2023 Design, Automation & Test in Europe Conference (DATE 2023) – Best Paper Award Candidate

Assessing Convolutional Neural Networks Reliability through Statistical Fault Injections

A. Ruospo*, G. Gavarini*, C. De Sio*, J. Guerrero*, L. Sterpone*, M. Sonza Reorda*, E. Sanchez*, R. Mariani†, J. Aribido† and J. Athavale†

*Politecnico di Torino, DAUIN, Torino, Italy

†NVIDIA, US

***Abstract*—Assessing the reliability of modern devices running CNN algorithms is a very difficult task. Actually, the complexity of the state-of-the-art devices makes exhaustive Fault Injection (FI) campaigns impractical and typically out of the computational capabilities. A possible solution consists of resorting to statistical FI campaigns that allow a reduction in the number of needed experiments by injecting only a carefully selected small part of it. Under specific hypothesis, statistical FIs guarantee an accurate picture of the problem, albeit selecting a reduced sample size. The main problems today are related to the choice of the sample size, the location of the faults, and the correct understanding of the statistical assumptions. The intent of this paper is twofold: first, we describe how to correctly specify statistical FIs for Convolutional Neural Networks; second, we propose a data analysis on the CNN parameters that drastically reduces the number of FIs needed to achieve statistically significant results without compromising the validity of the proposed method. The methodology is experimentally validated on two CNNs, ResNet-20 and MobileNetV2, and the results show that a statistical FI campaign on about 1.21% and 0.55% of the possible faults, provides very precise information of the CNN reliability. The statistical results have been confirmed by the exhaustive FI campaigns on the same cases of study.**

***Index Terms*—Statistical Fault Injection, Convolutional Neural Network, Reliability, Fault Injection**

I. Introduction

Nowadays, Deep Neural Networks (DNNs), and particularly Convolutional Neural Networks (CNNs), represent one of the most used solutions for addressing complex computational problems. Validating safety requirements of modern computing systems leveraging DNN algorithms is today a major concern in industry and academia. To safely deploy them in safety-critical systems, there is an urgent need of understanding their resilience against the occurrence of hardware faults [1], [2]. Parallel to reliability assessment issues, the complexity of these models and their memory requirements are also non-negligible.

As complexity increases, the computational effort required to perform the reliability assessment becomes unmanageable. Fault Injection (FI) campaigns have been accepted as valid assessment methodologies for DNNs; however, validating the safety properties by exhaustively fault simulating a DNN is typically out of the computational possibilities. Recently, a widely used technique adopted to determine DNN resilience consists in performing FIs on static parameters (weights), and checking their behaviour in response to the occurrence of faults [3], [4]. Indeed, when they are deployed on hardware devices, being read-only variables, weights and static data are stored in memories, which are the highest contributor of soft errors in the system [5], [6], in the case not additional mechanisms such as error correction code are present in the device. As mentioned, the costs for performing exhaustive FIs on DNNs are typically prohibitive as the complexity and the size of newer DNNs grow. Understanding the minimal number of experiments that a designer must perform to get significant results is currently one of the main issue. To address this problem, statistical approaches have been proposed over the past decades with the intent of reducing the cost of the fault simulation procedure while still achieving significant results (i.e., fault sampling) [7]–[13]. Among these, only [9] validates the proposed statistical method with exhaustive results, but none of these apply specifically to DNNs. Nevertheless, statistical injections are widely used in the research community also to perform reliability assessments on DNNs (e.g., [14], [15]). In particular, the gathered results are elaborated to identify the DNN criticalities, for example, the most critical layer, the most critical bit in the DNN weights, and so on. In this work, we experimentally demonstrate that fault sampling is an effective solution *if and only if* the statistical hypothesis and constraints are met and correctly applied.

This research work presents two main contributions. First, it presents a methodology to perform Statistical Fault Injections (SFIs) on CNNs, by defining not only *how many* faults need to be injected (i.e., the sample size), but also *where* they should be placed to achieve statistically significant results. Additionally, it describes a methodology to measure the probability of a fault to become a critical failure starting from the probability distribution of the golden data representation (the DNN weights). The proposed statistical approach is validated by comparing the results with exhaustive FI campaigns. Two CNN topologies are used: ResNet-20 and MobileNetV2, trained and tested on CIFAR-10. The rest of the paper is organized as follows: section II provides the reader with background knowledge about statistical inference; it details the motivations behind the research work, and then presents related studies. Section III describes the proposed approach and Section IV outlines the case study. Next, Section V reports on the experimental results. Finally, Section VI draws conclusions.

