To appear at the 24th Design, Automation and Test in Europe (DATE'21), February, 2021.

Securing Deep Spiking Neural Networks against Adversarial Attacks through Inherent Structural Parameters

Rida El-Allami[1,*], Alberto Marchisio[2,*], Muhammad Shafique[3], Ihsen Alouani[1]

[1] *IEMN CNRS-UMR8520, Université Polytechnique Hauts-De-France, Valenciennes, France*

[2] *Institute of Computer Engineering, Technische Universität Wien, Vienna, Austria*

[3] *Division of Engineering, New York University Abu Dhabi, UAE*

Email: rida.elallami@etu.uphf.fr, alberto.marchisio@tuwien.ac.at, muhammad.shafique@nyu.edu, ihsen.alouani@uphf.fr

***Abstract*—Deep Learning (DL) algorithms have gained popularity owing to their practical problem-solving capacity. However, they suffer from a serious integrity threat, i.e., their vulnerability to adversarial attacks. In the quest for DL trustworthiness, recent works claimed the inherent robustness of Spiking Neural Networks (SNNs) to these attacks, without considering the variability in their structural spiking parameters. This paper explores the security enhancement of SNNs through internal structural parameters. Specifically, we investigate the SNNs robustness to adversarial attacks with different values of the neuron's firing voltage thresholds and time window boundaries. We thoroughly study SNNs security under different adversarial attacks in the strong white-box setting, with different noise budgets and under variable spiking parameters. Our results show a significant impact of the structural parameters on the SNNs' security, and promising sweet spots can be reached to design trustworthy SNNs with 85% higher robustness than a traditional non-spiking DL system. To the best of our knowledge, this is the first work that investigates the impact of structural parameters on SNNs robustness to adversarial attacks. The proposed contributions and the experimental framework is available online[1] to the community for reproducible research.**

***Index Terms*—SNN, Spiking Neural Networks, Security, Machine Learning, Deep Learning, Neuromorphic, Adversarial Attacks, Robustness, Parameters, Optimization, Analysis.**

I. Introduction

The recent advances in Deep Neural Networks (DNNs) have made them the de-facto standard algorithm for several Machine Learning (ML) applications in sectors such as finance, robotics, healthcare and computer vision [1]. However, the trustworthiness of DNNs is threathened by adversarial attacks [2, 3, 4]. A malicious actor is able to jeopardize DNNs integrity with a small and imperceptible input perturbation. In safety-critical applications (e.g., automotive [5], medicine [6], banking [7]), these attacks can cause catastrophic consequences and considerable losses. For example, misclassifying a *stop* traffic sign as a *speed limit* sign could lead to material and human damages. Another scenario is in financial transactions and automatic bank check processing: automatic handwritten character reading of bank checks [8]. An attacker could easily fool the model to predict wrong bank account numbers or wrong amount of money.

On the other hand, Spiking Neural Networks (SNNs) recently emerged as an attractive alternative, due to their biological plausibility and similarity to the human brain's functionality [9]. Moreover, neuromorphic hardware can exploit the asynchronous communication between neurons and the event-based propagation of the information through layers to achieve high energy-efficiency [10, 11]. These characteristics led to an increasing interest in developing neuromorphic architectures such as IBM TrueNorth [12] and Intel Loihi [13]. More interestingly from a security perspective, recent studies claimed inherent robustness of SNNs compared to traditional DNNs [14, 15].

A. Target Research Problem and Research Challenges

While several systematic analyses of the reliability and security of traditional (i.e., non-spiking) DNNs against adversarial attacks have been thoroughly conducted [3, 4, 16, 17, 18, 19, 20], the corresponding analysis of SNNs robustness is still under-explored. Due to their bio-inspired aspect, higher behavioral dimensions are present in SNNs compared to non-spiking DNNs. Therefore, a more comprehensive study is required to understand the inherent behavior of SNNs, especially under adversarial attacks. Towards this, the following key questions need to be investigated:

(Q1) *How do the spiking structural parameters (i.e., threshold voltage and time window[2]) affect the SNNs' behavior under attack?*

(Q2) *Are SNNs inherently robust against adversarial attacks, regardless of the structural parameters?*

(Q3) *Does a combination of structural parameters that provides high accuracy also guarantee high robustness?*

B. Motivational Case Study

Before answering the above-discussed questions, we conducted a motivational case-study to highlight the importance of the problem. We trained a 5-layer Convolutional Neural Network (CNN), with 3 convolutional layers and 2 fully-connected layers on the MNIST dataset [21] using the PyTorch framework [22], and an SNN with the same number of layers and neurons per layer with the Norse framework [23]. We applied the white-box PGD attack [24] on both networks and monitored the accuracy variation w.r.t. the noise budget ε. The results reported in Figure 1 indicate that, while with low noise magnitude the CNN has higher accuracy (pointer ①), after the

*These authors contributed equally to this work.

