Harnessing Approximate Computing for Machine Learning

Salar Shakibhamedan Amin Aminifar Luke Vassallo Nima TaheriNejad

***Abstract*—This paper explores the integration and application of Approximate Computing (AxC) approaches to Machine Learning (ML), especially Deep Learning (DL) models. We focus on four principal techniques—quantization, approximate multiplication, approximate in-memory computing, and input-dependent AxC . We demonstrate how each contributes to reducing the energy demands of current Artificial Intelligence (AI) systems, while maintaining acceptable levels of computational accuracy. These techniques may be deployed on software or hardware platforms. Quantization and input-dependent techniques can be implemented through software on general-purpose systems, enhancing flexibility and ease of deployment. Approximate multiplier and in-memory computing require specialized hardware integration, e.g., as custom System-on-Chip (SoC) or System-in-Package (SiP) solutions. We also discuss the crucial aspect of reliability, emphasizing robust design and error resilience to ensure the operational integrity of AI applications. By thoroughly examining these AxC techniques, the paper discusses an approach to designing energy-efficient and reliable AI accelerators, especially for SoC/SiP systems, providing essential support for use cases such as mobile and edge devices.**

***Index Terms*—Approximate Computing, Energy-Efficient Deep Learning, AI Accelerators, AI System Reliability, SoC/SiP**

I. Introduction

As the digital era continues to evolve, the integration of Artificial Intelligence (AI) in everyday applications has become increasingly common [1]. From smartphones that expect users' needs to smart cities that optimize traffic and energy use, the impact of advanced Machine Learning (ML) and Deep Learning (DL) technologies is profound. However, these innovations come at a significant cost of increased computational demand, leading to intensive energy consumption that is quickly becoming unsustainable [2].

The growth in data-driven technologies, particularly ML and DL, has led to an exponential increase in computational requirements [3]. Deep Neural Networks (DNNs) require extensive training with millions of data to accurately predict outcomes [4]. Once trained, deploying these models in real-world applications often necessitates continuous data processing and real-time decision-making, further increasing their energy consumption. For instance, applications such as autonomous vehicles and real-time health monitoring systems rely on the constant processing of vast amounts of sensor data to function effectively [5], [6]. This continuous demand for high computational power significantly strains energy resources, raising concerns about environmental impact and long-term sustainability of current computing practices [7].

The challenge is deepened by the physical limitations of existing computational architectures. Traditional computing systems are based on the von Neumann architecture, where significant energy is spent in the movement of data between the memory units (where data is stored) and the processors (where computation occurs) [8]. This architecture is increasingly becoming a bottleneck in systems where speed and efficiency are paramount. Moreover, as data volumes grow and the complexity of algorithms increases, the energy required for data movement can often exceed that used for actual computation, leading to inefficiencies that are both costly and environmentally detrimental [9].

In this context, Approximate Computing (AxC) emerges as a promising solution to mitigate the energy consumption issues inherent in traditional computing systems [10]. AxC strategically introduce approximation into the computation process, where exact precision is not crucial, thus reducing the computational burden [11]. This approach is particularly suited to applications in image and video processing, sensor data analysis, and large-scale simulations where approximate results may suffice [12]–[14]. By allowing for controlled inaccuracies, AxC systems consume less power and execute tasks faster than traditional precise computations [15], [16].

The significance of AxC is particularly relevant in the deployment of ML and DL models on mobile and embedded systems, such as smartphones, wearable technology, and Internet of Things (IoT) devices [17]. These devices, often constrained by battery life and processing power [18], [19], stand to benefit immensely from the reduced energy requirements of approximate computing. For instance, a smartphone using AxC techniques can perform tasks like voice recognition and image processing more power-efficiently, extending battery life while still delivering satisfactory performance [16], [20].

