Spikewhisper: Temporal Spike Backdoor Attacks on Federated Neuromorphic Learning over Low-power Devices

Hanqing Fu[1], Gaolei Li[1], Jun Wu[1], Jianhua Li[1], Xi Lin[1], Kai Zhou[2], and Yuchen Liu[3]

[1] Shanghai Jiao Tong University
{fuhanqing,gaolei_li,junwuhn,lijh888,linxi234}@sjtu.edu.cn
[2] Hong Kong Polytechnic University
kaizhou@polyu.edu.hk
[3] North Carolina State University
yuchen.liu@ncsu.edu

Abstract. Federated neuromorphic learning (FedNL) leverages event-driven spiking neural networks and federated learning frameworks to effectively execute intelligent analysis tasks over amounts of distributed low-power devices but also perform vulnerability to poisoning attacks. The threat of backdoor attacks on traditional deep neural networks typically comes from time-invariant data. However, in FedNL, unknown threats may be hidden in time-varying spike signals. In this paper, we start to explore a novel vulnerability of FedNL-based systems with the concept of time division multiplexing, termed Spikewhisper, which allows attackers to evade detection as much as possible, as multiple malicious clients can imperceptibly poison with different triggers at different timeslices. In particular, the stealthiness of Spikewhisper is derived from the time-domain divisibility of global triggers, in which each malicious client pastes only one local trigger to a certain timeslice in the neuromorphic sample, and also the polarity and motion of each local trigger can be configured by attackers. Extensive experiments based on two different neuromorphic datasets demonstrate that the attack success rate of Spikewispher is higher than the temporally centralized attacks. Besides, it is validated that the effect of Spikewispher is sensitive to the trigger duration.

Keywords: Federated Learning · Backdoor Attacks · Spiking Neural Networks.

1 Introduction

Federated neuromorphic learning (FedNL), a combination of federated learning (FL) and spiking neural networks (SNNs), enables substantial low-power devices to quickly and energy-efficiently obtain Artificial Intelligence (AI) models from large amounts of distributed data while protecting data privacy. In federated

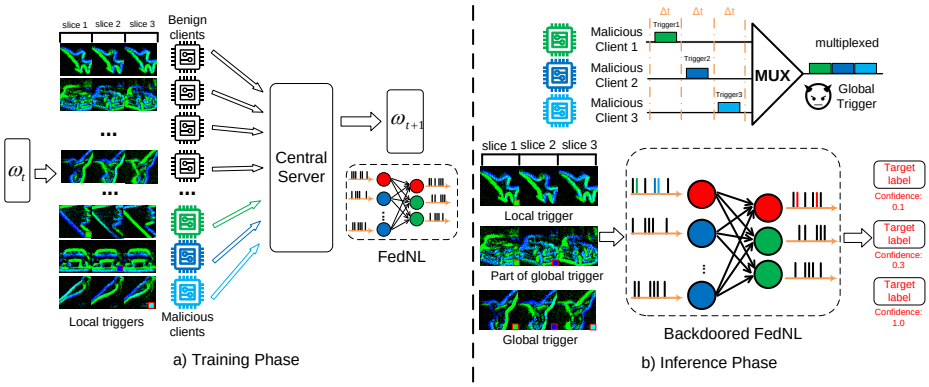


Fig. 1: Overview of Temporal Spike Backdoor Attacks on FedNL. In the training phase, the Central Server aggregates parameters from local benign and malicious participants in the previous round t, updating the global SNN model parameters ω_{t+1}. The attackers use only a subset of the global trigger's temporal sequence as the local trigger for implementing the backdoor attack. In the inference phase, all clients with the global SNN model will misclassify input with the global trigger into the target class.

learning, each device trains an AI model locally and then uploads the model parameters or gradients to a central server for global model aggregation. SNNs have been proposed and explored as a low-power neuromorphic alternative to traditional deep neural networks (DNNs) due to the event-driven and discrete features of signal processing [10,8]. In the era of thriving large models, for example, training the GPT-3 model consumed about 190,000 kWh of electricity [7], FedNL emerges with the potential for allowing low-power devices to collaboratively train the large model.

Nowadays, FedNL has garnered widespread attention [22,23,28]. However, in the same way as the federated learning with DNNs [18,5], federated neuromorphic learning is vulnerable to a variety of security threats, and one of the most critical threats is backdoor attack, which modifies the training set to inject triggers in certain examples. After training, the model will perform correctly in the main task. However, with the presence of triggers (backdoors) on the input samples, the model will go wrong and misclassify samples to the target label. Existing backdoor research has mainly considered DNNs rather than SNNs. Backdoor threats under FedNL urgently need further study.

