On the Privacy Risks of Spiking Neural Networks: A Membership Inference Analysis

Junyi Guan[*†1] **Abhijith Sharma**[*†1] **Chong Tian**[1] **Salem Lahlou**[1]

[1]Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.

Abstract

Spiking Neural Networks (SNNs) are increasingly explored for their energy efficiency and robustness in real-world applications, yet their privacy risks remain largely unexamined. In this work, we investigate the susceptibility of SNNs to Membership Inference Attacks (MIAs)—a major privacy threat where an adversary attempts to determine whether a given sample was part of the training dataset. While prior work suggests that SNNs may offer inherent robustness due to their discrete, event-driven nature, we find that its resilience diminishes as latency (T) increases. Furthermore, we introduce an input dropout strategy under black box setting, that significantly enhances membership inference in SNNs. Our findings challenge the assumption that SNNs are inherently more secure, and even though they are expected to be better, our results reveal that SNNs exhibit privacy vulnerabilities that are equally comparable to Artificial Neural Networks (ANNs). Our code is available at `https://github.com/sharmaabhijith/MIA_SNN`

1 INTRODUCTION

Spiking Neural Networks (SNNs) are a class of neural networks that emulate the discrete, event-driven mechanisms of biological neurons. Unlike Artificial Neural Networks (ANNs) that process continuous signals, SNNs communicate through discrete spikes, activating only when relevant information is present [Tavanaei et al., 2019]. The design of SNNs for handling complex tasks remains an active research area. However, their unique spiking operations impart certain inherent advantages across a wide range of practical applications [Nunes et al., 2022, Stan and Rhodes, 2024, Baek and Lee, 2024]. In particular, advances in neuromorphic computing have elicited greater traction in the use of SNNs for sparse and efficient hardware implementations of neural networks [Yao et al., 2024, Minhas et al., 2024].

The primary adoption of SNNs is driven by two key advantages. First, the event-driven nature of spikes allows for reduced computations, leading to significant energy savings. Hence, SNNs naturally emerge as a favorable choice for energy constrainted domains, such as edge devices [Yan et al., 2024], Internet of Things (IoT) [Liu and Zhang, 2022] and low-power embedded platforms [Syed et al., 2021]. Second, SNNs have demonstrated resilience against various adversarial threats. Their discrete spike-based processing enables enhanced security by capturing only the relevant information, and obscuring less important patterns which are often exploited by adversaries [Nagarajan et al., 2022].

Given these advantages, SNNs have emerged as a compelling choice for various real-world applications in recent years. Several studies have explored replacing ANN strategies with spiking dynamics, demonstrating their utility in domains such as image classification, object detection, and speech recognition [Yamazaki et al., 2022]. Their ability to process information efficiently and robustly has led to their integration into industrial systems, enhancing performance and reliability [Wang et al., 2023a, Zhou et al., 2023].

Simultaneously, with the rapid emergence of high-performance and efficient architectures [Lin et al., 2022, Vaswani, 2017, Brown et al., 2020, Sze et al., 2017], organizations worldwide are racing to adopt AI models across domains. As machine learning models are increasingly deployed in sensitive domains, data privacy has become a critical concern. Moreover, AI users are now more aware and apprehensive about potential data leaks [Jobin et al., 2019]. This shift has sparked an unprecedented demand for AI service providers to safeguard the privacy of the data they manage. Unfortunately, numerous studies have exposed the

[*] Equal Contribution

[†] Correspondence to: `junyi.guan@mbzuai.ac.ae`, `abhijith.sharma@mbzuai.ac.ae`

vulnerabilities of current systems to privacy attacks, highlighting the urgent need for private models [Liu et al., 2021, Rigaki and Garcia, 2023].

One of the most significant threats in this domain is Membership Inference Attacks (MIAs). In an MIA, an adversary aims to determine whether a specific data sample was part of a model's training set, potentially leading to privacy breaches [Hu et al., 2022]. Studying MIAs allows researchers to identify vulnerabilities and design robust models to withstand the threat. For instance, in healthcare, ensuring that patient data cannot be inferred from a predictive model is vital for maintaining confidentiality [Hu et al., 2022]. In addition, regulating organizations can leverage MIA as a metric to conduct predeployment audits of AI models to avoid leakage of sensitive data.

Given the increasing deployment of SNNs in various applications [Chu et al., 2022, Hussaini et al., 2024, Zhou et al., 2023, Bos and Muir, 2023], it is imperative to assess their vulnerability to MIAs. Preliminary studies suggest that the discrete nature of SNN outputs may offer some protection against such attacks. However, existing research is limited, with only a few studies addressing this issue, often using metrics that may not fully capture the nuances of SNN behavior [Carlini et al., 2022]. Motivated by our observations, we aim to investigate: Are SNNs truly as robust to MIAs as we might hope, or do they share the same privacy risks as their counterparts that we have yet to uncover?

To address the privacy concerns of SNNs, our work systematically analyzes their resilience against MIAs through the following contributions:

- Develop efficient empirical estimates of state-of-the-art MIAs to simulate the online attack setting.
- Propose a simple input dropout method to introduce stochasticity in the membership inference process, improving attack effectiveness in SNNs.
- Analyze the impact of SNNs' latency and training methods on their vulnerability to MIAs, and also compare them with ANNs.
- Investigate the feasibility of using ANNs to perform MIAs against SNNs, when attacker does not have the knowledge of SNNs' parameters.

To contextualize our contributions, we first review existing research on SNNs and MIAs, including concurrent work, in Section 2. Section 3 provides a concise overview of the SNN dynamics, training, and MIA formulation. In Section 4, we propose an input dropout-based strategy to efficiently evaluate SNNs against MIAs, while also comparing them with traditional ANNs. Finally, we validate our hypothesis through experiments and results in Section 5.

2 RELATED WORK

Spiking Neural Networks (SNNs): Designing and training SNNs is challenging due to their sensitivity to hyperparameters such as membrane threshold and synaptic latency, both of which significantly impact performance [Bouvier et al., 2019]. Consequently, many existing methods focus on achieving low-latency inference with improved convergence while maintaining accuracy [Meng et al., 2022]. Traditional approaches like surrogate gradient learning [Neftci et al., 2019] and temporal coding [Bellec et al., 2018] have been further enhanced by advanced techniques [Dampfhoffer et al., 2023]. Notably, Deng et al. [2022] introduced a gradient re-weighting mechanism to improve the temporal efficiency of SNN training. To eliminate the need for manual threshold selection, Bojkovic et al. [2024] proposed a data-driven approach for threshold selection and potential initialization. Their method also facilitates the conversion of trained ANNs into SNNs, enabling efficient and high-performance training even in low-latency settings ($T = 1, 2, 4$). These advancements are crucial for real-time applications. Beyond energy efficiency, SNNs have also been explored for their inherent robustness against adversarial [Nomura et al., 2022, El-Allami et al., 2021] and model inversion attacks [Kim et al., 2022], further reinforcing their potential towards robust AI.

