electronics

Article

Towards Privacy-Preserving Federated Neuromorphic Learning via Spiking Neuron Models

Bing Han †, Qiang Fu *,† and Xinliang Zhang †

China National Institute of Standardization, Beijing 100191, China; hanb@cnis.ac.cn (B.H.); zhangxinliang@cnis.ac.cn (X.Z.)
* Correspondence: fuqiang@cnis.ac.cn
† These authors contributed equally to this work.

Abstract: Federated learning (FL) has been broadly adopted in both academia and industry in recent years. As a bridge to connect the so-called "data islands", FL has contributed greatly to promoting data utilization. In particular, FL enables disjoint entities to cooperatively train a shared model, while protecting each participant's data privacy. However, current FL frameworks cannot offer privacy protection and reduce the computation overhead at the same time. Therefore, its implementation in practical scenarios, such as edge computing, is limited. In this paper, we propose a novel FL framework with spiking neuron models and differential privacy, which simultaneously provides theoretically guaranteed privacy protection and achieves low energy consumption. We model the local forward propagation process in a discrete way similar to nerve signal travel in the human brain. Since neurons only fire when the accumulated membrane potential exceeds a threshold, spiking neuron models require significantly lower energy compared to traditional neural networks. In addition, to protect sensitive information in model gradients, we add differently private noise in both the local training phase and server aggregation phase. Empirical evaluation results show that our proposal can effectively reduce the accuracy of membership inference attacks and property inference attacks, while maintaining a relatively low energy cost. blueFor example, the attack accuracy of a membership inference attack drops to 43% in some scenarios. As a result, our proposed FL framework can work well in large-scale cross-device learning scenarios.

Keywords: federated learning; privacy protection; spiking neural networks

Citation: Han, B.; Fu, Q.; Zhang, X. Towards Privacy-Preserving Federated Neuromorphic Learning via Spiking Neuron Models. *Electronics* **2023**, *12*, 3984. https://doi.org/10.3390/electronics12183984

Academic Editor: Dimitra I. Kaklamani

Received: 3 September 2023
Revised: 18 September 2023
Accepted: 19 September 2023
Published: 21 September 2023

1. Introduction

Multiparty machine learning paradigms are gaining increasing attention recently, as they power many data-driven applications while preserving data privacy [1–4]. Among the many algorithms, a widely adopted paradigm is FL, where multiple participants (or clients) jointly train a shared global model under the coordination of a central server. Since training datasets from each client never leave their holders, a privacy guarantee can be provided in FL. To date, instead of traditional centralized training methods, FL has been broadly implemented in a range of fields, including automatic driving, speech recognition, intelligent medical diagnoses, etc.

However, there remain several challenges to be addressed in FL [5–9]. In this paper, we tackle the problem of privacy protection and computing the energy cost. Many recent works have proven that FL is vulnerable to various kinds of privacy attacks [10,11]. For example, given the local update gradients (or local models), a malicious server can infer whether a specific data point belongs to the training set of a particular client [12]. Moreover, an attacker in the open public environment can precisely reconstruct small batches of training samples [13–15], if the updated gradients are intercepted and the local model architectures are known. Such attacks pose a server privacy threat to FL systems. As a response, many defense methods have been proposed, such as leveraging cryptographic techniques [16,17] or adding differential noise to model gradients [18].

Although these solutions provide effective or provable defenses to FL, they cannot be widely adopted in resource-constrained training scenarios, especially in large-scale and highly distributed FL, e.g., edge computing with massive amounts of unreliable devices. Since current privacy protection techniques incur extra computing overheads and affect model performance [19], data service providers as well as individual users are reluctant to adopt these techniques to ensure data privacy. On the other hand, energy consumption has also become one of the major bottlenecks in promoting FL in Internet of Things (IOT) [20]. Nevertheless, current researchers merely focus on constructing new training protocols to limit resource usage and hardly extend their attention to modifying local models in an energy-efficient manner.

To this end, in this paper, we propose a novel FL framework that offers both theoretically guaranteed privacy protection and low energy consumption. Our proposed framework models the local training in a discrete manner, which is similar to how nerve signals travel in the human brain. Specifically, we leverage spiking neural networks, also known as the third-generation neural network, and differential privacy, a promising technique to protect data privacy, to construct a new local training and server aggregation algorithm. The crux of our framework design lies in how to properly inject differentially private noise into the discontinuous forward propagation process, while maintaining the availability of gradients produced by backpropagation. Our evaluation results demonstrate that our proposal can effectively defend against two common privacy attacks, i.e., membership inference attacks and property inference attacks.

The contributions made in this paper can be summarized as follows.

- We leverage an SNN to tackle the conflict between privacy and efficiency in FL. blue In particular, SNNs enjoy naturally high computing efficiency, with a mathematically traceable computing process. Hence, differential private noise can be added to the training process, while maintaining acceptable model accuracy. Compared to previous works, our proposed framework significantly lowers the hardware requirements for clients.
- To the best of our knowledge, we propose the first FL framework that enjoys both low energy consumption and theoretically guaranteed privacy protection. Our framework protects both the local model and global model by injecting DP noise into the training and transmission process.
- We conduct extensive experiments to evaluate our proposed framework. Empirical evaluation results show that our proposal can effectively defend against common privacy attacks. Moreover, our training scheme can be implemented in large-scale cross-device training, and it does not incur a notable accuracy drop compared to typical FL paradigms with privacy protection.

This paper is organized as follows: Section 2 presents the preliminaries and background of our work, Section 3 formalizes the problem and threat models, Section 4 describes our proposed method, Section 5 shows our empirical evaluation results, Section 6 discusses previous works related to ours, and Section 7 concludes our work.

2. Preliminaries and Background

In this section, we introduce the related preliminaries and background of our work.

