An Effective Iterative Statistical Fault Injection Methodology for Deep Neural Networks

An Effective Iterative Statistical Fault Injection Methodology for Deep Neural Networks

Annachiara Ruospo, *Member, IEEE,* Matteo Sonza Reorda, *Fellow, IEEE,* Riccardo Mariani, *Senior Member, IEEE,* and Ernesto Sanchez, *Senior Member, IEEE*

Abstract—The complexity of the state-of-the-art devices makes reliability assessments approaches extremely complex and, sometimes, out of the timing constraints and computational capabilities. Fault Injections (FIs) are one of the most used approaches for evaluating the dependability of safety-critical systems. With billion-transistor hardware devices running trillion-parameter deep neural networks, injecting the entire fault universe is unfeasible. A widespread solution consists in performing statistical fault injections (SFIs), injecting a subset of faults to estimate a characteristic with an error margin and a confidence level. This research work presents an iterative SFI approach to estimate failure rates in convolutional neural networks (CNNs), i.e., the percentage of wrong predictions caused by random hardware faults affecting synaptic weights. SFIs at different granularities have been performed with margin of errors equal to 1%, 0.1%, and 0.01%. Results for two CNNs (ResNet20 and MobileNetV2) are presented and experimentally and statistically demonstrate the effectiveness of the proposed approach. For instance, to estimate the network-wise failure rate with an error margin of 0.01%, the proposed approach reduces the total injected faults by about 66% and 90% compared to conservative methods, and by 1.94% and 1.65% compared to iterative SFI methods in the literature, for ResNet20 and MobileNetV2, respectively.

Index Terms—Statistical Fault Injection, Convolutional Neural Network, Reliability, Fault Injection, AI Safety

1 Introduction

The constant demand of efficiency and performance in today's fast evolving computer systems has led to the creation of sophisticated and complex systems. The technological advances the society is witnessing are powerful and captivating. It is evident that artificial intelligence (AI)-powered systems have reached, in specific tasks, human-level capabilities. For their outstanding qualities, AI systems are today attractive also for safety-critical systems, where the dependability aspect is of paramount importance [1], [2]. The cost for the excellent performance is paid in terms of complexity: complexity of the software-hardware stack; complexity of the design; complexity of the dependability assessment phase; complexity of the market.

To safely deploy AI systems in safety-critical domains (e.g., self-driving cars, aerospace, robotics, healthcare monitoring systems, transportations), it is necessary to comply with rigorous safety standards. A standard on AI functional safety, i.e., the ISO/IEC TR 5469:2024, has been recently published [3]. The European Union is also making legislation regarding the safety and security of AI-based systems that have decision-making power, proposing a risk-based approach to strictly follow to produce/sell AI-powered systems in Europe (i.e., the AI Act [4]).

Among the different techniques, Fault Injection (FI)-based approaches are the most widely used for evaluating the dependability of AI systems [5], [6]. The system under assessment is subjected to a controlled set of faults and its response is observed. Not all faults lead to a system failure. Hence, the purpose of FI campaigns is to directly observe and measure the proportion of activated faults that result in a system failure. Although it constitutes a very effective technique to characterize the resilience of the system, its cost grows directly as the complexity of today's designs grows, often leading to unacceptably long FI experiments. As an example, modern Deep Neural Networks (DNNs) are made of billions or trillions of parameters (synaptic weights). The Google's Switch Transformer accounts for 1.6 trillion parameters [7]: carrying out an exhaustive FI is impractical and would require years to complete. Although the complexity of DNNs used in safety-critical systems is typically smaller, the reliability assessment process is still extremely time-consuming. The same reasoning can be applied to digital fault simulations: today's hardware technology is shrinking with the design of integrated circuits to a few nanometers, incorporating far more transistors in the same area. Simulating the entire fault list is becoming very complex and expensive, typically out of market time.

To address this problem, Statistical Fault Injection (SFI) approaches have been proposed with the intent of reducing the cost of the fault simulation procedure while still achieving statistically valid results [8], [9], [10], [11], [12], [13], [14], [15]. In general, SFI experiments inject a reduced number of faults to achieve an estimate of the target characteristic with a maximum margin of error and a specific confidence level. For dependability purposes, fixing a low margin of error is crucial. A 5% margin of error means that the statistical estimate will be close to the exact number by $\pm 5\%$. This value is good if the exact value has a medium-high failure rate: however, failure rate estimations in safety-critical systems are very low [16]. Hence, a $\pm 5\%$ of an estimate equal to 0.16% means that the real (unknown value) is in the range $[0, 5.16\%]$. A smaller margin of error must be

- *A. Ruospo, M. Sonza Reorda, E. Sanchez are with the Politecnico di Torino, Torino, Italy. E-mail<name.surname>@ polito.it.*
- *R. Mariani is with NVIDIA, US*

Manuscript received XX; revised YY.

used in a safety-critical scenario. However, the smaller the margin of error, the higher the number of faults needed to simulate. State-of-the-art (SoA) solutions, for a margin of error equal to 0.1% can require injecting, for deep AI models, more than 90% of the total fault list (which is still impractical). As an example, consider a fault list composed of 50k permanent faults. The intent of an SFI is to perform a reduced number of FIs ($<<$50k) to estimate the failure rate percentage of the unit under investigation. Let us suppose that a failure rate estimation with an error margin of 0.1% and 99% of confidence is needed. SoA solutions ([10] and Data-unaware [15]) would require injecting 48,542 faults, corresponding to 97% of the entire fault list. However, this number is too conservative if the exact failure rate is low (as it reasonably happens in dependability assessments). A possible SoA solution is to iterate by adding some faults at a time: so as soon as the desired margin of error is reached, the fault injection process is stopped.

This research work proposes an iterative SFI technique to provide a statistical estimate of the failure rate in Convolutional Neural Networks (CNNs), achieving the desired margin of error with a minimal number of iterations and a reduced number of fault injections. The proposed iterative SFI method introduces a formula for dynamically adjusting the margin of error based on the observed failure rate, iteration by iteration. Failure rate in CNNs is defined, in this work, as the percentage of injected faults affecting the static parameters (synaptic weights) that lead to wrong predictions.

The statistical investigations have been conducted targeting three margin of errors, particularly suitable for dependability assessments: 1%, and 0.1%, and 0.01%. We varied the margin of error while maintaining a fixed confidence level of 99%. However, in some experiments, we held the margin of error constant and adjusted the confidence level, ranging from 95% to 99.9%. The proposed statistical approach is validated by comparing the results with exhaustive FI campaigns. Two CNN models are used: ResNet-20 and MobileNetV2, trained and tested on CIFAR-10.

Moreover, statistical investigations have been conducted at different granularities, analysing a total of 140 different fault lists. Failure rates have been estimated at different levels: at the CNN level (i.e., the total number of faults affecting the entire CNN that lead to wrong predictions, *network-wise SFIs*); at the CNN layer level (i.e., the total number of faults affecting every layer of the CNN that lead to wrong predictions, *layer-wise SFIs*), and at the bit level (i.e., the total number of faults affecting every bit position of the CNN's weights that lead to wrong predictions, *bit-wise SFIs*).

Overall, the proposed manuscript provides a comprehensive analysis on statistical fault injections, comparing results with state-of-the-art works in terms of fault injections reduction and iteration reductions (for iterative approaches). To the best of the authors' knowledge, this is the first work to propose a formula for iteratively adjusting the margin of error based on the observed failure rate. Consequently, the estimate with the desired margin of error can be achieved with a significant reduction in both iterations and injected faults. The results experimentally demonstrate that the extent of the reduction is strongly influenced by the margin of error (with lower margins leading to a greater reduction in iterations) and the size of the fault list (with larger fault lists making the iterative SFI method more effective). For example, when estimating the failure rate of MobileNetV2 with a margin of error of 0.01%, the proposed SFI approach reduces the total injected faults by over 90% compared to one-step conservative methods (e.g., [10]), and by 1.65% compared to iterative SFI methods in the literature, while also reducing the total number of iterations by more than 57% relative to the iterative solution (e.g., [16]). For higher margins of error (e.g., 1%), the fault injection savings compared to state-of-the-art iterative methods can reach about 22%. Furthermore, the research highlights specific scenarios where an iterative approach is preferable for achieving a very low margin of error (e.g., e=0.01%), and others where a conservative approach reaches the same outcome more efficiently (about the same FI percentage with one single iteration). As the experimental results will demonstrate, the decisive factor is the size of the fault list: the larger the population of faults, the more effective iterative solutions become.

The rest of the paper is organized as follows. Section 2 provides the reader with some background knowledge about statistical inference and SFIs. Section 3 presents related studies, and Section 4 describes the proposed approach. The case study and the experimental results are given in Section 5. Finally, Section 6 draws conclusions and highlights future directions.

2 Background

This section provides background knowledge on statistical inferences (Section 2.1) and statistical FIs (Section 2.2).

2.1 Statistical Inference

Statistical Inference is the branch of statistics that deals with generalizing some characteristics of a population by observing only a reduced sample. A statistical **population** (N) refers to a set of all measurements corresponding to each unit in the entire population of units about which information is sought [17].

