Learning sustainability by making games. The experience of a challenge as a novel approach for Education for Sustainable Development

Pros and Cons of Fault Injection Approaches for the Reliability Assessment of Deep Neural Networks

Annachiara Ruospo*, Lucas Matana Luza†, Alberto Bosio‡, Marcello Traiola‡, Luigi Dilillo† and Ernesto Sanchez*
*Politecnico di Torino, Torino, Italy. Email: {firstname.lastname}@polito.it
†LIRMM, Univ Montpellier, CNRS, Montpellier, France. Email: {lucas.matana-luza, dilillo}@lirmm.fr
‡Univ Lyon, ECL, INSA Lyon, CNRS, UCBL, CPE Lyon, INL, UMR5270, France. Email: {firstname.lastname}@ec-lyon.fr

***Abstract*—In the last years, the adoption of Artificial Neural Networks (ANNs) in safety-critical applications has required an in-depth study of their reliability. For this reason, the research community has shown a growing interest in understanding the robustness of artificial computing models to hardware faults. Indeed, several recent studies have demonstrated that hardware faults induced by an external perturbation or due to silicon wear out and aging effects can significantly impact the ANN inference leading to wrong predictions. This work classifies and analyses the principal reliability assessment methodologies based on Fault Injection at different abstraction levels and with different procedures. Some of the most representative academic and industrial works proposed in the literature are described and the principal advantages, and drawbacks are highlighted.**

***Index Terms*—Reliability, Fault Injection, Neural Networks**

I. Introduction

For their outstanding computational capabilities, Artificial Neural Networks (ANNs) have shown to be effective in various areas such as robotics, automotive, gaming, medical wearable devices. Since some of them are considered safety-critical, in the last decades, the research community has shown a growing interest in investigating the reliability of systems based on artificial neural networks [1], [2].

ANNs are brain-inspired models, and it is claimed that they hold a certain degree of robustness for their parallel and distributed structure and for the redundancy due to over-provisioning [3]. Indeed, they are made by more neurons than the minimal number required to perform a computation.

However, these models, when deployed, are executed on hardware devices, and, as the shrinking of semiconductor technology continues, devices are getting more prone to physical errors: the probability that parts of the hardware fail increases. Hence, even though researchers claim that neural networks are potentially safe due to their inner resilience properties, there is a serious need for evaluating their robustness, especially if deployed on safety-critical systems [4], [5].

Fault Injections (FIs) have been longly considered as appealing methods for evaluating the dependability of systems under test among all known testing methodologies [6]. Specifically, faults or errors are injected in the system under test, and its behaviour is observed with respect to the golden scenario. Dealing with ANNs, many different injection scenarios might arise: errors can be placed in neurons, synaptic weights, activators, and input images; faults can be injected in the hardware architecture running the network, i.e., either the physical silicon implementation or a Hardware Description Language (HDL) model. They are intentionally modified or turned off to evaluate the ANN's resilience in incorrect conditions. FIs can be performed at very different levels of abstraction: they can target only the ANN software to achieve a technology-independent characterization, or they can consider the entire system, including the ANN and the final hardware implementation, for a technology-dependent characterization of the system.

In this work, we provide an insight into the state-of-the-art FI methodologies targeting the assessment of ANN-based applications and highlight the principal advantages and drawbacks of their adoption. In Section II, a classification is proposed for the FI methodologies, and some of the most representative works in the literature are described. Section III provides an analysis of the principal pros and cons of the above-described techniques with respect to three metrics, i.e., Cost, Precision, FI Time. In the same section, open challenges and future directions are discussed. Finally, Section IV concludes the paper.

II. Proposed Classification of FI Methodologies

Different FI methodologies are described in the literature to facilitate the ANN reliability assessment. As also proposed in [7], the methodologies can be classified in three main categories: (*i*) Simulation-based, (*ii*) Platform-based, and (*iii*) Radiation-based. The different classification considers the abstraction level where the FI is performed and also the faults sources, which can be simulated, emulated, and even physically induced on the target platform. In the following, we present an overview of the different methodologies and frameworks developed for the purpose.