2.1. Federated Learning

FL [21] allows multiple participants to cooperatively train a shared global model, while maintaining the confidentiality of their privately held local datasets. Assume that each participant in FL $c_i \in \mathcal{C}$ privately holds a dataset $\mathcal{D}_i \in \mathcal{D}$. Take surprised learning, for example: local dataset $\mathcal{D}_i$ consists of multiple data points d_i, which are characterized by a sample x_i and its corresponding label y_i, i.e., $\{x_i, y_i\} = d_i$. In the canonical FL paradigm,

each client locally trains a local model $\mathcal{F}_i(.)$ on $\mathcal{D}_i$ via minimizing a loss function $\mathcal{L}(.)$. This optimization process is typically achieved by gradient decent algorithms. Mathematically,

$$\theta_i^{p+1} \leftarrow \frac{\partial\, \mathcal{L}(D_i|\theta_i^p)}{\partial\, \theta_i^p}, \tag{1}$$

where p and θ denote the training epoch (or communication round in FL) and weights of local models, respectively. The loss function $\mathcal{L}(.)$ could be predefined before training or dynamically adjusted in the training process, based on practical training goals.

After each client finishes their local training, the central server collects local models from them. A new global model for the next training epoch $\mathcal{F}(.)$ is then created by averaging each local model, i.e.,

$$\mathcal{F}^{p+1}(.) \leftarrow \frac{1}{I}\sum_{c_i}\theta_{c_i}^p. \tag{2}$$

The new global model would be broadcasted to all clients, on which the next local training is based. The local model of each client could be a classical machine learning model or neural network. In this paper, we propose using spiking neuron models, also known as spiking neural networks (SNN), as local models of each client. The authors in [21] have explored the possibility of implementing SNNs in federated learning.

2.2. Spiking Neuron Models

Neuromorphic computing has gained considerable attention in recent years, as it requires lower energy compared to traditional neural networks. A basic model in neuromorphic computing is a spiking neural network (SNN), which is inspired by certain biological principles of the human brain. Different from current artificial neural networks (ANNs), SNNs model the forward propagation process in a discrete manner and are considered to be the third-generation neural network. Specifically, each neuron in the SNN accumulates the incoming spikes and generates a spike when its membrane potential exceeds a threshold. After firing, the membrane potential would be reset to the resting potential. Mathematically, at each time step t, for the i-th neuron, such a process can be modeled as

$$u_i^t = \lambda u_i^{t-1} + \sum_{j\in N} w_{ij}o_j^{t-1} \quad s.t. \quad o_j^{t-1} = \begin{cases} 1 & if\ \ u_i^{t-1} > v, \\ 0 & otherwise, \end{cases} \tag{3}$$

where u, o, and w_{ij} denote its membrane potential, the output of the previous neuron, and a weight between the i-th and j-th neurons, respectively. $\lambda < 1$ is a constant, indicating the reduction in membrane potential at every time step.

An SNN is typically converted from a well-trained ANN. However, such conversion is time-consuming and the model accuracy cannot be properly persevered. Recent research [22] has proposed Batch Normalization Through Time (BNTT), a novel training method that associates a local learning parameter with each time step. As a result, it is possible to train an SNN without an auxiliary fully trained ANN model. In particular, after the BNTT layer is applied, the forward propagation is modeled as

$$u_i^t = \lambda u_i^{t-1} + \mathrm{BNTT}_{\gamma_i^t}\left(\sum_{j\in N} w_{ij}o_j^t\right) = \lambda u_i^{t-1} + \gamma_i^t\left(\frac{\sum_{j\ N} w_{ij}o_j^t - \mu_i^t}{\sqrt{(\sigma_i^t)^2 + \tau}}\right), \tag{4}$$

where u and σ denote the mean and variance from a batch of samples, and τ is a small constant to ensure numerical stability. A parameter γ can therefore be learnt using backpropagation.

2.3. Differential Privacy

Differential privacy (DP) offers theoretically guaranteed privacy protection to confidential data, while not incurring a significant additional computing cost. The DP-SGD algorithm [23] is the first solution to implement DP in machine learning paradigms, which adds perturbing noise to the stochastic gradient descent. DP-SGD only modifies the training algorithm, not adding noise the dataset itself. As a result, the gradient in the model update (in the case of FL, local models) can be published without privacy leakages. In particular, we have the following definitions.

Definition 1 ([24]). *A randomized mechanism $\mathcal{M}$ with domain D and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if, for any two adjacent inputs $d \in D$, and for any subset of outputs $\mathcal{S} \in \mathcal{R}$, it holds that*

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} \Pr\left[\mathcal{M}\left(d'\right) \in S\right]. \tag{5}$$

Under this definition, each output of DP algorithm $\mathcal{M}$ is equally likely on two adjacent databases. The privacy budget, ϵ, controls the amount of the difference between d and d'. A smaller ϵ can provide a stronger privacy guarantee of $\mathcal{M}$. To make the effectiveness of DP measurable, we introduce Renyi Differential Privacy as follows.

Definition 2 ([25]). *A randomized mechanism $\mathcal{M}(d)$ is said to be (α, ϵ)-Renyi differentially private if its distribution over two adjacent inputs d and d' satisfies*

$$D_{\alpha}\left(\mathcal{M}(d) \| \mathcal{M}\left(d'\right)\right) \leq \epsilon, \tag{6}$$

where α tunes the amount of concern placed on unlikely large values of $c(o; \mathcal{M}, d, d')$ versus the average value of $c(o; \mathcal{M}, d, d')$. Here, $c(o; \mathcal{M}, d, d')$ denotes the the privacy loss at an outcome o, which is defined as

$$c\left(o; \mathcal{M}, d, d'\right) = \log \frac{\Pr[\mathcal{M}(d) = o]}{\Pr[\mathcal{M}(d') = o]}. \tag{7}$$

3. Problem Statement and Threat Model

In this paper, our objective is to protect the confidentiality of each participant's local training data. In particular, we consider the following two types of data leakages.

