FL-SNNs: Benchmarking the Byzantine-Robustness of Uniquely-Shaped Surrogate Gradients

Manh V. Nguyen, Liang Zhao, Bobin Deng, and Shaoen Wu
College of Computing and Software Engineering, Kennesaw State University, Georgia, USA
mnguy126@students.kennesaw.edu, {lzhao10, bdeng2, swu10}@kennesaw.edu

***Abstract*—The rise of Edge AI necessitates energy-efficient models like Spiking Neural Networks (SNNs), often trained using Federated Learning (FL) to preserve data privacy. However, FL is vulnerable to Byzantine attacks, where malicious clients disrupt training. While SNNs offer potential energy benefits due to their event-driven nature, their unique training mechanisms, particularly the use of surrogate gradients to handle non-differentiable spike events, raise questions about their inherent robustness in adversarial FL settings. We evaluate the robustness of SNNs employing 5 surrogate gradients (distinct by function shape) against 7 diverse Byzantine attacks and assess recovery potential using 5 robust aggregation rules (AGRs). Our extensive experiments (1032 runs) reveal that SNNs are not universally more robust than ANNs; they show resilience to certain structured attacks (e.g., *MinMax*) but vulnerability to others (e.g., *Label Flip*). We find a moderate positive correlation between surrogate gradient choice and recovery effectiveness using AGRs, with *Triangle* and *Rectangle* surrogates often enabling better recovery, though this advantage is context-dependent. Our results underscore that robust AGRs (like *DnC* and *RFA*) are essential for mitigating attacks in SNN-based FL, regardless of the surrogate gradient used. We conclude that achieving reliable SNN deployment in adversarial FL requires a holistic, context-aware approach, carefully considering the interplay between network type, surrogate gradient, threat model, and defense mechanisms. Our code is open-sourced for reproducibility [1].**

***Index Terms*—Surrogate gradients, SNNs, FL, Byzantine**

I. Introduction

The proliferation of intelligent devices at the network edge has spurred the development of Edge AI, enabling localized data processing and reducing reliance on centralized cloud infrastructure. However, edge devices often operate under strict energy constraints. Spiking Neural Networks (SNNs) represent a new class of brain-inspired neural network with promising fault-tolerance and energy-efficiency potentials [1], making them particularly attractive for deployment in resource-constrained edge environments. The inherent event-driven computation of SNNs, where processing only occurs when necessary (i.e., upon spike arrival), offers potential for significant power savings compared to traditional Artificial Neural Networks (ANNs) [2].

On edge devices, training powerful AI models directly presents significant challenges, particularly concerning data privacy. Edge devices often collect sensitive, user-specific data that cannot be shared directly due to privacy regulations and user expectations. Therefore, Federated Learning (FL) [3] has emerged as a key paradigm to address this, enabling multiple edge devices to collaboratively train a shared global model without exchanging their raw local data. Instead, devices typically train models locally and share only model updates (e.g., gradients or weights) with a central server for aggregation, thus preserving data locality and enhancing privacy.

Despite its privacy advantages, the distributed nature of FL introduces vulnerabilities, particularly to Byzantine attacks. Malicious participants (Byzantine clients) can disrupt the training process by sending corrupted or intentionally misleading updates to the central aggregator. These attacks manifest mainly in two forms: *model-poisoning*, where attackers aim to degrade the global model's performance or insert backdoors; and *data-poisoning*, where attackers manipulate their local data to influence the training outcome [4]. The literature landscape encompasses numerous attack strategies, ranging from simple noise injection and label flipping to more sophisticated attacks that mimic benign updates [4]–[6], posing significant threats to the reliability of FL systems.

SNNs operate fundamentally different from conventional ANNs, where as it mimics biological neurons in processing information in a spatiotemporal manner through spike trains instead of single real values in ANNs or binary values in the original Multi-layered Perceptrons (MLP). This temporal dimension and the use of non-differentiable spike activation functions necessitate specialized training algorithms. As elaborated by Zenke et al. and other work [7], due to its temporal nature, SNNs can be trained with a modified version of back-propagation through time (BPTT), a recurrent neural network (RNN) training technique, often referred to as *Spatio-Temporal Back-Propagation (STBP)* [8]. A core challenge in applying gradient descent is the non-differentiable nature of the spike generation event (Heaviside step function).

To deal with this problem, previous work has proposed that the partial derivative of the Heaviside step function can be replaced with a continuous approximation during the backward pass. This widely adopted technique is called *Surrogate Gradients* [7]. While essential for training, the choice and behavior of these surrogate gradients may impact SNN performance beyond optimization, particularly regarding robustness under adversarial settings like Byzantine Federated Learning (FL). This raises the following key questions: *Does the surrogate gradient choice influence an SNN's resiliency to malicious attacks? And how does SNN robustness compare to traditional ANNs under attack?*

[1] https://github.com/manh-vnguyen/surrogate_byzantine_bench

Contribution. In this study, we systematically investigate these questions. We evaluate SNNs employing 5 distinct surrogate gradients against 7 Byzantine attacks and assess recovery using 5 different Byzantine-robust aggregation rules (AGRs) across 4 benchmark datasets and two network architectures. Our main contributions are three-folds:

1. *Highlighting SNN Superiority Against Complex Attacks*: We demonstrate that SNNs exhibit superior resilience compared to ANNs when facing sophisticated, structured Byzantine attacks (such as *Sign Flip* and bounded attacks). Defending against these structured attacks is notably more challenging in federated learning, positioning SNNs as a strategically advantageous choice for enhancing security in these difficult adversarial scenarios.
2. *Optimizing SNN Robustness via Surrogate Gradients*: Our analysis reveals a clear positive correlation between the choice of surrogate gradient function and the ability of SNNs to recover from attacks. Specific gradients, notably Triangle and Rectangle, consistently lead to enhanced robustness, demonstrating that careful selection of the surrogate gradient is a key factor in maximizing SNNs' defensive capabilities against adversarial manipulations.
3. *Validating Robust AGRs as Essential for FL-SNNs Defense*: We provide quantitative evidence establishing that robust aggregation rules (AGRs), especially those leveraging clustering or robust statistics, are critical for bolstering SNN resilience in adversarial FL. These defenses significantly amplify the inherent strengths of SNNs, providing a powerful mechanism to counteract Byzantine threats effectively, complementing the benefits gained from optimized surrogate gradients.

Collectively, these findings underscore that SNNs offer a compelling advantage over ANNs in defending against complex, structured Byzantine attacks. This advantage can be further amplified through strategic selection of surrogate gradients and the implementation of robust aggregation rules, paving the way for more secure and resilient SNN-based federated learning systems.