A population is typically described by the distribution of its values (i.e., a probability distribution for finite populations or a probability density function for infinite ones). Since the investigation of the entire population's characteristics is typically very difficult (today unfeasible), it is usually necessary to observe a **sample** (n).

If the sample is accurately defined, it is possible to properly estimate features of the whole population by observing only a reduced portion of it. It is important to underline that the sample must be: internally heterogeneous (i.e., representative of the entire population) and externally homogeneous (i.e., having the same probability of being selected).

However, in conducting a statistical investigation, whether exhaustive or sample-based, it is essential to associate with it the **statistical error**, understood as the discrepancy between the true value and the value available from the statistical inquiry (Figure 1).

When a sample (n) is used to estimate the mean (μ) and the variance (σ^2) of a population N, the probability that the

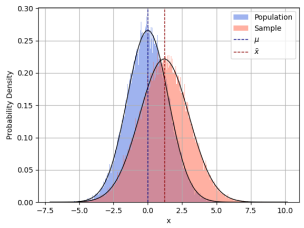

Fig. 1: Observing a statistical sample will always mean introducing an **error**.

mean and the variance of the estimation x will be equal to μ and σ^2 of the population are slim. For a single measurement, assuming that the true value is μ and the measured value is $\hat{x}$ (Figure 1), the statistical error of the estimate can be computed as $|\hat{x} - \mu|$.

Given that, for a large sample size (n $\geq$ 30), this error can be considered as a random variable having approximately the standard normal distribution, it is possible to compute the **maximum error of the estimate** (e) as:

$$e = z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \tag{1}$$

where:

- σ represents the standard deviation of the studied population N;
- n is the sample size;
- $z_{\frac{\alpha}{2}}$ is a constant that depends on the desired confidence level. It will be detailed in the following.

For a statistical inference, knowing (i) the maximum error of the estimate e given by (1), and (ii) the measured statistic $\hat{x}$, the unknown value of the population we would like to measure (μ) falls within the **confidence interval**:

$$\hat{x} - z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \leq \mu \leq \hat{x} + z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \tag{2}$$

with a $(1-\alpha)$ **confidence**:

$$P\Big(\hat{x} - z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}} \leq \mu \leq \hat{x} + z_{\frac{\alpha}{2}} * \frac{\sigma}{\sqrt{n}}\Big) = \Big(1-\alpha\Big) \tag{3}$$

TABLE 1: Most Used Confidence Intervals

Confidence Level (1-α)	95%	99%	99.9%
$z_{\frac{\alpha}{2}}$	1.96	2.58	3.09

A **confidence level** is a statistical measure of coverage. It says that, with a $(1-\alpha)$ confidence, the unknown parameter μ of the population will be covered by the confidence interval, Eq. (2). The most widely used values for $(1-\alpha)$ are 0.95 and 0.99, and the corresponding values of $z_{\frac{\alpha}{2}}$ are given in Table 1.

Therefore, to estimate the unknown value μ of a population, two main statistical inference approaches can be followed. In the first one (*n depending on e*), the inference is performed by fixing a maximum acceptable margin of error (e). Once e is fixed, the sample size (n) that allows to reach the real value μ with an error lower than e is known a-priori, and is computed as detailed in Eq. (4):

$$n = \frac{N}{1 + e^2 \cdot \frac{N-1}{t^2 \cdot p * (1-p)}} \tag{4}$$

In Eq. (4), N corresponds to the actual size of the population and p to the **probability of success**, i.e., the probability that the predefined "success" may occur. Being a probability, p assumes values between 0 and 1. When the probability of success is unknown, it is common to set p equal to 0.5, meaning that the single event under investigation has the same probability to success (p=50%) or to fail (p=50%). As it will be deepened in the following subsections, this is a conservative approach, because when p is set to 0.5, the number of statistical experiments is maximized.

In a second option (*e depending on n*), the sample size n is not defined a-priori. An arbitrary number of samples is chosen (typically using Eq. 4 with a big margin of error), and the final error is calculated as defined in Eq. (1). If it is considered acceptable, the statistical inference process can stop; otherwise, an iterative approach is set up.

2.2 Statistical Fault Injections

Leveraging statistical inferences (Section 2.1) for performing reliability analyses is a widely used approach in the literature. Among the different uses, FI processes find great advantages in relying on statistical inferences approaches. Indeed, this approach allows to *drastically reduce the number of possible FI experiments* to execute, while achieving statistically significant results. The sampling formula in (4) represents a very important finding, and, most importantly, it defines the number of statistical inferences (i.e., the number of random experiments or, in our case, the number of FIs) that must be performed ($n << N$) to run statistical inferences.

As mentioned, SFIs are widely used in the research community to perform reliability assessments also on DNNs (e.g., [18], [19]). In particular, the gathered results on the sample are elaborated to identify the artificial neural network's criticalities (for example, the most critical layer, the most critical bit in the CNN weights, and so on). As a matter of fact, performing exhaustive FIs on real CNNs is out of the computational possibilities: modern CNN models are made of millions or billions of parameters (e.g., weights) and operations (e.g., Multiply-Accumulate Operation MAC), the underlying hardware may easily include billions of transistors, and therefore sampling becomes necessary to get realistic simulation times.

In the FI context, N is used to indicate the population's size, i.e., the total number of possible faults in a system (e.g, the total number of stuck-at faults in a CPU, or the total number of soft errors in CNN weights). The term sample 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, the sample size (n) is much lower than 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 or x successes and $n-x$ unsuccesses in n attempts.

The reader should notice that, in the context of FIs, **a trial is a success when a fault produces a critical failure**.

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 producing a failure. Moreover, a Bernoulli trial grounds on these assumptions [17]:

1) There are only two possible outcomes for each trial (success and unsuccess).
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, Eq. (4) neither.

3 Related Works

This research work identifies two typologies of SFIs and divides state-of-the-art research works in two main categories, following the two paradigms mentioned in Section 2.1:

1) **One-step or Conservative *(n depending on e)***: The total sample size (n) is defined given a fixed margin of error (e) before running the fault injection campaigns. Once the desired margin of error and confidence level are defined, the sample size is computed as in (4). It is always a good practice to control that the obtained margin of error is within the defined margin of error. Using a probability of success (p) equal to 0.5 guarantees this result, only if the statistical hypotheses are respected.
2) **Iterative *(e depending on n)***: The sample size (n) is iteratively increased to achieve a specific margin of error (e_{goal}). The key aspect of this approach is the ability to monitor the margin of error and schedule new FI experiments to iteratively increase the sample size, while reducing the margin of error. Clearly, constantly monitoring the results is, at the same time, an advantage, and a drawback. The advantage is the ability to stop when the conditions (e.g., time or specific error margin) are met. The drawback is the monitoring activity itself that requires additional operations managed by a specific FI manager (not needed in the one-step SFI approaches).

The state-of-the-art SFI works have been classified as **one-step** or **iterative** approaches, as given in Table 2.

TABLE 2: Statistical Fault Injections techniques in the state of the art.

SFIs	One-step	Iterative
Research Works	[8], [9], [10], [11], [12], [13], [15], [20], [21], [22]	[16]

Among the one-step approaches, the following works are worth being mentioned. The authors in [20] propose a methodology based on fault sampling to reduce the dimension of the fault list. The statistical background is based on a previous work [8], 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 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., [12]). In 2005, a further statistical proposal was presented in [9], where the authors performed FIs on an Itanium-class processor to derive the logic derating for latches. The population size corresponded to 150 trillion flips that could happen during the application run. In 2008, the authors in [13] proposed a statistical analysis by comparing a method for SFI into arbitrary latches within a full system hardware-emulated model with particle-beam-accelerated SER testing for a modern microprocessor. This investigation was used to focus on statistically significant bit-flips into the system.

In 2009, a pivotal and very important method to select a statistically significant sample size was presented in [10]. The authors provide the fault sampling formula presented in Eq. (4) to calculate the sample size, given the confidence level and the error margin. The SFI approach is validated using a cryptographic coprocessor performing Advanced Encryption Standard (AES) computations. SFIs based on [10] have been widely exploited by the research community in the following years to perform reliability assessments also on DNNs (e.g., [18], [19], [23], [24]). In particular, the results collected on the sample n are elaborated to identify the DNN vulnerabilities.

However, the application of the SFI technique presented in [10] to DNN models has not always been statistically accurate in the literature. Results obtained at a specific granularity (e.g., fault sampling at the DNN level) have been used to lay out conclusions at lower granularities (e.g., at bit level), *ignoring the change in the margin of error.* This issue has been addressed in [15], where two methodologies to perform SFIs on CNNs have been presented. They allow performing complete vulnerability investigations on the whole neural network and its internal units, by defining not only *how many* faults need to be injected, but also *where* they should be placed to achieve statistically significant results. Additionally, the work in [15] describes a methodology to heuristically compute the probability of success (i.e., the *p*-value) of a population of faults without running FI campaigns. This analysis allows reducing the sample size by tuning the *p*-value within the fault sampling formula (4). Starting from the probability distribution of the golden data representation (the DNN synaptic weights), the Average Bit Flip Distance (ABFD) is computed to measure the probability of a fault to produce a critical failure (defined as a wrong classified image).