Moreover, AxC opens new avenues for the design of hardware and software that are inherently energy-efficient. This includes the development of specialized processing units that perform approximate calculations more efficiently [21], [22] and programming paradigms that prioritize energy efficiency [15], [16]. As the world moves towards more sustainable computing practices, the principles of AxC provide a crucial framework for future research and development [10].

In summary, AxC represents a paradigm shift in how we approach the problem of energy consumption in advanced computing systems. By embracing the trade-offs between accuracy and energy efficiency, AxC not only addresses the immediate challenges of power consumption but also sets the stage for the development of next-generation computing technologies that are both powerful and sustainable. This introductory overview sets the stage for a detailed discussion on various AxC strategies and their applications, as explored in the subsequent sections of this paper.

II. Methods

As computational demands continue to expand across various sectors, the exploration of efficient computing solutions has become a critical area of research. This section delves into the realm of AxC, highlighting innovative strategies and techniques designed to optimize the balance between computational accuracy and energy efficiency. The focus here is on four pivotal techniques that significantly reduce power consumption while maintaining sufficient performance for practical applications. These techniques include quantization, approximate multipliers, in-memory computing, and input-dependent AxC [5], each contributing uniquely towards achieving more sustainable computing practices.

These methods are not merely theoretical paradigms but are actively reshaping how computing tasks are approached in energy-constrained environments. By integrating these techniques and strategies, systems can achieve substantial reductions in power consumption—an essential advancement given the ever-increasing prevalence of mobile and embedded devices in our daily lives [23]. Furthermore, these techniques address critical issues inherent in traditional computing architectures, such as the high energy costs associated with data movement and the inefficiencies of general-purpose computing on large datasets [24].

The following subsections will explore each of these techniques in detail, discussing their principles, applications, and impact on ML and DL fields, where the need for efficient computing is particularly pronounced.

A. Quantization

Quantization is a pivotal technique in the field of ML and DL, especially when deploying models on resource-constrained devices [25]. By converting the high-precision floating-point numbers, typically used in model training, into lower-bit representations, quantization significantly reduces the memory footprint and bandwidth requirements [26]. This process not only accelerates inference times but also reduces the energy consumed per operation, making it particularly advantageous for embedded and mobile applications where power efficiency is crucial [27]. The effect of using quantization on memory storage of DNNs is illustrated in Table I.

TABLE I: The impact of quantization on DNNs.

DNN Model	Originial Model Size (MB)	Quantized Model Size (MB)
AlexNet [28]	240	6.9
SqueezeNet [28]	4.8	0.47
LeNet-300 [29]	1.07	0.027
LeNet-5 [29]	1.72	0.044
VGG16 [29]	552	11.3
ResNetV2-50 [30]	95.4	10.12
INCEPTIONV4 [30]	171.26	19.4
Tiny-YOLOV2 [30]	58.8	8.5
MobileNet-SSD [30]	63.28	30.48

The principle behind quantization lies in its ability to reduce the storage and computation of Neural Network (NN) parameters by mapping high dynamic value ranges into lower discrete ones, often at significantly reduced bit widths [31]. For example, moving from 32-bit floating-point to 8-bit integers can drastically cut down the amount of data that needs to be processed and stored, thereby minimizing the energy required for data retrieval and arithmetic computations [32]. This transition, however, must be managed carefully to ensure that the loss of precision does not negatively affect the overall accuracy of the model [30].

In the context of the ACE-CNN framework [33], quantization is utilized alongside approximate multipliers to further enhance energy efficiency. The ACE-CNN applies quantization at various stages of the Convolutional Neural Networks (CNNs), strategically reducing bit widths in a way that balances performance with computational efficiency. By integrating quantization with innovative approximate multipliers, the ACE-CNN approach demonstrates substantial power savings—up to 42% less energy consumption—while maintaining an acceptable accuracy level that is crucial for practical applications in fields like image and video processing.