Considerable attention has been paid to exploring backdoor attacks on traditional DNNs. However, there has been scant attention paid to investigating backdoor attacks targeting FedNL with SNNs. In this paper, we delve into the feasibility of backdoor attacks in FedNL. In response to the temporally distributed characteristic, a novel method with the concept of Time Division Multiplexing for backdoor attacks is designed. By splitting the global trigger into multiple spike timeslices, the local triggers are concealed within each spike timeslice of

the neuromorphic data, significantly enhancing the stealthiness and effectiveness of the backdoor attack. It exposes a novel security vulnerability for FedNL, which is crucial for the security of edge intelligence applications [4]. The main contributions of this paper are summarized as follows:

- A novel temporal spike backdoor attack scheme is proposed for FedNL over low-power devices, named Spikewhisper, which is distributed in the time dimension rather than the spatial dimension. To the best of our knowledge, this is the first work on the robustness and security of federated neuromorphic learning.
- Different from traditional neural backdoor attacks, we identify that the backdoor effect of Spikewhisper is extremely sensitive to not only local trigger size and location but also temporal duration.
- Extensive experiments on Attack Success Rate (ASR) and Main Task Accuracy (MTA) with two different neuromorphic datasets demonstrate that Spikewhisper achieves state-of-the-art attack effects against temporal centralized backdoor attacks.

The rest of this paper is organized as follows. Section 2 introduces the related work of FedNL and backdoor attacks. Section 3 introduces the Spikewhisper system model. The experiments are presented in Section 4. Section 5 provides a conclusion and outlook for our work.

2 Related work

Since our work primarily follows two research directions: Federated Neuromorphic Learning and Backdoor Attacks, a comprehensive introduction of recent advances in these two areas is as follows.

2.1 Federated Neuromorphic Learning

Federated Neuromorphic Learning (FedNL) is pioneered by Skatchkovsky et al. [22] to effectively train SNNs for low-power edge intelligence, providing an effective trade-off between communication overhead and training accuracy. To capture dynamic spike characteristics at time-domain and reduce the training cost, Venkatesha et al.[13,23] proposed a Batch Normalization Through Time (BNTT) algorithm, which decoupled the parameters of each layer of neurons on the time axis. On the basics of this, they also validated that the accuracy of FedNL is 15% higher than that of DNNs on CIFAR10 as well as the energy efficiency is 5.3 times higher. Xie et al. [27] applied the FedSNN-NRFE approach based on neuronal receptive field encoding in traffic sign recognition. In comparison to CNN, FedSNN-NRFE significantly reduced energy consumption. Meanwhile, Yang et al. [28] proposed a lead federated neuromorphic learning framework for wireless edge intelligence, designating devices with high computational capacity, communication, and energy resources as leaders to effectively accelerate the training process. Wang et al. [25] introduced SNNs into asynchronous federated learning, which adapts to the statistical heterogeneity of users and complex communication environments.

2.2 Backdoor Attacks

Gu et al. first introduced neural backdoor attacks, named Badnets [12]. This attack method involves training on images with square patches, creating a backdoor that can be triggered at will by the attacker. Bagdasaryan et al. introduced backdoor attacks into the field of FL [5]. In this context, attackers train the backdoor model locally and upload local model updates scaled by a constraint replacing the benign global model. Bhagoji et al. proposed a stealthy model poisoning method [6], with the use of an alternating minimization strategy that alternately optimizes for stealth and the adversarial objective. Xie et al. introduced the distributed backdoor attack (DBA) [26], wherein the global trigger is spatially decomposed into local triggers. These local triggers are then individually embedded into the training data of multiple malicious parties, enabling a distributed implementation of the backdoor attack.

Abad et al. first investigated the application of backdoor attacks in SNNs using neuromorphic datasets [1,2]. The subsequent work [3] by Abad et al. concurrently explored backdoor attacks on FedNL alongside our research. Compared to this work, we made more explorations in the design and duration of local triggers.

In this paper, we identify a very sophisticated attack path with the concept of Time Division Multiplexing, that is, poisoning by different clients on different frames of neuromorphic data with different local triggers, which further improves the awareness level about the security risks of FedNL.

3 System Model

In this section, we will present the comprehensive system model of Spikewhisper. Section 3.1 introduces the backdoor threat model under FedNL, while Section 3.2 discusses the variations that attackers face when transitioning from FL to FedNL. Section 3.3 provides an overview of Spikewhisper.

3.1 Threat Model

FedNL's goal is to train a global SNN model that can generalize well on test data D_{test} after aggregating over the distributed training results from N clients with their N local datasets D_i on a central server S. The objective of FedNL can be cast as a finite-sum optimization as below:

$$\min_{\omega \in R^d} \left[F(\omega) := \frac{1}{N} \sum_{i=1}^{N} f_i(\omega) \right] \tag{1}$$

where ω stands for the parameters of the model and f_i stands for the loss function $\sum_{(x,y)\in\mathcal{D}_i} \mathcal{L}(f_\omega(x), y)$.

Specifically, at round t, S dispatches the current global SNN model G_t to a subset of clients denoted by $n \in \{1, 2, ..., N\}$. The selected client, indexed as i, locally computes the loss function f_i and adopts surrogate gradient descent [20] for E local epochs.

