Privacy-Preserving EEG Data Frameworks for Brain-Computer Interfaces

Shouvik Paul

Master of Science in Computer Science

(Specialization in Artificial Intelligence)

Department of Computer Science

Faculty of Graduate Studies

Lakehead University

Thunder Bay, Ontario, Canada

September 15, 2024

A thesis submitted to Lakehead University in partial

fulfillment of the requirements of the degree of

Master of Science in Computer Science

with a specialization in Artificial Intelligence.

Statement of Originality

This is to certify that, to the best of my knowledge, the content of this thesis is my own work and has been carried out under the general supervision of my supervisor. This thesis has not been submitted to any other Institute for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Shouvik Paul

Signature: _____ Date:

Supervisory Committee

Supervisor:

Dr. Garima Bajwa

Department of Computer Science

Lakehead University

Thunder Bay, Ontario, Canada

Signature: _____ Date:

Internal Examiner:

Dr. Thiago E. Alves de Oliveira

Department of Computer Science

Lakehead University

Thunder Bay, Ontario, Canada

Signature: _____ Date:

External Examiner:

Dr. Muhammad Asaduzzaman

School of Computer Science

University of Windsor

Windsor, Ontario, Canada

Signature: _____ Date:

Acknowledgements

My M.Sc. program has been a fun, challenging, and an everlasting experience. Looking back, I realize that the relationships I have built along the way were what made this time truly special. I am truly grateful to have had the chance to work alongside and learn from outstanding researchers. I am also thankful for the friendships that formed and deepened during this time.

First and foremost, I am highly indebted to Dr. Garima Bajwa, my esteemed supervisor. I feel lucky to have benefited from her mentorship. I would like to show my heartfelt gratitude towards her for reshaping my research experience by providing constant feedback. Dr. Bajwa's guidance, support, and expertise have been invaluable throughout the journey of this thesis. Her insightful encouragement, and unwavering commitment have truly enriched my research experience.

I thank Dr. Thiago E. Alves de Oliveira and Dr. Muhammad Asaduzzaman for being a part of the committee for this thesis and offering their support and assistance.

I am also grateful to Dr. Todd Randall, Dr. Sabah Mohammed, Dr. Amin Safaei, Dr. Abedalrhman Alkhateeb, and Dr. Xing Tan for their encouragement and support. My academic experience has been enhanced by their assistance.

I thank Dr. Sourav De, who inspired me to do research after recognizing my potential

and I appreciate his dedicated efforts, as his encouragement and faith in my abilities have been climacteric to my academic success. His guidance during my Bachelor's thesis helped me to expand avenues and pursue an M.Sc. Additionally, I thank my Bachelor's thesis co-supervisor, Sandip Dey, whose mentorship was crucial to my success and provided a strong foundation for my M.Sc. journey.

Many thanks to all from Brain-Machine Interfaces Lab (BMI Lab), and the Smart Health Lab who always filled me with new energy and motivation to perform.

Handling administrative tasks would have been much more difficult without the help of Dr. Rachael Wang, Sheila Walsh, Patricia-Anne Sokoloski, Yvonne Elcheson, Allison Whately-Doucet, and Maegen Lavallee. I also extend my thanks to Darryl Willick from the Technology Services Centre for his indispensable technical assistance with the setup and maintenance of my Lakehead University High Performance Computing Centre (LUHPCC) account.

I am immensely thankful to the Faculty of Graduate Studies, the Vector Institute, and the Government of Ontario for their generous support through various scholarships including Vector Scholarship in Artificial Intelligence and Ontario Graduate Scholarship (OGS). These scholarships have greatly enabled me to pursue my research with dedication and have been significant in my academic progress. I would also like to acknowledge the opportunities I have been fortunate to have as a Graduate Teaching Assistant and Research Assistant. These roles have not only contributed to my academic growth but have also provided invaluable hands-on experience and insights into the field of computer science.

Many thanks to my friends, Vraj, Tuhin, Raj, Gaurav, Yash, Parth, and Erfan, for making my journey both joyful and meaningful by being a part of it. I also appreciate Varuna,

Manmeet, Aditya, Girijesh, Asif, and Alankrit for their emotional and mental support, which kept me determined in my research while bringing fun into the experience.

I will be always grateful to my parents, Shyamal Krishna Paul and Shikha Paul, who have been my biggest cheerleaders and who offered me all their love and enthusiastic support no matter what I set out to do. Thank you for being there for me and for believing in me.

Lastly, my heartfelt thanks go to my role model, my elder sister, Sharmistha Paul Dutta, her wonderful husband, Tamal Dutta, and their adorable daughter, Tannistha Dutta. I am incredibly grateful for their unwavering support and love throughout this journey. Witnessing Sharmistha's research journey has enlightened and inspired me the most, especially during the challenging times of my M.Sc. Loving thanks to my little niece, Tannistha, whose presence has brought so much joy and light into my life. The bond we share is truly special, and her smiles have been a source of happiness and inspiration. Thank you all for making my life purposeful and full of motivation along the way.

Thank you all for your kind contributions that made this journey an everlasting experience for me.

Abstract

Non-invasive brain-computer interface (BCI) systems rely on brainwave activity, predominantly captured through Electroencephalography (EEG), to facilitate seamless interactions with digital platforms. Throughout its development, EEG-driven BCIs have touched industries as diverse as entertainment, healthcare, and cybersecurity. However, despite improvements in functionality and accuracy, the critical issue of securing the vast amounts of sensitive EEG data collected by these systems has remained largely overlooked, posing significant privacy risks. While techniques like data anonymization, encryption, masking, and perturbation aim to protect privacy, they often degrade the quality of the data and fail to fully eliminate the risk of re-identification. In response, we have developed multiple privacy-preserving frameworks: a quantum-inspired Differential Privacy-based generative model, a Rényi Differential Privacy (RDP) based Federated model, and a privacy-adaptive Federated Split Learning framework, featuring Secure Aggregation and Autoencoders. Each framework is designed to generate synthetic EEG data that comply with privacy protection standards while ensuring robust data utility for downstream analysis. Modern defenses that focus on privacy frequently sacrifice performance or depend on large amounts of external data, which can limit their practicality. Our approach not only mitigates these limitations, but also significantly strengthens defenses against membership inference and reconstruction threats.

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

Brain-computer interface (BCI) technology is rapidly evolving and has allowed the creation of intricate neuroprosthetics and applications in the fields of healthcare, entertainment and even personal security. BCIs based on electroencephalography (EEG) register electrical activity in the brain and provide a bridge between human cognitive states and computerized procedures. EEG recordings were first demonstrated by Hans Berger in 1924 [1, 2] and have since provided a new way of interacting and controlling the human brain [3, 4, 5, 6]. BCIs were developed in the 1970s with the intention of capturing and analyzing brain signals from a user and translating them into actions to control external devices [4]. Brain-machine interface (BMI) systems, in turn, use only embedded sensor signals for input, excluding external equipment [4].

1.1 Landscape: EEG Data Privacy

BCI technologies that use EEG data range from cutting-edge brain-controlled prosthetics for people with severe motor disabilities to unique real-time mood forecasting systems in personalized entertainment and mental health coaching. All of these developments illustrate the increased use of EEG data across domains and represent a major advancement in user interface techniques [7, 8, 9].

The increasing reliance on EEG data for emerging BCI applications introduces significant privacy issues. Recent EEG data breaches have exposed many vulnerabilities and threats associated with current data protection measures, leading to identity theft, privacy violations, and unwarranted surveillance. Many researchers show that wearable BCI devices pose a significant risk to user privacy and could be detrimental if brain data is not adequately encrypted [7, 8, 10]. For example, unauthorized access to EEG data can expose user's mental health or emotional states to targeted advertising, or more harmful ones, such as coercion or manipulation [9, 11].

Beltran et al. [12] identified various cumulative threats indicating that noise-based attacks can occur in BCI systems that rely on P300 waves. To mitigate these threats, increasing the number of electrodes to monitor the EEG output of such systems is advantageous. Bernal et al. [13] found that these threats could also influence the natural activity of neurons. Their research demonstrated the use of neuronal scanning attacks and neuronal flooding attacks within a neuronal simulator.

The security landscape is further made complex by the integration of BCIs with other technologies such as cloud computing and IoT devices. Thus, it is necessary to protect privacy at the key stages of data collection, transmission, and storage [8, 10, 14]. The ongoing efforts to ensure the anonymity of EEG data to preserve its utility for BCI applications include techniques such as encryption, anonymization, and differential privacy preservation principles [7, 8, 15, 16].

1.2 Motivation

This thesis is mainly motivated by the need to address the fundamental problem of striking a balance between strong privacy guarantees and data value. Even state-of-the-art solutions such as differential privacy still have room for development. Traditional privacy-preserving techniques sometimes fail to preserve the usefulness of data while applying adequate privacy protections [16, 17, 18, 19]. More specifically, any privacy-preserving method must guarantee minimum loss in data quality in the context of BCI applications to facilitate trustworthy and insightful analysis. The goal of this thesis is to provide privacy assurance while maintaining the usefulness of EEG data through the development of privacy-preserving frameworks.

1.3 Objectives

The primary objective of this thesis is to develop and validate novel frameworks that ensure the privacy-preserving generation and processing of EEG data in brain-computer interfaces (BCIs), while maintaining high data utility. This thesis specifically seeks to achieve the following goals:

- To develop new privacy-preserving frameworks through the generation of synthetic EEG data that closely resemble the original data for subsequent BCI operations while addressing the gaps of privacy concerns.
- To systematically explore the privacy-utility trade-off by quantifying how varying privacy budgets ϵ and the amount of noise added affect the usability and accuracy of generated synthetic EEG data in our proposed frameworks.
- To develop secure communication protocols ensuring that sensitive EEG data remains protected even during network communication without exposing raw data.
- To validate the effectiveness of the proposed privacy mechanisms by testing them against adversarial attacks and comparing them to existing state-of-the-art techniques.

The section provides details on each of the objectives achieved in the subsequent chapters and the specific contributions.

1.4 Thesis Contributions

Focused on addressing the limitations of existing approaches, we reconfigure the architecture of generative adversarial networks (GANs) and present various novel privacy-preserving frameworks. Our approach seeks to achieve an equilibrium between privacy and utility in EEG-based BCI applications. The primary objective is to generate synthetic EEG data that preserve the fundamental attributes of the original EEG dataset while substantially mitigating the risk of privacy violations.

The main contributions of this study are:

- **Quantum inspired noise dynamics integration:** Our study is one of the pioneering efforts to integrate quantum inspired noise dynamics into the GAN training process by emulating the phenomenon of quantum decoherence. Dynamically adjustable noise not only meets, but significantly improves the criteria for differential privacy. It effectively reduces the risks of information leakage by demonstrating a transition from a quantum state with high uncertainty to a classical state with reduced uncertainty across multiple epochs in the training, which starts with a high noise level that gradually decreases.

- **Advanced privacy budget management:** To comply with the quantum uncertainty principle, the privacy budget is calculated by adding the decay factor to the total amount of the privacy budget accumulated during training epochs. The contribution of each epoch to the total privacy budget decreases as the model stabilizes, resulting in a more predictable privacy budget which is manageable for practical use where long-term data security is very important.

- **Privacy attacks evaluation:** We evaluate our system against membership inference in white-box and black-box configurations, as well as reconstruction attacks. These

tests confirm the model's effectiveness in masking real data sources. Our model is one of the early works in the field showing resilience to reconstruction attacks on EEG data.

- **Federated learning with Spiking Neural Networks:** We built a Federated Spiking GAN framework combining Spiking Neural Networks (SNNs) and Generative Adversarial Networks (GANs) with an advanced noise model for EEG data synthesis. The temporal dynamics of SNNs improve the authenticity of synthetic EEG data, while the ANN-based discriminator ensures accuracy for utility (classification). Incorporating a temporally correlated noise model with Renyi Differential Privacy (RDP) provides reliable privacy and high data utility. Our federated learning architecture enhances privacy by training models decentralized across client nodes without raw data exchange. Our assessments validate the privacy-utility trade-offs and effectiveness of the framework.

- **Federated Split Learning with anonymized latent variables :** We built a unique a framework using Hierarchical encoder-decoder networks, GFlowNet, and Federated Split Learning (FSL) for privacy-preserving EEG data creation. It ensures privacy by exchanging only anonymized latent variables with the server, keeping raw EEG data on the client. The hierarchical encoder-decoder network uses a multi-level latent space with RDP for privacy-utility balance. Secure Aggregation lets the server analyze only aggregated data, thus protecting individual contributions. Lastly, GFlowNet generates synthetic EEG data, ensuring temporal and spatial consistency with privacy. Our method offers better privacy-utility trade-offs and maintains high data quality for sensitive EEG data compared with existing models.

1.5 Thesis Roadmap

This dissertation is organized as follows:

- **Chapter 2** provides a comprehensive overview of foundational concepts and theories

essential for understanding the subsequent analysis. It provides the necessary context for the research discussed in the later chapters. It also reviews and discusses existing literature and previous research pertinent to the thesis topics. It highlights key findings and identifies gaps that the current research aims to address.

- **Chapters 3, 4, and 5** explore various techniques designed to preserve the privacy of EEG data. Each chapter presents and analyzes different methods and approaches to safeguarding sensitive information.

- **Chapter 6** concludes the dissertation by summarizing the key findings, discussing their implications, and reflecting on the contributions of the research.

- **Chapter 7** outlines potential directions for future research and proposes areas for further investigation.

2

Background and Literature Review

2.1 Generative Models:

2.1.1 Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. in 2014 [20], represent a significant innovation in machine learning by facilitating the generation of highly realistic synthetic data. GANs are composed of two competing neural networks: generator (G) and discriminator (D). The generator's objective is to create data that appear indistinguishable from real data, while the discriminator's task is to differentiate between real and synthesized data. This adversarial interaction encourages both networks to continuously improve, resulting in the generation of highly realistic data.

Mathematically, GANs operate on a minimax game principle. The generator (G) gen-

erates data samples $G(z;\theta_g)$ from a noise distribution $z \sim p_z(z)$, while the discriminator (D) evaluates these samples to determine their authenticity. The value function $V(G,D)$ defining this adversarial game is expressed as [20]:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] \ + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.1}$$

In this setup, $D(x)$ indicates the probability that x is a real sample. The generator (G) aims to minimize $\log(1-D(G(z)))$, whereas the discriminator (D) seeks to maximize $\log D(x) + \log(1-D(G(z)))$ [20].

The training process alternates updates to the generator and discriminator parameters, θ_g and θ_d, respectively, typically employing stochastic gradient descent. To improve training dynamics, the generator can maximize $\log D(G(z))$ rather than minimizing $\log(1-D(G(z)))$, providing more robust gradients during initial training phases:

$$\max_G E_{z\sim p_z(z)}[\log D(G(z))] \tag{2.2}$$

This adversarial framework ensures that the generator progressively enhances its ability to produce realistic data, while the discriminator refines its capability to distinguish real from synthetic data. Theoretically, with adequate capacity and training time, the generator's distribution p_g converges to the real data distribution p_{data} [20].

2.1.2 Sequential Generative Adversarial Networks (SGANs)

Sequential Generative Adversarial Networks (SGANs) [21] extend the GAN architecture to handle sequential data such as time series or text. SGANs are particularly useful for generating sequences that exhibit temporal dependencies, making them suitable for applications in natural language processing and bioinformatics.The primary innovation in SGANs is the integration of recurrent neural networks (RNNs) or long short-term memory (LSTM) networks within the generator and discriminator to capture temporal dependencies. The ob-

jective function remains similar to the original GAN formulation but is adapted to account for sequential data:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x_{1:T})] \ + E_{z\sim p_z(z)}[\log(1 - D(G(z_{1:T})))] \tag{2.3}$$

where $x_{1:T}$ denotes a sequence of data points, and $G(z_{1:T})$ represents the generated sequence [21].

2.1.3 Wasserstein Generative Adversarial Networks (WGANs)

Wasserstein GANs (WGANs) introduce a novel objective based on the Earth Mover's (Wasserstein-1) distance, which enhances training stability and mitigates mode collapse. The WGAN objective is defined as [22, 23, 24]:

$$\min_G \max_{D\in\mathcal{D}} E_{x\sim p_{data}(x)}[D(x)] - E_{z\sim p_z(z)}[D(G(z))] \tag{2.4}$$

where $\mathcal{D}$ denotes the set of 1-Lipschitz functions. This approach encourages the discriminator to learn a meaningful metric for comparing real and generated samples [22].

To enforce the Lipschitz constraint, WGANs utilize a gradient penalty. This penalty is expressed as:

$$E_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{2.5}$$

where $\hat{x}$ is sampled uniformly along straight lines between pairs of points drawn from the real data distribution and the generator distribution. This technique ensures stable training and reduces issues such as vanishing gradients and mode collapse [23].

WGANs have been successfully applied in various fields, including addressing data imbalance in classification tasks. Bhatia and Dahyot [25] demonstrated the effectiveness of WGANs in enhancing classifier performance on imbalanced datasets by generating realistic synthetic samples that help balance the class distribution.

2.1.4 Conditional Generative Adversarial Networks (CGANs)

Conditional GANs (CGANs) extend the GAN framework to conditional settings, where both the generator and discriminator receive additional information y (e.g., class labels). The objective function for CGANs is [26]:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x|y)] \ + E_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{2.6}$$

By incorporating conditional information, CGANs can generate samples that adhere to specific conditions, making them suitable for a variety of applications, including image-to-image translation and data augmentation [26].

2.2 Privacy-Preserving Mechanisms

2.2.1 Differential Privacy (DP)

Differential Privacy (DP) [27] offers a framework to quantify and control the loss of privacy incurred when releasing information derived from a dataset. By ensuring that the removal or addition of a single data point does not significantly affect the outcome of any analysis, DP provides strong privacy guarantees that are crucial in the context of sensitive data such as EEG records. The integration of DP into data analysis processes, particularly in machine learning models, has become a cornerstone for developing privacy-preserving technologies, underscoring the importance of balancing data utility with privacy considerations.

Here, we define the core concepts and theorems essential for our proposed methodology:

Definition 2.1 (Differential Privacy): A randomized algorithm $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets $d, d' \in \mathcal{D}$ (differing by one element), and for all subsets $S \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(d') \in S] + \delta \tag{2.7}$$

where ϵ (epsilon) represents the privacy loss, δ (delta) is the probability of this guarantee not holding, essentially allowing for a small chance of failure [28].

Theorem 2.2 (Gaussian Mechanism): For any function $f : \mathcal{D} \to R^k$ with sensitivity Δf, and parameters ϵ, δ where $0 < \epsilon < 1$ and $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism adds noise with standard deviation $\sigma \geq \frac{c\Delta f}{\epsilon}$ to the output of f to ensure (ϵ, δ)-differential privacy [28].

Theorem 2.3 (Composition Theorem): For a series of independent randomized mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, each providing (ϵ, δ)-differential privacy, the sequence of mechanisms provides $(k\epsilon, k\delta)$-differential privacy [29].

Theorem 2.4 (Enchanced Composition Theorem): For any $\epsilon, \delta, \delta' > 0$, the class of ϵ-differentially private mechanisms satisfies $(\epsilon', k\delta + \delta')$-differential privacy under k-fold adaptive composition [27], where:

$$\epsilon' = \sqrt{2k\ln(1/\delta')}\epsilon + k\epsilon(e^{\epsilon} - 1) \tag{2.8}$$

This theorem provides a tighter bound on cumulative privacy loss when multiple differentially private mechanisms are composed [27].

2.2.2 Rényi Differential Privacy

Rényi Differential Privacy (RDP) [30] is a relaxation of the standard differential privacy definition that allows for tighter privacy guarantees when combining multiple differentially private mechanisms. RDP uses the Rényi divergence to quantify the privacy loss, providing a more refined analysis of privacy guarantees.

Definition 2.5 (Rényi Differential Privacy): A randomized algorithm $\mathcal{M}$ satisfies (α, ϵ)-Rényi differential privacy if for any two adjacent datasets $d, d' \in \mathcal{D}$, the Rényi divergence of order $\alpha > 1$ between the outputs of $\mathcal{M}$ on d and d' is at most ϵ. Formally,

$$D_{\alpha}(\mathcal{M}(d) \parallel \mathcal{M}(d')) \leq \epsilon \tag{2.9}$$

where the Rényi divergence $D_\alpha(P \parallel Q)$ for two probability distributions P and Q is defined as [30]:

$$D_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} \log E_{x \sim Q} \left[\left(\frac{P(x)}{Q(x)} \right)^\alpha \right] \tag{2.10}$$

This definition provides a framework that can offer tighter privacy bounds under composition compared to the traditional (ϵ, δ)-differential privacy [30].

Theorem 2.6 (Advanced Composition for RDP): If an algorithm satisfies (α, ϵ_i)-RDP for $i = 1, 2, \ldots, k$, then the composition of these algorithms satisfies $(\alpha, \sum_{i=1}^{k} \epsilon_i)$-RDP [30].

2.2.3 Local Differential Privacy

Local Differential Privacy (LDP) is a variant of differential privacy that ensures privacy at the level of individual data contributors before any aggregation occurs. This model is particularly relevant for decentralized data collection scenarios where the data provider does not fully trust the data collector.

Definition 2.7 (Local Differential Privacy): A randomized algorithm $\mathcal{M}$ satisfies ϵ-local differential privacy if for any two inputs $x, y \in \mathcal{X}$, and for all outputs $z \in \mathcal{Z}$,

$$\Pr[\mathcal{M}(x) = z] \leq e^\epsilon \Pr[\mathcal{M}(y) = z] \tag{2.11}$$

LDP ensures that the randomized response provided by each individual does not reveal too much about their input, thereby protecting their privacy [31, 32, 33].

Theorem 2.8 (Gaussian Mechanism for LDP): For a function $f : \mathcal{X} \to R$ with sensitivity Δf, the Gaussian mechanism adds noise drawn from a Gaussian distribution with mean zero and standard deviation $\sigma = \frac{\Delta f}{\epsilon}$ to the output of f to ensure ϵ-local differential privacy [31, 33, 34].

2.3 Quantum Principles

Quantum computing uses the principles of quantum mechanics to perform computations in ways that are fundamentally different from classical computing. Fundamental principles include quantum bits (qubits), superposition, entanglement, quantum interference, the uncertainty principle, quantum decoherence, and quantum tunneling [35, 36, 37]. Each of these principles has profound implications for computational efficiency and data security. We specifically incorporated quantum decoherence and the quantum uncertainty principle due to their direct applicability to enhancing privacy in data synthesis.

2.3.1 Quantum Uncertainty Principle

The Heisenberg Uncertainty Principle is a fundamental concept in quantum mechanics, stating that certain pairs of physical properties, such as position (x) and momentum (p), cannot be known precisely simultaneously. This principle is mathematically expressed as [38]:

$$\Delta x \Delta p \geq \frac{\hbar}{2} \tag{2.12}$$

where Δx and Δp are the uncertainties in position and momentum, respectively, and $\hbar$ is the reduced Planck constant. In quantum computing, the uncertainty principle ensures that attempts to measure quantum states can disturb those states. This principle is exploited in quantum cryptographic systems to detect eavesdropping, since any measurement by an eavesdropper would disturb the quantum states and reveal their presence [39].

2.3.2 Quantum Decoherence

Quantum decoherence [40] describes the loss of quantum coherence as the quantum system interacts with its environment, causing the quantum system to transition from a coherent superposition state to an incoherent classical mixture. This process can be described by the evolution of the density matrix ρ of the quantum state, given by the Lindblad master equation [41].

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H, \rho] + \sum_n \varsigma_n \left(L_n \rho L_n^\dagger - \frac{1}{2}\{L_n^\dagger L_n, \rho\} \right) \tag{2.13}$$

where H is the Hamiltonian of the system, L_n are the Lindblad operators representing interactions with the environment, and $\{\cdot, \cdot\}$ denotes the anticommutator. Decoherence is a major challenge in quantum computing as it can destroy delicate quantum states that are needed for computation. However, understanding and controlling decoherence is key to developing quantum error correction techniques [42].

2.4 Federated Learning

With Federated Learning (FL), several clients may train together to build a common model without sharing their raw data due to a decentralized ML technique. By keeping the data on the local servers, lowering the possibility of security breaches, and guaranteeing adherence to the data protection laws, FL protects the privacy of user data. FL uses a variety of distributed datasets to improve the generalization capabilities of models [43].

In FL, each client trains locally on their own private data and only shares updates to the model with a central server (that aggregates these updates in order to update the global model). The federated averaging algorithm that is at the heart of FL, is defined as [44, 45]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \sum_{i=1}^{N} \frac{n_i}{n} \mathbf{u}_i^t \tag{2.14}$$

where $\mathbf{w}_t$ represents the model parameters at round t, η is the learning rate, $\mathbf{u}_i^t$ is the update from client i, n_i is the number of samples held by client i, and n is the total number of samples across all clients.

FL simultaneously tackles challenges such as the high communication cost, data diversity or heterogeneity and security. In addition to healthcare, other applications, such as privacy-aware traffic flow prediction, have been explored using FL by Liu et al. [46]. Mothukuri et al. The research by Mothukuri et al. [47] provides an overall survey on the security and

private issues in FL, arguing that a strong privacy preservation method must be integrated into FL systems.

2.5 Existing Privacy Preserving Approaches

Despite numerous efforts, existing methods face several challenges and limitations that hinder their effectiveness in protecting sensitive EEG data.

Homomorphic encryption is a popular technique that allows computations to be performed on encrypted data without decrypting it [48]. Popescu et al. [17] applied it to obtain privacy-preserving EEG data classification. However, the computational inefficiency and high latency of their method constrain its application for real-time EEG data processing in BCI applications.

Differential privacy (DP) provides a framework for adding noise to data or queries to prevent the disclosure of individual-specific information [49]. Debie et al. [50] proposed a method with Generative Adversarial Networks (GAN) to produce synthetic EEG data resembling real EEG data. Their GAN framework developed using a differential privacy approach, implemented a privacy budget to strike a balance between data utility and privacy. However, their model achieved slightly lower but similar performance to those of trained on original EEG data. Although DP is effective in ensuring privacy, the EEG data-specific mechanisms [16, 17] face difficulties in preserving the utility of the data, which significantly affected the accuracy of the downstream machine learning models [18, 19].

Federated learning (FL) enables the training of the decentralized model on local devices, ensuring that the raw data remain on the user's device, thus enhancing privacy [51]. However, FL also faces challenges related to communication overhead, model convergence, and heterogeneity of data between devices. Xia et al. [14] highlighted these limitations in their review of privacy-preserving brain-computer interfaces, noting that while FL reduces the risk of data breaches, it struggles with scalability and efficiency in processing large-scale EEG datasets.

Wang et al. [52] proposed a novel *privacy-preserving domain adaptation approach* (PPDA) for intracranial EEG classification that addresses individual differences and privacy concerns without accessing source data. This method showed promise in reducing domain shifts and preserving patient privacy but relies on pseudolabeling techniques introducing inaccuracies, which accumulate and degrade the model's performance over time, particularly when dealing with complex and noisy EEG signals.

Agarwal et al. proposed cryptographic techniques to ensure the confidentiality of multiple users [53]. Their approach relies on *secure multiparty computation* (SMC) [54], which ensures that no party has access to an individual's brain/EEG signal. However, this method involves substantial effort to make it feasible on less powerful machines, which presents a major disadvantage. Hanisch et al. [55] also noted the complexity of implementing SMC in EEG-based BCI, which poses a significant barrier to its widespread adoption.

Focusing on AI-driven *cybersecurity solutions*, Schiliro et al. [56] introduced a novel cognitive privacy technique that can protect EEG data from cyber threats. To ensure the security of brain data within the BCI architecture against unauthorized credentials, a normalized correlation analysis approach was used, which was also referenced in [2]. Pazouki et al. [57] built a model to mitigate common cyber threats such as flooding or jamming associated with the control of smart home devices with brain implants. Their ANN-based model showed effective performance against false date injection attacks and scanning attacks [13].

Bidgoly et al. [58] suggested a privacy protection technique for BCI authentication systems where instead of using raw brain signals, confidential user data can be conserved as an EEG fingerprint, similar to a cryptographic hash. Gui et al. [59] introduced a detection mechanism based on residual noise characteristics to determine replay attacks and input modifications. Initially, it recognizes the user using a convolutional neural network (CNN)and then classifies the replay attacks. By mixing the EEG channels, abnormalities in the communication channel caused by hacking can be determined. Mezzina et al. [60]

found vulnerabilities between the BCI device and the framework, and addressed them using a versatile cyber-secure architecture resistant to noise-based attacks.

Maiorana et al. [61] discovered a hill climbing attack, which is achieved by iterative modification of EEG data until success with the BCI-based biometric system. To protect the features of the EEG against such attacks, Wang et al. [62] proposed a system based on polynomial transformation with cosine functions that modulate the features of the graph and devised a template-corrupting process to improve to provide cancelable templates for improved EEG security.

Although synthetic data generation and privacy-preserving tools have advanced significantly recently, some notable limitations remain. Numerous studies highlight the considerable potential of employing SNNs to process EEG data [63, 64], yet none have successfully provided a comprehensive approach to preserving all forms of private information. For example, PATE-GANs have been developed to address privacy concerns but face challenges with scalability and high-dimensional data [65]. Furthermore, despite progress in the generation of temporal data with GANs [66, 67], these methods lack the incorporation of differential privacy. Several studies examined the self-adaptiveness of SNNs and federated frameworks for tasks such as anomaly detection (AD) and human activity recognition [68, 69], but indicated the need for privacy-preserving mechanisms.

Various studies have been conducted on privacy-preserving techniques for SNNs, for example, in homomorphic encryption and energy efficiency; however, this area of research typically does not offer direct applications to EEG data or include advanced temporal coding [70, 71]. Similarly, continuous sequential data GANs, including Spiking GANs, have reported significant progress in data generation quality, but do not address privacy concerns [72, 73]. Recent studies on GANs, equipped with temporal dynamics and differential privacy mechanism, showcase effectiveness while highlighting the need for further integration of federated learning and advanced temporal coding [74, 75].

Yan et al. [63] empirically proved that spiking neural networks (SNN) are energy effi-

cient and well-suited for real-time processing systems compared to classical artificial neural networks (ANN) for EMG/EEG classification, where SNN accuracies were competitive with state-of-the-art ANNs while leading to far lower power consumption. Based on this, Xu et al. [64] used the temporal properties of SNN to capture the intricate temporal characteristics possessed by the EEG data and used them for emotion recognition. To address privacy-related issues, a GAN model, named PATE-GAN, was proposed in [65], which applies differential privacy in the generative domain using the framework of private aggregation of teacher ensembles. Although this model achieved strong privacy guarantees, it was challenging to scale and introduced additional complexity via an ensemble of teacher models, as well as the need for secure aggregation. Moreover, PATE-GAN had difficulty working with high-dimensional data because the student discriminator needs to see at least somewhat realistic generated samples from the beginning, which becomes difficult in higher-dimensional spaces. McKenna et al. with the AIM algorithm [76], noted the importance of a balance between data utility and privacy, contributing an adaptive iterative strategy for differentially private synthetic data generation.

Esteban et al. [66] introduced RCGAN – a recurrent conditional GAN to generate realistic time series medical data, which showed the ability of GANS to work with temporal sequences. This work paved the path for follow-up studies like Yoon et al.'s [67] exploration, which was the first one, an implementation of TimeGAN with embedding and recovery functions to treat mixed datasets containing temporal data along with non-time-aware features. Both studies were able to process temporal data but did not include privacy-preserving technologies, creating an opportunity for additional improvement to secure data generation. Bäßler et al. extended the use of SNNs through an unsupervised anomaly detection framework over multivariate time series while focusing on its adaptive properties for online data streams [68]. Complementing this, Khan et al. [69] proposed a federated framework for human activity recognition, combining privacy and energy efficiency to process data from distributed edge devices. Together, these studies demonstrated the promise of using SNNs with federated

learning to increase privacy and efficiency.

In [70] a homomorphic encryption scheme for privacy-preserving SNN was proposed to provide strong secrecy guarantees by encrypting the input while leveraging the benefits of SNN in processing temporal data. Arsalan et al. [71] focused on energy efficiency and privacy preservation in forecasting user health data streams using SNN to demonstrate the trade-offs between accuracy and energy savings. Rather, in the GAN world, Mogren [72] introduced C-RNN-GAN to adversarial training from continuous sequential data such as music generation. This paper showed that these GANs have the potential to process temporal data. In this sense, Rosenfeld et al. [73] extended the concept with Spiking GANs to incorporate local training and Bayesian models, as well as continuous meta-learning to enhance adaptability and efficiency. This research underscored that continued advancements in GANs and SNN led to more efficient generation of synthetic data examples.

Shen et al. introduced temporal spiking GANs [74] for heading direction decoding, a form of EEG signal processing, and showed the importance of temporal dynamics in this research stream. Similarly, Wang and Zhao [75] proposed DPSNN to address the differential privacy of temporal data processing by using temporal enhanced pooling to improve utility and robustness with respect to differentially private training. These experiments showcased the power of privacy-preserving mechanisms when coupled with GANs while dealing particularly with temporal data. Furthermore, the concept of PPGAN by Liu et al. [77] deliberately combines differential privacy with GANs by adding noise to the gradient during generator training. Later, Xie et al. [78] extended this work as DPGAN, introducing differential privacy in data generation through noise addition, demonstrating that both privacy and the usefulness of the generated data are crucial.

2.6 Quantum Computing's Emerging Role

Although significant progress has been made in developing privacy-preserving techniques for EEG data, existing methods face several challenges that limit their effectiveness and

practicality. The dynamic nature of EEG data, characterized by its high dimensionality and temporal dependencies, poses unique challenges to preserving privacy. Many existing techniques fail to adequately address these characteristics, leading to oversimplification of the data or inadequate protection measures, especially with small datasets [79].

Quantum algorithms provide a unique advantage for their potential to implement differential privacy in a more secure manner, employing quantum noise mechanisms that naturally align with the probabilistic and unpredictable nature of quantum mechanics [10, 80].

The intersection of quantum computing and privacy is an emerging field, with ongoing research aimed at addressing the unique challenges posed by integrating quantum principles into existing privacy-preserving frameworks. The application of quantum computing in BCI has been explored in hybrid models, combining classical and quantum computing elements to enhance performance and security. For example, quantum neural networks (QNN) are being developed to improve the accuracy and robustness of EEG data processing, utilizing quantum properties to handle complex patterns and noise more effectively than traditional neural networks [80, 81].

Quantum enhanced BCIs can potentially revolutionize the way we approach brain signal processing, enabling faster and more secure data handling, which is crucial for applications such as neuroprosthetics and brain-controlled interfaces [82, 83, 84]. As technology matures, quantum-enhanced privacy techniques are expected to become increasingly viable to protect sensitive data, particularly in high-security applications such as EEG-based BCIs [85, 86, 82].

Limited research on the application of differential privacy, GANs, and quantum principles specifically to EEG data underscores a gap in the literature that requires further exploration of methods that can effectively protect EEG data without compromising their integrity.

2.7 Summary

The chapter discussed concepts essential to the thesis, including Generative Adversarial Networks (GANs), Sequential Generative Adversarial Networks (SGANs), Wasserstein Genera-

tive Adversarial Networks (WGANs), Conditional Generative Adversarial Networks (CGANs), Differential Privacy (DP), R'enyi Differential Privacy (RDP), Local Differential Privacy (LDP), Gaussian Mechanism, Quantum Uncertainty Principle, Quantum Decoherence, Federated Learning (FL). We highlight related works on privacy-preserving learning, generative models in the EEG-based BCI field along with existing privacy-preserving methods. In Chapter 3, the development of Quantum-Inspired Differential Privacy-Based GAN is covered, with the motive to obtain high-quality private EEG data that previous approaches failed to produce with a tight privacy budget.