[1] https://github.com/rda-ela/SNN-Adversarial-Attacks

[2] The *threshold voltage* is defined such that, when the spiking neuron's membrane potential overcomes such threshold, the neuron emits an output spike and resets its membrane potential. The *time window* represents the observation period in which the SNN receives the same input.

turnaround point of $\varepsilon = 0.5$ (pointer ②), the SNN clearly shows a more robust response to the attack than its CNN counterpart, with an accuracy gap higher than 50% (pointer ③). For $\varepsilon > 0.5$, the accuracy of the CNN decreases sharply, while the slope for the SNN is lower. This outcome motivated us to further investigate the inherent robustness of the SNNs.

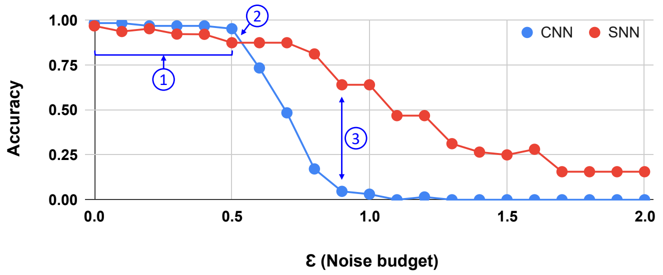

Fig. 1. PGD adversarial attack applied to a CNN and an SNN that have the same number of layers with equal size and equal number of neurons.

These experiments give a quick overview on the high potential of SNNs in terms of security when compared to traditional DNNs. However, while these experiments are run using the default SNN structural parameters, one cannot generalize this observation until a deeper analysis is made.

C. Our Novel Contributions

- We propose a systematic methodology (see Figure 2) for analyzing the SNN robustness. We are first to explore the impact of neurons' structural parameters (i.e., spiking threshold voltage V_{th} and time window boundary T) on the SNNs' robustness against the strong white-box adversarial attacks. **[Section V]**
- The SNN learnability and security studies show that the SNNs' inherent robustness is strongly conditioned by these structural parameters. **[Section VI]**
- We design trustworthy SNNs, by fine-tuning their structural parameters around the previously-found sweet spots. For instance, a 5-layer SNN trained on MNIST dataset achieves up to 84% accuracy improvement compared to a corresponding CNN, when the PGD attack with $\varepsilon = 1.5$ noise budget is applied. **[Section VI-C]**

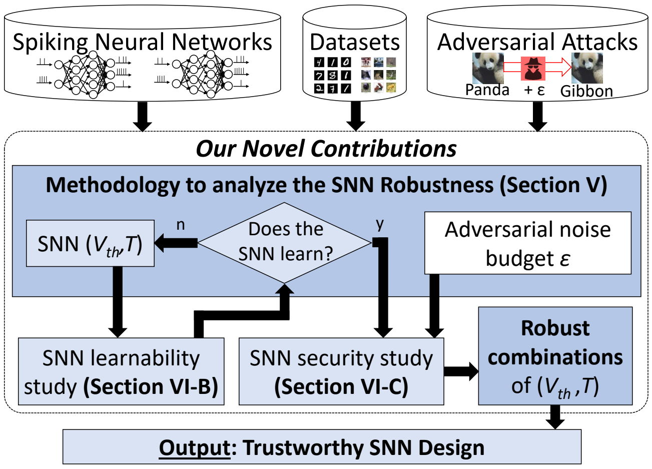
Spiking Neural Networks
Datasets
Adversarial Attacks
Panda
+ ε
Gibbon
Our Novel Contributions
Methodology to analyze the SNN Robustness (Section V)
SNN (V_{th},T)
n
Does the SNN learn?
y
Adversarial noise budget ε
SNN learnability study (Section VI-B)
SNN security study (Section VI-C)
Robust combinations of (V_{th},T)
Output: Trustworthy SNN Design

Fig. 2. Overview of our novel contributions (shown in blue boxes).

- **Open-Source Contribution:** for reproducible research, we release the complete source code of our methodology, including all the examples and robust SNN models, at https://github.com/rda-ela/SNN-Adversarial-Attacks.

