Exploring Approximate Comparator Circuits on Power Efficient Design of Decision Trees

Pedro Aquino Silva, Mateus Grellert, Cristina Meinhardt
Departamento de Informática e Estatística - PPGCC
Universidade Federal de Santa Catarina (UFSC), Brazil
pedro.aquino@grad.ufsc.br, {mateus.grellert, cristina.meinhardt}@ufsc.br

***Abstract*—In recent years, Approximate Computing has been gaining space as a technique for tackling energy requirements in error-resilient applications, while the usage of Machine Learning systems has steadily increased. This work explores two different approaches for approximation in comparator circuits and their impact on Decision Tree applications, observing power and accuracy metrics. Two gate-level architectures are proposed for dedicated comparators, approximating 25% or 50% of least significant bits using different techniques. The circuits were described in 7 nm FinFET technology. The approximate comparators were then evaluated in a Decision Tree classification model using five continuous and mixed attribute datasets. The 25% LSB approximate comparator proposed improves the energy efficiency in Decision Tree applications, reducing from 12% up to 84% the power per inference while presenting minor deviations in accuracy compared to the exact baseline.**

***Index Terms*—approximate computing, decision trees, energy efficiency, comparators**

I. Introduction

Nowadays, there is an increasing demand for energy-efficient designs, as battery life has become a significant factor in embedded and Internet of Things (IoT) devices. Moreover, there is growing concern regarding energy savings of applications in accordance with Green Computing practices [1] [2]. In this context, dedicated hardware solutions are demanded, as they can be associated with architectures optimized for energy efficiency. With the massive amounts of data currently created, many Machine Learning (ML) applications are available and inserted in the our daily life. This also brings the necessity for energy efficient ML systems [3], recalling on the demand for dedicated hardware solutions. ML algorithms generally involve a large number of operations that are independent, which enlarges the design space to include hardware optimization. These applications are also very resilient to errors, making Approximate Computing (AxC) [4] [5] a suitable approach.

AxC is an emerging research area that exploits the fact that many applications have soft constraints in terms of accuracy, trading the exactness of operations for significant energy savings [4] [6]. AxC techniques have been explored in both hardware and software in different contexts, such as Internet of Things (IoT) devices, video and audio processing, Machine Learning and other error-tolerant environments [7] [8] [9].

Recent projects have looked into developing low-power approximate solutions for ML applications, focusing particularly on Neural Networks (NNs) [10] [11]. However, the usage of NNs may still be costly in energy-restricted environments, for example due to the large number of multiplication operations required. In this scenario, simpler and less costly learning models might be preferred over NNs. For instance, Decision Trees (DTs) can provide satisfactory and concise results for a large number of inference problems [5] [12].

During the classification stage of Decision Trees, one of the most frequently executed operations is the comparison. These operations are processed to determine the target path on the tree for the value under classification, guiding the tree traversal. Optimizations on delay and power of comparator circuits can significantly reduce the resource requirements on a decision tree synthesis. Thus, optimizations and improvements in comparator circuits are critical for tree-based models, including functional approximation.

While most studies evaluating the usage of AxC techniques in ML models investigate architectural and software approaches, such as [13] [14], there are also some works investigating AxC arithmetic blocks [15]. However, most of these works concentrate on the proposal and development of approximate adders and multipliers, with little to no literature investigating AxC approaches in the design of energy-efficient comparators. To the best of our knowledge, this is the first prospective study about the design of power efficient comparator circuits for decision trees.

This work investigates the design of approximated comparator circuits and their usage in Decision Trees classification models, focusing on the gains in power efficiency within acceptable accuracy constraints. Since many ML applications specifically require the inference (or classification) step to operate in low-power environments [5], we focused on the classification operation of pre-trained trees. The main contributions of this work are: 1) proposing two energy efficient approximate comparator dedicated circuits; 2) exploring approximation in classification with Decision Trees; and 3) introducing a workflow for analyzing the energy savings of AxC comparators in tree-based applications of continuous and mixed datasets.

The remainder of this paper is organized as follows: Section II introduces the approximate comparator circuits designed. The approximation workflow for decision trees classification are detailed in Section III. Section IV discusses the evaluation of the approximated comparators on decision trees. Finally,

978-1-6654-9005-4/22/$31.00 ©2022 IEEE

the main conclusions are presented in Section V.

II. Proposed Approximate Comparator Circuits

The comparison operation is traditionally implemented in two main forms: with dedicated circuits or with subtractors. Traditional dedicated comparators are designed to perform greater, equal or lesser-than computations for a given number of bits, and may be chained to accommodate desired bit widths for the inputs. These architectures are generally used for unsigned operands. Some designs for AxC comparator blocks provide support for larger bit width, however, they are typically implemented following a architectural approach targeting timing optimization [16].