Attacker ability We consider the attacker's ability as follows:

- The attacker has full control over the local training data of any compromised participant. All compromised participants conspire under the attacker's control to conduct backdoor attacks against the FedNL system, which is consistent with FL settings. [24]
- The attacker can manipulate the local training process, such as updating hyperparameters like the number of epochs and learning rate.
- The attacker can not tamper with any aspects of the benign participants' training.
- The attacker does not have control over the central server's aggregation algorithm used to combine participants' updates into the joint SNN model.

Attack Objective The attacker wants FedNL to produce a joint backdoor SNN model that has good performance on both the main task and the backdoor task. In other words, the SNN model predicts normally on any clean input while predicting a target label $\hat{y}$ on any input that has a global trigger. The adversarial objective for attacker i in round t with local dataset D_i and target label $\hat{y}$ is:

$$\omega_{t+1} = \arg\min_{\omega}\Big(\sum_{j \in D_i^{cln}} \mathcal{L}(f_\omega(x_j), y_j) + \sum_{j \in D_i^{poi}} \mathcal{L}(f_\omega(\hat{x_j})), y_t)\Big) \tag{2}$$

where the clean dataset D_i^{cln} and poisoned dataset D_i^{poi} satisfy $D_i^{cln} \cup D_i^{poi} = D_i$ and $D_i^{cln} \cap D_i^{poi} = \emptyset$. The $\hat{x}$ is the backdoored, and y_t is the backdoor target label.

3.2 Changes Faced by Attackers

The transition from FL to FedNL introduces a series of changes that pose challenges for attackers. These changes are outlined as follows.

The biggest difference between SNNs and traditional DNNs is the feature of information processing. DNNs process continuously changing real-value, whereas SNNs process discrete events that occur at certain time points due to using spiking neurons. The Leaky-Integrate-and-Fire (LIF) model [11] is frequently used to simulate neuronal functions in SNNs.

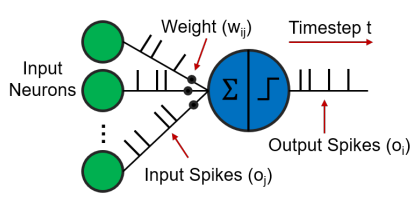


Fig. 2: The Leaky-Integrate-and-Fire Behavior of Spiking Neuron i.

As shown in Fig. 2, for given Spiking Neuron i, a set of spikes from N input neurons is accumulated through the weight w_{ij} for all $j \in N$, forming the membrane potential of the neuron. Once the membrane potential reaches the threshold v, an output spike is triggered. After the occurrence of a spike, the membrane potential is reset to the resting potential u_{rest}, or in the case of a soft reset, the membrane potential is decreased by the threshold v. The entire process persists for T timesteps.

occurrence of a spike, the membrane potential is reset to the resting potential u_{rest}, or in the case of a soft reset, the membrane potential is decreased by the threshold v. The entire process persists for T timesteps. Formally, the Leaky Integrate-and-Fire (LIF) mechanism can be expressed as follows:

$$u_i^t = \lambda u_i^{t-1} + \sum_{j \in N} w_{ij} o_j^{t-1} \tag{3}$$

where u_i^t represents the membrane potential of neuron i at timestep t, λ is a constant leakage factor, indicating how much the membrane potential decreases per timestep. The discrete nature of information processing makes SNNs employ Surrogate Gradient Descent [20] for training.

Neuromorphic data are widely considered to be the most suitable data for SNNs today. Neuromorphic data consists of many spiking events that are captured by the Dynamic Vision Sensor (DVS) sensing the intensity change (increase or decrease) of each pixel in the environment, e.g. ON channel indicates an increase and OFF channel indicates a decrease. The entire spiking sequence can be represented as an event of size $T \times P \times H \times W$, where H and W are the height and width, T denotes the length of the recording time, and P denotes two channels of polarity. For ordinary data in the image domain, the triggers commonly are encoded in 256 possibilities per pixel per channel (usually 3 channels), which allows for many color combinations. In neuromorphic data, however, each pixel could only take the value 0 or 1 in two channels (On and OFF polarity channel). In other words, the trigger space is reduced to 4 possibilities encoded by the two different polarities in SNNs, which limits the backdoor trigger design greatly. Additionally, neuromorphic data is time-encoded, which is temporal and contains T frames while ordinary images are static and non-temporal.

After being aware of the discrete nature of information processing, the limitations of color combinations, and the flexibility in the time dimension, we propose Spikewhisper to inject backdoors in FedNL.