3

Quantum-Inspired Differential Privacy-based GAN

The proliferation of Brain-Computer Interfaces (BCIs) using EEG data introduces significant privacy risks, particularly due to the susceptibility of these data to inference and reconstruction attacks. Traditional privacy techniques such as data anonymization, encryption, and data perturbation often compromise the utility of the data or fail under sophisticated attack scenarios. Recognizing the limitations of existing approaches, which often result in trade-offs between data utility and privacy, we developed a quantum inspired, differential privacy based generative adversarial network (Q-DP-GAN). Although classical GANs can generate high-quality synthetic data, they usually lack in dynamically modifying the privacy parameters during training, which leaves them open to privacy violations over time. By emulating inherent randomness of quantum processes in the noise adjustment, our findings

demonstrate the Q-DP-GAN's ability to protect EEG data against both membership inference and reconstruction attacks compared to traditional privacy-preserving methods. It effectively generates synthetic EEG data that maintain high utility while ensuring confidentiality and security of the underlying training data during BCI classification tasks validated with seminal BCI datasets.

3.1 Methodology

3.1.1 Hybrid quantum inspired differential privacy Model

The Q-DP-GAN model uniquely combines differential privacy, enforced by stochastic gradient descent (DP-SGD), with dynamically adjusted quantum-inspired noise dynamics within a generative adversary network (GAN) designed for the synthesis of EEG data. This integration is designed to achieve robust privacy guarantees while maintaining the utility of the generated EEG data. Our methodology is outlined in Algorithm 3.1 and depicted in Figure 3.1.

Generator and Discriminator Architecture

The generator (G) and the discriminator (D) form the backbone of our model, in which the generator is tasked with creating synthetic EEG data and the discriminator is responsible for the validation of its authenticity.

Generator Architecture: The generator is designed to convert a latent noise vector into synthetic EEG data that resembles the real EEG recordings both in structure and dynamics. It comprises a series of transposed convolutional layers, each designed to progressively upscale the input vector into a full EEG data structure. The generator function $G(\mathbf{z}; \theta_G)$ maps the latent space vector $\mathbf{z}$ to the data space, where θ_G denotes the generator parameters. The output is activated by a tanh function to ensure the output matches the amplitude characteristics typical of EEG signals. The generator architecture is designed as shown in Table 3.1.

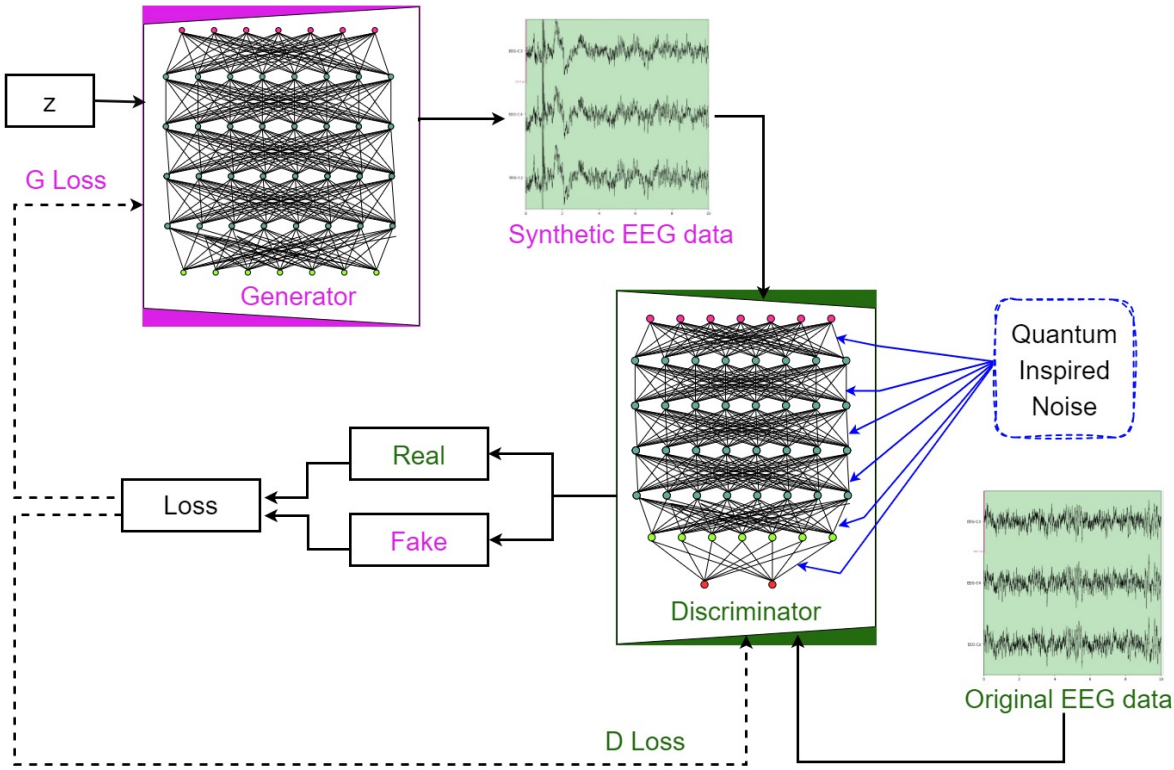


Figure 3.1: System architecture of the proposed Q-DP-GAN (quantum-inspired differential privacy) model.

Discriminator Architecture: The discriminator assesses the authenticity of EEG data, classifying it as either real or generated. It utilizes convolutional layers to extract features from the input data, which are then used to compute the probability of the data being real. The function $D(\mathbf{x}; \theta_D)$ evaluates the input $\mathbf{x}$, with θ_D representing the discriminator parameters. A sigmoid function at the output layer provides a probabilistic estimate of authenticity. The discriminator architecture is designed as shown in Table 3.2 and depicted in Figure 3.2.

Table 3.1: Generator architecture details of our proposed model

Block	Layer Name	Units/Size	Activation	Output Shape
1	Input (latent vector)	-	-	(100, 1)
2	Dense	128	LeakyReLU	(128, 1)
3	Reshape	-	-	(64, 2, 1)
4	GRU Layer 1	256	Tanh	(64, 256)
5	GRU Layer 2	64	Tanh	(64, 64)
6	Flatten	-	-	(4096,)
7	Dense	3000	ReLU	(3000,)
8	Reshape/Output Layer	-	Tanh	(3, 1000)

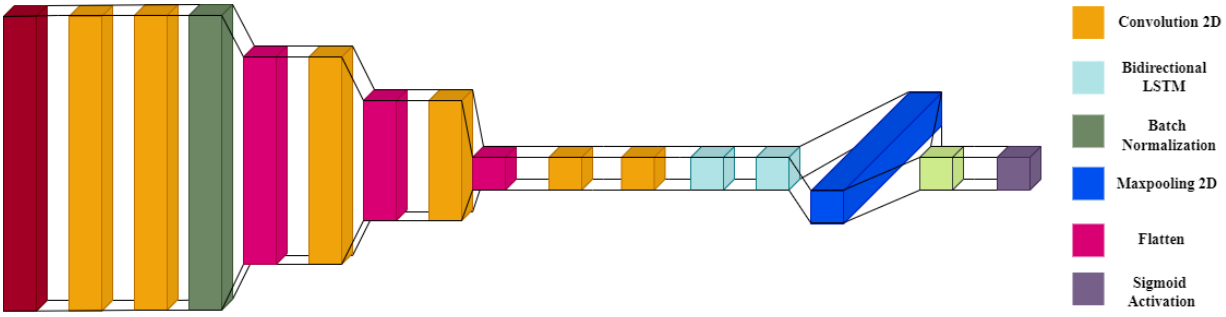


Figure 3.2: Discriminator architecture of the proposed model. For simplicity we omitted the activation and dropout layers.

Table 3.2: Discriminator architecture details of our proposed model

Block	**Layer Name**	**Unit**	**Kernel Size**	**Activation**	**Output Shape**
1	Input	-	-	-	(3, 1000)
2	Reshape	-	-	-	(1, 3, 1000)
3	2D Convolution	10	(1,10)	ReLU	(10, 3, 991)
4	2D Convolution	10	(3,1)	ReLU	(10, 1, 991)
5	Dropout [Dropout rate: 0.25]	-	-	-	(10, 1, 991)
6	Batch Normalization	-	-	-	(10, 1, 991)
7	Max Pooling 2D	-	(1,2)	-	(10, 1, 495)
8	2D Convolution	16	(1,10)	ReLU	(16, 1, 486)

Table 3.2: Discriminator architecture details of our proposed model (continued)

Block	**Layer Name**	**Unit**	**Kernel Size**	**Activ-ation**	**Output Shape**
9	Max Pooling 2D	-	(1,3)	-	(16, 1, 162)
10	2D Convolution	32	(1,10)	ReLU	(32, 1, 153)
11	Max Pooling 2D	-	(1,2)	-	(32, 1, 76)
12	2D Convolution	64	(1,10)	ReLU	(64, 1, 67)
13	Dropout [Dropout rate: 0.25]	-	-	-	(64, 1, 67)
14	Batch Normalization	-	-	-	(64, 1, 67)
15	Bidirect- ional LSTM	64	-	Tanh	(64, 67)
16	Bidirect- ional LSTM	128	-	Tanh	(128, 67)
17	Flatten	-	-	-	(8704,)
18	Dense	64	-	ReLU	(64,)
19	Dropout [Dropout rate: 0.5]	-	-	-	(64,)
20	Dense (Output Layer)	1	-	Sigmoid	(1,)

Quantum Decoherence and Noise Dynamics

Reflecting the behavior of quantum systems, where interaction with the environment leads to a loss of quantum coherence, the model initially applies variable high-intensity noise that adapts over time, analogous to the uncertainty in quantum measurements.The mathematical formulation of noise dynamics is;

$$\sigma(t) = \sigma_{\text{initial}} e^{-\lambda t} \tag{3.1}$$

$$C(t) = C_{\text{initial}} e^{-\beta t} \tag{3.2}$$

where $\sigma(t)$ represents the standard deviation of the noise applied to the gradients at epoch t. This noise is dynamically adjusted, decreasing as t increases from 1 to $\text{epochs}_{\text{total}}$. This decrease simulates the transition from a quantum state (characterized by high uncertainty) to a more classical state (characterized by reduced uncertainty). Similarly, $C(t)$ denotes the clipping threshold for the gradients at each epoch t. It is also adjusted dynamically, ensuring that the influence of any individual training sample remains bounded throughout the training, which helps to mitigate the risk of revealing individual contributions in the synthesized data. Here, λ and β are the decay rates. We performed several trials, changing λ and β to examine how different rates affected the model's accuracy and privacy, in order to find the optimal values. In this study, λ and β were set to 0.05 and 0.03, respectively, as these values showed the right balance between privacy and accuracy (Section 3.2.3).

Quantum Uncertainty Principle in Later Phases:

To ensure persistent privacy protection as the model stabilizes, the Quantum Uncertainty Principle maintains a baseline level of indeterminacy, preventing an exact determination of individual data contributions. The constant privacy parameters are defined as follows:

$$\sigma = \sigma_{\text{final}} \tag{3.3}$$

$$C = C_{\text{final}} \tag{3.4}$$

These parameters remain unchanged after the transition (described in the following section). By maintaining a steady level of noise and minimizing the influence of individual data points, they reliably ensure data privacy.

3.1.2 Transition Algorithm and Privacy Budget Computation

Transition Algorithm:

The model employs an algorithmic approach to determine when to switch from dynamic to static privacy parameters based on the stabilization of the model's performance.

- Algorithm Details:
  - if $\Delta\text{Loss} < \theta$ for n consecutive epochs, then transition to static phase.

    This condition checks if the change in loss ΔLoss falls below a threshold θ over n epochs, indicating that the model has reached a sufficiently stable state and requires reduced dynamism in privacy controls.

To ensure effective execution of these parameters, ΔLoss is calculated as follows:

$$\Delta\text{Loss} = \frac{1}{n} \sum_{i=t-n+1}^{t} |\text{Loss}_i - \text{Loss}_{i-1}| \tag{3.5}$$

This formula calculates the average absolute change in loss over n epochs, providing a smooth and representative measure of the changes in loss in this period. It effectively captures both the magnitude and consistency of loss changes, providing a reliable trigger for the parameter transition.

Privacy Budget Computation:

To incorporate quantum noise while ensuring consistency with the principles of differential privacy, the privacy budget [87, 88, 89] at a given epoch t is modified as follows:

$$\varepsilon(t) = \sqrt{2\log(1.25/\delta)} \cdot \sum_{i=1}^{t} \omega(i) \cdot \frac{q \cdot C(i)}{\sigma(i)}, \tag{3.6}$$

where, $w(i)$ is a weighting function defined as $w(i) = e^{-\gamma i}$, with γ representing the decay rate. This rate controls how rapidly the influence of each epoch reduces, highlighting the diminishing privacy risk as the model stabilizes. Here, $\gamma = 2\lambda\beta/(\lambda + \beta)$. $\sigma(i)$ and $C(i)$ are dynamically adjusted noise and clipping parameters, respectively, up to the stabilization point, after which they remain constant, and q represents the proportion of the dataset used in each training batch.

Algorithm 3.1 Training Q-DP-GAN with dynamic privacy

Input: EEG dataset, total number of epochs $\text{Epoch}_{\text{total}}$, initial and final privacy parameters $(\sigma_{\text{initial}}, C_{\text{initial}}, \sigma_{\text{final}}, C_{\text{final}})$, transition threshold (θ), number of stable epochs required (n), learning rates for discriminator (lr_D) and generator (lr_G)
Output: Trained GAN model with privacy guarantees
Initialize parameters: Generator parameters θ_G, Discriminator parameters θ_D
Initialize $\sigma_{\text{current}} = \sigma_{\text{initial}}$, $C_{\text{current}} = C_{\text{initial}}$
Initialize λ (decay rate for σ), β (decay rate for C)
Initialize *loss_stabilization_counter* $= 0$, *previous_loss* $= \infty$
for each epoch $t = 1$ to $\text{Epoch}_{\text{total}}$ **do**
    Shuffle and batch the dataset
    for each batch **do**
        Sample noise vector z from a normal distribution
        Generate synthetic data $G(z; \theta_G)$ using generator
        Train discriminator (D) on both real and generated data:
            Compute discriminator loss L_D on real data and $G(z; \theta_G)$
            Clip gradients of L_D by norm C_{current}
            Add Gaussian noise $\sim N(0, \sigma^2_{\text{current}})$ to the gradients
            Update θ_D using gradient descent with learning rate lr_D
        Train generator (G) to fool discriminator:
            Compute generator loss L_G using D's response to $G(z; \theta_G)$
            Update θ_G using gradient descent with learning rate lr_G
    end for
    Compute and record the discriminator loss for this epoch (*current_loss*)
    Assess model stability:
    if $|\text{previous\_loss} - \text{current\_loss}| < \theta$ **then**
        loss_stabilization_counter += 1
    else
        loss_stabilization_counter $= 0$
    end if
    if *loss_stabilization_counter* $\geq n$ **then**
        Transition to static privacy parameters:
        $\sigma_{\text{current}} = \sigma_{\text{final}}$
        $C_{\text{current}} = C_{\text{final}}$
    end if
    Update *previous_loss*:
    previous_loss = *current_loss*
    if not yet transitioned **then**
        Adjust privacy parameters dynamically:
        $\sigma_{\text{current}} * = e^{-\lambda}$
        $C_{\text{current}} * = e^{-\beta}$
    end if
end for
Return: trained GAN model (θ_G, θ_D)

3.2 Experimental Setup

3.2.1 Dataset Description

To improve the reliability, robustness, and generalizability of our model on various datasets, we selected two different datasets. Our study focuses on the detailed examination of the BCI Competition IV 2008 datasets, 2A [90] and 2B [91], which are seminal in the field of BCI. These datasets are not only validated through numerous research applications, but also provide a reliable benchmark for motor imagery tasks, facilitating insightful comparisons with other similar datasets. Collecting an original EEG-based BCI dataset was not feasible within the scope of this work due to several constraints. Building a new EEG dataset requires substantial resources, including time, funding, and specialized equipment such as EEG recording hardware, fMRI scanners, and MEG systems, which are expensive and difficult to acquire. In addition, there are major logistical concerns in arranging human subject studies for the duration of this thesis work.

Dataset 2A contains EEG recordings of nine participants who participated in four distinctive motor imagery tasks that included movements of the left hand, right hand, both feet, and tongue during multiple sessions. These sessions are meticulously structured, offering insight into neural patterns associated with motor tasks. Dataset 2B also contains EEG data from 9 subjects performing left- and right-hand movements, with and without feedback mechanisms, unlike dataset 2A. A detailed comparison of these two datasets is shown in Table 3.3.

Table 3.3: Comparison of BCI Competition IV Dataset 2A and 2B

Attribute	**Details for BCI-2A**	**Details for BCI-2B**
Dataset Origin	Graz University of Technology, 2008	Graz University of Technology, 2008

Attribute	Details for BCI-2A	Details for BCI-2B
Subjects	9 healthy subjects	9 right-handed subjects with normal or corrected vision
Motor Imagery Tasks	4 tasks: Left hand, right hand, both feet, tongue	2 tasks: Left hand, right hand
Number of Sessions	2 per subject	5 per subject
Runs per Session	6 runs	Screening: 6 runs, Feedback: 4 runs
Trials per Run	48 trials (12 per task)	Screening: 20 trials (10 per task), Feedback: 40 trials (20 per task)
Total Trials per Session	288 trials	Screening: 120 trials, Feedback: 160 trials
Total Trials per Subject	576 trials (2 sessions × 288 trials)	720 trials per subject (240 from screening, 480 from feedback)
EEG Electrodes	22 electrodes	3 bipolar electrodes (C3, Cz, C4)
EOG Channels	3 monopolar electrodes, for artifact processing	3 monopolar electrodes, for artifact processing
Sampling Rate	250 Hz	250 Hz

Attribute	Details for BCI-2A	Details for BCI-2B
Filtering	Bandpass 0.5-100 Hz, 50 Hz notch filter	Bandpass 0.5-100 Hz, 50 Hz notch filter
Feedback Provided	No	Yes, in last 3 sessions
Artifact Handling	Expert visual inspection, trials with artifacts marked	Trials containing artifacts marked, required artifact removal
Data File Format	GDF (General Data Format)	GDF (General Data Format)
File Distribution	Training and evaluation sets separate	First 3 sessions for training, last 2 sessions for evaluation
Data Access Software	BioSig toolbox (Octave/FreeMat/MATLAB, C/C++)	BioSig toolbox (Octave/MATLAB, C/C++)

3.2.2 Data Preprocessing

For Dataset 2A, our focus was on three channels, C3, Cz, and C4, which efficiently capture brain patterns associated with identifying imagined movement states [92]. Among the four initial motor imagery tasks, the movements of the feet and tongue were omitted, and the study focused solely on the movements of the left and right hands. To ensure data quality, we excluded trials labeled with event type 1023, which were marked as artifacts in the dataset by expert reviewers [90, 91]. Table 3.4 shows the accepted and rejected trials for Sessions I and II of all subjects. Session I was used as the train set and Session II as the test set.

For Dataset 2B, we used all three EEG channels (C3, Cz and C4) and tasks (left- and right-hand movements). Artifacts are also present in this dataset and we again excluded

Table 3.4: Accepted and rejected/artifact trials by subject and task of Dataset 2A

Dataset: 2A Session I (train set)										
Task	**Status**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**
Left Hand	Accepted	69	67	69	62	63	56	67	66	53
	Rejected	3	5	3	10	9	16	5	6	19
Right Hand	Accepted	69	69	68	67	66	57	66	66	63
	Rejected	3	3	4	5	6	15	6	6	9
Dataset: 2A Session II (test set)										
Left Hand	Accepted	71	71	67	59	70	59	71	66	65
	Rejected	1	1	5	13	2	13	1	6	7
Right Hand	Accepted	70	71	70	57	65	55	69	68	65
	Rejected	2	1	2	15	7	17	3	4	7

Table 3.5: Accepted and rejected/artifact trials by subject and task of Dataset 2B

Dataset: 2B Session I (train set)										
Task	**Status**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**
Left Hand	Accepted	51	48	47	57	51	35	55	52	43
	Rejected	9	12	13	3	9	25	5	8	17
Right Hand	Accepted	51	50	43	60	52	41	52	40	48
	Rejected	9	10	17	0	8	19	8	20	12
Dataset: 2B Session II (train set)										
Left Hand	Accepted	48	51	43	56	58	42	55	49	45
	Rejected	12	9	17	4	2	18	5	11	15
Right Hand	Accepted	50	49	45	53	58	43	56	41	46
	Rejected	10	11	15	7	2	17	4	19	14
Dataset: 2B Session III (train set)										
Left Hand	Accepted	60	64	56	78	74	65	68	54	69
	Rejected	20	16	24	2	6	15	12	26	11
Right Hand	Accepted	64	67	62	75	69	69	70	52	66
	Rejected	16	13	18	5	11	11	10	28	14
Dataset: 2B Session IV (test set)										
Left Hand	Accepted	51	51	62	78	79	71	51	69	56
	Rejected	29	29	18	2	1	9	29	11	24
Right Hand	Accepted	61	51	55	75	77	66	56	57	59
	Rejected	19	29	25	5	3	14	24	23	21
Dataset: 2B Session V (test set)										
Left Hand	Accepted	58	71	60	76	55	52	63	50	65
	Rejected	22	9	20	4	25	28	17	30	15
Right Hand	Accepted	58	72	53	78	62	62	62	54	65
	Rejected	22	8	27	2	18	18	18	26	15

trials labeled with event type 1023. Table 3.5 shows the accepted and rejected trials for the five sessions. Sessions I, II and III were used as train sets, and rest as test sets.

In the preprocessing stage of our study, initial analysis was performed using 7.5-second trials to capture a broad spectrum of neural activity. To improve the efficiency of model training and optimize the use of computational resources, we systematically tested the efficacy of shorter time windows. Our methodological approach involved segmenting the data into 6- and 4-second intervals. This segmentation was intended to identify the time frames that are most critical for motor imagery classification tasks. Specifically, the 6-second segments were obtained by excluding the first 1 second and the last 0.5 seconds of the trials, reducing the data from 1875 (7.5 X 250) data points to 1500 (6 X 250) data points. Further refinement led us to test 4-second segments, where the initial 2 seconds and the final 1.5 seconds were removed, leaving 1000 data points. These truncated segments were analyzed to determine if they contained sufficient information for accurate classification.

Our initial findings validated the 4-second window, as it provided classification accuracy comparable to longer durations. This indicates that the key neural signatures of the motor imagery tasks are within this interval. Using these segments, we achieved improved model training and real-time processing efficiency without compromising classification quality. Thus, the 4-second window was chosen for the subsequent analysis, balancing computational efficiency and empirical effectiveness.

3.2.3 Model Training: details, hyperparameters, and configurations

Selecting the appropriate parameter values is crucial to balance the effectiveness of the privacy mechanism and the model's learning capability. The following paragraphs explain the parameters selected for the hybrid quantum-inspired model that were empirically determined through extensive experimental trials.

Initial Phase (quantum decoherence)

- We chose the initial noise standard deviation (σ_{initial}) as 2.0. The chosen value provides robust privacy protection at the beginning of the training process, where the model is most susceptible to leaking information. A higher initial noise level effectively obscures the contributions of individual training samples, thereby reducing the risk of overfitting to specific data points, which could compromise privacy.

- We have set the initial clipping threshold (C_{initial}) to 1.0, a value determined through experimentation. Beginning with a higher clipping threshold also ensures that the impact of any single data point remains constrained during the early stages of training. This configuration helps to mitigate the risk of outliers exerting undue influence on the model's learning process, thereby enhancing privacy protection.

Decay Rates (λ for noise and β for clipping)

- Similarly, following our experimental testing, we set decay rates, λ to 0.05 and β to 0.03. The decay rate of 0.05 for noise ensures that the model gradually transitions to less noisy updates as it stabilizes, balancing privacy protection with learning efficiency. The slightly lower decay rate of 0.03 for clipping helps prevent the model from becoming overly sensitive to individual samples too quickly, thereby maintaining a conservative approach to privacy for a longer duration during training.

Stabilization Phase (quantum uncertainty)

- We have defined the final noise standard deviation (σ_{final}) as 0.5. It provides sufficient privacy protection as the model stabilizes without significantly impairing the model's fine-tuning capabilities. The reduced noise level prevents precise inferences about individual data points while allowing the GAN to generate high-quality synthetic EEG data.

- We have set the final clipping threshold (C_{final}) to 0.1. This lower threshold refines the model's sensitivity to the training data during stabilization. It improves the quality of generated outputs by limiting the influence of individual training samples without overly constraining model learning.

Privacy Guarantee (δ)

- We set δ at $1/\ N$, where N is the total number of data-points. For our 4-seconds data window, N is 1000 (4 x 250).

Privacy Budget (ϵ)

- For our method, the final ϵ values for Dataset 2A were 4.291, 4.365, 4.246, 4.381, 4.309, 4.226, 4.317, 4.403, and 4.192 for Subjects A1 through A9, respectively. For Dataset 2B, the final values for Subjects B1 through B9 were 4.457, 4.324, 4.355, 4.410, 4.493, 4.445, 4.397, 4.382, and 4.363, respectively.

Total epochs and number of consecutive epochs (n)

- We set the $\text{Epoch}_{\text{total}}$ as 1000.

- For a total of 1000 epochs, n should be large enough to ensure that the detection of model stabilization is robust against normal fluctuations in training loss but not so large that it delays the transition unduly. A reasonable choice for n is about 1% to 2% of $\text{Epoch}_{\text{total}}$, which translates to 10 to 20 epochs. This range typically offers a good balance by smoothing out regular variations in loss while being responsive enough to changes indicating true model stabilization. Setting $n = 20$ allowed the model to demonstrate consistent stabilization over a sufficient period to mitigate the effects of short-term fluctuations but is short enough to ensure timely adaptation to stabilized conditions.

Loss Stabilization Threshold (θ)

- Considering a typical GAN loss scale, values such as $\theta = 0.0001$ to 0.1 are effective in ensuring model stabilization. After initial experiments, we chose $\theta = 0.005$. This value guarantees that reductions in privacy parameters are triggered only by stabilization during training. Thus, aligning decrements with a steady phase of the model.

3.3 Results and Discussion

3.3.1 Classification Performance: Varying Original (real) and Synthetic Data

We first examined the effects of varying proportions of original (train data) and synthetic EEG data to train the model and evaluate its classification performance (utility) only on real EEG samples from the test data. Dataset 2A's original training data comprises only session I, while Dataset 2B's original training data include sessions I-III. We used four neural network models: ATCNet[93], EEGNet[94], ShallowNet [95] and CapsNet [96] to recognize hand movements using (1) original EEG data and (2) mixed data (original and synthetic EEG). The experiment, shown in Figure 3.3, aimed to assess how these data proportions impact the performance of the model.

We tested five specific ratios of original to synthetic data for training the model:

- 100% original data and 0% synthetic data. This model was trained exclusively on the original training dataset (Tr) for optimal comparison with augmented models.
- 50% original data and 50% synthetic data. This model used a balanced combination of 50% original training data and 50% synthetic data (S) to augment the training set while maintaining the total number of samples constant.
- 30% original data and 70% synthetic data.
- 20% original data and 80% synthetic data.
- 10% original data and 90% synthetic data.

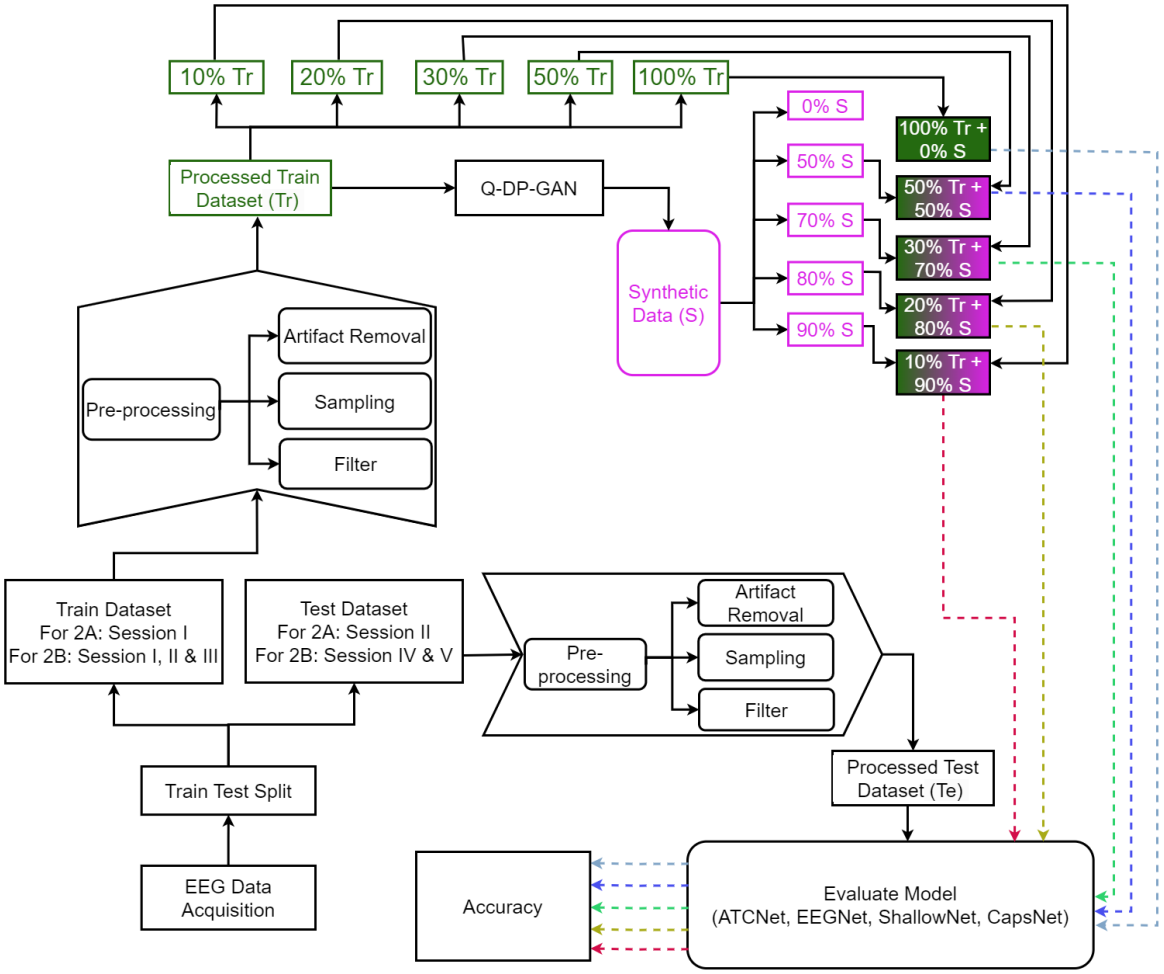


Figure 3.3: Overview of our experiments using original and synthetic data. The quantity of synthetic data (S) that we produced was equal to the number of original training samples (Tr). We then used varying proportions of Tr and S, merging them as training data for further analysis.

The classification performance of each model was evaluated over 20 runs to ensure the reliability and robustness of our results. For training, we used the training dataset specified in the original publication of the dataset resource (For Dataset 2A: Session I, and for Dataset 2B: Sessions I-III.) and tested it with the remaining sessions. Using this dataset, we applied our method to generate synthetic data and then trained via four different neural network models (ATCNet, EEGNet, ShallowNet, and CapsNet) with different combinations of original train and synthetic data , then tested on the original test dataset. Specifically, we

selected $k\%$ from original train data and $l\%$ from synthetic data, where $k = 100, 50, 30, 20,$ and 10, and $l = 0, 50, 70, 80,$ and 90. Given the variations in sample sizes across different subjects and datasets, we implemented a rounding procedure to determine the exact number of data points to use. The percentages were rounded to the nearest whole number using a threshold of 0.50.

The results of this experiment are illustrated in Figures 3.4 and 3.5. Model accuracy changes were more noticeable when the percentage of synthetic data is increased progressively, as discussed below:

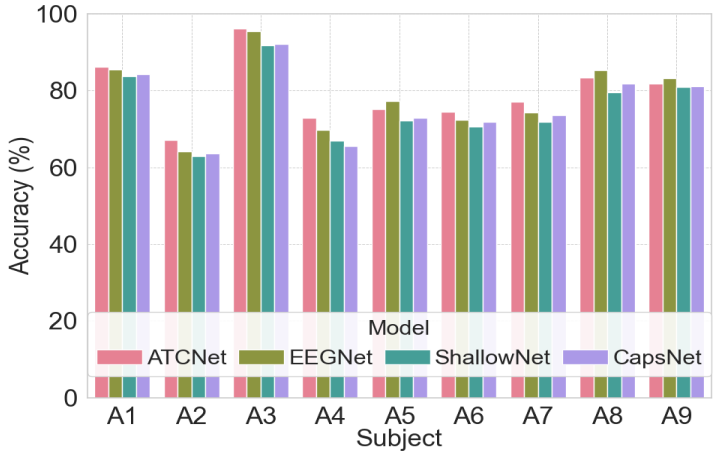
- Models trained on 100% original data with performed similarly to those trained on 50% original data and 50% synthetic data, with the accuracy dropping by only 1-2.5% on the test data.

- The model accuracy was reduced between 2.1-4.1% using 30% of original data mixed with 70% synthetic data.

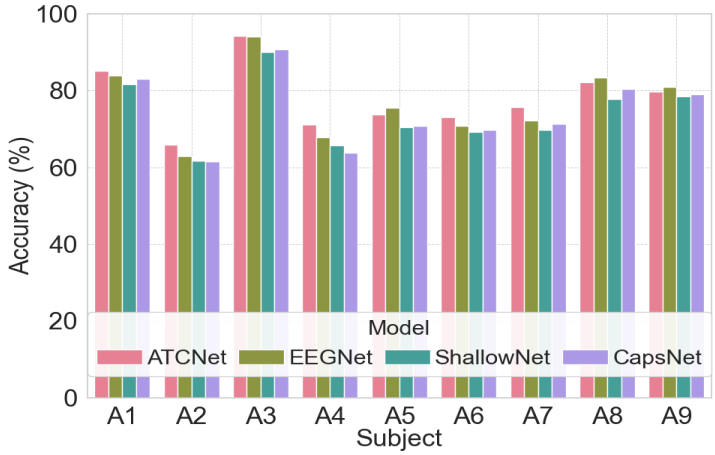
- A ratio of 20% of original data mixed with 80% synthetic data resulted in 3.2-5.6% accuracy drops.

- The greatest change was observed using 10% of the original data mixed with 90% synthetic data resulting in 5.1-7.2% change in model accuracy.