Membership Inference Attacks: The threat of Membership Inference Attacks (MIAs) was first demonstrated by Shokri et al. [2017] in a simple Machine Learning-as-a-Service (MLaaS) black-box setting. Since then, extensive research has explored the privacy risks associated with diverse neural network architectures for a wide range of applications [Hu et al., 2022, Yeom et al., 2018, Salem et al., 2019]. Despite significant advancements and robustness characteristics of SNNs [Kim et al., 2022], their vulnerability to MIAs remains largely unclear and underexplored [Sharmin et al., 2019].

The inconsistencies in evaluation metrics and experimental settings in existing studies have made direct comparisons of MIA techniques challenging [Hu et al., 2023]. However, Carlini et al. [2022] presented MIA from first principles, emphasizing the importance of analyzing the Receiver Operating Characteristic (ROC) curve in attack's assessments. The ROC fully captures the tradeoff between True Positive Rate (TPR) and False Positive Rate (FPR) of the membership data across different classification thresholds. Reporting TPR under extremely low FPR conditions ($\leq$1% and $\leq 0.1\%$) is particularly crucial, as attackers prioritize confidently identifying members over overall accuracy. More recently, Zarifzadeh et al. [2024] proposed a state-of-the-art attack, called robust MIA (RMIA), and generalized all other existing MIAs under the umbrella of their attack formulation. RMIA also achieved highly effective attack performance with a limited number of shadow/reference models - auxiliary models trained on data with similar properties to the target model's training data.

Concurrent Work: While existing research primarily focuses on traditional ANNs, the membership privacy risks in SNNs remain largely unexamined. A recent study by Li et al. [2024] explored the robustness of SNNs against MIAs, incorporating diverse experimental settings and assessing the impact of data augmentation. However, despite these contributions, the study suffers from several critical limitations. It relies on biased evaluation metrics such as balanced accuracy, which can obscure the true effectiveness of MIAs, and employs outdated training techniques for SNNs [Carlini et al., 2022]. Nowadays, in many research papers, AUC and TPR at very low FPR are the main metrics to study the performance of MIAs. Additionally, the evaluation is conducted on simple datasets, failing to provide meaningful insights into real-world scenarios. More importantly, the study neglects key advancements in attack methodologies, such as RMIA, limiting the comprehensiveness of its findings. Furthermore, the analysis is restricted to timestep variations in neuromorphic datasets, lacking a systematic investigation of static datasets. These shortcomings underscore the insufficiency of existing efforts in rigorously assessing membership privacy risks in SNNs. A more sound evaluation is necessary to bridge this gap and uncover the true privacy vulnerabilities of SNNs.

3 PRELIMINARIES

3.1 SPIKING NEURAL NETWORK

3.1.1 Data Encoding

Unlike ANNs, SNNs have an additional temporal dimension to represent the data. This means that, in the case of image processing, the input to the network is not a single image, but is instead encoded over time into a sequence of T images, where T is known as *time step* or *latency*. There are various methods to encode images [Almomani et al., 2019, Wang et al., 2023b], however, in our work, we use the classical *constant encoding* that simply replicates the original image T times, and the resulting data is fed into the SNN.

3.1.2 Forward Pass

The forward process of SNN begins with encoding the input data. As the spikes are fed into the network, each neuron updates its membrane potential based on the spikes it receives. If a neuron's membrane potential exceeds a certain threshold, the neuron fires a spike and sends it to the next layer through synapses. The Equation 1 and 2 together describes the dynamics of a neuron in an SNN.

$$U_l^t = \lambda U_l^{t-1} + w_l O_{l-1}^t \tag{1}$$

$$O_l^t = H(U_l^t - \theta_{tr}), \qquad U_l^t = U_{reset} \tag{2}$$

Equation 1 represents the membrane potential U_l^t of a neuron at layer l and time t. The membrane potential is updated by combining two components: the decayed potential from the previous time step λU_l^{t-1}, where λ is a decay factor between 0 and 1, and the weighted input from the previous layer's output $w_l O_{l-1}^t$, where w_l is the synaptic weight and O_{l-1}^t is the output (spike) from the previous layer at time t. If the decay factor λ is 1, the neuron is called an *Integrate-and-Fire (IF)* neuron; otherwise, it is a *Leaky Integrate-and-Fire (LIF)* neuron.

The Equation 2 describes the neuron's output at time t, where H is the Heaviside step function. If the membrane potential U_l^t exceeds a certain threshold θ_{tr}, the neuron generates an output spike ($O_l^t = 1$); otherwise, no spike is generated ($O_l^t = 0$). Once the neuron fires a spike, its membrane potential is reset to a specific value U_{reset}, preparing the neuron for further spike processing in subsequent time steps. For the last layer L, the output of the network is U_L^T, which is the membrane potential for the final time T.

3.1.3 SNN Training

The training of SNNs can be categorized into two main approaches: direct backpropagation and ANN-to-SNN conversion (hybrid). However, in MIAs, the probability vector output by the neural network is the most critical feature for determining the membership status of a specific data point. Therefore, when studying the effectiveness of MIAs, we should focus more on the forward propagation process of SNNs rather than their backpropagation process.

Training SNNs directly from scratch can be challenging sometimes as the performance is sensitive to hyperparameters, like membrane threshold. On the other hand, SNN converted from trained ANNs along with hybrid training can help achieve good accuracy due to better initialization of threshold from the trained ANNs [Kim et al., 2022]. For example, a recent work from Bojkovic et al. [2024] shows how the activation values from the layers of a trained ANN can be utilized to initialize the membrane threshold in SNN. This method coupled with hybrid training [Rathi et al., 2021] enables achieving good performance even for low-latency SNNs, which is beneficial for diverse real-time applications.

3.2 MEMBERSHIP INFERENCE

In binary classification, let $D = \{(x_i, y_i)\}_{i=1}^n$ be a dataset where each $x_i \in \mathbb{R}$ represents a one-dimensional feature value, and $y_i \in \{0, 1\}$ is the class label, where $y_i = 1$ indicates a positive sample and $y_i = 0$ indicates a negative sample. A threshold-based classifier with threshold t classifies a sample as positive if $x_i \geq t$ and negative otherwise. For a given threshold t, the true positive rate (TPR) and false positive rate (FPR) is defined as shown in Equation 3

$$\text{TPR}(t) = \frac{\sum_{i=1}^{n} \mathbf{1}(x_i \geq t, y_i = 1)}{\sum_{i=1}^{n} \mathbf{1}(y_i = 1)}, \quad \text{FPR}(t) = \frac{\sum_{i=1}^{n} \mathbf{1}(x_i \geq t, y_i = 0)}{\sum_{i=1}^{n} \mathbf{1}(y_i = 0)} \quad (3)$$

The Receiver Operating Characteristic (ROC) curve is the set of points $\{(\text{FPR}(t), \text{TPR}(t)) \mid t \in \mathbb{R}\}$, and the Area Under the Curve (AUC) is given by the integral $\int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$.