II. Statistical Background

In statistics, the term population (N) refers to a set of measurements and is typically described by the distribution of its values [16]. Since the investigation of the entire population's characteristics is typically very difficult, it is usually necessary

978-3-9819263-7-8/DATE23/© 2023 EDAA

to observe a sample (n) which is a representation of the population. The branch of statistics that deals with generalizing from a sample is also referred to as *statistical inference*. Noteworthy, when a sample n is used to estimate the mean (μ) and the variance (σ^2) of a population N, the probability that the mean μ_{x} and the variance ${\sigma^2}_{\mathrm{x}}$ of the estimation x will be equal to μ and σ^2 of the population are slim. It is possible to compute the maximum error of the estimate, which, for finite populations, must be adjusted by applying the finite population correction factor. Starting from the maximum error of the estimate, and considering specific assumptions (detailed in the following), it is possible to derive the sample size n.

$$e = t * \frac{\sigma}{\sqrt{n}} \longrightarrow n = \frac{N}{1 + e^2 \cdot \frac{N-1}{t^2 \cdot p*(1-p)}} \tag{1}$$

In (1), σ represents the standard deviation, n the sample size, e the desired error margin, and t is a constant that depends on the desired confidence level. N corresponds to the actual size of the population and p to the probability of an event to success. Being a probability, p assumes values between 0 and 1.

A. Motivations

This subsection discusses the applicability of (1) to CNNs.

In the FI context, the term population (N) is used to indicate the total number of possible faults in a system. The term sample (n) is adopted to indicate the subset of random faults that must be injected in a system to extract the characteristics of the entire population. Typically, $n << N$. How the sample is selected, as well as its size, are the focus of the science of statistical sampling. In practice, a FI process is merely a set of repeated trials n, where we are interested in the probability of getting x successes in n trials. The reader should notice that, in the context of FIs, a trial is a success when a fault becomes a critical failure. Since the number of trials is finite, x is a realization of a discrete random variable X that follows a binomial distribution $X \sim B(n, p)$, where p is the probability of success of each trial. X can take on values between 0 and n. According to the Central Limit Theorem, when n is large enough, a binomial distribution can be approximated by a normal distribution [16]. Additionally, the variance of a binomial distribution with parameters n and p is given by:

$$\sigma^2 = n * p * (1 - p) \tag{2}$$

If replacing (2) in (1), and applying the finite population correction factor, n is obtained. Noteworthy, each single trial in a binomial distribution is a Bernoulli trial $X \sim B(p)$. Particularly, one single experiment is performed, and the fault has a p probability of becoming a failure. In repeated trials (i.e., binomial distributions), the probability of success p is the same every time the experiment is performed. Noteworthy, a Bernoulli trial grounds on these assumptions [16]:

1) There are only two possible outcomes for each trial (success and failure).
2) The outcomes from different trials are independent.
3) There are a fixed number n of Bernoulli trials conducted.
4) The probability of success is the same for each trial.

If these assumptions cannot be met, Bernoulli trials should not be used, and, as a consequence, (1) neither. Checking the adequacy of the Bernoulli trials assumptions is necessary to determine the validity of the statistical inference. In the following, an example from [16] is given with the intent of clarifying when the Bernoulli assumptions are valid and when they are not. *At a checkpoint, drivers will be screened to see if they are wearing or not a seatbelt. If all vehicles are treated the same, every driver has the same likelihood of not wearing a seatbelt. If drivers are categorized by age, you may require different probability for those under the age of 20 than those between the ages of 50 and 60. There would be no Bernoulli trials then.* In the CNN field, assuming that all faults have the same probability of success in each trial (4th Bernoulli assumption) is a very strong assumption. The probability that a fault becomes a failure (p) in CNNs is *not* the same for each injected fault. Based on the literature, it is known that it depends on several factors, such as the layer, the faulty location, the bit position, etc. Indeed, it is well-known that units inside a CNN have different vulnerabilities. With reference to the example (the seatbelt check), it means that we can not use Bernoulli trials to identify the most critical layer, the most critical bit inside weights, and so on. They would have different p probabilities. If we can not use Bernoulli trials, we can not use (1) to carry out the above-mentioned vulnerability analyses (critical layers, critical units inside the CNN, etc). For the sake of clarity, **it does not mean that** (1) **cannot be used with CNNs: if we treat the CNN as a black box, the only information that we can retrieve is overall information of the behavior of the CNN to faults, but not how vulnerable are the network's internal units (e.g., layers) to faults, since they have known different vulnerabilities (i.e., p probabilities) and the last Bernoulli assumption falls.**