II. Background

A. Spiking Neural Networks

SNNs are considered as the third generation of neural networks [25]. Their event-based communication scheme between different neurons is inspired by the human brain's functionality with a higher level of similarity than the (non-spiking) DNNs. Such a bio-inspired computation not only provides biologically-plausible deep learning, but also represents a huge potential for bridging the energy-efficiency gap between the human brain and the supercomputers executing complex deep learning applications, as motivated by the recent advances of neuromorphic architectures, such as IBM TrueNorth[12] and Intel Loihi[13].

An example of the SNNs' modus operandi is illustrated in Figure 3. The input information has to be properly coded using spikes. While other possible coding schemes can be based on the delay between consecutive spikes or the latency between the beginning of the stimulus to the first spike, the most commonly adopted mechanism is the *rate encoding* [9], where the activation intensity corresponds to the mean firing rate over a certain *time window*. Such a time window represents the observation period in which the SNN receives the same input. A wider window gives more time for the spikes to propagate towards the output, but incurs in higher latency. When an incoming spike s_i arrives at the input of the neuron, it is multiplied by its associated synaptic weight w_i and integrated into the membrane potential V, following Equation 1.

$$V = \sum_{i=1}^{N} w_i \cdot s_i \tag{1}$$

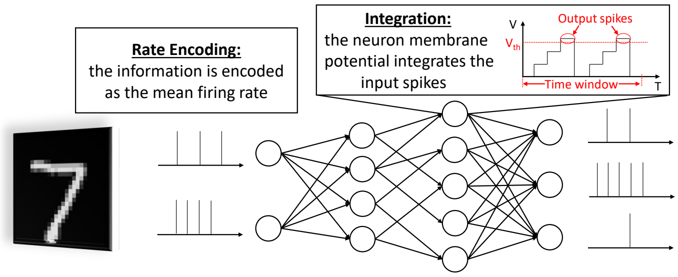
Rate Encoding:
the information is encoded as the mean firing rate
Integration:
the neuron membrane potential integrates the input spikes
V
V_{th}
Output spikes
Time window
T

Fig. 3. Overview of the functionality of an SNN, with a focus on the rate encoding of the information and the integration of the spikes into the membrane potential.

For a Leaky-Integrate-and-Fire (LIF) spiking neuron model, when the membrane potential exceeds the *threshold voltage* V_{th}, the neuron emits an output spike and resets its membrane potential. In this way, the information is propagated to the output. For the example of rate encoding for image classification, the higher the firing rate of the output spike train is, the higher the output probability of the associated class.

Moreover, supervised learning for SNNs is complex, because of the non-differentiability of the SNN loss function [26].

Therefore, the standard backpropagation procedure cannot be applied. To overcome these challenges, there are two possibilities. The first solution consists of training a correspondent DNN with standard backpropagation, and then converting it to the spiking version [27]. However, a certain accuracy is typically lost in the conversion process. Another possibility is to approximate the SNN derivatives and learn based on the temporal information in the spiking domain [28]. *The latter option is adopted in our work.*

B. Adversarial Attacks

DNNs are now deployed in a wide range of sectors, including safety-critical applications such as Intelligent Transportation Systems [29]. Despite their performance, DNNs suffer from a critical challenge, i.e., adversarial attacks. Many studies [2, 4, 19, 24, 30, 31] have shown that DNNs are vulnerable to carefully crafted inputs designed to fool them, very small imperceptible perturbations added to the data can completely change the output of the model.

Minimizing injected noise: It is essential for an attacker to minimize the added adversarial noise to avoid detection. Formally, given an original input x with a target label l with a classification model $f()$, the problem of generating an adversarial example x^* can be formulated as a constrained optimization problem [32], i.e.,

$$\begin{aligned} x^* = \arg\min_{x^*} \quad & \mathcal{D}(x, x^*), \\ s.t. \quad & f(x) = l, \;\; f(x^*) = l^*, \;\; l \neq l^* \end{aligned} \tag{2}$$

Where $\mathcal{D}$ is the distance between two images and the optimization objective is to minimize this adversarial noise to make it stealthy. x^* is considered as an adversarial example if and only if $f(x) \neq f(x^*)$ and the noise is bounded ($\mathcal{D}(x, x^*) < \epsilon$, where $\epsilon \geqslant 0$).

III. Related Work

Recent studies analyzed the SNNs' robustness. Schuman et al. [33] analyzed the resilience of SNNs to faults, varying the training method. In NeuroAttack [30], the impact of the bit-flips on the accuracy of SNNs was studied. The adversarial perturbation was introduced to fire a target neuron to trigger a hardware Trojan.

