A survey of hardware error fault tolerance for deep learning

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This literature review surveys the landscape of hardware fault tolerance for deep learning, with a focus on strategies to mitigate the impact of hardware errors on deep learning computations. The paper explores various thematic sections, including training and retraining strategies for fault tolerance, hardware-software co-design for error resilience, fault detection, prediction, and correction techniques, evaluation and mitigation of permanent and transient hardware faults, advanced techniques and frameworks for fault tolerance, and broader perspectives on reliability and robustness in deep learning systems. Each section provides a comprehensive review of relevant literature, highlighting the significance of hardware fault tolerance in deep learning and presenting advanced techniques and frameworks that contribute to the fault tolerance of deep learning systems. The survey aims to provide a comprehensive understanding of the current state of research in hardware fault tolerance for deep learning and to identify potential avenues for future exploration and development in this critical area.**

***Index Terms*—**

I. Introduction

Deep learning has emerged as a powerful tool for various applications, including safety-critical domains such as automotive, avionics, medical, and industrial systems. However, the reliability of deep learning models can be compromised by hardware errors, leading to computational errors and potential safety hazards. In response to this challenge, researchers have been exploring hardware fault tolerance techniques to enhance the resilience of deep learning models against such errors.

This literature survey explores the landscape of hardware fault tolerance for deep learning, focusing on strategies to mitigate the impact of hardware errors on deep learning computations. The survey is organized into several thematic sections, each addressing specific aspects of hardware fault tolerance in deep learning.

The first section, "Training and Retraining Strategies for Fault Tolerance," delves into various training and retraining methodologies designed to enhance the fault tolerance of deep learning models against hardware errors. It discusses fault-aware training, incremental network approximation, and retraining-based timing error mitigation techniques.

The subsequent sections cover hardware-software co-design for error resilience, fault detection, prediction, and correction techniques, evaluation and mitigation of permanent and transient hardware faults, advanced techniques and frameworks for fault tolerance, and broader perspectives on reliability and robustness in deep learning systems.

Each section provides a comprehensive review of relevant literature, highlighting the significance of hardware fault tolerance in deep learning and presenting advanced techniques and frameworks that contribute to the fault tolerance of deep learning systems.

Overall, this literature survey aims to provide a comprehensive understanding of the current state of research in hardware fault tolerance for deep learning and to identify potential avenues for future exploration and development in this critical area.

II. Training and Retraining Strategies for Fault Tolerance

This section explores various training and retraining methodologies designed to enhance the fault tolerance of deep learning models against hardware errors. We will discuss fault-aware training, incremental network approximation, and retraining-based timing error mitigation techniques.

A. Fault-Aware Training

Zahid et al. proposed a novel methodology called fault-aware training (FAT) to enhance the resilience of quantized neural networks (QNNs) against hardware faults [1]. The authors addressed the need for functional safety in safety-critical applications, such as automotive, avionics, medical, and industrial systems, where hardware faults must be mitigated to en-

Identify applicable funding agency here. If none, delete this.

sure reliable inference. By injecting faults in the convolutional layers during training, highly accurate convolutional neural networks (CNNs) were trained, exhibiting much better error tolerance compared to the original. Furthermore, redundant systems built from QNNs trained with FAT achieved higher worst-case accuracy at lower hardware cost. This methodology provides a domain-specific solution to exploit the inherent features of DNNs, thereby decreasing the hardware cost for achieving functional safety.

B. Incremental Network Approximation

Liu et al. introduced the Incremental Network Approximation (INA) algorithm, a hardware-software co-design algorithm aimed at improving fault tolerance in DNNs [2]. INA addresses the convergence problem and promotes fault tolerance of DNNs, yielding tradeoffs between accuracy and implementation cost. The experiments conducted by the authors demonstrated that the approximate inference models retrained by INA could achieve significant hardware reduction while maintaining high classification accuracy. This algorithm provides an example of co-design algorithms that enhance fault tolerance in deep learning hardware.

C. Training Techniques for Fault Tolerant Neural Networks

Chiu et al. explored training techniques to improve fault tolerance in neural networks, presenting methods to coerce weights to have low magnitudes during the backpropagation training process and to add artificial faults to various components of a network during training [3]. The experimental results showed that these methods can obtain better robustness than backpropagation training and compare favorably with other approaches. This paper provides insights into training strategies that can lead to more fault-tolerant deep learning models.