B. Related Works

Recently, the problem of reducing the computational effort associated with FI campaigns has gained growing interest in many areas. The statistical background is based on a previous work [7], and allows defining a probabilistic model to find out the probability that r faults are detected in a random sample of R faults. The concept of the binomial distribution is presented, but, the main drawback is that a comparison with exhaustive results is missing. Statistical sampling was also used to investigate the effects of transient faults that propagate through processor cores (e.g., [11]). In 2008, the authors in [12] proposed a statistical analysis to perform focused statistically significant bit-flips into the system. In [10], the authors propose a machine learning (ML)-based vulnerability model to reduce the number of FIs. The results demonstrate the validity of the approach, but the effort for setting up the ML model is non-negligible. In 2009, a widely used method to select a statistically significant sample size was presented in [9]. The authors provide the same formula presented in (1) to calculate the sample size, the confidence and the error interval on the results. They state that, the number of FIs (n) that is necessary to inject in a system to obtain statistically significant results is defined by (1). Moreover, the authors in [9] assume that the characteristics of a population follow a normal distribution. But, it is necessary to specify that the normal approximation to the binomial distribution is applied.

III. Proposed Approach

In this section, a methodology to perform statistical FIs on CNNs that allows performing complete vulnerability investigations on the whole network and its internal units is presented in Section III-A. Then, Section III-B describes a data representation aware optimization to reduce the number of FIs.

A. Data-unaware Statistical Fault Injections on CNNs

To do comprehensive vulnerability investigations on CNN internal units, the sampling process must be modified. It is necessary to *change the granularity to identify subpopulations where the 4th Bernoulli assumption applies*, and where the probability p can be assumed equal among the trials (binomial distribution constraint). For example, to investigate up to the bit granularity in the CNN weights (i.e., to figure out the most critical layers and the most critical bits), subpopulations where (1) can be applied should be defined. To this end, it is reasonable to assume that a fault affecting the least significant bit (LSB) of a weight within a layer has the same probability to succeed (i.e., to cause a critical failure) as a fault affecting any other weight in the same bit position, within the same layer.

Therefore, if considering a CNN of L layers where each weight is represented using I bits, the subpopulation that we find by reducing the granularity to the bit level is the set of all faults in a specific bit position ($i \in I$) within that layer ($l \in L$). Consequently, the subpopulation will be $N_{\text{(i,l)}}$, and the sample $n_{\text{(i,l)}}$ (Fig. 1, right). The size of $N_{\text{(i,l)}}$ depends on the adopted fault model: if permanent faults are studied, $N_{\text{(i,l)}}$ would be equal to the number of weights in that layer multiplied by 2 (suck-at-0 and stuck-at-1). In the end, the final sample size in a CNN (the total number of FIs n_{TOT}) is computed as follows.

$$n_{\text{(i,l)}} = \frac{N_{\text{(i,l)}}}{1 + e^2 \cdot \frac{N_{\text{(i,l)}} - 1}{t^2 \cdot p * (1-p)}} \longrightarrow n_{\text{TOT}} = \sum_{l=0}^{L-1} \sum_{i=0}^{I-1} n_{\text{(i,l)}} \qquad (3)$$

The reader should note that the configurations of the parameters e, t, and p are kept equal among all the different subpopulations. Particularly, the probability of success p and failure $(1-p)$ are both equal to 0.5. Performing FIs at this granularity allows not only extracting the number of successes (faults leading to critical failures), but also comprehensively inspecting the most critical units inside the CNN model.

B. Data-aware Statistical Fault Injections on CNNs

Assigning to a fault the same probability of success and failure (i.e., $p = 0.5$) is the safest choice because it leads to the highest amount of FIs. As illustrated in Fig. 1 (left), when p equals 0.5 (x-axis), the multiplication between p and $(1-p)$ assumes the highest value (y-axis). The higher $p * (1-p)$, the higher the sample size n (1). This means that when the probability of success is different from 0.5 ($p! = 0.5$), the sample size n reduces, and in our context the number of FIs.