Adversarial attacks have been tested on SNNs. Bagheri et al. [34] studied the sensitivity of SNN w.r.t. different types of encoding, when subjected to white-box adversarial attacks. Marchisio et al. [14] applied black-box adversarial attacks to DNNs and SNNs, and the comparison showed that the SNNs are more robust. Sharmin et al. [15] proposed a methodology to perform the adversarial attack on (non-spiking) DNNs, and then the DNN-to-SNN conversion made the adversarial examples craft the SNNs. Liang et al. [35] proposed a gradient-based adversarial attack methodology for SNNs, and showcased the impact of the adversarial attack success rate on the type of loss function and threshold voltage of the second-last layer only. The work of [36] analyzed the adversarial accuracy of SNNs trained with different inference latency and leak factor in LIF spiking neurons. However, this work did not explore the impact of membrane voltage threshold along with the time window. *Note, in contrast to the work in [36], we explore the impact of spiking parameters of the neurons on SNNs robustness to question the generalization of inherent robustness observation.*

Recently, Massa et al. [27] tuned the threshold voltage and the time window with the purpose of minimizing the accuracy loss in the DNN-to-SNN conversion process. This work did not study robustness to adversarial attacks. DIET-SNN [37] proposed to tune the membrane threshold and membrane leak to jointly optimize the accuracy and the latency. *However, none of these prior works analyzed the impact of structural parameters, i.e., membrane threshold and time window, from the security perspective, as we target in this paper.*

IV. Threat Model

A. Adversary Knowledge

In our experiments, we assume the strongest case where an attacker is attempting to design adversarial attacks to fool a SNN classifier in a white-box attack scenario. In fact, we assume a powerful attacker who has the full knowledge of the victim classifier's architecture and parameters (including the structural parameters V_{th} and T). The attacker uses this knowledge to create adversarial examples. Figure 4 gives an overview of the attack scenario.

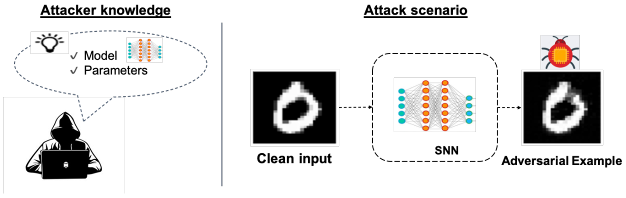

Fig. 4. White-box setting: The attacker has full access to the SNN structure, input & output, hyper parameters, weights, and even structural parameters.

B. Attack Generation

We evaluate the SNNs robustness using one of the most widely used attacks, namely PGD [24]. It is one of the strongest iterative variant of the FGSM where the adversarial example is generated as follows:

$$x^{t+1} = \mathcal{P}_{\mathcal{S}_x}(x^t + \alpha \cdot sign(\nabla_x \mathcal{L}_\theta(x^t, y))) \tag{3}$$

Where $\mathcal{P}_{\mathcal{S}_x}()$ is a projection operator projecting the input into the feasible region $\mathcal{S}_x$ and α is the additive noise at each iteration. PGD attack tries to find the perturbation that maximizes the loss of a model on a given sample while keeping the perturbation magnitude lower than a given budget. PGD is an iterative gradient-based attack that is considered as a high-strength attack [3, 4].

V. Proposed Methodology

Our study aims at the exploration of SNNs' robustness under different adversarial noise budgets, and this, for different (V_{th}, T) parameters combinations. Figure 5 gives an overview of different components of our methodology. It is composed of the following two main steps. **(1)** The first step of the exploration is meant to exclude combinations that are not propitious for efficient learning in SNNs. There is indeed no interest in studying the robustness of SNNs with low baseline performance. **(2)** In the second step, for all (V_{th}, T) settings that enable the SNN training to converge efficiently, we proceed to a robustness exploration.

Algorithm 1 details our robustness exploration methodology. Line 1 and 2 browse the n threshold voltages and m time windows to explore. Once the training is launched (Line 3), we proceed to the SNN learnability study for the given combination (V_{th}, T). As shown in Line 4, the learnability is quantitatively verified by setting a minimum baseline accuracy level below which we consider the SNN learning inefficient. This value depends on the given SNN architecture, its learning method, the dataset and the application. In our case study, we use this accuracy threshold equal to 70% as it is typically achieved by state-of-the-art SNNs. The security analysis starts from Line 5; it consists of generating adversarial examples with different noise budgets to fool the SNN. The noise budget models the aggressiveness allowed within the attack generation; the higher the noise budget, the more aggressive the attack is considered. First, the counter of successful attack generation cases is initialized (Line 6). Then, we browse the dataset $\mathcal{D}$ (Line 7) to generate the adversarial attacks using PGD, as shown in Line 8. Afterwards, the algorithm verifies if the generated example is able to fool the SNN (Lines 9 - 13), i.e., if the attack succeeded to force the output to a wrong label, and accordingly increment the adversarial success counter. Then, the robustness is then evaluated for every ε value as the rate of attacks for which the adversary failed to generate an adversarial example that fools the victim SNN (Line 15). Hence, by tracking the accuracy slope with regard to ε, we can compare the robustness of each model to adversarial attacks.