II. Related Work

A recent survey by Dampfhoffer et al. [9] explores different techniques for training SNNs, highlighting spike-based backpropagation with surrogate gradients as a prominent and widely discussed topic. In [10], Guo et al. provided a survey on the advancement in direct training SNNs with surrogate gradients to enhance the model quality. In which, techniques such as: adaptive threshold/leak-constant [11], [12], dynamic/learnable surrogate gradient [13], [14] and batch-normalization [15], are discussed. Other such work as from Zenke et al. discussed and demonstrate the inherent robustness in training stability and accuracy of different surrogate gradients [16]. However, recent research highlights the vulnerability of surrogate gradients to adversarial attacks, namely: FGSM and PGD [17], [18]; SNN-tailored rate and temporal based (HART) [19]; a set of DVS-attacks [20]; and rate gradient approximation attack (RGA) [21]. In terms of adversarial robustness of surrogate gradients, Sharmin et al. [22] conclude that directly training SNN with surrogate gradients instead conversion from ANNs results in SNNs that are more robust against the FGSM and PGD attacks. Building upon this inherent robustness of surrogate gradients, Kundu et al. [17] introduce crafted input noise during training; Ding et al. [23], [24] explore a dynamic LIF framework and a stochastic gating mechanism; and Liu et al. [25] propose a gradient sparsity regularization scheme; each of them to enhance the robustness against the aforementioned attacks. Moshruba et al. [26] empirically explore the privacy-preserving properties of common surrogate functions and quantization levels under membership inference attacks (MIA).

In the context of FL with SNNs, recent work such as Abad et al. [27], Fu et al. [28], Walter et al. [29], and Riano et al. [30], explore the vulnerability of SNNs against backdoor attacks in FL. Thus, highlighting the need improve the robustness of SNN in such training context. With regards to resiliency of surrogate gradients in FL, Venkatesha et al. [31], demonstrates the better accuracy and energy efficiency of SNNs over ANNs in large-scale FL. While extensive research addresses Byzantine attacks and defenses for ANNs in FL [4]–[6], [32]–[35], [35]–[39], this critical area remains largely unexplored for SNNs trained with surrogate gradients. This gap is particularly significant given that our previous work [40], [41] has demonstrated SNNs to be inherently more resilient in various FL settings. Filling in this critical gap, our study undertakes a systematic exploration of the inherent robustness of these SNNs against major Byzantine threats, benchmarking their performance against ANNs and evaluating the effectiveness of robust aggregation rules.

III. Preliminaries

In this section, we first introduce SNNs training with surrogate gradients, focusing on the most common surrogate functions—each characterized by distinct shapes—which would be benchmarked in our experiments. We then review prevalent Byzantine attack strategies in FL and discuss robust aggregation rules designed to defend against such attacks. These concepts provide the foundation for the analyses and evaluations in the subsequent sections.

A. Surrogate Gradients in Spiking Neural Networks

In spiking neural networks, each neuron utilize a spike train perceived over a determined number of timesteps (T) to represent the input and output. With spike encoding, the leaky-integrate-and-fire (LIF) mechanism is commonly used for spike activation [8]. In which, the membrane potential (U) is accumulated over time with a decay rate (β), the neuron fires an output spike (S) and reset the membrane once its potential exceed a threshold (ϑ). Since spike signal is discrete over time instead of analog and continuous as in ANNs, computation of the gradients of the synaptic weights is hindered since calculating $\partial S/\partial U$ is intractable. Thus, surrogate gradients as functions to approximate the derivative of the spike function. In this work, we provide a survey of existing auxiliary function

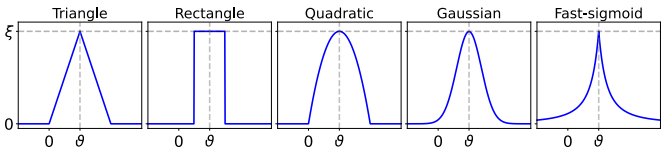

Figure 1. Visualizing the shapes of different surrogate gradient functions

and for surrogate gradients and categorize them based on their shape, we select one formulation of each shape to benchmark its robustness in Byzantine attacks and defense scenarios. Their shapes are visualized in Fig. 1 and formulations are discussed in the following.

Triangle. This surrogate gradient shape is constructed using piece-wise linear approximation:

$$\frac{\partial S}{\partial U} \approx \xi \cdot \max\left(0, 1 - \frac{|U - \vartheta|}{\omega}\right) \tag{1}$$

Rectangle: The rectangular-shaped surrogate gradient Also constructed using piece-wise linear approximation:

$$\frac{\partial S}{\partial U} \approx \begin{cases} \xi \cdot \frac{1}{2\omega} & \text{if } |U - \vartheta| < \omega, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Quadratic: We utilize the formulation that produces the parabolic shape:

$$\frac{\partial S}{\partial U} \approx \begin{cases} \xi \cdot \left(1 - \left(\frac{U-\vartheta}{\omega}\right)^2\right) & \text{if } |U - \vartheta| < \omega, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Gaussian: There are multiple variations of the bell-curve shape, namely: SoftLIF, Sigmoid, Artan, Spike Rate Escape. We utilize the Gaussian formula, which is given as follows:

$$\frac{\partial S}{\partial U} \approx \xi \cdot \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(\frac{-(U - \vartheta)^2}{2\sigma^2}\right) \tag{4}$$

Fast-sigmoid: The concave-wedge shape also have many approximation: Fast-sigmoid, Slayer, and Exponential. We utilize the Fast-sigmoid formulation as follows:

$$\frac{\partial S}{\partial U} \approx \xi \cdot \frac{1}{1 + |\alpha(U - \vartheta)|^2} \tag{5}$$

In which, $\xi < 0$ is a dampening factor adjusted in proportional to T, it is used to prevent the gradients from exploding as they accumulates over the timesteps. In the rectangular, triangular and parabolic formula, ω determine the window size where $\partial S/\partial U > 0$ around ϑ. In the bell-curve formula, σ denotes the standard deviation, which controls the width of the "bell". In the concave formula, α denotes the slope of the Fast-sigmoid approximation.