B. Approximate Multipliers

Approximate multipliers are components designed to reduce computational complexity and energy consumption, yet this comes at the expense of introducing inaccuracies into multiplication operations [11], [21], [34]. These components are vital in applications where high throughput is more critical than absolute accuracy, such as multimedia processing and NN computation tasks [11], [35].

The concept of approximate multipliers revolves around the idea that many applications can tolerate some imprecision without significant degradation in final output quality [36]. By simplifying the multiplier design, either by skipping certain carry operations or truncating parts of binary numbers, these multipliers consume less energy and provide faster results than their precise counterparts [11]. This trade-off is particularly effective in DL where the inherent redundancy of NNs often masks the small errors introduced by approximation.

The ACE-CNN study introduces a novel signed approximate multiplier, the Signed Carry Disregard Multiplier (SCDM8), which optimizes energy consumption by selectively ignoring carry operations that have minimal impact on the overall computation accuracy [33]. This method showcases a significant reduction in power usage, with a slight drop in accuracy that is often negligible in real-world applications. The development and integration of such multipliers into computational architectures underscore the potential of approximate computing to transform energy efficiency in electronic systems, paving the way for more sustainable computing practices. The beneficial results of using SCDM8 approximate multiples are illustrated in Figure 1. As Shown, blue bars depict the approximate version accuracy to initial accuracy.

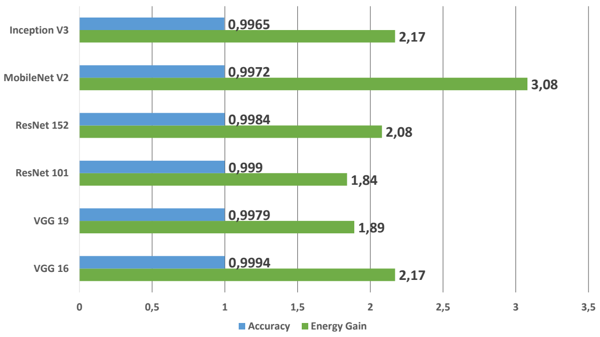

Fig. 1: Effects of SCDM8 on DNNs (Green: Energy Gain, Blue: Accuracy ratio) [33]

TABLE II: VGG16 energy usages and gains per inference [38].

	Energy (mJ/inference)	Relative Energy Gain
von Neumann system	16.77	1
3D-PIM	2.27	7.37

C. In-Memory Computing

In-memory computing has recently emerged as a powerful technique to counteract the energy inefficiencies caused by traditional data movement between processors and memory units [37]. This approach integrates processing capabilities directly within memory arrays, allowing data to be processed where it is stored. By minimizing the distance data travels, in-memory computing significantly reduces the energy and time overheads associated with data transfer, thus enhancing overall system performance.

The 3D-PIM technique represents a cutting-edge development in this area by utilizing a novel digital-to-time modulation approach within static random-access memory (SRAM) to perform multiplication and accumulation operations directly within the memory cells [38]. This technique eliminates the need for digital-to-analog converters (DACs), reducing the complexity and power consumption of the memory modules. By processing data directly at its storage location, 3D-PIM reduces latency and energy expenditure, offering substantial improvements over traditional computing paradigms.

In practical applications, the 3D-PIM method is especially beneficial for energy-constrained environments such as mobile devices and edge computing platforms, where power efficiency is paramount. The integration of processing within memory modules also opens up new possibilities for the development of more compact and efficient hardware designs, potentially transforming the landscape of computing technology. Table II displays the energy efficiency of 3D-PIM for a DL application (VGG16 [39] inference).

D. Input-Dependent Approximate Computing

Input-dependent AxC adjust the processing power and computations based on the relevance and complexity of the incoming data, optimizing resource allocation and energy consumption. This adaptive approach ensures that computational efforts are concentrated on data segments that require more intensive processing, while less critical data receives minimal attention. Such strategies are particularly useful in real-time systems and applications involving large datasets where processing all data at a uniform level of precision is inefficient and unnecessary.