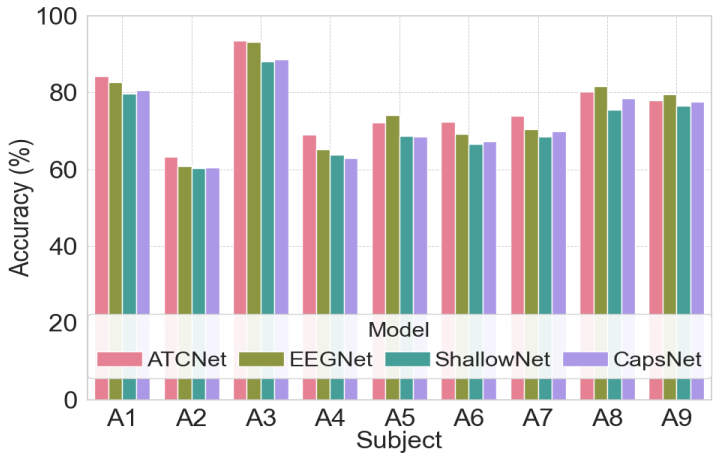
These findings suggest that while a balanced mix of 50% original and 50% synthetic data can effectively augment EEG datasets while maintaining robust model performance, other configurations such as 30% or 20% original data with synthetic data are also viable, although with slightly higher accuracy drops. The correct balance between the original and synthetic data depends on the error-rate tolerance of the BCI applications. Our findings show that varying proportions of synthetic EEG data is helpful to achieve comparable results in training processes that use real EEG data.



(a) Model trained on 100% original training data (Tr)

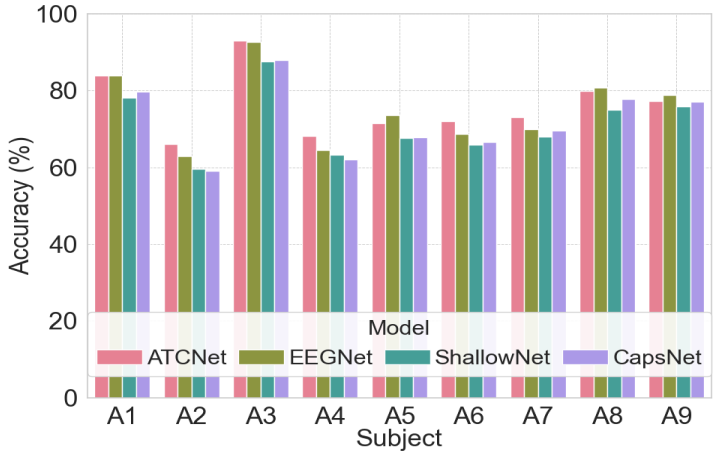


(b) Model trained on 50% original training data (Tr) and 50% synthetic data (S).

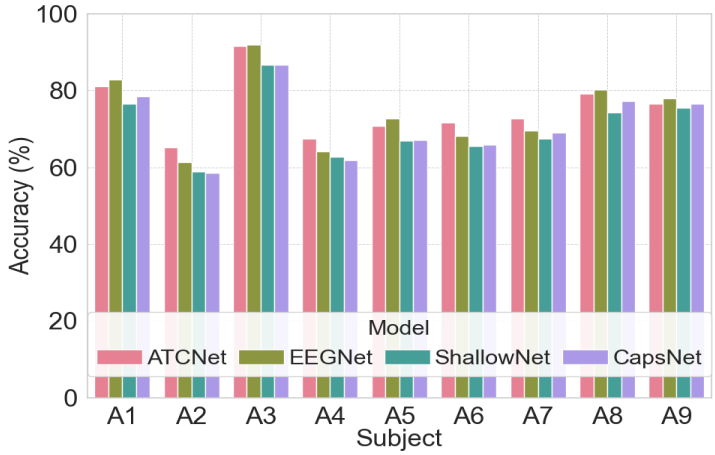


(c) Model trained on 30% original training data (Tr) and 70% synthetic data (S).

Figure 3.4: Classification performance of BCI Dataset IV 2A using various training scenarios. All models are evaluated on test data using 100% of real EEG data samples composed of session II. (Part 1)



(d) Model trained on 20% original training data (Tr) and 80% synthetic data (S).



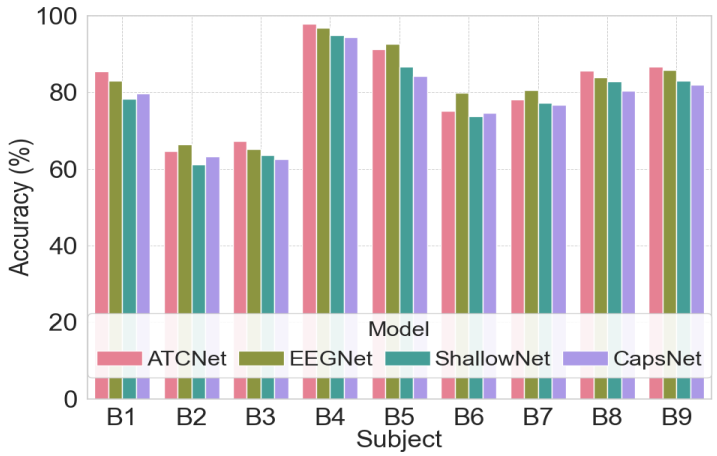
(e) Model trained on 10% original training data (Tr) and 90% synthetic data (S).

Figure 3.4: Classification performance of BCI Dataset IV 2A using various training scenarios. All models are evaluated on test data using 100% of real EEG data samples composed of session II. (Part 2)

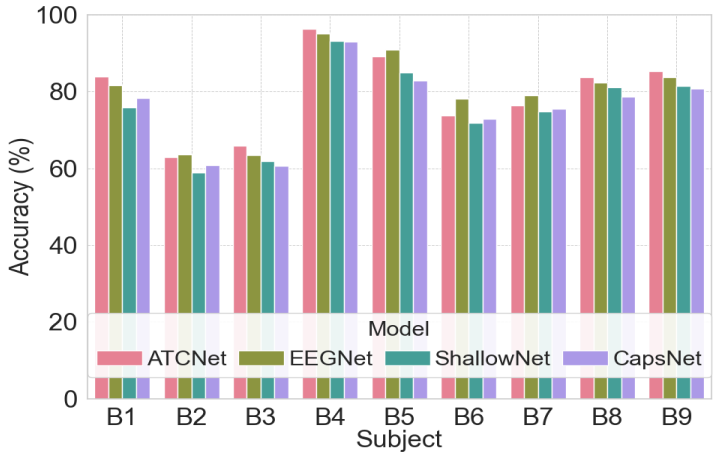
3.3.2 Privacy Analysis: attack scenarios

After evaluating the classification performance, we conducted simulations focusing on two specific attack scenarios to explore the preservation of privacy. To assess the effectiveness of these methodologies, we evaluated and compared four state-of-the-art methods - WGAN [24], CGAN [97], DPGAN [78], and LDPGAN [98].

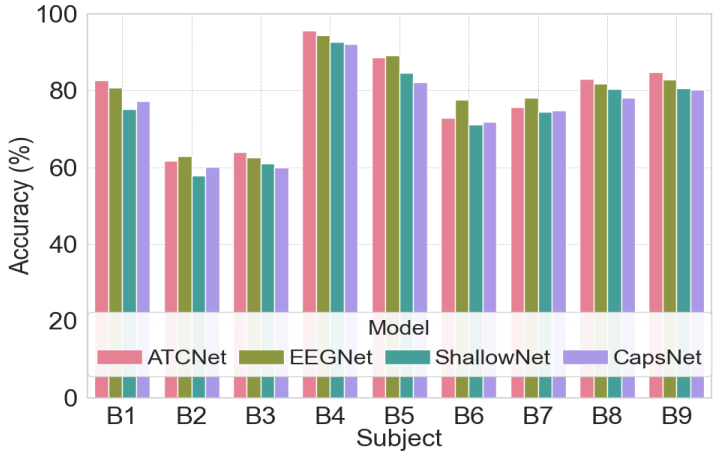
We ensured that every model was set using the ϵ values derived from our method, so that each model maintained the same privacy budget for each subject, allowing a fair comparison



(a) Model trained on 100% original training data (Tr)

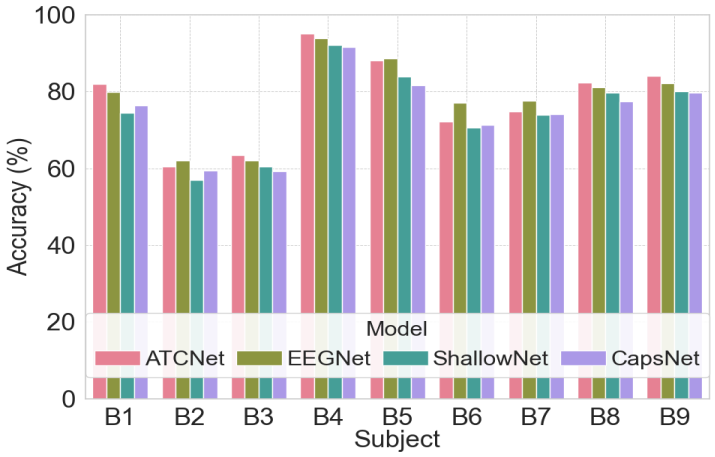


(b) Model trained on 50% original training data (Tr) and 50% synthetic data (S).

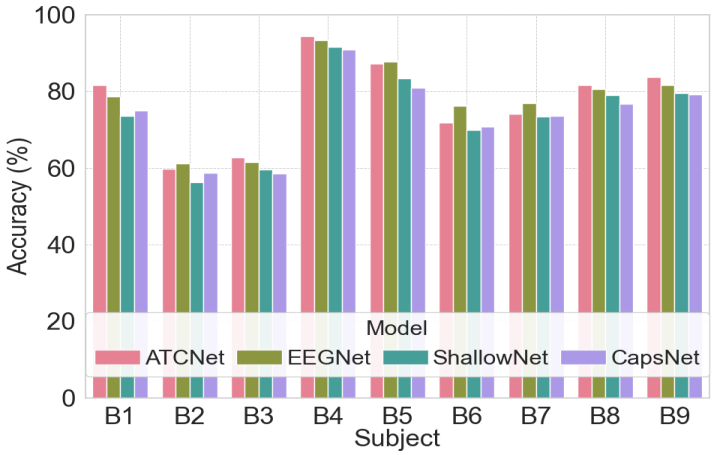


(c) Model trained on 30% original training data (Tr) and 70% synthetic data (S).

Figure 3.5: Classification performance of BCI Dataset IV 2B using various training scenarios. All models are evaluated on test data using 100% of real EEG data samples composed of sessions IV and V. (Part 1)



(d) Model trained on 20% original training data (Tr) and 80% synthetic data (S).



(e) Model trained on 10% original training data (Tr) and 90% synthetic data (S).

Figure 3.5: Classification performance of BCI Dataset IV 2B using various training scenarios. All models are evaluated on test data using 100% of real EEG data samples composed of sessions IV and V. (Part 2)

across the datasets. For each method, synthetic data was generated in quantities equivalent to the original training dataset and then exposed to the two attack scenarios.

Membership Inference Attacks

To assess the privacy preservation capabilities of our Q-DP-GAN, we implemented membership inference attacks (MIA) under black box (BB) and white box (WB) conditions [99, 79]. These attacks are designed to assess if an attacker can identify whether specific data were part of the model's training set. Three subsets of data were used to gauge the model's ability to secure EEG data against potential inference threats.

- S - Synthetic data generated by different models.
- Tr - Real training data used to train different models (Dataset 2A session I and Dataset 2B sessions I-III).
- Te - Real data not used in training, serving as test data (Dataset 2A session II and Dataset 2B sessions IV-V)).

To improve training, we combine 100% of the original training data (Tr) with a corresponding amount of synthetic data (S), essentially doubling the amount of the dataset to train the model. The effectiveness of the model is assessed using the unaltered testing dataset (Te). We are able to precisely evaluate the model's capacity to generalize from the training set to new untested data.

MIAs are conducted in two main modes based on the attacker's access level:

- *Black-Box Membership Inference Attacks:* The attacker does not have access to Q-DP-GAN 's internals. Black-Box (BB) scenarios simulate an external adversary using shadow models trained on various subsets of S and Tr, mimicking the target model's behavior without accessing its actual parameters. The BB attack is detailed in Algorithm 3.2.

Algorithm 3.2 Black-Box Membership Inference Attack

Require: Real training data T_r, Synthetic data S, Real test data T_e, Number of shadow models N_s

Ensure: Attack model to predict membership inference

Combine T_r and S to create T_{combined}
Split T_{combined} into N_s subsets: $\{T_{\text{combined}_1}, T_{\text{combined}_2}, \ldots, T_{\text{combined}_{N_s}}\}$
for each subset T_{combined_i} ($i = 1$ to N_s) **do**
    Train shadow model S_i on T_{combined_i}
end for
Generate Predictions:
for each shadow model S_i ($i = 1$ to N_s) **do**
    Predict probabilities on T_e: $P_i^{\text{test}} = S_i.\text{predict}(T_e)$
    Predict probabilities on S: $P_i^{\text{syn}} = S_i.\text{predict}(S)$
end for
Create and Label Attack Dataset:
Combine all P_i^{test} to form X_{test} { X_{test} contains the predictions for the real test data}
Combine all P_i^{syn} to form X_{syn} { X_{syn} contains the predictions for the synthetic data}
Label X_{test} with 1s (real) and X_{syn} with 0s (synthetic)
Form attack dataset $X_{\text{attack}} = \{X_{\text{test}}, X_{\text{syn}}\}$ and labels $Y_{\text{attack}} = \{1s, 0s\}$
Train Attack Model:
Split X_{attack} into training and testing sets
Train binary classifier *AttackModel* on the training set
Evaluate Attack Model:
Test *AttackModel* on the testing set
Output accuracy of *AttackModel*

- *White-Box Membership Inference Attacks:* In this scenario, the adversary uses complete information about Tr and the structure of Q-DP-GAN to examine the influence of training data on the model's outputs for Te and S. It can reveal vulnerabilities in the model's processing and learning from Tr. The WB attack is outlined in Algorithm 3.3.

The results for both types of MIA are imperative for evaluating the ability of Q-DP-GAN to obfuscate the origin of data points, particularly from Tr and S, as well as its efficacy in preventing adversaries from deducing the membership status of Te. The MIA results, depicted in Figures 3.6, 3.7, 3.8, and 3.9, illustrate that Q-DP-GAN offers robust protection against both external and internal threats, thereby ensuring the confidentiality of EEG data. The low predictive accuracy observed in the BB and WB scenarios indicates substantial data protection, which highlights the effectiveness of the model in protecting privacy.

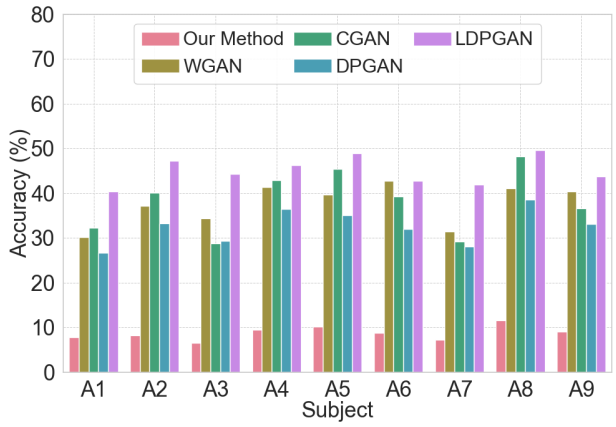


Figure 3.6: Black Box attack simulated on Dataset 2A. The accuracy indicates whether or not the model is able to identify the test data (Te). Reduced accuracy scores signify a more robust model against potential attacks. By adding an equivalent quantity of synthetic data (S) to 100% original data (Tr), the total amount of original training data is doubled.

Algorithm 3.3 White-Box Membership Inference Attack

Require: Real training data T_r, Synthetic data S, Real test data T_e, Labels y_r (for T_r) and y_e (for T_e), Trained discriminator model discriminator, List of intermediate layer names intermediate_layer_names

Ensure: Attack model to predict membership inference

Load the trained discriminator model discriminator
List the layers of discriminator to choose appropriate intermediate layers
for each layer in discriminator **do**
    Print the layer index, name, and output shape
end for
Choose intermediate layers for feature extraction based on the output from step 2
Split the real training data into training and testing sets:
Split T_r and y_r into $T_{r_{\text{train}}}$, $T_{r_{\text{test}}}$, $y_{r_{\text{train}}}$, $y_{r_{\text{test}}}$ with a test size of 20%
Prepare attack data:
Extract intermediate outputs from the chosen layers of discriminator for $T_{r_{\text{train}}}$, S, and T_e
Build an intermediate model with the chosen layers
Predict the features using the intermediate model on $T_{r_{\text{train}}}$, S, and T_e
Concatenate features from multiple intermediate layers
Create and Label Attack Dataset:
Combine features from T_e and label as 1 (real test data)
Combine features from S and label as 0 (synthetic data)
Form attack dataset $X_{\text{attack}} = \{X_t, X_s\}$ and labels $Y_{\text{attack}} = \{1s, 0s\}$
Build the white-box attack model:
Define the input shape based on the shape of X_{attack}
Build the attack model with several dense, batch normalization, and dropout layers
Compile the attack model with Adam optimizer, binary cross-entropy loss, and accuracy metric
Train the white-box attack model:
Split X_{attack} and Y_{attack} into attack training and testing sets
Train the attack model on the attack training set with validation on the attack testing set
Evaluate the attack model:
Evaluate the attack model on the attack testing set
Output the accuracy of the attack model

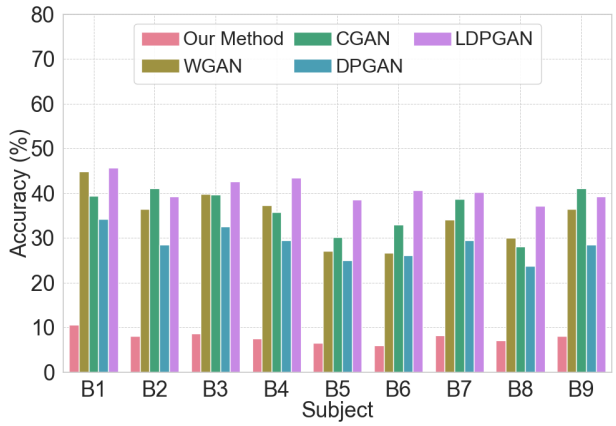


Figure 3.7: Black Box attack simulated on Dataset 2B. Our model's reduced accuracy scores signify that it is robust against potential attacks compared to other GAN models.

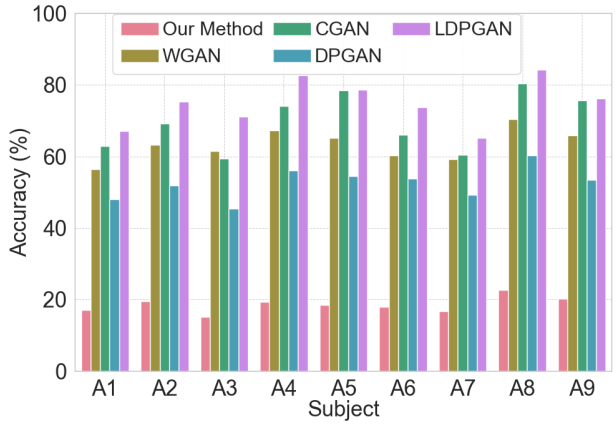


Figure 3.8: White Box attack simulated on Dataset 2A. The accuracy indicates whether or not the model is able to identify the test data (Te). Reduced accuracy scores signify a more robust model against potential attacks. We are using 100% synthetic data from each model. By adding an equivalent quantity of synthetic data (S) to 100% original data (Tr), the training dataset is doubled.

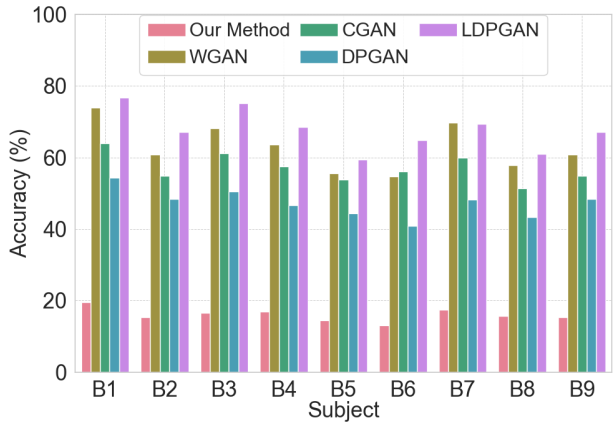


Figure 3.9: White Box attack simulated on Dataset 2B. Our model demonstrated similar performance as the Black Box attack results (lower accuracy scores), indicating its robustness to potential attacks in comparison to other GAN models.

Reconstruction Attacks Analysis

The second phase of our privacy assessment involved conducting reconstruction attacks [100] to test the resilience of Q-DP-GAN against attempts to recreate the original EEG data from synthetic output. These attacks present a significant challenge, as they aim to reverse the generative mechanism of Q-DP-GAN to retrieve or infer the original training data. The methodology for these attacks incorporates sophisticated models designed to approximate the inverse function of Q-DP-GAN's generator, scrutinizing the synthetic data's fidelity to its original counterpart and revealing potential weaknesses in the anonymization process.

We used advanced deep learning models to emulate the reverse generation process. It involved optimizing a latent space representation to match the synthetic data output back to its presumed original form. The attack model architecture was meticulously designed with dense layers for feature extraction, dropout layers to prevent overfitting, batch normalization for consistent training performance, and regularization to ensure generalization beyond the training data. This comprehensive setup was pivotal in evaluating Q-DP-GAN's capability to produce synthetic EEG data that not only retain utility but also resist direct reconstruction attempts for data privacy.

For reconstruction attacks, a higher mean squared error value is desirable. It signifies

challenges and complexity in accurately reconstructing original data from synthetic data, which provides increased privacy [100]. Figures 3.10 and 3.11 show the superior performance of Q-DP-GAN against this attack compared to other state-of-the-art methods.

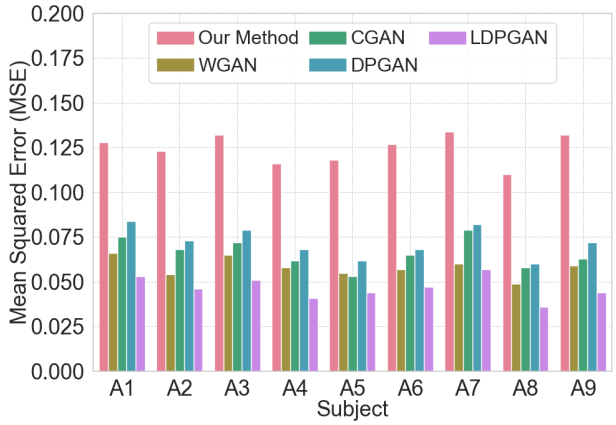


Figure 3.10: Reconstruction attack simulated on Dataset 2A to assess the model's ability to reconstruct original input data x from its outputs $M(x)$. Synthetic data (S), equivalent in quantity to the original dataset (Tr) is used for this assessment.

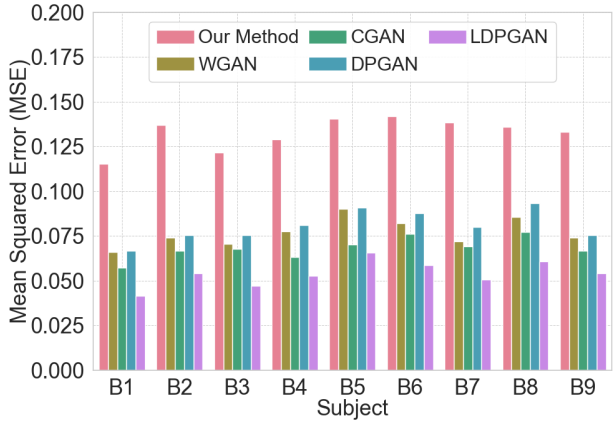


Figure 3.11: Reconstruction attack simulated on Dataset 2B. A higher mean squared error value achieved by our model, similar to Dataset 2A, shows resilience against reconstructing original data from synthetic data.

3.4 Summary

We designed a new generative adversarial network (GAN) model using quantum uncertainty and decoherence principles to ensure the privacy of EEG data in BCI applications. The un-

predictability of quantum uncertainty introduces randomness that is hard to reverse-engineer, enhancing security. Quantum decoherence allows dynamic noise adaptation during training, progressively aligning privacy measures with the model's learning and stability. This integration enhances the privacy of EEG data without sacrificing utility, addressing trade-offs between privacy-utility. Our quantum inspired differential privacy model (Q-DP-GAN) showed resilience to membership inference and reconstruction attacks, effectively obscuring data origins and preventing accurate reconstruction. The proposed model demonstrated that generated synthetic EEG data maintain high utility for BCI applications while hiding EEG data sources. Thus, the success of our model in balancing utility, privacy, and adversarial threats makes it a valuable resource for BCI researchers and practitioners.

Chapter 4 explores privacy techniques with the help of the Federated Learning approach. The Federated Privacy in Spiking GANs manages the pitfalls of the single-layer privacy framework by distributing the data processing at multiple nodes, and limiting access to raw data by the central server, which in turn reduces the risk of intrusion. This transition addressed the need for a federated structure capable of enhancing security without affecting the quality of EEG data. Using Spiking GANs with Federated Learning, we investigate the privacy preservation approach and the effects on the utility of the data.

4

Federated Privacy using Spiking GANs

This chapter provides a unique approach that uses temporal noise correlation and a Federated Generative Adversarial Network (GAN) architecture with Spiking Neural Networks (SNNs) to generate synthetic EEG data that preserve anonymity. Our method guarantees data privacy and produces high-quality synthetic data utilizing the federated learning paradigm, the intrinsic temporal dynamics of SNNs, and an Artificial Neural Network (ANN)-based discriminator for effective classification. Unlike conventional GANs, which employ CNNs in both the generator and the discriminator, our method employs SNNs in both scenarios, enabling efficient analysis of temporal data, such as EEG signals. We implement Renyi Differential Privacy (RDP) by introducing controlled noise into spike trains and membrane potentials to provide robust privacy guarantees. Quantitative evaluations through extensive

testing show that the proposed method can effectively synthesize realistic EEG data and therefore preserves both data privacy and utility. This research adds to the disciplines of neural network-based synthetic data creation and privacy-preserving machine learning, providing a potential option for applications that need privacy in addition to data fidelity.

4.1 Methodology

Our proposed methodology introduces an advanced GAN framework designed to generate privacy-preserving synthetic EEG data. This framework uniquely integrates a Spiking Neural Network (SNN)-based generator with an Artificial Neural Network (ANN)-based discriminator. Our model excels in producing high-fidelity synthetic EEG signals with the temporal dynamics of SNNs and the computational efficiency of ANNs. The spiking neurons within the generator simulate the behavior of biological neurons, while the discriminator's multi-layered ANN architecture is optimized to assess the authenticity of the generated data.

The architecture of the GAN involves several layers in generator and discriminator, each designed to play a critical role in processing and capturing the complex patterns of EEG signals. The SNN-based generator is built on a modified Leaky Integrate and Fire (LIF) neuron model, which is ideal for capturing the temporal dynamics of EEG signals. The ANN-based discriminator complements this by efficiently distinguishing between real and synthetic EEG data through a series of fully connected layers. Our methodology is detailed in Algorithm 4.1.

We developed a federated learning framework that allows several clients to use their own private EEG datasets to independently train Spiking GAN models. This methodology guarantees the decentralization of sensitive EEG data, safeguards privacy, and facilitates the improvement of collaborative models. At the beginning of every communication round, the server sends the global model parameters to each client. Then, using spike-timing-dependent learning algorithms and temporally correlated noise mechanisms to mimic the spiking behavior of neurons, the client processes the EEG data by training a local Spiking

GAN model. Sensitive information is kept localized because, upon completion of the local training, the client sends the updated model's parameters (gradients) back to the server without releasing any raw EEG data.

The fundamental component of this framework, the server, is in charge of combining the local updates from every client. Without gaining access to any client's raw data, the server uses Federated Averaging to merge the locally updated model parameters into a single global model. Over several rounds, this method iteratively improves the global model, with each communication cycle improving model performance. The server incorporates RDP privacy-preserving technique and temporally correlated noise into the updates to secure EEG data without compromising data security, ensuring that sensitive data is secured while still achieving effective model training.

4.1.1 Data Generation Process

The data generation process in our framework is composed of several stages, each crucial to ensure that the synthetic EEG data produced is realistic and preserve privacy. These stages begin with the theoretical foundation and model design, which are deeply rooted in the principles of spiking neural networks. In contrast to traditional neural networks that rely on continuous activation functions, spiking neurons operate based on discrete events—spikes. This discrete nature allows SNNs to capture the temporal dynamics inherent in EEG signals more effectively.

4.1.2 Membrane Potential Dynamics of the Neurons

The neurons in our generator are modeled using a modified version of the Leaky Integrate-and-Fire (LIF) model [101, 102]. The membrane potential u_i of the neuron i at any time $\bar{t}$ is described by Equation 4.1:

$$\tau_m \frac{du_i}{d\bar{t}} = -(u_i - u_{\text{rest}}) + RI_i(\bar{t}) \tag{4.1}$$

where, τ_m is the membrane time constant, u_{rest} is the resting potential, R is the input resistance, and $I_i(\bar{t})$ is the input current.

When the membrane potential u_i reaches a threshold ϑ, the neuron fires a spike, and the potential is reset to u_{rest}. The spike generation and reset mechanism are incorporated through the spike train $S_i(\bar{t})$, leading to the following equation [102]:

$$\tau_m \frac{du_i}{d\bar{t}} = -(u_i - u_{\text{rest}}) + RI_i(\bar{t}) + S_i(\bar{t})(u_{\text{rest}} - \vartheta) \tag{4.2}$$

where, the spike train $S_i(\bar{t})$ is defined as:

$$S_i(\bar{t}) = \sum_k \delta(\bar{t} - \bar{t}_k^i) \tag{4.3}$$

where $\bar{t}_i^k$ represents the k-th spike time of neuron i.

4.1.3 Discretization of Membrane Potential for Implementation

To implement the neuron dynamics in discrete time, we approximate the continuous-time differential equation using small time steps $\Delta\bar{t}$. The discretized update rule for the membrane potential at time step q is given by [102]:

$$u_i[q+1] = \tilde{\beta} u_i[q] + I_i[q] - S_i[q] \tag{4.4}$$

where $\tilde{\beta} \equiv \exp\left(-\frac{\Delta\bar{t}}{\tau_m}\right)$, $S_i[q]$ represents the spike at the q-th time step, and $S_i[q] = 1$ (if spike), $S_i[q] = 0$ (if no spike).

The input current $I_i[q]$ is updated similarly:

$$I_i[q+1] = \tilde{\alpha} I_i[q] + \sum_j W_{ij} S_j[q] \tag{4.5}$$

where $\tilde{\alpha} \equiv \exp\left(-\frac{\Delta\bar{t}}{\tau_s}\right)$, and W_{ij} is the synaptic weight between neuron j and neuron i.

4.1.4 Spike Generation and Membrane Reset

When the membrane potential u_i reaches the threshold ϑ, the neuron generates a spike, modeled by the Dirac delta function $\delta(\bar{t} - \bar{t}_k^i)$, and the potential is reset to u_{rest}. The spike reset process is represented by:

$$u_i[q+1] = \tilde{\beta} u_i[q] + I_i[q] - S_i[q] \tag{4.6}$$

where $S_i[q]$ indicates whether the neuron spikes at time step q. This formulation allows the neuron to fire multiple spikes in response to sustained input. $S_i[q] = 1$ forces the reset after the spike.

4.1.5 Synaptic Filtering for Spike Trains

To convert discrete spike trains into biologically plausible continuous signals, a double exponential synaptic filter [103, 102] is applied. The filtered response r_j of neuron j is governed by:

$$\dot{r_j} = -\frac{r_j}{\tau_d} + h_j \tag{4.7}$$

$$\dot{h_j} = -\frac{h_j}{\tau_r} + \frac{1}{\tau_r \tau_d} \sum_{\bar{t}_j < \bar{t}} \delta(\bar{t} - \bar{t}_j^k) \tag{4.8}$$

where, τ_r and τ_d are the synaptic rise and decay time constants, h_j is an intermediary signal, $\delta(\bar{t} - \bar{t}_j^k)$ is the spike train of neuron j.

This synaptic filtering process produces continuous signals from discrete spikes, mimicking the biological processes of real neurons.

4.1.6 Privacy Preservation Using Temporally Correlated Noise

To maintain the privacy of the generated EEG data, temporally correlated noise is introduced into the dynamics of the membrane potential during the training process. The noise $\text{Noise}_n(\mu)$ for neuron n is modeled as Equation 4.9:

$$\text{Noise}_n(\bar{t}) = \zeta \cdot \text{Noise}_n(\bar{t} - \Delta\bar{t}) + \xi(\bar{t}) \tag{4.9}$$

where, ζ is the correlation coefficient determining the degree of temporal correlation in the noise, $\xi(\bar{t}) \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise with mean 0 and variance σ^2.

This noise is then incorporated into the membrane potential equation as follows:

$$u_n(\bar{t} + \Delta\bar{t}) = \tilde{\beta} u_n(\bar{t}) + I_n(\bar{t}) - S_n(\bar{t}) + \text{Noise}_n(\bar{t}) \tag{4.10}$$

This noise is crucial to ensure that the generated data remain private, as it obscures the specific details of the input data while preserving the overall structure and temporal dynamics of the EEG signals.

4.1.7 Federated Learning and Model Aggregation

In this work, we implemented a federated learning framework by simulating both client-side and server-side processes on a single personal laptop. Although federated learning typically involves separate devices, we replicated the federated environment entirely on one machine for practical purposes. The dataset was partitioned between five simulated clients, each receiving an equal portion of the data. Each client initialized local instances of the SNN-based generator and ANN-based discriminator models, synchronized with the global model at the start of each training round. Clients trained their local models for five local epochs, processing data in mini-batches of size 64. After training, each client computed its contribution to the global model by evaluating the performance of its local discriminator, quantifying its contribution through a function that measures the difference between real and generated spike train outputs.

The server-side aggregation was simulated on the same machine. After receiving updates from all clients, we used contribution-weighted aggregation to update the global model, ensuring that clients with higher contributions had a greater impact. This process was repeated over 300 global epochs. We also dynamically adjusted each client's privacy budget based on

their contribution, ensuring a fair balance between data privacy and model contribution. To maintain privacy, we incorporated Rényi Differential Privacy (RDP), calculating the RDP epsilon value at the end of each epoch (described in the next subsection). By setting appropriate parameters, we ensured strong privacy guarantees while simulating a federated learning environment on a single laptop.

In our federated learning setup [104], the EEG dataset $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m\}$ was partitioned into multiple subsets among clients m, ensuring that each client only has access to its own local data. This decentralized structure improves privacy by keeping the raw EEG data localized, and only model updates are communicated to the central server. In the beginning, the global generator $\mathcal{G}$ and discriminator $\mathcal{D}$ were initialized on the central server, with initial weights $\Theta_0^{\mathcal{G}}$ and $\Theta_0^{\mathcal{D}}$. The noise parameters ζ and σ^2 were applied during the generation of synthetic EEG data to obscure sensitive information while preserving the temporal structure of the signals.