A. Simulation-based

The simulation-based is the most commonly used technique, and it targets the FI on a model of the ANN, which can assume

This study has been achieved thanks to the financial support of the VAN ALLEN Foundation (Contract No. UM 181387) and the Region Occitanie (Contract No. UM 181386), and by the projects "IDEX Lyon OdeLe".

2021 IEEE 22nd Latin American Test Symposium (LATS) | 978-1-6654-2057-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/LATS53581.2021.9651807

two different abstraction levels:

- **Software-level:** targets a high-level model of the ANN. In this case, no architectural aspects are considered.
- **Hardware-level:** targets a model of the ANN considering the hardware architecture aspects, e.g., register transfer level (RTL) or gate level.

The software-level FI approach provides great controllability [8], and it is usually faster when compared with other approaches. Software-level FIs rely on the creation of a high-level model of the ANN and the construction of a FI framework that can inject faults on the model, which can target, for example, the ANN's weights and biases. Since the controllability usually is high, this approach can characterize the vulnerabilities of the neural network, in which it is possible to determine the most sensitive layers, nodes, and data representations. It is possible to say that the FI at the software level may incur in low cost.

In the literature, software-level approaches have been proposed in several works. For example, in [9], the authors evaluated the reliability of two CNNs: LeNet-5 [10] and YOLO [11], by randomly injecting permanent faults on the weights of the CNNs at the software level. The outcomes from the FI campaign are then compared with a golden prediction to analyse the effects of the permanent faults. Furthermore, in [12], the impact of approximate computing techniques, e.g., the use of a reduced data representation, is also explored through simulation-based FIs. Additionally, we also record the growing interest in high-level open-source machine learning and deep learning frameworks, such as TensorFlow [13]. This framework is commonly used as a base for the development of simulation-based FI frameworks, as presented in [14]–[17].

Similar to the software-level, the hardware-level approach works on a simulation of an ANN model. However, this model should consider the hardware architecture where the ANN is intended to be deployed. At the software level, the FI is limited on the parameters of the neural networks (such as synaptic weights, activations values) and the high-level programming logic. Conversely, simulations at the hardware level enable a broader spectrum of possibilities for selecting weaknesses and faulty locations based on the hardware architecture. The hardware-level approach is achieved by running simulations in HDL (Hardware Description Language) models of the target hardware architecture running the ANN-based application. Therefore, the simulation time tends to increase.

The authors in [17] present a resilience analysis framework to study transient hardware errors in deep learning accelerators leveraging on high-level design information obtained from architectural descriptions [17]. Also, Li *et at* in [18] propose a framework to simulate the propagation of soft errors in DNN-based systems. Moreover, based on RTL models, Salami *et al.* in [19] present a simulation-based hardware-level FI framework for the vulnerability analysis of a hardware accelerator.

B. Platform-based

The platformed-based FI approach assesses the reliability of the target neural network by running it in the actual physical platform, e.g., FPGAs, embedded GPUs, CPUs, and memories. The faults are emulated in the platform by different means, such as injecting bit flips in the FPGA bitstream or configuration memory or inducing bit flips in registers and memory cells.

The platform-based category is present in a vast part of the state-of-the-art projects, where several frameworks have been proposed. For example, in [20], the authors propose a framework, named *Ares*, to inject faults in the memory hosting the network parameters. The framework injects bit-flips in the weights, the activations and the hidden states of DNNs. Furthermore, in [21], it is proposed the vulnerability assessment of an automotive detection neural network deployed in a Volta GPU, where single bit-flips are injected in the weights and inputs images. Also, De Sio *et al.* in [22] exploit the FPGA reconfigurability to emulate faults affecting the hardware, with a framework (i.e., *FireNN*) that applies random bit-flip injections in the configuration memory related to specific resources of the network architecture.

C. Radiation-based

The radiation-based FI category relies on the exposure of the system to an accelerated radiation source, e.g., atmospheric-like neutrons. Radiation is a well-known source of perturbations in electronics devices, inducing Single-Event Effects like Single-Event Upsets (SEUs), Single-Event Functional Interrupt (SEFI), among others [23]. Therefore, this methodology guarantees a highly accurate reliability assessment for specific environments of application such as space, atmosphere, nuclear power plants, medical irradiation facilities and particle accelerators. On the other hand, the hardware resources and facility access are costly.