D. Retraining-Based Timing Error Mitigation

Deng et al. discussed retraining neural network accelerators to mitigate timing errors, a specific aspect of hardware fault tolerance in deep learning [4]. The authors leveraged the error resiliency of neural networks to mitigate timing errors in NN accelerators, proposing to retrain the accelerators to update their weights when timing errors significantly affect the output results. The experimental results demonstrated that timing errors in NN accelerators can be effectively mitigated for different applications. This paper illustrates how retraining can be used as a strategy for hardware fault tolerance in neural network accelerators.

III. Hardware-Software Co-Design for Error Resilience

In this section, we delve into the intersection of hardware and software design to achieve fault tolerance in deep learning systems. We will examine co-design algorithms, predictive design paradigms, and the role of algorithm-hardware co-design in resilient deep learning inference.

A. Co-Design Algorithms for Fault Tolerance

The concept of hardware-software co-design is crucial for achieving fault tolerance in deep learning systems. Liu et al. proposed the Incremental Network Approximation (INA) algorithm, which is a hardware-software co-design approach aimed at improving fault tolerance in Deep Neural Networks (DNNs) [2]. INA addresses the convergence problem associated with highly approximate arithmetics in DNNs, promoting fault tolerance and offering tradeoffs between accuracy and implementation cost. The experiments conducted by Liu et al. demonstrated that the approximate inference models retrained by INA could achieve significant hardware reduction while maintaining high classification accuracy, showcasing the effectiveness of co-design algorithms in enhancing fault tolerance in deep learning hardware.

Furthermore, Tambe et al. introduced an algorithm-hardware co-design centered around a novel floating-point inspired number format, AdaptivFloat, which dynamically maximizes and optimally clips its available dynamic range to create faithful encodings of neural network parameters [5]. This approach consistently produced higher inference accuracies compared to conventional quantization methods at low bit precision, demonstrating the potential of algorithm-hardware co-design in improving fault tolerance and resilience in deep learning inference.

B. Predictive Design Paradigms for Error Resilience

Predictive design paradigms play a significant role in improving error resilience in deep learning hardware. Pandey et al. presented GreenTPU, a predictive design paradigm for improving timing error resilience of a near-threshold Tensor Processing Unit (TPU) [6]. By identifying patterns in error-causing activation sequences and intermittently boosting the operating voltage of specific multiplier-and-accumulator units, GreenTPU enables higher performance in an NTC TPU with minimal loss in prediction accuracy. This predictive design approach showcases the potential of hardware design paradigms in enhancing error resilience and energy efficiency in deep learning hardware.

C. Balancing Software Solutions and Hardware Endurance

Song et al. proposed a software and hardware co-design methodology to address imperfections in RRAM-crossbar-based DNN accelerators, aiming to preserve classification accuracy with few on-device training iterations [7]. This approach leverages the inherent self-healing capability of the neural network and dynamic adjustment mechanisms to prevent and mitigate errors induced by imperfect memristors. By balancing software solutions and hardware endurance, this co-design methodology effectively guarantees minimal loss of accuracy even in the presence of resistance variations and stuck-at-faults (SAFs). This work highlights the importance of integrating software and hardware solutions to achieve fault tolerance and error resilience in deep learning hardware.

In summary, the integration of hardware and software design is essential for achieving fault tolerance and error resilience in

deep learning systems. Co-design algorithms, predictive design paradigms, and the balance between software and hardware solutions play crucial roles in enhancing fault tolerance and resilience in deep learning hardware. These approaches not only address the challenges posed by hardware errors but also pave the way for more efficient and reliable deep learning systems.

IV. Fault Detection, Prediction, and Correction Techniques

This section focuses on the mechanisms for detecting, predicting, and correcting faults in deep learning hardware. We will cover methods for neuron resilience prediction, fault detection and remedy frameworks, and safety design techniques for error localization and correction.

A. Neuron Resilience Prediction

Accurate prediction of neuron resilience is crucial for managing reliability in neural network hardware accelerators. Schorn et al. proposed a method for predicting the error resilience of neurons in deep neural networks, which significantly improves upon existing methods in terms of accuracy and interpretability [8]. By simulating hardware faults in networks trained on image classification benchmarks, the authors demonstrated the effectiveness of their resilience prediction method and its potential for a flexible trade-off between reliability and efficiency in neural network hardware accelerators.