We propose an exact comparator targeting reduction in area and, consequently, power consumption, named Exact Dedicated Comparator (EDC). This circuit was designed by reducing a CMOS unsigned comparator circuit to perform only the *lesser-or-equal than* function. We note that other comparison operations can be trivially implemented by switching the input order or inverting the output. In this circuit, the branches that computed equal and less-than were removed, and the greater-than branch was fed through an inverter.

The general equation for n-bits of the EDC is presented in Eq. 1, comprised of three parts. The first line performs an equality test for both inputs, producing each EQ_i; the terms $\overline{G_i}$ represent a larger than test in each bit, accounting for all equality tests in more significant bits; and the last line computes the operation $A \leq B$.

For the sake of compactness, a 4-bit version of this exact dedicated comparator is presented in Fig. 1. The diagram structure follows the same idea as the n-bit comparators. This circuit is taken as the starting point and baseline for the proposed dedicated approximated comparators.

$$\begin{aligned} EQ_i &= \overline{A_i \oplus B_i} && ,\ 0 < i < n \\ \overline{G_i} &= \overline{[\textstyle\prod_{k=i+1}^{n-1} EQ_k] \cdot (A_i \cdot \overline{B_i})} && ,\ 0 \leq i < n \\ A \leq B &= \textstyle\prod_{i=0}^{n-1} \overline{G_i} \end{aligned} \tag{1}$$

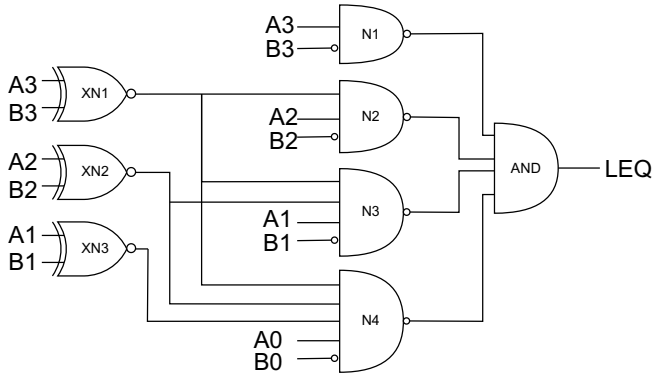


Fig. 1: 4-bit Exact Dedicated Comparator (EDC)

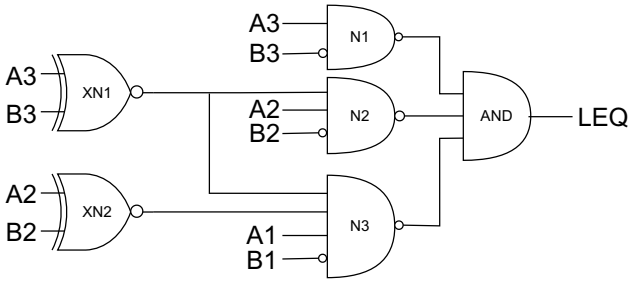
A. Dedicated Approximate Comparators

We propose two architectures for Approximate Dedicated Comparators (AxDCs): the AxDC1 and the AxDC2. The functional approximation technique involved removing logic gates from the EDC circuits while truncating the inputs. This approach is employed to improve power consumption while having little impact on the application accuracy. The for 4-bit versions of both comparators are shown in Fig. 2, as a simplification to introduce the idea, and to allow a direct comparison with the 4-bit EDC example.

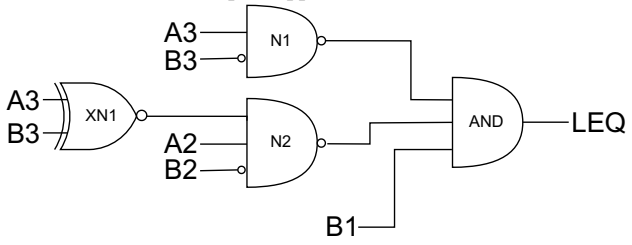
The first dedicated approximate comparator is the AxDC1. The general equation to n-bit is presented in Eq. 2, in which the difference from the Eq. 1 is that the range for the operations is reduced to $n/4$ of the n more significant bits (MSB). The approximation was achieved by truncation of the Least Significant Bits (LSBs). Thus, in the 4-bit example, the NAND4 and XNOR3 gates were removed from the EDC circuit.

$$\begin{aligned} EQ_i &= \overline{A_i \oplus B_i} && ,\ \tfrac{n}{4} < i < n \\ \overline{G_i} &= \overline{[\textstyle\prod_{k=i+1}^{n-1} EQ_k] \cdot (A_i \cdot \overline{B_i})} && ,\ \tfrac{n}{4} \leq i < n \\ A \leq B &= \textstyle\prod_{i=\frac{n}{4}}^{n-1} \overline{G_i} \end{aligned} \tag{2}$$

The AxDC2 implementation is shown in Eq. 3. This version limits the range of exact operations to $n/2$ of the n more significant bits (MSB), also truncating the $n/4$ LSBs. For the 4-bit version, the gates XNOR2 and NAND3 were also removed from the EDC circuit, and the bit 1 from the second input (B) was routed directly to the last AND gate. This resulted in an approximation of 50% of input bits and further reduction in area.



