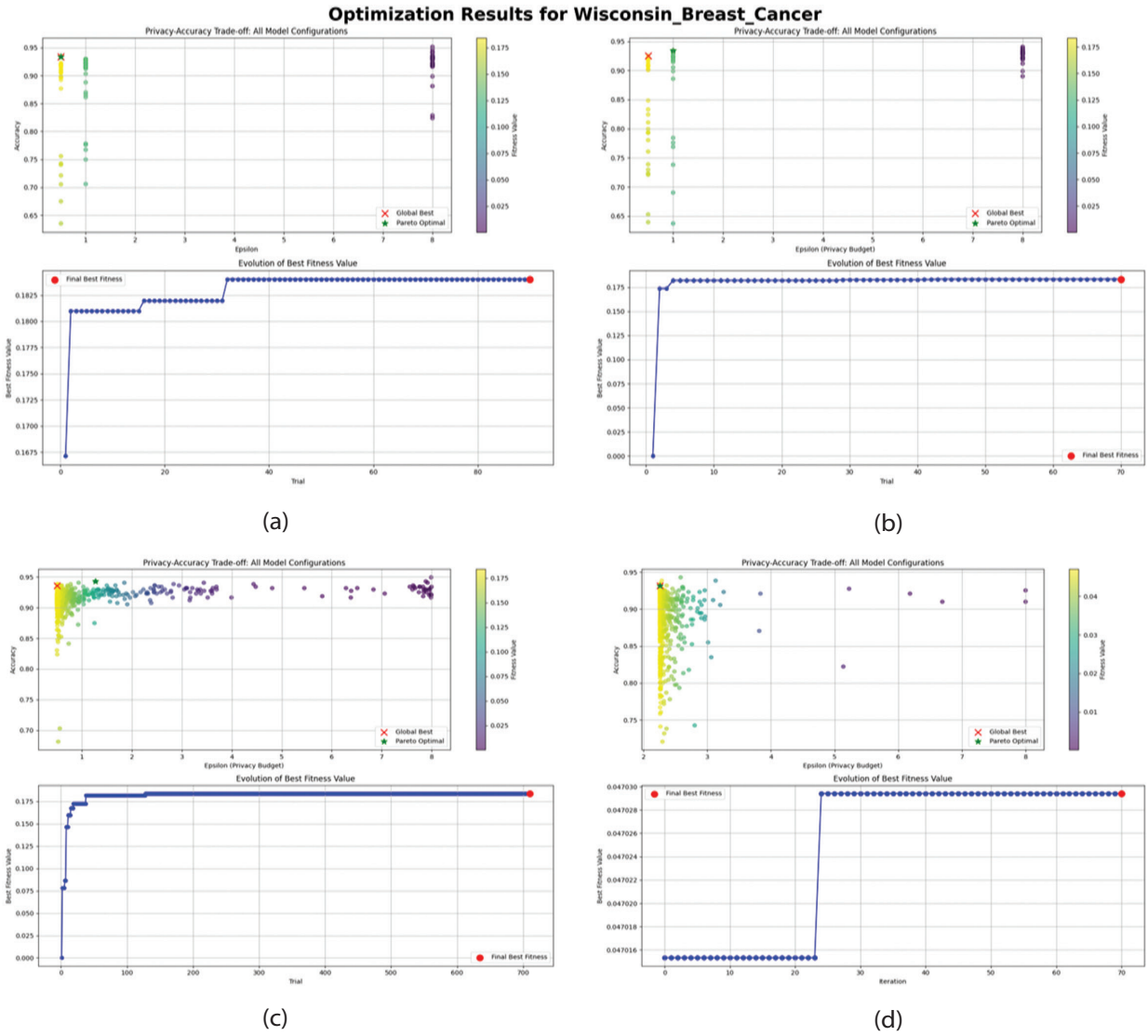

Fig. 3. Optimization Results for Wisconsin Breast Cancer Dataset: (a) Grid Search, (b) Random Search, (c) Bayesian Optimization, and (d) Bat Algorithm. The top row shows privacy-accuracy trade-off plots with color indicating fitness values. The bottom row shows the evolution of the best fitness value over iterations.

5.6.2. BreastMNIST Dataset

The optimization results for the BreastMNIST dataset reveal distinctive characteristics and performance patterns across the four optimization methods, with notable differences from the Wisconsin Breast Cancer dataset analysis.