Definition 1 (Membership Inference Game)**.** *Let π be an underlying data distribution and let $\mathcal{A}$ be a training algorithm. The game proceeds between two entities: a* challenger *and an* attacker *[Carlini et al., 2022, Zarifzadeh et al., 2024] .*

1. ***Dataset Sampling and Model Training.*** *The challenger samples a dataset S from π, and trains a model θ using $\mathcal{A}$ on S. Here, θ is the target model.*
2. ***Challenge Sample Selection.*** *The challenger tosses a fair coin $b \in \{0, 1\}$.*
   - *If $b = 1$, it picks a point x uniformly at random from S. In this case, x is a member of θ.*
   - *If $b = 0$, it draws a sample x from π (ensuring $x \notin S$). In this case, x is not a member of θ.*

   The challenger then provides both the target model θ and the sample x to the attacker.
3. ***Attacker's Inference.*** *The attacker, having access to θ (and potentially query access to the distribution π), computes a* membership score $\text{Score}_{MIA}(x; \theta)$ *indicating how likely it believes x was contained in S. Based on a chosen threshold β, the attacker issues a membership decision*

   $$\hat{b} \leftarrow \mathbf{1}\big[\text{Score}_{MIA}(x;\theta) \geq \beta\big].$$

   Here, $\mathbf{1}[\cdot]$ is the indicator function, hence, $\hat{b} = 1$ (member) if the score exceeds the threshold, and 0 (non-member) otherwise.
4. ***Outcome and Evaluation.*** *The attacker's prediction $\hat{b}$ is compared against the true bit b. Hence, membership is inferred by the attacker if $\hat{b} = b$. By repeating this experiment over many trials, we can plot the ROC for each β. The leakage of the model is often characterized by the achievable trade-off between TPR and FPR across all possible threshold values. The attack method is considered good if it achieves a high AUC score.*

4 METHODOLOGY

In this section, we describe our methodology for evaluating SNNs against MIA. Our attack setup follows prior research [Zarifzadeh et al., 2024, Carlini et al., 2022]. In MIAs, an adversary uses shadow models (or reference models) to replicate the target model's behavior. These reference models are trained on the target model's data distribution. If a reference model's training data contains the target query, it is an *IN model*; otherwise, it is an *OUT model*.

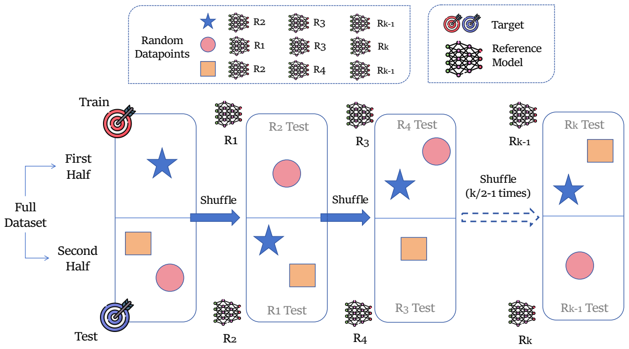

Figure 1: Illustration of Data Splitting Strategy

A MIA is classified as an online attack if half of the reference models are *IN model*, ensuring an unbiased estimate. However, this approach is computationally expensive since *IN model* must be trained for each target query. In contrast, offline attacks use only *OUT model* and apply a scaling factor as a hyperparameter to approximate the online setting. This creates a trade-off between attack strength and computational cost.

To rigorously evaluate the vulnerability of SNNs to Membership Inference Attacks (MIAs), we prioritize **simulating an online attack scenario**. Online MIAs are preferred as they more realistically reflect real-world attack settings and are often more revealing of subtle vulnerabilities compared to offline approaches. To achieve this efficiently, we propose a data splitting strategy that emulates an online attack without incurring the computational cost of repeatedly retraining models. Our method begins by partitioning the dataset D into two equal halves: one to train our target SNN model and the other to serve as its test set. Subsequently, we shuffle the original dataset D. This shuffled dataset is then iteratively divided to train n pairs of reference models, resulting in a total of $2n$ reference models. Crucially, while each reference model is trained on a substantial portion of D derived from this process, the data splitting is carefully designed to ensure that each query from the target model's test set is present in approximately half of the reference models. This balanced exposure is essential for unbiased MIA evaluation.

This approach effectively simulates an online attack environment by providing a diverse set of pre-trained reference models against which to assess membership, all without the need for computationally expensive online model retraining. Consistent with standard MIA assumptions in the literature [Zarifzadeh et al., 2024], we acknowledge that our reference models are designed under the common assumption that attackers possess knowledge of the target model's architecture. Our approach is depicted in Figure 1, showcasing an efficient and fair data-splitting strategy tailored for online attack settings. The dataset D is initially balanced and split into training (Tr) and test (Ts) sets for the target model. By iteratively shuffling and re-splitting D, a total of $2n$ reference models are trained. Each target query is included in exactly half of these reference models, ensuring a fair distribution and enabling robust evaluation. This method avoids redundant training and ensures computational efficiency while maintaining unbiased experimental results.

4.1 MIA ATTACK METHODS: BACKGROUND AND MODIFICATIONS

We start by describing the common MIA attacks: *Attack-P, Attack-R, and RMIA* [Ye et al., 2022, Zarifzadeh et al., 2024] and adapt them to our experimental setting. These three attack methods exploit different assumptions about model confidence to infer the membership. The parameters for the attack methods can be defined as follows. Let θ be the target model and θ' a reference model. The audited sample is x, while z is drawn from the dataset D of size N. The model's confidence output is $\Pr(\cdot|\theta)$. Reference models trained with and without x are denoted as θ'_x and $\theta'_{\bar{x}}$, respectively. The confidence of any model m, $\Pr(d|m)$, is given by the softmax score of the true label of input d.

Based on this setting, the original *Attack-P* is defined in Table 1. Note that *Attack-P* operates without reference models, assuming that if x belongs to θ's training data, the model's confidence in x is typically higher than for an arbitrary sample z from the overall distribution. The empirical estimation of the MIA score is given by:

$$\frac{1}{N}\sum_{j=1}^{N}\mathbf{1}\Big(\Pr(x \mid \theta) \geq \Pr(z_j \mid \theta)\Big) \tag{4}$$

However, since our setting provides access to the full dataset D, we refine the MIA score by considering all $x \in D$. Furthermore, by Proposition 1, we adopt $\Pr(x|\theta)$ as the modified *Attack-P* score, ensuring preservation of the ROC curve. While the weakest among the three, it provides insights into the model's confidence distribution.