In summary, [15] provides two SFI methodologies to run vulnerability investigations of CNNs:

- **Data-unaware SFI**: The sample size n is determined by applying Eq. (4) at the specific granularity of the analysis. The p probability is always set to 0.5, meaning that every injected fault has the same probability of success or fail. This approach can be considered the most conservative one (leading to a high sample size) but guarantees the minimum margin of error at the given granularity.
- **Data-aware SFI**: The sample size n is determined

by using Eq. (4) at every bit position within each layer of the CNN. The p probability is not equal for each population of faults: it is computed by means of a preliminary analysis consisting in the ABFD computation. The *p*-value used in Eq. (4) approaches the real one. This leads to considerably reducing the sample size, at the cost of estimating the *p*-value before running the FI campaigns.

FI approaches that rely on preliminary analyses of the fault population (in line with the abovementioned data-aware SFI [15]) also fall into the same one-step category. For example, [22] and [21] propose techniques and frameworks to reduce the fault list size, and consequently the cost for FI. However, they require domain-specific knowledge of the application under assessment. More in detail, they propose *application-dependent* fault pruning approaches to reduce the fault list space, relying on application characteristics running on GPGPUs [21] and microprocessors [22].

Overall, this methodology shows that by leveraging statistical properties, one-step FI approaches can take into account the specific failure rate behaviour of the design/model under consideration, without configuring preliminary pruning strategies. Indeed, it is possible to benefit from initial statistical estimations available after few experiments to further minimize the number of injected faults. The dynamic adjustment of this number should occur in real-time, responding to the ongoing accuracy of estimations during experimentation. Consequently, the FI campaign concludes when the accuracy of estimations reaches or falls below the desired margin of error. To do so, in contrast to one-step SFI approaches, iterative approaches iteratively increase the sample size to achieve a specific margin of error.

The authors in [16] present a technique to run iterative SFIs to estimate the failure rates of the LEON3 processor, a 32-bit processor compliant with the SPARC V8 architecture. They show that it is possible to conduct a detailed statistical investigation of an implementation-based HDL model of the targeted LEON3 processor, by reducing the time devoted to experimentation from 23 days to a little more than 1 day (27 hours). They propose two variants of the approach: error-driven and time-driven. In the first case, their iterations stop when the margin of error is equal or lower than the desired one; in the second case, when the time devoted to experimentations expires. As it will be described, the proposed approach takes the best from [16] and [15].

4 Proposed Approach

Reducing the cost and time required to perform reliability assessments is becoming of fundamental relevance given the increasing size of modern AI models. This research work proposes a statistical fault injection methodology to iteratively reach an estimation of the failure rate within a targeted margin of error. To fulfil the goal with the lowest number of iterations and fault injections, the proposed SFI method iteratively benefits from the values obtained in every iteration, in order to better determine the subsequent number of fault injections, limiting the possibility of overestimating the number of injected faults through reduction of iterations. In other words, iterations are guided by the probability of success p. A success is defined as the occurrence of a specific event: in the FI context, it is the occurrence of a failure. Similarly, the probability of success p is the probability that a fault produces a failure ($0 < p < 1$). Refer to Table 3 for comprehensiveness of terminology.

TABLE 3: SFI Terminology

Statistical Term	Definitions in the fault injection field
N	Population under investigation. Entire fault list.
Success	Occurrence of a specific event.
Estimation	Fault Injection campaign.
Estimate	Statistical measure. Result of a FI campaign.
p	Probability of success of the estimate. In the FI context, it is the probability that a fault becomes a failure, also known as *failure rate*.
e	Margin of error of the estimate.
t	Confidence level. Also referred to as $z_{\frac{\alpha}{2}}$
n	Sample size. In the FI context, it defines the number of faults to inject and is computed as in (4).
$\hat{x}$	Number of successes obtained in a FI campaign.
$\hat{p}$	Probability of success measured. It is computed as $\hat{p} = \frac{\hat{x}}{n}$
$\hat{e}$	Margin of error of the estimate measured. It is computed as $\hat{e} = t * \sqrt{\frac{\hat{p}*(1-\hat{p})}{n}} * \sqrt{\frac{(N-n)}{(N-1)}}$ $\sqrt{\frac{(N-n)}{(N-1)}}$ is the Finite Population Correction Factor.

To estimate a characteristic of a population under investigation, a very conservative approach consists in assuming that an individual (i.e., a fault in the FI context) may or may not exhibit the characteristic being estimated with the same probability p=0.5. In other words, an event has a 50% probability of occurrence. Given a specific margin of error e and confidence level t, this conservative assumption leads to the highest sample size n, turning out to be the worst-case scenario. When the event being estimated is more ($p \to 1$) or less ($p \to 0$) probable, the number of necessary experiments (i.e., fault injections) required to achieve a desired margin of error considerably reduces [16]. *It should be noted that in the dependability community, failure estimations in safety-critical systems generally have very low failure rates.* This property can be explored to reduce the costs associated with FIs and, generally, with reliability assessment approaches. In [15], the feasibility of adjusting p to reduce the sample size was explored by means of a preliminary domain-dependent heuristic analysis. However, relying on preliminary information (to estimate the p to measure the $\hat{p}$) is not always easy, and it is strictly related to the deep knowledge of the user about the characteristic under investigation.

This research work proposes an iterative SFI process that starts from no knowledge of the population (p = 0.5) to iteratively approach the real estimation of the failure rate ($\hat{p}$) in *the shortest possible time.* A flowchart of the proposed approach is illustrated in Fig. 4. The SFI process starts by taking into account the worst-case scenario (equal probability of success and unsuccess), and from a big margin of error. The goal of the SFI is to obtain an accurate estimation of the failure rate of the population with a maximum error margin e_{goal}. When the measured error margin is lower than the desired one, iterations stop.

At the first iteration ($i = 0$), no indications are available for the population of faults (N). So, a first FI campaign of n faults is executed with p_0=0.5, e_0=e_{start}, and confidence level t (Fig. 2a). Adopting a p_0=0.5 is essential at the beginning of

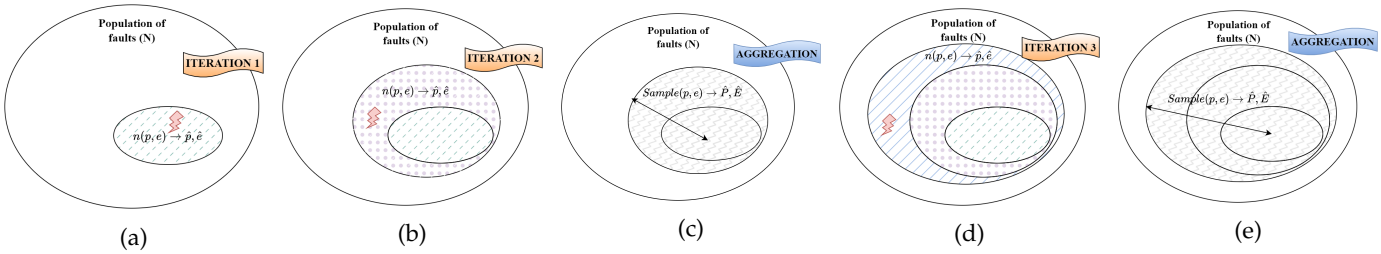

Fig. 2: Illustration of the proposed approach.

the iterations to guarantee that no preliminary knownledge is necessary on the criticality of the system being investigated. Indeed, each fault has the same probability (p) of producing a failure ($p_0 = 50\%$) or not. Moreover, when p_0=0.5 is used, the sample size (n) is maximised ([10], [16]), and this is useful at the first iteration to obtain an estimate close to the real value (this estimate will guide subsequent iterations).

So, the first iteration will provide an estimation $\hat{p}_0$ with an error $\hat{e}_0$ that will be lower than e_0 (because p_0=0.5 was used). This initial estimation is extremely useful because it provides an estimate (an initial *flavour*) of the actual failure rate of the population, that can be leveraged for the subsequent iteration (Fig. 2b) to avoid using p_{i}=0.5 ($i! = 0$) to compute n in scenarios where $p \rightarrow 0$ or $p \rightarrow 1$. The first time, this set of injected faults is added to the list of Samples. When we refer to the Sample, capital letters $\hat{P}$ and $\hat{E}$ are used. Then, the margin of error $\hat{E}_{\text{i}}$ is computed and if it is higher than the targeted e_{goal}, a new iteration starts ($i++$). Every time a new FI experiment is performed, estimates are aggregated (Fig. 2c) in order to compute the total failure rate measured on the total sample ($\hat{P} = \frac{\hat{P}_{\text{TOT}}}{n_{\text{TOT}}}$). Then, the margin of error on ($\hat{E}$) must be recomputed as in Table 3. The cost of this phase is significant, so it should be minimized by reducing the number of iterations as much as possible. The confidence level t is constant during the entire SFI process. The SFI process continues (Fig. 2d and 2e) as the measured margin of error reaches e_{goal}. When this occurs, with a given confidence (e.g., 95% or 99%) the estimate, including its margin of error, will cover the true value sought (e.g., the true failure rate of the population N).