Grid Search (Fig. 4a) shows a structured exploration pattern with evenly distributed sampling points across the privacy-accuracy space. The convergence plot displays a step-like pattern, indicating discrete improvements in fitness values at specific intervals. This suggests that the method systematically identified better solutions through its predefined search grid.

Random Search (Fig. 4b) demonstrates a more scattered distribution of solutions, yet maintains coverage across the solution space. The fitness evolution plot shows rapid initial improvement followed by sustained performance, suggesting early discovery of promising regions in the search space.

Bayesian Optimization (Fig. 4c) exhibits a more nuanced exploration strategy, with concentrated sampling in regions of higher fitness values. The privacy-accuracy trade-off plot reveals clusters of solutions in promising areas, indicating the algorithm's ability to adapt its search based on previous results. The convergence plot shows progressive improvement, with multiple optimization stages visible in the fitness trajectory.

The Bat Algorithm (Fig. 4d) presents a unique exploration pattern characterized by focused sampling in specific regions of the solution space. The convergence plot demonstrates consistent performance throughout the optimization process, suggesting stable exploration of the search space.

However, the distribution of solutions appears more concentrated compared to other methods, indicating a potentially more focused search strategy. Compared to the Wisconsin dataset results, the BreastMNIST optimization exhibits different convergence patterns