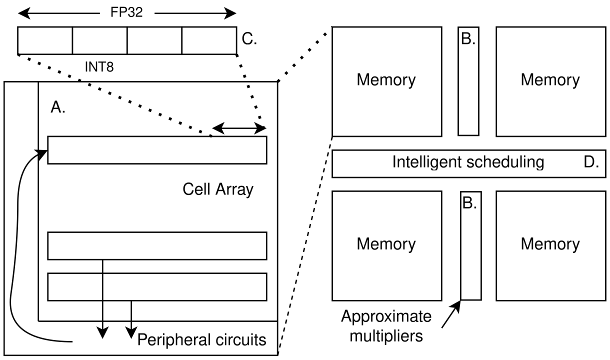

Fig. 2: Summary of approximate computing techniques

Recent studies adopted the input-dependent AxC approach in the development of ultra-low energy wearable devices for long-term monitoring of patients' vital signs and real-time event-driven classification prediction of medical conditions, aiming to prolong battery life while maintaining high prediction performance [40]–[45].

In another case, recent research [46], [47] proposes input-dependent AxC to enhance the reliability of wearable systems for monitoring physiological signals. Using machine learning algorithms, these studies identify and prioritize processing data segments that lack significant noise interference, directing computational resources to crucial segments for accurate health monitoring. This selective processing not only improves diagnostic accuracy but also reduces energy expenditure by avoiding unnecessary computations on noise-corrupted data.

Furthermore, research detailed in [45] introduces the Iterative CNN (ICNN), an innovative adaptation of traditional CNNs into a sequence of smaller, sequentially executed networks. This approach enhances classification accuracy by processing subsets of the input and features from prior networks, with the ability to terminate early once acceptable confidence is achieved. This design significantly reduces computational needs by curtailing operations when sufficient accuracy is reached, demonstrating that ICNN can achieve the efficiency of larger networks with fewer computational resources.

Figure 3 visually demonstrates the ICNN process, highlighting its sequential network execution and early termination capability once adequate classification confidence is achieved. This illustration emphasizes the ICNN's effectiveness in reducing computational load and energy use.

Overall, input-dependent computations represent a crucial strategy for managing the computational demands of modern systems, enabling more intelligent, efficient, and context-aware processing. By dynamically adjusting computational efforts based on data significance, these techniques can significantly enhance the performance and energy efficiency of a wide range of applications, from healthcare, computer vision, and natural language processing, to autonomous driving [48]–[50].

III. Discussion

In this section, we discuss the system integration and reliability of AxC techniques within AI accelerators and System-on-Chip (SoC) or System-in-Package (SiP) architectures. The focus is on how the efficient computing strategies discussed in section II: Quantization, Approximate Multipliers, In-Memory Computing, and Input-Dependent AxC, can be effectively integrated and implemented to enhance both the performance and reliability of AI systems.

Recent trends in the deployment of AxC techniques indicate a significant shift toward their implementation in specialized computing units. According to a survey [5], 57% of recent adaptive approximate techniques have been implemented on AI accelerators and non-conventional processing units such as SoCs, Tensor Processing Units (TPUs), and Field Programmable Gate Arrays (FPGAs). This trend underscores the growing recognition of the benefits these platforms offer in terms of processing power and energy efficiency. Figure 4 illustrates the distribution of these implementations, highlighting the predominant use of these technologies in cutting-edge AI systems. The integration of AxC techniques in these units also aligns with the strategies discussed later, focusing on how computing is implemented in various hardware configurations to improve both efficiency and reliability. This sets the stage for exploring specific hardware adaptations such as approximate multipliers and in-memory computing.

A. System Integration

The shift towards using SoCs, TPUs, and FPGAs, shown in Figure 4, reflects a broader industry movement towards optimizing AI applications for specific performance needs and energy efficiency. The integration of AxC techniques into AI systems involves a combination of software and hardware approaches, each contributing uniquely to the overall efficiency and functionality of the system. We discuss here, which methods are software or hardware compatible.