Each client trains its own local copy of the generator $\mathcal{G}_m$ and discriminator $\mathcal{D}_m$ using its local subset $\mathcal{X}_t^{(m)}$ of the EEG dataset. During local training, the generator produces synthetic EEG data by sampling latent vectors $\mathcal{Z}$ from a predefined distribution $\mathcal{P}(\mathcal{Z})$. At the same time, the local discriminator $\mathcal{D}_m$ was trained to differentiate between real and synthetic EEG data. Real $\mathcal{L}_{\text{real}}$ and fake EEG data losses $\mathcal{L}_{\text{fake}}$ were computed using binary cross-entropy loss functions. The discriminator was updated using gradient descent to minimize these loss functions, allowing it to improve its ability to classify real and synthetic data. The real and fake losses are given by [105, 106, 107]:

$$\mathcal{L}_{\text{real}} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\mathcal{X}_{bt}^{(m)}), 1) \tag{4.11}$$

$$\mathcal{L}_{\text{fake}} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\tilde{\mathcal{X}}_t^{(m)}), 0) \tag{4.12}$$

After training the discriminator, the generator was updated based on the feedback from

the discriminator. The generator's goal is to improve the realism of its generated EEG data so that the discriminator cannot easily distinguish between real and synthetic data. The generator loss was calculated similarly using binary cross-entropy loss, defined as

$$\mathcal{L}_{bt}^{\mathcal{G}_m} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\tilde{\mathcal{X}}_t^{(m)}, \mathcal{L}_t^{(m)}), 1) \tag{4.13}$$

Once local training was completed, the updated model weights $\Theta_{bt}^{\mathcal{G}_m}$ and $\Theta_{bt}^{\mathcal{D}_m}$ were sent to the central server, where they were aggregated. The central server [43] performs model aggregation by averaging the weights received from each client, updating the global generator and discriminator as follows.

$$\Theta_t^{\mathcal{G}} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \Theta_{bt}^{\mathcal{G}_m}, \quad \Theta_t^{\mathcal{D}} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \Theta_{bt}^{\mathcal{D}_m} \tag{4.14}$$

This aggregation process allowed the global model to improve without the central server needing access to the actual EEG data. After aggregation, the updated global models were sent back to the clients, and the process was repeated over multiple global epochs $\mathcal{T}$. The entire process is repeated until the model converges or until a predefined number of global epochs are completed.

4.1.8 Differential Privacy with Renyi Differential Privacy (RDP)

To quantify the loss of privacy during training, we employ Renyi Differential Privacy (RDP) [30], which provides a formal measure of privacy leakage. The RDP loss is calculated as Equation 4.15:

$$\mathcal{L}_{\text{RDP}} = \sum_{\alpha>1} \left(\frac{\alpha \cdot \mathcal{P}^2 \cdot \mathcal{L}}{2 \cdot \sigma_{\text{eff}}^2} + \frac{\log(1/\delta)}{\alpha - 1} \right) \tag{4.15}$$

where, α is the order of Renyi divergence, $\mathcal{P}$ is the is the sampling rate, $\mathcal{L}$ is the number of local epochs, σ_{eff} is the effective noise scale, which controls the balance between privacy and utility, and δ is the privacy parameter, related to the probability of re-identification.

This approach ensures that the synthetic data generated through this model preserves the essential features of the original EEG data while providing strong privacy guarantees against potential re-identification attacks.

4.2 Experimental Setup

4.2.1 Dataset

We used the BCI IV 2A dataset, which contains EEG recordings of nine subjects identified as A1 through A9, performing four motor imagery tasks (MIT) [90]. The data of each subject are divided into two sessions: Session I and Session II. For Dataset 2A, both Session I and Session II are characterized by the same data shape (including artifacts), denoted $(E_{ld}, Trial_N, D_{tp})$. In this notation, E_{ld} represents the number of electrodes or EEG channels, which is 22. The signals are filtered using a bandpass filter (b_{pf}) between 0.5 Hz and 100 Hz, and a notch filter (n_f) is applied at 50 Hz. $Trial_N$ signifies the total number of trials per session, which is 2,592 including artifacts. D_{tp} refers to the number of data points per trial, calculated as $D_{tp} = time_s \times s_f$, where s_f is the sampling rate. Given that the MIT lasts for 4 seconds with a s_f of 250 Hz, D_{tp} is $4 \times 250 = 1000$. Therefore, for each session (containing nine subjects), the shape of the dataset including artifacts is (22,2592,1000), where for each subject's data, the shape is (22,288,1000).

We used C3, C4, and Cz EEG channels in our study instead of 22 EEG channels as these channels are effective in capturing brain patterns associated with the recognition of imaginary movement states [92]. We removed artifacts from both sessions, and used Session I as train data (Tr), and Session II as test data (Te). Therefore, each subject had a variable shape of input data for train and test sets as shown below:

- Subject A1: Tr – $(273, 3, 1000)$, labels – $(273, 1)$, Te – $(281, 3, 1000)$, labels – $(281, 1)$
- Subject A2: Tr – $(270, 3, 1000)$, labels – $(270, 1)$, Te – $(283, 3, 1000)$, labels – $(283, 1)$
- Subject A3: Tr – $(270, 3, 1000)$, labels – $(270, 1)$, Te – $(273, 3, 1000)$, labels – $(273, 1)$

Algorithm 4.1 Privacy-Preserving Synthetic EEG Data Generation using Federated Spiking GAN

Require: EEG dataset $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m\}$, noise parameters ω, ζ, σ^2, model parameters: global epochs $\mathcal{T}$, local epochs $\mathcal{L}$, number of clients $\mathcal{M}$, learning rate η, sampling rate $\mathcal{P}$, privacy parameter γ, batch size $\mathcal{B}$, initial weights $\Theta_0^{\mathcal{G}}, \Theta_0^{\mathcal{D}}$

Ensure: Synthetic EEG dataset $\mathcal{X}_{\text{synthetic}}$

Initialize global generator $\mathcal{G}(\Theta^{\mathcal{G}})$ and discriminator $\mathcal{D}(\Theta^{\mathcal{D}})$ with $\Theta_0^{\mathcal{G}}, \Theta_0^{\mathcal{D}}$
Initialize temporally correlated noise parameters
Initialize Renyi Differential Privacy (RDP) parameters
Federated Learning Setup:
Partition $\mathcal{X}$ into $\mathcal{M}$ subsets: $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m \rightarrow \mathcal{X}_1^{(m)}, \mathcal{X}_2^{(m)}, \ldots, \mathcal{X}_t^{(m)}$ for each client m
For each client m, initialize local models $\mathcal{G}_m(\Theta_0^{\mathcal{G}_m}), \mathcal{D}_m(\Theta_0^{\mathcal{D}_m})$
for each global epoch $t \in \{1, 2, \ldots, \mathcal{T}\}$ **do**
  for each client $m \in \{1, 2, \ldots, \mathcal{M}\}$ **do**
    Local Training on Client m**:**
    for each local epoch $\ell \in \{1, 2, \ldots, \mathcal{L}\}$ **do**
      for each batch of data $(\mathcal{X}_{bt}^{(m)}, \mathcal{L}_t^{(m)}) \in \mathcal{X}_t^{(m)}$ **do**
        Generate latent vector $\mathcal{Z}_{bt}^{(m)} \sim \mathcal{P}(\mathcal{Z})$
        Generate temporally correlated noise:
        Initialize noise at time 0: $N_n(0) \sim \mathcal{N}(0, \sigma^2)$
        for each timestep μ_t **do**
          $N_n(\mu_t) = \zeta \cdot N_n(\mu_{t-1}) + \mathcal{N}(0, \sigma^2)$
        end for
        Forward pass through the local generator $\mathcal{G}_m$**:**
        Synthetic EEG data $\tilde{\mathcal{X}}_t^{(m)} = \mathcal{G}_{bt}^{(m)}(\mathcal{Z}_{bt}^{(m)}, \mathcal{L}_t^{(m)}; \Theta_{bt}^{\mathcal{G}_m}) + N_n(\mu_t)$
        Train the local discriminator $\mathcal{D}_m$**:**
        Real data loss: $\mathcal{L}_{\text{real}} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\mathcal{X}_{bt}^{(m)}, \mathcal{L}_t^{(m)}), 1)$
        Fake data loss: $\mathcal{L}_{\text{fake}} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\tilde{\mathcal{X}}_t^{(m)}, \mathcal{L}_t^{(m)}), 0)$
        Total discriminator loss: $\mathcal{L}_{bt}^{\mathcal{D}_m} = \mathcal{L}_{\text{real}} + \mathcal{L}_{\text{fake}}$
        Update $\Theta_{bt}^{\mathcal{D}_m}$ via gradient descent: $\Theta_{bt}^{\mathcal{D}_m} \leftarrow \Theta_{bt}^{\mathcal{D}_m} - \eta \nabla \mathcal{L}_{bt}^{\mathcal{D}_m}$
        Train the local generator $\mathcal{G}_m$**:**
        Generator loss: $\mathcal{L}_{bt}^{\mathcal{G}_m} = \mathcal{L}_{\text{bce}}(\mathcal{D}_{bt}^{(m)}(\tilde{\mathcal{X}}_t^{(m)}, \mathcal{L}_t^{(m)}), 1)$
        Update $\Theta_{bt}^{\mathcal{G}_m}$ via gradient descent: $\Theta_{bt}^{\mathcal{G}_m} \leftarrow \Theta_{bt}^{\mathcal{G}_m} - \eta \nabla \mathcal{L}_{bt}^{\mathcal{G}_m}$
      end for
    end for
    Send updated weights $\Theta_{bt}^{\mathcal{G}_m}, \Theta_{bt}^{\mathcal{D}_m}$ to the central server
  end for
  Aggregation at Central Server:
  Compute global parameters:

$$\Theta_t^{\mathcal{G}} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \Theta_{bt}^{\mathcal{G}_m}, \quad \Theta_t^{\mathcal{D}} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \Theta_{bt}^{\mathcal{D}_m}$$

  Update RDP Privacy Loss:
  Compute RDP loss:

$$\mathcal{L}_{\text{RDP}} = \sum_{\alpha > 1} \left(\frac{\alpha \mathcal{P}^2 \mathcal{L}}{2\sigma_{\text{eff}}^2} + \frac{\log(1/\delta)}{\alpha - 1} \right)$$

end for
Generate Synthetic EEG Data:
Generate new latent vector $\mathcal{Z}_{\text{new}} \sim \mathcal{P}(\mathcal{Z})$
Forward pass through global generator $\mathcal{G}$ to produce synthetic EEG data:

$$\mathcal{X}_{\text{synthetic}} = \mathcal{G}(\mathcal{Z}_{\text{new}}, \mathcal{L}_{\text{new}}; \Theta^{\mathcal{G}})$$

Output: Final synthetic EEG data $\mathcal{X}_{\text{synthetic}}$

- Subject A4: Tr – $(262, 3, 1000)$, labels – $(262, 1)$, Te – $(228, 3, 1000)$, labels – $(228, 1)$
- Subject A5: Tr – $(262, 3, 1000)$, labels – $(262, 1)$, Te – $(276, 3, 1000)$, labels – $(276, 1)$
- Subject A6: Tr – $(219, 3, 1000)$, labels – $(219, 1)$, Te – $(221, 3, 1000)$, labels – $(221, 1)$
- Subject A7: Tr – $(271, 3, 1000)$, labels – $(271, 1)$, Te – $(277, 3, 1000)$, labels – $(277, 1)$
- Subject A8: Tr – $(264, 3, 1000)$, labels – $(264, 1)$, Te – $(271, 3, 1000)$, labels – $(271, 1)$
- Subject A9: Tr – $(237, 3, 1000)$, labels – $(237, 1)$, Te – $(264, 3, 1000)$, labels – $(264, 1)$

The labels in the dataset correspond to four motor imagery tasks. These tasks are classified as follows [90]: Class 1 (left), represented by event type 769 (0x0301) and referred to as CL1; Class 2 (right), represented by event type 770 (0x0302) and referred to as CL2; Class 3 (foot), represented by event type 771 (0x0303) and referred to as CL3; and Class 4 (tongue), represented by event type 772 (0x0304) and referred to as CL4. For the purpose of generating synthetic data, only the train data from Session I was utilized.

4.2.2 Data Pre-processing

The EEG data for each channel is normalized to the range $[0, 1]$ using the min-max normalization method. The normalized value, E_i, is calculated as follows:

$$E_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \tag{4.16}$$

where X_i represents the data point from channel i, and $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the EEG data for channel i, respectively.

4.2.3 Model Architecture

The Spiking GAN architecture consists of a Spiking generator and a ANN-based discriminator. Using the temporal patterns present in spiking neurons, the Spiking Generator creates synthetic EEG data. It preserves important temporal properties that resemble biological

cerebral activity, while it converts a low-dimensional latent space, a 100-dimensional vector, into high-dimensional EEG signals. Starting from a typical normal distribution, this latent vector is then concatenated with a label embedding to guarantee that the generator outputs EEG data specific to a class. By adding pertinent information to the latent space, this combination conditions the generating process on the intended label.

Three fully connected layers make up the architecture of the generator. In order to begin the transformation into a high-dimensional EEG representation, the first layer processes the concatenated latent vector and label embedding, increasing the dimensionality of the input. The second layer continues this process by increasing the complexity of the features. The third layer, which peaks at 1000 neurons, fully contains the features required to generate the EEG signal. Leaky Integrate-and-Fire (LIF) neurons generate spiking behavior after each fully connected layer, simulating the dynamics of membrane potentials and guaranteeing that the resulting data resemble actual EEG signals. The last layer utilizes a Sigmoid activation function to scale the output to the range [0, 1], corresponding to the normalized EEG data format. It has 3000 neurons, or 3 channels with 1000 datapoints each.

The generated EEG data authenticated by the ANN-based discriminator gradually reduces the dimensionality of the input while extracting important features that differentiate real from fake data. In the first fully connected layer, it starts with the entire 3000-dimensional EEG vector, concatenates it with the label embedding, and reduces it to 1000 neurons. Subsequent layers of this downsampling technique concentrate on fine-tuning the features suggestive of data authenticity, with the second layer lowering the dimensionality to 750 neurons and the third layer to 500 neurons. Before the final layer brings the dimensionality back to 1000 neurons, the fourth layer modifies the dimensionality significantly to 750 neurons, enabling a wider exploration of the feature space. Before the final categorization, this symmetric structure guarantees a rich representation of the input data. The output layer, which consists of a single neuron, uses a Sigmoid activation function to output a value between 0 and 1. This value indicates the chance that the input data is real and signals the

generator to produce more realistic synthetic data.

4.3 Results and Discussion

We modified several existing GAN-based models to align with the Renyi Differential Privacy (RDP) framework, ensuring privacy preservation. Among these models were TimeGAN [67], RCGAN [66], CRNN-GAN [72], and Clare-GAN [108]. By incorporating gradient clipping and Gaussian noise into their original architectures, we developed RDP-TimeGAN, RDP-RCGAN, RDP-CRNN-GAN, and RDP-Clare-GAN, respectively. A clipping norm of 1.0 was employed with varying noise multipliers [0.100, 0.311, 0.522, 0.733, 0.944, 1.156, 1.367, 1.578, 1.789, 2.000] in all methodologies. The other parameter details are shown in Table 4.1. This configuration allowed for the evaluation of the privacy-preserving objective, facilitating a fair comparison among the adapted RDP models.

Table 4.1: Common parameters used in this experiment.

Parameter	**Value**
Privacy loss parameter (δ)	10^{-3}
Rényi divergence order α	10
Number of local epochs ($\mathcal{L}$)	5
Number of global epochs ($\mathcal{T}$)	300
Sampling rate ($\mathcal{P}$)	0.2
Number of clients ($\mathcal{M}$)	5
Batch size ($\mathcal{B}$)	64
Learning rate (η)	0.0002
Noise scale (σ)	[0.100, 0.311, 0.522, 0.733, 0.944, 1.156, 1.367, 1.578, 1.789, 2.000]
Data split among clients	Equally among the clients
Total epochs for other models	300

4.3.1 Evaluation Scenarios

We performed the following tests to evaluate the usability of the generated EEG data using two well-known classification architectures; EEGNet [94], which generalizes across the BCI paradigms, and ATCNet [93], which shows excellent performance with motor imagery Dataset 2A.

- $Train_{(Tr+Sy)}, Test_{(Te)}$**:** Trained on real EEG samples of train set plus the synthetic dataset (augmented dataset), and tested on real EEG samples of test dataset.

- $Train_{(Tr_{50}+Sy_{50})}, Test_{(Te)}$**:** Trained on 50% real EEG samples from train set plus the 50% synthetic dataset, and tested on real EEG samples of test dataset.

- $Train_{(Tr)}, Test_{(Sy)}$**:** Trained only on the real EEG samples of train dataset, and tested on synthetic dataset.

- $Train_{(Sy)}, Test_{(Tr)}$**:** Trained on synthetic dataset, and tested on real EEG samples of train dataset.

- $Train_{(Sy)}, Test_{(Te)}$**:** Trained on synthetic dataset, and tested on real EEG samples of test dataset.

We evaluated the classification performance of each model over 30 runs to ensure the validity of our findings. The real test set (Te) and the real training set (Tr) for each subject were denoted by the A0PE and A0PT datasets, respectively, with 'P' representing the subject identifier. We created synthetic data using the entire A0PT dataset for our proposed method and the comparison approaches. Following the creation of the synthetic data, we used the EEGNet and ATCNet classifiers for training and analyzed them in a variety of scenarios. We conducted experiments using various temporal correlation coefficients (ρ values) ranging from 0.1 to 0.9. Our observations indicated that data quality, measured in terms of accuracy for the five different evaluation scenarios, from $Train_{(Tr+Sy)}, Test_{(Te)}$ to $Train_{(Sy)}, Test_{(Te)}$, was optimal at a ρ value of 0.55. Consequently, we used this value of ρ for all subsequent comparisons with other methods.

The experimental results for our proposed Spiking GAN model, RDP-TimeGAN, RDP-RCGAN, RDP-CRNN-GAN, and RDP-Clare-GAN are shown in Figures 4.1,4.2,4.3,4.4, and 4.5. These figures show the comparison of four state-of-the-art models evaluated using ATC-Net and EEGNet for subject A1. These figures illustrate the models' performance across

various noise multipliers for subject A1, and these also illustrate how different levels of noise impact the models' ability to preserve privacy while maintaining efficient data generation capabilities. The rest of the graphs for other subjects evaluated with both models are shown in Appendix A. These evaluations highlight the crucial trade-off between privacy (quantified by ϵ) and model accuracy. We observed that both architectures give comparable performance for most of the subjects but with a high degree of variability highlighting the need for individualized utility optimization (deep learning models) for similar privacy budgets. Secondly, our model achieves a balance of privacy and utility for the noise multiplier range of 1.1 to 1.6, with corresponding privacy budget values ranging from 1.88 to 2.63.

We also found that for our method, the accuracy of both classifiers (EEGNet and ATC-Net) for scenario 1, $Train_{(Tr+Sy)}, Test_{(Te)}$ was comparable to or slightly less than that for scenario 2, $Train_{(Tr_{50}+Sy_{50})}, Test_{(Te)}$. This suggests that reducing the real training data by 50% and augmenting with 50% synthetic data maintains or even slightly improves accuracy. This indicates that our privacy-preserving model can be deployed in BCI environments which are less sensitive to accuracy variability, such as BCI games. In contrast, other models such as RDP-TimeGAN, RDP-RCGAN, RDP-CRNN-GAN, and RDP-Clare-GAN showed a decrease in accuracy under similar conditions. The difference in accuracy between $Train_{(Tr)}, Test_{(Sy)}$ and $Train_{(Sy)}, Test_{(Tr)}$ is more noticeable compared to other methods. However, in our approach, this difference is smaller, indicating the high quality of the synthetic EEG data generated. Furthermore, in the case of $Train_{(Sy)}, Test_{(Te)}$, our method produces better accuracy than other approaches, suggesting a better data utility.

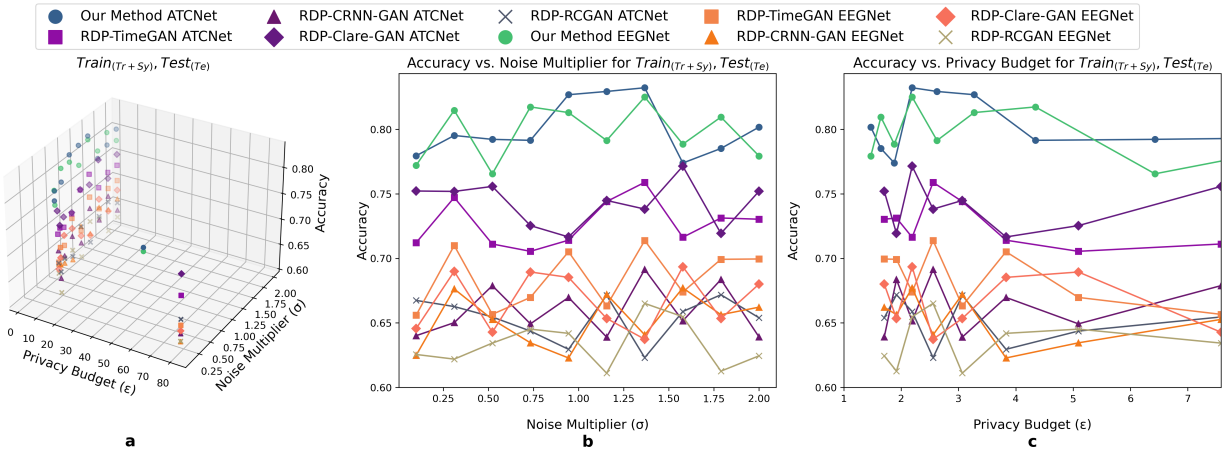


Figure 4.1: Comparison of different models under various privacy budgets and noise multipliers for subject A1 ($Train_{(Tr+Sy)}, Test_{(Te)}$).

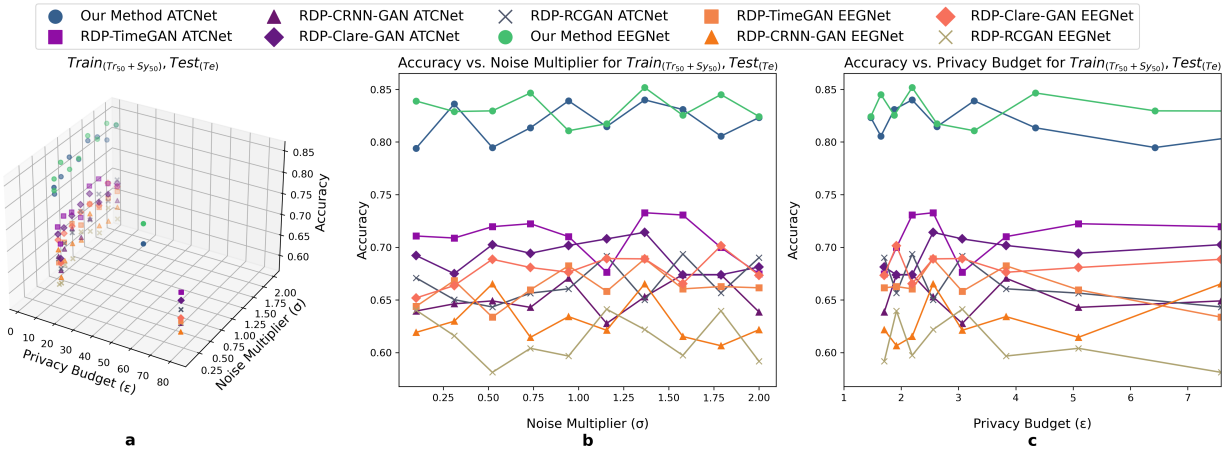


Figure 4.2: Comparison of different models under various privacy budgets and noise multipliers for subject A1 ($Train_{(Tr50+Sy50)}, Test_{(Te)}$).

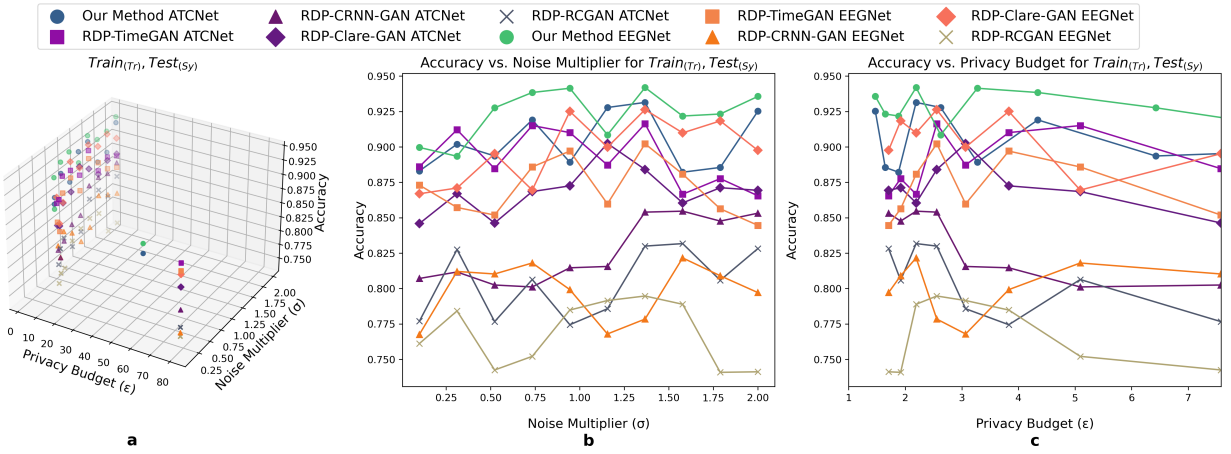


Figure 4.3: Comparison of different models under various privacy budgets and noise multipliers for subject A1 ($Train_{(Tr)}, Test_{(Sy)}$).

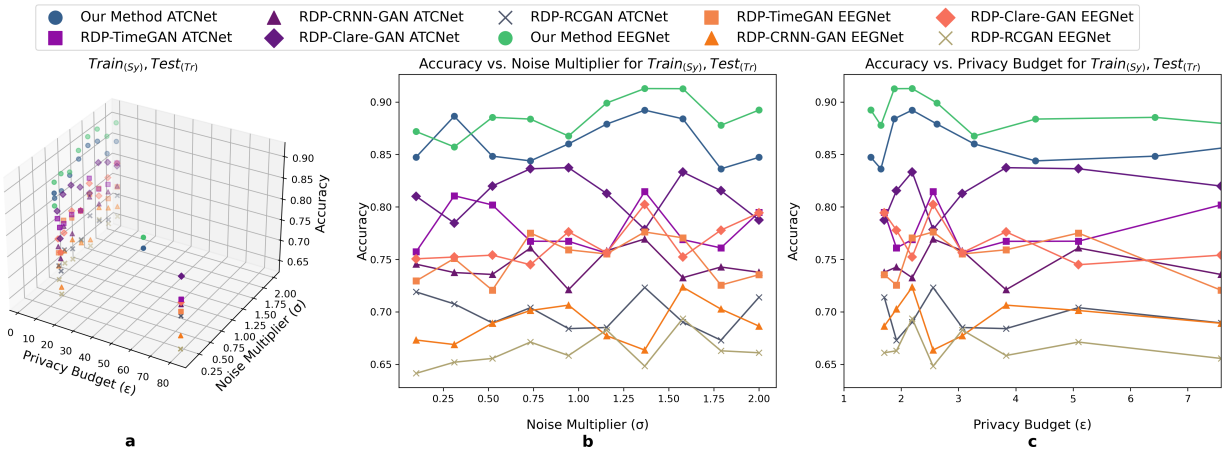


Figure 4.4: Comparison of different models under various privacy budgets and noise multipliers for subject A1 ($Train_{(Sy)}, Test_{(Tr)}$).

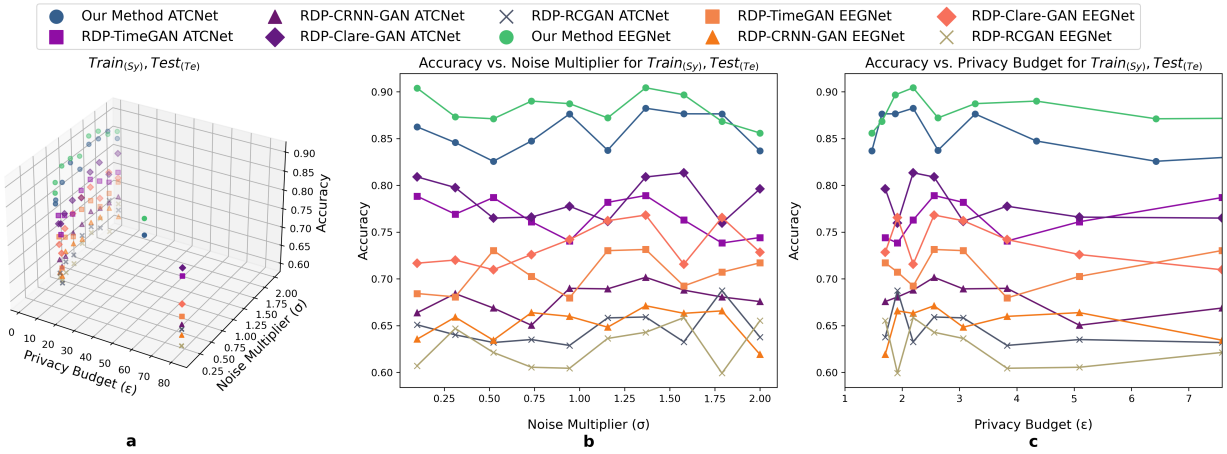


Figure 4.5: Comparison of different models under various privacy budgets and noise multipliers for subject A1 ($Train_{(Sy)}, Test_{(Te)}$).

4.3.2 Data Fidelity

The fidelity and usability of the generated EEG data is evaluated using Dynamic Time Warping (DTW) [109, 110] and Spectral Similarity (SS) [111] techniques. DTW aligns the sequences to minimize the overall distance in order to determine how similar the actual and synthetic EEG signals are to each other. DTW is useful for assessing the temporal dynamics of the data. Low values indicate that the temporal dynamics of the synthetic data closely follows the real data. SS gives a comparison of the power spectral density (PSD) between synthetic and actual EEG data. A high value of spectral similarity means that the synthetic data maintain the frequency characteristics of the genuine data. The SS score ranges from 0 to 1. The range of DTW and SS metrics obtained for the generated EEG data, shown in Figures 4.6 and 4.7, demonstrates that our method produces high-fidelity synthetic data that closely resemble the real data [112]. The inter-subject variability observed in the DTW and SS scores suggests that the synthetic data may not have fully integrated the individual-specific characteristics of the EEG signals. This variability could potentially be reduced by using more subject-specific modeling approaches in the synthetic data generation process. Overall, the fidelity of the synthetic data is high, despite considerable variation between subjects. This implies that our technique preserves the essential features of EEG signals,

such as frequency content and temporal dynamics, while adapting well to different subjects.

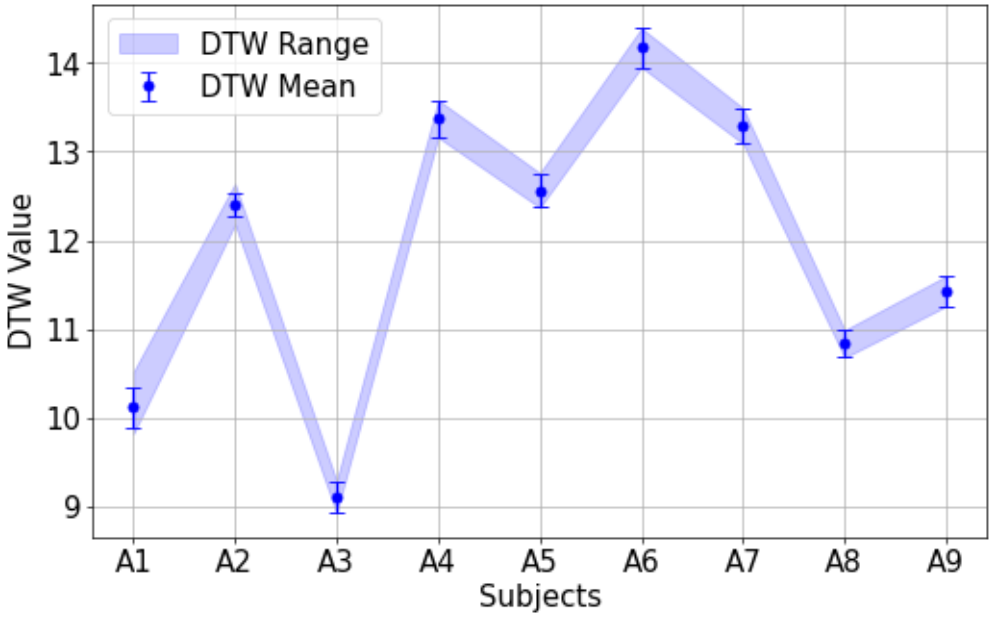


Figure 4.6: The plots shows the mean Dynamic Time Warping (DTW) values with standard deviations and ranges obtained for the generated EEG data of all subjects, A1 to A9.

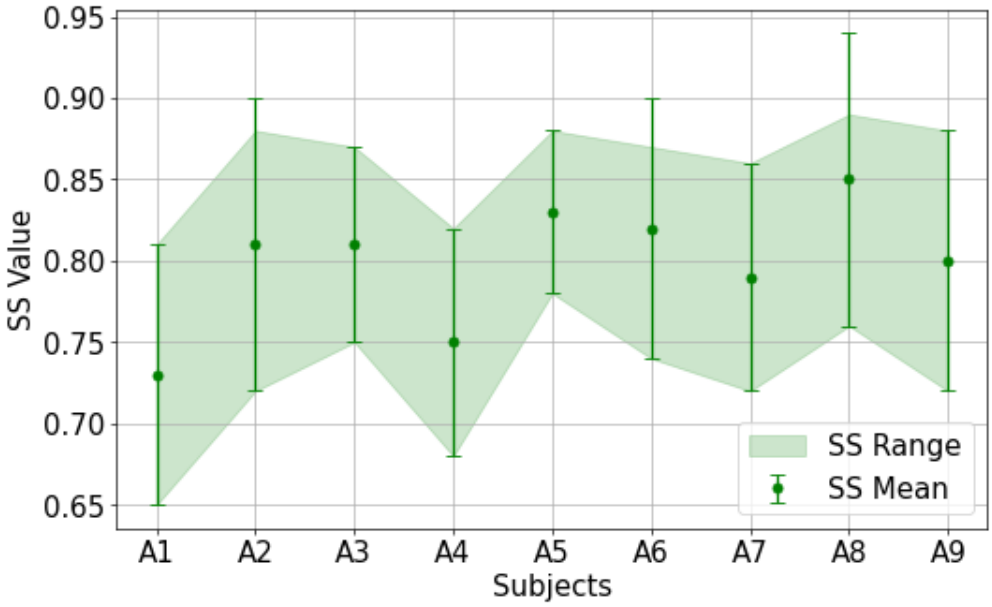
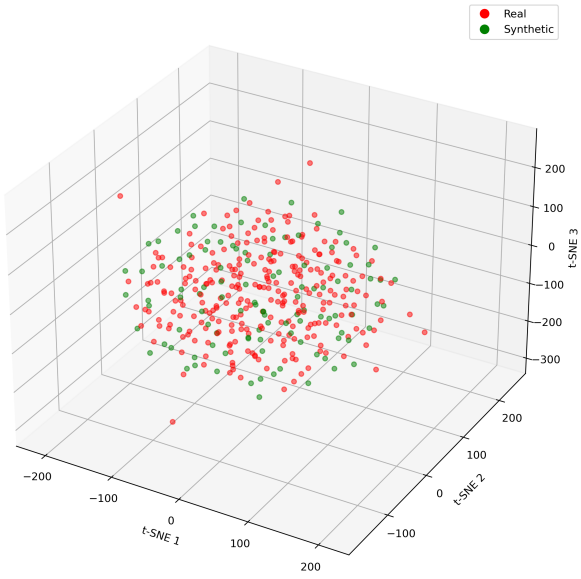


Figure 4.7: The plot shows the mean Spectral Similarity Score (SS) values with standard deviations ranges obtained for the generated EEG data of all subjects, A1 to A9.