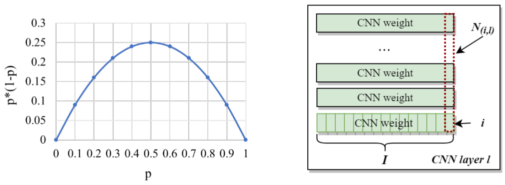

Fig. 1: Figure on the left: Probability of success (p). Figure on the right: illustration of the proposed approach.

In some data type representations, the probability p that a fault in a specific bit position of a weight becomes a critical failure is well-known. As an example, the probability that a fault affecting, in 32-bit floating-point (FP) representation, the LSB of the mantissa could result in a critical failure is almost non-existent. In this case, to reduce the number of FIs (the sample size n), we can assign to p a lower value: $p < 0.5$. Conversely, it has been proven that the probability that a fault affecting the most significant bit of the exponent part of the 32-bit FP representation could lead to a critical failure is extremely high. In this case, the probability of success is very high ($p > 0.5$), and the sample size greatly reduces. Please notice that different probabilities to faults on different bit positions can be given because subpopulations $N_{\text{(i,l)}}$ are independent.

This research work proposes a methodology to determine the p parameter starting from the CNN weights distribution with the aim of reducing the sample size n. In this work, the Single Precision IEEE 754 FP Standard is addressed. The idea is that the larger the variation a bit-flip introduces into a weight, the more likely the fault will cause an incorrect prediction or a critical fault. This variation is measured as the average distance produced by a bit-flip in a given bit position i for all the CNN weights. An example is reported in Fig. 2, where the distance value produced by a bit-flip on the 28th bit is illustrated. Assuming that the CNN weights are represented using I bits, for every bit $i \in I$, a criticality value (D_{avg}) is computed (4). $D_{\text{0-1}}(i)$ represents the average distance between all the golden and the faulty weights produced by a bit-flip from 0 to 1 on the bit i^{th}. Similarly, $D_{\text{1-0}}(i)$, is computed when the bit i^{th} is corrupted by a bit-flip from 1 to 0. The average distance does not determine the criticality by itself. This parameter is used in conjunction with the frequency with which each bit is either a logical 0 ($f_0(i)$) or a logical 1 ($f_1(i)$). In other words, for every weight within the distribution, we first compute the frequency for each bit to be a logical 0 or a logical 1 ($f_0(i)$ or $f_1(i)$), i.e., the number of time that the bit is set at 0 or 1.

The effect of a bit-flip on a specific bit i within a specific distribution of weights is represented by $D_{\text{avg}}(i)$:

$$\forall_{\text{i}} \in I \qquad D_{\text{avg}}(i) = D_{\text{0-1}}(i) * f_0(i) + D_{\text{1-0}}(i) * f_1(i) \qquad (4)$$

The p parameter used for determining the sample size represents the probability for a fault to become a critical failure: the closer p is to 0.5, the higher the number of FIs (Fig. 1). In line with this, we assume that the higher the average distance caused by a bit-flip in a weight, the higher the likelihood the fault will lead to a misprediction. For this reason, the final p is computed by performing a min-max normalization of D_{avg} between $a = 0$

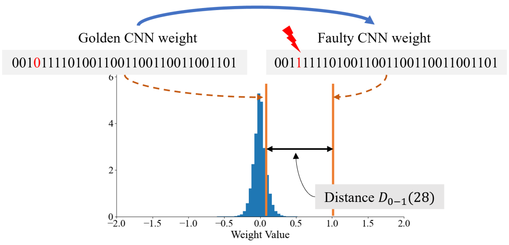

Fig. 2: Bit-Flip Distance.

and $b = 0.5$, without considering outliers (5). Since we assume that a high distance corresponds to a high criticality p, we can suppose outliers to have the highest criticality ($p = 0.5$).

Therefore, considering the maximum average distance $D_{\text{avg-max}}$ and the minimum average distance $D_{\text{avg-min}}$ measured for all the bits (I), the p can be computed as follows:

$$\forall_{\text{i}} \in I \qquad p(i) = a + \frac{(D_{\text{avg}}(i) - D_{\text{avg-min}})(b-a)}{(D_{\text{avg-max}} - D_{\text{avg-min}})} \qquad (5)$$

Once $p(i)$ is computed, $n(i, l)$ is obtained as shown in (3).