Furthermore, Wang et al. introduced an online fault detection method, Adversarial Testing (AT), tailored for neural network accelerator chips. This function-level testing method exhibits negligible run-time overhead and super sensitivity to subtle hardware variations, ensuring the normal use of deep learning accelerators during their lifetime [9].

These predictive methods for neuron resilience provide valuable insights into enhancing the reliability and fault tolerance of deep learning hardware.

B. Fault Detection and Remedy Frameworks

Li et al. presented RRAMedy, a framework for in-situ fault detection and network remedy for memristor-based neural accelerators. The proposed Adversarial Example Testing accurately detects defected cells and memristor soft faults, with the model accuracy being restored through edge-cloud collaborative fault-masking retraining and model updating mechanisms [10]. This framework effectively protects the neural accelerator from accuracy and performance degradation throughout its life cycle.

Moreover, Khoshavi et al. proposed SHIELDeNN, an end-to-end inference accelerator framework that synergizes the mitigation approach and computational resources to realize a low-overhead error-resilient Neural Network (NN) overlay. By developing a rigorous fault assessment paradigm, SHIELDeNN improves the error-resiliency magnitude of neural network parameters, thereby enhancing fault tolerance [11].

These fault detection and remedy frameworks offer practical solutions for mitigating errors in neural network hardware and improving fault tolerance.

C. Safety Design Techniques for Error Localization and Correction

Xu et al. introduced safety design techniques, including Algorithm Based Atomic Error Checking (ABAEC-1 and ABAEC-2), for a Weight Stationary Convolutional Neural Network (CNN) accelerator. These techniques focus on low latency and low overhead error detection and correction without performance degradation. The proposed design not only detects errors on-the-fly but also performs error diagnosis to localize the errors for on-line fault management and recovery [12].

Additionally, He et al. presented FIdelity, a resilience analysis framework for deep learning accelerators that accurately and quickly analyzes the behavior of hardware errors. By modeling transient errors in logic components, FIdelity ensures the reliability requirements are met for safe deployment in a wide range of applications, including safety-critical scenarios such as self-driving cars [13].

These safety design techniques provide insights into error detection, localization, and correction without compromising the performance of CNN hardware accelerators.

In summary, the literature reviewed in this section demonstrates a variety of approaches for detecting, predicting, and correcting faults in deep learning hardware. These methods contribute to the development of fault-tolerant hardware for deep learning systems, ensuring reliability and resilience in safety-critical applications.

V. Evaluation and Mitigation of Permanent and Transient Hardware Faults

Here, we review the impact of both permanent and transient faults on deep learning hardware and discuss various mitigation strategies. We will consider the resilience of different neural network architectures to hardware errors and the role of fault-tolerant design in accelerators.

A. Evaluation of Permanent Faults in Neural Network Accelerators

Permanent faults in hardware components can significantly impact the performance and reliability of neural network accelerators. The work by Gambardella et al. [14] provides valuable insights into the evaluation of permanent faults affecting Quantized Neural Networks (QNNs) and methods to decrease their effects in hardware accelerators. The study utilizes FPGA-based hardware accelerated error injection to evaluate the impact of permanent faults on QNNs, demonstrating that QNNs containing convolutional layers are not as robust to faults as commonly believed. The findings emphasize the importance of assessing the robustness of neural network architectures to permanent faults and highlight the need for effective fault-tolerant design in hardware accelerators.

Additionally, Mahdiani et al. [15] discuss the development of relaxed fault-tolerant techniques for VLSI implementation of neural networks, focusing on cost-effective fault tolerance that leverages the inherent resilience of neural networks. The proposed relaxed fault-tolerant techniques offer insights into mitigating the impact of permanent faults in neural network hardware implementations, providing a valuable perspective on achieving fault tolerance without significant performance degradation or cost escalation.

B. Mitigation of Transient Faults in Hardware Accelerators

Transient faults, such as single event upsets (SEUs) in FPGA-based accelerators, pose significant challenges to the fault tolerance of deep learning hardware. Li et al. [16] present a fault-tolerant design for Convolutional Neural Network (CNN) accelerators on FPGAs, addressing the impact of SEUs and proposing error mitigation techniques. The study analyzes the sensibility of CNNs to SEUs and introduces fault-tolerant design strategies, offering practical methods to achieve fault tolerance in FPGA-based hardware accelerators.