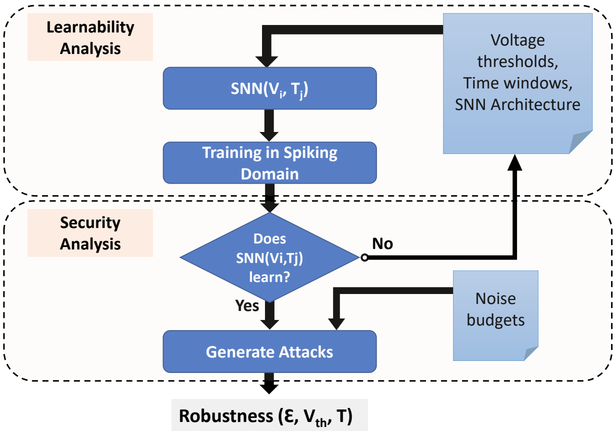

Fig. 5. A flowchart illustrating the key steps of SNN robustness exploration.

VI. Evaluation

A. Experimental Setup

Our experiments are performed using Norse [23] which is a library that expands PyTorch [22] with primitives for bio-inspired neural components, thereby allowing to train and run SNNs in the spiking domain. The adversarial attacks are implemented using Foolbox v3.1.1 [38]. The SNN architecture is a Lenet-5 adapted to the spiking domain, and trained on the

Algorithm 1: Robustness Exploration Algorithm.

```
Result: Robustness Level
Data:
Membrane Voltage Thresholds: $Vth = V_i \ /i \in [1, n]$;
Spiking Time Windows: $T = T_j / j \in [1, m]$;
Adversarial Noise Budgets: $\varepsilon = \varepsilon_k \ /k \in [1, p]$;
SNN Architectures: $S_{ij} = SNN(V_i, T_j)$;
Labeled Test Set: $\mathcal{D} = (X_t, L_t)$ ;
Accuracy threshold: $A_{th}$
1  for $i \leftarrow 1$ to $n$ do
2      for $j \leftarrow 1$ to $m$ do
3          Train($S_{ij} = SNN(V_i, T_j)$);
4          if Accuracy($S_{ij}$) $\geq A_{th}$ then
               // $S_{ij}$ learns
5              for $k \leftarrow 1$ to $p$ do
6                  Adv = 0;
7                  for $X_t \leftarrow 1$ to $< \mathcal{D} >$ do
                       // Adversarial Attack
8                      $X_t^* = PGD(S_{ij}, \varepsilon_k, X_t)$;
9                      if $S_{ij}(X_t^*) \neq L_t$ then
10                         Adv++ ;
11                     else
12                         NOP;
13                     end
14                 end
15                 Robustness ($\varepsilon_k$)= $1 - \frac{Adv}{<\mathcal{D}>}$;
16             end
17         else
18             NOP;
19         end
20     end
21 end
```

MNIST database [21]. Note: unlike DNNs, SNNs are still in a relatively early phase of adoption. We adopt the same test conditions as widely used by the SNN research community where the typical evaluation settings [39] use datasets like MNIST and Fashion MNIST. The used neuron model is the Leaky-Integrate-and-Fire (LIF) Neuron and the experiments were run on an Nvidia TESLA P100 GPU with a 16GB memory.

B. Learnability Study

Before studying the robustness with respect to varying the structural parameters, we need to define our exploration space. In fact, the default values of the threshold voltage and time window parameters are $(V_{th}, T) = (1, 64)$. Therefore, we focus on having an overview of the learnability of SNNs in the neighborhood of this setting. Figure 6 shows the accuracy heat map for different (V_{th}, T) combinations. The horizontal and vertical axes denote V_{th} and T, respectively. Different colors denote the accuracy of the SNN. Note from pointer ① of Figure 6 that the highest-accuracy combination tends to be towards the top-left corner, i.e., low V_{th} and high T. However, the heat map is clearly not monotonic. For example, as indicated by pointer ②, there are combinations with an accuracy lower than 16% which are surrounded by combinations with accuracy higher than 89%.

While it is obvious that studying robustness for the non-learnable combination is not useful, we use this map as a reference to track the behavior of SNNs under attack with different noise budgets.