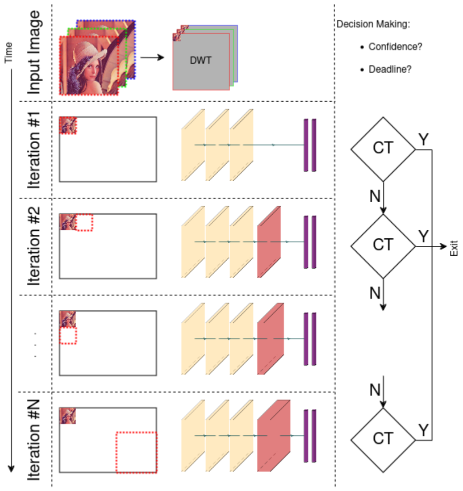

Fig. 3: ICNN iterative expansion of computations [45].

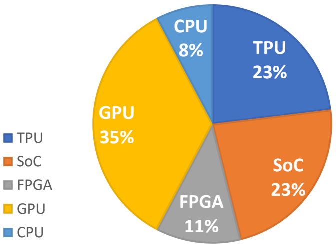

Fig. 4: Statistical distribution of operated hardware platform

Software Integration: Techniques like quantization and input-dependent AxC can often be implemented at the software level [5]. Software-based quantization involves modifying the data precision within the algorithms running on general-purpose computing systems. This allows for flexibility and ease of deployment across various platforms without the need for specialized hardware. Similarly, input-dependent computations can be programmed into the software to dynamically adjust the computational resources based on the significance or complexity of the input data, enhancing the system's efficiency and adaptive capabilities.

Hardware Integration: On the other hand, techniques such as approximate multipliers and in-memory computing require more specialized hardware designs. Approximate multipliers can be integrated into custom silicon or SoC designs to provide energy-efficient computation at the hardware level. These components are designed to perform specific tasks with reduced precision, which can significantly decrease power consumption and increase processing speed, making them ideal for integration into dedicated AI accelerators.

In-memory computing, involving techniques like the 3D-PIM approach, necessitates a rethinking of traditional memory architectures [38]. These strategies are incorporated directly into the memory hardware to perform computations where data is stored, drastically reducing the energy and time costs associated with data movement. Integrating these memory technologies into SoC or SiP can lead to significant improvements in the speed and energy efficiency of the system, particularly for applications requiring large-scale data processing like DNNs and real-time analytics.

Hybrid Approaches: Most advanced AI systems benefit from a hybrid approach that combines both software and hardware solutions [5], [51]. For example, an SoC for AI applications might use software-implemented quantization for flexibility and hardware-based in-memory computing for performance-critical tasks. This hybrid strategy ensures that the system is not only energy efficient but also maintains the versatility needed to handle a wide range of AI applications.

B. Reliability

Reliability in AI systems is crucial, particularly in applications where decisions must be made based on the computed results, such as autonomous driving or health diagnostics [5], [6]. In autonomous systems, the adaptive optimization of extra-functional properties, such as power efficiency or thermal management, leads to emergent behaviors that enhance system reliability without compromising functionality [52]. Therefore, integration of AxC techniques poses unique challenges and opportunities for enhancing the reliability of AI systems.

Quantization and Input-Dependent AxC : While these techniques can be implemented in software, ensuring their reliability involves rigorous testing and validation to confirm that the reduced precision or selective processing does not lead to unacceptable errors or biases in decision-making. Specific methodologies, such as cross-validation with different datasets and real-time monitoring of performance metrics, are critical in verifying the stability of these approaches under various operational conditions [53]. It is essential to develop robust frameworks for dynamically adjusting these parameters incorporating adaptive algorithms that can respond to data variability and system feedback without compromising the system's integrity [54]. Such frameworks should also include error detection and correction mechanisms that activate corrective measures automatically, maintaining the accuracy and reliability of outputs despite the inherent approximations [55].