(a) 25% inputs approximation (AxDC1)



(b) 50% inputs approximation (AxDC2)

Fig. 2: Approximate Dedicated Comparators (AxDCs)

$$
\begin{aligned}
EQ_i &= \overline{A_i \oplus B_i} && , \ \tfrac{n}{2} < i < n \\
\overline{G_i} &= \overline{[\textstyle\prod_{k=i+1}^{n-1} EQ_k] \cdot (A_i \cdot \overline{B_i})} && , \ \tfrac{n}{2} \leq i < n \\
A \leq B &= \textstyle\prod_{i=\frac{n}{4}}^{\frac{n}{2}-1} B_i \ \cdot \ \prod_{i=\frac{n}{2}}^{n-1} \overline{G_i}
\end{aligned}
\tag{3}
$$

These approximations can be explored in n-bits circuits, which are constructed by obeying the same conditions of approximation shown in Eqs. 2 and 3. For these experiments, we employed 8-bit versions of the approximate circuits, analogous to those presented in the schematics for 4-bits. For example, the AxDC2 would ignore bits 0 and 1 from both inputs, and pass bits B2 and B3 to the last AND gate.

B. Subtraction-based Comparators

In this work, the proposed dedicated approximate comparators are confronted with comparators based on subtraction. These circuits use full adders (FAs) to perform subtraction and analyze a specific output for the operation required, for either unsigned or signed inputs. In the case of unsigned operands and the lesser-or-equal-than operation, we evaluate the carry-out of the most significant bit (MSB) full adder (FA) as the desired output. This approach also presents the benefit of easy scalability, achieved simply by increasing the size of the subtractor implemented.

For our case study, the subtraction-based comparators were designed with a simple Ripple Carry Adder (RCA). In these circuits, the approximation was achieved by substituting exact FA cells with three different approximate topologies: the Simplified Mirror Adder (SMA), the Approximate Mirror Adder 1 (AMA1), and the Approximate Mirror Adder 2 (AMA2) [17]. The circuits of these approximate FAs are presented in Fig. 3. These AxC FAs are explored due to the good outcomes of lower error distance and power savings compared to the Mirror and SERF Full Adders [18].

While dedicated circuits tend to be faster and more energy efficient, the downsides of subtraction-based comparators can be minimized using specialized low-power or faster architectures, as the RCA design used generates longer carry chains which result in increased delay.

III. Electrical Evaluation of the Comparators

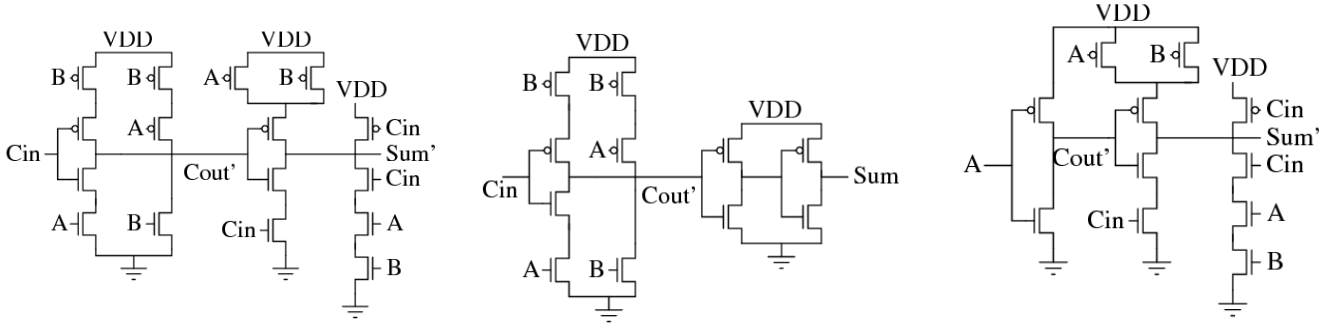
The electrical evaluation of the comparator circuits begins with the description of the Exact Dedicated Comparator and each proposed circuits for 8-bit extensions of the EDC and AxDCs, as well as the RCA with each approximated FA topologies for the subtractor-based comparators. We note that the proposed approaches can be easily adapted to larger inputs according to the range of the data in the datasets, but for our case study, 8-bit versions were justified by little impact in accuracy from quantization of the data.