- Membership Leakage. Membership privacy [26,27] concerns indicating whether a specific training sample was involved in the training set. In a membership inference attack (MIA) [28], an attacker manages to obtain black-box access to the target model and query multiple times with its local samples D' to infer which part of D' belongs to the confidential training set. MIA is critical in FL, since each participant as well as the central server can access the shared global model. Moreover, recent researchers [12,28] have proposed more effective MIA pipelines for FL, which significantly increases the risk of privacy leakage.
- Class Leakage. Class leakage exists when the attacker manages to obtain the class distribution of the target dataset. For example, a property inference attack (PIA) [29,30] infers the properties of the training data that are irrelevant to the learning objective. Such properties typically include the proportion of each type of sample in the training set. Similar to MIA, in FL, the shared global model would also cause severe class leakage problems. In particular, with the assistance of a data poisoning attack, an existing attack [30] achieved high accuracy in inferring the general properties of each participant's confidential training dataset.

In FL, since each entity (including the central server) could potentially become compromised, we mainly presume participants to be honest-but-couriers. In other words, each entity would strictly follow the predefined protocol, while trying to infer secret information as much as possible. Therefore, our proposal provides a privacy guarantee in the FL scenario where one or all of the following unexpected conditions are applied.

- (T1) The central server becomes malicious and actively infers confidential information from the received local models. Since the central server has full visibility to all local models, this is the most threatening scenario. Moreover, in order to further extract secret information, the central server could send carefully designed fake global models to a victim participant and infer secret information from their responses.
- (T2) A third party manages to intercept but cannot tamper with the communication between the central server and participants. Likewise, they could actively infer confidential information from updated local models. Different from the malicious central server, the global model is dependable.
- (T3) A number of participants become compromised and cooperatively infer a training set that they do not possess from each training round's global model.

To eliminate these privacy threats, our proposal leverages DP in both local training and server aggregation. Both local models and global models are protected by DP noise, which makes a compromised entity less able to infer sensitive information from the local models and global models.

4. Proposed Method

In this section, we describe our proposed privacy-preserving federated learning by spiking neuron models. A federated learning scheme can generally be divided into two iterative phases: (1) a local search by each participant, denoted by $\mathcal{S}(.)$, and (2) server aggregation, denoted by $\mathcal{A}(.)$. In a communication round, each client firstly performs a local search on their private dataset D, i.e., $\mathcal{F}_i(.) \leftarrow \mathcal{S}(D)$. The trained model $\mathcal{F}(.)$ would be uploaded to a central server, where model aggregation is performed, i.e., $\mathcal{F}(.) \leftarrow \mathcal{A}(\mathcal{F}_1(.) \ldots, \mathcal{F}_i(.), \ldots, \mathcal{F}_c(.))$. In this paper, we design novel local search and server aggregation algorithms for SNNs that provide a privacy guarantee. Figure 1illustrates our proposal at a high level. In the following paragraph, we separately describe our proposed two algorithms.

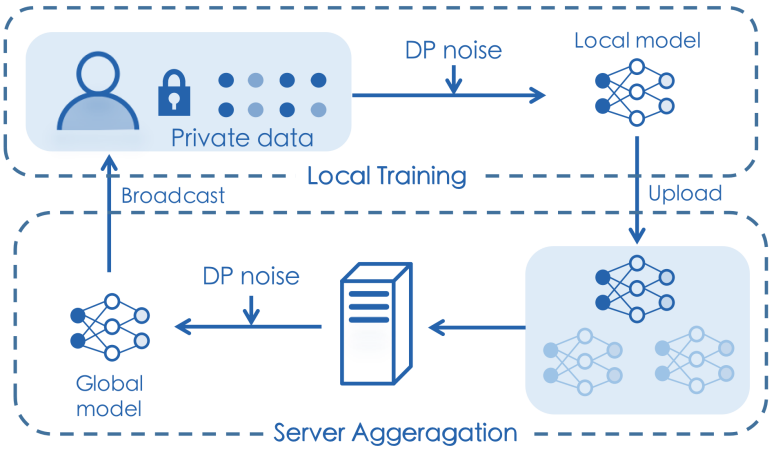

Figure 1. Our proposed training framework at a high level.

4.1. Local Search

In a local search, each client updates their local model by minimizing a loss function $\mathcal{L}(.)$. A widely used loss function is the cross-entropy loss, defined as

$$\mathcal{L}(.) = -\sum_i P_A(x_i) log(P_B(x_i)), \tag{8}$$

where P_A and P_B denote two probability distributions. From the accumulated membrane potential, the cross-entropy loss for SNNs can be defined as

$$\mathcal{L}(.) = -\sum_i y_i \log\left(\frac{e^{u_i^T}}{\sum_k e^{u_k^T}}\right). \tag{9}$$

After modeling the discrete propagation process in a continuous manner, i.e., applying surrogate gradient, $\frac{do}{du} = e^{2(u-u_{th})^2}$, gradient descent can be adopted to compute the model updates at each time step. According to the chain rule, the gradient of the corresponding time and layer can be computed as

$$\nabla_w \mathcal{L}(.) = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial o_i^t} \frac{\partial o_i^t}{\partial u_i^t} \frac{\partial u_i^t}{\partial w_{ij}}. \tag{10}$$

To protect training data privacy, we add Gaussian noise $\mathcal{N}(.)$ to the gradients computed at each training epoch p. As a result, differentially private gradients can be obtained, which makes the attacker less able to extract sensitive information. We provide a privacy guarantee under the framework of GDP theory [31]. In particular, distinguishing between two adjacent datasets D and D' is harder than distinguishing between two Gaussian distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(\kappa,1)$. Therefore, even if a malicious party intercepts the local update gradient or global model, they cannot directly infer or reconstruct the corresponding training set. Mathematically, in the p-th training epoch, the local model for client c is updated by

$$w_c^{p+1} \leftarrow w_c^p - \eta \frac{1}{\mathcal{B}}\left(\left(\mathcal{N}\left(0, \kappa^2 C^2 \mathbf{I}\right) + \nabla_w \mathcal{L}_c^p(.) / \max\left(1, \frac{\left\|\nabla_w \mathcal{L}_c^p(.)\right\|_2}{C}\right)\right) - \nabla_w \mathcal{L}_c^p(.)\right), \tag{11}$$

where η is the learning rate, C is the gradient norm bound, κ is the noise scale, $\mathcal{B}$ is the batch size, and $\mathcal{N}(0, \kappa^2 C^2 \mathbf{I})$ is a normal Gaussian distribution. To avoid privacy leakage, participants usually upload the gradients of each training epoch, instead of uploading the full local model, i.e., $\Delta w_c^{p+1} = \Delta w_c^{p+1} - w_c^p$. This is equal to uploading the full local model, as the central server can directly compute the average on such received gradients.