Approximate Multipliers and In-Memory Computing: The hardware-specific essence of these strategies necessitates designing them with a focus on fault tolerance and error resilience. For approximate multipliers, it is vital to ensure that the imprecision introduced does not accumulate in a way that significantly degrades the overall system performance. Similarly, for in-memory computing, the integration of error-checking and correction mechanisms directly within the memory hardware can help maintain the reliability of the computations, even with the architectural changes that facilitate processing in memory.

AxC can potentially enhance the security and privacy of DNNs by altering their mathematical computational principles, which obscures precise computations and increases resistance to attacks [56]–[58]. This obfuscation protects sensitive data and reduces the efficacy of adversarial attacks aimed at exploiting exact outputs [56]. Furthermore, the variability introduced by approximation techniques as a defense mechanism improves data privacy by complicating reverse-engineering efforts [57]. Thus, employing AxC in AI models not only boosts efficiency but also strengthens their security and privacy measures [59].

Furthermore, reliability also relates to the physical and operational durability of hardware components. Custom SoC and SiP designed for AI must not only be energy efficient and functionally adaptive but also capable of operating reliably under vast environmental conditions and over extended periods.

IV. Conclusion

In conclusion, the integration of AxC techniques into AI accelerators and SoC/SiP systems necessitates a balanced approach that carefully considers both the computational efficiency and the reliability of the system. By effectively merging software flexibility with hardware performance, these strategies accommodate the dynamic demands of various applications, from regular data processing to critical real-time decision-making tasks.

Robust error management and system stability are crucial to ensure that these systems operate reliably under diverse conditions and maintain their integrity over time. Such robustness is essential for applications where failures could have significant consequences, such as in autonomous vehicles, healthcare monitoring, and other critical infrastructure.

By encompassing both the technological advancements in AxC and the practical considerations of implementing these techniques in real-world systems, it is possible to develop AI systems that are not only more energy-efficient but also robust and reliable enough for widespread deployment in critical applications. This approach not only contributes to the sustainability of computing resources but also ensures that the advancements in AI and ML continue to deliver tangible benefits across industries.

The discussion and insights presented in this paper underscore the potential of AxC to reshape the future of computing, providing a pathway towards more sustainable and efficient computational practices that do not compromise the quality or reliability of outcomes. As this field evolves, continued research and collaboration across the disciplines of hardware engineering, software development, and system design will be key to realizing the full potential of energy-efficient and reliable AI systems.

V. Acknowledgement

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 956090 (APROPOS: Approximate Computing for Power and Energy Optimisation, http://www.apropos.eu/).

References

[1] E. Baccour *et al.* Pervasive ai for iot applications: A survey on resource-efficient distributed artificial intelligence. *IEEE Communications Surveys & Tutorials*, 24(4):2366–2418, 2022.

[2] H. Liu *et al.* Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology*, 14(1):1–59, 2022.

[3] I. H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *Sn Computer Science*, 2, 2021.

[4] G. Amato *et al.* Large scale indexing and searching deep convolutional neural network features. In *Big Data Analytics and Knowledge Discovery: 18th International Conference, DaWaK 2016, Porto, Portugal, September 6-8, 2016, Proceedings 18*, pp. 213–224. Springer, 2016.

[5] S. Shakibhamedan *et al.* Ease: Energy optimization through adaptation–a review of runtime energy-aware approximate deep learning algorithms. *Authorea Preprints*, 2024.

[6] H. J. Damsgaard *et al.* Adaptive approximate computing in edge ai and iot applications: A review. *Journal of Systems Architecture*, 150:103114, 2024.

[7] L. Lannelongue *et al.* Ten simple rules to make your computing more environmentally sustainable, 2021.

[8] P. Kogge and J. Shalf. Exascale computing trends: Adjusting to the ”new normal”' for computer architecture. *Computing in Science Engineering*, 15(6):16–26, 2013.