Moreover, Libano et al. [17] evaluate the impact of radiation-induced errors in neural networks implemented in FPGAs and propose a selective hardening strategy to mitigate the effects of such errors. The selective hardening approach, which triplicates only the most vulnerable layers of the neural network, provides a targeted and efficient method for mitigating transient faults in hardware implementations. This work serves as a case study for hardware error fault tolerance strategies in neural network implementations on FPGAs, shedding light on effective mitigation techniques for transient faults.

C. Resilience of Neural Network Architectures to Hardware Errors

Understanding the resilience of neural network architectures to hardware errors is crucial for developing effective fault-tolerant designs. Arechiga et al. [18] investigate the robustness of modern deep learning architectures against single event upset errors, focusing on the resilience of Convolutional Neural Networks (CNNs) to bit flips in their weights. The study provides valuable insights into the resilience of various neural network architectures to hardware errors, offering a comprehensive analysis of the impact of errors on network performance and identifying architectural factors that contribute to greater robustness.

Furthermore, Salami et al. [19] study fault characterization and mitigation in Register-Transfer Level (RTL) models of neural network accelerators, providing a detailed analysis of the vulnerability of various components of RTL neural network implementations and proposing low-overhead fault mitigation techniques. This work serves as an example of fault characterization and mitigation techniques in hardware accelerators for neural networks, contributing to the understanding of fault tolerance at the architectural level.

In summary, the evaluation and mitigation of permanent and transient hardware faults in neural network accelerators are essential for ensuring the reliability and robustness of deep learning hardware. By considering the insights and strategies presented in the referenced literature, researchers and practitioners can advance the development of fault-tolerant designs and enhance the resilience of hardware implementations for deep learning applications. Additionally, the investigation of the resilience of neural network architectures to hardware errors provides valuable guidance for the design and optimization of fault-tolerant hardware accelerators.

VI. Advanced Techniques and Frameworks for Fault Tolerance

This section presents advanced techniques and frameworks that contribute to the fault tolerance of deep learning systems. We will explore the use of clipped activation, sensitivity-based techniques, dynamic quantization, and Bayesian approaches for assessing and improving hardware error resilience.

A. Clipped Activation for Improving Fault Tolerance

Hoang et al. proposed a novel error mitigation technique, FT-ClipAct, which focuses on improving the resilience of Deep Neural Networks (DNNs) to hardware faults [20]. The technique involves replacing the unbounded activation functions with their clipped versions to alleviate the impact of high-intensity faulty activation values. By systematically defining the clipping values of the activation functions, the resilience of the networks against faults is significantly improved. Experimental results on the AlexNet and VGG-16 DNNs trained for the CIFAR-10 dataset demonstrated a substantial improvement in classification accuracy, particularly at higher fault rates.

This approach is valuable for enhancing fault tolerance in deep learning systems by addressing the impact of hardware faults on DNN parameters. It provides a practical method for mitigating errors caused by hardware circuit faults, thereby contributing to the overall fault tolerance of deep learning systems.

B. Sensitivity-Based Techniques for Energy-Efficient DNN Accelerators

Choi et al. introduced sensitivity-based error resilient techniques for energy-efficient DNN accelerators, focusing on enabling aggressive voltage scaling by exploiting different levels of error resilience within DNN layers, filters, and channels [21]. The proposed techniques leverage the sensitivity variation among filter weights to design DNN accelerators that assign computations with more sensitive weights to more robust processing units, thereby achieving energy savings without significant accuracy loss.

This work is relevant to the topic of hardware fault tolerance as it addresses the energy-efficient design of DNN accelerators while considering the impact of hardware errors on the resilience of deep learning computations. By incorporating sensitivity-based techniques, DNN accelerators can effectively mitigate the effects of hardware faults, contributing to improved fault tolerance in deep learning systems.

C. Dynamic Quantization for DNN Acceleration

Song et al. presented a dynamic region-based quantization technique, DRQ, for deep neural network acceleration, which dynamically adjusts the precision of a DNN model based on sensitive regions in the feature map to achieve greater acceleration while preserving accuracy [22]. The proposed technique identifies sensitive regions in the feature map and utilizes a variable-speed mixed-precision convolution array to enable dynamic quantization, resulting in significant performance gains and energy reduction without substantial accuracy loss.

While not directly addressing hardware faults, dynamic quantization techniques like DRQ are essential for maintaining deep learning performance in the presence of hardware limitations. By dynamically adjusting precision based on feature map dynamics, these techniques indirectly contribute to fault tolerance by ensuring accurate and efficient DNN computations despite potential hardware errors.