In the literature, several radiation-based approaches have been proposed. For example, in [24], the authors evaluated the impact of neutron-induced SEEs on a CNN (LeNet-5) implemented with three different levels of approximation on the data representation. In this study, the target hardware is the memory device that hosts the network parameters and input images. A similar approach is presented in [25], where a 2- and 3-D Flash memory storing the weights of an ANN is exposed to X-ray irradiation. Also, in [26], three different NVIDIA GPU architectures are exposed to a neutron beam targeting the study of error propagation in computing resources. Finally, targeting an FPGA-based architecture, Libano *et al.* in [27] analyse the SEEs influence on three versions of a MNIST CNN implemented in a SRAM-based FPGA. Most of the works that assess the reliability of ANN applications via radiation-based approaches use atmospheric-like neutrons as a radiation source. However, as mentioned above, for a specific environment, specific radiation sources like protons, heavy-ions, electrons, among others, should be considered.

III. Comparison among FI Methodologies

The intent of this section is to compare and underline the principal advantages as well as the limitations of the

above-mentioned fault injection methodologies. Next, the open challenges are discussed.

We introduce three different metrics for presenting the trade-offs between the state-of-the-art fault injection approaches:

- **Costs:** It refers to the costs needed to carry out the reliability assessment, including both resources and time.
- **Precision:** It means how much the FI procedure is close to reality, and the obtained results are accurate and realistic.
- **Fault Injection Time:** The amount of time that the injection process takes to complete a single injection cycle.

In Fig. 1, we assign a specific level to these metrics and grade them as *Low*, *Medium-Low*, *Medium*, *High-Medium*, *High*, and *Very High*.

As for the costs, because simulation-based approaches do not need the development and purchase of specific electronic devices to conduct the assessments, they are the most cost-effective. Moreover, when the HDL model is available, the costs are low and, anyhow, reduced compared to other FI techniques. When working with platform-based techniques, economic costs increase since they rely on the purchase and usage of specific validation or emulation devices (e.g., GPUs, CPUs, and FPGAs). A further benefit is also due to the fact that (i) they can be utilized again once the FI campaigns are completed, and (ii) they can be parallelized to increase the performance. The radiation-based procedures are the most expensive ones for three principal reasons: access to an irradiation facility, hardware setup development, and the low possibility of reusing the irradiated devices.

The precision with which the four FI procedures provide findings varies and is dependent on how well these approaches simulate the incidence of realistic system faults and how near they are to real. Radiation-based FI techniques achieve the maximum level of precision, as radiation-induced faults directly impact the silicon implementation of the device under test. This enables the DNN model to be accurately characterized. On the other hand, simulation-based hardware-level FIs have a good level of precision. Due to the adoption of the HDL model (either RTL or gate-level), their injection procedure can be considered close to the actual silicon implementation. For this reason, they are credited with a medium-high precision level. Contrarily, platform-based and simulation-based software-level FIs both have a low-medium precision level for the following reasons. The incidence of realistic hardware faults is mimicked using sophisticated software fault models, where errors are injected at the software or algorithmic level. Specifically, when a DNN model is written in C or C++, it can be compiled and executed directly on a physical hardware device; hence, the injected software errors can be close to the faults they attempt to reproduce.

In higher-level programming languages or tools, such as Python, PyTorch, and TensorFlow, FI frameworks introducing errors at the algorithmic level are exposed to a more elaborate compilation chain. Consequently, the lower the programming language level adopted for the DNN application, the higher the precision. One of the advantages of conducting simulation-based reliability assessments at the software level is the possibility of characterizing the vulnerability of neural networks independently from the target hardware device and, in particular, driving analyses on layers, data types, weights, and network's parameters. However, when a more thorough reliability evaluation is necessary, the injection campaigns should additionally encompass the target hardware that will ultimately execute the DNN under test, clearly when the device's HDL model, whether RTL or gate-level, is provided. In this second scenario (such as in [7], [19]), hardware-level FIs can achieve better accuracy of the results, closer to the silicon implementation. In a recent paper [28], the authors propose using realistic fault models (retrieved from radiation test campaigns) to inject at the software level in a CNN application, with the aim of enhancing the precision level of simulation-based FIs at the software level.