4.2. Server Aggregation

The central server aggregates all received local updates and forms a new global model. To ensure that all local models have a sufficient impact on the global model, the central server would usually select a subset $\mathbb{S} \in \mathcal{C}$ from all participants. The selection algorithm, for example, could be solving the following maximization problem [32]:

$$\max_{\mathbb{S}} |\mathbb{S}| \quad \text{s.t.} \quad T_{\text{round}} \geq T_{\text{cs}} + T_{\mathbb{S}}^{\text{d}} + \Theta_{|\mathbb{S}|} + T_{\text{agg}}, \tag{12}$$

where $\mathbb{S}$ is the selected client set, T_{round} is the deadline for each round, T_{cs} is the time required for the client selection step, T_{agg} is the time required for the aggregation steps, $T_{\mathbb{S}}^{\text{d}}$ is the time required for global model distribution, and $\Theta_{|\mathbb{S}|}$ is the time for participants in $\mathbb{S}$ required to update and upload the local models.

After client selection, the central server initiates the aggregation process. Denoting the count of non-zero elements of a set by $|.|$, Equation (2) can be expanded as follows:

$$\mathcal{F}^{p+1}(.) \leftarrow \mathcal{F}^p(.) + \frac{1}{\sum_{c\in\mathbb{S}}\left|D_{c_i}^{p+1}\right|} \sum_{c\in\mathbb{S}} \left|D_{c_i}^{p+1}\right| \Delta w_{c_i}^{p+1}. \tag{13}$$

Here, participants with more data samples are given more weight in aggregation, which helps the global model to converge faster. To enhance the privacy of the global model, i.e., defend against third parties intercepting the broadcasted global model, DP noise can be added before it is published:

$$\mathcal{F}^{p+1}(.) \leftarrow \mathcal{F}^{p+1}(.) + \mathcal{N}\left(0, \kappa'^2 C'^2 \mathbf{I}\right). \tag{14}$$

In particular, to limit the performance penalty incurred by adding DP noise, noise is only applicable to model weights that do not change significantly compared to the last training round. The central server also selects some particular training terminal conditions, which indicate when the FL ends. This also helps to reduce the overfitting of the model. The detailed algorithm of our proposed training scheme is shown in Algorithm 1.

Algorithm 1: Privacy-Preserving Federated Neuromorphic Learning

Input : Total communication round $\mathcal{P}$, client set $\mathcal{C}$, local learning rate η, training terminal conditions ϕ, local training batch size $\mathcal{B}$, client selection algorithm $\mathcal{F}_s(.)$,

Initialize: Model parameter of local models w_i^0.

for $p = 1 : \mathcal{P}$ **do**
  $\mathbb{S} \leftarrow \mathcal{F}_s(\mathcal{C})$
  for *each* $c_i \in \mathbb{S}$ **do in parallel**
    $w_{c_i}^{p+1} \leftarrow$ **Equation (11)**
    $\Delta w_{c_i}^{p+1} = \Delta w_{c_i}^{p+1} - w_{c_i}^{p}$
    upload $\Delta w_{c_i}^{p+1}$ to central server
  $\mathcal{F}^{p+1}(.) \leftarrow \mathcal{F}^{p}(.) + \frac{1}{\sum_{c \in \mathbb{S}} \left| D_{c_i}^{p+1} \right|} \sum_{c \in \mathbb{S}} \left| D_{c_i}^{p+1} \right| \Delta w_{c_i}^{p+1}$
  $\mathcal{F}^{p+1}(.) \leftarrow \mathcal{F}^{p+1}(.) + \mathcal{N}\left(0, \kappa'^2 C'^2 \mathbf{I}\right)$
  if *training terminal conditions* ϕ *are matched* **then**
    return $\mathcal{F}^{p+1}(.)$
return $\mathcal{F}^{p+1}(.)$

5. Experiments

In this section, we present the empirical evaluation results of our proposal. We test the final accuracy of the well-trained global model, as well as its ability in defending different privacy attacks.

5.1. Settings

Our evaluation is conducted on the PyTorch Framework 3.8. All experiments are conducted 5 times, and the average results are reported. Detailed settings are described as follows.

Dataset and Learning Scale. We conduct experiments to evaluate the performance of our FL framework on the MNIST and CIFAR-10 datasets. MNIST consists of 60,000 training samples and 10,000 test samples, which are 28 × 28 grayscale images. CIFAR-10 consists of 60,000 training samples and 10,000 test samples, which are 32 × 32 RGB images. To simulate three practical learning scenarios [33] (cross-silo FL and cross-device FL), the datasets are split according to the following.

1. cross-silo FL: Participants are different organizations (e.g., hospitals or banks) or geo-distributed data centers. Typically, there are 2–100 participants in this scenario, and all of them are reliable, i.e., no failure or drop out. Our simulation of this scenario splits the dataset with the following settings: (1) *CS_1*: 6 clients, with 10,000 training samples each, (2) *CS_2*: 20 clients, with 2000–5000 samples each, and (3) *CS_3*: 50 clients, with 1000–2000 samples each. All local gradients would aggregated by the central server, i.e., client selection is disabled.
2. cross-device FL: Participants are a very large number of massively parallel mobile or IoT devices, typically up to 10^5. During training, only a small of participants are available at any one time, i.e., some of the participants would have a probability not responding to the central server. Our simulation of this scenario splits the dataset with the following settings: (1) *CD_1*: 200 clients, each with 500 training samples randomly selected from the training set, and (2) *CD_2*: 1000 clients, each with 250 training

samples. randomly selected from the training set. The central server only selects 30 local models for aggregation according to its selection protocol.