D. Bayesian Approach for Assessing Fault Tolerance

Banerjee et al. presented a Bayesian Deep Learning based Fault Injection (BDLFI) methodology for assessing the fault tolerance of neural networks, using Bayesian Deep Learning to model the propagation of faults and Markov Chain Monte Carlo inference to quantify the effect of faults on the outputs of a NN [23]. This advanced technique provides a novel approach to fault injection and assessment in deep learning systems, challenging pre-existing results in the field.

The Bayesian approach presented in this work offers a sophisticated method for evaluating the fault tolerance of deep neural networks, which is crucial for understanding the impact of hardware faults on deep learning computations. By leveraging Bayesian Deep Learning, this methodology provides valuable insights into the resilience of neural networks to hardware errors, contributing to the broader understanding of fault tolerance in deep learning systems.

In summary, the advanced techniques and frameworks discussed in this section offer valuable insights and methodologies for improving the fault tolerance of deep learning systems in the presence of hardware errors. These approaches provide practical strategies for mitigating the impact of hardware faults on deep neural networks, ultimately enhancing the resilience and reliability of deep learning computations. Additionally, they pave the way for further research and development in fault tolerance techniques for deep learning systems.

VII. Broader Perspectives on Reliability and Robustness in Deep Learning Systems

In the final section, we provide a broader perspective on the challenges and opportunities in building reliable deep learning systems. We will discuss the importance of testing methodologies, the impact of storage media errors, and the need for robust machine learning systems that encompass hardware fault tolerance.

A. State-of-the-Art in DNN Reliability and Hardware Error Resilience

The reliability of Deep Neural Network (DNN) algorithms and accelerators has become increasingly crucial, especially as they are being deployed in mission-critical applications [24]. Mittal et al. present a comprehensive survey of techniques for studying and optimizing the reliability of DNN accelerators and architectures, emphasizing the importance of designing for reliability as the first principle, rather than retrofitting for it. The paper underscores the significance of considering soft/hard errors arising due to process variation, voltage scaling, timing errors, and DRAM errors, highlighting the need for robust hardware error resilience in DNN systems.

Torres-Huitzil et al. provide a detailed review of fault tolerance in neural networks, focusing on passive techniques and briefly touching upon active fault tolerance methods [25]. The paper categorizes fault types, models, and measures used to evaluate performance, providing a taxonomy of the main techniques to enhance the intrinsic properties of neural models for fault tolerance. This review is valuable for understanding the principles and taxonomy of fault tolerance techniques in neural networks, which are directly relevant to hardware error fault tolerance in deep learning.

B. Specific Methods for Hardware Error Resilience

Ozen et al. introduce Sanity-Check, a method for enhancing the reliability of DNNs against hardware-level faults, using spatial and temporal checksums to protect fully-connected and convolutional layers [26]. The proposed method can be purely implemented in software and seamlessly integrated with modern DNN accelerators, delivering perfect error-caused misprediction coverage. This specific method of using checksums for fault tolerance in deep learning is essential for ensuring the reliability of DNNs, especially in safety-critical systems like autonomous driving.

Hari et al. focus on algorithmic error detection techniques for convolutions in Convolutional Neural Networks (CNNs), which are crucial for hardware fault tolerance in deep learning [27]. The paper explores low-cost, algorithmic error detection techniques for CNNs, demonstrating their ability to detect transient hardware errors while incurring low runtime overheads. These techniques are essential for ensuring the resilience of CNNs against hardware faults.

C. Impact of Storage Media Errors and Process Variation

Qin et al. study the trade-offs between storage/bandwidth and prediction accuracy of neural networks stored in noisy media, highlighting the vulnerability of more sophisticated models and datasets to errors in their trained parameters [28]. The proposed detection approach universally improves the robustness of deep neural networks, providing valuable insights into the impact of storage media errors on neural network robustness.

Ma et al. address the issue of process variation in CNN accelerators and propose mitigation techniques to ensure consistent performance, which is a crucial aspect of hardware

fault tolerance in deep learning systems [29]. The proposed sub-matrix reformation mechanism and weight transfer technique enable CNN accelerators to tolerate low-frequency PEs, achieving significant processing speed improvement with negligible accuracy loss. This work provides specific examples of fault tolerance techniques in the context of CNN accelerators affected by process variation.