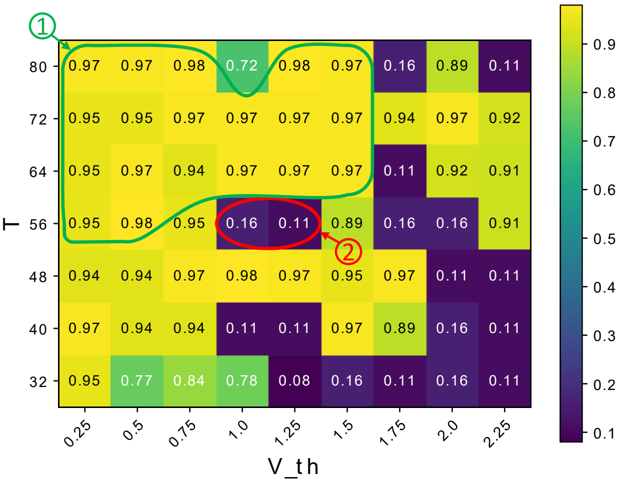

Fig. 6. A heat map showing the accuracy of SNNs trained on MNIST dataset under different combinations of V_{th} and T.

C. Security Study

In this section, we investigate the robustness of SNNs while increasing the attacks' adversarial noise magnitude in a white-box scenario. We first proceed to a holistic exploration under all previous combinations of V_{th} and T. Figures 7 and 8 show the accuracy degradation of SNNs under PGD attack with noise magnitudes of 1 and 1.5, respectively.

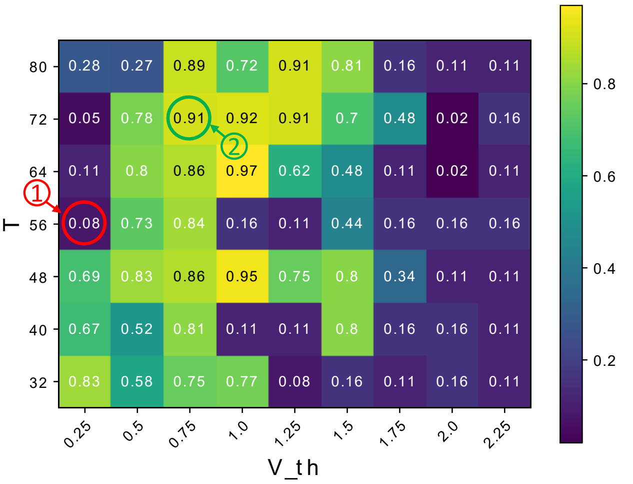

Fig. 7. A heat map showing the accuracy of SNNs trained on MNIST datatset under different combinations of V_{th} and T under PGD attack with $\underline{\varepsilon = 1}$.

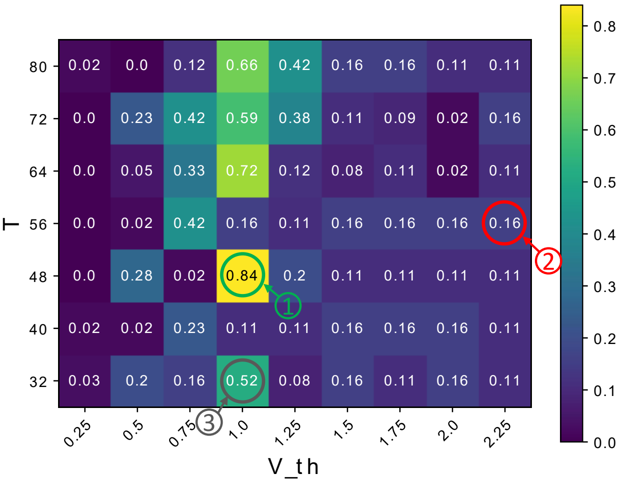

Fig. 8. A heat map showing the accuracy of SNNs trained on MNIST dataset under different combinations of V_{th} and T under PGD attack with $\underline{\varepsilon = 1.5}$.