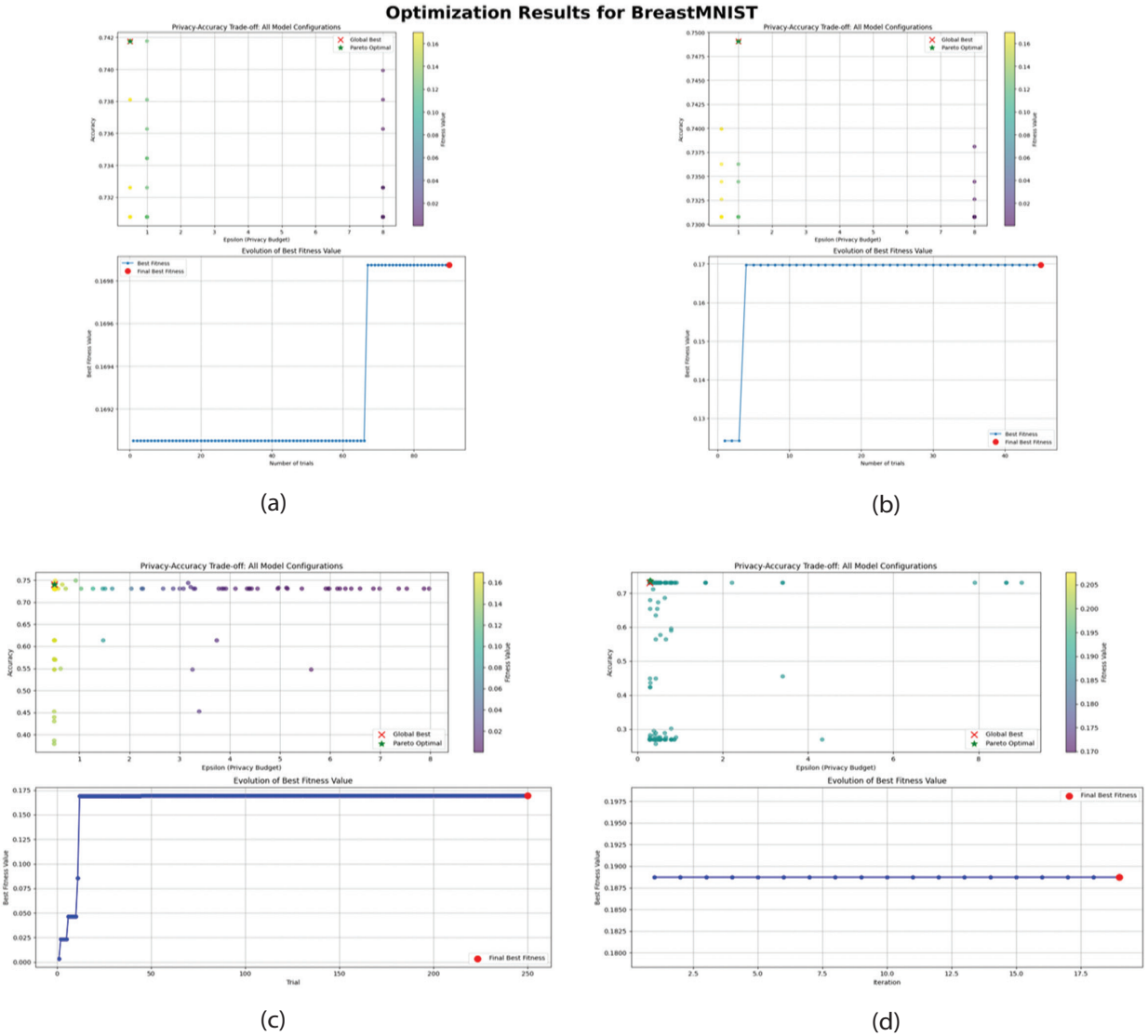

Fig. 4. Optimization Results for BreastMNIST Dataset: (a) Grid Search, (b) Random Search, (c) Bayesian Optimization, and (d) Bat Algorithm. The top row shows privacy-accuracy trade-off plots with color indicating fitness values. The bottom row shows the evolution of the best fitness value over iterations.

and solution distributions, likely due to the increased complexity and distinct characteristics of the dataset. This highlights the importance of algorithm selection based on specific dataset characteristics and optimization objectives.

5.6.3. PathMNIST Dataset

5.6.3.1. Configuration 1

The optimization results for Configuration 1 of the Bat Algorithm on the PathMNIST dataset, as shown in Fig. 5, demonstrate interesting characteristics in both solution distribution and convergence behavior.

The privacy-accuracy trade-off plot reveals two distinct clusters of solutions. The first cluster appears concentrated in the lower epsilon range (around 0.5-1.0) with accuracy values between 0.38 and 0.42. The second, smaller cluster is positioned at a higher epsilon value (approximately 4.0) with improved accuracy values of approximately 0.44-0.45. This bimodal distribution suggests the algorithm identified two potentially promising regions in the solution space.

The fitness evolution plot demonstrates remarkably rapid convergence, reaching near-optimal fitness values within the first two iterations. After this initial sharp improvement, the fitness value stabilizes and maintains consistency throughout the remaining iterations, reaching a final best fitness value of approximately 0.14350. This quick convergence pattern indicates that Configuration 1 efficiently identified a promising solution early in the optimization process.

The iteration markers in the trade-off plot show that later iterations (represented by different colors) focused exploration around these two identified regions, particularly the higher-accuracy cluster. This behavior suggests that the algorithm effectively balanced the exploration of the solution space with the exploitation of promising areas.