Attack-R builds on *Attack-P*'s approach, but with the help of reference models. Unlike *Attack-P*, the *Attack-R* compares θ's confidence on x with that of the reference models (θ'), assuming that the target model assigns higher confidence to samples it has seen during training. Similar to the *Attack-P*'s empirically estimated score, the modified *Attack-R* can be computed as shown in Table 1.

Attack	Original	Modified
Attack-P	$\Pr_{z}\left(\frac{\Pr(x\|\theta)}{\Pr(z\|\theta)} \geq 1\right)$	$\Pr(x\|\theta)$
Attack-R	$\Pr_{\theta'}\left(\frac{\Pr(x\|\theta)}{\Pr(x\|\theta')} \geq 1\right)$	$\frac{1}{2n}\sum_{\theta'}\mathbf{1}\left(\Pr(x\|\theta) \geq \Pr(x\|\theta')\right)$
RMIA	$\Pr_{z}\left(\frac{\Pr(\theta\|x)}{\Pr(\theta\|z)} \geq 1\right)$	$\frac{\Pr(x\|\theta)}{\frac{1}{2n}\left(\sum_{\theta'_x}\Pr(x\|\theta'_x) + \sum_{\theta'_{\bar{x}}}\Pr(x\|\theta'_{\bar{x}})\right)}$

Table 1: Attack formulations for computational efficiency by removing dependence on z.

Finally, *RMIA* also known as robust-MIA is the strongest of all. It exploits the assumption that if θ is trained on x, the likelihood of θ given x is higher than for any unrelated sample z. Similar to *Attack-P*, by Proposition 1, one can easily find the equivalence between the original and the modified *RMIA*. Moreover, with the data-splitting method described above, each x is used to train an equal number of reference models with (θ'_x) and without ($\theta'_{\bar{x}}$) it, ensuring an unbiased estimation of $\Pr(x)$. This setup follows Zarifzadeh et al. [2024] and provides all the necessary information for an optimal online RMIA attack. Together, these attacks provide a progressive understanding of how a model's confidence can reveal membership information.

Proposition 1. *Let S_1 and S_2 be two MIA scoring functions:*

$$S_1(x) = \frac{1}{N}\sum_{j=1}^{N}\mathbf{1}\Big(\Pr(x \mid \theta) \geq \Pr(z_j \mid \theta)\Big), \tag{5}$$

$$S_2(x) = \Pr(x \mid \theta), \tag{6}$$

where $\{z_j\}_{j=1}^{N}$ is a fixed set of samples. Suppose $x_1, x_2 \in D$. Then, S_1 and S_2 are equivalent, meaning that, $S_2(x_1) > S_2(x_2)$, if and only if $S_1(x_1) > S_1(x_2)$, and their resulting ROC curves are similar.

4.2 DROPOUT-ENHANCED MIA METHOD: INTRODUCING PREDICTION STOCHASTICITY

Numerous existing works have demonstrated the robustness of SNNs against adversaries due to their discrete spiking behavior [Kim et al., 2022, Sharmin et al., 2019]. Similarly, we expect MIAs to be ineffective, especially on low-latency Spiking SNNs (also validated by our results of Section 5.1.2). It is not because SNNs fail to remember the training data, but because their discrete representations cause member and non-member samples to overlap. Interestingly, we observe that MIA and the out-of-distribution (OOD) detection task share the same principle: the model exhibits higher confidence on in-distribution data (members) compared to the out-of-distribution data (non-members) [Hendrycks and Gimpel, 2017]. Fundamentally, both methods are related to epistemic uncertainty, which arises due to lack of knowledge or data, and can be reduced with additional information or improved modeling [Der Kiureghian and Ditlevsen, 2009, Lahlou et al., 2023].

In practice, dropout is a widely used and effective technique for regularization [Srivastava et al., 2014], uncertainty estimation [Gal and Ghahramani, 2016], [Sun et al., 2023], and OOD detection [Hendrycks and Gimpel, 2017, Nguyen et al., 2022]. Techniques like Monte Carlo (MC)-Dropout [Gal and Ghahramani, 2016] introduce stochasticity during inference, with variations in output softmax probabilities approximating model uncertainty. In the context of MIA, directly applying MC-Dropout is not feasible, as it requires access to model parameters, which is unavailable in a black-box setting. To address this, we propose using input dropout, as an approximation of MC-Dropout (in the first layer), introducing stochasticity to the predictions without requiring access to the model's internals. As evidenced by our experiments, low-latency SNNs often struggle to differentiate subtle distinctions between member and non-member data. By perturbing the input through dropout, we aim to reduce the confidence of non-member data while maintaining the relative confidence of member data, thus enhancing the separation and improving the MIA's effectiveness.

Similarities between weight and input dropout: Consider a simple single-layered model given by $f_W : \mathbb{R}^d \to \mathbb{R}, \quad f_W(X) = \varphi(W^T \cdot X)$, where $W \in \mathbb{R}^d$ is the weight vector, $X \in \mathbb{R}^d$ represents the input and φ is the non-linear activation. During dropout, a Bernoulli mask $M \in \mathbb{R}^d$ with probability $(1 - p)$ modifies the input and weight as $\tilde{X} = M \odot X$ and $\tilde{W} = M \odot W$, where $\odot$ denotes the Hadamard product. The equivalence between the expected outputs of input dropout $\mathbb{E}_M[f_W(\tilde{X})]$ and weight dropout $\mathbb{E}_M[f_{\tilde{W}}(X)]$ is shown in Equation 7:

$$\begin{aligned}\mathbb{E}_M[f_W(\tilde{X})] &= \mathbb{E}_M\left[\varphi\left(W^T \cdot (M \odot X)\right)\right] \\ = \mathbb{E}_M\left[\varphi\left(\sum_i w_i(M_i x_i)\right)\right] &= \mathbb{E}_M\left[\varphi\left(\sum_i (M_i w_i) x_i\right)\right] \\ = \mathbb{E}_M\left[\varphi\left((M \odot W)^T \cdot X\right)\right] &= \mathbb{E}_M[f_{\tilde{W}}(X)]\end{aligned} \tag{7}$$

However, in case of multi-layered networks, weight dropout is applied to each layer individually, which differentiates it from input dropout. Nevertheless, both methods are capable of introducing stochasticity into the model's predictions. Hence, based on our hypothesis, instead of modifying the internal layers, we randomly drop parts of the input data. For each forward pass $i \in \{1, \ldots, N\}$, a binary mask with elements sampled from the Bernoulli (p) mask inputs: $\tilde{X} = X \odot M$ is fed into the model θ. We then compute the confidence $\Pr(\tilde{X} \mid \theta)$ for each pass, and the final estimate is obtained by averaging over n passes:

$$\Pr(X \mid \theta) = \frac{1}{N} \sum_{i=1}^{N} \Pr(\tilde{X}_i \mid \theta). \tag{8}$$

This method enhances existing MIA techniques by incorporating the mean confidence value, which can be optimized for different attack strategies. When selecting the hyperparameters, we use the following approach: we randomly consider one of the reference models as the target model and use the other reference models to attack it. Then, we find the optimal hyperparameters p and N by grid search to maximize the attack's AUC. Finally, we use the hyperparameters selected from the reference models to attack the original target model.