B. Byzantine Attacks in Federated Learning

To briefly describe our *system model*, we assume an FL training apparatus [3] of n clients, of which f are Byzantine clients. Without the loss of generality, we assume that the first $n - f$ clients are benign, and the last f clients, i.e. $[n - f + 1, n]$, are malicious. We denote $\nabla_i^{(r)}$ as the model update submitted by client $i \in [n]$ at round r. We denote the coordinate-wise average of the benign update as $\nabla_{\text{avg}} = f_{\text{avg}}\left(\nabla_{\{i\in[n-f]\}}\right)$, which is also known as the true update. With the mix of benign and malicious updates submitted, we denote $\widetilde{\nabla} = \texttt{AGR}\left(\nabla_{\{i\in[n]\}}\right)$ as the aggregation result performed by the parameter server. Section III-C lists different methods of aggregation. In this work, we experiment with seven *untargeted Byzantine attacks*, categorized by level of knowledge the adversaries, that are: *non-omniscient* and *omniscient*. Among which, one of them is *data-poisoning* and the rest are of *model-poisoning* capabilities. These are briefly described as follows.

1) Non-omniscient: In this type of attack, the adversaries can modify the updates of the Byzantine clients. However, they are unaware of the benign updates.

Label Flip (L.F.) [32]: This type of attack is called *data-poisoning*. Namely, for each data point, label l is flipped to $L - l - 1$, where L is the number of classes.

Gaussian Random (G.R.) [5]: The attacker injects random vectors drawn from a Gaussian distribution with zero mean and variance to tune potency of attack:

$$\nabla_i = \mathcal{N}(0, \sigma^2), \text{ with } \forall i \in [n - f + 1, n] \tag{6}$$

Sign Flip (S.F.) [33]: After the process of local training, the adversary intercept the communication process and flip the sign of the obtained gradients being submitted:

$$\nabla_i = -\nabla_i, \text{ with } \forall i \in [n - f + 1, n] \tag{7}$$

2) Omniscient: The adversaries assume *full knowledge* of the benign updates and can collude with each others to produce stronger attacks. With the colluding scheme, we denote the malicious update calculated by the adversary and is submitted by all Byzantine clients as $\forall \nabla_{\{i\in[n-f+1,n]\}} = \nabla_{\text{m}}$. The formula for ∇_{m} in different attacks are given as follows.

Mimic [34]: The adversary copies a benign update and amplifies its influence while suppressing others, making the attack nearly indistinguishable from normal activity. This subtle manipulation skews the overall update aggregation in their favor and evades standard anomaly detection measures, posing a serious challenge to robust systems.

$$\nabla_{\text{m}} = \nabla_{\text{random}([n-f])} \tag{8}$$

IPM [6]: Aims to invert the sign of the benign average (∇_{avg}) in the malicious updates, yet, scaled down by the factor γ (estimated manually) so they remain undetected:

$$\nabla_{\text{m}} = -\gamma * \nabla_{\text{avg}} \tag{9}$$

The authors also highlight the need for γ to be large enough such that the inner product of the benign aggregation vector ∇_{avg} and the actual aggregation vector is negative, which would break the gradient descent convergence.

Fang [4]: Still aim to negate the benign update (∇_{avg}) coordinate-wise. However, there are two main difference from IPM. First, the value of the perturbation vector at each coordinate is either -1 or $+1$, depending on the corresponding

sign of ∇_{avg}. Second, γ is dynamically optimized so as to bypass the filtering of the known AGRs (e.g. Krum [5]):

$$\nabla_{\text{m}} = \nabla_{\text{avg}} - \gamma * \text{sign}(\nabla_{\text{avg}}) \tag{10}$$

MinMax [35]: Craft the malicious update by scaling the perturbation calculated by coordinate-wise standard deviation across the benign update. The authors propose an adaptive scaling factor (γ) based on the maximum distance between the malicious updates and the benign updates, and the maximum distance among themselves. With $\nabla_{\text{std}} = f_{\text{std}}\left(\nabla_{i\in[n-f]}\right)$, the optimization problem for γ is formulated as follows:

$$\underset{\gamma}{\text{argmax}} \max_{i\in[n-f]} \|\nabla_{\text{m}} - \nabla_i\|_2 \leq \max_{i,j\in[n-f]} \|\nabla_i - \nabla_j\|_2 \quad \text{with } \nabla_{\text{m}} = \nabla_{\text{avg}} - \gamma\nabla_{\text{std}} \tag{11}$$

C. Byzantine Defense with AGRs

Robust Aggregation Rules (AGRs) are evaluated based on how effective they are at neutralizing the effect of the malicious updates. The default aggregation in FL is coordinate-wise averaging which is ∇_{avg}. In this work, we benchmark five recently proposed AGRs of different basis of reasoning. They are briefly described in the following.

Norm Clipping (N.C.) [36] (Clipping-based): This AGR postulate that the L2 norm of an update should not surpass a predefined threshold M. Then, it clips the update by the ratio of its L2 norm over M if the L2 norm of an update is higher than M, as the following formula shows:

$$\widetilde{\nabla} = \sum_{i\in[n]} \frac{\nabla_i}{\max(1, \|\nabla_i\|_2/M)} \tag{12}$$

Centered Clipping (C.C.) [37] (History, clipping-based): In similar manner to momentum stochastic gradient descent, this approach also incorporate historical updates to make the aggregated update more robust. The formula of such incorporation is given as follows:

$$\widetilde{\nabla}^{(r+1)} = \widetilde{\nabla}^{(r)} + \frac{1}{n}\sum_{i\in[n]} \left(\nabla_i^{(r+1)} - \widetilde{\nabla}^{(r)}\right) \cdot \min\left(1, \frac{\tau_r}{\left|\nabla_i^{(r+1)} - \widetilde{\nabla}^{(r)}\right|}\right) \tag{13}$$

in which, τ_r is the clipping threshold parameter for round r. This acts as a safeguard, preventing any client's update from straying too far from the collective historical direction, thus enhancing robustness to outliers and adversarial updates.

DnC [35] (Composite statistic-based): This approach identifies the malicious update using the outlier filtering method via singular value decomposition (SVD) based spectral. To mitigate the computational cost of apply SVD, the authors proposed selecting a random subset of dimensions then centering the sub-samples by subtracting away their coordinate-wise mean. These centered "sub-updates" are then projected to their top-right singular eigenvector to compute the outlier score. Finally, with the filtering fraction $\kappa \leq 1$, the algorithm filters out $\kappa \cdot f$ vectors of the highest outlier scores, and return the mean of the remaining ones.