IV. Case Study

To experimentally demonstrate the effectiveness of the proposals, software FIs are performed on two CNNs: ResNet-20 and MobileNetV2, pretrained and tested on CIFAR-10 by using PyTorch. ResNet-20 reaches an accuracy equal to 91.7% and MobileNetV2 to 92.01% on the test set. The architectural details of the two CNNs are included in Table I and Table II, respectively. For reasons of space, only the total figures for MobileNetV2 are given. The considered fault models are permanent faults affecting the CNN weights. The FI process has been executed by exploiting the open-source PyTorchFI tool. Faults have been classified as *Critical* or *Non-critical*, depending on whether the top-1 prediction is correct. In Tables I and II, the number of faults (i.e., n) that are injected in each layer is given: along with the exhaustive FI details (where $n == N$), the following SFI campaigns are reported:

1) **Network-wise SFI**: The sample size n is determined by applying (1) to the entire CNN.
2) **Layer-wise SFI**: The sample size n is determined by applying (1) to each layer of the CNN.
3) **Data-unaware SFI**: The sample size n is determined by applying (1) at every bit position within each layer of the CNN (i.e., (3)). The p probability is always set to 0.5.
4) **Data-aware SFI**: The sample size n is determined by using (1) at every bit position within each layer of the CNN (i.e., (3)). The p probability is not equal for each population, and is computed as described in Section III-B.

V. Experimental Results

To validate the proposed SFI approaches with real comparisons, exhaustive FIs have been carried out: under the single fault assumption, a total of 17,174,144 stuck-at faults have been injected on all the weights of ResNet-20, and 141,029,376 on MobileNetV2. For every injected fault, the inferences of the entire test set (10k images) were run. Experiments have been run on an Intel(R) Xeon(R) Gold 6238R CPU @2.20GHz equipped with a GPU NVIDIA GeForce RTX 3060 Ti with 8 GB of Memory. The exhaustive FIs on ResNet-20 lasted about 37 days, while the exhaustive FIs on MobileNetV2 about 54.

TABLE I: ResNet-20: Exhaustive vs Statistical FIs.

Layer	Parameters (32-bit FP)	Exhaustive FI	Statistical FI			
			Network-wise [9] (e=1%, t=99%)	Layer-wise (e=1%, t=99%)	Proposed Data-unaware (p==0.5)	Proposed Data-aware (p!=0.5)
0	432	27,648	27	10,389	26,272	2,732
1	2,304	147,456	143	14,954	115,488	6,258
2	2,304	147,456	143	14,954	115,488	6,258
3	2,304	147,456	143	14,954	115,488	6,258
4	2,304	147,456	143	14,954	115,488	6,258
5	2,304	147,456	143	14,954	115,488	6,258
6	2,304	147,456	143	14,954	115,488	6,258
7	4,608	294,912	285	15,752	189,792	8,744
8	9,216	589,824	571	16,184	279,872	11,652
9	9,216	589,824	571	16,184	279,872	11,652
10	9,216	589,824	571	16,184	279,872	11,652
11	9,226	590,464	572	16,185	280,000	11,656
12	9,216	589,824	571	16,184	279,872	11,652
13	18,432	1,179,648	1,142	16,410	366,912	14,425
14	36,864	2,359,296	2,284	16,524	434,464	16,563
15	36,864	2,359,296	2,284	16,524	434,464	16,563
16	36,864	2,359,296	2,284	16,524	434,464	16,563
17	36,864	2,359,296	2,284	16,524	434,464	16,563
18	36,864	2,359,296	2,284	16,524	434,464	16,563
19	640	40,960	40	11,834	38,048	3,309
Total	268,346	**17,174,144**	16,625	307,650	**4,885,760**	**207,837**

TABLE II: MobileNetV2: Exhaustive vs Statistical FIs.

Total Layers	Total Parameters (32-bit FP)	Exhaustive FI (total)	Statistical FI (total numbers)			
			Network-wise [9] (e=1%, t=99%)	Layer-wise (e=1%, t=99%)	Proposed Data-unaware (p==0.5)	Proposed Data-aware (p!=0.5)
54	2,203,584	**141,029,376**	16,639	838,988	**14,894,400**	**778,951**

A. Defining the p parameter for a Data-aware SFI

One of the main contributions of this research work is the proposal of a methodology that, starting from the golden distribution of the CNN weights (i.e. not affected by faults), allows the criticality of the specific bit position (p) to be represented and the sample size of the SFI to be reduced. The procedure is described for ResNet-20, but the same is applied to MobileNetV2. ResNet-20 exploits a 32-bit FP representation for the weights. For every bit position, the number of times the bit is 0 ($f_0(i)$) or 1 ($f_1(i)$) has been calculated, for all the weights. (Fig. 3). Next, we computed for each bit position i, the average distance $D_{\text{avg}}(i)$ between the golden and the faulty weights that a bit-flip causes in that specific bit position in both directions ($D_{\text{0-1}}(i)$ and $D_{\text{1-0}}(i)$). Therefore, given the average