Some MIA methods also incorporate input noise [Carlini et al., 2022, Zarifzadeh et al., 2024]. However, these methods do not directly estimate the confidence term $\Pr(X \mid \theta)$; they rely on other components of the MIA metric. Our focus here is not to compare various uncertainty estimation methods but rather to evaluate whether input dropout can significantly enhance MIA performance on SNNs. Additionally, by estimating the confidence term with input dropout, we can extend this approach to all attack methods that require confidence estimation.

5 EXPERIMENTS AND RESULTS

The sections describe the detailed outline of our experimental setup and discussion of our results.

5.1 MODEL TRAINING

5.1.1 Hybrid Training via ANN-to-SNN Conversion

In this approach, we first train an ANN and subsequently convert it into SNN using the threshold extraction and fine-tuning method proposed by Bojkovic et al. [2024]. This conversion facilitates efficient SNN training while maintaining high performance, even under low-latency constraints. We consider both CIFAR-10 and CIFAR-100 datasets [Krizhevsky et al., 2009], where CIFAR-10 represents a relatively simpler classification task, whereas CIFAR-100 presents a more challenging scenario due to its increased class diversity. To ensure a comprehensive evaluation, we train ResNet18 [He et al., 2016] on both the datasets.

All models are trained for 200 epochs. For CIFAR-10, we initialize the learning rate at 0.1, while for CIFAR-100, we use 0.02. Optimization is performed using the stochastic gradient descent (SGD) algorithm with a weight decay of 5×10^{-4} and a momentum parameter of 0.1. Once the ANN training is complete, we convert the models into SNNs. Following Bojkovic et al. [2024], we begin by training the converted SNN models at $T = 1$ for 50 epochs using surrogate

Table 2: MIA Results with SNN (left) and ANN (right) ResNet18 on CIFAR-10 and CIFAR-100.

	Drop Out	**Attack**	**SNN (T=1)**			**SNN (T=2)**			**SNN (T=4)**		
			AUC	0.1%	1%	AUC	0.1%	1%	AUC	0.1%	1%
CIFAR-10	**False**	Attack-P	54.58	0.08	0.87	54.90	0.00	0.72	55.34	0.00	0.00
		Attack-R	57.59	0.00	0.00	58.22	0.00	0.00	59.14	0.00	0.00
		RMIA	59.89	0.84	3.82	60.83	1.00	4.43	62.61	1.09	5.01
	True	Attack-P	54.64	0.06	0.84	54.75	0.05	0.83	55.08	0.00	0.70
		Attack-R	59.54	0.00	0.00	59.99	0.00	0.00	60.21	0.00	0.00
		RMIA	**63.84**	**1.86**	**6.34**	**64.28**	**2.03**	**6.23**	**64.65**	**1.94**	**6.61**
CIFAR-100	**False**	Attack-P	59.47	0.14	1.24	61.67	0.12	1.13	64.85	0.12	1.13
		Attack-R	66.25	0.00	0.00	69.58	0.00	0.00	73.23	0.00	0.00
		RMIA	69.06	1.11	6.29	72.81	1.87	8.24	77.57	2.61	12.01
	True	Attack-P	59.70	0.14	1.20	61.85	0.13	1.30	64.82	0.12	1.21
		Attack-R	71.69	0.00	0.00	73.64	0.00	0.00	75.47	0.00	0.00
		RMIA	**75.82**	**2.80**	**11.14**	**78.31**	**3.55**	**13.06**	**80.76**	**4.36**	**16.16**

	Attack	**ANN**		
		AUC	0.1%	1%
CIFAR-10	Attack-P	56.61	0.00	0.00
	Attack-R	59.49	0.00	0.00
	RMIA	**63.84**	**2.84**	**5.82**
CIFAR-100	Attack-P	68.75	0.10	1.22
	Attack-R	76.93	0.00	0.00
	RMIA	**82.51**	**6.75**	**20.25**

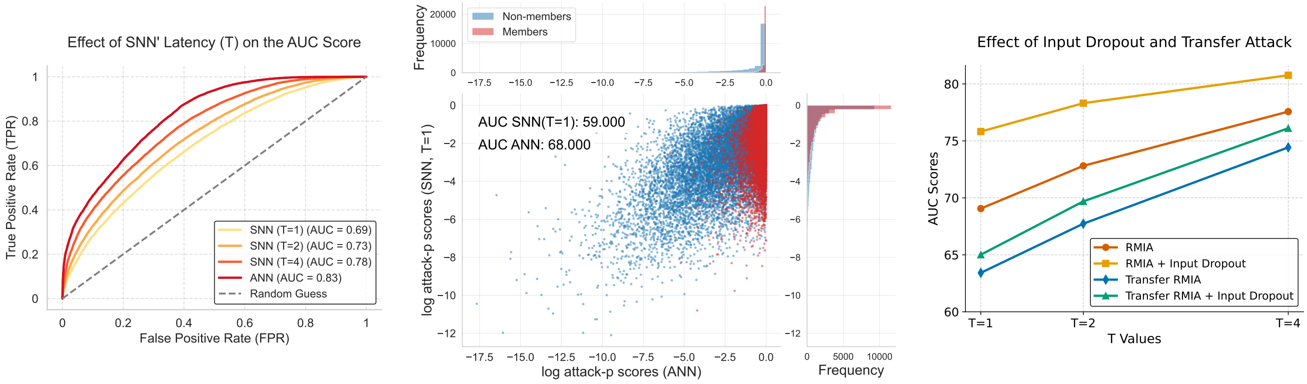

Figure 2: **Left**: ROC curves and AUC values for all-latency SNNs and an ANN under RMIA on CIFAR-100. **Middle**: A scatter plot showing the log attack p-scores and distributions for member and non-member data on SNN (T=1) and ANN for CIFAR-100. **Right**: RMIA performance for all-latency SNNs under different conditions: with or without input dropout, and using or not using a transfer model (ANN).

gradient learning (with a learning rate of 5×10^{-3}). Higher latency SNN models ($T = 2, 3, 4$) are then obtained through sequential training, where the weights of the SNN model at latency T are initialized from the previously trained SNN at $T - 1$. Each additional latency level is trained for 30 epochs. Table 6 shows the test accuracy of trained models.

Table 3: AUC results for the SNN model (left) and ANN model (right) on CIFAR-10.