Regarding the last metric (i.e., the fault injection time), it is worth considering that it is very difficult to compare exactly the time required to run a single FI among the existing fault injection approaches. Indeed, many variables are involved and responsible for determining the fault injection time, such as the parallelization of the experiments, the adopted tools, the specific radiation source.

The problem associated with simulation-based FIs at the hardware level is that HDL simulations are extremely time-consuming. Clearly, it depends on the complexity of the neural networks under assessment and their HDL description. For example, a small CNN with only seven layers can take about 25 minutes to run a single inference [7]. Furthermore, existing commercial fault simulation tools are not tuned and neither optimized to face the complexity of the state-of-the-art DNN applications [29] (with billions of neuronal computations). This means that a FI at the hardware level is accurate but very costly in terms of simulation time. Therefore, reliability assessments at the hardware level typically consider only neural networks of limited size: a 6-layer fully connected classifier in [19] and a 7-layer CNN in [7]. By contrast, simulation-based FIs at the software level are not concerned with this non-negligible limitation. Actually, the neural network under consideration can range from 2-layer neural networks to more complex and deep networks, such as VGGNet and ResNet.

A. Open Challenges

Even though a lot of effort has been spent so far to develop specific FI frameworks for the reliability assessment of neural networks, in this section, we would like to highlight the open issues in this field that may need significant research and innovation efforts. A systematic FI into the configuration memory of FPGAs, more specific SRAM-based ones, can be satisfactory for some specific cases. Still, it can not be generalized to assess the device or system reliability since, as shown in [30], the results vary from device to device. Furthermore, the temperature has shown influence in the FIs results. The study was based on sixteen Xilinx Artix-7 and ten Lattice iCE40, which have their bitstream documentation

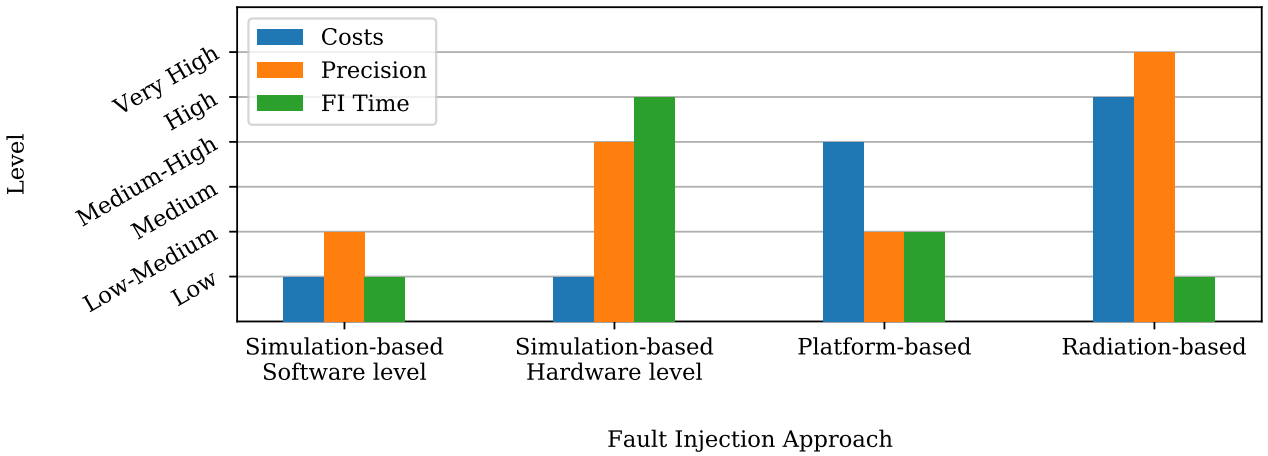

Fig. 1. Comparing the Fault Injection Methodologies.

publicly available. Overall, it means that several aspects should be considered, and parallelising the FI procedure may lead to unrealistic results. In the literature, several works can be seen on the effects of transient and permanent faults. However, a deeper investigation on other fault models, such as delays, bridging, and open lines, is still an open challenge and would be able to cover the newer fault mechanisms in deep submicrometer technologies. Moreover, the use of the FI methodologies brings pros and cons, and the research community is also looking forward to a hybrid solution, which could use the advantages of different methodologies and merge into one. An interesting example is presented in [31], where the reliability of CNNs executed in GPUs is investigated using microarchitectural simulations and software FI. Also, FI tools, as the one presented in [32], have been used for the reliability assessment of generic applications. We believe that for future work, this study can serve as a basis for researchers wishing to identify the best hybrid strategy that considers all the metrics examined (eg Cost, Precision, Fault Injection Time).