Table I
MODEL ARCHITECTURE AND CONFIGURATION

Dataset	Network	Model	Structure
CIFAR10 CIFAR100	VGG9	ANN	Conv (64-64-128-128-256-256-256 filters) AvgPool & BatchNorm FC (conv_output, 1024, num_classes) ReLU
		SNN	Conv (64-64-128-128-256-256-256 filters) AvgPool & BatchNorm FC (conv_output, 1024, num_classes) LIF ($\beta = 0.95; \vartheta = 1.5; T = 15$)
MNIST FMNIST	FCNet	ANN	2 FC layers (Input, 1000, num_classes) ReLU
		SNN	2 FC layers (Input, 1000, num_classes) LIF ($\beta = 0.95; \vartheta = 1.5; T = 15$)

RFA [38] (Single statistic-based): This AGR aims to calculate the weighted geometric median vector is closer to most received updates. The smoothed Weiszfeld algorithm to approximate this solution. The objective function to find this optimal vector is:

$$\widetilde{\nabla} = \underset{\widetilde{\nabla}\in\mathbb{R}^d}{\text{argmin}} \sum_{i\in[n]} \left\|\widetilde{\nabla} - \nabla_i\right\| \tag{14}$$

Sign Guard (S.G.) [39] (Pattern-based): This defense aim to use clustering algorithms (e.g. k-means, DBSCAN, mean-shift) to identify the malicious updates using sign statistics (i.e. total number of positive, zero, negative value of an update) as the input features of the received updates.

IV. EXPERIMENTAL SETUP

A. Surrogate gradient configurations

We implement five surrogate gradients functions in our experiment: *Triangle, Rectangle, Quadratic, Gaussian, Fast-sigmoid*. To reduce the risk of exploding gradient, we set the damping factor of all surrogate gradients to $\xi = 0.3$. For *Triangle* and *Quadratic*, we set the width of piece-wise linear and quadratic formulations as $\omega = 1$. For *Rectangle*, the width is set as $\omega = 0.5$. For the *Gaussian* surrogate gradient, the root-mean-square width, i.e. the width of the "bell", is set as $\sigma = 0.4$. As for the *Fast-sigmiod* formulation, the slope of the concave-wedge is set as $\alpha = 1.5$.

B. Model Architectures, Datasets & Training

This study evaluates MNIST, FMNIST, CIFAR10 and CIFAR100 datasets. For all datasets, we follow the default train/test splits provided, and apply the z-score normalization, resulting in each input channel a mean of 0 and standard deviation of 1. MNIST and FMNIST are gray scale images, i.e. one input channels, of size 28×28, while CIFAR10 and CIFAR100 are colored images with three input channels per pixel and of size 32×32. We handle the data pre-processing with the TorchVision library. We implemented the models and

trainings using PyTorch. We use fully connected network (FC-Nets) architecture to train on MNIST and FMNIST datasets, and convolutional neural networks architecture, specifically, the VGG9 model to train on CIFAR10 and CIFAR100. The detailed configurations of these model architectures for both ANNs and SNNs are presented in Table I. For activation functions, we use ReLU for ANNs and Leaky Integrate and Fire (LIF) neurons (with the membrane decaying rate of $\beta = 0.95$, membrane firing threshold of $\vartheta = 1.5$, number of timesteps to propagate spikes as $T = 15$) for SNNs. For the FL training configurations, we set the number of clients as $n = 20$, in which the number of Byzantine clients is $f = 8$. We implement the FedSGD algorithm where the clients updates are the gradients, and set the number of training rounds as 2000. These hyperparameters are carefully selected so that the baseline training accuracy of ANN and SNN models with different surrogate gradients are roughly similar.

C. Byzantine Attacks & Robust AGRs Configuration

In terms of Byzantine attacks, we evaluate seven different types: *Label Flip (L.F.), Gaussian Random (G.R.), Sign Flip (S.F.), Mimic, IPM, Fang* and *MinMax*. For *G.R.* we set the standard deviation of the noise distribution as $\sigma = 1.0$. For *IPM* we set the scale of attack of $\lambda = 1.1$, which is concluded empirically to be the optimal value in the original paper [6]. For *Fang*, we set the filtering AGR to optimized against as *Krum* [5], the search for optimized λ starts at 1.0 and the stop threshold is set as 10^{-5}. For the *MinMax* attack, we utilize the binary search algorithm as suggested by the authors to find the optimized γ, with the search space $[0, 5]$ and tolerance of 0.01. In terms of Byzantine defenses, we consider coordinate-wise mean (∇_{avg}) to be the baseline for comparison. We evaluate five different robust AGRs (not including the default coordinate-wise average): *Norm Clipping (N.C.), Centered Clipping (C.C.), DnC, RFA*, and *Sign Guard (S.G.)*. For *N.C.*, we set the L2 norm threshold $M = 3$. For *C.C.*, the L2 norm threshold is set as 100, and number of iteration to optimize the local update as 1. For *S.G.* we utilize the k-means algorithm with $k = 2$ to separate the updates into two clusters. For *DnC*, the size of the sub-sampling is set as 10000, the filtering fraction as $\kappa = 1.0$ and number of subset iterations as 10. For *RFA*, we set the number of optimizing iterations for the weighted geometric median vector as 20.

V. Results & Discussion

In this section, we present our experiment results, arranging to answer four questions of increasing complexity. In each subsections, we present the question, summarized metrics, data analysis and our overall discussion to answer them.

A. Is FL-SNNs with surrogate gradients inherently more Byzantine-robust than FL-ANNs?

Table II presents a comparative analysis of the robustness of FL-SNN and FL-ANN under various Byzantine attack strategies across four datasets. The robustness comparison score $\left(\text{acc}_{\text{drop}}^{\text{SNNvANN}}\right)$ is quantified using the difference in accuracy degradation due to an attack $(\text{acc}_{\text{drop}})$ between ANNs and SNNs. As shown in the following equations:

$$\text{acc}_{\text{drop}} = \text{acc}_{\text{baseline}} - \text{acc}_{\text{attacked}} \tag{15}$$

$$\text{acc}_{\text{drop}}^{\text{SNNvANN}} = \text{acc}_{\text{drop}}^{\text{ANN}} - \text{acc}_{\text{drop}}^{\text{SNN}} \tag{16}$$

Positive values indicate that SNNs are more robust (i.e., experience a smaller accuracy drop), while negative values indicate that ANNs are more robust. Note that, SNN accuracy under attack is averaged across five commonly used surrogate gradient functions, as follows:

$$\text{acc}_{\text{drop}}^{\text{SNN}} = \frac{1}{5} \cdot \Big(\text{acc}_{\text{drop}}^{\text{Triangle}} + \text{acc}_{\text{drop}}^{\text{Rectangle}} + \text{acc}_{\text{drop}}^{\text{Quadratic}} + \text{acc}_{\text{drop}}^{\text{Gaussian}} + \text{acc}_{\text{drop}}^{\text{FastSigmoid}} \Big) \tag{17}$$