Model Architectures. We implement two local model architectures: (1) *M1*, a lightweight fully connected neural network with 5 hidden layers; and (2) *M2*, VGG13, which consists of 10 convolutional filters 3×3 and two fully connected layers with pooling. As a comparison, both architectures would be implemented as an ANN and SNN. The training batch size $\mathcal{B}$ for each participant is 32, and the learning rate η is 0.005. In VGG16, we also use a momentum of 0.9 to avoid falling into local optima. The activation function in ANNs is set with Sigmoid. In each communication round, the local model is trained by 10 epochs before updating the gradients on the central server. The DP noise added in the local training phase and aggregation phase is $\mathcal{N}(0, 0.01)$ and $\mathcal{N}(0, 0.005)$, respectively.

5.2. Training Accuracy

We report the training accuracy of the well-trained global model, as well as the communication rounds needed under different settings, in Table 1 (MNIST) and Table 2 (CIFAR-10). In general, the global model can converge well under both FedAvg and our method. Please note that our method provides privacy protection, which would cause a performance penalty. We will present the evaluation results in defending against privacy attacks in the next subsection.

Table 1. The converged global model accuracy (Acc) and communication rounds (CR) in different settings for the MNIST dataset.

		FedAvg M1		**FedAvg M2**		**Ours M1**		**Ours M2**	
		Acc	**CR**	**Acc**	**CR**	**Acc**	**CR**	**Acc**	**CR**
CS_1	SNN	98.9%	126	99.1%	178	97.1%	152	97.2%	219
	ANN	99.2%	97	99.6%	169	97.4%	128	97.8%	143
CS_2	SNN	98.8%	134	98.9%	189	97.0%	180	97.4%	269
	ANN	99.2%	106	99.3%	170	97.2%	144	97.6%	178
CS_3	SNN	98.3%	150	98.5%	195	96.3%	208	96.8%	285
	ANN	99.0%	111	99.3%	182	96.8%	160	97.1%	197
CD_1	SNN	96.5%	213	96.9%	245	94.2%	323	94.8%	329
	ANN	97.8%	178	98.0%	190	94.9%	219	95.5%	230
CD_2	SNN	95.7%	256	95.8%	281	93.8%	348	94.4%	367
	ANN	97.1%	220	97.5%	243	94.0%	245	95.2%	277

Table 2. The converged global model accuracy (Acc) and communication rounds (CR) in different settings for the CIFAR-10 dataset.

		FedAvg M1		**FedAvg M2**		**Ours M1**		**Ours M2**	
		Acc	**CR**	**Acc**	**CR**	**Acc**	**CR**	**Acc**	**CR**
CS_1	SNN	96.2%	186	96.8%	240	95.7%	187	95.9%	199
	ANN	96.6%	170	97.6%	226	96.1%	165	96.3%	186
CS_2	SNN	95.9%	193	96.3%	258	95.2%	196	95.6%	215
	ANN	96.5%	179	97.0%	239	95.5%	173	95.5%	194
CS_3	SNN	95.1%	205	95.9%	262	94.6%	228	94.9%	254
	ANN	95.7%	186	96.5%	245	94.8%	212	94.3%	210
CD_1	SNN	93.2%	240	94.3%	280	94.2%	347	94.6%	363
	ANN	94.1%	210	94.7%	255	93.6%	236	94.7%	274
CD_2	SNN	92.6%	269	93.8%	293	93.3%	366	94.0%	370
	ANN	93.4%	238	94.2%	276	93.4%	250	93.3%	280

From both tables, we can observe that, in both simple and complex network architectures (M1 and M2), our proposed FL scheme can provide a global SNN model with testing accuracy similar to the ANN. In both network architectures, the number of communication rounds for the SNN is 1.5 times more than for the ANN on average, especially in large

training scenarios. Moreover, the decrease in model accuracy compared to the ANN does not vary with the learning scale. It can be estimated that, in even larger learning scenarios with more uncertainties, our proposal can achieve a global model that is similar to an ANN in terms of accuracy, while enjoying the unique benefits of SNNs, e.g., low computing energy cost. From Table 2, we can see that the SNN also performs well on RGB images. Next, we report how the global model accuracy increases with the communication rounds in Figure 2. The tested global model accuracies under the setting of M2 CIFAR-10 are shown. For cross-silo training, the global model accuracy increases faster than for cross-device training.

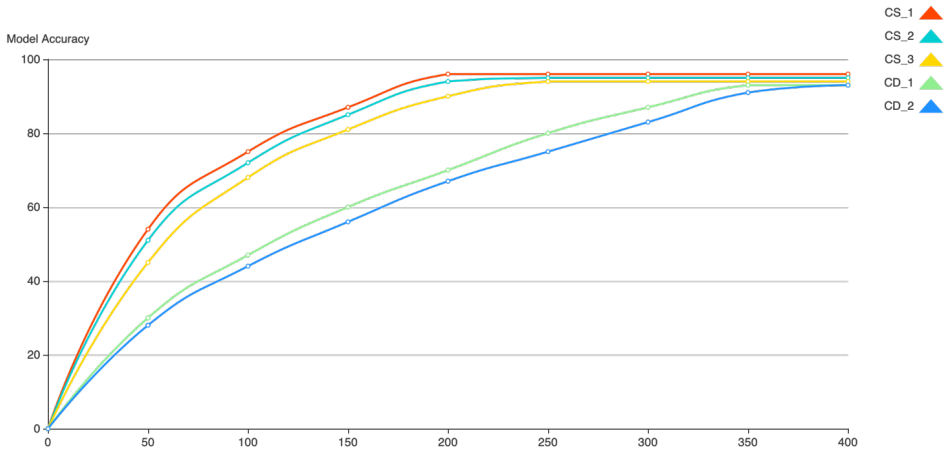

Figure 2. Change in global model accuracy with the increase in communication rounds (CIFAR-10, M2).