3.3 Spikewhisper Framework

Our proposed Spikewhisper utilizes the concept of Time Division Multiplexing (TDM) to enhance the backdoor efficacy and stealthiness in FedNL as shown in Fig. 1. In TDM, different signals are interleaved within different time slots to share a channel. Similarly, in Spikewhisper, the neuromorphic data is segmented into multiple timeslices. Let T represent the total duration of the neuromorphic

data. The allocation of timeslices can be represented as follows:

$$T = \{t_1, t_2, ..., t_K\} \text{ while } T = \sum_{i=1}^{K} \Delta t_i \tag{4}$$

Where Δt_i represents the duration of timeslice t_i and K represents the total number of timeslices, consistent with the number of malicious participants controlled by the attacker. The length of timeslice allocated to each malicious participant can be freely distributed according to demand, constrained only by the total length T. In the field of communications, there is the concept of channel utilization rate. In Spikewhisper, the redundant space for backdoor triggers of neuromorphic data can be analogized as a 'temporal channel', and the temporal utilization rate U, can be represented as:

$$U = \sum_{i=1}^{K} \frac{\Delta L_i}{\Delta t_i} \tag{5}$$

Where ΔL_i stands for the local trigger's duration of t_i. Subsequent experiments in Section 4.5 illustrate that high utilization produces a stronger backdoor effect.

In the poisoned data design stage, the attacker has the flexibility to configure the polarity and motion of the local triggers in a wide range of cases. As stated in Section 3.2, We denote these four polarity possibilities as p_0, p_1, p_2, and p_3. Consequently, for different polarity combinations, triggers show different colors, i.e. black, green, dark blue, or light blue. Due to the multiple time frames in neuromorphic data, we can change the position of local triggers frame by frame, creating a sense of motion. This aligns more with the motion (illumination changes) nature of neuromorphic data and facilitates more stealthy and more natural triggers. Formally, the process of generating a poisoned data is as follows:

$$(\hat{x}, y_t) = R_i((x, y), p, m, s) \tag{6}$$

Where $\hat{x}$ represents the modified input, y_t is the target label, p denotes the trigger polarity, m represents the trigger's motion trajectory, and s represents the size of the trigger.

For the backdoor training phase, the malicious client adopts the batch poison approach. Based on the poisoning rate r, the malicious client i poisons $r \times BatchSize$ clean data, only poisoning the data at the i-th timeslice. Additionally, the malicious client uses a smaller learning rate and more local epochs to achieve a better backdoor effect. Formally, the training process is represented as follows:

$$\Delta w_{ij} = \sum_{t=1}^{T} \frac{\partial L}{\partial w_{ij}^t} = \sum_{t=1}^{T} \frac{\partial L}{\partial o_i^t} \frac{\partial o_i^t}{\partial u_i^t} \frac{\partial u_i^t}{\partial w_{ij}^t} \tag{7}$$

Where Δw_{ij}, the gradient of the weight connecting neuron i and j is accumulated over T timesteps. The loss function L is to evaluate the mean square error between output fire rates and sample label y_i, which is given by

$$L = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{T} \sum_{t=1}^{T} S_i(t) - y_i \right)^2 \tag{8}$$

In the inference phase of FedNL, the attacker uses a global trigger to implement a backdoor attack. The global trigger is formed by aggregating all local triggers, which can be represented as $T_{\text{global}} = \sum_{i=1}^{K} T_{\text{local},i}$

In the Spikewhisper framework, the allocation of timeslices and the design of triggers exhibit a high degree of flexibility. For ease of experimental demonstration, we adopt equally sized time slice allocations. Specifically, we segmented neuromorphic data into 3 timeslices and designed two types of global triggers, namely static trigger and moving trigger, as illustrated in Fig. 3. Taking the static

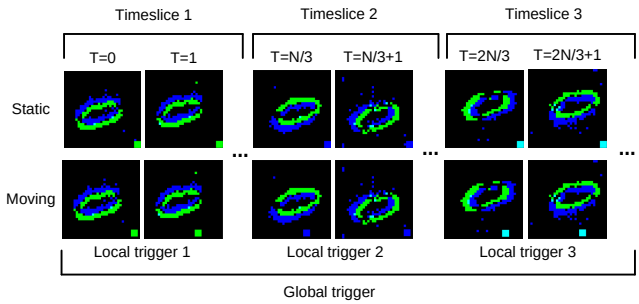


Fig. 3: Static trigger and moving trigger used in Spikewhisper.

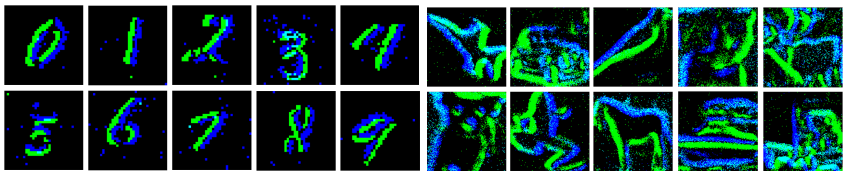
trigger as an example, this trigger is inspired by BadNets [12] in DNNs, where a fixed-position square trigger is on all frames of neuromorphic data. However, the polarity of the trigger varies over timeslices. As the polarity changes from p_1 to p_3 (except the polarity p_0 representing the color black), the color of the static trigger changes on different timeslices too. The moving trigger is inspired by the previous backdoor work on SNNs [1,2]. The trigger horizontally and smoothly changes the location from frame to frame, creating an effect of moving among the actions of the image, making the procedure more stealthy and more natural. As for the variation in the polarity for each timeslice of moving triggers, it remains consistent with that of static triggers, which is different from the previous SNN backdoors.