Dropout	**Attack**	**T=1**	**T=2**	**T=4**
False	Attack-P	55.13	55.57	55.95
	Attack-R	58.33	59.90	60.25
	RMIA	61.44	61.97	63.06
True	Attack-P	55.98	56.01	56.22
	Attack-R	60.97	60.73	61.07
	RMIA	**64.79**	**65.12**	**65.36**

Attack	**AUC**
Attack-P	56.61
Attack-R	59.49
RMIA	**63.84**

To demonstrate the generalizability of our method beyond ResNet, we also provide a brief set of experimental results using both Spiking Neural Networks (SNNs) and Artificial Neural Networks (ANNs). Table 3 reports the AUC scores for an SNN and ANN model based on VGG-16 architecture on the CIFAR-10 dataset, highlighting the impact of incorporating dropout and varying the number of time steps.

5.1.2 Direct SNN Training

To validate the findings obtained from our hybrid training approach, we conduct an additional set of experiments using directly trained SNNs, following the methodology proposed in Mukhoty et al. [2023]. For this experiment also, we focus on the CIFAR-10 and CIFAR-100 dataset and train a ResNet18 SNN model. The model is trained for 250 epochs using the Adam optimizer with a learning rate of 0.001. We evaluate the SNN at $T = 4$ to assess its impact on privacy risks. Unlike the sequential training strategy employed in

Table 4: MIA results on SNNs with varying latencies (T=1,2,4) using ANNs as reference models, with and without dropout

Data	**Attack**	**Dropout = False**									**Dropout = True**								
		SNN(T=1)			**SNN(T=2)**			**SNN(T=4)**			**SNN(T=1)**			**SNN(T=2)**			**SNN(T=4)**		
		AUC	0.1%	1%	AUC	0.1%	1%	AUC	0.1%	1%	AUC	0.1%	1%	AUC	0.1%	1%	AUC	0.1%	1%
CIFAR 10	**Attack-R**	53.26	0.00	1.74	53.88	0.00	0.49	55.04	0.00	0.00	53.31	0.25	1.56	54.10	0.00	0.50	55.50	0.00	0.00
	RMIA	58.52	0.69	3.40	59.68	**0.81**	3.86	61.32	**1.05**	4.55	**59.32**	**0.74**	**3.77**	**60.52**	0.76	**4.34**	**62.29**	0.89	**4.98**
CIFAR 100	**Attack-R**	58.99	0.00	0.80	62.71	0.00	1.89	67.98	0.00	0.00	59.41	0.00	0.49	63.31	0.00	1.40	68.58	0.00	0.00
	RMIA	63.42	**0.62**	3.85	67.73	**1.02**	**6.29**	74.43	**1.37**	9.65	**65.01**	0.38	**3.89**	**69.69**	0.72	6.13	**76.11**	1.10	**10.39**

Table 5: Directly trained SNN Attack Results (T=4).

Dataset	**Dropout**	**Attack**	AUC	0.1%	1%
CIFAR-10	**False**	Attack-P	59.76	0.05	1.05
		Attack-R	63.53	0.00	0.00
		RMIA	63.85	1.04	5.76
	True	Attack-P	58.87	0.06	0.96
		Attack-R	64.64	0.00	0.00
		RMIA	**67.58**	**3.56**	**8.86**
CIFAR-100	**False**	Attack-P	56.54	0.00	0.74
		Attack-R	56.15	0.00	0.00
		RMIA	59.62	0.44	2.39
	True	Attack-P	56.58	0.00	1.06
		Attack-R	57.65	0.00	0.00
		RMIA	**62.11**	**0.62**	**3.13**

the hybrid approach, direct SNN training allows for parallel training of models at different latencies, as each latency level is independently optimized. To ensure fair comparison, we apply the same data preprocessing and transformations as in the hybrid training setup. All training and experiments are conducted on NVIDIA A100 GPUs.

Table 6: ResNet18 classification test accuracy (%) for ANN-to-SNN converted models and directly trained SNNs.

	Hybrid			**Direct**	**ANN**
Dataset	**SNN (T=1)**	**SNN (T=2)**	**SNN (T=4)**	**SNN (T=4)**	–
CIFAR-10	86.1	87.3	88.2	87.0	**90.9**
CIFAR-100	59.5	62.1	63.9	66.9	**69.5**

5.2 SNN PERFORMANCE AGAINST MIA

Tables 2 and Figure 2 (**Left**) summarize the performance of trained ANNs and their corresponding SNNs ($T = 1, 2, 4$) on CIFAR-10 and CIFAR-100 when subjected to MIAs. Our findings highlight three key insights:

Effectiveness of Attacks: RMIA consistently outperforms both Attack-R and Attack-P, confirming its superiority as a membership inference technique. This aligns with prior studies [Zarifzadeh et al., 2024], reinforcing RMIA's superior privacy threat in ANNs and SNNs.

Impact of Latency on SNN Vulnerability: The privacy risk of SNNs increases with latency (T), across all attack metrics. As the T increases, the SNN behaves similar to an ANN and hence, the difference in membership scores of member and non-member data widens.

ANNs vs. SNNs in MIA Susceptibility: ANNs consistently exhibit the highest attack metric scores, indicating greater susceptibility to MIAs compared to SNNs. This suggests that SNNs offer an intrinsic advantage in privacy protection, particularly at lower latencies.

Comparing directly trained SNNs: Table 5 highlights the MIA threat to directly trained SNNs (T=4), showing AUC trends similar to hybrid training. This confirms that SNN vulnerability to MIA is independent of the training method.

The resilience of SNNs at low T stems from their discrete processing. Unlike ANNs, which use continuous values, SNNs encode information as spike trains over time steps. When $T = 1$, each neuron outputs a single binary value, significantly limiting expressiveness and leading to overlapping confidence distributions for member and non-member data. This overlap reduces the attacker's ability to distinguish between them, thereby mitigating MIA effectiveness (see Fig. 2, **Middle**). However, as T increases, SNNs accumulate more information over time, improving expressiveness and decision boundary separation (Fig. 3), which in turn makes them more vulnerable to MIAs. This reveals an accuracy-privacy trade-off: lower T enhances privacy but slightly reduces accuracy, while higher T improves accuracy at the cost of privacy (Table 6).

5.3 ATTACKING SNN USING ANN AS REFERENCE MODELS

SNNs consists of critical hyperparameters, such as latency (T) and membrane threshold, which significantly influence their performance. Unlike ANNs, these parameters are not easily accessible to an adversary, adding an inherent layer of complexity for the attack. Consequently, an attacker may resort to using a conventionally trained ANN with the same architecture as a reference model to infer membership on the target SNN. As expected, our experiments indicate that this approach is less effective (Fig. 2, **Right**).