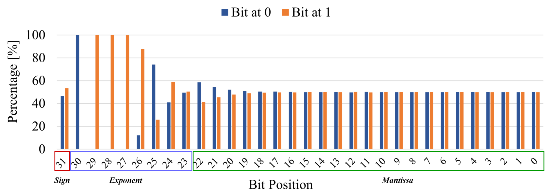

Fig. 3: $f_1(i)$ and $f_0(i)$) for ResNet-20 weights distribution.

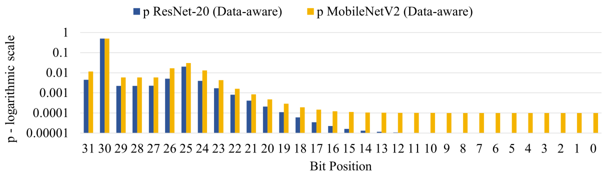

Fig. 4: Data-aware SFI: p for ResNet-20 and MobileNetV2.

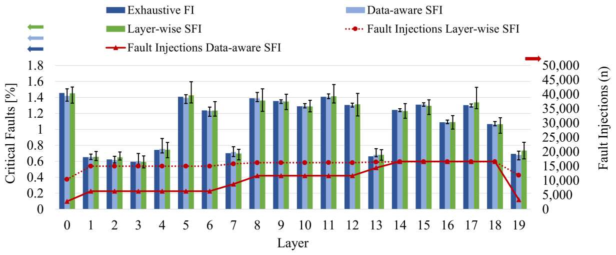

Fig. 5: Layer-wise and Data-aware SFIs.

distance $D_{avg}(i)$ the $p(i)$ of each bit $i \in I$ can be computed according to (5). Fig. 4 shows the probabilities p computed with this procedure, for both CNNs.

B. Complete Statistical Fault Injection Results

In this subsection, the four SFI approaches are compared, and the trade-offs are discussed for both CNNs. As mentioned, the validity of the approaches is measured by comparing the statistical results with the exhaustive ones. Before describing the overall findings, a first detailed analysis is provided for the first convolutional layer of ResNet-20. As illustrated in Fig. 6, ten random samples ($S0 - S9$) are extracted for each SFI approach (x-axis); the sample size (n) is specified in Table I (first row). The right y-axis defines the number of FIs (red line). The left y-axis represents the percentage of critical faults in that layer: the dark blue bar is the exhaustive result (obtained injecting all the N possible faults), while the light blue bar is the statistical result that we obtain by injecting a reduced number of faults (n). For every sample ($S0 - S9$), a thin black vertical bar represents the error margin of the statistical inference. In more details, the error margin delimits the range within which the exhaustive result would fall into (considering that, in real scenarios, the exhaustive result is not available). *If the exhaustive results fall into the error margin indicated in the statistical results, the statistical approach is valid and correctly predicts the final percentage of critical faults.* Clearly, the higher the sample size, the smaller the error margin. It is evident from Fig. 6 that the error margin is not acceptable for the network-wise SFI; it reduces in the layer-wise and data-unaware SFI as the sample size increases; and slightly increases in the data-aware scenario, albeit it keeps lower than the 1% (the initial requirement). It means that, by trading-off the costs of the FI campaign (i.e., number of FIs) and statistical accuracy, the proposed data-aware SFI approach might be considered the most effective. These data suggest that layer-wise and data-aware SFIs are the most effective in terms of number of FIs as well as accuracy (the error margin is below the predefined 1%). We extended this investigation to the entire CNN: in Fig. 5, the complete analysis on all the layers of ResNet-20 is given. As shown, in layers where almost the same number of faults is injected (i.e., L15, L16, L17, L18), the proposed data-aware method is, on average, more accurate (i.e., the error margin is smaller). Very interestingly, in layers where the proposed data-aware SFI injects a reduced number of faults (e.g., L9, L10), the accuracy of the estimate highly increases, on average. Fig. 5 shows that a layer-wise SFI provides good results in profiling the per-layer criticality. However, the problem with applying a layer-wise SFI is that it is not possible to investigate units inside the layer entity: for instance, according to the motivation of this research work, it is not possible to determine the most vulnerable bits inside a CNN. The same analyses performed on MobileNetV2 confirm the same trend. Additionally, results in Fig. 7 show that, compared to a network-wise SFI, the proposed data-aware SFI can correctly estimate the critical rate of layers. The data-aware SFI injects only the 0.55% of total faults, and provides results that are closest to the exhaustive (with an average error margin equal to 0.008%, Table III).