If we transplant the traditional FL backdoor attack to FedNL, we would obtain a form of temporally centralized attack. In this scenario, each malicious user employs the same global trigger to poison the data, and this trigger spans every frame of the neuromorphic data. In contrast to Spikewhisper, which conceals local triggers within specific data timeslices, this attack method significantly increases the risk of poisoning exposure, exhibits poor stealthiness, and overlooks the temporal distribution characteristics of FedNL.

4 Experiment

4.1 Datasets & Network Architectures

We evaluate Spikewhisper on two neuromorphic datasets: N-MNIST [21] and CIFAR10-DVS [16], which are converted from the most popular benchmarking datasets for AI security in computer vision.



(a) N-MNIST (b) CIFAR10-DVS

Fig. 4: Neuromorphic Data Samples (one frame of each sample).

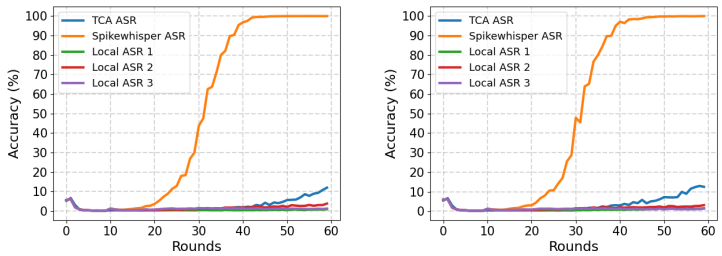
The N-MNIST dataset is a spiking version of the MNIST [15] dataset. It comprises 60000 training samples and 10000 test samples for 10 classes, maintaining the same scale as the original MNIST dataset, but with a sample size of 34×34 instead of 28×28. Likewise, the CIFAR10-DVS dataset is a spiking version of the CIFAR10 [14] dataset. It contains 10000 128×128 samples, and 1000 samples per class, corresponding to 10 classes. The sample size for both the N-MNIST and CIFAR10-DVS datasets is represented as $T \times P \times H \times W$, where T is the time steps (we set $T = 18$ in the experiments), P is the polarity, H is the height, and W is the width.

We employed distinct network architectures for the two datasets. For the N-MNIST dataset, the network comprises two convolutional layers followed by batch normalization and max pooling layers, then two linear layers with dropout, concluding with a voting layer aimed at enhancing classification robustness. In the case of the CIFAR10-DVS dataset, the network consists of four convolutional layers incorporating batch normalization and max pooling layers, along with two linear layers incorporating dropout, and similarly concludes with a voting layer.

4.2 Experiment Setup

We used SpikingJelly [9] framework to implement the SNN model and partition the N-MNIST and CIFAR10-DVS datasets into T=18 frames. Within FedNL, we employed the Adam optimizer with a local learning rate l_r and batch size of B for training over E local epochs.

Following the multiple-shot attack setup of Bagdasaryan et al. [5], the attackers need to undergo multiple rounds of selection, and the accumulation of malicious updates is necessary for the success of the attack. Otherwise, the backdoor will be weakened by benign updates and quickly forgotten by the global SNN model.



(a) Static Trigger (b) Moving Trigger

Fig. 5: Spikewhisper and Temporally Centralized Attacks (TCA) on the N-MNIST dataset

To expedite the convergence speed of backdoor learning and quickly observe the distinctions between temporally centralized attacks and Spikewhisper, we consistently select attackers in each training round. Then we randomly choose benign participants to form a total of 10 participants. Furthermore, we expedite the training speed by employing IID distribution to allocate the neuromorphic datasets among a total of 50 participants. To ensure training stability, malicious participants engage in backdoor training after a certain number of benign training rounds (10 for N-MNIST, and 25 for CIFAR10-DVS).

In our experiments, we employ the same static and moving global triggers to assess the attack success rates of Spikewhisper and temporally centralized attacks. To ensure a fair comparison, we make certain that the total number of backdoor trigger pixels for Spikewhisper attackers is identical to that of temporally centralized attackers.

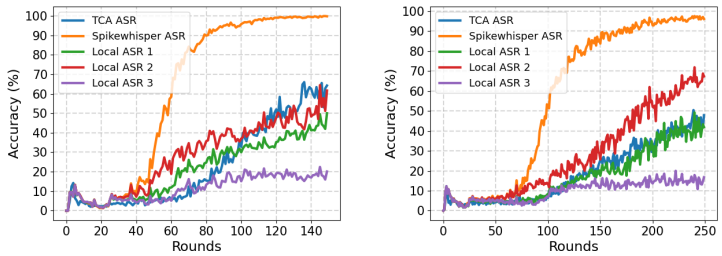
4.3 Evaluation Metrics

We evaluate Spikewhisper and Temporally Centralized Attacks with the commonly used metrics:

- **Attack Success Rate (ASR)** represents the percentage of attacked samples that the compromised model successfully predicts as the desired target label.
- **Main Task Accuracy** signifies the precision with which the infected model predicts benign test samples.