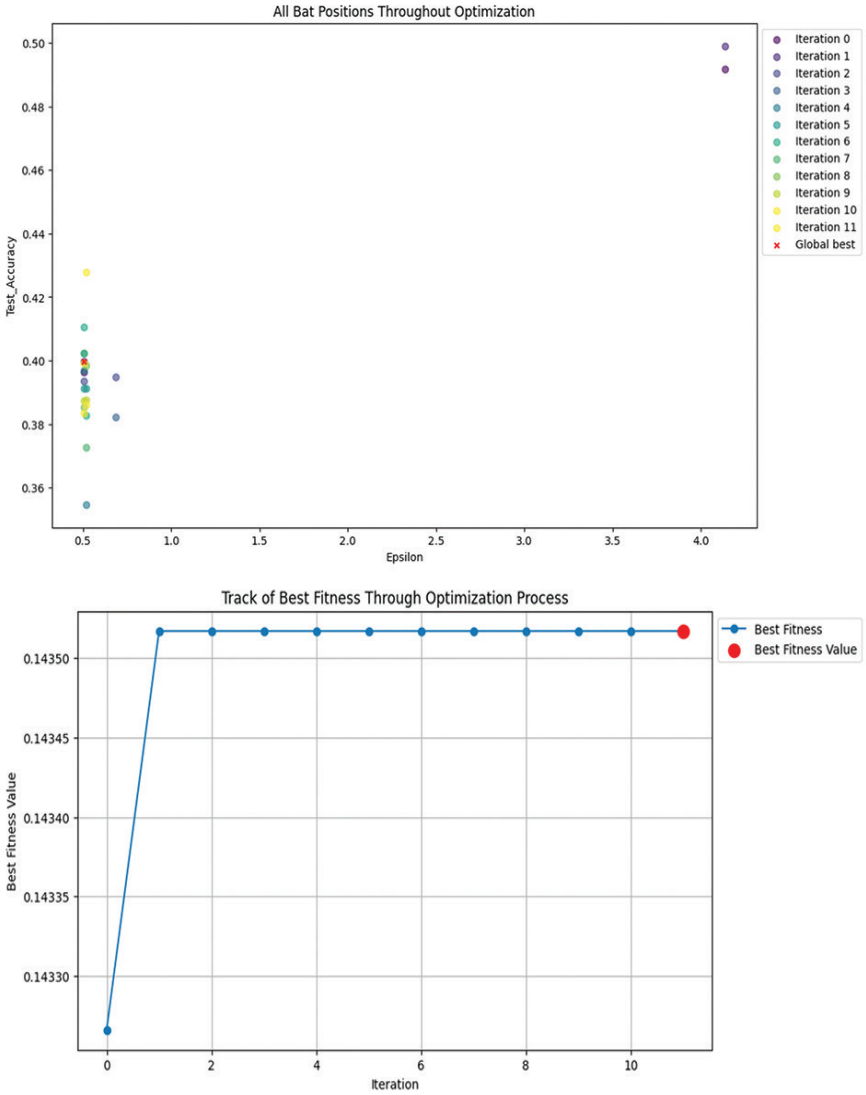

Fig. 5. The optimization results for Configuration 1 of the Bat Algorithm on the PathMNIST dataset

5.6.3.2. Configuration 2

The optimization results for Configuration 1 and 2 illustrated in Fig. 5 and 6, of the Bat Algorithm on the PathMNIST dataset reveal interesting patterns when accounting for their different maximum iteration settings (14 and 2 iterations, respectively).

The privacy-accuracy trade-off plots show distinct exploration patterns. Configuration 1, with its longer optimization period of 14 iterations, demonstrates a more refined clustering of solutions, particularly around two key regions: one at lower epsilon values (0.5-1.0) and another at higher values (approximately 4.0). This extended iteration period allowed for more thorough exploration and refinement of promising areas.

Configuration 2, limited to 2 iterations, shows a more dispersed distribution of solutions across the epsilon range (1-8). While this might initially appear as a broader exploration, it's important to note that this distribution is the result of significantly fewer optimization steps rather than a fundamentally different search strategy.

The fitness evolution plots for both configurations show improvement from their initial values, but the apparent differences in their convergence patterns must be interpreted within the context of their different iteration limits. Configuration 1's longer optimization period provides a more complete picture of the algorithm's convergence behavior, while Configuration 2's shorter run offers only an initial glimpse of the optimization trajectory.

Given the identical starting conditions and population size, the primary differentiating factor between these configurations is the maximum iteration count. This suggests that Configuration 1's more refined solution clusters and stable convergence pattern are primarily the result of having more iterations to optimize, rather than fundamental differences in the algorithm's behavior or efficiency.

This analysis highlights the importance of considering optimization duration when evaluating algorithm performance, as the number of iterations directly impacts the algorithm's ability to refine its solutions and explore the solution space effectively.