Although there are architectural similarities, the confidence distributions of ANNs and SNNs exhibit notable differences, particularly in how they separate member and non-member samples (Fig. 2, **Middle**). Despite this, we observe a pattern similar to that of the same-model scenario in Table 4. Furthermore, the trends in attack metrics for RMIA and Attack-R across both datasets closely mirror those in the same-model setting. At lower SNN latencies, the confidence scores for member and non-member data become increasingly indistinguishable compared to ANNs (Fig. 3).

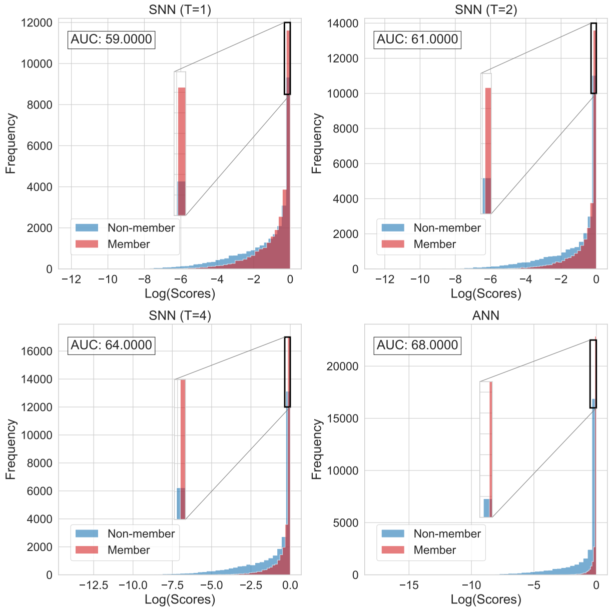

Figure 3: Attack-P(confidence) distribution of member and non-member for all latency SNNs & ANNs in CIFAR-100

5.4 IMPACT OF INPUT DROPOUT ON ATTACK PERFORMANCE

To enhance the attack on SNNs, we apply the input dropout method. For both CIFAR-10 and CIFAR-100, our experiments demonstrate that incorporating input dropout significantly increases the overall attack performance. We observe that the vulnerability of SNNs becomes quite similar to that of ANNs for all attacks as the input dropout strategy is adopted. This result is clearly evident for the most potent RMIA attack on both datasets.

Interestingly, the AUC improvement also occurs even when the adversary uses ANNs as reference models to attack the target SNN model, as shown in Table 4 and Figure 2. This highlights the transferability of our hypothesis regarding SNN vulnerabilities, reinforcing the concern that adversarial threats can persist even without direct knowledge of the model's structure. Furthermore, similar enhancements in attack performance are also evident for directly trained SNNs, particularly for low-latency configurations as shown in Table 5. This shows that the technique adopted to train SNNs is inconsequential to the MIA threat and our input dropout strategy, highlighting the universal susceptibility of SNN to the privacy threat.

6 CONCLUSION

Our results show that SNNs exhibit greater resilience to MIAs than ANNs. However, it becomes increasingly vulnerable as the latency increases. Furthermore, our proposed naive input dropout technique can significantly increase the risk of MIA, even when the adversary lacks knowledge of the SNN's architecture and uses ANN as reference models. This effect also persists in SNNs trained directly. Therefore, we argue that the assumption of SNNs' inherent privacy protection should be critically evaluated, especially for sensitive tasks. Our work also highlights the need for robust techniques to enhance privacy assurance in SNNs.

Acknowledgements

The authors express their sincere gratitude to Velibor Bojkovic, Ruben Solozabal, and Xingyu Qu for their valuable insights and constructive feedback, which greatly contributed to the improvement of this work.

References

Ammar Almomani, Mohammad Alauthman, Mohammed Alweshah, Osama Dorgham, and Firas Albalas. A comparative study on spiking neural network encoding schema: implemented with cloud computing. *Cluster Computing*, 22:419–433, 2019.

Suwhan Baek and Jaewon Lee. Snn and sound: a comprehensive review of spiking neural networks in sound. *Biomedical Engineering Letters*, 14(5):981–991, 2024.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

Velibor Bojkovic, Srinivas Anumasa, Giulia De Masi, Bin Gu, and Huan Xiong. Data driven threshold and potential initialization for spiking neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4771–4779. PMLR, 2024.

Hannah Bos and Dylan Muir. Sub-mw neuromorphic snn audio processing applications with rockpool and xylo.

In *Embedded Artificial Intelligence*, pages 69–78. River Publishers, 2023.

Maxence Bouvier, Alexandre Valentian, Thomas Mesquida, Francois Rummens, Marina Reyboz, Elisa Vianello, and Edith Beigne. Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(2):1–35, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

Haoming Chu, Yulong Yan, Leijing Gan, Hao Jia, Liyu Qian, Yuxiang Huan, Lirong Zheng, and Zhuo Zou. A neuromorphic processing system with spike-driven snn processor for wearable ecg classification. *IEEE Transactions on Biomedical Circuits and Systems*, 16(4):511–523, 2022.

Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=_XNtisL32jv`.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Rida El-Allami, Alberto Marchisio, Muhammad Shafique, and Ihsen Alouani. Securing deep spiking neural networks against adversarial attacks through inherent structural parameters. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 774–779. IEEE, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. Defenses to membership inference attacks: A survey. *ACM Computing Surveys*, 56(4):1–34, 2023.

Somayeh Hussaini, Michael Milford, and Tobias Fischer. Applications of spiking neural networks in visual place recognition. *IEEE Transactions on Robotics*, 2024.

Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature machine intelligence*, 1(9):389–399, 2019.

Youngeun Kim, Yeshwanth Venkatesha, and Priyadarshini Panda. Privatesnn: privacy-preserving spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1192–1200, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=eGLdVRvvfQ`. Expert Certification.

Jiaxin Li, Gorka Abad, Stjepan Picek, and Mauro Conti. Membership privacy evaluation in deep spiking neural networks. *arXiv preprint arXiv:2409.19413*, 2024.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI open*, 3:111–132, 2022.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

Qi Liu and Zhixuan Zhang. Ultralow power always-on intelligent and connected snn-based system for multimedia iot-enabled applications. *IEEE Internet of Things Journal*, 9(17):15570–15577, 2022.

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12444–12453, 2022.

Mishal Fatima Minhas, Rachmad Vidya Wicaksana Putra, Falah Awwad, Osman Hasan, and Muhammad Shafique. Continual learning with neuromorphic computing: Theories, methods, and applications. *arXiv preprint arXiv:2410.09218*, 2024.

Bhaskar Mukhoty, Velibor Bojkovic, William de Vazelhes, Xiaohan Zhao, Giulia De Masi, Huan Xiong, and Bin Gu. Direct training of snn using local zeroth order method. *Advances in Neural Information Processing Systems*, 36: 18994–19014, 2023.