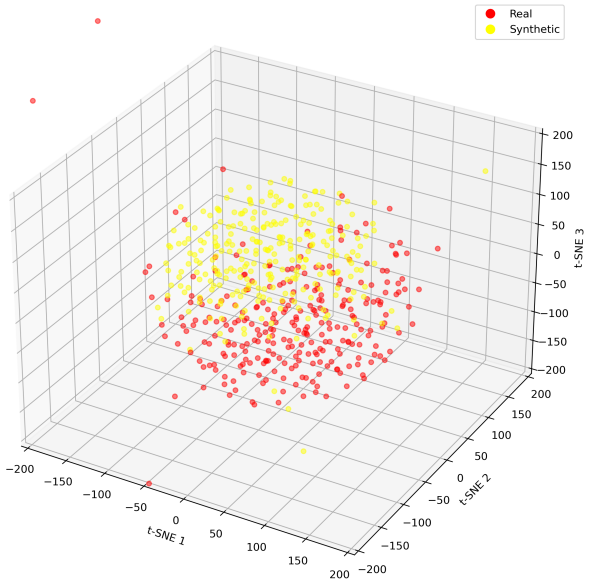
4.3.3 Visualization of High-Dimensional EEG Data Using 3D t-SNE

In order to compare the original with generated EEG data, we used t-Distributed Stochastic Neighbor Embedding (t-SNE), which visualizes high-dimensional datasets by projecting from higher dimensions to lower ones [113]. Through the t-SNE visualizations, the degree of convergence of the synthetic data generation process can be assessed, facilitating the identification of similarities or dissimilarities between the real and synthetic datasets.

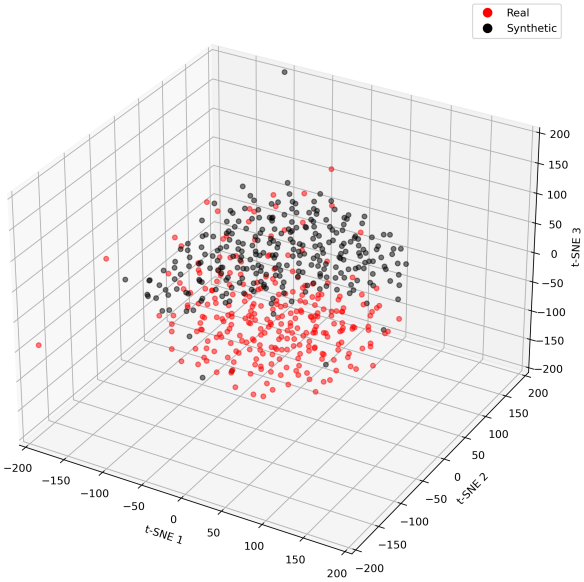
Figure 4.8a illustrates the spatial distribution of the real and synthetic EEG data points generated by our proposed Spiking GAN model and Figures 4.8b to 4.8e of four other models including RDP-TimeGAN, RDP-RCGAN, RDP-CRNN-GAN and RDP-Clare-GAN. Due to the large number of images and their file size, only one subject's results are shown as a representation of the dataset. Additional figures for other subjects are included in Appendix B.



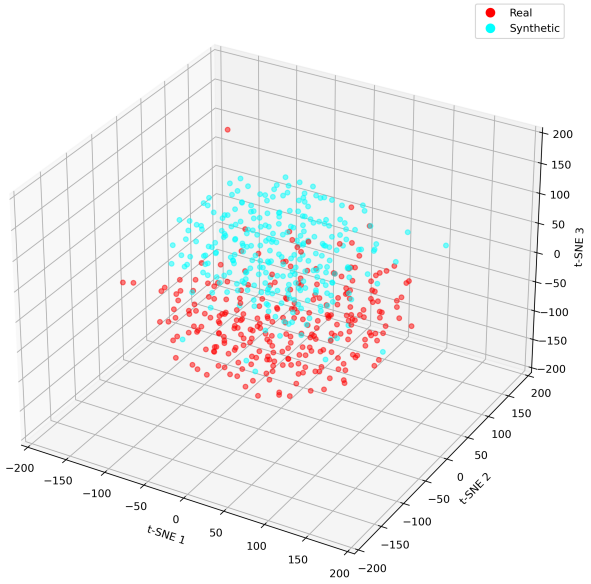
(a) Real data vs. proposed Spiking GAN



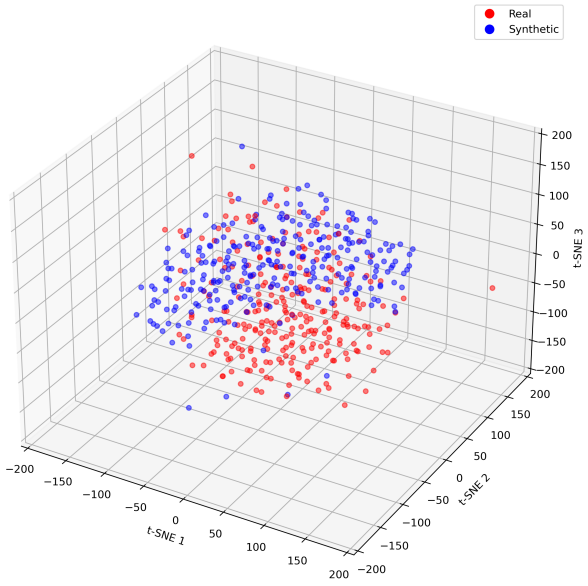
(b) Real data vs. RDP-TimeGAN



(c) Real data vs. RDP-CRNN-GAN



(d) Real data vs. RDP-Clare-GAN



(e) Real data vs. RDP-RCGAN

Figure 4.8: 3D t-SNE visualization of high-dimensional EEG data for subject A1. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

The clusters and patterns observed in the t-SNE plots demonstrate that our Spiking GAN model generates synthetic EEG data that closely approximate the distribution of authentic EEG data, ensuring privacy preservation while maintaining high data fidelity. The performance is also considerably better than that of the RDP-CRNN-GAN and RDP-RCGAN models, which exhibit a greater degree of homogeneity relative to those of RDP-TimeGAN and RDP-Clare-GAN.

Experimental results consistently indicate that our proposed Spiking GAN model strikes a balance between privacy and accuracy more effectively than the baseline models. Furthermore, the lower epsilon values achieved signify enhanced privacy preservation while maintaining high data fidelity and usability.

4.4 Summary

In this chapter, we introduce a novel methodology for generating privacy-preserving synthetic EEG data through the application of a Federated Spiking GAN framework, which incorporates Spiking Neural Networks (SNNs) in the generator, an Artificial Neural Network (ANN)-based discriminator, and a temporally correlated noise model. Unlike other methods, ours takes advantage of the temporal dynamics inherent in SNNs to effectively capture the essential characteristics of EEG signals. By integrating Renyi Differential Privacy (RDP) within the federated learning paradigm, we ensure robust privacy guarantees without compromising the utility of the data. Our experiments demonstrate the efficacy of our approach in producing high-fidelity synthetic EEG data while maintaining the balance of privacy-utility. Extensive comparative analysis with other GAN-based privacy-preserving methodologies, such as RDP-TimeGAN, RDP-CRNN-GAN, RDP-RCGAN, and RDP-Clare-GAN, underscores the performance of our method in preserving the temporal dynamics of EEG signals and providing strong privacy.

Although we presented a federated framework that incorporates Spiking GANs to decentralize and strengthen EEG data security, there is a growing demand for more granular and adaptive privacy control mechanisms. This need lays the foundation of the next chapter which proposes a Hierarchical Privacy Framework using GFlowNet and Federated Split Learning (FSL). Chapter 5, built upon federated privacy concepts discussed in this chapter, introduces hierarchical privacy layers which allow abstraction of at different levels, adjusting privacy at multiple latent levels, showing a more enhanced balance between privacy and utility while maintaining privacy concern. GFlowNet's generative capabilities further extend the communication security and its overheads with the help of dividing the learning process among clients and servers. Thus, addressing current limitations through multi-layered privacy mechanisms, with adaptive sensitivity factor incorporated at these levels.

5

Hierarchical Privacy using GFlowNet and Federated Split Learning

In this work, we present a novel approach to privacy-preserving EEG data generation, combining Federated Split Learning (FSL) with Hierarchical Privacy-Adaptive Autoencoders, Secure Aggregation, and Generative Flow Networks (GFlowNet). Our method is designed to ensure strong privacy guarantees while maintaining high data utility, specifically tailored for the complex, spatio-temporal nature of EEG data. The hierarchical architecture of the autoencoder enables multi-level feature extraction, effectively capturing both spatial and temporal dependencies in the EEG signals. By leveraging Rényi Differential Privacy (RDP) and adaptive noise scaling, our model anonymizes sensitive brain signals during the data generation process. The FSL architecture allows client-side processing of raw EEG data, followed by server-side reconstruction and synthetic data generation using GFlowNet. To enhance

privacy further, Secure Aggregation is applied, ensuring that individual data contributions are protected even during communication between clients and the server. We evaluate our approach under various privacy budgets, demonstrating a balanced privacy-utility trade-off. Our findings show that this method provides robust privacy protection, maintaining both spatial and temporal coherence in the generated synthetic EEG data, while offering flexibility in real-world privacy-sensitive applications such as healthcare and neuroscience.

5.1 Methodology

This section outlines our approach for generating privacy-preserving synthetic EEG data using Federated Split Learning (FSL) [114] employing a hierarchical encoder-decoder architecture inspired by [94, 115] and Generative Flow Networks (GFlowNet) [116]. To achieve a balance between high data utility and strong privacy guarantees, we integrate Rényi Differential Privacy (RDP) [30] and Secure Aggregation [117], protecting sensitive EEG data while enabling the generation of high-quality synthetic data. The methodology is outlined in Algorithm 5.1.

5.1.1 Federated Split Learning (FSL)

Federated Split Learning (FSL) divides the learning process between the client and the server. Clients process raw EEG data locally, ensuring that the data never leave the client's device [114]. Only anonymized latent representations are shared with the server, which performs the remaining computation without accessing the raw EEG data.

In our FSL setup, both the server and client components were simulated on a personal computer, to replicate a federated learning environment. We simulated 5 clients, each representing an independent entity in the network, with each client handling its unique subset of the EEG dataset. The raw EEG data was divided into non-overlapping segments, ensuring that each client processed a different portion of the dataset. This configuration simulates real-world situations in which various devices gather data independently.

Using the hierarchical encoder, each client processed its local data to create latent vari-

ables l_1, l_2, l_3 that captured various temporal and spatial characteristics of the EEG data. These latent variables were then anonymized using RDP to ensure that they could not be traced back to the original EEG signals. To further improve privacy, each client added a random mask m_i to anonymized latent variables after implementing RDP. After that, a centralized server received the masked latent variables $(l'_i + m_i)$ for aggregation. The server, also simulated on the same machine, acted as the central aggregator. It executed secure aggregation after receiving the masked latent variables from each client to guarantee that no client's data was revealed. After aggregation, the server reconstructed the EEG signals using the hierarchical decoder. Finally, the server used GFlowNet to create synthetic EEG data while preserving the original EEG data's temporal and spatial structure.

5.1.2 Client-Side: Processing EEG Data with Hierarchical Encoder

On the client side, the raw EEG data is processed using a hierarchical encoder architecture inspired by [94, 115], designed to capture both spatial and temporal features of the EEG data across multiple levels of abstraction.

The encoder processes the data in three stages, producing latent variables l_1, l_2, l_3, which capture different aspects of the EEG signals:

- **First Block (Temporal Filter)**: The initial stage captures basic temporal patterns using depth-wise temporal convolution, producing the latent variable l_1, modeled as:

$$l_1 \sim \mathcal{N}(\mu_1, \sigma_1^2) \tag{5.1}$$

  where μ_1 and σ_1 are the mean and variance learned from the data.

- **Second Block (Spatial Filter)**: This stage applies parallel convolutions to capture

spatial features across EEG channels, resulting in the latent variable l_2:

$$l_2 \sim \mathcal{N}(\mu_2, \sigma_2^2) \tag{5.2}$$

- **Third Block (Separable Convolution)**: The final block refines both spatial and temporal features, producing l_3, which captures the remaining dependencies in the EEG data:

$$l_3 \sim \mathcal{N}(\mu_3, \sigma_3^2) \tag{5.3}$$

These hierarchical latent variables, l_1, l_2, l_3, capture progressively more abstract representations of the EEG data. These variables are then prepared for transmission to the server after privacy mechanisms are applied. For details on the encoder configuration, refer to Table 5.1.

5.1.3 Anonymization with Rényi Differential Privacy (RDP)

To protect latent variables before transmission, Rényi Differential Privacy (RDP) [30] is applied. RDP ensures that latent representations cannot be traced back to the original EEG data by adding controlled Gaussian noise. The privacy budget is distributed evenly across the latent spaces to balance privacy and utility.

The total privacy budget ϵ_{total} is divided equally across the three latent spaces:

$$\epsilon_1 = \epsilon_2 = \epsilon_3 = \frac{\epsilon_{\text{total}}}{3} \tag{5.4}$$

This approach ensures a consistent privacy guarantee across the different levels of feature abstraction.

Noise Addition to Latent Variables: Noise is added to each latent variable l_i (where $i = 1, 2, 3$) as follows:

$$l_i' = l_i + \mathcal{N}(0, \sigma_i^2) \tag{5.5}$$

where $\mathcal{N}(0, \sigma_i^2)$ represents Gaussian noise with variance σ_i^2. The noise scale σ_i is determined by the privacy budget ϵ_i and the data sensitivity Δf:

$$\sigma_i = \frac{\Delta f}{\epsilon_i} \tag{5.6}$$

Here, Δf represents the sensitivity of the data, ensuring that each latent variable is protected while preserving data utility.

5.1.4 Ensuring Privacy with Secure Aggregation

To further enhance privacy, we implement Secure Aggregation [117], which ensures that individual client data remains protected during communication with the server. Each client applies a random mask m_i to anonymized latent variables before transmission. The uniform distribution over the range $[-1, 1]$ was used to generate the random masks m_i. This ensures that even if the server or an adversary attempts to intercept the communication, it cannot access the latent variables of any individual client.

The masked latent variables are sent as:

$$l_i'' = l_i' + m_i \tag{5.7}$$

Upon receiving the masked latent variables l_i'' from all clients, the server aggregates the masked variables and removes the masks using a process called mask cancellation [118]. The server only works with the aggregated data, preserving client privacy:

$$\sum_{i=1}^{n} l_i'' - \sum_{i=1}^{n} m_i = \sum_{i=1}^{n} l_i' \tag{5.8}$$

This process ensures that the server can never access individual client data, as it only deals with the aggregated results of the masked latent variables, further enhancing the overall privacy of the system.

5.1.5 Server-Side: Decoding and Reconstruction

Once the anonymized latent variables are received by the server, the hierarchical decoder, mirroring the encoder structure, reconstructs the original EEG signals. The decoder is designed to ensure accurate reconstruction of the temporal and spatial features.

- **First Block (Separable Convolution)**: This stage uses separable transposed convolutions to upsample the latent variables and reconstruct the spatial-temporal features.
- **Second Block (Spatial Filter)**: Applies parallel transposed convolutions to reconstruct spatial features.
- **Third Block (Temporal Filter)**: Reconstructs temporal dynamics in the EEG data using transpose convolution.

The reconstruction loss is computed as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + D_{KL}(q(l|d)\|p(l)) \tag{5.9}$$

where $\mathcal{L}_{\text{recon}}$ measures temporal alignment using Dynamic Time Warping (DTW) [119], and D_{KL} is the Kullback-Leibler (KL) Divergence [120], ensuring that the latent variables follow a Gaussian distribution.

5.1.6 Generating Synthetic EEG Data with GFlowNet

After verifying the quality of the latent variables, the server uses Generative Flow Networks (GFlowNet) [116] to generate synthetic EEG data. GFlowNet models the generation process as a flow through latent states, ensuring that the generated data is spatially and temporally coherent.

The generative process for the entire sequence of EEG data points is defined as [121, 122]:

$$P(y_1, y_2, \ldots, y_n|l') = P(y_1|l') \prod_{i=2}^{n} P(y_i|y_{i-1}, l') \tag{5.10}$$

where, the first data point y_1 is generated independently based on latent variables l', and each subsequent data point y_i is generated conditionally based on the previous point y_{i-1} and the latent variables l'. This structure ensures that the generative process begins with the independent generation of y_1 and then follows a conditional sequence for the subsequent points.

Table 5.1: Encoder Layer Configuration

Block No.	**Blocks**	**SL No.**	**Layers**	**Kernel**	**I/P depth**	**O/P depth**
1	First block (temporal filter)	1	Convolution 2D [Depth-wise convolution (Temporal Filter)]	(1, 125)	1	8
		2	Batch Norm 2D [Default parameters]	-	-	-
		3	Attention Layer [Adds temporal attention scores]	-	8	8
2	Second block (spatial filter)	4	Convolution 2D (Parallel 1) [Depth-wise convolution (Spatial Filter)]	(1, 3)	8	16
		5	Batch Norm 2D [Default parameters]	-	-	-
		6	Activation [ELU]	-	-	-
		7	Dropout [p = 0.5]	-	-	-

Continued on next page

Block No.	Blocks	SL No.	Layers	Kernel	I/P depth	O/P depth
3	Third Block (Separable Conv.)	8	Convolution 2D [Depth-wise convolution (Separable Conv.)]	(1, 32)	16	16
		9	Convolution 2D [Pointwise convolution]	(1, 1)	16	16
		10	Activation [ELU]	-	-	-
		11	Average pooling [Default parameters]	(1, 8)	-	-
		12	Dropout [p = 0.5]	-	-	-
4	Sample layer	13	Convolution 2D [Pointwise convolution]	(1, 1)	16	32

Table 5.2: Decoder Layer Configuration

Sl. No.	Blocks	SL No.	Layers	Kernel	I/P depth	O/P depth
1	Third Block (Separable Convolution)	1	Dropout [p = 0.5]	-	-	-

Continued on next page

Sl. No.	Blocks	SL No.	Layers	Kernel	I/P depth	O/P depth
		2	Upsample [Default parameters]	(1, 8)	-	-
		3	Activation [ELU]	-	-	-
		4	Batch Norm 2D [Default parameters]	-	-	-
		5	Transpose Convolution 2D [Pointwise convolution]	(1, 1)	32	16
		6	Transpose Convolution 2D [Depth-wise convolution]	(1, 32)	16	16
2	Second Block (Spatial Filter)	7	Dropout [p = 0.5]	-	-	-
		9	Activation [ELU]	-	-	-
		10	Batch Norm 2D [Default parameters]	-	-	-
		11	Transpose Convolution 2D [Depth-wise convolution (Spatial Filter)]	(1, 3)	16	8
3	First Block (Temporal Filter)	12	Batch Norm 2D [Default parameters]	-	-	-

Continued on next page

Sl. No.	Blocks	SL No.	Layers	Kernel	I/P depth	O/P depth
		13	Transpose Convolution 2D [Depth-wise convolution (Temporal Filter)]	(1, 125)	8	8

5.2 Experiments and Results

5.2.1 Dataset

We used the BCI IV 2B dataset, which contains EEG recordings of nine subjects identified as B1 through B9, performing two motor imagery tasks (MIT) with and without feedback [91]. We used the recordings from all the EEG channels (C3, Cz and C4).The signals were filtered using a bandpass filter between 0.5 Hz and 100 Hz, and a notch filter was applied at 50 Hz. All subjects participated in five sessions. Three sessions, S_{III} through S_{V}, had real-time feedback, while the first two, S_{I} and S_{II}, consisted of training data without any feedback. Each subject completed 60 trials for each MI class during the non-feedback motor imagery (MI) sessions, for a total of 120 trials. During the feedback sessions, there were 80 trials for each MI class, for a total of 160 trials in a session. The average duration of the trial was 4 seconds. Each participant completed 720 tests in total, although some were not completed due to differences in the experiment. Signal data from each trial were collected, with a focus on a segment of approximately 4 seconds. A 4-second frame sampled at a frequency of 250 Hz corresponded to 1000 data points each trial.

5.2.2 Experimental Setup

We divided the overall dataset $\bar{D}$ into training and testing subsets, denoted as Tr (for the real training data) and Te (for the real test data), respectively. Here, (1) $\bar{D}$: sessions (S_{I}

Algorithm 5.1 Privacy-Preserving EEG Data Generation Using FSL, Hierarchical Encoder-Decoder, and GFlowNet

Input: EEG data $D = \{d_1, d_2, \ldots, d_K\}$ for K clients
Output: Synthetic EEG data Y generated by GFlowNet
Initialization:
Initialize hierarchical encoder-decoder at clients and server
Initialize GFlowNet model at server
Set privacy budget ϵ_{total} for RDP, divide as $\epsilon_1 = \epsilon_2 = \epsilon_3 = \frac{\epsilon_{\text{total}}}{3}$
Set sensitivity Δf for each latent space
Set communication rounds $\bar{R}$
Federated Split Learning (FSL) - For each round $r \in R$:
for each client k **do**
  (i) Input EEG data d_k into hierarchical encoder to compute latent variables:
    $l_1^k, l_2^k, l_3^k = \text{Encoder}(d_k)$
  (ii) Apply RDP to anonymize latent variables:
    $l_i'^k = l_i^k + \mathcal{N}(0, \sigma_i^2), \quad \sigma_i = \frac{\Delta f}{\epsilon_i}$ for $i = 1, 2, 3$
  (iii) Apply secure aggregation by adding random masks:
    $l_i''^k = l_i'^k + m_i^k \quad$ for $i = 1, 2, 3$
  (iv) Transmit masked, anonymized latent variables $l_1''^k, l_2''^k, l_3''^k$ to server
end for
Server-Side:
(i) Aggregate masked latent variables from clients using secure aggregation:
    $\text{Aggregate}_{l_i'} = \sum_{k=1}^{K} l_i''^k - \sum_{k=1}^{K} m_i^k \quad$ for $i = 1, 2, 3$
(ii) Decode aggregated latent variables to reconstruct EEG data:
    $\hat{d} = \text{Decoder}(\text{Aggregate}_{l_1'}, \text{Aggregate}_{l_2'}, \text{Aggregate}_{l_3'})$
(iii) Compute reconstruction loss:
    $\mathcal{L}_{\text{recon}} = \text{DTW}(D_{\text{Aggregate}}, \hat{d})$
GFlowNet Training:
for each time step i in EEG sequence **do**
  Generate next EEG point y_i using flow transition
  Update GFlowNet parameters to minimize flow-based loss
end for
Model Update:
(a) Update client encoder weights using $\mathcal{L}_{\text{recon}}$
(b) Update server decoder and GFlowNet weights using $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{flow}}$
Output:
Output synthetic EEG data Y generated by GFlowNet

through S_{V}), (2) Tr : sessions (S_{I} through S_{III}), (3) Te : sessions (S_{IV} and S_{V}). After splitting $\bar{D}$ into Tr and Te, we removed the trials with artifacts. Figures 5.1, 5.2, 5.3, 5.4 and 5.5 represent the accepted and rejected trials for Session I, II, III, IV and V, respectively.

We used the subset Tr for training and created synthetic data, Sy, equal in number to the training data samples in Tr. In addition, we combined Sy with Tr to create an augmented dataset Aug. We tested our approach and other state-of-the-art techniques with a range of ϵ values: 0.5, 1, 3, 6, 9, 12, and 15. The experiments were carried out using CUDA 11.8, cuDNN 9.3.0, PyTorch 2.4.1+cu118, Python 3.8.19, and an NVIDIA GeForce RTX 4050 GPU.

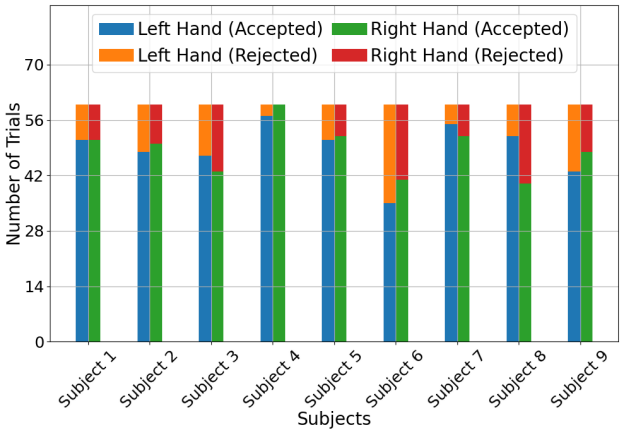


Figure 5.1: 2B (Session I): Accepted and Rejected/Artifact Trials by Subject and Task

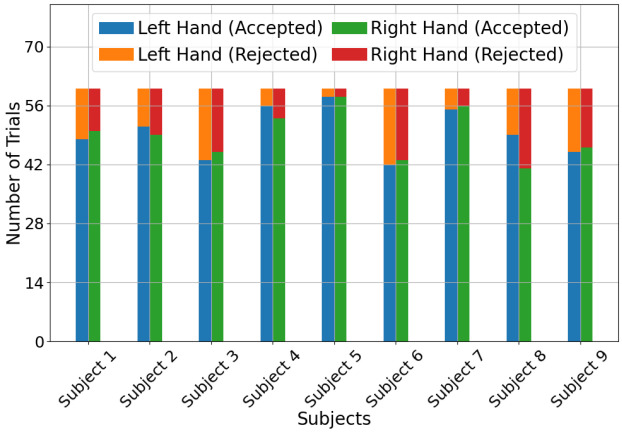


Figure 5.2: 2B (Session II): Accepted and Rejected/Artifact Trials by Subject and Task

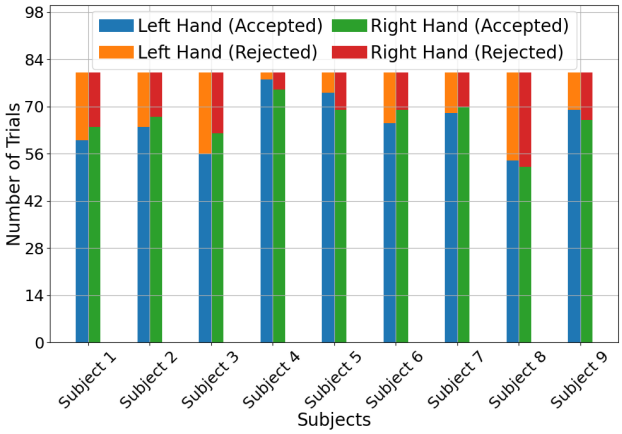


Figure 5.3: 2B (Session III): Accepted and Rejected/Artifact Trials by Subject and Task

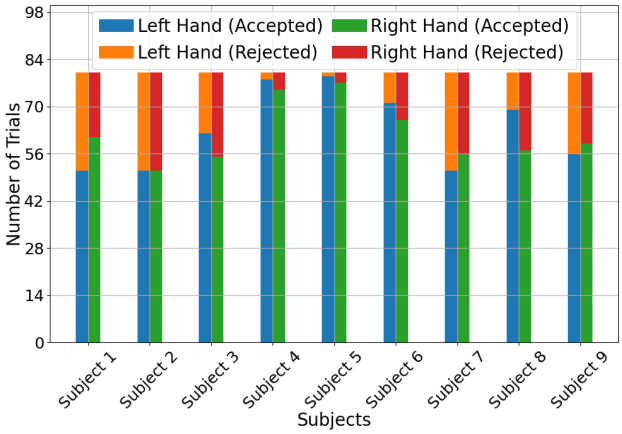


Figure 5.4: 2B (Session IV): Accepted and Rejected/Artifact Trials by Subject and Task

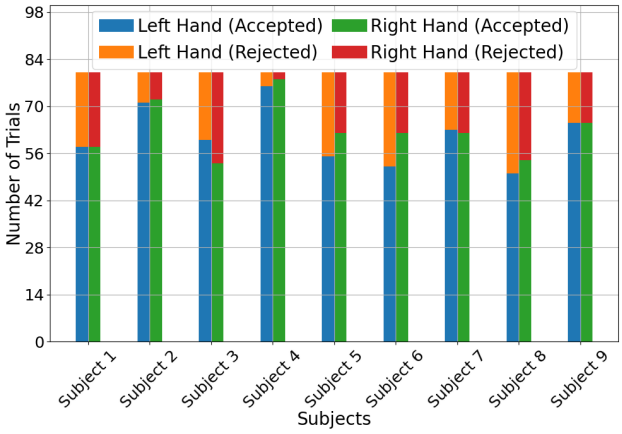


Figure 5.5: 2B (Session V): Accepted and Rejected/Artifact Trials by Subject and Task

5.2.3 State-of-the-art Methods

We evaluated our proposed method against two popular state-of-the-art methods as discussed below.

- DP-GAN [50]: To create synthetic EEG data, DP-GAN employs a GAN (consists of a generator, a discriminator) with Differential Privacy which uses DP-SGD (Differentially Private Stochastic Gradient Descent). To reduce the influence of individual data points, the model limits training by adding noise to the gradients. To improve data quality and privacy, it uses convolutional neural networks in the discriminator to learn spatio-temporal EEG patterns.

- RDP-CGAN [97]: It uses Convolutional GANs (CGANs) with Rényi Differential Privacy (RDP) for data generation. It includes Convolutional Autoencoders (CAEs) to handle discrete and continuous data, capturing temporal and feature correlations.

5.2.4 Classification Performance

The computational evaluation is broken down into three distinct scenarios: $Tr \rightarrow Sy$: $(Train_{(Tr)}, Test_{(Sy)})$, $Sy \rightarrow Tr$: $(Train_{(Sy)}, Test_{(Tr)})$, and $Aug \rightarrow Te$: $(Train_{(Aug)}, Test_{(Te)})$ [66].

- $Tr \rightarrow Sy$: $(Train_{(Tr)}, Test_{(Sy)})$: Deep learning models are first trained on Tr (real EEG train subset), and then tested on the corresponding Sy (generated synthetic EEG). This scenario provides insights into how well the model generalizes from real-world samples to synthetic ones, which is critical to understanding the effectiveness of synthetic data for inference tasks when training is done on real datasets.

- $Sy \rightarrow Tr$: $(Train_{(Sy)}, Test_{(Tr)})$: In contrast, this scenario involves training the same models using Sy data and testing them with Tr data. This test is particularly important because it shows whether the synthetic data are robust enough to be useful for

training models that can later perform well on real-world data. The model's performance in this configuration would indicate whether or not the synthetic data produced by the privacy-preserving frameworks is of a high quality suitable for real-world use.

- $Aug \rightarrow Te$: $(Train_{(Aug)}, Test_{(Te)})$: Here, the models are trained on augmented data Aug $(Tr + Sy)$. The model is then tested on Te (real EEG samples of the test subset). This scenario helps to evaluate the model's generalization ability after training on a mixed dataset. The use of augmented data makes the model robust by diversifying the training set, while testing on unseen data Te evaluates the model's ability to handle new, real-world variations.

A comprehensive understanding of the performance of models trained with synthetic, real, and enhanced data was achieved in various settings by examining the results in three evaluation scenarios. These evaluations provide valuable information on the practical applicability of Sy data in real-world situations, the efficacy of augmented data, and the overall dependability of our proposed privacy-preserving method. In all instances, the ShallowNet [95] and CapsNet [96] models were utilized for classification tasks. Their performance was compared with those of other state-of-the-art methods (DP-GAN and RDP-CGAN). To ensure a fair comparison of all techniques, we used 500 rounds ($\bar{R}$) for our method and 500 total epochs for the other models, along with an identical clipping norm $C = 0.5$, $\alpha = 10$, and the privacy parameter $\delta = 10^{-3}$.

Figure 5.6 illustrates the test accuracy of the raw data (baseline scenario, $(Train_{(Tr)}, Test_{(Te)})$), where deep learning models were trained on Tr (real EEG) and tested on Te (real EEG). Table 5.3 shows the test accuracy for the three evaluation scenarios - $(Train_{(Tr)}, Test_{(Sy)})$, $(Train_{(Sy)}, Test_{(Tr)})$, and $(Train_{(Aug)}, Test_{(Te)})$ - across different subjects and methods with $\epsilon = 3$. Our method achieves higher accuracy in all three scenarios compared to the popular methods, DP-GAN and RDP-GAN.

		$(Train_{(Tr)}, Test_{(Sy)})$		$(Train_{(Sy)}, Test_{(Tr)})$		$(Train_{(Aug)}, Test_{(Te)})$	
		ShallowNet	**CapsNet**	**ShallowNet**	**CapsNet**	**ShallowNet**	**CapsNet**
B1	Our Method	**85.54**	**86.82**	**82.47**	**83.71**	**75.11**	**77.43**
	DP-GAN	75.21	76.49	70.82	72.26	67.42	68.78
	RDP-CGAN	72.74	73.66	67.19	69.32	64.27	63.56
B2	Our Method	**80.42**	**81.88**	**77.53**	**76.61**	**59.27**	**61.79**
	DP-GAN	68.26	70.80	62.40	63.55	48.12	49.68
	RDP-CGAN	70.93	69.21	58.74	56.48	47.65	46.59
B3	Our Method	**78.71**	**79.33**	**74.85**	**73.47**	**60.92**	**61.26**
	DP-GAN	66.15	65.87	59.31	60.50	47.61	45.22
	RDP-CGAN	61.76	63.48	56.94	54.20	43.57	44.83
B4	Our Method	**91.39**	**92.84**	**90.51**	**88.72**	**92.28**	**92.63**
	DP-GAN	79.22	80.69	69.75	71.86	77.41	78.58
	RDP-CGAN	74.87	76.16	66.52	65.87	76.31	78.74
B5	Our Method	**85.61**	**84.37**	**82.82**	**81.55**	**84.29**	**82.74**
	DP-GAN	76.20	74.78	66.59	68.95	68.41	67.67
	RDP-CGAN	73.12	75.66	69.83	67.38	65.94	68.50
B6	Our Method	**80.52**	**78.84**	**77.29**	**76.76**	**71.41**	**72.68**
	DP-GAN	63.39	65.28	59.77	61.44	55.69	57.55
	RDP-CGAN	62.71	64.14	56.88	59.30	53.64	54.92
B7	Our Method	**81.43**	**83.57**	**80.89**	**80.66**	**75.31**	**75.74**
	DP-GAN	66.94	64.37	62.62	61.51	57.79	56.21
	RDP-CGAN	63.86	65.28	58.55	59.90	59.12	61.77
B8	Our Method	**80.82**	**82.41**	**76.67**	**78.53**	**81.29**	**79.75**
	DP-GAN	67.65	68.91	64.30	63.45	62.74	61.18
	RDP-CGAN	66.29	64.77	60.54	58.85	57.33	58.92
B9	Our Method	**86.57**	**87.82**	**81.41**	**83.69**	**81.28**	**80.94**
	DP-GAN	72.64	73.26	63.93	62.45	61.78	59.31
	RDP-CGAN	68.18	70.72	59.53	58.90	57.22	58.67

Table 5.3: Performance comparison of models across various scenarios using ShallowNet and CapsNet architectures. Bold values indicate the highest performance.