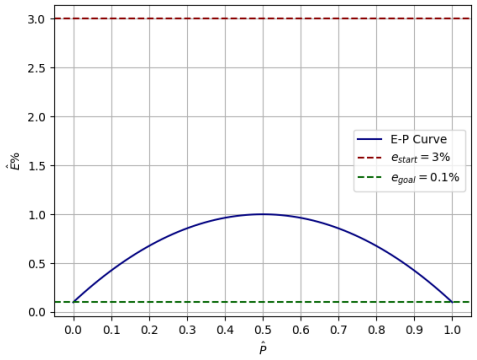

Fig. 3: Margin of error driven by $\hat{P}$.

The proposed iterative approach primarily focuses on two key aspects, which are visually represented in the yellow box of the flowchart (Fig 4):

1) **$\hat{\text{P}}$-guided iterations**: Every time a new SFI experiment starts, the $\hat{P}$ obtained from the previous iteration is used to compute the next sample size, as in (4). Equal probability of success $p = 0.5$ is used only for the first iteration, when no failure rate indications are available.

2) **Margin of error driven by $\hat{\text{P}}$**: Every time a new SFI experiment starts, the new margin of error e must be reduced iteratively to approach the target e_{goal}. To boost the number of FIs, for the very first time, we propose an E-P curve that iteratively reduces e according to $\hat{P}$, as in (5), and as shown in Fig. 3.

The aim is to quickly approach e to e_{goal} for a non-critical and critical population of faults ($p \rightarrow 0$ and $p \rightarrow 1$), taking into account that although we are lowering the error a lot, under these circumstances ($p \rightarrow 0$ and $p \rightarrow 1$) the number of injections to be made is lower, so the sample size is balanced. In other words, as the measured failure rate ($\hat{P}$) of a specific iteration tends to 0 or 1, the error margin of the next iteration tends to e_{goal}; as the measured failure rate ($\hat{P}$) of a specific iteration tends to 0.5, the error margin of the next iteration tends to $\frac{\hat{E}}{3}$. Therefore, the curve that best represented this requirement was a parabola passing through three (x, y) points: (0, e_{goal}), (1, e_{goal}), (0.5, $\frac{\hat{E}}{3}$). By considering the parabola formula ($y = a * x^2 + b * x + c$) passing through the aforementioned three points, Equation 5 was mathematically derived. At a given iteration i, the margin of error of the subsequent iteration $e_{(\text{i+1})}$ is computed as follows:

$$e_{(\text{i+1})} = \begin{cases} -k\hat{P}_{\text{i}}^2 + k\hat{P}_{\text{i}} + e_{\text{goal}}; & if\ \frac{\hat{E}_{\text{i}}}{3} > e_{\text{goal}} \\ e_{\text{goal}}. \end{cases} \quad (5)$$

where $k = 4 * (\frac{\hat{E}_{\text{i}}}{3} - e_{\text{goal}})$.

The proposed E-P curve offers benefits over the SoA iterative approach, which only halves the margin of error at each iteration (i.e., $e_{\text{i+1}} = \frac{\hat{E}_{\text{i}}}{2}$). Indeed, for populations having a low failure rate (p→0) or a high failure rate (p→1), halving the margin of error at each iteration may lead to extra FIs and iterations. On the contrary, for populations having a failure rate close to 50% (p→0.5), halving the margin of error at each iteration leads to extra iterations. A practical example and concrete motivations are given in the experimental section. The proposed E-P curve shows benefit in (i) reducing the number of FIs and additionally (ii) reducing the iterations when p→0.5. An example of the

E-P curve is illustrated in Fig. 3 for a generic SFI using an initial error margin of 3% and an $e_{\text{goal}} = 0.1\%$.

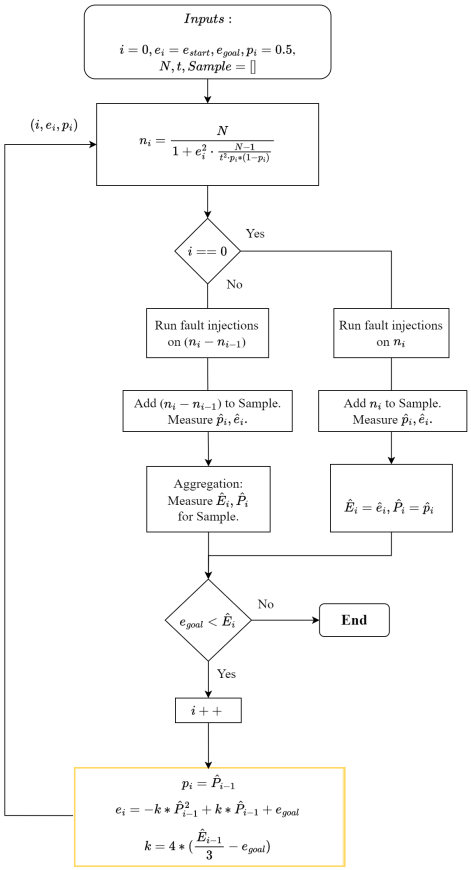

Fig. 4: Flowchart of the proposed approach.

5 Experimental Analysis and Results

This section shows the effectiveness of the proposed approach. Section 5.1 provides details about the adopted fault model and CNNs. Next, it describes how the different SFI investigations are configured. Finally, experimental results are given in 5.2, and concluding remarks are given in Section 5.3.

5.1 Configuration of SFI experiments

To show the effectiveness of the proposed SFI methodology, this work investigates the impact of permanent faults on CNN weights. Their investigation is very common in the research community: many authors have addressed the same problem and exploited the same fault model in the last decade [25], [26]. Moreover, the authors in [26] have highlighted that permanent faults in CNN accelerators have a major impact on CNN accuracy with respect to, for instance, temporary faults (soft errors). The modification of the weights of a CNN mimics the occurrence of stuck-at faults in the memories, which are very common due to aging effects, radiations, etc. [27], [28], [29], [30].

Comprehensive statistical investigations have been performed on two CNNs: ResNet20 and MobileNetV2, trained and tested on CIFAR-10. For every single injected fault, the entire test set of CIFAR-10 is executed (10k inferences for every fault). Therefore, injected stuck-at faults are independent, in line with the second Bernoulli's assumption (Section 2). ResNet-20 encompasses 268,346 32-bit floating-point (FP32) weights, that account for a total of 17,174,144 permanent faults. MobileNetV2 is deeper compared to ResNet-20, and includes a total of 2,203,584 FP32 weights. It means that to exhaustively test this second CNN, more than 141 million FIs must be performed. 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, exploiting a Pytorch-based FI tool [31]. The exhaustive FIs on ResNet-20 lasted about 37 days, while the exhaustive FIs on MobileNetV2 about 54 days.

Overall, the intent of the statistical investigation is to estimate the percentage of critical faults in image classification CNNs. Injected faults have been classified as *Critical* or *Non-critical*, depending on whether the top-1 prediction is correct. Faults affecting weights and propagating to the application's output are named as Silent Data Corruptions (SDCs). Critical faults changing the top-1 prediction are referred to as SDC-1 [23].

The following SDC-1 estimations have been done considering three different scenarios, maintaining the same fault model (permanent faults on synapic weights):

- **Network-wise SFI**: the percentage of SDC-1 affecting the CNN has been estimated considering the entire list of faults of the entire CNN. The SFIs considers one single population. This scenario is adopted when a statistical SDC-1 estimate for the entire CNN model is required (e.g., also known as model-wise assessment [32]).
- **Layer-wise SFI**: the percentage of SDC-1 affecting individual layers of the CNN has been estimated. To meet the statistical hypothesis and constraints, the total fault list (network-wise) has been divided at the layer granularity. Experiments have been run on every single list of faults referring to every single layer in the CNN (one population corresponds to a single layer). This scenario is adopted when a statistical SDC-1 estimate for individual layers of the CNN model is needed. In the literature, this configuration is employed to investigate the criticality of specific layers and architectural components (e.g., [33], [34]). For a given margin of error, it is necessary to sample faults within each layer-wise fault list.
- **Bit-wise SFI**: the percentage of SDC-1 corrupting every bit position of the CNN has been estimated. Since both CNNs adopt a FP32 data representation, a total of 32 fault lists have been analysed. This scenario is adopted when the reliability assessment aims at investigating, through SFIs, the most critical bits in a given data representation (e.g., [23], [35]).

The reader should be aware that the proposed methodology is a statistical method applicable to a variety of scenarios. The scenarios included in this manuscript were chosen because they are among the most commonly used in the literature.

Data in Table 4 report the number of populations in each analysed scenario, and the total number of faults considered

TABLE 4: Details of the SFI experiments performed and the size of the total fault lists. A total of 140 fault lists have been analysed. In layer-wise experiments, every layer comes with a different number of FP32 weights. Hence, the dimension of the fault lists varies accordingly. In this table, the minimum and the maximum size are reported.