TABLE III: Comparing the FI methodologies.

ResNet-20	**FIs (n)**	**Injected Faults [%]**	**Avg Error Margin [%] (acceptable<1%)**
Exhaustive FI	17,174,144	100	-
Network-wise SFI [9]	16,625	0.09	**1.57**
Layer-wise SFI	307,650	1.79	0.19
Data-unaware SFI	4,885,760	28.45	0.06
Data-aware SFI	207,837	1.21	0.08
MobileNetV2	**FIs (n)**	**Injected Faults [%]**	**Avg Error Margin [%] (acceptable<1%)**
Exhaustive FI	141,029,376	100	-
Network-wise SFI [9]	16,639	0.01	**3.28**
Layer-wise SFI	838,988	0.59	0.01
Data-unaware SFI	14,894,400	10.56	0.001
Data-aware SFI	778,951	0.55	0.008

In Table III the four SFI approaches are compared in terms of number of injected faults (n), and the average error margin (the error margin averaged over all layers). It is necessary to underline that all the statistical approaches have been performed by pre-defining the error margin at 1% (as shown in Tables I and II). Data in Table III demonstrate the motivation behind this work: a network-wise SFI produces on ResNet-20 an error margin equal to 1.56% ($> 1\%$), and on MobileNetV2 an error margin equal to 3.28% ($> 1\%$). This means that the approach can not be considered statistically valid, and that reducing the granularity allows respecting the constraints and obtaining correct results. Moreover, it is clear that the data-unaware SFI approach leads to the lowest error margin, but the number of injected faults is higher compared to the layer-wise and the data-aware SFI techniques. It can be argued that the best compromise might be the data-aware technique, which, when compared to the layer-wise one, leads to a reduced margin of error and a smaller sample size.

VI. Conclusions

As the complexity of CNN models increases, the problem of reducing the costs of reliability assessment procedures

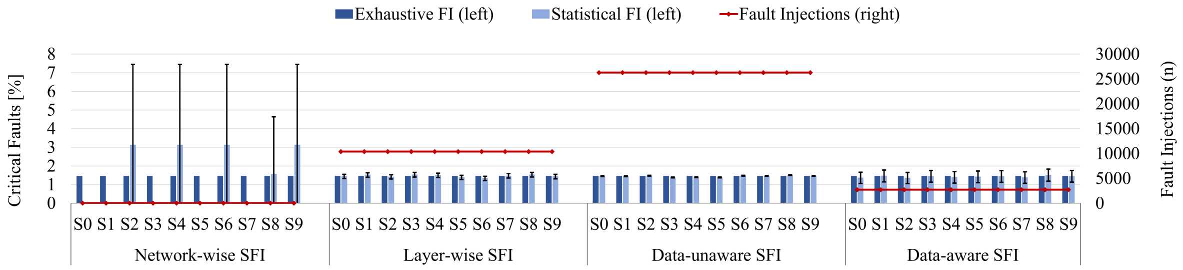

Fig. 6: ResNet-20: first layer.

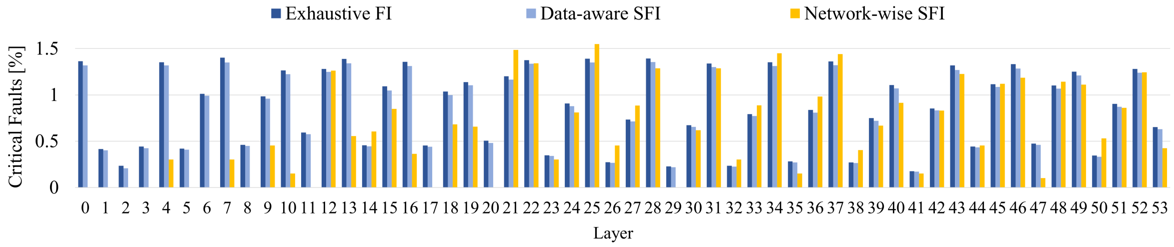

Fig. 7: MobileNetV2: a data-aware SFI yields statistically significant results, and correctly depicts the per-layer criticality.