The first interesting insight that we extract from Figures 7 and 8 is that, high baseline learnability (without adversarial attacks) is not a guarantee of robustness. Moreover, we notice a different evolution of the SNNs w.r.t. adversarial attacks based on their respective structural parameters. More specifically, two SNNs with a starting comparable accuracy may have different behaviors under attack. For example, combinations $(V_{th}, T) = (0.25, 56)$ and $(V_{th}, T) = (0.75, 72)$ start with respective accuracy of 95% and 97%. However, while the accuracy of the first combination (pointer ① of Figure 7) drops drastically to 8% under $\varepsilon = 1$ attack budget, the second (pointer ②) looses only 6% of its initial accuracy under the same attack noise magnitude. Moreover, from Figure 8 we can observe different types of robustness to the attack:

1) $(V_{th}, T) = (1, 48)$: high robustness
2) $(V_{th}, T) = (2.25, 56)$: low robustness
3) $(V_{th}, T) = (1, 32)$: medium robustness

In the following, we track a set of insightful (V_{th}, T) combinations and track their impact on SNNs robustness compared to the Lenet-5 CNN trained on the same dataset. Figure 9 compares the robustness of SNNs with different structural parameters w.r.t. its correspondent CNN. This figure shows, in a more detailed fashion, the impact of structural parameters on SNNs' security. In fact, while the combination $(V_{th}, T) = (2.25, 56)$ achieves lower robustness than the CNN, up to 85% higher robustness is reached by the combination $(V_{th}, T) = (1, 48)$. Another interesting case is represented by the combination $(V_{th}, T) = (1, 32)$, whose clean accuracy is only 78% (see pointer ① in Figure 9), while, as indicated by pointer ②, it has 75% higher accuracy than the CNN when a strong noise budget (i.e., $\varepsilon > 1$) is applied.

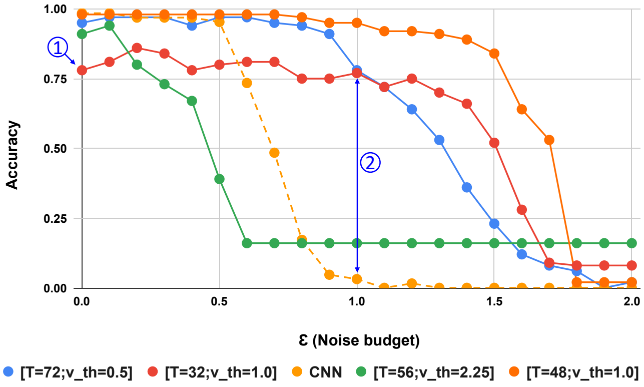

Fig. 9. Robustness of SNNs tested on MNIST with different V_{th} and T parameters under PGD attack compared to the Lenet-5 CNN.

VII. Concluding Remarks

This paper investigated the security of SNNs from a new perspective., i.e., a systematic exploration of the impact of structural parameters of the bio-inspired neurons on the robustness of SNNs to adversarial attacks. Under a strong attack scenario, i.e., white-box setting, we performed extensive security analysis with respect to three variant parameters: spiking voltage threshold (V_{th}), spiking time window (T) and attack's noise budget (ε). By tracking the robustness considering those parameters, we found a high impact of V_{th} and T on the robustness of SNNs. While this work confirms state-of-the-art claims about the inherent SNNs robustness, it generalizes those findings and shows their relativity to spiking structural parameters, which is the first study revealing these valuable insights. In summary, this work answers the three questions raised in the introduction (Section I-A) as follows:

(A1) *Structural parameters* (V_{th}, T) *do have a significant impact on the robustness of SNNs*, and a careful exploration needs to be carried out before deployment in safety-critical or security-sensitive applications.

(A2) Yes, SNNs have inherent robustness against adversarial attacks. *However, this inherent robustness is highly conditioned by the choice of* (V_{th}, T) *combination.*

(A3) No, a combination of (V_{th}, T) parameters that learns efficiently and gives high baseline accuracy is *not* a guarantee of robustness.

These findings consolidate the positioning of SNNs as an interesting solution towards efficient and robust machine learning systems. SNNs' high power efficiency makes them even more interesting, especially for embedded systems and at the Edge.

The observed possible *negative* impact of some parameters combinations *even with high initial accuracy* is a counterexample of the previously assumed unconditional inherent robustness. We believe that this is an interesting finding that can enable more comprehensively secure SNNs design. In future work, we will include deeper networks in our experiments, as we believe that the findings of this paper can be generalized to other SNNs and datasets. More complex behavior might be witnessed, but the very impact of structured parameters on SNNs robustness should be comparable.

Acknowledgment

This work has been partially supported by the Doctoral College Resilient Embedded Systems which is run jointly by TU Wien's Faculty of Informatics and FH-Technikum Wien. This work is also partially supported by Intel Corporation through Gift funding for the project "Cost-Effective Dependability for Deep Neural Networks and Spiking Neural Networks".

References

[1] M. Capra *et al.*, "An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks," *Future Internet*, 2020.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013.

[3] M. Shafique *et al.*, "Robust machine learning systems: Challenges,current trends, perspectives, and the road ahead," *IEEE Design & Test*, 2020.