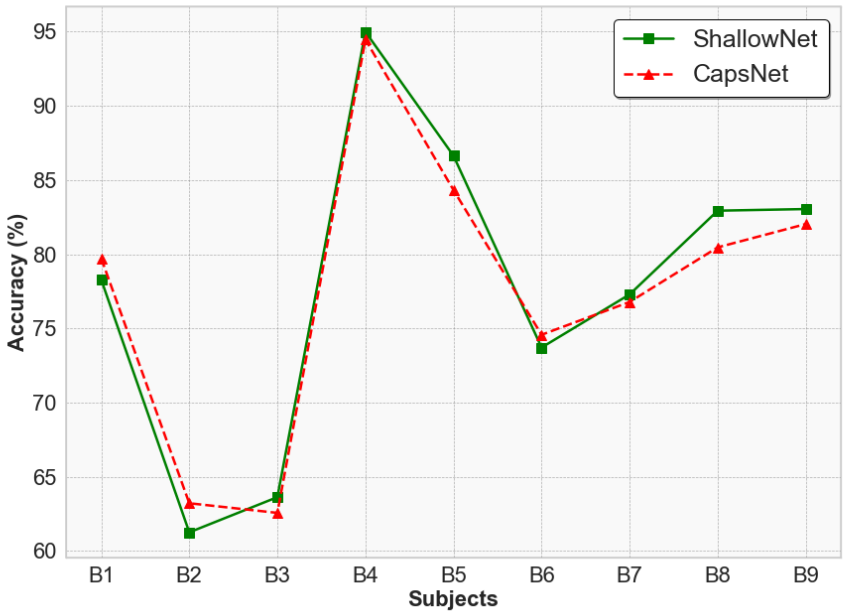


Figure 5.6: the test accuracy of the raw data (baseline scenario, $(Train_{(Tr)}, Test_{(Te)})$), where the deep learning models were trained on Tr (real EEG) and tested on Te (real EEG).

5.2.5 Full Black-box Attack

To thoroughly evaluate the privacy guarantees of our synthetic EEG data generation model, we performed a Full Black-Box Attack (FBA) similar to [98]. Here, an adversary has very restricted access, solely to the synthetic data generated by the model. The objective of the adversary is to infer whether a specific real data point X was used during the training of the model by analyzing synthetic data.

$$R = \arg\min_{\hat{X}} L(X, \hat{X})$$

where, Euclidean distance, L is the metric used to calculate the difference between the generated data point $\hat{X}$ and the real data point X and R denotes the reconstructed data point, which is the data point of the generated set that is closest to the target data point. The adversary is informed of a potential privacy violation when the calculated distance $L(X, R)$

is found to be less than a preset threshold T_κ. This is because it raises the possibility that the original data point X was included in the training dataset. The following is a mathematical expression for the attack:

$$A(X) = \begin{cases} 1 & \text{if } L(X, R) \leq T_\kappa \\ 0 & \text{otherwise} \end{cases}$$

where, $A(X)$ is a binary function that returns 1 if the distance is within the threshold T_κ, indicating a successful inference, and 0 otherwise.

Full Black-box Attack Results

The number of samples leaked during the Full Black-box Attack is measured as a part of the total dataset based on the output of the evaluation function, A(X). The results of the attack are summarized in Figure 5.7. Due to the large number of images and their file size, only one subject's results are shown as a representation of the dataset. Additional figures for other subjects are included in Appendix B. One important finding from the data is that our model shows great resilience against the FBA with few successful inferences when the ϵ is kept low, that is, below 3. In line with the anticipated trade-off between privacy and utility in differential privacy frameworks, the model becomes more vulnerable to inference attacks as ϵ increases.

For this evaluation, we set T_κ at 0.05, also observed in [98], which balances the trade-off between the sensitivity of the attack and the practical utility of the generated data. Similarly, our findings indicate that to achieve the balance between privacy and utility of the model, ϵ should be maintained within a range. Too low ϵ values increase privacy but reduce utility, and too high values increase susceptibility to attacks, as shown in Figure 5.7.

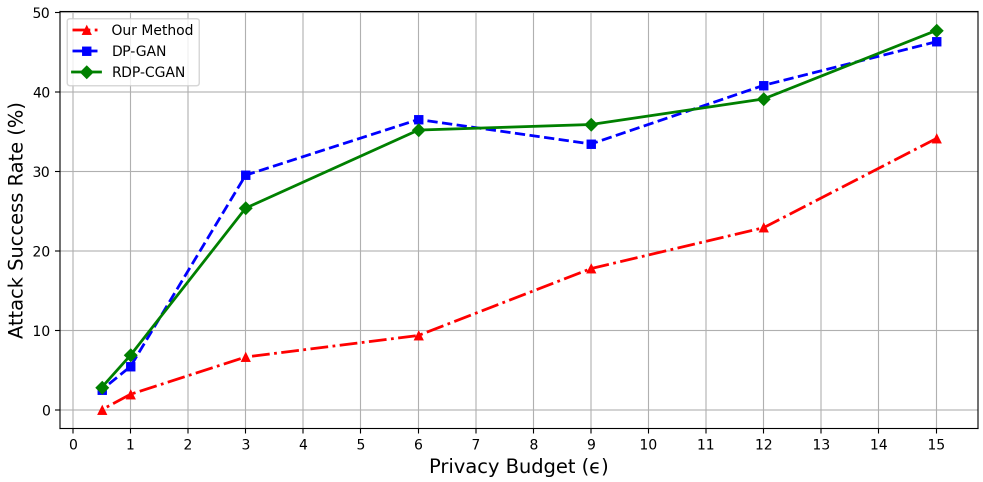


Figure 5.7: Attack success rate (%) for subject B1 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

5.3 Summary

In this work, we present a framework based on a federated learning approach for the generation of synthetic EEG data. It combines Hierarchical Privacy Adaptive Autoencoders, Secure Aggregation, and Generative Flow Networks (GFlowNet) enhanced by Rényi Differential Privacy (RDP) to achieve data utility and strong privacy. The hierarchical architecture of the autoencoders allows for the efficient extraction of multi-level spatial and temporal characteristics from EEG signals, essential for preserving the quality of the generated synthetic EEG data. Our approach tackles the issue of safeguarding sensitive brain signals while producing high-fidelity synthetic data.

Through Federated Split Learning (FSL), we decouple the learning process into hierarchical feature extraction on the client side and data generation on the server side. This reduces computational resource demand on the client side and ensures that raw EEG data never leave the client's device. The use of adaptive autoencoders and RDP further strengthens privacy by dynamically applying noise based on the sensitivity of the data. The Secure Aggregation mechanism ensures that individual client contributions remain private even during commu-

nication to the server. Our results demonstrate the effectiveness of the proposed method, which offers a balance between privacy and utility measured with varying privacy budgets. This makes it suitable for real-world applications where privacy is paramount, such as in medical diagnostics, brain-computer interfaces, and other EEG-based systems. Future work could explore the extension of this framework to other types of physiological data and the potential for real-time privacy-preserving analytics in distributed environments.

6

Conclusions

The problem of protecting personal data is becoming more and more pressing in a world where AI systems are rapidly developing. Despite their strength, deep learning models carry the risk of disclosing personal data if not properly handled. To navigate this challenge, this thesis presents three substantial frameworks that deal with model and data privacy. These approaches not only protect user personal information, but also promote the use of diverse datasets, ultimately leading to stronger model performance and trust.

In this thesis, in the first part, we developed a Generative Adversarial Network (GAN) enhanced with quantum-inspired differential privacy techniques to secure EEG data for Brain Machine Interface applications. The GAN architecture integrates differential privacy within the training process, applying dynamically adjusted Gaussian noise based on the principles of quantum decoherence and uncertainty to the discriminator's gradients. Our first method ensures a balance between the data utility and privacy by managing the privacy budget across

training epochs using a custom stochastic gradient descent. The discriminator, equipped with convolutional and bidirectional LSTM layers, validates the authenticity of the synthetic EEG data generated, while the generator is trained to produce high-utility data under privacy restrictions. The effectiveness of the model is validated by rigorous testing against standard privacy attacks (reconstruction attacks and membership inference attacks), demonstrating its ability to protect sensitive EEG data while maintaining its utility for biomedical research.

In the second part of this thesis, we designed a privacy-preserving synthetic EEG data generation framework that integrates a Spiking Neural Network (SNN)-based generator with an Artificial Neural Network (ANN)-based discriminator deployed within a federated learning environment. The SNN-based generator, modeled using a modified Leaky Integrate-and-Fire (LIF) neuron, simulates the temporal dynamics of biological neurons to produce realistic synthetic EEG signals. The ANN-based discriminator classifies the signals generated as real or artificial through a series of FC layers. Temporally correlated noise is added throughout the data synthesis process to protect privacy by preventing the synthetic data from being connected to the original samples. This noise is applied dynamically across time steps, preserving the temporal structure of the EEG signals. By aggregating local models to a central server, federated learning ensures privacy without centralizing sensitive data while the model is taught across numerous clients. Renyi Differential Privacy (RDP) is employed to measure and enforce privacy guarantees throughout the training process. The results of this method demonstrate that it can provide strong privacy guarantees and efficiently captures the temporal dynamics of EEG signals. This method maintains secrecy while guaranteeing that the synthetic data closely mimic the real data.

Our third approach integrates Federated Split Learning (FSL), Hierarchical Privacy-Adaptive Autoencoders, Secure Aggregation, and Generative Flow Networks (GFlowNet) with RDP to generate privacy-preserving synthetic EEG data. In FSL, clients process raw EEG data locally through a hierarchical autoencoder that extracts spatial and temporal features. The server receives anonymized latent representations, from which GFlowNet creates

artificial EEG signals. RDP adds noise to preserve privacy depending on how sensitive the data is, and Secure Aggregation encrypts the data while it is being transmitted. Our results demonstrate an optimal balance of privacy and utility, preserving the spatio-temporal structure of EEG data in various privacy budgets, making the approach suitable for sensitive applications such as healthcare and neuroscience.

The three distinct privacy-preserving frameworks were explored and compared on the basis of their balance between privacy guarantees and data utility. Chapter 3 offers the strongest privacy guarantees, but at the cost of slightly reduced data utility due to the dynamic quantum-inspired noise addition. Chapter 4 allowed better data utility by keeping raw EEG data decentralized at the client level while applying Rényi Differential Privacy locally to maintain privacy. By distributing the privacy budget across multiple latent spaces, Chapter 5 achieved the best trade-off between privacy and data quality, ensuring that the generated EEG data retained both spatial and temporal coherence.

Our rigorous experiments have verified that these approaches deliver an ideal trade-off between utility and privacy, positioning them as practical solutions for real-world BCI or BMI applications. Optimizing these frameworks for even more versatility and effectiveness in a range of use situations will be the focus of my future research.

7

Future Work

Evaluating the model's scalability in real-world scenarios is essential, despite its development in this study with two seminal EEG-BCI datasets. A user-friendly framework or toolkit based on Q-DP-GAN can improve its adoption among researchers and practitioners who lack expertise in privacy-preserving technologies. Although our focus has been on BCI applications based on EEG data, applying Q-DP-GAN to finance, healthcare, and social media is promising. These fields can benefit significantly from the privacy-preserving capabilities of the model.

Moreover, I aim to integrate federated learning frameworks with generative models to enable decentralized data generation while ensuring privacy. This approach has the potential to expand the applicability of synthetic data in privacy-sensitive domains, including healthcare and finance, where protecting data confidentiality is paramount. I am also interested in applying these techniques to real-time data generation, which could substantially improve

predictive modeling accuracy and decision-making processes in these critical fields.

In addition, I intend to improve the validation of the suggested models using a series of assessment methods that emphasize the resilience, precision, and utility of the generated data. To be more precise, I will assess how effectively the models capture dependencies across different data dimensions using Dimension-wise prediction (DWpre) [78]. This will enhance our understanding of inter-dimensional relationships. Evaluate the overall quality of the newly generated data using the Generate Score [77] to make sure that it satisfies strict accuracy requirements under differential privacy conditions. I will use dimension-wise statistics and dimension-wise average (DWA) to further verify the statistical integrity of synthetic data [98]. This will provide a thorough comparison of the statistical characteristics and averages in each dimension of the created data with the original data. Correlation analysis [98] can be used to verify that the resulting data preserve these dimensional relationships, guaranteeing their utility. In order to ensure that the statistical distributions of the synthetic and real data are almost identical, the maximum mean difference (MMD) [123, 97] can provide comparisons. Lastly, an ablation study similar to [97], systematically altering model elements such as autoencoders and convolutional layers, will help to evaluate their respective contributions to overall performance.

Another direction for the future is to expand my research to incorporate blockchain technologies, exploring their potential to enhance the security and traceability of synthetic data transactions. This integration could offer a robust solution to ensure data integrity and compliance, particularly in regulated industries such as healthcare. Furthermore, I intend to explore privacy-preserving techniques like homomorphic encryption and secure multiparty computation to strengthen the privacy guarantees of my models.

Lastly, the scope of my research can extend to include other types of neural data, such as fMRI and MEG, and integrate multimodal data. This will allow for more comprehensive neural data synthesis, potentially leading to breakthroughs in areas like neuroimaging and personalized medicine. Finally, I plan to make these advanced techniques more accessible by

developing user-friendly frameworks, thereby enabling a broader range of practitioners and researchers to leverage privacy-preserving synthetic data generation in their work.

Publications

The work presented in this thesis is being submitted for publication in IEEE, ACM, Elsevier, and Springer journals with the following titles.

- Paul, S., Bajwa, G.: Improving EEG data privacy through quantum-inspired differential privacy-based GAN.
- Paul, S., Bajwa, G.: Enhancing thought privacy using federated learning with spiking GANs for high-fidelity EEG data generation.
- Paul, S., Bajwa, G.: Privacy-preserving EEG data generation: A federated split learning approach using privacy-adaptive autoencoders and secure aggregation with GFlowNet.

Bibliography

[1] P. Chaudhary and R. Agrawal, "Emerging threats to security and privacy in brain computer interface," *International Journal of Advanced Studies of Scientific Research*, vol. 3, no. 12, 2018.

[2] O. Landau, R. Puzis, and N. Nissim, "Mind your mind: Eeg-based brain-computer interfaces and their security in cyber space," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–38, 2020.

[3] G. Pfurtscheller, D. Flotzinger, and J. Kalcher, "Brain-computer interface—a new communication device for handicapped persons," *Journal of microcomputer applications*, vol. 16, no. 3, pp. 293–299, 1993.

[4] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan, *et al.*, "Brain-computer interface technology: a review of the first international meeting," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 2, pp. 164–173, 2000.

[5] S. G. Mason and G. E. Birch, "A general framework for brain-computer interface design," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 11, no. 1, pp. 70–85, 2003.

[6] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, no. 6725, pp. 297–298, 1999.

[7] R. Janapati, V. Dalal, and R. Sengupta, "Advances in modern eeg-bci signal processing: A review," *Materials Today: Proceedings*, vol. 80, pp. 2563–2566, 2021.

[8] E. P. Torres, E. A. Torres, M. Hernández-Álvarez, and S. G. Yoo, "Eeg-based bci emotion recognition: A survey," *Sensors*, vol. 20, no. 18, p. 5083, 2020.

[9] K. Douibi *et al.*, "Toward eeg-based bci applications for industry 4.0: Challenges and possible applications," *Frontiers in Human Neuroscience*, vol. 15, p. 705064, 2021.

[10] J. Li *et al.*, "Meta-learning for fast and privacy-preserving source knowledge transfer of eeg-based bcis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 123–134, 2023.

[11] X. Brocol *et al.*, "Brain-computer interfaces in safety and security fields: Risks and applications," *Journal of Neuroscience*, vol. 40, pp. 123–135, 2021.

[12] E. T. Martínez Beltrán, M. Quiles Pérez, S. López Bernal, A. Huertas Celdran, and G. Martínez Pérez, "Noise-based cyberattacks generating fake p300 waves in brain–computer interfaces," *Cluster Computing*, vol. 25, no. 1, pp. 33–48, 2022.

[13] S. L. Bernal, A. H. Celdran, L. F. Maimo, M. T. Barros, S. Balasubramaniam, and G. M. Perez, "Cyberattacks on miniature brain implants to disrupt spontaneous neural signaling," *IEEE Access*, vol. 8, pp. 152204–152222, 2020.

[14] Y. Xia *et al.*, "Privacy-preserving brain-computer interfaces: A systematic review," *Journal of Biomedical Informatics*, vol. 136, p. 104172, 2023.

[15] T. Varbu *et al.*, "Past, present, and future of eeg-based bci applications," *Journal of Neural Engineering*, vol. 19, p. 045003, 2022.

[16] K. Xia, W. Duch, Y. Sun, K. Xu, W. Fang, H. Luo, Y. Zhang, D. Sang, X. Xu, F.-Y. Wang, *et al.*, "Privacy-preserving brain–computer interfaces: A systematic review," *IEEE Transactions on Computational Social Systems*, 2022.

[17] D. Popescu, R. Voicu, and I. Bichindaritz, "Privacy-preserving classification of eeg data using machine learning and homomorphic encryption," *IEEE Access*, vol. 9, pp. 25979–25989, 2021.

[18] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer, and K. Muralidhar, "A critical review on the use (and misuse) of differential privacy in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–16, 2022.

[19] B. Amira, "Evaluating differential privacy in machine learning models: Methods, applications, and challenges," *International Journal of Intelligent Automation and Computing*, vol. 7, pp. 11–20, May 2024.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[21] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, pp. 5767–5777, 2017.

[24] F. Faisal, N. Mohammed, C. K. Leung, and Y. Wang, "Generating privacy preserving synthetic medical data," in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, IEEE, 2022.

[25] S. Bhatia and R. Dahyot, "Using wgan for improving imbalanced classification performance," *Proceedings of the International Conference on Machine Learning*, 2019.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[27] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, IEEE, 2010.

[28] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Journal of Privacy and Confidentiality*, vol. 7, no. 3, pp. 17–34, 2006.

[29] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, IEEE, 2017.

[31] T. Wang, X. Zhang, J. Feng, and X. Yang, "A comprehensive survey on local differential privacy toward data statistics and analysis," *Sensors*, vol. 20, no. 24, p. 7030, 2020.

[32] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy, data processing inequalities, and statistical minimax rates," *arXiv preprint arXiv:1302.3203*, 2013.

[33] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proceedings of the 2018 International Conference on Management of Data*, pp. 1655–1658, 2018.

[34] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 425–438, 2017.

[35] R. L. Amoroso, *Universal quantum computing: supervening decoherence-surmounting uncertainty*. World Scientific, 2017.

[36] L. Bi, K. Liang, G. Czap, H. Wang, K. Yang, and S. Li, "Recent progress in probing atomic and molecular quantum coherence with scanning tunneling microscopy," *Progress in Surface Science*, vol. 98, no. 1, p. 100696, 2023.

[37] I. Abdikhakimov, "The uncertainty principle: How quantum mechanics is transforming jurisprudence," *International Journal of Cyber Law*, vol. 1, no. 7, 2023.

[38] P. Busch, P. Lahti, and R. F. Werner, "Colloquium: Quantum root-mean-square error and measurement uncertainty relations," *Reviews of Modern Physics*, vol. 86, no. 4, pp. 1261–1281, 2014.

[39] M. Moore and A. Narayanan, "Quantum-inspired computing," *Dept. Comput. Sci., Univ. Exeter, Exeter, UK*, 1995.

[40] M. Schlosshauer, "Quantum decoherence," *Physics Reports*, vol. 831, pp. 1–57, 2019.

[41] D. Manzano, "A short introduction to the lindblad master equation," *Aip advances*, vol. 10, no. 2, 2020.

[42] G. Jaeger, *Quantum information*. Springer, 2007.

[43] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[44] X. Yao, T. Huang, R.-X. Zhang, R. Li, and L. Sun, "Federated learning with unbiased gradient aggregation and controllable meta updating," *arXiv preprint arXiv:1910.08234*, 2019.

[45] Y. Xu, W. Ma, C. Dai, Y. Wu, and H. Zhou, "Generalized federated learning via gradient norm-aware minimization and control variables," *Mathematics*, vol. 12, no. 17, p. 2644, 2024.

[46] K. Liu, H. Yu, J. Hu, and C. Wang, "Privacy-preserving traffic flow prediction: A federated learning approach," *arXiv preprint arXiv:2006.05592*, 2020.

[47] P. R. M. P. S. H. Y. D. A. Mothukuri, Venkata and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[48] A. Wood, K. Najarian, and D. Kahrobaei, "Homomorphic encryption for machine learning in medicine and bioinformatics," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[49] C. Dwork, "Differential privacy. automata, languages and programming," in *33rd International Colloquium, ICALP*, 2006.

[50] E. Debie, N. Moustafa, and M. T. Whitty, "A privacy-preserving generative adversarial network method for securing eeg brain signals," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[51] S. Alshebli, M. Alshehhi, and C. Y. Yeun, "Investigating how data poising attacks can impact an eeg-based federated learning model," in *2024 2nd International Conference on Cyber Resilience (ICCR)*, pp. 1–6, IEEE, 2024.

[52] K. Wang, M. Yang, C. Li, A. Liu, R. Qian, and X. Chen, "Privacy-preserving domain adaptation for intracranial eeg classification via information maximization and

gaussian mixture model," *IEEE Sensors Journal*, vol. 23, no. 21, pp. 26390–26400, 2023.

[53] A. Agarwal, R. Dowsley, N. D. McKinney, D. Wu, C.-T. Lin, M. De Cock, and A. C. Nascimento, "Protecting privacy of users in brain-computer interface applications," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 8, pp. 1546–1555, 2019.

[54] A. Agarwal, R. Dowsley, N. D. McKinney, D. Wu, C.-T. Lin, M. De Cock, and A. Nascimento, "Privacy-preserving linear regression for brain-computer interface applications," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5277–5278, IEEE, 2018.

[55] C. Hanisch, B. Barth, and D. Kuhn, "Privacy-preserving data analytics for eeg signals," *Journal of Neuroscience Methods*, vol. 352, p. 109027, 2021.

[56] F. Schiliro, N. Moustafa, and A. Beheshti, "Cognitive privacy: Ai-enabled privacy using eeg signals in the internet of things," in *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, pp. 73–79, IEEE, 2020.

[57] S. Pazouki, N. A. Golilarz, S. M. Kazemi-Razi, and A. Aydeger, "A self-healing cybersecurity mechanism for cyberattacks targeting artificial neural network-based human brain implants controlling smart homes," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6, IEEE, 2022.

[58] A. J. Bidgoly, H. J. Bidgoly, and Z. Arezoumand, "Towards a universal and privacy preserving eeg-based authentication system," *Scientific Reports*, vol. 12, no. 1, pp. 1–12, 2022.

[59] Q. Gui, W. Yang, Z. Jin, M. V. Ruiz-Blondet, and S. Laszlo, "A residual feature-based replay attack detection approach for brainprint biometric systems," in *2016*

IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6, IEEE, 2016.

[60] G. Mezzina, V. F. Annese, and D. De Venuto, "A cybersecure p300-based brain-to-computer interface against noise-based and fake p300 cyberattacks," *Sensors*, vol. 21, no. 24, p. 8280, 2021.

[61] E. Maiorana, G. E. Hine, D. La Rocca, and P. Campisi, "On the vulnerability of an eeg-based biometric system to hill-climbing attacks algorithms' comparison and possible countermeasures," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, IEEE, 2013.

[62] M. Wang, S. Wang, and J. Hu, "Polycosgraph: A privacy-preserving cancelable eeg biometric system," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[63] J. Yan, F. Liu, Y. Xiao, and L. Yang, "Eeg classification with spiking neural network: Smaller, better, more energy efficient," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1306–1310, IEEE, 2022.

[64] Y. Xu, M. Zhang, X. Wang, and Y. Luo, "Eescn: A novel spiking neural network method for eeg-based emotion recognition," in *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 2147–2154, IEEE, 2024.

[65] J. Jordon, J. Yoon, and M. van der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *2019 International Conference on Learning Representations (ICLR)*, pp. 1–10, ICLR, 2019.

[66] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017.

[67] J. Yoon, D. Jarrett, and M. van der Schaar, "Timegan: Time-series generative adversarial networks," in *NeurIPS*, pp. 1–10, NeurIPS, 2019.

[68] D. Bäßler, T. Kortus, and G. Gühring, "Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks," *Machine Learning*, vol. 111, no. 4, pp. 1377–1408, 2022.

[69] A. Khan, I. Aziz, and A. Bhatti, "A privacy and energy-aware federated framework for human activity recognition," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 458–463, IEEE, 2023.

[70] M. Nikfam, H. Liu, and J. Zhang, "A homomorphic encryption framework for privacy-preserving spiking neural networks," in *2023 International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 678–687, IEEE, 2023.

[71] M. Arsalan, D. Di Matteo, S. Imtiaz, Z. Abbas, V. Vlassov, and V. Issakov, "Energy-efficient privacy-preserving time-series forecasting on user health data streams," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 541–548, IEEE, 2022.

[72] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," in *Constructive Machine Learning Workshop (NIPS 2016)*, pp. 1–6, NIPS, 2016.

[73] J.-P. Rosenfeld, M. U. Ahmed, and T. Durrani, "Spiking generative adversarial networks with a neural network discriminator: Local training, bayesian models, and continual meta-learning," in *2022 International Conference on Artificial Neural Networks (ICANN)*, pp. 1–10, IEEE, 2022.

[74] J. Shen, K. Wang, W. Gao, J. Liu, Q. Xu, G. Pan, X. Chen, and H. Tang, "Temporal spiking generative adversarial networks for heading direction decoding," *Available at SSRN 4757430*.

[75] J. Wang and Y. Zhao, "Dpsnn: A differentially private spiking neural network," in *2022 International Conference on Machine Learning (ICML)*, pp. 3147–3156, IEEE, 2022.

[76] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau, "Aim: An adaptive and iterative mechanism for differentially private synthetic data," *arXiv preprint arXiv:2201.12677*, 2022.

[77] Y. Liu, J. Peng, J. J. Yu, and Y. Wu, "Ppgan: Privacy-preserving generative adversarial network," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1–8, IEEE, 2019.

[78] L. Xie, "Differentially private generative adversarial network," in *NIPS*, 2018.

[79] M. Shateri, F. Messina, F. Labeau, and P. Piantanida, "Preserving privacy in gans against membership inference attack," *IEEE Transactions on Information Forensics and Security*, 2023.

[80] D. Huang, M. Wang, and J. Wang, "A survey of quantum computing hybrid applications with brain-computer interface," *Cognitive Robotics*, vol. 2, pp. 164–176, 2022.

[81] A. Ullah, S. Ullah, and A. Rafiq, "Quantum machine learning revolution in healthcare: A systematic review of emerging perspectives and applications," *Journal of Healthcare Engineering*, vol. 2023, pp. 1–15, 2023.

[82] D. Singh, Y. Kanathey, Y. Waykole, R. K. Mishra, R. Walambe, K. H. Aqeel, and K. Kotecha, "Exploring the usability of quantum machine learning for eeg signal classification," in *International Advanced Computing Conference*, pp. 427–438, Springer, 2023.

[83] G. Tasci, M. V. Gun, T. Keles, B. Tasci, P. D. Barua, I. Tasci, S. Dogan, M. Baygin, E. E. Palmer, T. Tuncer, *et al.*, "Qlbp: Dynamic patterns-based feature extraction

functions for automatic detection of mental health and cognitive conditions using eeg signals," *Chaos, Solitons & Fractals*, vol. 172, p. 113472, 2023.

[84] J. H. Kim, Y. Cho, Y.-A. Suh, and M.-S. Yim, "Development of an information security-enforced eeg-based nuclear operators' fitness for duty classification system," *Ieee Access*, vol. 9, pp. 72535–72546, 2021.

[85] H. Liao, X. Zhang, and X. Zhou, "Exploring the intersection of brain-computer interfaces and quantum sensing: A review of opportunities and challenges," *IEEE Sensors Journal*, vol. 23, no. 5, pp. 1045–1058, 2023.

[86] M. Sangeetha, P. Senthil, A. H. Alshehri, S. Qamar, H. Elshafie, and V. P. Kavitha, "Neuro quantum computing based optoelectronic artificial intelligence in electroencephalogram signal analysis," *Optical and Quantum Electronics*, vol. 56, no. 544, pp. 1–18, 2024.

[87] D. K. Saha, V. D. Calhoun, Y. Du, Z. Fu, S. M. Kwon, A. D. Sarwate, S. R. Panta, and S. M. Plis, "Privacy-preserving quality control of neuroimaging datasets in federated environments," *Human Brain Mapping*, vol. 43, no. 7, pp. 2289–2310, 2022.

[88] P. Wang, Y. Lei, Y. Ying, and H. Zhang, "Differentially private sgd with non-smooth losses," *Applied and Computational Harmonic Analysis*, vol. 56, pp. 306–336, 2022.

[89] D. Wang and J. Xu, "Differentially private high dimensional sparse covariance matrix estimation," *Theoretical Computer Science*, vol. 865, pp. 119–130, 2021.

[90] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set a," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, pp. 1–6, 2008.

[91] R. Leeb, C. Brunner, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set b," *Graz University of Technology, Austria*, vol. 16, pp. 1–6, 2008.

[92] S. Hu, H. Wang, J. Zhang, W. Kong, and Y. Cao, "Causality from cz to c3/c4 or between c3 and c4 revealed by granger causality and new causality during motor imagery," in *2014 International joint conference on neural networks (IJCNN)*, pp. 3178–3185, IEEE, 2014.

[93] H. Altaheri, G. Muhammad, and M. Alsulaiman, "Physics-informed attention temporal convolutional network for eeg-based motor imagery classification," *IEEE transactions on industrial informatics*, vol. 19, no. 2, pp. 2249–2258, 2022.

[94] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[95] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[96] K.-W. Ha and J.-W. Jeong, "Motor imagery eeg classification using capsule networks," *Sensors*, vol. 19, no. 13, p. 2854, 2019.

[97] A. Torfi, E. A. Fox, and C. K. Reddy, "Differentially private synthetic medical data generation using convolutional gans," *Information Sciences*, vol. 586, pp. 485–500, 2022.

[98] H. Gwon, I. Ahn, Y. Kim, H. J. Kang, H. Seo, H. Choi, H. N. Cho, M. Kim, J. Han, G. Kee, *et al.*, "Ldp-gan: Generative adversarial networks with local differential privacy

for patient medical records synthesis," *Computers in Biology and Medicine*, vol. 168, p. 107738, 2024.

[99] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," *arXiv preprint arXiv:1705.07663*, 2017.

[100] Z. Li, M. Yang, Y. Liu, J. Wang, H. Hu, W. Yi, and X. Xu, "Gan you see me? enhanced data reconstruction attacks against split inference," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[101] H. Gao, J. He, H. Wang, T. Wang, Z. Zhong, J. Yu, Y. Wang, M. Tian, and C. Shi, "High-accuracy deep ann-to-snn conversion using quantization-aware training framework and calcium-gated bipolar leaky integrate and fire neuron," *Frontiers in Neuroscience*, vol. 17, p. 1141701, 2023.

[102] S. Singanamalla, S. R. Rajasekaran, K. Dinesh, and V. Narayanan, "Spiking neural network for augmenting electroencephalographic data for brain-computer interfaces," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5005–5009, IEEE, 2022.

[103] W. Nicola and C. Clopath, "Supervised learning in spiking neural networks with force training," *Nature communications*, vol. 8, no. 1, p. 2208, 2017.

[104] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[105] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE access*, vol. 8, pp. 4806–4813, 2019.

[106] S. Li, V. Dutta, X. He, and T. Matsumaru, "Deep learning based one-class detection system for fake faces generated by gan network," *Sensors*, vol. 22, no. 20, p. 7767, 2022.

[107] A. Tabassum, A. Erbad, W. Lebda, A. Mohamed, and M. Guizani, "Fedgan-ids: Privacy-preserving ids using gan and federated learning," *Computer Communications*, vol. 192, pp. 299–310, 2022.

[108] H. Arnout, J. Bronner, and T. Runkler, "Clare-gan: Generation of class-specific time series," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2021.

[109] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[110] H. Li, "Time works well: Dynamic time warping based on time weighting for time series data mining," *Information Sciences*, vol. 547, pp. 592–608, 2021.

[111] K. Gupta, D. Thomas, S. Vidya, K. V. Venkatesh, and S. Ramakumar, "Detailed protein sequence alignment based on spectral similarity score (sss)," *BMC bioinformatics*, vol. 6, pp. 1–16, 2005.

[112] D. Geng and Z. S. Chen, "Auxiliary classifier generative adversarial network for interictal epileptiform discharge modeling and eeg data augmentation," in *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 1130–1133, IEEE, 2021.

[113] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[114] Z. Zhang, A. Pinto, V. Turina, F. Esposito, and I. Matta, "Privacy and efficiency of communications in federated split learning," *IEEE Transactions on Big Data*, vol. 9, no. 5, pp. 1380–1391, 2023.

[115] G. Cisotto, A. Zancanaro, I. Zoppis, S. Manzoni, *et al.*, "hveegnet: exploiting hierarchical vaes on eeg data for neuroscience applications," 2023.

[116] S. Lahlou, T. Deleu, P. Lemos, D. Zhang, A. Volokhova, A. Hernández-Garcıa, L. N. Ezzine, Y. Bengio, and N. Malkin, "A theory of continuous generative flow networks," in *International Conference on Machine Learning*, pp. 18269–18300, PMLR, 2023.

[117] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame, *et al.*, "Safelearn: Secure aggregation for private federated learning," in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 56–62, IEEE, 2021.

[118] Z. Zhang, J. Li, S. Yu, and C. Makaya, "Safelearning: Enable backdoor detectability in federated learning with secure aggregation," *arXiv preprint arXiv:2102.02402*, 2021.

[119] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[120] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4, pp. IV–317, IEEE, 2007.

[121] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1017–1024, 2011.

[122] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[123] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," *arXiv preprint arXiv:1806.07755*, 2018.