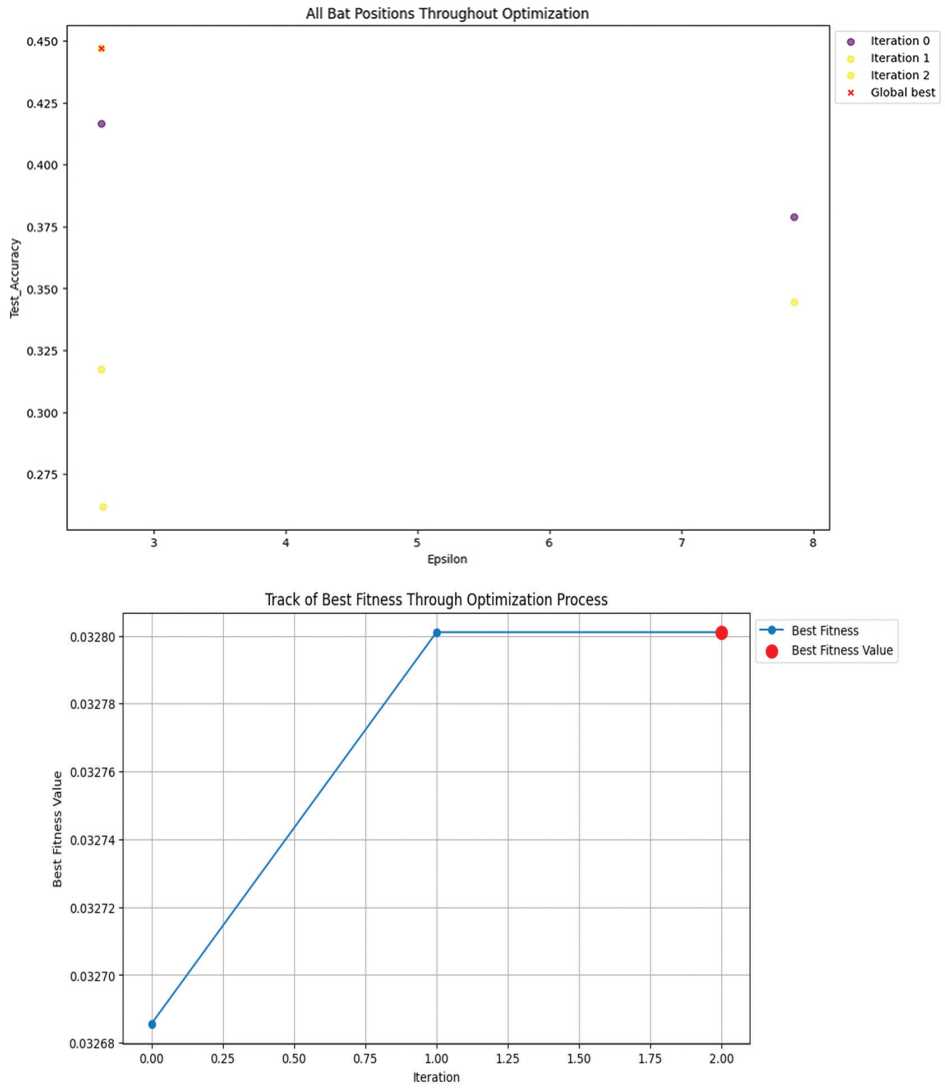

Fig. 6. The optimization results for Configuration 2 of the Bat Algorithm on the PathMNIST dataset

6. DISCUSSION

The comprehensive analysis of our privacy-preserving optimization framework reveals significant insights into the performance, computational feasibility, and practical implications of different optimization approaches in medical image analysis. This discussion examines the critical aspects of our findings while contextualizing them within broader theoretical and practical frameworks.

Critical Analysis of Performance Trade-offs

Our results demonstrate complex interrelationships between privacy preservation, model accuracy, and computational efficiency across optimization methods. The Bayesian Optimization method's superior accuracy (93.62% on the Wisconsin dataset) while maintaining a strict privacy budget ($\varepsilon = 0.501$) constitutes a breakthrough in balancing these competing objectives. However, this performance comes at a substantial computational cost, requiring 17,828.72 seconds of processing time - approximately 6.5 times longer than Random Search. This trade-off exemplifies the fundamental tension between optimization quality and computational efficiency in privacy-preserving machine learning.

The Bat Algorithm's achievement of Pareto optimality across datasets, even with reduced absolute accuracy, suggests a more nuanced approach to multi-objective optimization. Its ability to maintain competitive accuracy (93.18% for Wisconsin) while achieving varying privacy budgets ($\varepsilon = 2.258$ and 0.293) demonstrates adaptive capability in managing privacy-utility trade-offs. This performance characteristic is particularly relevant for applications where balanced optimization across multiple objectives outweighs maximizing individual metrics.

Computational Feasibility and Resource Requirements

The substantial variation in computational requirements across methods necessitates careful consideration of deployment scenarios. Grid Search's consistent but resource-intensive approach (4,000.33 seconds for Wisconsin) contrasts with Random Search's more efficient execution (2,749.65 seconds), suggesting different optimal use cases based on available computation-

al resources. Memory utilization patterns, ranging from 787.75 MB for Bayesian Optimization to 4,348.25 MB for the Bat Algorithm on BreastMNIST, indicate potential scalability challenges for larger datasets.