assumes great significance. This work describes how to perform statistical inferences on CNNs to obtain statistically significant results. Moreover, it proposes an optimization which allows to further reduce the costs of the reliability assessment (in terms of fault simulations), while achieving accurate results. This data-aware SFI applies different criticalities to different subpopulations, computed by only observing the golden data distribution of a CNN. Overall, this article shows that, by reducing the sample size to only the 1.21% (ResNet-20) or 0.55% (MobileNetV2) of the entire possible experiments, it is possible to achieve an estimate of the CNN reliability close to the exhaustive result with an error always lower than 1%. It is important to underline that the fault model selection does not impact the statistical methodology: the only difference is the size of the total population of faults (N). As an example, if transient faults are selected, N must also consider all possible instants of time when a fault may be injected into the system.

To conclude, this research work underlines the importance of properly applying statistical approaches: not only the hypotheses must be met, but also understanding which type of information is possible to retrieve, is extremely important. In the future, the method will be applied to other fault models, and the data-aware SFI methodology will be extended to CNNs that use different architectures, datasets, and data representations.

References

[1] C. Alippi, V. Piuri, and M. Sami, "Sensitivity to errors in artificial neural networks: a behavioral approach," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*.

[2] M. A. Neggaz, I. Alouani, S. Niar, and F. Kurdahi, "Are cnns reliable enough for critical applications? an exploratory study," *IEEE Design Test*, vol. 37, no. 2, pp. 76–83, 2020.

[3] A. Ruospo, E. Sanchez, M. Traiola, I. O'Connor, and A. Bosio, "Investigating data representation for efficient and reliable convolutional neural networks," *Microprocessors and Microsystems*, vol. 86, p. 104318, 2021.

[4] G. Li *et al.*, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Denver, Colorado: ACM, 2017, pp. 1–12.

[5] M. Peña-Fernandez, A. Lindoso, L. Entrena, and M. Garcia-Valderas, "The use of microprocessor trace infrastructures for radiation-induced fault diagnosis," *IEEE Transactions on Nuclear Science*, vol. 67, no. 1, pp. 126–134, 2020.

[6] S. E. Damkjar, I. R. Mann, and D. G. Elliott, "Proton beam testing of seu sensitivity of m430fr5989srgcrep, efm32gg11b820f2048, at32uc3c0512c, and m2s010 microcontrollers in low-earth orbit," in *2020 IEEE Radiation Effects Data Workshop (in conjunction with 2020 NSREC)*, 2020, pp. 1–5.

[7] W. Daehn, "Fault simulation using small fault samples," *Journal of Electronic Testing*, vol. 2, 1991.

[8] H. Nguyen, Y. Yagil, N. Seifert, and M. Reitsma, "Chip-level soft error estimation method," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 365–381, 2005.

[9] R. Leveugle, A. Calvez, P. Maistri, and P. Vanhauwaert, "Statistical fault injection: Quantified error and confidence," in *2009 Design, Automation Test in Europe Conference Exhibition*, 2009, pp. 502–506.

[10] Y. Zhang, H. Itsuji, T. Uezono, T. Toba, and M. Hashimoto, "Estimating vulnerability of all model parameters in dnn with a small number of fault injections," in *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2022, pp. 60–63.

[11] E. Cheng *et al.*, "Clear: Cross-layer exploration for architecting resilience: Combining hardware and software techniques to tolerate soft errors in processor cores," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2016, pp. 1–6.

[12] P. Ramachandran, P. Kudva, J. Kellington, J. Schumann, and P. Sanda, "Statistical fault injection," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, 2008.

[13] N. Wang, J. Quek, T. Rafacz, and S. Patel, "Characterizing the effects of transient faults on a high-performance processor pipeline," in *International Conference on Dependable Systems and Networks, 2004*, 2004.

[14] Y. He, P. Balaprakash, and Y. Li, "Fidelity: Efficient resilience analysis framework for deep learning accelerators," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. Athens, Greece: IEEE, 2020, pp. 270–281.

[15] A. Bosio, P. Bernardi, A. Ruospo, and E. Sanchez, "A reliability analysis of a deep neural network," in *2019 IEEE Latin American Test Symposium (LATS)*, Mar., 2019, pp. 1–6.

[16] R. Johnson, I. Miller, and J. Freund, *Miller & Freund's Probability and Statistics for Engineers*, ser. Pearson Modern Classics for Advanced Statistics Series. Pearson Education, 2018.