5.3. Defense against Privacy Attacks

In this subsection, we present the evaluation results in terms of defending against privacy attacks. As mentioned in Section 3, we mainly consider membership inference attacks and property inference attacks. We posit that (1) the central server or (2) some of the clients are malicious. The MIA follows the implementation of [34], and the attack accuracy is reported in Table 3. The PIA follows the implementation of [35], and the attack accuracy is reported in Table 4. We use the same metric for attack accuracy as in [35].

The MIA accuracy in Table 3 is computed by

$$\mathrm{MA}_{\mathcal{A}} = 2\mathbb{P}[\, \mathcal{F}_{\mathcal{A}}(\mathrm{Y}_t \mid \boldsymbol{aux}\,) = \epsilon\,] - 1, \tag{15}$$

where $\boldsymbol{aux}$ denotes the information available to the attacker (i.e., attacker's advantage), and $\mathcal{F}_{\mathcal{A}}$ is the chosen attack algorithm. Since a randomly initialized attack algorithm achieves 50% accuracy (random guess), this measurement of the performance of MIA would be more rational. From Table 3, we can observe that our proposed FL framework can effectively prevent MIA in different learning scenarios. The attack accuracy drops significantly (65% in average) in simple and complex datasets. In large models, our proposal can provide even better privacy protection.

As for defense against PIA, our training scheme also achieves similar defense performance compared to the results in MIA. From Table 4, we can observe that the attack accuracy decreases considerably with our training scheme. In different settings, the distribution of only a small portion of confidential data can be inferred by the attacker. From all tables, we can conclude that our proposed training scheme can provide strong privacy protection in different threat models, while only causing a slight drop in model performance and convergence speed.

Table 3. MIA accuracy under FedAvg and our proposed training scheme in MNIST (D1) and CIFAR-10 (D2).

MIA	FedAvg M1		Ours M1		FedAvg M2		Ours M2	
	D1	D2	D1	D2	D1	D2	D1	D2
CS_1	67.3%	61.1%	23.0%	26.6%	69.9%	63.5%	23.5%	25.9%
CS_2	65.9%	61.0%	21.9%	26.3%	68.1%	62.9%	23.2%	25.3%
CS_3	65.1%	60.4%	20.8%	29.5%	67.5%	62.3%	22.7%	24.8%
CD_1	52.1%	48.2%	19.6%	18.6%	59.1%	51.2%	17.6%	20.2%
CD_2	49.3%	42.1%	18.5%	17.9%	58.3%	47.0%	17.1%	19.6%

Table 4. PIA accuracy under FedAvg and our proposed training scheme in MNIST (D1) and CIFAR-10 (D2).

PIA	FedAvg M1		Ours M1		FedAvg M2		Ours M2	
	D1	D2	D1	D2	D1	D2	D1	D2
CS_1	78.6%	75.0%	30.3%	28.7%	73.9%	71.8%	27.4%	25.7%
CS_2	77.2%	74.2%	29.4%	26.6%	73.1%	70.9%	26.7%	25.0%
CS_3	76.4%	73.7%	28.7%	26.0%	72.0%	70.1%	26.1%	24.6%
CD_1	72.1%	71.0%	26.9%	24.9%	67.5%	65.7%	24.0%	22.4%
CD_2	71.2%	69.5%	26.1%	24.0%	64.3%	63.1%	22.9%	21.8%

5.4. Ablation Study

We also conduct an ablation study to further verify the effectiveness of our proposal. We test the defense performance when (1) the scale of added noise is changed, and (2) we only enable DP in local training or server aggregation. We select the most challenging scenario in the cross-silo and cross-device context, respectively, i.e., *CS_3* and *CD_2*. The results are reported in Tables 5 and 6, respectively.

In Table 5, we further show the distribution of DP noise added in local training. We evaluate $\mathcal{N}(0,0.02)$ and $\mathcal{N}(0,0.005)$, and the global model accuracy (Acc), communication rounds (CR), and privacy attack accuracy (PA) are reported. PA reflects the model's general ability in defending against privacy attacks and is computed from the average attack accuracy of MIA and PIA. Generally speaking, adding DP noise with a wider range would cause a penalty in both model accuracy and convergence speed, but provide more powerful privacy protection. In practical training, participants can choose the appropriate scale of noise.

Table 5. Results showing how the scale of added noise in local training impacts global model accuracy (Acc), communication rounds (CR), and privacy attack accuracy (PA). PA equals the average attack accuracy of MIA and PIA.

		CS_3 M1		CD_2 M1		CS_3 M2		CD_2 M2	
		D1	D2	D1	D2	D1	D2	D1	D2
$\mathcal{N}(0,0.01)$	Acc	96.3%	94.6%	93.8%	93.3%	96.8%	94.9%	94.4%	94.0%
	CR	208	228	348	366	285	254	367	370
	PA	24.8%	27.8%	22.3%	21.0%	24.4%	24.7%	20.0%	20.7%
$\mathcal{N}(0,0.02)$	Acc	95.6%	93.3%	92.9%	92.0%	95.1%	93.7%	92.6%	91.3%
	CR	213	237	365	378	298	265	388	386
	PA	23.6%	25.7%	20.9%	19.8%	19.7%	18.9%	16.5%	16.0%
$\mathcal{N}(0,0.005)$	Acc	97.0%	95.6%	94.4%	94.6%	97.2%	95.8%	95.7%	95.2%
	CR	196	207	328	332	250	239	354	348
	PA	27.5%	26.7%	23.9%	23.6%	29.8%	30.6%	28.6%	29.5%

Table 6. Results when only applying DP noise in local training phase or server aggregation phase.