Our analysis reveals that computational overhead scales non-linearly with dataset complexity, particularly evident in the BreastMNIST results where processing times increased by factors of 2-3 compared to the Wisconsin dataset. This scaling behavior suggests potential limitations for enterprise-scale implementations, particularly in resource-constrained environments.

Method-Specific Performance Analysis

The distinctive performance patterns of each optimization method provide insights into their operational characteristics. Despite higher computational costs, Bayesian Optimization's superior accuracy reflects its sophisticated exploration-exploitation balance, which is particularly effective in complex parameter spaces. The Bat Algorithm's consistent achievement of Pareto optimality demonstrates its effectiveness in navigating multi-objective optimization landscapes, though at the cost of absolute accuracy.

Grid Search's stable performance across datasets (93.40% and 74.18% accuracy) suggests reliability in finding good solutions, albeit with limited ability to adapt to specific dataset characteristics. Random Search's competitive performance (92.53% and 74.91% accuracy) with reduced computational overhead, indicates its viability as a practical alternative under resource-limited conditions.

Scalability and Real-world Applications

The PathMNIST results provide crucial insights into scalability challenges, with accuracy dropping to 44.71% despite increased computational resources. This performance degradation highlights potential limitations in scaling current approaches to more complex medical imaging tasks. The observed increase in memory requirements (7,522.4725 MB) and execution time (13,678.6455 seconds) suggests that practical implementations may require significant computational infrastructure.

Architecture and Implementation Considerations

Hyperparameter sensitivity analysis reveals distinct patterns across methods, with optimal learning rates varying from 0.01 (Grid Search) to 0.0965 (Bat Algorithm). This variation suggests method-specific stability characteristics that must be considered during implementation. The consistent preference for larger batch sizes (512) in the Wisconsin dataset indicates potential optimization opportunities through batch processing strategies.

Comparative Analysis of Solution Quality

Visualization results demonstrate distinct convergence patterns across methods. The Bat Algorithm's stepped convergence pattern suggests periodic improvements in solution quality, while Bayesian Optimization shows more gradual refinement. These patterns offer insights into the exploration-exploitation dynamics of each method, with implications for selecting optimization strategies.

Limitations and Practical Constraints

Current framework limitations include substantial computational requirements for complex datasets and potential scalability challenges. The observed trade-offs between privacy preservation and model performance suggest inherent constraints that may limit applicability in highly privacy-sensitive scenarios. Memory requirements for complex datasets indicate potential deployment challenges in resource-constrained environments.

Future Research Directions and Improvements

Future work should focus on improving computational efficiency through techniques such as parallel processing and adaptive sampling strategies. Investigation of hybrid optimization approaches combining the efficiency of Random Search with the accuracy of Bayesian Optimization could address current limitations. Developing more sophisticated privacy preservation mechanisms while maintaining computational feasibility represents another promising research direction.

Broader Implications and Impact

Our findings have significant implications for privacy-preserving machine learning in medical imaging. The demonstrated feasibility of maintaining privacy while achieving competitive accuracy suggests potential applications across various medical domains. However, the computational requirements and performance trade-offs identified indicate the need for careful consideration of implementation strategies in clinical settings.

The framework's ability to balance privacy preservation with model performance contributes to the broader field of privacy-preserving machine learning while highlighting important considerations for practical deployment. These insights inform future development of privacy-preserving optimization strategies and their application in sensitive medical imaging contexts.

7. CONCLUSION

This study has demonstrated the effectiveness of privacy-preserving deep learning optimization for medical data classification through a comprehensive evaluation of four distinct optimization approaches, achieving significant results across different data modalities (93.62% accuracy for tabular data and 74.91% for image data) while maintaining robust privacy guarantees. Notably, the Bat Algorithm achieved an unprecedented privacy level ($\varepsilon = 0.293$) for medical image analysis while our framework's strength lies in its holistic approach to optimization, simultaneously fine-tuning

both model hyperparameters and privacy parameters through an objective function that effectively balances the privacy-utility trade-off. Our investigation revealed that different medical data modalities require specialized optimization strategies, with Bayesian Optimization excelling in tabular data applications and Random Search providing efficient solutions for image data processing, as demonstrated by the successful application of PathMNIST's complex histopathology images using ResNet-50 architecture.