IV. Conclusions

Following the growing trend on ANNs, which have been used in various sectors, such as computer science, medicine, safety-critical applications, we have presented in this article the pros and cons of the different approaches for assessing the reliability of ANN-based applications. Among the existing testing techniques, fault injections are the most commonly used solutions to measure the dependability of systems under test. We have identified three main categories: simulation-, platform-, and radiation-based FI methodologies. This classification is based on their abstraction level, the source of errors (internal or external), and the overall FI procedure. These methodologies were compared in terms of cost, precision and FI time, exploring open challenges. Our overview of the topic reveals that state-of-the-art ANNs still require more accurate metrics and tools to manage their complexity and shows that this discussion is relevant to the advancement of this area.

References

[1] C. Torres-Huitzil and B. Girau, "Fault and error tolerance in neural networks: A review," *IEEE Access*, vol. 5, pp. 17 322–17 341, Aug. 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2742698

[2] E. B. Tchernev, R. Mulvaney, and D. Phatak, "Investigating the fault tolerance of neural networks," *Neural Computation*, vol. 17, no. 7, pp. 1646–1664, Jul. 2005. [Online]. Available: https://doi.org/10.1162/0899766053723096

[3] S. Lawrence, C. Giles, and A. Tsoi, "What size neural network gives optimal generalization? convergence properties of backpropagation," Institute for Advanced Computer Studies, Univ. of Maryland, Tech. Rep. UMIACS-TR-96-22 and CS-TR-3617, 1996.

[4] 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*, vol. 42, no. 6, pp. 358–361, Jun. 1995. [Online]. Available: https://doi.org/10.1109/81.390269

[5] A. Ruospo and E. Sanchez, "On the reliability assessment of artificial neural networks running on ai-oriented mpsocs," *Applied Sciences*, vol. 11, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/14/6455

[6] H. Ziade, R. Ayoubi, and R. Velazco, "A survey on fault injection techniques," *Int. Arab J. Inf. Technol.*, vol. 1, no. 2, pp. 171–186, Jul. 2004.

[7] A. Ruospo, A. Balaara, A. Bosio, and E. Sanchez, "A pipelined multi-level fault injector for deep neural networks," in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. Frascati, Italy: IEEE, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/DFT50435.2020.9250866

[8] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, "Fault injection techniques and tools," *Computer*, vol. 30, no. 4, p. 75–82, Apr. 1997. [Online]. Available: https://doi.org/10.1109/2.585157

[9] 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.: IEEE, 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/LATW.2019.8704548

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998, doi: 10.1109/5.726791.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

[12] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933121004786

[13] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. USA: ACM, USENIX Association, 2016, p. 265–283. [Online]. Available: https://dl.acm.org/doi/10.5555/3026877.3026899

[14] Z. Chen, N. Narayanan, B. Fang, G. Li, K. Pattabiraman, and N. DeBardeleben, "Tensorfi: A flexible fault injection framework for tensorflow applications," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. Coimbra, Portugal: IEEE, Oct. 2020, pp. 426–435. [Online]. Available: https://doi.org/10.1109/ISSRE5003.2020.00047

[15] M. Beyer, A. Morozov, K. Ding, S. Ding, and K. Janschek, "Quantification of the impact of random hardware faults on safety-critical AI applications: CNN-based traffic sign recognition case study," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. Oct.: IEEE, 2019, pp. 118–119. [Online]. Available: https://doi.org/10.1109/ISSREW.2019.00058

[16] Z. Chen, G. Li, K. Pattabiraman, and N. DeBardeleben, "Binfi: An efficient fault injector for safety-critical machine learning systems," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. New York, NY, USA: ACM, 2019, pp. 1 – 23. [Online]. Available: https://doi.org/10.1145/3295500.3356177

[17] 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. [Online]. Available: https://doi.org/10.1109/MICRO50266.2020.00033