[9] T.-J. Yang *et al.* A method to estimate the energy consumption of deep neural networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 1916–1920, 2017.

[10] A. Ometov and J. Nurmi. Towards approximate computing for achieving energy vs. accuracy trade-offs. In *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 632–635, 2022.

[11] N. Amirafshar *et al.* Carry disregard approximate multipliers. pp. 1–14, 2023.

[12] N. TaheriNejad and S. Shakibhamedan. Energy-aware adaptive approximate computing for deep learning applications. In *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 328–328, 2022.

[13] C. Ossimitz and N. TaheriNejad. A fast line segment detector using approximate computing. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2021.

[14] S. E. Fatemieh *et al.* Approximate in-memory computing using memristive imply logic and its application to image processing. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2022.

[15] A. S. Baroughi *et al.* Axe: An approximate-exact multi-processor system-on-chip platform. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pp. 60–66, 2022.

[16] S. Huemer *et al.* Approximation-aware task partitioning on an approximate-exact mpsoc (axe). In *IEEE Nordic Circuits and Systems Conference (NorCAS 2023)*, pp. 1–7, 2023.

[17] H. J. Damsgaard *et al.* Approximation opportunities in edge computing hardware: Anbsp;systematic literature review. *ACM Comput. Surv.*, 55(12), mar 2023.

[18] E. Shamsa *et al.* User-centric resource management for embedded multi-core processors. In *The 33rd International Conference on VLSI Design and The 19th International Conference on Embedded Design*, pp. 1–6, 2020.

[19] E. Shamsa *et al.* Ubar: User and battery aware resource management for smartphones. *ACM Transactions on Embedded Computing Systems (TECS)*, pp. 1–23, 2021.

[20] N. Chandolikar *et al.* Voice recognition: A comprehensive survey. In *2022 International Mobile and Embedded Technology Conference (MECON)*, pp. 45–51, 2022.

[21] N. Amirafshar *et al.* An approximate carry disregard multiplier with improved mean relative error distance and probability of correctness. In *Euromicro Conference on Digital Systems Design 2022 (DSD2022)*, pp. 1–7, 2022.

[22] S. E. Fatemieh *et al.* Fast and compact serial imply-based approximate full adders applied in image processing. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 13(1):175–188, 2023.

[23] M. Loukadakis *et al.* Accelerating deep neural networks on low power heterogeneous architectures. In *11th International Workshop on Programmability and Architectures for Heterogeneous Multicores (MULTIPROG-2018)*, 2018.

[24] C.-Y. Chen *et al.* Exploiting approximate computing for deep learning acceleration. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 821–826. IEEE, 2018.

[25] F. Daghero *et al.* Energy-efficient deep learning inference on edge devices. In *Advances in Computers*, volume 122, pp. 247–301. Elsevier, 2021.

[26] H. Wu *et al.* Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020.

[27] M. M. H. Shuvo *et al.* Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1):42–91, 2023.

[28] F. Iandola. *Exploring the design space of deep convolutional neural networks at large scale*. University of California, Berkeley, 2016.

[29] S. Han *et al.* Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[30] P. Nayak *et al.* Bit efficient quantization for deep neural networks. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pp. 52–56. IEEE, 2019.

[31] D. Becking *et al.* Ecq x: explainability-driven quantization for low-bit and sparse dnns. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 271–296. Springer, 2020.

[32] A. Demidovskij and E. Smirnov. Effective post-training quantization of neural networks for inference on low power neural accelerator. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

[33] S. Shakibhamedan *et al.* ACE-CNN: Approximate carry disregard multipliers for energy-efficient cnn-based image classification. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–14, 2024.

[34] Y. Wu *et al.* A survey on approximate multiplier designs for energy efficiency: From algorithms to circuits. *ACM Transactions on Design Automation of Electronic Systems*, 29(1):1–37, 2024.