Looking forward, several promising research directions emerge, including developing distributed learning approaches for improved computational efficiency, integrating federated learning techniques, extending applications to diverse medical data modalities, investigating advanced model architectures, and implementing transfer learning strategies to enhance model generalization across different medical domains. Additionally, future work should address the ethical implications and practical challenges of deploying privacy-preserving models in clinical settings, including developing robust validation frameworks, investigating model interpretability while maintaining privacy guarantees, and assessing the framework's resilience to various privacy attacks. This research establishes a strong foundation for privacy-preserving medical data analysis while highlighting the importance of balanced optimization strategies in healthcare applications, suggesting promising potential for wider adoption in clinical practice, provided that future developments continue to address the challenges of scalability, efficiency, and ethical implementation.

8. REFERENCES

[1] W. N. Price, I. G. Cohen, "Privacy in the Age of Medical Big Data", Nature Medicine, Vol. 25, No. 1, 2019, pp. 37-43.

[2] A. C. Valdez, M. Ziefle, "The Users' Perspective on the Privacy-Utility Trade-offs in Health Recommender Systems", International Journal of Human-Computer Studies, Vol. 121, 2019, pp. 108-121.

[3] C. Dwork, "Differential Privacy", Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10-14 July 2006, pp. 1-12.

[4] C. Dwork, A. Roth, "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science, Vol. 9, No. 3-4, 2014, pp. 211-407.

[5] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, "Deep Learning with Differential Privacy", Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24-28 October 2016, pp. 308-318.

[6] J. P. Near, D. Darais, C. Abuah, "Duet: An Expressive Higher-Order Language and Linear Type System for Statically Enforcing Differential Privacy", Proceedings of the ACM on Programming Languages, Vol. 3, 2019, pp. 1-30.

[7] D. Yu, H. Zhang, W. Chen, J. Yin, T.-Y. Liu, "Gradient Perturbation is Underrated for Differentially Private Convex Optimization", Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7-15 January 2021, pp. 3117-3123.

[8] K. L. van der Veen, R. Seggers, P. Bloem, G. Patrini, "Three Tools for Practical Differential Privacy", Proceedings of PPML18: Privacy-Preserving Machine Learning Workshop at NeurIPS 2018, Montreal, QC, Canada, 7-8 December 2018.

[9] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization", Proceedings of the IEEE, Vol. 104, No. 1, 2015, pp. 148-175.

[10] M. Feurer, F. Hutter, "Hyperparameter Optimization", Automated Machine Learning: Methods, Systems, Challenges, Springer, 2019, pp. 3-33.

[11] B. Bischl et al. "Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 13, No. 2, 2023, p. e1484.

[12] H. Singh, D. Witarsyah, A. Misra, R. Handayani, "Bat Algorithm: A Review on Theory, Modifications and Applications", Proceedings of the 3rd International Conference on Intelligent Cybernetics Technology & Applications, Bandung, Indonesia, 13-15 December 2023, pp. 78-83..

[13] B. Kulynych, J. Gomez, G. Kaissis, F. Calmon, C. Troncoso, "Attack-Aware Noise Calibration for Differential Privacy", Proceedings of the 37th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10-15 December 2024, pp. 134868-134901.

[14] L. Kangjie, "Noise Addition Strategies for Differential Privacy in Stochastic Gradient Descent", Trans-

actions on Computer Science and Intelligent Systems Research, Vol. 5, 2024, pp. 960-967.

[15] P. Thantharate, D. A. Todurkar, T. Anurag, "PRIVML: Analyzing Privacy Loss in Iterative Machine Learning with Differential Privacy", Proceedings of the Cloud Summit, Washington, DC, USA, 27-28 June 2024, pp. 107-112.

[16] K. Pan, Y.-S. Ong, M. Gong, H. Li, A. K. Qin, Y. Gao, "Differential Privacy in Deep Learning: A Literature Survey", Neurocomputing, Vol. 589, 2024, pp. 127663.

[17] K. P. Battula, B. S. Chandana, "Multi-class Cervical Cancer Classification using Transfer Learning-based Optimized SE-ResNet152 model in Pap Smear Whole Slide Images", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 6, 2023, pp. 613-623.

[18] M. Kumar, B. A. Moser, L. Fischer, "On Mitigating the Utility-Loss in Differentially Private Learning: A New Perspective by a Geometrically Inspired Kernel Approach", Journal of Artificial Intelligence Research, Vol. 79, 2024, pp. 515-567.