Table II
Comparing the robustness of SNNs and ANNs against different Byzantine attacks

Dataset	**L.F.**	**G.R.**	**S.F.**	**Mimic**	**IPM**	**Fang**	**MinMax**
MNIST	-0.03	-0.31	-0.02	0.00	0.02	0.04	0.61
FMNIST	-0.06	0.04	-0.01	-0.01	0.03	0.01	-0.07
CIFAR10	-0.04	0.07	0.03	0.01	-0.05	0.06	0.05
CIFAR100	-0.03	0.03	0.01	0.02	-0.03	0.07	0.03

Specific patterns emerge favouring one architecture over the other depending on the attack type and dataset. SNNs consistently demonstrated slightly enhanced robustness against the Label Flipping (L.F.) attack across all four datasets, with $\text{acc}_{\text{drop}}^{\text{SNNvANN}}$ values ranging from -0.03 to -0.06. This suggests SNNs might possess some inherent resilience to simple forms of data poisoning involving label manipulation. The most pronounced advantage for SNNs was observed under the *Gaussian Random (G.R.)* attack on the MNIST dataset, registering a substantial difference of -0.31. Minor advantages for SNNs were also noted for *Sign Flipping (S.F.)* on MNIST and FMNIST, *IPM* on CIFAR10 and CIFAR100, and *MinMax* on FMNIST, indicating context-specific benefits.

Conversely, ANNs exhibited superior robustness in several key instances. The most striking result is the significantly better performance of ANNs against the MinMax attack on the MNIST dataset, where the $\text{acc}_{\text{compare}}^{\text{SNNvANN}}$ reached 0.61. This indicates that, for this specific sophisticated attack on a relatively simpler dataset, the conventional ANN architecture withstood the attack with substantially less accuracy degradation compared to the averaged SNN performance. ANNs also generally showed better resilience against the Fang attack (except on FMNIST) and the G.R. attack (except on MNIST), although the margins were typically smaller than the *MinMax*-MNIST case. These positive values highlight scenarios where the mechanisms within ANNs might be better suited to mitigating certain types of adversarial manipulations introduced by Byzantine workers.

These findings collectively underscore that the relative robustness of SNNs versus ANNs against Byzantine attacks is not absolute but highly contingent on the specific attack

Table III
RECOVERY-EFFECTIVENESS OF SNNS VS ANNS WITH DIFFERENT AGRS APPLIED

Attacks	MNIST					FMNIST					CIFAR10					CIFAR100				
	C.C.	DnC	N.C.	RFA	S.G.	C.C.	DnC	N.C.	RFA	S.G.	C.C.	DnC	N.C.	RFA	S.G.	C.C.	DnC	N.C.	RFA	S.G.
L.F.	−0.15	−0.03	−0.96	−0.03	−0.19	−0.30	−0.02	−0.51	−0.03	−0.38	0.02	−0.05	−0.26	−0.05	−0.87	0.05	1.27	−0.19	0.01	−0.17
G.R.	0.55	−0.01	−0.08	−0.01	−0.01	−0.17	0.00	−0.08	−0.01	−0.01	−0.33	0.00	−0.02	−0.01	0.02	−0.09	0.01	0.01	−0.01	−0.02
S.F.	−0.04	0.09	−0.30	−0.02	−0.26	0.02	0.02	−0.35	−0.08	−0.24	0.02	0.02	−0.20	−0.38	−0.24	0.02	0.63	−0.07	−0.06	−0.27
Mimic	−0.12	−1.53	0.26	−15.90	−4.63	−0.63	−0.03	−1.01	−0.41	−0.54	−1.44	−4.11	−3.46	−81.04	−8.28	−0.21	−0.47	−0.05	−10.47	−0.04
IPM	0.00	−1.26	0.06	−32.71	−15.26	−6.07	70.95	−1.55	658.01	465.85	0.08	1.61	35.77	30.22	−3.28	−0.43	−3.17	27.04	18.64	−5.48
Fang	0.12	0.00	−0.88	−0.23	−0.10	−0.33	0.17	−0.60	−0.62	−0.58	0.13	0.09	0.30	0.16	−0.09	0.53	0.37	0.52	0.27	−0.41
MinMax	−0.02	−0.02	−0.87	−0.37	−0.02	1.85	−0.01	−0.13	1.79	−0.02	0.00	0.00	0.03	0.01	0.01	0.00	0.03	0.01	0.01	0.01

methodology and the data characteristics. Neither architecture demonstrates universal superiority across all tested conditions. An important consideration is that the SNN values are averaged across five surrogate gradient functions. This averaging ensures that the robustness evaluations reflect the intrinsic architectural characteristics of SNNs, rather than the idiosyncrasies of any one optimization method. In later sections, we present the data with regards to specific surrogate gradients and compare the effectiveness among them.

B. Do common Byzantine-robust AGRs more effective when applied to FL-SNNs than to FL-ANNs?

Table III provides a detailed view of how SNNs with surrogate gradients compares to ANNs $\left(\text{acc}_{\text{effective}}^{\text{SNNvANN}}\right)$ in terms of recovery-effectiveness under various Byzantine attacks and robust AGR mechanisms. Across all datasets, the metric $\text{acc}_{\text{effective}}$ captures the relative gain when recovering lost accuracy due to attacks when robust aggregation rules (AGRs) are applied. The following equations provide insight into how these metrics are computed:

$$\text{acc}_{\text{effective}} = \frac{\text{acc}_{\text{defended}} - \text{acc}_{\text{attacked}}}{\text{acc}_{\text{drop}}} \tag{18}$$

$$\text{acc}_{\text{effective}}^{\text{SNNvANN}} = \text{acc}_{\text{effective}}^{\text{SNN}} - \text{acc}_{\text{effective}}^{\text{ANN}} \tag{19}$$

A positive value suggests SNNs are more resilient than ANNs in that specific setting. Once again, the recovery-effectiveness of SNN is also averaged across five surrogate gradients, in a similar fashion as demonstrated in Eq. (17).