4.4 Experiment Result

The experiment conducted involved attacks using both static and moving triggers on the N-MNIST and CIFAR10-DVS datasets, evaluating the impact on federated neuromorphic learning through multiple rounds of SNN model aggregation. The results demonstrate the efficacy of Spikewhisper and the comparative performance of Spikewhisper versus the centralized approach.



(a) Static Trigger (b) Moving Trigger

Fig. 6: Spikewhisper and Temporally Centralized Attacks (TCA) on the CIFAR10-DVS dataset

As shown in Fig. 5, for the N-MNIST dataset, Spikewhisper with static trigger exhibited a rapid escalation in the attack success rate (ASR) for the global trigger, surpassing 99% by the 42nd round. In contrast, the local static triggers showed significantly lower ASRs at 0.49%, 1.99%, and 1.27%, respectively. The centralized static attack yielded a relatively low ASR of 2.09% at this point, highlighting the inferiority compared to Spikewhisper. Spikewhisper with moving trigger on N-MNIST also achieved high ASRs for the global trigger, exceeding 99% by the 46th round. The centralized attack achieved an ASR of only 12.46% in the final round, indicating a failure to establish a backdoor in the global SNN model, while Spikewhisper reached 100% ASR.

Switching to the CIFAR10-DVS dataset as shown in Fig. 6, Spikewhisper with static trigger also demonstrated a rapid rise in the ASR of the global trigger, exceeding 99% by the 120th round, with local static triggers exhibiting varying low ASRs. The temporally centralized attack lagging behind at 50% ASR at this point. The moving trigger attacks on CIFAR10-DVS were more challenging, with a slower growth rate in ASR compared to static triggers. In Spikewhisper, the global moving trigger reached 95% ASR by the 190th round, with the ASR for the temporally centralized attack less than 40%. Even in the final round, the temporally centralized attack achieved an ASR of only 47.89%, emphasizing the effectiveness of Spikewhisper with its higher ASR.

Table 1: Main Task Accuracy (%)

Attack Types	N-MNIST	CIFAR10-DVS
Baseline (No Attack)	98.96	64.30 (150 Round) 65.60 (250 Round)
Spikewhisper w/ Static Trigger	98.87	62.20 (150 Round)
Temporally Centralized Attack w/ Static Trigger	98.99	62.80 (150 Round)
Spikewhisper w/ Moving Trigger	98.76	64.30 (250 Round)
Temporally Centralized Attack w/ Moving Trigger	98.92	65.50 (250 Round)

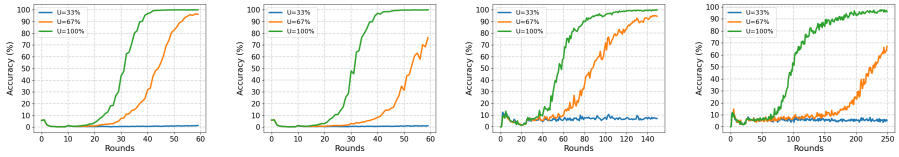
As for the impact of Spikewhisper and temporally centralized attacks on main task accuracy, please refer to Table 1. From the table, we can observe that both Spikewhisper and temporally centralized attacks have almost negligible effects on the accuracy of the main task. The decrease in main task accuracy caused by Spikewhisper does not exceed 2.1%, while the accuracy under temporally centralized attacks at times, even surpasses the baseline due to fluctuation.

In summary, Spikewhisper successfully injected a backdoor into the global SNN model, demonstrating its potential threat to federated neuromorphic learning systems. By implementing Spikewhisper on the N-MNIST and CIFAR10-DVS datasets, we achieved attack success rates (ASRs) of over 99%, with little impact on the accuracy of the main task. Compared to temporally centralized attacks, the ASR of Spikewhisper was significantly higher in all experimental instances. In the N-MNIST dataset, even with local trigger attack success rates consistently below 10%, the global trigger achieved a 99% ASR. This indicates that temporally centralized attacks are inefficient in federated neuromorphic learning.

4.5 Ablation Study

In this subsection, we investigated the impact of the temporal length of Spikewhisper triggers on the backdoor effects based on ASRs. Additionally, we experimented with the attack performance of Spikewhisper in the Non-IID setting, further exploring the generality of Spikewhisper.

Temporal Utilization of Trigger In the preceding experiment setup, each data sample consists of $T = 18$ frames. The global trigger also spans 18 frames, distributing to three local triggers, each lasting for 6 frames, achieving a 100% temporal utilization rate.