[18] 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. [Online]. Available: https://doi.org/10.1145/3126908.3126964

[19] 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. [Online]. Available: https://doi.org/10.1109/CAHPC.2018.8645906

[20] 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 *Proceedings of the 55th Annual Design Automation Conference*. San Francisco, California, USA: Association for Computing Machinery, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3195970.3195997

[21] A. Lotfi, S. Hukerikar, K. Balasubramanian, P. Racunas, N. Saxena, R. Bramley, and Y. Huang, "Resiliency of automotive object detection networks on GPU architectures," in *2019 IEEE International Test Conference (ITC)*. Washington, DC, USA: IEEE, 2019, pp. 1–9. [Online]. Available: https://doi.org/10.1109/ITC44170.2019.9000150

[22] C. De Sio, S. Azimi, and L. Sterpone, "An emulation platform for evaluating the reliability of deep neural networks," in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. Frascati, Italy: IEEE, 2020, pp. 1–4. [Online]. Available: https://doi.org/10.1109/DFT50435.2020.9250872

[23] P. E. Dodd *et al.*, "Impact of heavy ion energy and nuclear interactions on single-event upset and latchup in integrated circuits," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, pp. 2303–2311, 2007. [Online]. Available: https://doi.org/10.1109/TNS.2007.909844

[24] L. Matana 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)*, Oct. 2020, pp. 1–6, doi: 10.1109/DFT50435.2020.9250865.

[25] M. M. Hasan, M. Raquibuzzaman, I. Chatterjee, and B. Ray, "Radiation tolerance of 3-d nand flash based neuromorphic computing system," in *2020 IEEE International Reliability Physics Symposium (IRPS)*. Dallas, TX, USA: IEEE, 2020, pp. 1–4. [Online]. Available: https://doi.org/10.1109/IRPS45951.2020.9128219

[26] F. Fernandes dos Santos, L. Draghetti, L. Weigel, L. Carro, P. Navaux, and P. Rech, "Evaluation and mitigation of soft-errors in neural network-based object detection in three GPU architectures," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. Denver, CO, USA: IEEE, 2017, pp. 169–176. [Online]. Available: https://doi.org/10.1109/DSN-W.2017.47

[27] F. Libano, P. Rech, B. Neuman, J. Leavitt, M. J. Wirthlin, and J. S. Brunhaver, "How reduced data precision and degree of parallelism impact the reliability of convolutional neural networks on fpgas," *IEEE Transactions on Nuclear Science*, pp. 1–1, 2021, (Early Access). [Online]. Available: https://doi.org/10.1109/TNS.2021.3050707

[28] L. M. Luza, A. Ruospo, A. Bosio, E. Sanchez, and L. Dilillo, "A model-based framework to assess the reliability of safety-critical applications," in *2021 24th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, 2021, pp. 41–44. [Online]. Available: https://doi.org/10.1109/DDECS52668.2021.9417059

[29] A. Floridia, E. Sanchez, and M. Sonza Reorda, "Fault grading techniques of software test libraries for safety-critical applications," *IEEE Access*, vol. 7, pp. 63 578–63 587, May 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2917036

[30] C. Fibich, R. Obermaisser, and M. Horauer, "Device- and temperature dependency of systematic fault injection results in artix-7 and ice40 fpgas," in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*. Grenoble, France: IEEE, 2021, pp. 1600–1605. [Online]. Available: https://doi.org/10.23919/DATE51398.2021.9474161

[31] J. E. R. Condia, F. F. dos Santos, M. S. Reorda, and P. Rech, "Combining architectural simulation and software fault injection for a fast and accurate cnns reliability evaluation on gpus," in *2021 IEEE 39th VLSI Test Symposium (VTS)*, 2021, pp. 1–7.

[32] A. Ejlali, S. G. Miremadi, H. R. Zarandi, G. Asadi, and S. B. Sarmadi, "A hybrid fault injection approach based on simulation and emulation co-operation," in *2003 International Conference on Dependable Systems and Networks, DSN 2003. Proceedings.* San Francisco, CA, USA: IEEE, 2003, pp. 479–488. [Online]. Available: https://doi.org/10.1109/DSN.2003.1209958