For simpler datasets like MNIST, SNNs often show limited or even negative recovery-effectiveness. In several cases—such as under RFA against IPM and Mimic attacks-SNNs perform significantly worse than ANNs, with values dropping to -32.71 and -15.90 respectively. This suggests that in low-complexity datasets where ANNs can exploit stable, dense patterns, SNNs may struggle to recover once perturbed, possibly due to their inherent temporal sparsity and reduced representational density. However, under *Centered Clipping (C.C.)* and *Gaussian Random (G.R.)* attacks, SNNs occasionally show competitive or slightly better robustness, hinting that their noise-resilient spiking mechanisms might still provide some defense against random perturbations.

In contrast, for FMNIST, the recovery-effectiveness of SNNs dramatically improves. Under attacks like *IPM* and with defenses such as *DnC* or *RFA*, SNNs achieve large positive values, even reaching $+70.95$ and $+658.01$. This suggests that on datasets with moderate visual complexity, the sparse, temporal encoding of SNNs offers substantial benefits when paired with cluster-based or structure-aware aggregation. These results highlight a strong synergy between the SNN model and certain defenses that better isolate or suppress corrupted updates, aligning well with the temporal dynamics of SNNs.

On CIFAR10 and CIFAR100, the comparison becomes more nuanced. CIFAR10 shows a mix of positive and negative values, with SNNs showing some robustness under *Norm Clipping* and *RFA*, especially against *IPM* attacks ($+35.77$ and $+30.22$), but also struggling in extreme cases like the *Mimic* attack with *RFA* (-81.04). CIFAR100, being the most complex dataset, shows smaller but more balanced differences. In several instances e.g., under Fang attack with *Centered Clipping* or *DnC*—SNNs slightly outperform ANNs, although the margins are relatively modest. These observations suggest that on high-complexity datasets, both architectures face significant challenges, and the effectiveness of defense mechanisms becomes even more critical. Overall, these findings indicate that SNNs are not universally more robust than ANNs but their recovery-effectiveness can outperform them under specific adversarial and defense conditions. Their performance is especially promising when paired with defenses that exploit clustering (like DnC) or robust aggregation (like RFA), particularly on datasets with moderate feature diversity.

C. Would different surrogate gradients offer various levels of inherent robustness against Byzantine attacks?

In Fig. 2, we provide detailed data and analysis of comparing the accuracy degradation, as shown in Eq. (15), of SNNs training with five different surrogate gradients under seven different Byzantine attacks. The results reveal that across the spectrum of attacks and datasets, no surrogate gradient demonstrates universal superiority. For each, catastrophic accuracy drops occur under the most severe attacks (such as *MinMax* and *Gaussian Random*), and especially on complex datasets like CIFAR-10 and CIFAR-100, where all surrogates typically lose more than 0.6 in accuracy. The difference in robustness between surrogates here becomes indistinct. This suggests that for the harshest and most effective Byzantine attacks, the particular surrogate gradient employed does little to protect the learning process, with attack strategy and dataset complexity

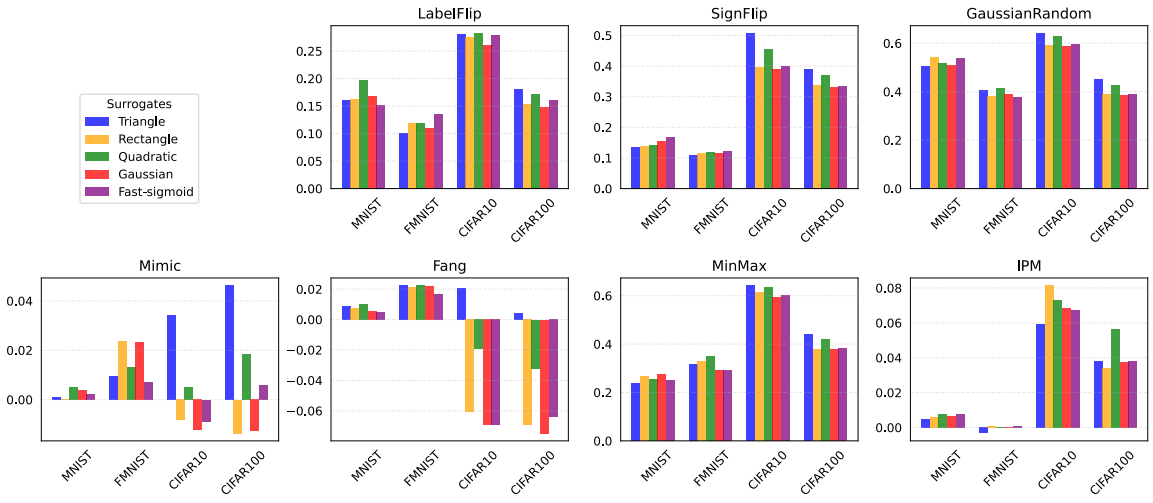

Figure 2. Accuracy degradation (acc_{drop}) under Byzantine attacks of different surrogate gradients (higher value is worse)

being the overwhelming factors. In scenarios with moderate threats, such as *Label Flip* and *Sign Flip*, some variability in surrogate robustness becomes more apparent, though differences remain subtle. For instance, under *Label Flip* across all datasets, *Gaussian* and *Fast-sigmoid* exhibit marginally reduced accuracy losses compared to others, particularly on FMNIST and CIFAR-100. Meanwhile, the Triangle surrogate is sometimes more vulnerable to *Sign Flip* on CIFAR-10, with its accuracy drop peaking compared to its peers. Nonetheless, for these attack settings, all surrogates still experience substantial performance degradation, indicating that choice of surrogate alone is insufficient as a robust defense.

The relative strength of each surrogate emerges most clearly under weak or ineffective attacks. With *Mimic* and *Fang* attacks, all surrogate gradients maintain high accuracy, with losses near zero or even slight improvements observed—especially for MNIST and FMNIST. *Rectangle*, for example, shows the largest (but still small) accuracy loss under *Mimic* in CIFAR-100, but in general all surrogates are robust under these adversarial conditions. This indicates that in low-threat environments, the type of surrogate gradient used is largely irrelevant to overall robustness: all perform well.

D. Would different surrogate gradients impact the recovery-effectiveness of FL-SNNs when AGRs are applied?

In Fig. 3, we shows how well SNNs training with different surrogate gradients recover from various adversarial attacks when applied robust AGRs (rows), across four datasets (columns). In each 3D plot, the x-axis lists surrogate gradients (colored lines), the y-axis denotes attack types (marker shapes), and the z-axis shows recovery-effectiveness, as shown in Eq. (18), the higher the line, the better the recovery.