Appendix A: Supplementary Figures of Chapter 4

Evaluation Scenarios:

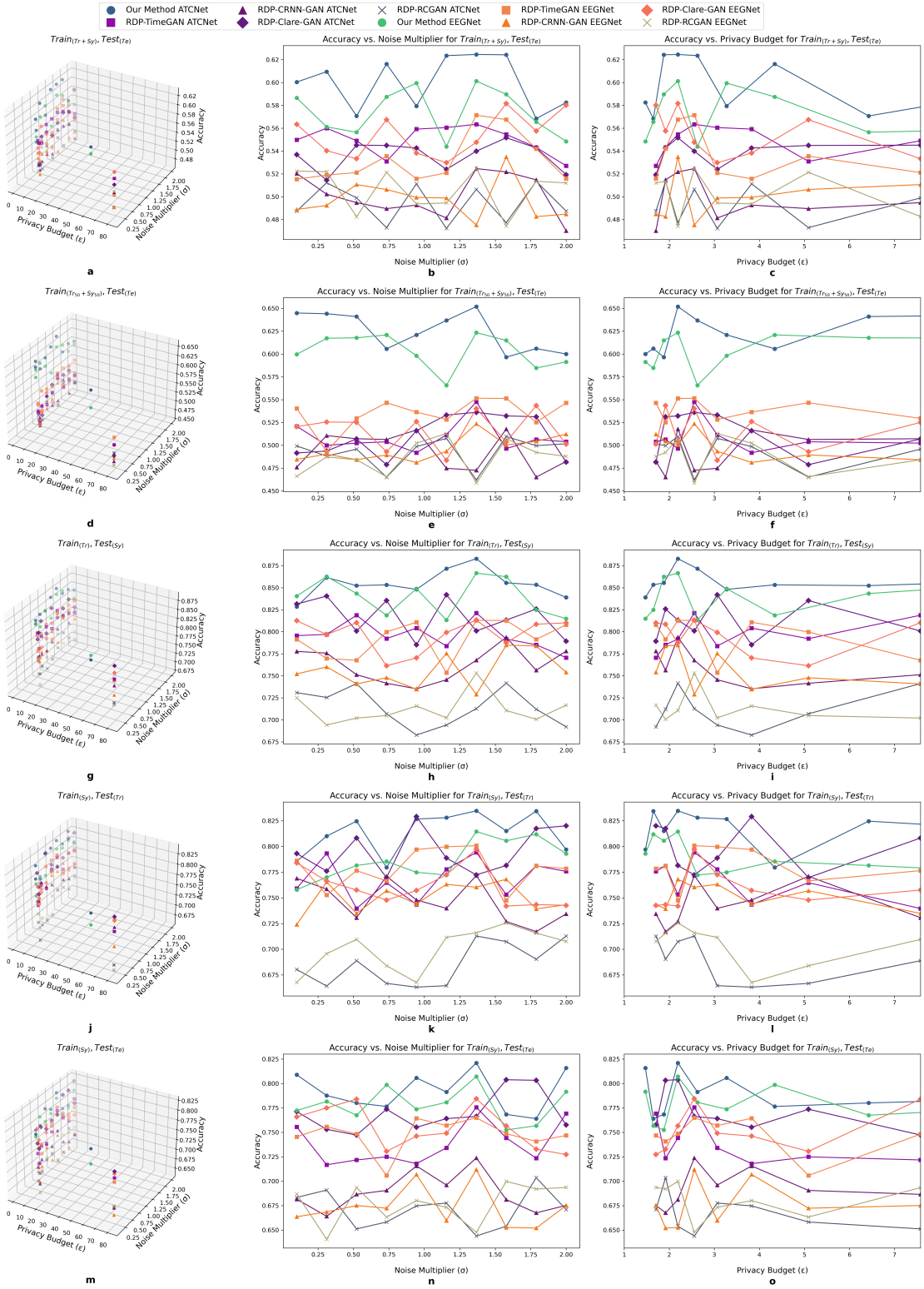
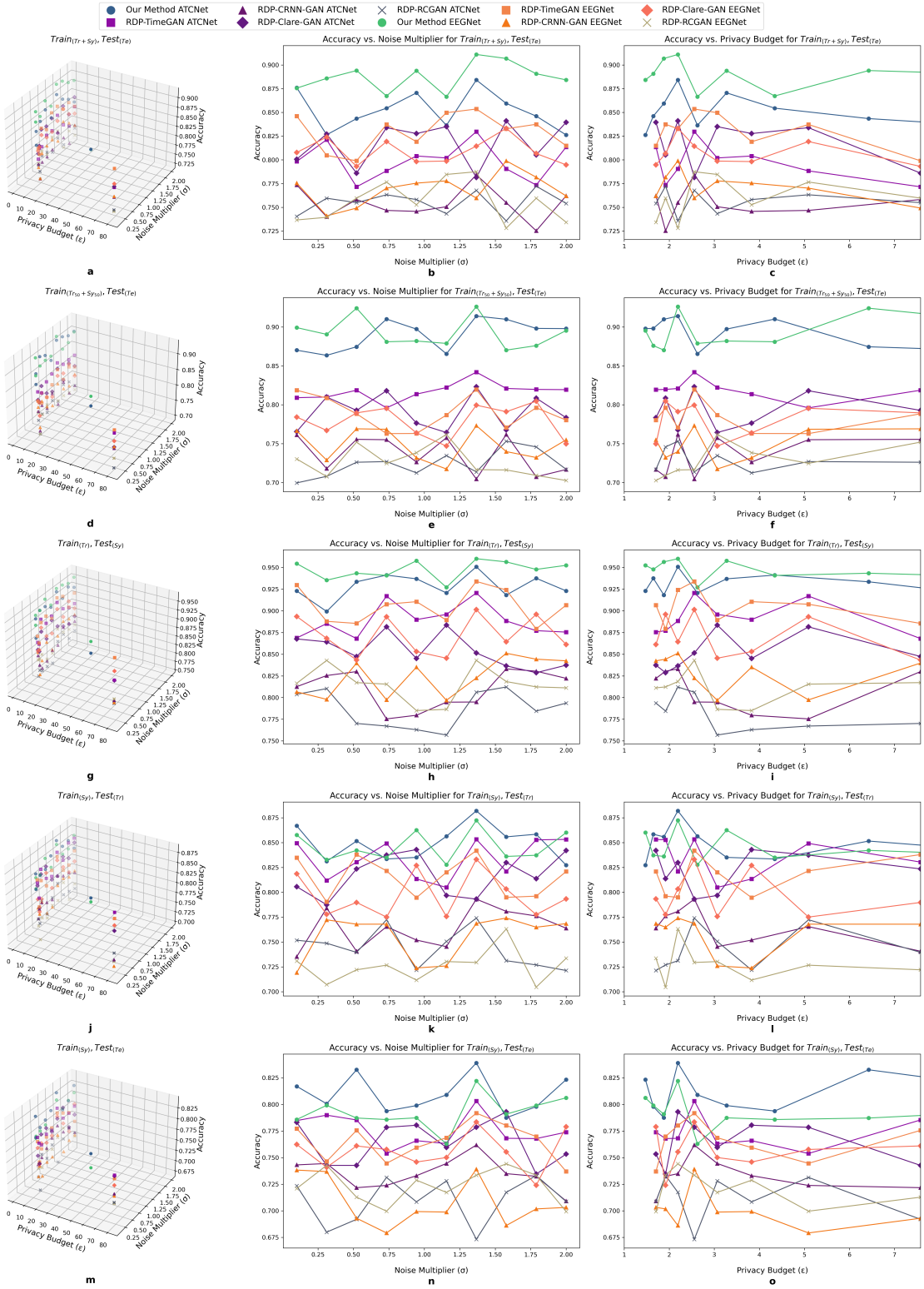
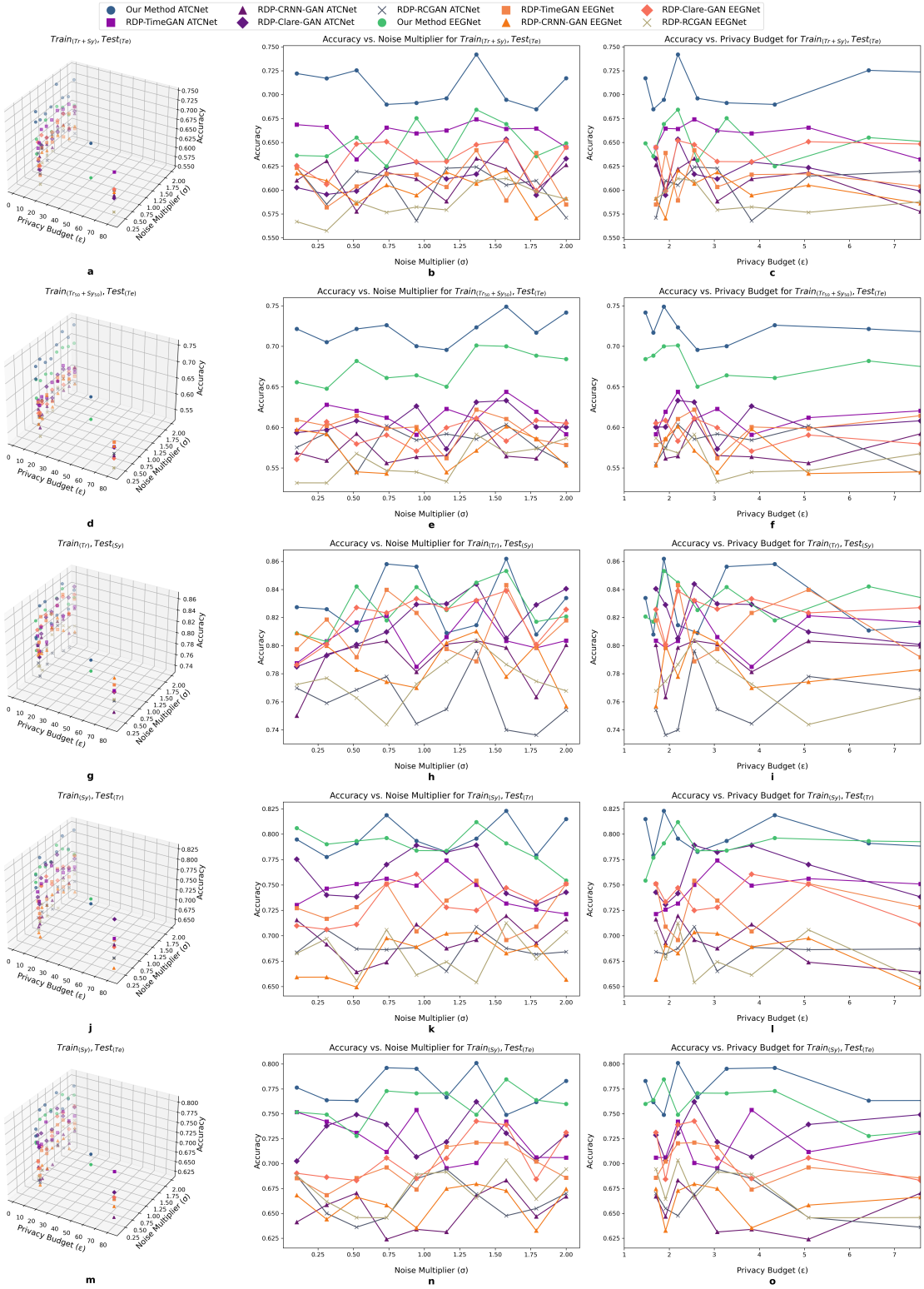


Figure 1: Comparison of different models under various privacy budgets and noise multipliers for subject A2.



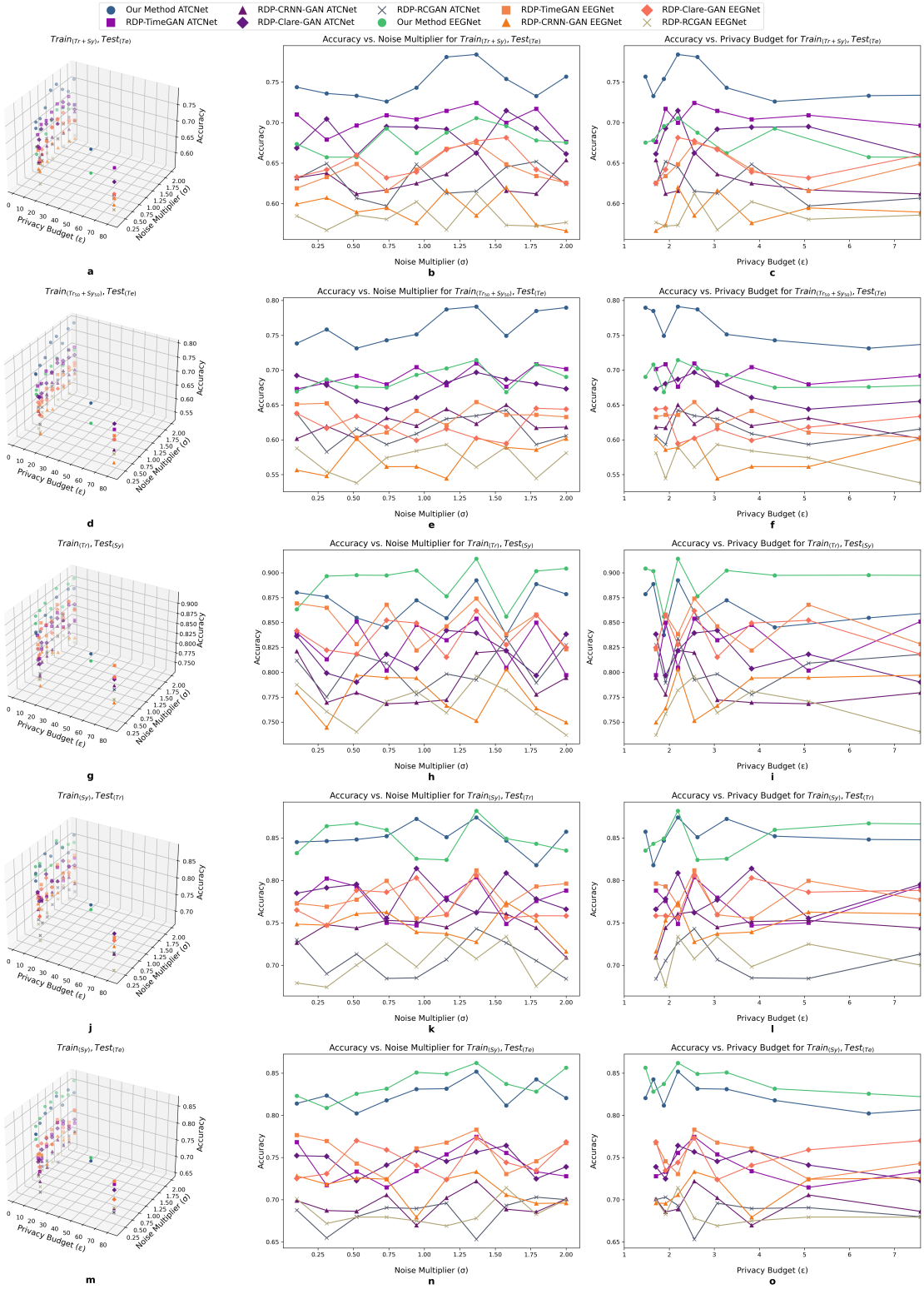
*

Figure 2: Comparison of different models under various privacy budgets and noise multipliers for subject A3.



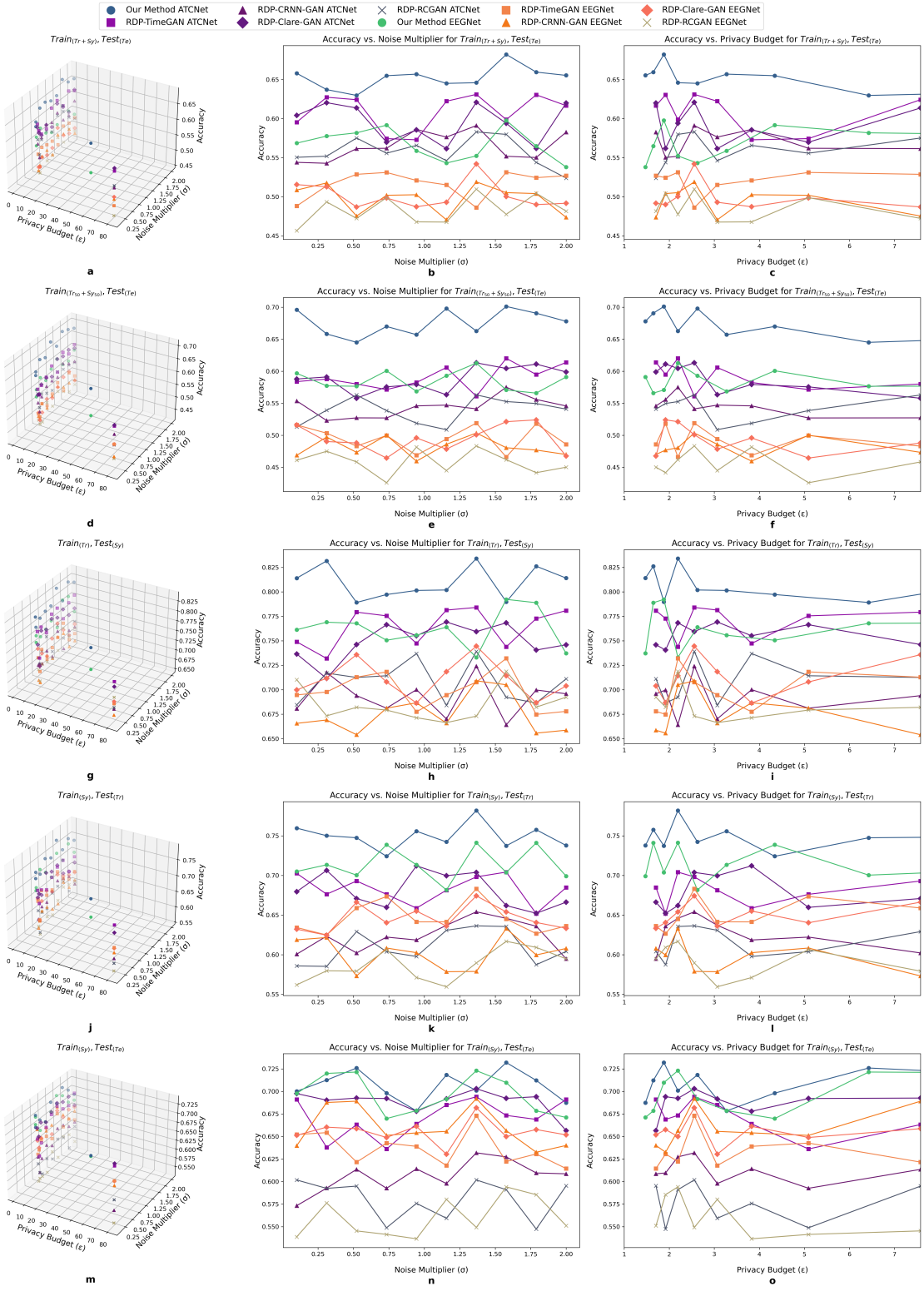
*

Figure 3: Comparison of different models under various privacy budgets and noise multipliers for subject A4.



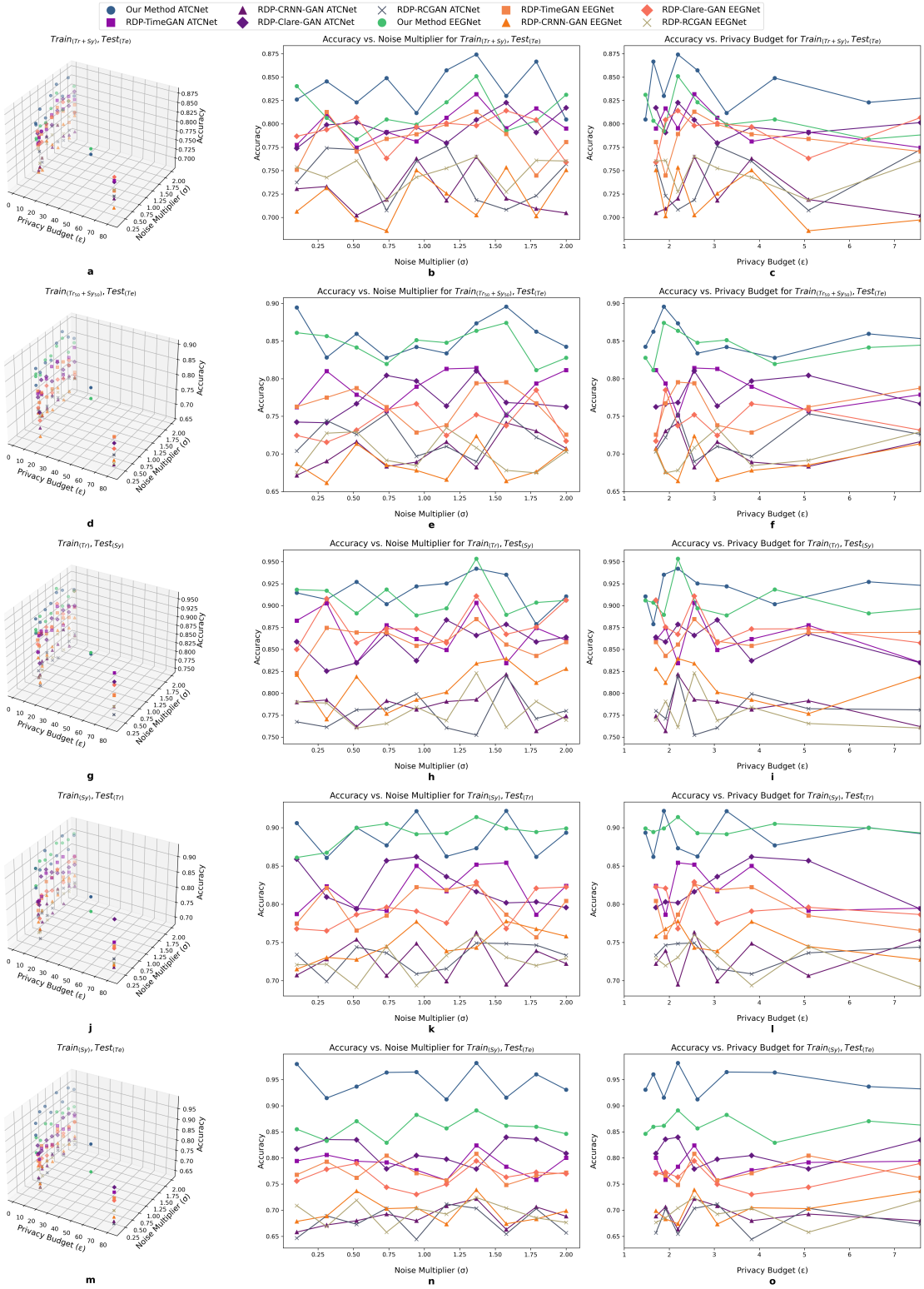
*

Figure 4: Comparison of different models under various privacy budgets and noise multipliers for subject A5.



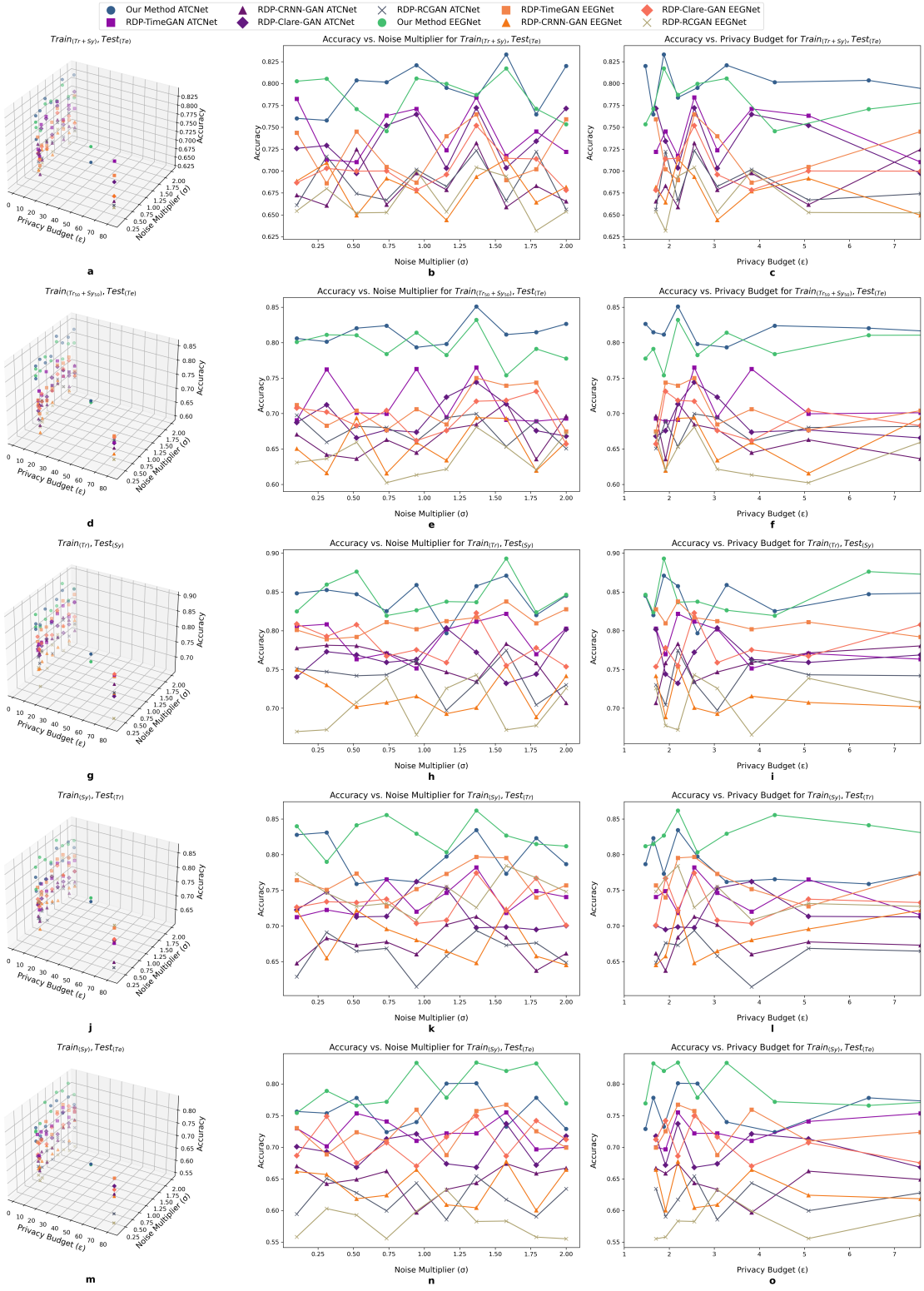
*

Figure 5: Comparison of different models under various privacy budgets and noise multipliers for subject A6.



*

Figure 6: Comparison of different models under various privacy budgets and noise multipliers for subject A7.



*

Figure 7: Comparison of different models under various privacy budgets and noise multipliers for subject A8.

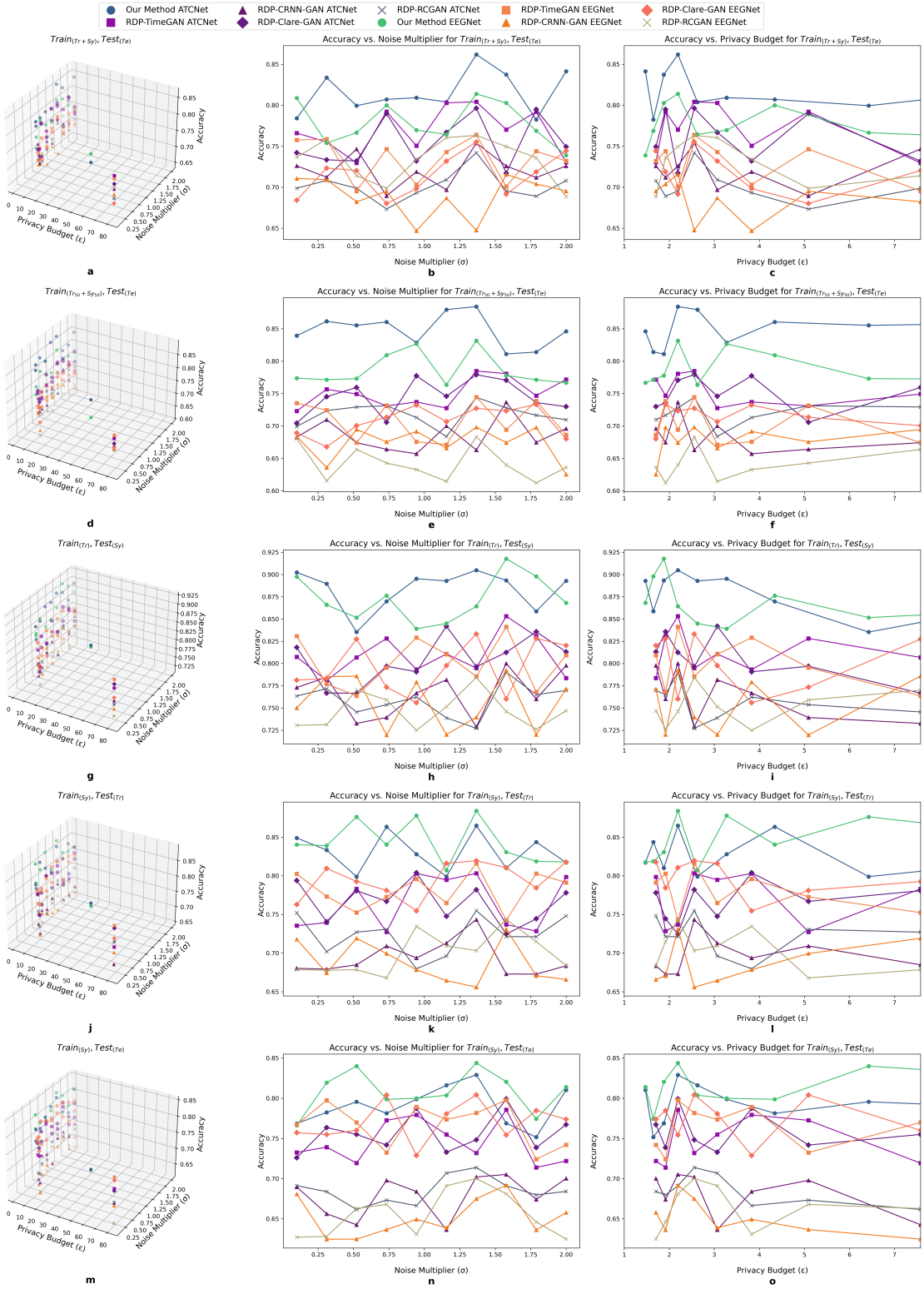


Figure 8: Comparison of different models under various privacy budgets and noise multipliers for subject A9.

Visualization of High-Dimensional EEG Data Using 3D t-SNE:

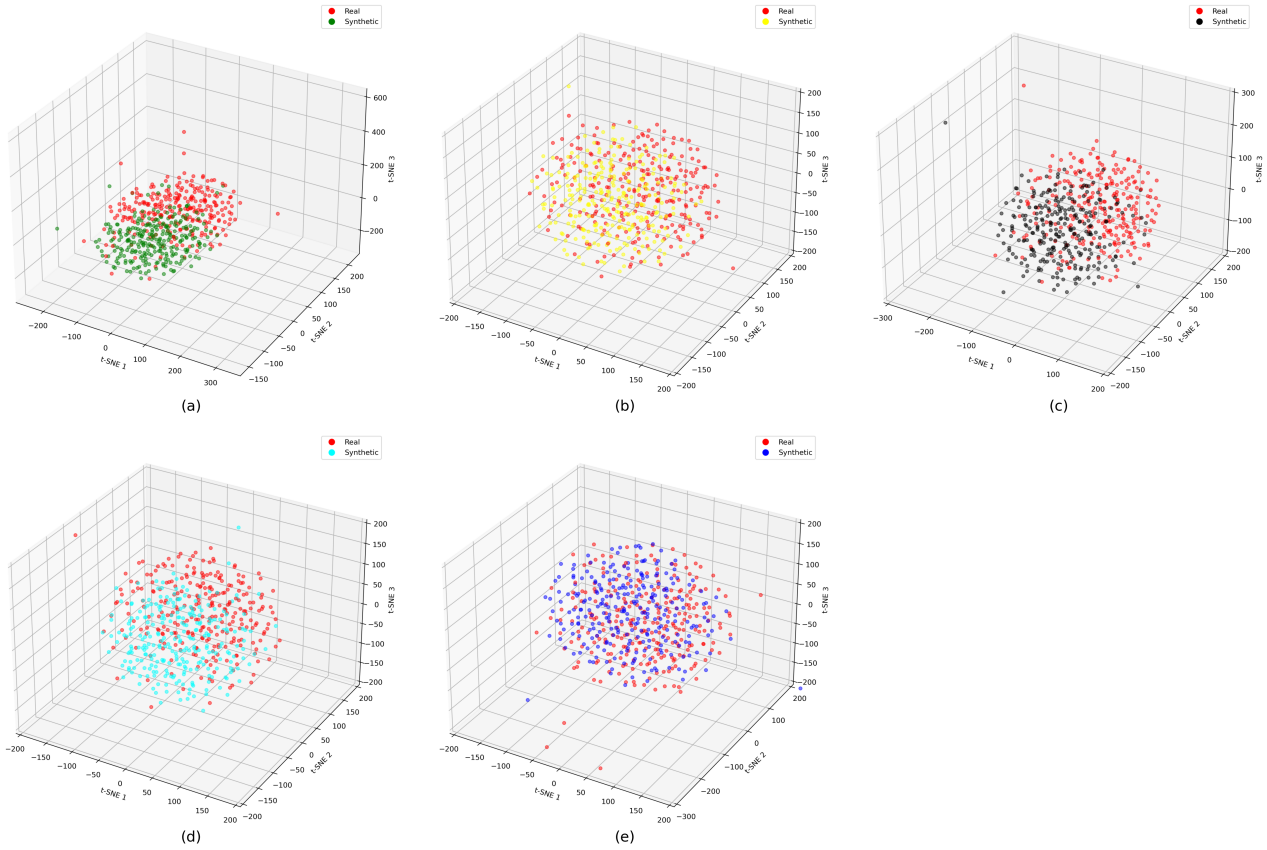


Figure 9: 3D t-SNE visualization of high-dimensional EEG data for subject A2. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

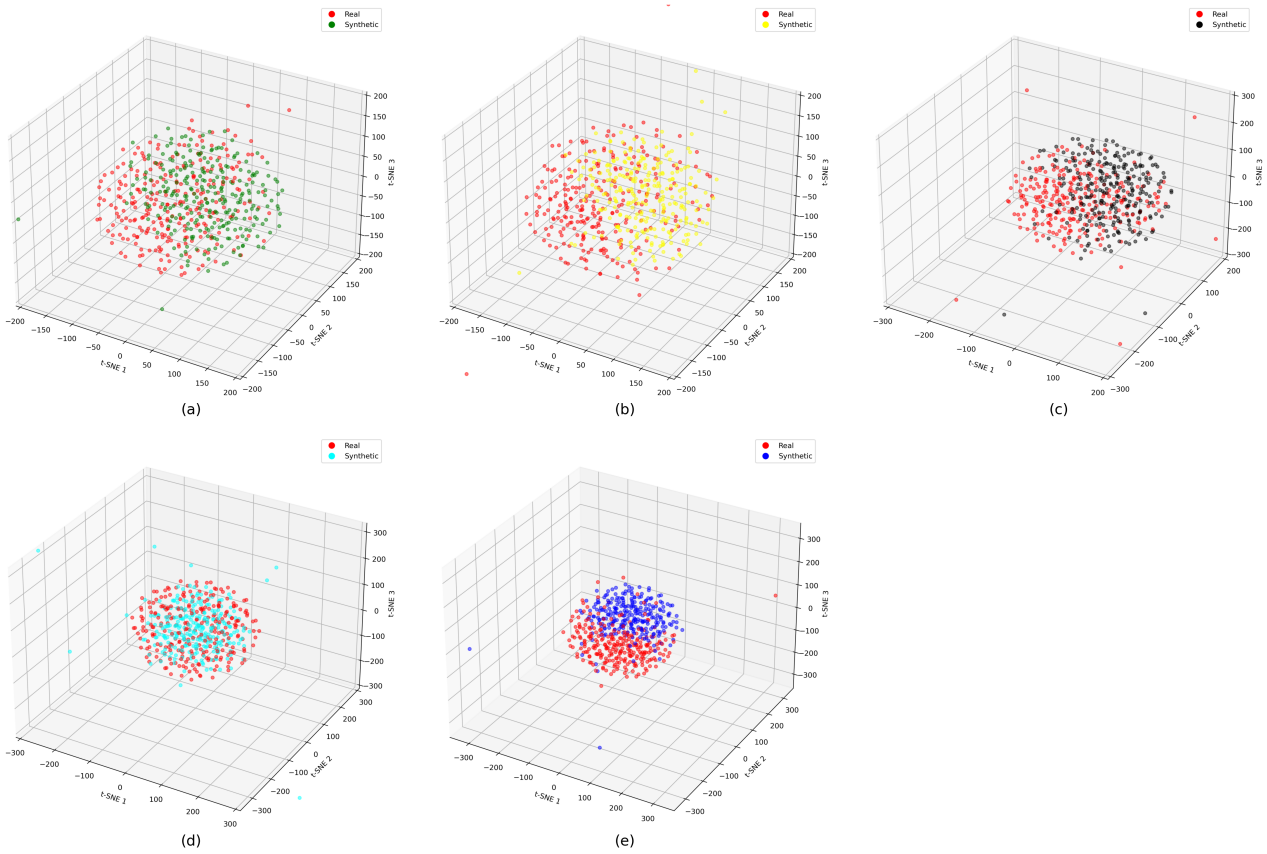


Figure 10: 3D t-SNE visualization of high-dimensional EEG data for subject A3. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