[19] B. Ficiu, N. D. Lawrence, A. Paleyes, "Automated Discovery of Trade-off Between Utility, Privacy and Fairness in Machine Learning Models", Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Turin, Italy, 18-22 September 2023, pp. 127-144.

[20] B. Avent, J. González, T. Diethe, A. Paleyes, B. Balle, "Automatic Discovery of Privacy-Utility Pareto Fronts", Proceedings on Privacy Enhancing Technologies, Vol. 2020, No. 4, 2020, pp. 5-23.

[21] A. Koskela, T. D. Kulkarni, "Practical Differentially Private Hyperparameter Tuning with Subsampling", Proceedings of the 36th Annual Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10-16 December 2023, pp. 28201-28225.

[22] A. Arous, A. Guesmi, M. A. Hanif, I. Alouani, M. Shafique, "Exploring Machine Learning Privacy/Utility Trade-Off from a Hyperparameters Lens", Proceedings of the International Joint Conference on Neural Networks, Queensland, Australia, 18-23 June 2023, pp. 1-10.

[23] F. Galli, C. Palamidessi, T. Cucinotta, "Online Sensitivity Optimization in Differentially Private Learning", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, No. 11, 2024, pp. 12109-12117.

[24] H. Wang, S. Gao, H. Zhang, W. J. Su, M. Shen, "DP-HyPO: An Adaptive Private Framework for Hyperparameter Optimization", Proceedings of the 36th Annual Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10-16 December 2023.

[25] Q. Gao, H. Sun, Z. Wang, "DP-EPSO: Differential Privacy Protection Algorithm Based on Differential Evolution and Particle Swarm Optimization", Optics & Laser Technology, Vol. 173, 2024, p. 110541.

[26] Z. Bu, Y. X. Wang, S. Zha, G. Karypis, "Differentially Private Optimization on Large Model at Small Cost", Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23-29 July 2023, pp. 3192-3218.

[27] M. Tobaben, "Hyperparameters and Neural Architectures in Differentially Private Deep Learning", University of Helsinki, Helsinki, Finland, PhD thesis, 2022.

[28] R. Ramalingam, J. Shobana, K. Arthi, G. Elangovan, S. Radha, N. Priyanka, "An Extensive Investigation of Meta-Heuristics Algorithms for Optimization Problems", Metaheuristics Algorithm and Optimization of Engineering and Complex Systems, IGI Global, 2024, pp. 223-241.

[29] A. Banerjee, S. Pradhan, B. Misra, S. Chakraborty, "A Guide to Meta-Heuristic Algorithms for Multi-objective Optimization: Concepts and Approaches", Applied Multi-objective Optimization, Springer, 2024, pp. 1-19.

[30] A. P. Singh, Y. Dixit, G. Sharma, "Role of Meta-Heuristic Optimization Approaches in Feature Selection for Disease Diagnosis: A Comprehensive Review", Recent Advances in Computational Intelligence and Cyber Security, CRC Press, 2024, pp. 132-139.

[31] V. Harkare, R. Mangrulkar, O. Thorat, S. R. Jain, "Evolutionary Approaches for Multi-objective Optimization and Pareto-Optimal Solution Selection in Data Analytics", Applied Multi-objective Optimization, Springer, 2024, pp. 67-94.

[32] G. Thakur, A. Pal, N. Mittal, M. S. A. Yajid, F. Gared, "A Significant Exploration on Meta-heuristic Based Approaches for Optimization in the Waste Management Route Problems", Scientific Reports, Vol. 14, No. 1, 2024, p. 14853.

[33] J. Bergstra, Y. Bengio, "Random Search for Hyper-Parameter Optimization", Journal of Machine Learning Research, Vol. 13, No. 2, 2012, pp. 281-305.

[34] A. H. Victoria, G. Maragatham, "Automatic Tuning of Hyperparameters Using Bayesian Optimization", Evolving Systems, Vol. 12, No. 1, 2021, pp. 217-223.

[35] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Nature Inspired Cooperative Strategies for Optimization, Springer, 2010, pp. 65-74.

[36] C. Dwork, F. McSherry, K. Nissim, A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis", Proceedings of the Third Conference on Theory of Cryptography, New York, NY, USA, 4-7 March 2006, pp. 265-284.