[35] S. K B and R. Raj. *Approximate Multiplier for Power Efficient Multimedia Applications*, pp. 395–405. 02 2023.

[36] G. Zervakis *et al.* Approximate computing for ml: State-of-the-art, challenges and visions. In *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 189–196, 2021.

[37] N. Verma *et al.* In-memory computing: Advances and prospects. *IEEE Solid-State Circuits Magazine*, 11(3):43–55, 2019.

[38] S. Seyedfaraji *et al.* 3D-PIM: DAC-less, Digital-to-Time modulated, and Data-Aware in-SRAM MAC Accelerator. *IEEE Transactions on Computers*, pp. **, 2024.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] D. Sopic *et al.* Real-time classification technique for early detection and prevention of myocardial infarction on wearable devices. In *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4. IEEE, 2017.

[41] D. Sopic *et al.* Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems. *IEEE transactions on biomedical circuits and systems*, 12(5):982–992, 2018.

[42] F. Forooghifar *et al.* Self-aware anomaly-detection for epilepsy monitoring on low-power wearable electrocardiographic devices. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1–4. IEEE, 2021.

[43] H. A. Kholerdi *et al.* Enhancement of classification of small data sets using self-awareness — an iris flower case-study. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.

[44] A. Mozelli *et al.* A study on confidence: An unsupervised multiagent machine learning experiment. *IEEE Design Test*, 39(3):54–62, 2022.

[45] K. Neshatpour *et al.* Icnn: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 551–556, 2018.

[46] A. Aminifar *et al.* Recognoise: Machine-learning-based recognition of noisy segments in electrocardiogram signals. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2024.

[47] S. Khooyooz *et al.* A novel machine-learning-based noise detection method for photoplethysmography signals. In *46th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. IEEE, 2024.

[48] M. C. Florkow *et al.* Deep learning–based mr-to-ct synthesis: the influence of varying gradient echo–based mr images as input channels. *Magnetic resonance in medicine*, 83(4):1429–1441, 2020.

[49] M. Collier *et al.* Correlated input-dependent label noise in large-scale image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1551–1560, 2021.

[50] R. Zhu *et al.* Using deep learning based natural language processing techniques for clinical decision-making with ehrs. *Deep learning techniques for biomedical and health informatics*, pp. 257–295, 2020.

[51] H. Sharif *et al.* Approxtuner: a compiler and runtime system for adaptive approximations. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '21, pp. 262–277, New York, NY, USA, 2021. Association for Computing Machinery.

[52] N. TaheriNejad *et al.* Autonomous systems, trust, and guarantees. *IEEE Design Test*, 39(1):42–48, 2022.

[53] G. Macin *et al.* An accurate multiple sclerosis detection model based on exemplar multiple parameters local phase quantization: Exmplpq. *Applied Sciences*, 12(10):4920, 2022.

[54] Q. Huang *et al.* Codenet: Efficient deployment of input-adaptive object detection on embedded fpgas. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 206–216, 2021.

[55] T.-L. Nghiem *et al.* Applying bayesian inference in a hybrid cnn-lstm model for time-series prediction. In *2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6. IEEE, 2022.

[56] W. Liu *et al.* Approximate computing and its application to hardware security. *Cyber-Physical Systems Security*, pp. 43–67, 2018.

[57] A. Guesmi *et al.* Defensive approximation: securing cnns using approximate computing. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '21, pp. 990–1003, New York, NY, USA, 2021. Association for Computing Machinery.

[58] D. Le Quoc *et al.* Privacy Preserving Stream Analytics: The Marriage of Randomized Response and Approximate Computing. *arXiv e-prints*, pp. arXiv:1701.05403, January 2017.

[59] H. Li *et al.* Security enhancements for approximate machine learning. In *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, GLSVLSI '21, pp. 461–466, New York, NY, USA, 2021. Association for Computing Machinery.