(a) Static Trigger on N-MNIST (b) Moving Trigger on N-MNIST (c) Static Trigger on CIFAR10-DVS (d) Moving Trigger on CIFAR10-DVS

Fig. 7: Effects of the temporal utilization rate U on Attack Success Rate of Spikewhisper. $U = 0\%$ means no data poisoning, and $U = 100\%$ means each local trigger persists throughout the entire 6 frames, collectively forming an 18 frames global trigger.

We maintain the presence of three malicious participants, inserting triggers in the first, middle, and last thirds of the data frames (each comprising 6 frames).

$U = 0\%$ indicates no insertion of local triggers, meaning no data poisoning. U=33% and U=67% respectively represent inserting 2 frames and 4 frames of local triggers per timeslice, and $U = 100\%$ signifies that each local trigger persists throughout the entire 6 frames, collectively forming an 18 frames global trigger.

The experimental results of two triggers on two neuromorphic datasets are shown in Fig. 7. It can be observed that all four plots exhibit similar phenomena, where the backdoor effect of Spikewhisper gradually strengthens with the increase of U. When $U = 33\%$, both trigger types on the two datasets fail to inject backdoors into the global SNN model, resulting in minimal backdoor effects. When $U = 67\%$, successful backdoor effects can be achieved, but more rounds are required to fully inject the backdoor into the global SNN model. When $U = 100\%$, the injection speed of the backdoor reaches its maximum, and the number of rounds required for backdoor learning convergence is also minimized.

Based on the experimental analysis of the impact of U on the backdoor effect, it can be concluded that the longer the duration of the trigger, the more likely the success of the attack in the federated neuromorphic learning scenario. This success leads to the injection of a backdoor into the global SNN model.

Trigger Size and Location In backdoor attacks on DNNs, the size and location of the trigger can significantly impact the attack effectiveness. To identify the difference of Spikewhisper against these backdoors in DNNs, in Section 4.5, we observe that Spikewhisper is only sensitive to the temporal duration of the trigger, which is a unique feature under FedNL. In this section, we will explore whether the size and location of the trigger similarly affect the effectiveness of Spikewhisper.

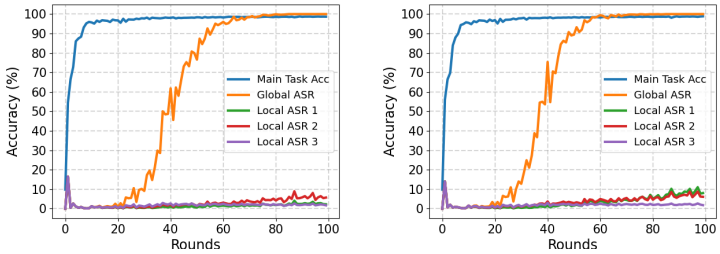
Table 2: ASR of Spikewhisper with different trigger sizes and locations on N-MNIST.

Trigger size	Position	Type	ASR (60 Rounds)
1×1	bottom-right	static	0.28%
1×1	bottom-right	moving	0.90%
2×2	bottom-right	static	56.44%
2×2	bottom-right	moving	87.96%
3×3	bottom-right	static	100.00%
3×3	bottom-right	moving	100.00%
3×3	middle	static	0.26%
3×3	middle	moving	0.26%
3×3	top-left	static	100.00%
3×3	top-left	moving	99.91%

Under different trigger sizes and position configurations, the ASR of Spikewhisper on the N-MNIST is presented in Table 2. Under 60 rounds, it can be observed that the size and position of triggers significantly influence the backdoor performance of Spikewhisper. In terms of size, the ASR increases sequentially as

the trigger size grows. Interestingly, in previous experiments, we observed that the backdoor convergence speed with moving triggers was slower than that of static triggers. However, in the case of smaller triggers, we found that moving triggers could induce a stronger backdoor effect. Regarding position, we moved the trigger from its original position in the bottom right corner of the data along the diagonal. We tested three positions: bottom-right, middle, and top-left. It can be seen that triggers at corners exhibit excellent ASR, while those in the middle perform poorly. We attribute this to the fact that the main body of N-MNIST samples is located in the middle of the images, affecting the effectiveness of triggers in the middle.

Non-IID Scenario In real-world application scenarios of federated learning, the data distribution among each participant is often non-independent and identically distributed (Non-IID) [17]. Therefore, we also evaluate the Spikewhisper's performance under Non-IID settings to validate its practicality.



(a) Static Trigger (b) Moving Trigger

Fig. 8: Spikewhisper on N-MNIST in Non-IID Scenario.

To evaluate the performance of Spikewhisper in the Non-IID scenario, we utilized a Dirichlet distribution [19] with $\alpha = 0.5$ hyperparameter to partition the 60,000 training samples of the N-MNIST dataset.

It can be observed that in the Non-IID setting, Spikewhisper successfully injected a backdoor into the global SNN model, whether using the static trigger or the moving trigger. In comparison to the IID setting, the communication rounds required for the ASR to reach 99% increased from around 40 rounds to approximately 60 rounds. However, the consistent phenomenon remains: when Spikewhisper successfully injects the backdoor, the ASR for each local trigger remains at an extremely low level.