The data reveals that the choice of surrogate gradient is a significant factor in determining the recovery effectiveness. Numerically, the *Triangle* and *Rectangle* surrogates yield the highest and most consistent recovery effectiveness scores, frequently ranging from 0.5 up to 1 on simpler datasets such as MNIST and FMNIST, particularly when paired with *Norm Clipping (N.C.)* or *Centered Clipping (C.C.)*. In several instances, *Triangle* achieves scores near or above 1.0, suggesting that, in these settings, the defended model not only recovers from the attack but may marginally outperform the attacked baseline. On these datasets, the inter-surrogate difference can reach as much as a factor of two to four in recovery effectiveness, indicating a pronounced correlation between surrogate selection and defense performance. In contrast, the *Fast-sigmoid* surrogate demonstrates marked under performance, with recovery effectiveness frequently dropping into negative values, especially for complex datasets such as CIFAR10 and CIFAR100 and under aggregation rules like *Sign Guard* and *DnC*. For example, under *RFA* or *Sign Guard* defenses and the CIFAR-class datasets, the effectiveness scores for *Fast-sigmoid* and, at times, *Gaussian*, can fall well below zero, with some values observed in the -100 to -600 range. These negative values indicate not only unsuccessful recovery but also possible exacerbation of attack effects, reinforcing the notion of incompatibility between these surrogates and certain robust aggregation rules.

The role of surrogate gradient diminishes in extremely adversarial or complex scenarios, as evidenced by the narrowing performance margins and universally lower recovery effectiveness on CIFAR10 and CIFAR100, regardless of the surrogate employed. The advantage of *Triangle* and *Rectangle* is less consistent in these regimes, and variability increases across all surrogates. Furthermore, the relative effect of surrogate gradients is modulated by both the chosen aggregation rule and the nature of the attack. For instance, aggressive attacks such as *MinMax* and *Fang* consistently reduce overall recovery, compressing the spread of effectiveness scores among different surrogates. Nevertheless, even under these adverse conditions, *Triangle* and *Rectangle* surrogates tend to retain a relative edge in most settings.

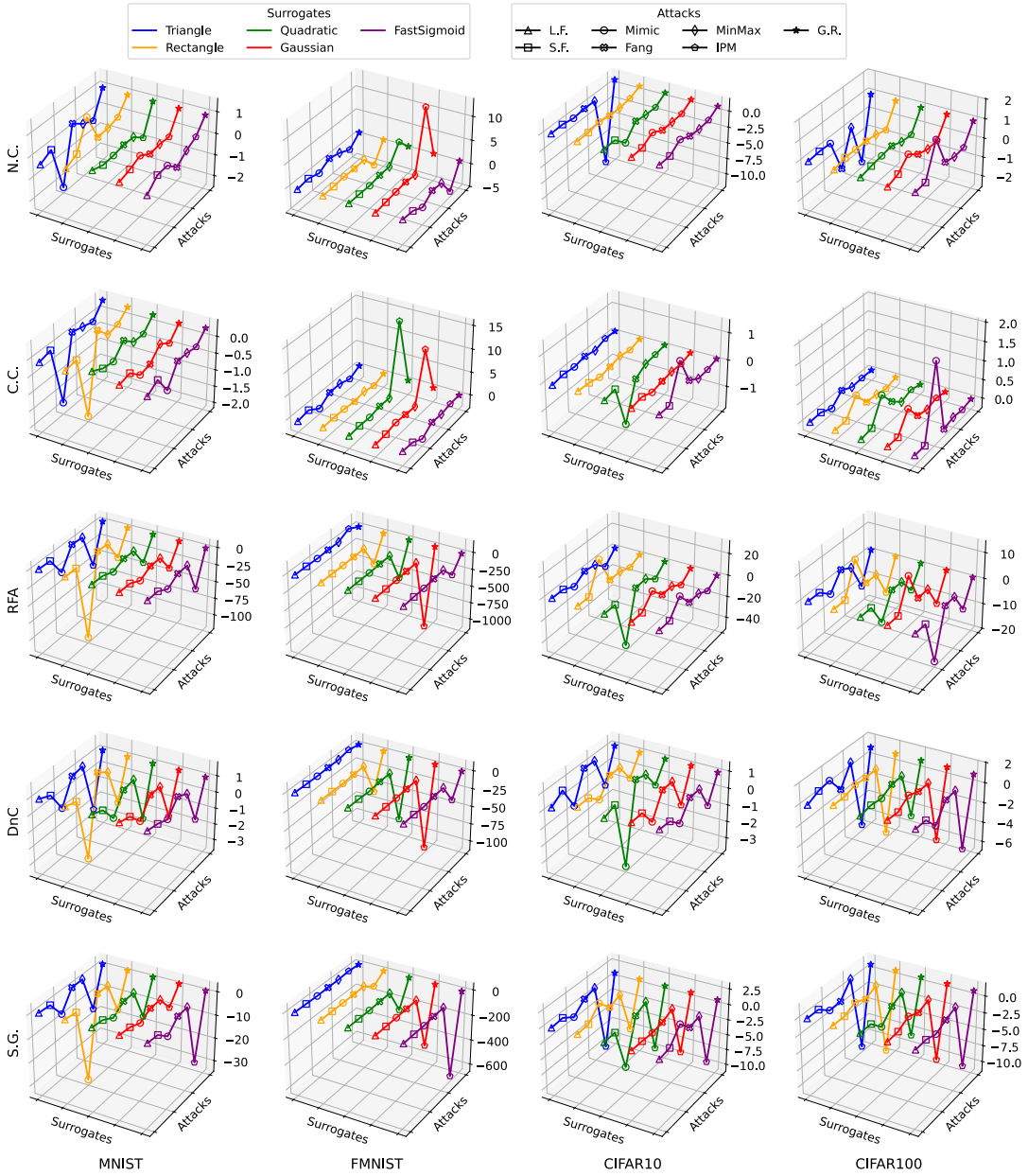

Figure 3. Recovery-effectiveness scores are shown for different surrogate gradients (colored lines) under various Byzantine attacks (marker shapes). Each row represents a different defense method, and each column corresponds to a specific dataset, allowing comparison across defenses, attacks, and datasets.