[4] J. J. Zhang *et al.*, "Building robust machine learning systems: Current progress, research challenges, and opportunities," in *DAC*, 2019.

[5] A. Luckow *et al.*, "Deep learning in the automotive industry: Applications and tools," *2016 IEEE International Conference on Big Data*, 2016.

[6] H. Jang and K. Cho, "Applications of deep learning for the analysis of medical data," *Archives of Pharmacal Research*, 2019.

[7] A. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications : A survey," *ArXiv*, vol. abs/2002.05786, 2020.

[8] M. B. A. Miah, M. A. Yousuf, M. S. Mia, and M. P. Miya, "Handwritten courtesy amount and signature recognition on bank cheque using neural network," *International Journal of Computer Applications*, 2015.

[9] F. Ponulak and A. Kasiński, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta neurobiologiae experimentalis*, 2011.

[10] M. Capra *et al.*, "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead," *IEEE Access*, 2020.

[11] R. V. W. Putra and M. Shafique, "Fspinn: An optimization framework for memory-efficient and energy-efficient spiking neural networks," *IEEE TCAD*, 2020.

[12] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, 2014.

[13] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[14] A. Marchisio *et al.*, "Is spiking secure? a comparative study on the security vulnerabilities of spiking and deep neural networks," in *IJCNN*, 2020.

[15] S. Sharmin *et al.*, "A comprehensive analysis on adversarial robustness of spiking neural networks," in *IJCNN*, 2019.

[16] M. A. Neggaz, I. Alouani, S. Niar, and F. Kurdahi, "Are cnns reliable enough for critical applications? an exploratory study," *IEEE Design Test*, 2019.

[17] M. A. Hanif and M. Shafique, "Salvagednn: salvaging deep neural network accelerators with permanent faults through saliency-driven fault-aware mapping," *Philosophical Transactions of the Royal Society A*, 2019.

[18] L. H. Hoang, M. A. Hanif, and M. Shafique, "Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," in *DATE*, 2020.

[19] F. Khalid *et al.*, "Fademl: Understanding the impact of pre-processing noise filtering on adversarial machine learning," in *DATE*, 2019.

[20] F. Khalid *et al.*, "Qusecnets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks," in *IOLTS*, 2019.

[21] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[22] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[23] C. Pehle and J. E. Pedersen, "Norse: A library to do deep learning with spiking neural networks," *github*, 2019.

[24] A. Madry *et al.*, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[25] W. Maas, "Networks of spiking neurons: The third generation of neural network models," *Trans. Soc. Comput. Simul. Int.*, 1997.

[26] B. Rückauer *et al.*, "Closing the accuracy gap in an event-based visual recognition task," *CoRR*, 2019.

[27] R. Massa, A. Marchisio, M. Martina, and M. Shafique, "An efficient spiking neural network for recognizing gestures with a dvs camera on the loihi neuromorphic processor," in *IJCNN*, 2020.

[28] J. C. Thiele, O. Bichler, and A. Dupret, "Spikegrad: An ann-equivalent computation model for implementing backpropagation with spikes," in *International Conference on Learning Representations*, 2020.

[29] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural networks : the official journal of the International Neural Network Society*, 2012.

[30] V. Venceslai *et al.*, "Neuroattack: Undermining spiking neural networks security through externally triggered bit-flips," in *IJCNN*, 2020.

[31] A. Marchisio *et al.*, "Capsattacks: Robust and imperceptible adversarial attacks on capsule networks," *ArXiv*, vol. abs/1901.09878, 2019.

[32] X. Yuan *et al.*, "Adversarial examples: Attacks and defenses for deep learning," *CoRR*, vol. abs/1712.07107, 2017.

[33] C. D. Schuman *et al.*, "Resilience and robustness of spiking neural networks for neuromorphic systems," in *IJCNN*, 2020.

[34] A. Bagheri, O. Simeone, and B. Rajendran, "Adversarial training for probabilistic spiking neural networks," in *SPAWC*, 2018.

[35] L. Liang *et al.*, "Exploring adversarial attack in spiking neural networks with spike-compatible gradient," *ArXiv*, vol. abs/2001.01587, 2020.

[36] 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," *ArXiv*, vol. abs/2003.10399, 2020.

[37] N. Rathi and K. Roy, "Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks," 2020.

[38] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models," *CoRR*, 2017.

[39] J. M. Allred and K. Roy, "Controlled forgetting: Targeted stimulation and dopaminergic plasticity modulation for unsupervised lifelong learning in spiking neural networks," *Frontiers in Neuroscience*, vol. 14, 2020.