5 Conclusions & Future Work

This paper aims to investigate a novel temporal spike backdoor attack named Spikewhisper in FedNL over low-power devices, particularly utilizing the dis-

tributed nature of federated learning and the temporal characteristics of spiking neural networks with the concept of time division multiplexing. We evaluate Spikewhisper using static and moving triggers on two different neuromorphic datasets: 1) N-MNIST and 2) CIFAR10-DVS. The results indicate that Spikewhisper outperforms temporally centralized attacks significantly, achieving an attack success rate of over 99%. We study the temporal duration of triggers in Spikewhisper, revealing that the more frames the triggers occupy, the stronger the resulting backdoor effect, facilitating faster injection of backdoors into the global SNN model. Furthermore, we explore the impact of trigger size and location on Spikewhisper. Lastly, we also evaluate the performance of Spikewhisper in the Non-IID setting. Our research indicates that federated neuromorphic Learning is susceptible to temporal spike backdoor attacks (Spikewhisper), yet there is currently a lack of dedicated backdoor defense measures in this domain. The development of corresponding defense strategies specifically targeting spiking neural networks and neuromorphic data in FedNL is poised to become a crucial direction for future research.

References

1. Abad, G., Ersoy, O., Picek, S., Ramírez-Durán, V.J., Urbieta, A.: Poster: Backdoor attacks on spiking nns and neuromorphic datasets. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 3315–3317 (2022)
2. Abad, G., Ersoy, O., Picek, S., Urbieta, A.: Sneaky spikes: Uncovering stealthy backdoor attacks in spiking neural networks with neuromorphic data. In: NDSS (2024)
3. Abad, G., Picek, S., Urbieta, A.: Time-distributed backdoor attacks on federated spiking learning. arXiv preprint arXiv:2402.02886 (2024)
4. Ansari, M.S., Alsamhi, S.H., Qiao, Y., Ye, Y., Lee, B.: Security of distributed intelligence in edge computing: Threats and countermeasures. The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing pp. 95–122 (2020)
5. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International conference on artificial intelligence and statistics. pp. 2938–2948. PMLR (2020)
6. Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S.: Analyzing federated learning through an adversarial lens. In: International Conference on Machine Learning. pp. 634–643. PMLR (2019)
7. Dhar, P.: The carbon impact of artificial intelligence. Nat. Mach. Intell. **2**(8), 423–425 (2020)
8. Eshraghian, J.K., Ward, M., Neftci, E.O., Wang, X., Lenz, G., Dwivedi, G., Bennamoun, M., Jeong, D.S., Lu, W.D.: Training spiking neural networks using lessons from deep learning. Proceedings of the IEEE (2023)
9. Fang, W., Chen, Y., Ding, J., Yu, Z., Masquelier, T., Chen, D., Huang, L., Zhou, H., Li, G., Tian, Y.: Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. Science Advances **9**(40), eadi1480 (2023)
10. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks.

In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2661–2671 (2021)
11. Gerstner, W., Kistler, W.M.: Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press (2002)
12. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
13. Kim, Y., Panda, P.: Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. Frontiers in neuroscience **15**, 773954 (2021)
14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
15. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
16. Li, H., Liu, H., Ji, X., Li, G., Shi, L.: Cifar10-dvs: an event-stream dataset for object classification. Frontiers in neuroscience **11**, 309 (2017)
17. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). pp. 965–978. IEEE (2022)
18. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 691–706. IEEE (2019)
19. Minka, T.: Estimating a dirichlet distribution (2000)
20. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. IEEE Signal Processing Magazine **36**(6), 51–63 (2019)
21. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience **9**, 437 (2015)
22. Skatchkovsky, N., Jang, H., Simeone, O.: Federated neuromorphic learning of spiking neural networks for low-power edge intelligence. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8524–8528. IEEE (2020)
23. Venkatesha, Y., Kim, Y., Tassiulas, L., Panda, P.: Federated learning with spiking neural networks. IEEE Transactions on Signal Processing **69**, 6183–6194 (2021)
24. 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. Advances in Neural Information Processing Systems **33**, 16070–16084 (2020)
25. Wang, Y., Duan, S., Chen, F.: Efficient asynchronous federated neuromorphic learning of spiking neural networks. Neurocomputing **557**, 126686 (2023)
26. Xie, C., Huang, K., Chen, P.Y., Li, B.: Dba: Distributed backdoor attacks against federated learning. In: 8th International Conference on Learning Representations, ICLR 2020 (2020)
27. 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 Transactions on Vehicular Technology **71**(9), 9980–9992 (2022)
28. Yang, H., Lam, K.Y., Xiao, L., Xiong, Z., Hu, H., Niyato, D., Vincent Poor, H.: Lead federated neuromorphic learning for wireless edge artificial intelligence. Nature communications **13**(1), 4269 (2022)