VI. Conclusion

This study provided a comparative analysis of SNNs and traditional ANNs in Byzantine FL, and specifically the influence of different surrogate gradients. Our findings reveal that SNNs possess distinct and valuable robustness characteristics: they exhibit strong resilience against structured attacks (such as *MinMax*), demonstrating a clear advantage in these challenging scenarios. While vulnerabilities to certain noise-based or data-poisoning attacks exist, this finding still positions SNNs as a promising alternative for specific adversarial landscapes. Crucially, we show that SNNs can indeed surpass ANN performance under attack conditions, particularly when integrated with powerful aggregation rules (AGRs) like *DnC* or *RFA*. Our analysis also indicates that the choice of surrogate gradient, with functions like *Triangle* and *Rectangle* positively contributes to the SNN's inherent Byzantine-resiliency. Ultimately, our results underscore that integrating robust AGRs is critical for unlocking and enhancing the adversarial resilience of SNN-based FL, requiring a strategic consideration of the interplay between the network architecture, effective surrogate gradient selection, and powerful aggregation defenses. Future work will build upon these findings with deeper theoretical and empirical analyses to fully realize the potential of robust neuromorphic FL frameworks.

REFERENCES

[1] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[2] H. Xue, "Neuromorphic Computing with Large Scale Spiking Neural Networks," Mar. 2025. [Online]. Available: https://www.preprints.org/manuscript/202503.1505/v1

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[5] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[6] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 261–270.

[7] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[8] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.

[9] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, "Backpropagation-based learning techniques for deep spiking neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[10] Y. Guo, X. Huang, and Z. Ma, "Direct learning-based deep spiking neural networks: a review," *Frontiers in Neuroscience*, vol. 17, p. 1209795, 2023.

[11] S. Wang, T. H. Cheng, and M.-H. Lim, "Ltmd: learning improvement of spiking neural networks with learnable thresholding neurons and moderate dropout," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 350–28 362, 2022.

[12] Q. Yu, J. Gao, J. Wei, J. Li, K. C. Tan, and T. Huang, "Improving multispike learning with plastic synaptic delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 254–10 265, 2022.

[13] Y. Guo, Y. Chen, L. Zhang, X. Liu, Y. Wang, X. Huang, and Z. Ma, "Im-loss: information maximization loss for spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 156–166, 2022.

[14] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," *Advances in neural information processing systems*, vol. 34, pp. 23 426–23 439, 2021.

[15] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 062–11 070.

[16] F. Zenke and T. P. Vogels, "The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks," *Neural computation*, vol. 33, no. 4, pp. 899–925, 2021.

[17] S. Kundu, M. Pedram, and P. A. Beerel, "Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 5189–5198. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00516

[18] J. Ding, T. Bu, Z. Yu, T. Huang, and J. Liu, "Snn-rat: Robustness-enhanced spiking neural network through regularized adversarial training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 780–24 793, 2022.

[19] Z. Hao, T. Bu, X. Shi, Z. Huang, Z. Yu, and T. Huang, "Threaten spiking neural networks through combining rate and temporal information," in *The Twelfth International Conference on Learning Representations*, 2023.

[20] A. Marchisio, G. Pira, M. Martina, G. Masera, and M. Shafique, "Dvs-attacks: Adversarial attacks on dynamic vision sensors for spiking neural networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–9.

[21] T. Bu, J. Ding, Z. Hao, and Z. Yu, "Rate gradient approximation attack threats deep spiking neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7896–7906.

[22] S. Sharmin, N. Rathi, P. Panda, and K. Roy, "Inherent Adversarial Robustness of Deep Spiking Neural Networks: Effects of Discrete Input Encoding and Non-Linear Activations," Jul. 2020, arXiv:2003.10399 [cs]. [Online]. Available: http://arxiv.org/abs/2003.10399

[23] J. Ding, Z. Pan, Y. Liu, Z. Yu, and T. Huang, "Robust Stable Spiking Neural Networks," May 2024, arXiv:2405.20694 [cs]. [Online]. Available: http://arxiv.org/abs/2405.20694

[24] J. Ding, Z. Yu, T. Huang, and J. K. Liu, "Enhancing the Robustness of Spiking Neural Networks with Stochastic Gating Mechanisms," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, pp. 492–502, Mar. 2024, number: 1. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/27804

[25] Y. Liu, T. Bu, J. Ding, Z. Hao, T. Huang, and Z. Yu, "Enhancing Adversarial Robustness in SNNs with Sparse Gradients," May 2024, arXiv:2405.20355 [cs]. [Online]. Available: http://arxiv.org/abs/2405.20355

[26] A. Moshruba, S. Snyder, H. Poursiami, and M. Parsa, "On the Privacy-Preserving Properties of Spiking Neural Networks with Unique Surrogate Gradients and Quantization Levels," Feb. 2025, arXiv:2502.18623 [cs]. [Online]. Available: http://arxiv.org/abs/2502.18623

[27] G. Abad, S. Picek, and A. Urbieta, "Time-distributed backdoor attacks on federated spiking learning." [Online]. Available: http://arxiv.org/abs/2402.02886

[28] H. Fu, G. Li, J. Wu, J. Li, X. Lin, K. Zhou, and Y. Liu, "Spikewhisper: Temporal spike backdoor attacks on federated neuromorphic learning over low-power devices."

[29] K. Walter, M. Mohammady, S. Nepal, and S. S. Kanhere, "Mitigating distributed backdoor attack in federated learning through mode connectivity," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, 2024, pp. 1287–1298.

[30] R. Riaño, G. Abad, S. Picek, and A. Urbieta, "Flashy backdoor: Real-world environment backdoor attack on snns with dvs cameras," *arXiv preprint arXiv:2411.03022*, 2024.

[31] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated learning with spiking neural networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6183–6194, 2021.

[32] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[33] G. Damaskinos, R. Guerraoui, R. Patra, M. Taziki *et al.*, "Asynchronous byzantine machine learning (the case of sgd)," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1145–1154.

[34] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via bucketing," *arXiv preprint arXiv:2006.09365*, 2020.

[35] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[36] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can You Really Backdoor Federated Learning?" Dec. 2019, arXiv:1911.07963 [cs]. [Online]. Available: http://arxiv.org/abs/1911.07963

[37] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for byzantine robust optimization," in *International conference on machine learning*. PMLR, 2021, pp. 5311–5319.

[38] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[39] J. Xu, S.-L. Huang, L. Song, and T. Lan, "Byzantine-robust federated learning through collaborative malicious gradient filtering," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 1223–1235.

[40] M. V. Nguyen, L. Zhao, B. Deng, W. Severa, H. Xu, and S. Wu, "The robustness of spiking neural networks in communication and its application towards network efficiency in federated learning," *arXiv preprint arXiv:2409.12769*, 2024.

[41] M. V. Nguyen, L. Zhao, B. Deng, and S. Wu, "The robustness of spiking neural networks in federated learning with compression against non-omniscient byzantine attacks," *arXiv preprint arXiv:2501.03306*, 2025.