Karthikeyan Nagarajan, Junde Li, Sina Sayyah Ensan, Mohammad Nasim Imtiaz Khan, Sachhidh Kannan, and Swaroop Ghosh. Analysis of power-oriented fault injection attacks on spiking neural networks. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 861–866. IEEE, 2022.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Andre T Nguyen, Fred Lu, Gary Lopez Munoz, Edward Raff, Charles Nicholas, and James Holt. Out of distribution data detection using dropout bayesian neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022.

Osamu Nomura, Yusuke Sakemi, Takeo Hosomi, and Takashi Morie. Robustness of spiking neural networks based on time-to-first-spike encoding against adversarial attacks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(9):3640–3644, 2022.

Joao D Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S Cardoso. Spiking neural networks: A survey. *IEEE Access*, 10:60738–60764, 2022.

Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=H1xwNhNFDH`.

Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4):1–34, 2023.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*, San Diego, CA, USA, February 2019. The Internet Society. doi: 10.14722/ndss.2019.23119.

Saima Sharmin, Priyadarshini Panda, Syed Shakib Sarwar, Chankyu Lee, Wachirawit Ponghiran, and Kaushik Roy. A comprehensive analysis on adversarial robustness of spiking neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Matei-Ioan Stan and Oliver Rhodes. Learning long sequences in spiking neural networks. *Scientific Reports*, 14(1):21957, 2024.

Tao Sun, Bojian Yin, and Sander Bohté. Efficient uncertainty estimation in spiking neural networks via mc-dropout. In *International Conference on Artificial Neural Networks*, pages 393–406. Springer, 2023.

Tehreem Syed, Vijay Kakani, Xuenan Cui, and Hakil Kim. Exploring optimized spiking neural network architectures for classification tasks on embedded platforms. *Sensors*, 21(9):3240, 2021.

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural networks*, 111:47–63, 2019.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Huan Wang, Yan-Fu Li, and Konstantinos Gryllias. Brain-inspired spiking neural networks for industrial fault diagnosis: A survey, challenges, and opportunities. *arXiv preprint arXiv:2401.02429*, 2023a.

Kuanchuan Wang, Xinyu Hao, Jiang Wang, and Bin Deng. Comparison and selection of spike encoding algorithms for snn on fpga. *IEEE Transactions on Biomedical Circuits and Systems*, 17(1):129–141, 2023b.

Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain Sciences*, 12(7):863, 2022.

Zhanglu Yan, Zhenyu Bai, and Weng-Fai Wong. Reconsidering the energy efficiency of spiking neural networks. *arXiv preprint arXiv:2409.08290*, 2024.

Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Forty-first International Conference on Machine Learning*, 2024.

Deming Zhou, Weitong Chen, Kongyang Chen, and Bing Mi. Fast and accurate snn model strengthening for industrial applications. *Electronics*, 12(18):3845, 2023.

On the Privacy Risks of Spiking Neural Networks: A Membership Inference Analysis (Supplementary Material)

Junyi Guan[*†1] **Abhijith Sharma**[*†1] **Chong Tian**[1] **Salem Lahlou**[1]

[1]Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.

A PROOFS

Proposition 1: *Let S_1 and S_2 be two MIA scoring functions:*

$$S_1(x) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\Big(\Pr(x \mid \theta) \geq \Pr(z_j \mid \theta)\Big), \tag{9}$$

$$S_2(x) = \Pr(x \mid \theta), \tag{10}$$

where $\{z_j\}_{j=1}^{N}$ is a fixed set of samples. Suppose $x_1, x_2 \in D$. Then, S_1 and S_2 are equivalent, meaning that, $S_2(x_1) > S_2(x_2)$, if and only if $S_1(x_1) > S_1(x_2)$, and their resulting ROC curves are similar.

Proof. Assume $S_2(x_1) > S_2(x_2)$, meaning

$$\Pr(x_1 \mid \theta) > \Pr(x_2 \mid \theta). \tag{11}$$

Define the indicator set:

$$A_x = \{z_j \mid \Pr(x \mid \theta) \geq \Pr(z_j \mid \theta)\}. \tag{12}$$

Since $\Pr(x_1 \mid \theta) > \Pr(x_2 \mid \theta)$, we have $A_{x_2} \subset A_{x_1}$. Hence,

$$S_1(x_1) = \frac{|A_{x_1}|}{N} > \frac{|A_{x_2}|}{N} = S_1(x_2). \tag{13}$$

This establishes the forward order-preserving property.

Reverse direction: Assume $S_1(x_1) > S_1(x_2)$. This implies

$$\frac{|A_{x_1}|}{N} > \frac{|A_{x_2}|}{N}. \tag{14}$$

Since A_x is defined based on comparisons with $\Pr(x \mid \theta)$, if $S_1(x_1) > S_1(x_2)$, then x_1 ranks strictly higher than x_2 in terms of how often it exceeds reference points z_j. The only way for this to happen is if

$$\Pr(x_1 \mid \theta) > \Pr(x_2 \mid \theta), \tag{15}$$

which implies $S_2(x_1) > S_2(x_2)$. Since both directions hold, we conclude

$$S_2(x_1) > S_2(x_2) \iff S_1(x_1) > S_1(x_2). \tag{16}$$

Hence the S_1 and S_2 are equivalent. By Lemma 1, the ROC for them are the same □

[*] Equal Contribution

[†] Correspondence to: junyi.guan@mbzuai.ac.ae, abhijith.sharma@mbzuai.ac.ae

[*] Equal Contribution

[†] Correspondence to: junyi.guan@mbzuai.ac.ae, abhijith.sharma@mbzuai.ac.ae

Lemma 1. *Let S_1 and S_2 be two equivalent scoring functions, meaning that they satisfy, for every x_1, x_2:*

$$S_2(x_1) > S_2(x_2) \iff S_1(x_1) > S_1(x_2). \tag{17}$$

Then the ROC curves for S_1 and S_2 on the same dataset are identical.

Proof. Since S_1 and S_2 preserve the same ranking of data points, their ordering in terms of classification thresholds remains unchanged. That is, for any threshold t applied to S_2, there exists a corresponding threshold s applied to S_1 such that the classification outcomes remain identical:

$$\hat{y}_i = \begin{cases} 1, & S_2(x_i) \geq t \\ 0, & S_2(x_i) < t \end{cases} \iff \hat{y}_i = \begin{cases} 1, & S_1(x_i) \geq s \\ 0, & S_1(x_i) < s. \end{cases} \tag{18}$$

Since both functions yield the same classification outcomes for all possible threshold values, they result in the same TPR and FPR values at each threshold:

$$\text{TPR}_{S_1}(s) = \text{TPR}_{S_2}(t), \quad \text{FPR}_{S_1}(s) = \text{FPR}_{S_2}(t). \tag{19}$$

Since the ROC curve is defined as the parametric plot of $(\text{FPR}(t), \text{TPR}(t))$, it follows that:

$$\text{ROC}_{S_1} = \text{ROC}_{S_2}. \tag{20}$$

□