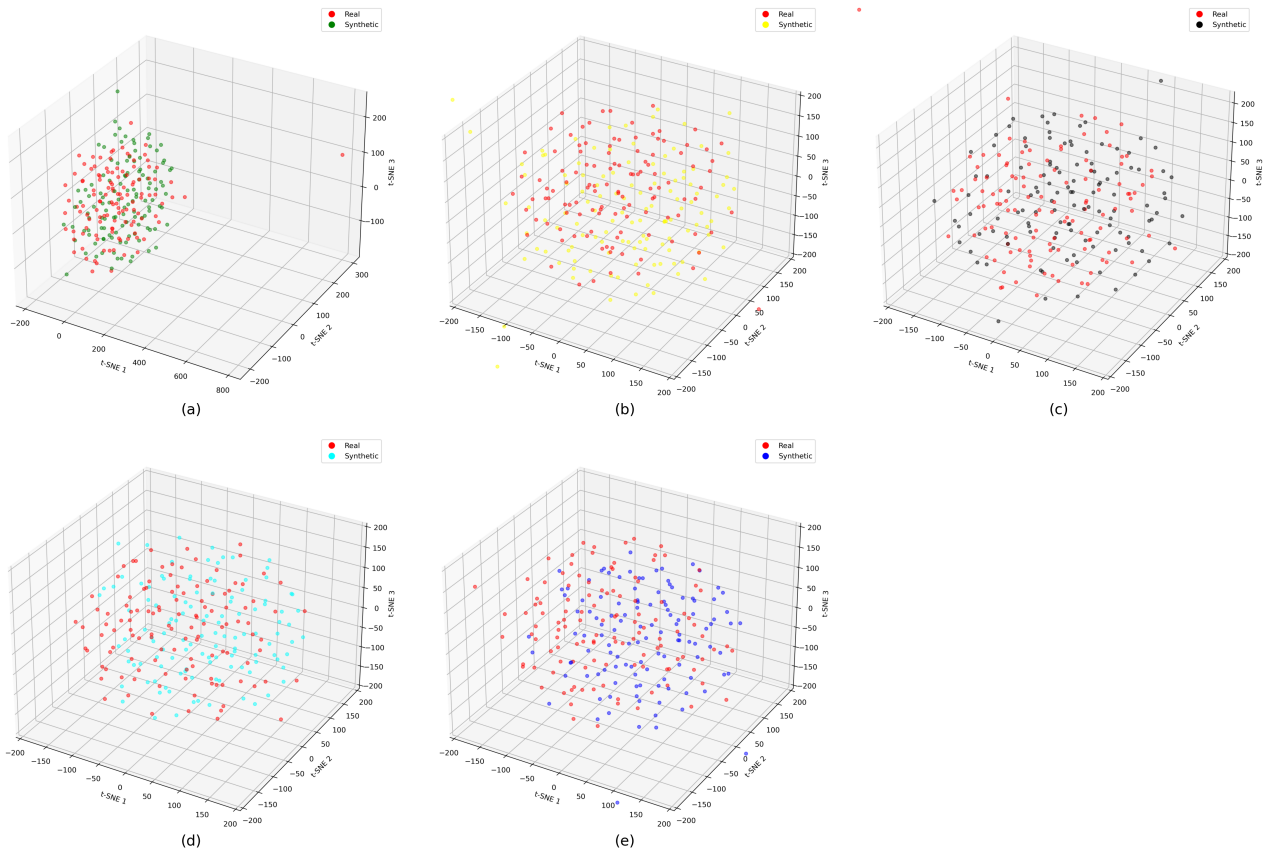


Figure 11: 3D t-SNE visualization of high-dimensional EEG data for subject A4. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

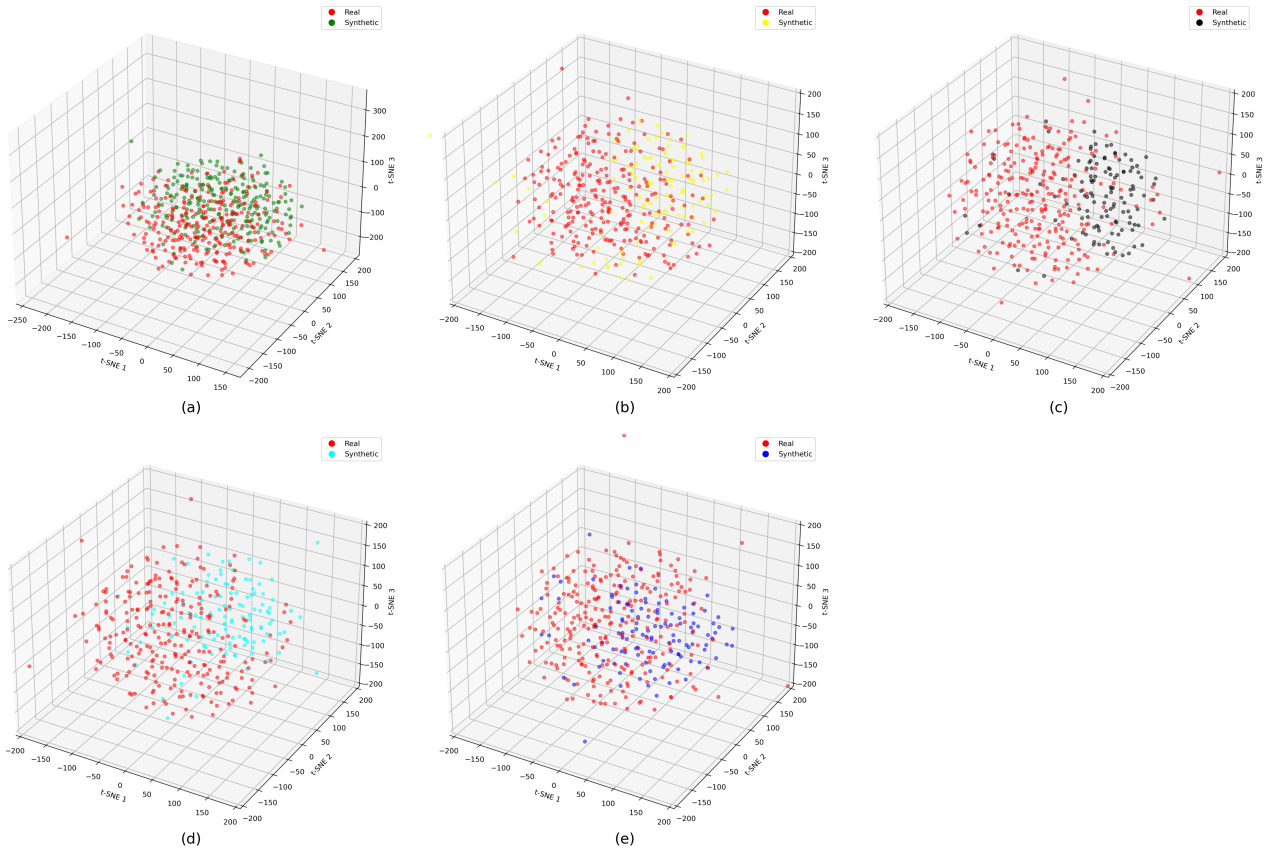


Figure 12: 3D t-SNE visualization of high-dimensional EEG data for subject A5. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

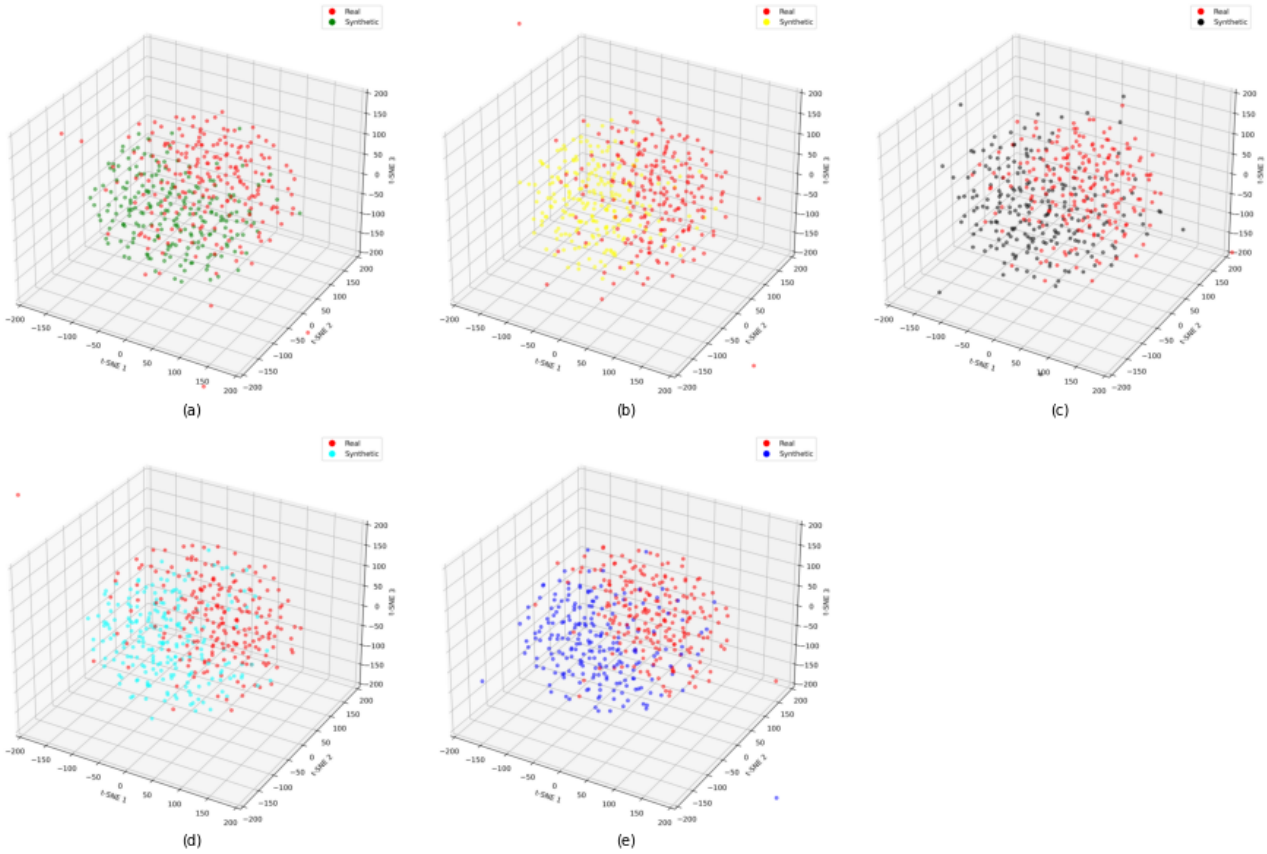


Figure 13: 3D t-SNE visualization of high-dimensional EEG data for subject A6. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

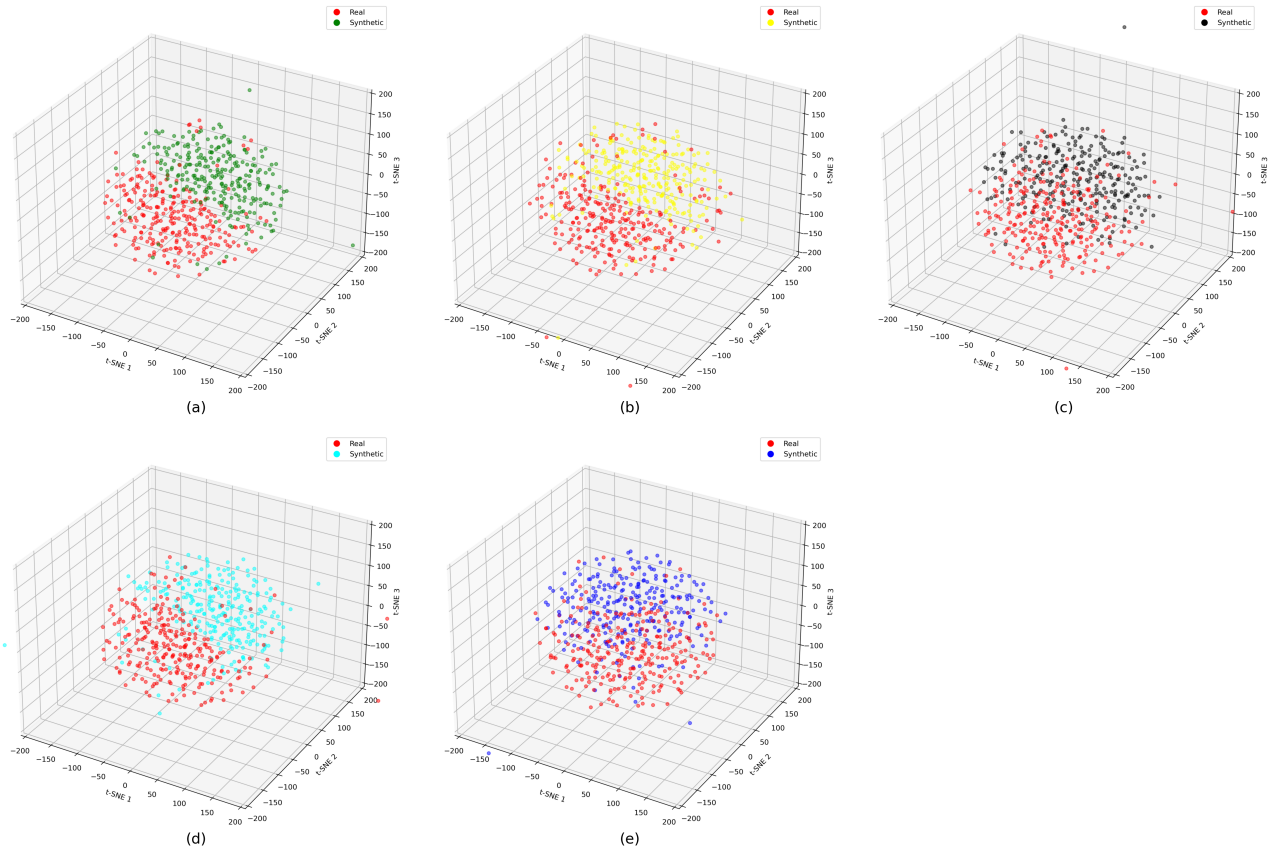


Figure 14: 3D t-SNE visualization of high-dimensional EEG data for subject A7. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

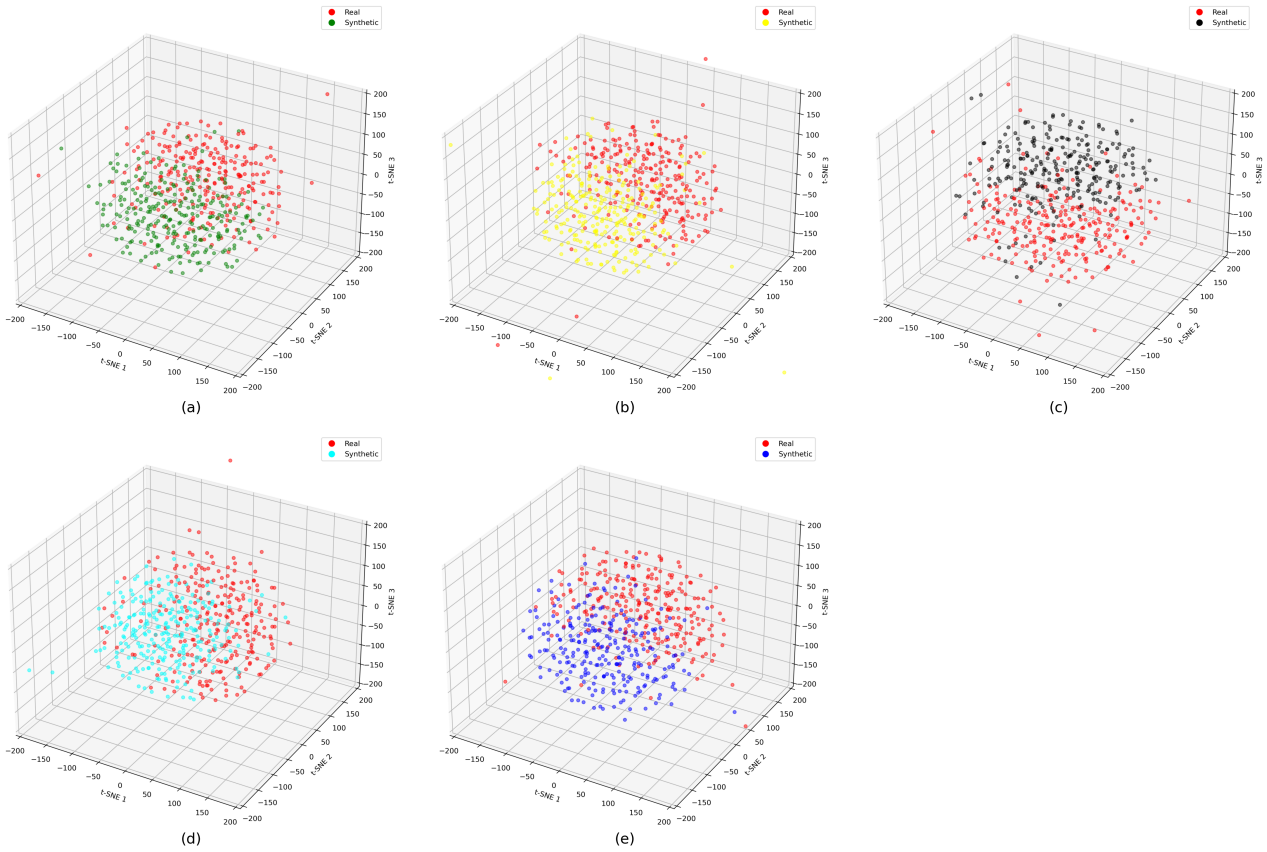


Figure 15: 3D t-SNE visualization of high-dimensional EEG data for subject A8. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

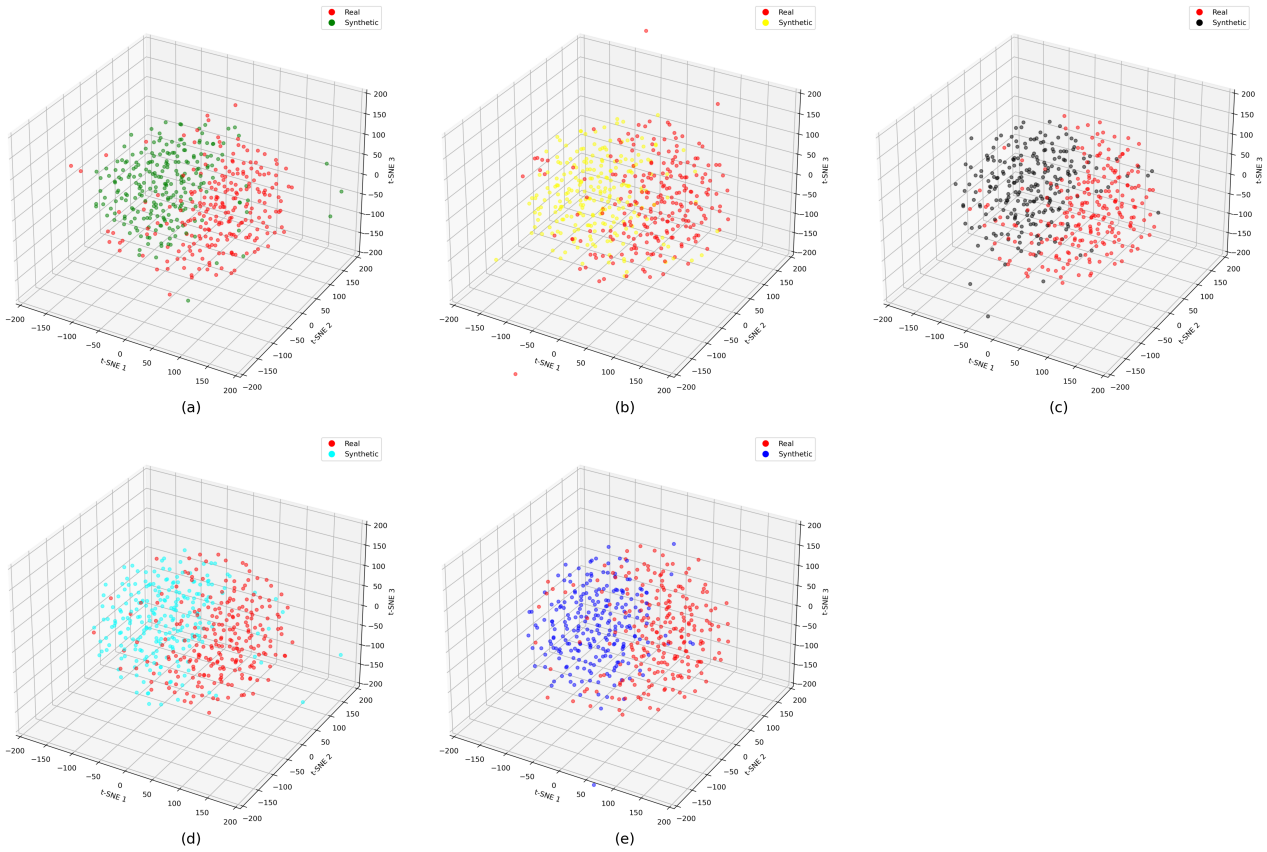


Figure 16: 3D t-SNE visualization of high-dimensional EEG data for subject A9. Each point in red color represents a real EEG data sample, and other synthetic data color-coded by the model used for generation. (a) Real data vs. proposed Spiking GAN (b) Real data vs. RDP-TimeGAN (c) Real data vs. RDP-CRNN-GAN (d) Real data vs. RDP-Clare-GAN and (e) Real data vs. RDP-RCGAN.

Appendix B: Supplementary Figures of Chapter 5

Full Black-box Attack Results:

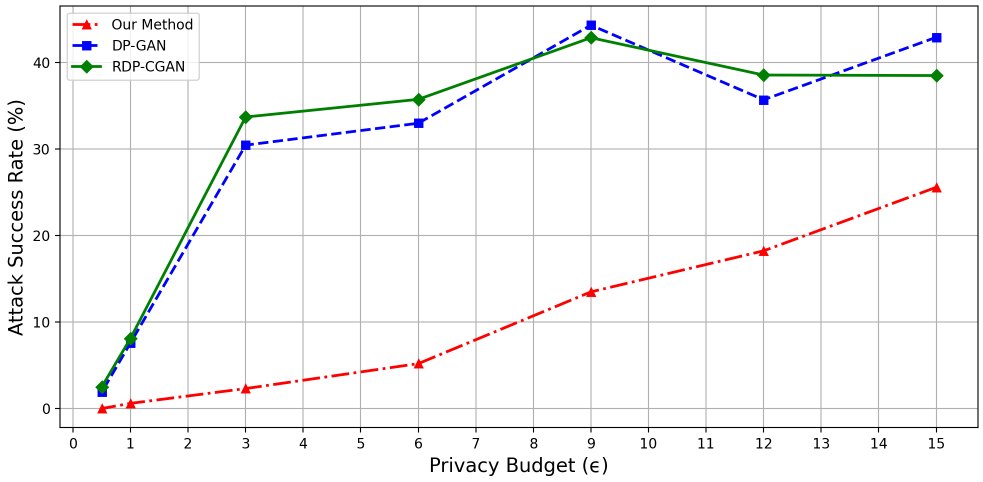


Figure 1: Attack success rate (%) for subject B2 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

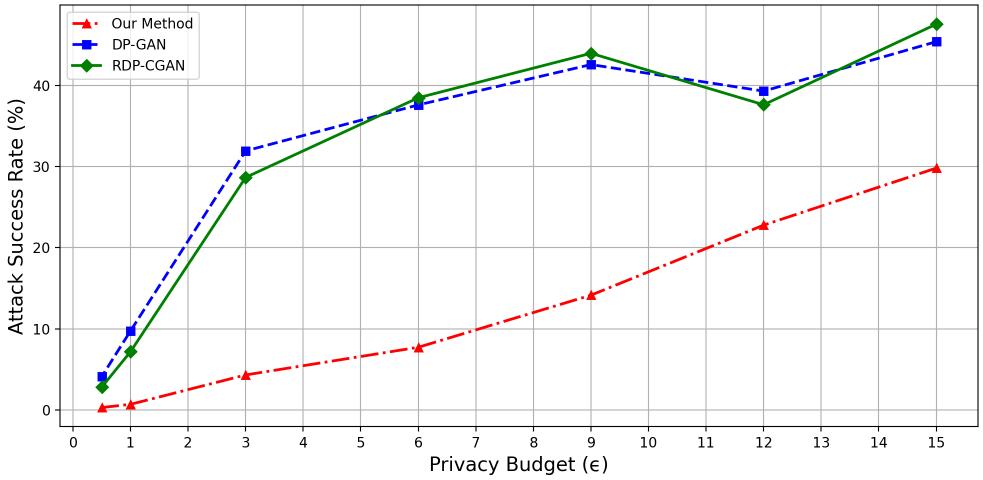


Figure 2: Attack success rate (%) for subject B3 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

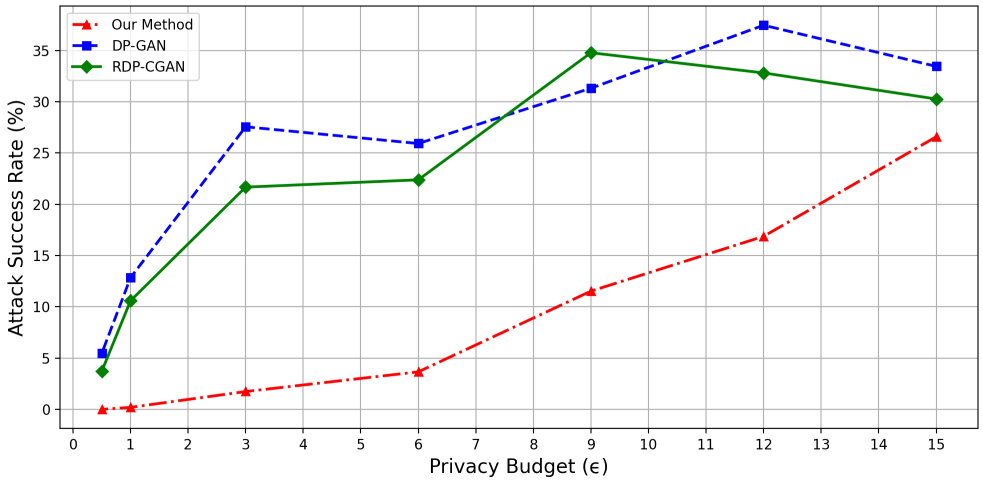


Figure 3: Attack success rate (%) for subject B4 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

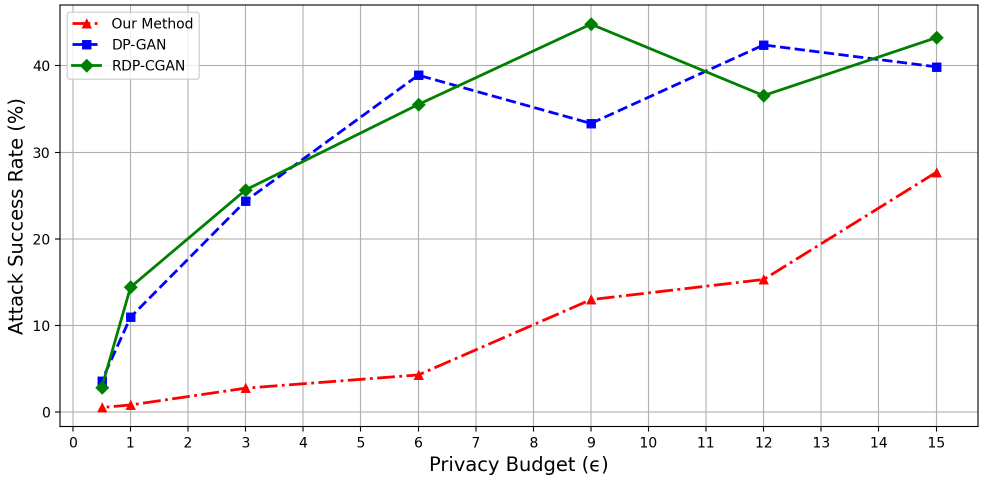


Figure 4: Attack success rate (%) for subject B5 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

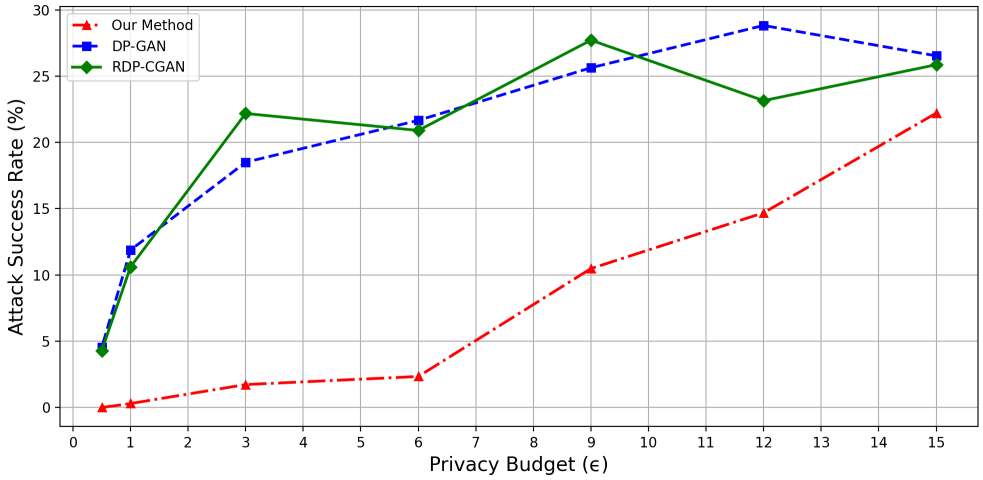


Figure 5: Attack success rate (%) for subject B6 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

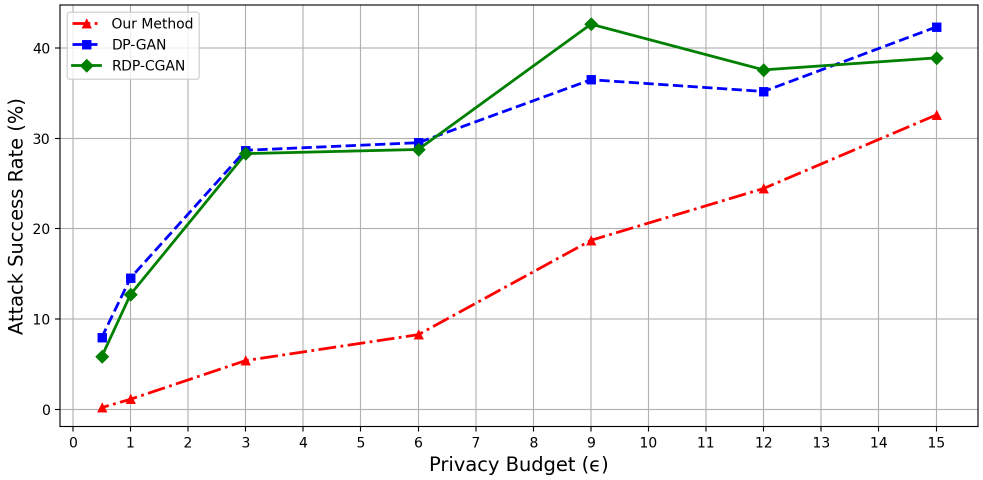


Figure 6: Attack success rate (%) for subject B7 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

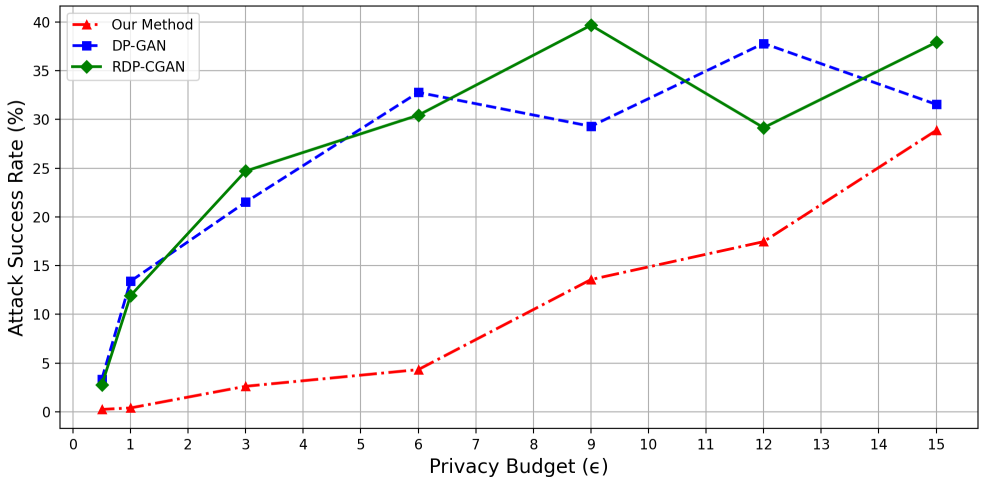


Figure 7: Attack success rate (%) for subject B8 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

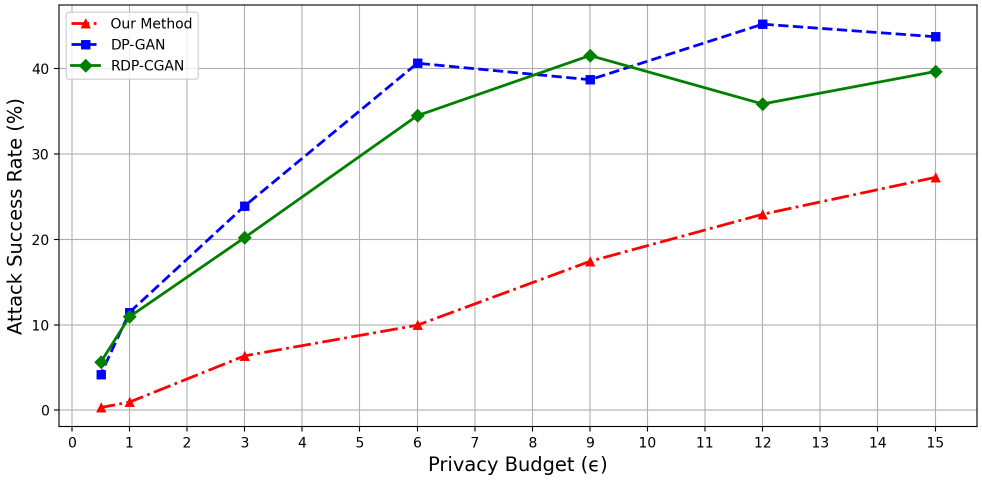


Figure 8: Attack success rate (%) for subject B9 across different ϵ values. A higher percentage signifies a more successful attack, whereas a lower percentage indicates better privacy protection.

Appendix C: Code Resources

Code Availability

The code used for the experiments and methodologies discussed in this thesis is available on request via the following link:

Google Drive Code Repository

For any further information or to request access to specific parts of the code, please feel free to contact me at:

- spaul4@lakeheadu.ca
- shouvik28paul@gmail.com
- sp.cgec@gmail.com