		CS_3 M1		CD_2 M1		CS_3 M2		CD_2 M2	
		D1	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**
DP in local training	Acc	95.1%	93.8%	93.2%	93.0%	95.7%	94.4%	93.9%	93.5%
	CR	187	198	306	328	246	289	345	356
	(PA	34.9%	36.6%	31.7%	32.8%	30.9%	31.7%	30.1%	32.6%
DP in server aggregation	Acc	95.3%	94.5%	93.9%	93.4%	95.7%	95.2%	95.0%	93.7%
	CR	181	196	294	310	239	265	310	327
	PA	32.7%	31.9%	31.3%	31.0%	31.8%	32.0%	31.7%	32.2%

Next, we investigate the model's defense capability when only adding DP noise in local training or server aggregation. In Tables 1 and 6, we can observe that applying DP noise in only the local training phase or server aggregation phase can allow resistance to privacy attacks. However, such privacy protection is weaker than when injecting noise in both phases.

6. Related Work

Our work mainly follows two lines of research: federated learning and spiking neuron models. The following paragraph briefly introduces recent advances in these two areas.

6.1. Federated Learning

Federated learning has quickly become one of the gold standards in privacy-preserving machine learning solutions since it was first introduced by Google in 2016 [19]. The pioneering work in FL typically simply considers all clients to be honest to each other, which makes directly aggregating local gradients possible [36]. However, the existence of malicious clients has prompted the proposal of more secure and robust FL frameworks [18].

Specifically, a number of works have designed different kinds of attacks to manipulate the training process or infer privacy data in FL. Zhu et al. proved that an attacker is able to reconstruct the actual training set of each participant if the local update gradient can be accessed [13]. Similarly, a malicious subset of the participants has been proven to be able to poison the global model by sending model updates derived from mislabeled data [37]. Moreover, FL can also easily be backdoored if the attacker manages to continuously send corrupted gradients to the aggregation server [38].

Nevertheless, in regard to these attacks, a wide range of researchers have provided many defense solutions. For example, Moriai et al. consider using homomorphic encryption to ensure the confidentiality of the whole training process [39]. In order to prevent different kinds of data inference attacks, adding differential noise to the training process has also been considered in a number of secure training schemes [10].

6.2. Spiking Neuron Models

The recent increasing need for the autonomy of machines in the real world has promoted the application of SNNs [40,41]. Inspired by biological neural networks, SNNs leverage plasticity and backpropagation as the main training methods [42]. Despite great advantages in energy consumption and learnability, SNNs are still difficult to train in many cases, since there exist complex dynamics of neurons, and the spike operations are naturally non-differentiable. Recent studies of SNNs can broadly be divided into the following streams.

Research stream 1: Investigating and advancing the connection between neural networks and SNNs. For example, Jason K. et al. explored the delicate interplay between encoding data as spikes and the learning process and investigated how deep learning might move towards biologically plausible online learning [43]. Yufei et al. designed an information maximization loss that aims at maximizing the information flow in the SNN, which ensures sufficient model updates at the beginning and accurate gradients at the end of the training [44].

Research stream 2: Boosting the convergence of SNNs. For example, Fang et al. proposed the spike-element-wise (SEW) ResNet to realize residual learning in deep SNNs, which overcomes the vanishing/exploding gradient problems of spiking ResNet [45]. To overcome the challenge of discrete spikes prohibiting gradient calculation, Yuhang et al. proposed a surrogate gradient approach that serves as continuous relaxation [46]. By encoding the spike trains into a spike representation using (weighted) firing rate coding, Qingyan et al. proposed the differentiation on spike representation method, which provides competitive accuracy to ANNs yet with low latency [47].

Research stream 3: Promoting the practical application of SNNs. For example, Xie et al. designed a novel network structure for SNNs based on neuron receptive fields, which extract information from the pixel and spatial dimensions of traffic signs [48]. Moreover, Viale et al. used an SNN connected to an event-based camera to tackle the classification problem between cars and other objects [49].

7. Conclusions and Future Work

In this paper, we propose the first privacy-preserving FL framework with SNNs. Our proposal can simultaneously provide theoretically guaranteed privacy protection and achieve low energy consumption. Our framework injects DP noise into the training and transmission process. We conducted extensive experiments to evaluate our proposed framework. Empirical evaluation results have shown that our proposal can effectively defend against common privacy attacks. blueFor example, the attack accuracy of membership inference attacks drops to 43% in some scenarios. Moreover, our training scheme can be implemented in large-scale cross-device training, and it does not incur a significant accuracy drop compared to typical FL paradigms with privacy protection.

Author Contributions: Conceptualization, B.H. and Q.F.; methodology, B.H.; software, B.H.; validation, B.H., Q.F. and X.Z.; formal analysis, B.H., Q.F. and X.Z.; writing—original draft preparation, B.H. and Q.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Plan Program, grant number 2021YFF0602203; the State Administration for Market Regulation Science and Technology Plan Project, grant number 2022MK187; and the President Funding Key Project of China National Institute of Standardization, grant number 242022Y-9470.

Data Availability Statement: The data can be shared upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
2. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4245–4257. [CrossRef]
3. Ren, H.; Li, H.; Liu, D.; Xu, G.; Shen, X.S. Enabling Secure and Versatile Packet Inspection with Probable Cause Privacy for Outsourced Middlebox. *IEEE Trans. Cloud Comput.* **2022**, *10*, 2580–2594. [CrossRef]
4. Ren, H.; Li, H.; Liu, D.; Xu, G.; Cheng, N.; Shen, X. Privacy-Preserving Efficient Verifiable Deep Packet Inspection for Cloud-Assisted Middlebox. *IEEE Trans. Cloud Comput.* **2022**, *10*, 1052–1064. [CrossRef]
5. Wang, Y.; Chen, K.; Tan, Y.; Huang, S.; Ma, W.; Li, Y. Stealthy and Flexible Trojan in Deep Learning Framework. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 1789–1798. [CrossRef]
6. Wang, Y.; Tan, Y.; Baker, T.; Kumar, N.; Zhang, Q. Deep Fusion: Crafting Transferable Adversarial Examples and Improving Robustness of Industrial Artificial Intelligence of Things. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7480–7488. [CrossRef]
7. Wang, Y.; Wu, S.; Jiang, W.; Hao, S.; Tan, Y.; Zhang, Q. Demiguise Attack: Crafting Invisible Semantic Adversarial Perturbations with Perceptual Similarity. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Montreal, QC, Canada, 19–27 August 2021; pp. 3125–3133. [CrossRef]
8. Pan, Z.; Hu, L.; Tang, W.; Li, J.; He, Y.; Liu, Z. Privacy-Preserving Multi-Granular Federated Neural Architecture Search–A General Framework. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 2975–2986. [CrossRef]
9. Liu, Z.; Hu, C.; Shan, C.; Yan, Z. ADCaDeM: A Novel Method of Calculating Attack Damage Based on Differential Manifolds. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4070–4084. [CrossRef]

10. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
11. Yin, X.; Zhu, Y.; Hu, J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36. [CrossRef]
12. Zhang, J.; Zhang, J.; Chen, J.; Yu, S. Gan enhanced membership inference: A passive local attack in federated learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
13. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 14774–14784.
14. Huang, W.; Sun, M.; Zhu, L.; Oh, S.K.; Pedrycz, W. Deep fuzzy min–max neural network: Analysis and design. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–12. [CrossRef] [PubMed]
15. Huang, W.; Xiao, Y.; Oh, S.K.; Pedrycz, W.; Zhu, L. Random Polynomial Neural Networks: Analysis and Design. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–11. [CrossRef] [PubMed]
16. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. Efficient homomorphic encryption for {Cross-Silo} federated learning. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 20), Virtual, 5–17 July 2020; pp. 493–506.
17. Huang, W.; Zhang, Y.; Wan, S. A sorting fuzzy min-max model in an embedded system for atrial fibrillation detection. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2022**, *18*, 1–18. [CrossRef]
18. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
19. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]
20. Shi, D.; Li, L.; Chen, R.; Prakash, P.; Pan, M.; Fang, Y. Towards energy efficient federated learning over 5g+ mobile devices. *IEEE Wirel. Commun.* **2022**, *29*, 44–51. [CrossRef]
21. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečnỳ, J.; Mazzocchi, S.; McMahan, B.; et al. Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **2019**, *1*, 374–388.
22. Kim, Y.; Panda, P. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Front. Neurosci.* **2020**, *15*, 773954. [CrossRef]
23. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
24. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Theory of Cryptography Conference, New York, NY, USA, 4–7 March 2006; pp. 265–284.
25. Mironov, I. Rényi differential privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–15 August 2017; pp. 263–275.
26. Li, N.; Qardaji, W.; Su, D.; Wu, Y.; Yang, W. Membership privacy: A unifying framework for privacy definitions. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 889–900.
27. Long, Y.; Bindschaedler, V.; Gunter, C.A. Towards measuring membership privacy. *arXiv* **2017**, arXiv:1712.09136.
28. Zari, O.; Xu, C.; Neglia, G. Efficient passive membership inference attack in federated learning. *arXiv* **2021**, arXiv:2111.00430.
29. Wang, Z.; Huang, Y.; Song, M.; Wu, L.; Xue, F.; Ren, K. Poisoning-assisted property inference attack against federated learning. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 3328–3340. [CrossRef]
30. Xu, M.; Li, X. Subject property inference attack in collaborative learning. In Proceedings of the 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 22–23 August 2020; pp. 227–231.
31. Dong, J.; Roth, A.; Su, W.J. Gaussian differential privacy. *arXiv* **2019**, arXiv:1905.02383.
32. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
33. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
34. Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; Backes, M. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv* **2018**, arXiv:1806.01246.
35. Ganju, K.; Wang, Q.; Yang, W.; Gunter, C.A.; Borisov, N. Property inference attacks on fully connected neural networks using permutation invariant representations. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 619–633.
36. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
37. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In Proceedings of the Computer Security—ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, 14–18 September 2020; pp. 480–501.
38. Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.Y.; Lee, K.; Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16070–16084.

39. Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. In Proceedings of the 2019 IEEE 26th Symposium on Computer Arithmetic (ARITH), Kyoto, Japan, 10–12 June 2019; p. 198.
40. Ghosh-Dastidar, S.; Adeli, H. Spiking neural networks. *Int. J. Neural Syst.* **2009**, *19*, 295–308. [CrossRef]
41. Yamazaki, K.; Vo-Ho, V.K.; Bulsara, D.; Le, N. Spiking neural networks and their applications: A Review. *Brain Sci.* **2022**, *12*, 863. [CrossRef]
42. Basu, A.; Deng, L.; Frenkel, C.; Zhang, X. Spiking neural network integrated circuits: A review of trends and future directions. In Proceedings of the 2022 IEEE Custom Integrated Circuits Conference (CICC), Newport Beach, CA, USA, 24–27 April 2022; pp. 1–8.
43. Eshraghian, J.K.; Ward, M.; Neftci, E.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training spiking neural networks using lessons from deep learning. *arXiv* **2021**, arXiv:2109.12894.
44. Guo, Y.; Chen, Y.; Zhang, L.; Liu, X.; Wang, Y.; Huang, X.; Ma, Z. IM-Loss: Information Maximization Loss for Spiking Neural Networks. In *Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 156–166.
45. Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; Tian, Y. Deep residual learning in spiking neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21056–21069.
46. Li, Y.; Guo, Y.; Zhang, S.; Deng, S.; Hai, Y.; Gu, S. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23426–23439.
47. Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; Luo, Z.Q. 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, New Orleans, LA, USA, 19–20 June 2022; pp. 12444–12453.
48. Xie, K.; Zhang, Z.; Li, B.; Kang, J.; Niyato, D.; Xie, S.; Wu, Y. Efficient federated learning with spike neural networks for traffic sign recognition. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9980–9992. [CrossRef]
49. Viale, A.; Marchisio, A.; Martina, M.; Masera, G.; Shafique, M. Carsnn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–10.