SFI Setup	Network-wise	Layer-wise	Bit-wise
ResNet20			
Number of populations (i.e., fault lists)	1	20	32
Fault list size	17,174,144	27k − 2.3M	536,692
MobileNetV2			
Number of populations (i.e., fault lists)	1	54	32
Fault list size	141,029,376	18k − 26M	4,407,168

in every *network-wise, layer-wise,* or *bit-wise* SFI experiments. Additionally, for each scenario, three different margins of errors are considered (e_{goal} = 1%, e_{goal} = 0.1%, e_{goal} = 0.01%). Even though it is highly common to set the error margin equal to 5%, it is worth to underline that in safety-critical systems, a 5% error may be too high. As a consequence, it is important to adopt low margin of errors, especially when the quantity to estimate has a low probability of success. Clearly, the lower the margin of error (e), the higher the sample size (n). The selected margin of errors are in line with those adopted in state-of-the-art works, e.g, [10], [15], [16]. This research work, leveraging an iterative approach that requires an initial error margin (e_{start}) and a final error margin (e_{goal}), proposes three analysis, with three levels of precision:

- level-1: e_{start} = 5%, e_{goal} = 1%
- level-2: e_{start} = 3%, e_{goal} = 0.1%
- level-3: e_{start} = 3%, e_{goal} = 0.01%

All the SFI investigations have been performed using a confidence level of 99% (t=2.58, Table 1). Additionally, the last set of experiments (Section 5.2.3) has been performed by keeping the error margin fixed at 1% and varying the confidence level to 95%, 99%, and 99.9% (following data in Table 1).

5.2 Experimental Results

Experimental results of a network-wise (Section 5.2.1), layer-wise (Section 5.2.2), and bit-wise (Section 5.2.3) statistical fault injection are given in the sections below.

5.2.1 *Network-wise SFI*

The purpose of a network-wise SFI analysis is to estimate the total percentage of SDC-1 faults in the CNNs under investigations with the three levels of precisions (i.e., *level-1*, *level-2*, and *level-3*). Given the exhaustive FI results, the total number of SDC-1 faults in ResNet20 is equal to 1.16%, and 1.11% in MobileNetV2. The statistical experiments below aim at estimating this quantity with an e_{goal}=1%, e_{goal}=0.1% and an e_{goal}=0.01%. For the sake of clarity, in Table 5 a step-by-step description of the proposed approach applied to ResNet20 is given. In the first iteration, no information is available about the characteristic under investigation of the population: so, a 50% probability of success is used, and a big error margin equal to 5%, injecting a total of 670 permanent faults. The results of this first iteration give a $\hat{P} = 1.32\%$ and an error margin slightly higher than the desired one ($\hat{E} = 1.14\%$). It means that a new iteration is needed. Hence, the previous estimation is used to define the next number of faults to inject ($p = \hat{P}$), and e as (5). After injecting new 206 faults, the results are aggregated (Fig. 2c) and the final $\hat{P}$ and $\hat{E}$ are computed. As shown in Table 5, $\hat{E} < 1\%$, then the SFI process stops.

A comparison with SoA works using *level-1*, *level-2* and *level-3* precisions is provided in Tables 6, 7 and 8 for both CNNs: ResNet20 and MobileNetV2. Comparisons are performed in terms of total number of injected faults and total number of iterations performed to achieve a desired margin of error. To avoid dependence on a single sampling process, we conducted iterative SFI experiments ten times to observe a general trend. The average values are reported. Additionally, the gains in terms of FI reduction and total number of iterations are given in Table 9 and 10, respectively.

TABLE 5: An example for the *level-1* proposed SFI process (e_{goal}=1%, t=99%).

Proposed Approach	Before the FI			After the FI	
	p [%]	e [%]	n	$\hat{P}$ [%]	$\hat{E}$ [%]
Iteration 1	50	5	670	1.32	1.14
Iteration 2	1.32	1	876	1.28	**0.98**
Total			876		

Overall, results show that iterative SFI methods dramatically reduce the number of injected faults, at the cost of performing more than one iteration to achieve a specific margin of error. As shown in Table 9, compared to One-Step approaches, the proposed iterative method can save more than about 94% of injected faults to achieve an error margin of 1% or 0.1%, and more than about 66% to achieve an error margin of 0.01%. On the other side, compared to the SoA iterative approach, the proposed *EP-guided* iterations can reduce the total number of FIs (from about 2% to more than 13%, Table 9) and, at the same time, reduce the total number of iterations needed to achieve the goal. Table 6 does not include the comparison with the SoA iterative because, after just the first iteration, the margin of error in all 10 runs was very close to the targeted 1%. In both algorithms (proposed and the SoA), the next error corresponds to the e_{goal}, obtaining similar outcomes, an example is given in Table 5. For a reduced margin of error (e=0.1%), Figure 5 shows in detail the iteration-by-iteration SFI execution of one (random) run (i.e., *Run 1*) out of the ten performed (the

TABLE 6: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with an error margin lower than 1%.

SFI Methods	e_{goal} = 1%	
	One-Step [10] and [15]	Proposed Iterative (Average over 10 runs)
ResNet20		
Total FIs	16,625	980
Total Iterations	1	1.91
MobileNetV2		
Total FIs	16,639	908
Total Iterations	1	1.90

TABLE 7: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with an error margin lower than 0.1%.

SFI Methods	e_{goal} = 0.1%		
	One-Step	Iterative (Average over 10 runs)	
	[10] and Data-unaware [15]	SoA [16]	Proposed Approach
ResNet20			
Total FIs	1,517,100	80,734	69,959
Total Iterations	1	4.3	3.12
MobileNetV2			
Total FIs	1,644,693	80,582	70,714
Total Iterations	1	4.5	3.5

TABLE 8: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with an error margin lower than 0.01%.

SFI Methods	e_{goal} = 0.01%		
	One-Step	Iterative (Average over 10 runs)	
	[10] and Data-unaware [15]	SoA [16]	Proposed Approach
ResNet20			
Total FIs	15,567,517	5,359,619	5,255,416
Total Iterations	1	7	3
MobileNetV2			
Total FIs	76,336,021	7,020,572	6,931,371
Total Iterations	1	7	3

goal is $\hat{E} \leq 0.1\%$). The SoA iterative achieves a final margin of error of 0.094% with a total of four iterations and 84,731 injected faults. The measured probability of success ($\hat{P}$) and margin of error ($\hat{E}$) correspond to (iteration$_1$: $\hat{P}$=1.92%, $\hat{E}$=0.82%), (iteration$_2$: $\hat{P}$=1.22%, $\hat{E}$=0.33%), (iteration$_3$: $\hat{P}$=1.29%, $\hat{E}$=0.17%), (iteration$_4$: $\hat{P}$=1.14%, $\hat{E}$=0.094%), respectively. In contrast, the proposed iterative approach achieves a final margin of error of 0.099% with only two iterations and 72,506 injected faults, reducing the number of injected faults by 12,225 faults and requiring two fewer iterations. The measured $\hat{P}$ and $\hat{E}$ correspond to (iteration$_1$: $\hat{P}$=1.24%, $\hat{E}$=0.67%), (iteration$_2$: $\hat{P}$=1.10%, $\hat{E}$=0.1%). In both cases, the final estimates $\hat{P} \pm \hat{E}$ *cover* the exact SDC-1 rate of 1.16%.

One of the key advantages of the proposed solution, aside from the decrease in injected faults, is the reduction in the number of iterations required to reach the desired outcome. As we will discuss, this reduction in iterations may also be viewed as a contributing factor to the decrease in injected faults. Iteratively checking the status of the SFI experiments means monitoring the margin of error: as soon as it meets the desired e_{goal}, the entire SFI process stops. This monitoring activity at each iteration is costly. Indeed, for a single iteration it involves:

1) running the FIs;
2) stopping the FIs;
3) analysing the results (e.g, computing the SDC-1 metric);
4) aggregating the results and computing the margin of error over the sample ($\hat{E}$);
5) restarting the FI process if $\hat{E} > e_{goal}$: the FI process restarts from 1).

TABLE 9: Gains in terms of FI reduction %.

Proposed	Avg. FI Reduction [%], confidence 99%					
	e=1%		e=0.1%		e=0.01%	
	Res.	Mob.	Res.	Mob.	Res.	Mob.
SoA One-Step	94.10	94.54	95.38	95.70	66.24	90.91
SoA Iterative	-	-	13.34	12.24	1.94	1.65

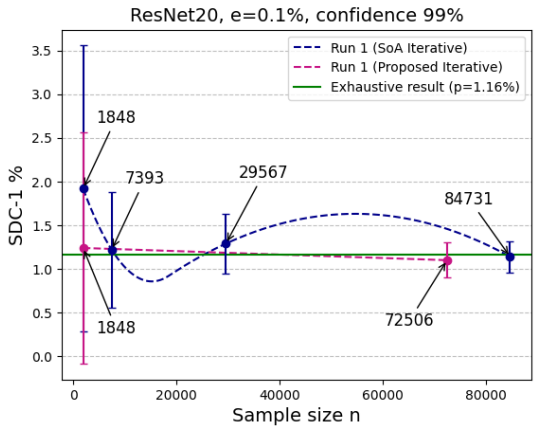

Fig. 5: Iterative SFI methods: a detailed execution of *Run 1*.

TABLE 10: Gains in terms of iterations reduction %.