D. Testing Methodologies and Robustness in Deep Learning Systems

Gerasimou et al. introduce a systematic testing methodology accompanied by an Importance-Driven (IDC) test adequacy criterion for DNN systems, emphasizing the importance of testing methodologies in ensuring the reliability of deep learning systems [30]. The IDC criterion enables the establishment of a layer-wise functional understanding of the importance of DNN system components, contributing to the assessment of the semantic diversity of a test set. This methodology is crucial for evaluating the dependability of DNN systems, especially in the context of hardware faults.

Zhang et al. discuss the broader topic of building robust machine learning systems, including the importance of robustness in machine learning systems, which encompasses hardware fault tolerance [31]. The paper provides context on the importance of robustness in machine learning systems, highlighting the need for reliable and fault-tolerant systems, especially in safety-critical applications.

E. Conclusion and Future Outlook

In conclusion, the surveyed literature provides valuable insights into the state-of-the-art in DNN reliability and hardware error resilience, specific methods for hardware error resilience, the impact of storage media errors and process variation, testing methodologies, and the broader context of building robust machine learning systems. Moving forward, future research should focus on integrating these diverse perspectives to develop comprehensive and robust fault tolerance strategies for deep learning systems, especially in safety-critical and mission-critical applications. Additionally, exploring the intersection of fault tolerance with energy efficiency and model protection in neural network accelerators presents an exciting avenue for further investigation.

REFERENCES

[1] U. Zahid, G. Gambardella, N. J. Fraser, M. Blott, and K. Vissers, "FAT: Training Neural Networks for Reliable Inference under Hardware Faults," *Proceedings - International Test Conference*, vol. 2020-Novem, 2020.

[2] Z. Liu, K. Jia, W. Liu, Q. Wei, F. Qiao, and H. Yang, "INA: Incremental network approximation algorithm for limited precision deep neural networks," *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, vol. 2019-Novem, 2019.

[3] C.-t. Chiu, K. Mehrotra, C. K. Mohan, and S. Ranka, "Training Techniques to Obtain Fault Tolerant Neural Networks," *Proceedings of IEEE 24th International Symposium on Fault-Tolerant Computing*, pp. 360—-369, 1994.

[4] J. Deng, Y. Rang, Z. Du, Y. Wang, H. Li, O. Temam, P. Ienne, D. Novo, X. Li, Y. Chen, and C. Wu, "Retraining-based timing error mitigation for hardware neural networks," *Proceedings -Design, Automation and Test in Europe, DATE*, vol. 2015-April, no. 2011, pp. 593–596, 2015.

[5] T. Tambe, E. Y. Yang, Z. Wan, Y. Deng, V. Janapa Reddi, A. Rush, D. Brooks, and G. Y. Wei, "Algorithm-hardware co-design of adaptive floating-point encodings for resilient deep learning inference," *Proceedings - Design Automation Conference*, vol. 2020-July, 2020.

[6] P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "GreenTPU: Predictive Design Paradigm for Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 7, pp. 1557–1566, 2020.

[7] Z. Song, Y. Sun, L. Chen, T. Li, N. Jing, X. Liang, and L. Jiang, "ITT-RNA: Imperfection Tolerable Training for RRAM-Crossbar-Based Deep Neural-Network Accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 1, pp. 129–142, 2021.

[8] C. Schorn, A. Guntoro, and G. Ascheid, "Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators," *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, vol. 2018-Janua, pp. 979–984, 2018.

[9] Y. Wang, "Adversarial Testing : A Novel On-line Testing Method for Deep Learning Processors," *VTS*, 2021.

[10] W. Li, Y. Wang, H. Li, and X. Li, "RRAMedy: Protecting ReRAM-based neural network from permanent and soft faults during its lifetime," in *Proceedings - 2019 IEEE International Conference on Computer Design, ICCD 2019*, 2019, pp. 91–99.

[11] N. Khoshavi, A. Roohi, C. Broyles, S. Sargolzaei, Y. Bi, and D. Z. Pan, "SHIELDeNN: Online accelerated framework for fault-tolerant deep neural network architectures," in *Proceedings - Design Automation Conference*, vol. 2020-July, 2020.

[12] Z. Xu and J. Abraham, "Safety design of a convolutional neural network accelerator with error localization and correction," in *Proceedings - International Test Conference*, vol. 2019-Novem, 2019, pp. 1–10.