Proposed	Avg. Iterations Reduction [%], confidence 99%					
	e=1%		e=0.1%		e=0.01%	
	Res.	Mob.	Res.	Mob.	Res.	Mob.
SoA Iterative	18.14	30.33	27.44	22.22	57.14	57.14

For each experiment, Table 10 reports the average reduction in total number of iterations (from an about 18% to more than 57% of fewer iterations when the margin of error is 0.01%).

Clearly, an optimal SFI approach is a good tradeoff between number of iterations (the lower, the better) and number of injected faults (the lower, the better). Injecting a few faults at a time may allow you to reduce the number of faults injected, but will dramatically increase the number of iterations. Actually, having many iterations means recalculating the failure rate p and the margin of error at the end of each iteration throughout the aggregation step (as shown in Figure 4).

As a final note, it is important to underline that the exact number of injected faults in *Iterative* approaches shown in all the Tables is strictly dependent on the measured probability of success ($\hat{p}$). This reasoning is sound: with a 99% confidence level, our sampling provides an estimate that encompasses the true (unknown) value within a range of $\hat{p} \pm$ one statistical error ($\hat{e}$). When the estimate of a given iteration falls in the positive range ([$\hat{p}$, $\hat{p}$ + $\hat{e}$]), dealing with very low failure rates, the number of faults to be injected in the next iteration is greater (the closer p is to 0.5, the more the number of faults to be injected [10]). When the estimate falls in the negative range ([$\hat{p}$ - $\hat{e}$, $\hat{p}$]), the number of faults to be injected in the next iteration is smaller. *Having more iterations increases the probability of sampling from the positive range, and thus of injecting more faults.* For the sake of generalization, iterative experiments in this research work

TABLE 11: Total number of FIs and iterations to achieve an estimate of SDC-1 faults in all layers with an error margin lower than 1%.

SFI Methods	e_{goal} 1%			
	One-Step		Iterative (Average over **10 runs**)	
	Data-unaware [15]	Data-aware [15]	SoA Iterative [16]	Proposed Approach
ResNet20				
Total FIs	307,650	201,079	19,708	15,416
Max. Iterations	1	1	6	4
MobileNetV2				
Total FIs	14,894,400	778,951	42,785	41,891
Max. Iterations	1	1	5	3

TABLE 12: Total number of FIs and iterations to achieve an estimate of SDC-1 faults in all layers with an error margin lower than 0.1%.

SFI Methods	e_{goal} 0.1%			
	One-Step		Iterative (Average over **10 runs**)	
	Data-unaware [15]	Data-aware [15]	SoA Iterative [16]	Proposed Approach
ResNet20				
Total FIs	8,877,535	1,933,997	1,177,447	1,102,509
Max. Iterations	1	1	5	4
MobileNetV2				
Total FIs	117,242,240	14,159,433	2,725,946	2,708,742
Max. Iterations	1	1	9	5

TABLE 13: Total number of FIs and iterations to achieve an estimate of SDC-1 faults in all layers with an error margin lower than 0.01%.

SFI Methods	e_{goal} 0.01%			
	One-Step		Iterative (Average over **10 runs**)	
	Data-unaware [15]	Data-aware [15]	SoA Iterative [16]	Proposed Approach
ResNet20				
Total FIs	16,989,188	11,497,921	14,000,886	13,963,231
Max. Iterations	1	1	8	3.3
MobileNetV2				
Total FIs	131,913,403	70,383,855	66,137,102	65,977,468
Max. Iterations	1	1	8	4

(both the SoA and the proposed) have been performed ten times. Changing the seed of the SFI experiments, numbers can oscillate, but the positive trend (i.e., reduced number of FIs and reduced number of iterations) is always confirmed.

For example, the exhact SDC-1% of MobileNetV2 is 1.11%. To achieve an estimate within the range of 1.11% $\pm$ 0.01% with 99% confidence, the real minimum and maximum sample size n can be computed, corresponding to 6,887,825 and 7,005,490, respectively. Although these figures most accurately reflect the experimental outcomes from running the iterative SFI (Table 8), they represent idealized values, computed with knowledge of the actual SDC-1 rate. In practice, estimates can deviate from this range during each iteration, either falling below [0, $\hat{p}$-$\hat{e}$) or exceeding ($\hat{p}$ + $\hat{e}$, 100%], depending on the confidence level. Increasing the number of iterations raises the likelihood of obtaining positive in-range estimates (i.e., [$\hat{p}$, $\hat{p}$ + $\hat{e}$]) or, in a worse scenario, positive out-of-range estimates (i.e., ($\hat{p}$ + $\hat{e}$, 100%]), both of which contribute to the growth of the sample size in subsequent iterations.

To conclude, as previously mentioned, in safety-critical scenarios the failure rate number is typically very low. An iterative approach dramatically reduces the number of fault injections without the need for preliminary investigation of the feature sought (e.g., as it happened with the data-aware approach [15]). As the margin of error decreases (e.g., e_{goal} =0.01%), the proposed SFI approach reduces the number of iterations by over 57% (Table 10), with only a slight drop in the number of injections (Table 9). Conversely, as the margin of error increases, the reduction in injections becomes more significant (Table 9). Notably, even a 1.65% of FIs reduction translates to approximately 117,000 fewer injected faults, compared with the SoA iterative solution (Table 9, Column 7$^{\text{th}}$).

5.2.2 *Layer-wise SFI*

As mentioned in Section 5.1, the intent of the layer-wise SFI analysis is to estimate the percentage of SDC-1 faults in every layer of the CNNs under investigations with three levels of precisions. A total of 20 and 54 separate fault lists (populations) have been considered for ResNet20 and MobileNetV2, respectively (as shown in Table 4). It is important to underline that fault lists have been divided because it is important to *guarantee the same maximum margin of error in every population*, as explained in [15]. For each experimental result, the flowchart describing the proposed approach, shown in Fig. 4 has been followed.

Tables 11, 12, and 13 report results in terms of total injected faults and average of the maximum number of iterations over ten different runs in both scenarios: one-step SFI approaches (Data-unaware and Data-aware [15]), and iterative SFI approaches ([16] and the proposed one). Layer-wise SFIs have been performed for both ResNet20 and MobileNetV2 with the intent of achieving e=1%, e=0.1% and e=0.01% margin of errors. It is fundamental to underline that in a layer-wise SFI an iteration is considered the complete execution of *all layers* of the CNN, until all layers have reached the desired margin of error. Therefore, the maximum value of iterations is calculated, and the average over ten different runs is reported in Tables 11, 12, and 13.

The proposed iterative SFI solution provides great advantages in fastly converging toward the statistical estimate with the desired margin of error with the minimal injected faults and minimal iterations, so providing benefits compared to state-of-the-art research works. The gains in terms of FI reduction and maximum interations reduction are reported in Tables 14 and 15, respectively.

Among the ten iterative SFI experiments, one random execution (*Run 4*) has been extracted: the layer-by-layer outcome is illustrated in Figure 6. The graph reports the results of a SFI performed to achieve the SDC-1% estimate on all layers with a 0.1% margin of error. As observed, the proposed iterative can reduce the total number of injected

TABLE 14: Gains in terms of FI reduction %.

Proposed	**Avg. FI Reduction [%], confidence 99%**					
	e=1%		**e=0.1%**		**e=0.01%**	
	Res.	**Mob.**	**Res.**	**Mob.**	**Res.**	**Mob.**
SoA One-Step	94.98	99.71	87.58	97.68	17.81	49.98
SoA One-Step with preliminary analysis	92.33	94.62	42.99	80.86	-21.44	6.26
SoA Iterative	21.77	2.08	6.36	0.63	0.26	0.24

TABLE 15: Gains in terms of iterations reduction %.

Proposed	Avg. Iterations Reduction [%], confidence 99%					
	e=1%		e=0.1%		e=0.01%	
	Res.	Mob.	Res.	Mob.	Res.	Mob.
SoA Iterative	33.33	40	20	44.44	58.75	50

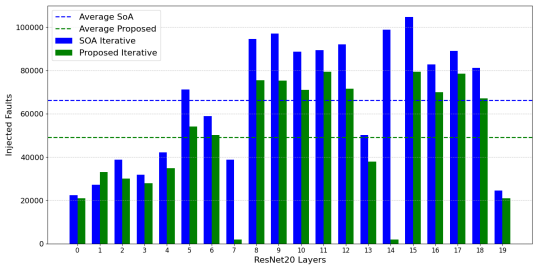

Fig. 6: Layer-wise SFI, *Run 4*: detailed information of layer-by-layer number of injected faults for achieving a 0.1% of margin of error in each ResNet20's layer.

faults in all layers except for layer1 (where 3% more faults are injected). This was due to an outlier in estimation (a FI result yielding a value outside the statistical range) that has increased the next sample size. As mentioned, iterative statistical injections are very susceptible to variations in $\hat{P}$. As for the remaining layers (all except layer1), the FI reduction obtained with the proposed iterative approach goes from 0.44% in layer17 to 12.51% in layer7. The other runs follow a similar trend.

Observing Tables 11, 12, and 13, it may appear that, overall, iterative approaches are the ones guaranteeing the minimal number of injected faults, for a specific error margin. Conceptually, it is true that iterative approaches have been created to optimize the number of injected faults, compared to generic one-step approaches (such as data-unaware with $p = 0.5$). However, if the one-step approach relies on a preliminary phase where p is pre-estimated based on selected data-characteristics (e.g., average-bit-flip-distance [15]), the one-step can yield superior performance than simple iterative approaches. This can be noted in Table 13 for ResNet20. Clearly, the cost for performing the pre-estimation of p is non-negligible, and it is highly domain- and data-dependent. The proposed approach does not require any preliminary information about the populations under study nor any heuristic approach to pre-estimate p as in the data-aware method proposed in [15].

A final observation must be done considering the results in Table 13 with a *level-3* precision (i.e., e=0.01%). With such a low error margin value, the number of injections to be performed is very large, regardless of the SFI technique. So, the reader may think that there is no great advantage of an iterative SFI over a One-Step solution. However, unrolling the SFI execution layer by layer (by assessing the percentage of injected faults required in each layer to achieve a 0.01% margin of error) clearly highlights the advantages of an iterative solution. Table 16 reports the population size for each layer (3rd Column) and the needed FIs for that error margin. A One-Step approach is less sensitive to population size, consistently injecting about 99% of faults in every layer. In contrast, an iterative solution can reduce the number of

TABLE 16: ResNet20: the larger the population of faults to be examined, the more advantageous it is to use an iterative approach to reduce the number of injections to be run.

LayerID	LayerType	Fault List Size (N)	Injected Faults [%] (e=0.01%, t=99%)	
			One-Step	Iterative Proposed
1	conv1	27,648	99.98	99.71
2	conv2	147,456	99.91	96.65
3	conv3	147,456	99.91	96.57
4	conv4	147,456	99.91	96.36
5	conv5	147,456	99.91	97.08
6	conv6	147,456	99.91	98.44
7	conv7	147,456	99.91	98.21
8	conv8	294,912	99.82	94.13
9	conv9	589,824	99.64	93.98
10	conv10	589,824	99.64	93.86
11	conv11	589,824	99.64	93.58
12	conv12	590,464	99.64	94.01
13	conv13	589,824	99.64	93.57
14	conv14	1,179,648	99.29	*79.02*
15	conv15	2,359,296	98.60	*77.85*
16	conv16	2,359,296	98.60	*78.49*
17	conv17	2,359,296	98.60	*75.31*
18	conv18	2,359,296	98.60	*78.37*
19	conv19	2,359,296	98.60	*75.59*
20	linear	40,960	99.97	99.11
Tot.		17,174,144	99.48	81.54

injections by up to 23% (*layer17*). This is noteworthy because it suggests that as models grow in size, an iterative approach becomes increasingly suitable. This aligns with the current trend of developing heavier deep learning models.

For the sake of completeness, we have performed a further experiment to discuss the benefit of One-Step and Iterative statistical FI experiments. We have selected two fault lists of two MobileNetV2's layers (i.e., *layer0* and *layer53*) having similar *exhaustive* failure rate (i.e., 1.36% and 1.27%, respectively) but different population (N) size. As shown in Table 17, *Layer0* accounts for a total of 55,296 stuck-at faults, while *Layer53* for a total of 26,214,400 faults. To achieve an estimate of the failure rate with an error margin of 0.01% and a confidence of 99%, statistical fault injections have been performed, and the experiment has been repeated three times (three runs are reported in each separate row). Data in Table 17 suggest that when the error margin and the population size are relatively low (e=0.01% and the population in the order of thousands), iterative FI methods iteratively approach the one-step number of FIs (around 99.38%). On the opposit, with bigger fault lists (*layer53*), iterative methods bring significant advantages in reducing the number of injections. Additionally, the proposed iterative method also dramatically reduces the total iterations (by more than 57%). Although the percentage reduction of FIs between the state-of-the-art iterative and the proposed approach may seem negligible, it is important to note that injecting 24.14% of the total faults instead of 26.04% in run-2 (*layer53*) translates to saving approximately 498,074 injected faults out of a population of over 26 million faults. And this is obtained with a 62.5% reduction of total iterations (from 8 to only 3 iterations).

TABLE 17: MobileNetV2: comparing two layers with similar success probabilities but different population sizes, highlighting when an iterative approach is beneficial. The analysis is carried out with a 0.01 % margin of error and 99% confidence.

Run	N	One-Step		SoA Iterative		Proposed	
		FIs [%]	Tot Iter.	FIs [%]	Tot Iter.	FIs [%]	Tot Iter.
Layer0							
run-0	55,296	99.38	1	99.38	8	99.39	3
run-1	55,296	99.38	1	99.41	8	99.38	3
run-2	55,296	99.38	1	99.45	8	99.37	3
Layer53							
run-0	26,214,400	86.39	1	25.55	7	24.25	3
run-1	26,214,400	86.39	1	25.8	8	24.21	3
run-2	26,214,400	86.39	1	26.04	8	24.14	3

5.2.3 *Bit-wise SFI*

The intent of a bit-wise SFI analysis is to estimate the SDC-1% of permanent faults in each bit-position of the FP32 representation. Each fault list includes all faults affecting a specific bit position of the entire set of CNN weights. As an example, all faults affecting the MSB (bit 31st) of the CNN weights constitute a single population of faults. Given that both CNNs use a FP32 bit-width representation, the exhaustive FIs have been performed by analysing 32 populations (fault lists) for each CNN, each sized as reported in Table 4. As for ResNet20, the exhaustive results show that from populations bit-0 (Least Significant Bit-LSB) to bit-19, the percentage of SDC-1 faults is zero: no faults affecting those bit positions lead to wrong predictions. Similarly, populations of faults from bit-0 to bit-22 (MobileNetV2) feature a SDC-1% equal to zero. In the literature, it is well-documented that incorrect predictions in CNNs, specifically SDC-1, are solely due to faults impacting the high-order exponent bits (e.g., [36], [32]). Faults in the sign and mantissa bits of FP32 synaptic weights do not result in CNN wrong predictions. Nevertheless, the proposed SFI flow has been applied, and experimental results are given in Tables 18, 19 and 20 for both cases of study (ResNet20 and MobileNetV2).

In this last section, instead of varying the error margin (as performed in Section 5.2.1 and 5.2.2), the confidence level was tuned, setting up three configurations: 95%, 99%, 99.9%. The confidence level, represented by the t variable in the statistical sampling formula (Eq. 4) assumes contant values (given in Tables 1). Having an estimate of the true value with a margin of error of 1% and a confidence of 95% means that in the 95% of cases, the estimate $\pm 1\%$ will cover the true (unknown) value. As shown in Tables 18, 19 and 20, tuning the confidence values does not have high impact on the sample size, but confirms the benefit of the proposed iterative solution over state-of-the-art approaches. However, it is important to note that the reductions in FI and the decrease in the number of iterations are not notably large. This is due to the characteristics of the 32 populations of faults and their exhaustive SDC-1% rate. Indeed, in all the populations (except for bit-30, i.e., the most significant bit of the 8-bit exponent part), the real probability of success is close to 0, or equal 0. It means that in one single iteration the e_{goal} is achieved, and the first iteration is the same among the SoA iterative and the proposed iterative. After one single iteration, for all of them the measured $\hat{e}$ already respected the e_{goal}.

TABLE 18: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with a 1% error margin and 95% confidence level.

SFI Methods	e_{goal} = 1%, confidence 95%		
	One-Step	Iterative (Average over 3 runs)	
	[10] and Data-unaware [15]	SoA [16]	Proposed Approach
ResNet20			
Total FIs	301,926	25,470	21,229
Total Iterations	1	4	3
MobileNetV2			
Total FIs	306,660	21,559	20,522
Total Iterations	1	4.33	4

TABLE 19: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with a 1% error margin and 99% confidence level.

SFI Methods	e_{goal} = 1%, confidence 99%		
	One-Step	Iterative (Average over 3 runs)	
	[10] and Data-unaware [15]	SoA [16]	Proposed Approach
ResNet20			
Total FIs	516,496	39,728	35,198
Total Iterations	1	4	3
MobileNetV2			
Total FIs	530,509	37,325	35,526
Total Iterations	1	4.33	4

5.3 Discussion

Overall, the research paper provides an overview of a variety of SFI techniques, comparing the proposed iterative approach with state-of-the-art solutions. Additionally, experimental results show that, when dealing with very small margins of error (such as 0.01%) and big fault lists, employing an iterative SFI approach is particularly beneficial. Therefore, the research work highlights a novel important aspect: it is not always advantageous to adopt iterative solutions. With small fault lists and small error margins, one-step methods achieve the final estimate with one single iteration, while iterative SFI ones inject about the same

TABLE 20: Total number of FIs to achieve an estimate of SDC-1 of the CNNs with a 1% error margin and 99.9% confidence level.