[13] Y. He, P. Balaprakash, and Y. Li, "Fidelity: Efficient resilience analysis framework for deep learning accelerators," *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, vol. 2020-Octob, pp. 270–281, 2020.

[14] G. Gambardella, J. Kappauf, M. Blott, C. Doehring, M. Kumm, P. Zipf, and K. Vissers, "Efficient Error-Tolerant Quantized Neural Network Accelerators," *arXiv*, 2019.

[15] H. R. Mahdiani, S. M. Fakhraie, and C. Lucas, "Relaxed fault-tolerant hardware implementation of neural networks in the presence of multiple transient errors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1215–1228, 2012.

[16] W. Li, G. Ge, K. Guo, X. Chen, Q. Wei, Z. Gao, Y. Wang, and H. Yang, "Soft Error Mitigation for Deep Convolution Neural Network on FPGA Accelerators," *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020*, pp. 1–5, 2020.

[17] F. Libano, B. Wilson, J. Anderson, M. J. Wirthlin, C. Cazzaniga, C. Frost, and P. Rech, "Selective hardening for neural networks in FPGAs," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 216–222, 2019.

[18] A. P. Arechiga and A. J. Michaels, "The Robustness of Modern Deep Learning Architectures against Single Event Upset Errors," *2018 IEEE High Performance Extreme Computing Conference, HPEC 2018*, 2018.

[19] B. Salami, O. S. Unsal, and A. C. Kestelman, "On the Resilience of RTL NN Accelerators: Fault Characterization and Mitigation," *Proceedings - 2018 30th International Symposium on Computer Architecture and High Performance Computing, SBAC-PAD 2018*, pp. 322–329, 2019. [Online]. Available: http://arxiv.org/abs/1806.09679

[20] L. H. Hoang, M. A. Hanif, and M. Shafique, "FT-ClipAct: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," *arXiv*, pp. 1241–1246, 2019.

[21] W. Choi, D. Shin, J. Park, and S. Ghosh, "Sensitivity based error resilient techniques for energy efficient deep neural network accelerators," *Proceedings - Design Automation Conference*, 2019.

[22] Z. Song, B. Fu, F. Wu, Z. Jiang, L. Jiang, N. Jing, and X. Liang, "DRQ: Dynamic Region-based Quantization for Deep Neural Network Acceleration," *Proceedings - International Symposium on Computer Architecture*, vol. 2020-May, pp. 1010–1021, 2020.

[23] S. S. Banerjee, J. Cyriac, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, "Towards a Bayesian Approach for Assessing Fault Tolerance of Deep Neural Networks," *Proceedings - 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume, DSN-S 2019*, pp. 25–26, 2019.

[24] S. Mittal, "A survey on modeling and improving reliability of DNN algorithms and accelerators," *Journal of Systems Architecture*, vol. 104, no. August 2019, p. 101689, 2020. [Online]. Available: https://doi.org/10.1016/j.sysarc.2019.101689

[25] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," *IEEE Access*, vol. 5, pp. 17 322–17 341, 2017.

[26] E. Ozen and A. Orailoglu, "Sanity-Check: Boosting the Reliability of Safety-Critical Deep Neural Network Applications," *Proceedings of the Asian Test Symposium*, vol. 2019-Decem, pp. 7–12, 2019.

[27] S. K. S. Hari, M. B. Sullivan, T. Tsai, and S. W. Keckler, "Making Convolutions Resilient via Algorithm-Based Error Detection Techniques," pp. 1–12, 2020.

[28] M. Qin, C. Sun, and D. Vucinic, "Robustness of Neural Networks against Storage Media Errors," *arXiv*, 2017.

[29] M. Ma, J. Tan, X. Wei, and K. Yan, "Process variation mitigation on convolutional neural network accelerator architecture," *Proceedings - 2019 IEEE International Conference on Computer Design, ICCD 2019*, no. Iccd, pp. 47–55, 2019.

[30] S. Gerasimou, H. F. Eniser, A. Sen, and A. Cakan, "Importance-Driven Deep Learning System Testing," in *Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: Companion, ICSE-Companion 2020*, 2020, pp. 322–323.

[31] J. J. Zhang, K. Liu, F. Khalid, M. A. Hanif, S. Rehman, T. Theocharides, A. Artussi, M. Shafique, and S. Garg, "INVITED: Building robust machine learning systems: Current progress, research challenges, and opportunities," in *Proceedings - Design Automation Conference*, 2019.