SFI Methods	e_{goal} = 1%, confidence 99.9%		
	One-Step	Iterative (Average over 3 runs)	
	[10] and Data-unaware [15]	SoA [16]	Proposed Approach
ResNet20			
Total FIs	731,323	63,347	58,330
Total Iterations	1	4	3
MobileNetV2			
Total FIs	759,733	53,512	50,931
Total Iterations	1	5	4

percentage with more iterations. Finally, three x-wise approaches have been discussed in the manuscript: they have been selected as representative SFI methods in the literature. The aim of the work is not to compare them in terms of performance and resilience. However, it is interesting to note that the bit-wise approach is the worst-case scenario, because the only fault list that requires more than one iteration is the population of bit 30. This population has a p close to 50%. However, even though the reduction is minimal, it confirms the effectiveness of the technique even with higher p values (much more uncommon in safety critical scenarios). To conclude, it is important to underline that the key contribution of the approach consists in the statistical iterative methodology, that can be applied to different types of dependable systems. Finally, we also want to emphasize that with the increasing complexity of modern chips and AI models, it becomes impossible to exhaustively test the entire fault list. Therefore, it is becoming increasingly essential to investigate innovative statistical techniques that allow for reducing the fault list while still obtaining statistically significant results. A statistical testing does not require any preliminary application- or data-specific knowledge or analysis or pruning of the population of faults under investigation. Its intent is not to find all critical faults, nor to prune the fault injection space: its goal is to provide a fast *system-agnostic* statistical measure of resilience.

6 Conclusions

The challenge of keeping acceptable the cost for reliability evaluation becomes increasingly important as CNN models get more complicated. In order to provide statistically meaningful findings, this paper proposes an iterative technique to execute statistical fault injections on CNNs. Compared to state-of-the-art approaches, it considerably reduces the number of simulations by (i) leveraging the initial estimate available after a few injected faults to further reduce the sample size (ii) reducing the number of iterations by proposing a margin of error that varies according to the measured failure rate p. The gains in terms of reduction in the number of faults to be injected depends on the target probability of success (the lower, the lower the sample size), the size of populations under investigations (the higher, the higher the benefit of the proposed technique), and the desired error margin. The effectiveness of the method was compared against previously proposed methods, and the provided experiments assessed the SFI proposal ability to significantly reduce the number of faults to be injected whereas guaranteeing a given error margin, and reducing the quantity of iterations.

Future works will target different fault models, different DNN typologies and datasets, different systems and designs (e.g., hardware units). Furthermore, we are examining the statistical relationship between the margin of error and population size to identify an optimal ratio that can be used as a stopping criterion.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

The authors would like to sincerely thank Professor Roberto Fontana, full professor of the Department of Mathematical Sciences "G. L. Lagrange" (DISMA) of the Politecnico di Torino, for valuable and stimulating conversations on statistics. The authors would like to sincerely thank the editors and the reviewers for their outstanding reviews and the help in improving the manuscript.

References

[1] A. E. Goodloe, "Assuring safety-critical machine learning-enabled systems: Challenges and promise," Computer, vol. 56, no. 09, pp. 83–88, sep 2023.

[2] J. Athavale, A. Baldovin, R. Graefe, M. Paulitsch, and R. Rosales, "Ai and reliability trends in safety-critical autonomous systems on ground and air," in 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2020, pp. 74–77.

[3] ISO/IEC JTC 1/SC 42, "ISO/IEC TR 5469:2024: Artificial intelligence, functional safety and ai systems," https://www.iso.org/standard/81283.html, 2024, accessed: February 12.

[4] European Commission, "AI Act," https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai, 2024, accessed: February 12.

[5] F. Su, C. Liu, and H.-G. Stratigopoulos, "Testability and dependability of ai hardware: Survey, trends, challenges, and perspectives," IEEE Design Test, vol. 40, no. 2, pp. 8–58, 2023.

[6] K. Pattabiraman, G. Li, and Z. Chen, "Error resilient machine learning for safety-critical systems: Position paper," in 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2020, pp. 1–4.

[7] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: scaling to trillion parameter models with simple and efficient sparsity," J. Mach. Learn. Res., vol. 23, no. 1, jan 2022.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. Ruospo, G. Gavarini, C. de Sio, J. Guerrero, L. Sterpone, M. Sonza Reorda, E. Sanchez, R. Mariani, J. Aribido, and J. Athavale, "Assessing convolutional neural networks reliability through statistical fault injections," in 2023 Design, Automation Test in Europe Conference Exhibition (DATE), 2023, pp. 1–6.

[16] I. Tuzov, D. de Andrés, and J.-C. Ruiz, "Accurate robustness assessment of hdl models through iterative statistical fault injection," in 2018 14th European Dependable Computing Conference (EDCC), 2018, pp. 1–8.
[17] 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.
[18] 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.
[19] 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.
[20] F. Goncalves, M. Santos, I. Teixeira, and J. Teixeira, "Self-checking and fault tolerance quality assessment using fault sampling," in 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2002. DFT 2002. Proceedings., 2002, pp. 216–224.
[21] B. Nie, L. Yang, A. Jog, and E. Smirni, "Fault site pruning for practical reliability analysis of gpgpu applications," in 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2018, pp. 749–761.
[22] S. K. S. Hari, S. V. Adve, H. Naeimi, and P. Ramachandran, "Relyzer: exploiting application-level fault equivalence to analyze application resiliency to transient faults," SIGPLAN Not., vol. 47, no. 4, p. 123–134, mar 2012. [Online]. Available: https://doi.org/10.1145/2248487.2150990
[23] 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.
[24] A. Lotfi et al., "Resiliency of automotive object detection networks on gpu architectures," in 2019 IEEE International Test Conference (ITC), 2019, pp. 1–9.
[25] B. Salami, O. S. Unsal, and A. C. Kestelman, "On the resilience of RTL NN accelerators: Fault characterization and mitigation," in 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). Lyon, France: IEEE, 2018, pp. 322–329.
[26] J. J. Zhang, T. Gu, K. Basu, and S. Garg, "Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator," in 2018 IEEE 36th VLSI Test Symposium (VTS), 2018, pp. 1–6.
[27] J. Leray, "Effects of atmospheric neutrons on devices, at sea level and in avionics embedded systems," Microelectronics Reliability, vol. 47, no. 9, pp. 1827–1835, 2007, 18th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis.
[28] P. H. Hochschild et al., "Cores that don't count," in Proceedings of the Workshop on Hot Topics in Operating Systems, ser. HotOS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 9–16.
[29] D. Xu et al., "Reliability evaluation and analysis of fpga-based neural network acceleration system," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 3, pp. 472–484, 2021.
[30] L. M. Luza et al., "Investigating the impact of radiation-induced soft errors on the reliability of approximate computing systems," in 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2020, pp. 1–6.
[31] G. Gavarini, A. Ruospo, and E. Sanchez, "Sci-fi: a smart, accurate and unintrusive fault-injector for deep neural networks," in 2023 IEEE European Test Symposium (ETS), 2023, pp. 1–6.
[32] B. Reagen, U. Gupta, L. Pentecost, P. Whatmough, S. K. Lee, N. Mulholland, D. Brooks, and G.-Y. Wei, "Ares: A framework for quantifying the resilience of deep neural networks," in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1–6.
[33] Y.-J. Jung, S.-H. Han, and H.-J. Choi, "Explaining cnn and rnn using selective layer-wise relevance propagation," IEEE Access, vol. 9, pp. 18 670–18 681, 2021.
[34] C. Bolchini, L. Cassano, A. Miele, and A. Nazzari, "Selective hardening of cnns based on layer vulnerability estimation," in 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2022, pp. 1–6.
[35] 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.
[36] G. Li, S. K. S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, and S. W. Keckler, "Understanding error propagation in deep learning neural network (dnn) accelerators and applications," in SC17: International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, pp. 1–12.

Annachiara Ruospo received the M.Sc. degree in computer engineering from the Politecnico di Torino, Italy, in 2018, where she is currently an Assistant Professor with the Department of Control and Computer Engineering. Her main research interests include safety, reliability, and security aspects of AI systems, with a focus on artificial neural networks, and test and verification of modern embedded devices. She is a Member of the AI Existential Safety Community of the Future of Life Institute.

Matteo Sonza Reorda (Fellow, IEEE) received the M.Sc. degree in electronics and the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 1986 and 1990, respectively. He is currently a Full Professor with the Department of Control and Computer Engineering, Politecnico di Torino. He is involved in numerous research projects with companies and other research centers worldwide. He published more than 400 papers in the area of test and fault-tolerant design of reliable circuits and systems. He received several best paper awards at major international conferences.

Riccardo Mariani , head of industry safety at NVIDIA, is widely recognized as an expert in functional safety. He is responsible for driving safety alignment across NVIDIA automotive and embedded teams, including for industrial, robotics and healthcare. Riccardo develops cohesive safety strategies and cross-segment safety processes, architecture and products that can be used across NVIDIA AI-based hardware and software platforms. He has leadership positions in international standardization initiatives such as part lead of ISO 26262-11, convenor of MT 61508-1/2, subgroup lead of ISO/PAS 8800 and editor of ISO/IEC TR 5469. He is chair of the IEEE Functional Safety Standards Committee. He received the 2021 Ron Waxman DASC Meritorious Service Award.

Ernesto Sanchez is an Associate professor at Politecnico di Torino, Italy. His research interests include digital circuits and systems reliability, evolutionary computation and ANN reliability. He received his Ph.D. degree in computing engineer from Politecnico di Torino in 2006. He is an IEEE senior member.