The circuits were described adopting the ASAP7 7 nm FinFET PTM provided by Arizona State University (ASU) [19]. All devices use 3 fins, recommended as the minimum sizing for standard cell design [19]. Delay and power consumption were extracted for each circuit by simulation with Synopsys HSPICE. The circuits were electrically characterized under nominal voltage of 0.7 V, utilizing a 1 fF capacitor as load, equivalent to a fan-out of 4, to emulate a more realistic scenario.

Due to the large number of inputs of 8-bit comparators, the critical delay and power were analytically estimated, using the critical path and logic cell electrical characterization. Finally, the Total Error Distances (ED) and Error Rates (ER) were also calculated through the Truth Tables of the 8-bit approximations and considered in the analysis.

Table I presents the electrical characteristics of delay and power for the 8-bit comparator circuits evaluated in this work, as well as the ED and ER for each circuit. The first line describes the EDC. The next two entries present the results for the approximate dedicated versions (AxDC1 and AxDC2) proposed in this work. Finally, the remaining lines contain the results for the subtractor-based comparators with approximation (SMA, AMA1 and AMA2).

The AxDC1 shows the best results for error metrics among all the circuits evaluated, while presenting 24% of delay reduction and 11% of power savings. Higher improvements on power and delay are achieved with the version 2 of the proposed approximations, AxDC2, which reduces over 44%



(a) Simplified Mirror Adder (SMA) (b) Approximate Mirror Adder 1 (AMA2) (c) Approximate Mirror Adder 2 (AMA2)

Fig. 3: Approximate FAs inspired by the Exact Mirror Adder (EMA) [17] [18]

TABLE I: Electrical Characteristics and Error Metrics for the 8-bit Comparator Circuits

Circuit	Delay (ps)	Power (nW)	ED †	ER (%)
EDC	79.45	1040.73	0	0
AxDC1	60.25	916.21	384	0.59
AxDC2	44.05	720.75	24,384	37.21
SMA	104.56	4,319.44	10,795	16.47
AMA1	75.60	5,278.00	10,795	16.47
AMA2	69.76	3,346.48	16,384	25.00

† The maximum ED for 8 bit comparators is 65,536.

and 30% for delay and power respectively. However, this circuit has the largest ED due the 50% approximation of the inputs.

The Mirror Adder inspired approximations explored in the subtrator-based comparators show delay reduction for the AMA versions. However, all subtractor comparators evaluated had increased power consumption compared to the EDC, and presented ER superior to 16%. These results indicate that the dedicated approach is more advantageous for limited energy environments and the approximations proposed in this work can elevate the power efficiency, observing the quality restrictions of the target application, while subtractor-based comparators are more suitable for applications which already provide approximate adder/subtractor modules.

IV. Approximation Workflow for Decision Trees Classification

As a case study for this analysis, we used the General Public License (GPL) version of the C5.0 Decision Tree implementation [20]. C5.0 is an improved version of one of the most popular algorithms for training Decision Trees developed by Ross Quinlan, called C4.5. The C5.0 version contains improvements in terms of memory and computing resources, as well as extended features like boosting. For continuous attributes, C5.0 compares their value with a threshold that is tuned in the classification step. For categorical features, C5.0 maps their value as indices to an array of sub-trees (the branches of a node). Therefore, our approximate operators only work on continuous attributes. The comparison operations performed in the classification step were modified in the C5.0 source code in order to enable precise and approximate comparators.

This experiment consisted in implementing a method for evaluation of each architecture in the context of C5.0 classification. Our workflow, summarized in Fig. 4 was comprised of 4 stages: (1) pre-processing, (2) model training and classification, (3) power/energy estimation, and (4) comparison of results.

In this study, we employed 8-bit comparators, as they were large enough to maintain performance in accuracy, and provide good results in energy savings. Due to the fixed bit width in the datasets, a quantization pre-processing stage (1) was necessary. This step prepared the datasets for use as inputs of the unsigned 8-bit versions of the comparators proposed. The quantization consisted of two steps: a scaling step, fitting the continuous-attribute data in the range of 0 and $2^8 - 1$; then truncating the values to remove fractional information that went beyond the limits of representation.

The model training and classification stage (2) consisted of running classification for each tested dataset. The baseline results are generated training and testing the original dataset on the default C5.0. The approximate versions are trained with the default C5.0 version adopting the results from the pre-processing performed in the Stage 1. Thus, the test of the classification adopts the EDC and approximated versions of the comparator on the C5.0, emulating the results of using each approximation circuit in the classification step.

The approximate versions of C5.0 were trained on 5 continuous and mixed attribute datasets provided by the University of California Irvine (UCI) Machine Learning Repository [21]. The results were then analyzed using the unaltered C5.0 version as benchmark, focusing on the error rates (ER) on both training and test data, which correspond respectively to 80% and 20% of each complete dataset. The impact on the quality of the Decision Tree output was analyzed using the accuracy obtained on each approximation.

The third stage includes an electrical evaluation of all comparisons done on the approximate operations added to

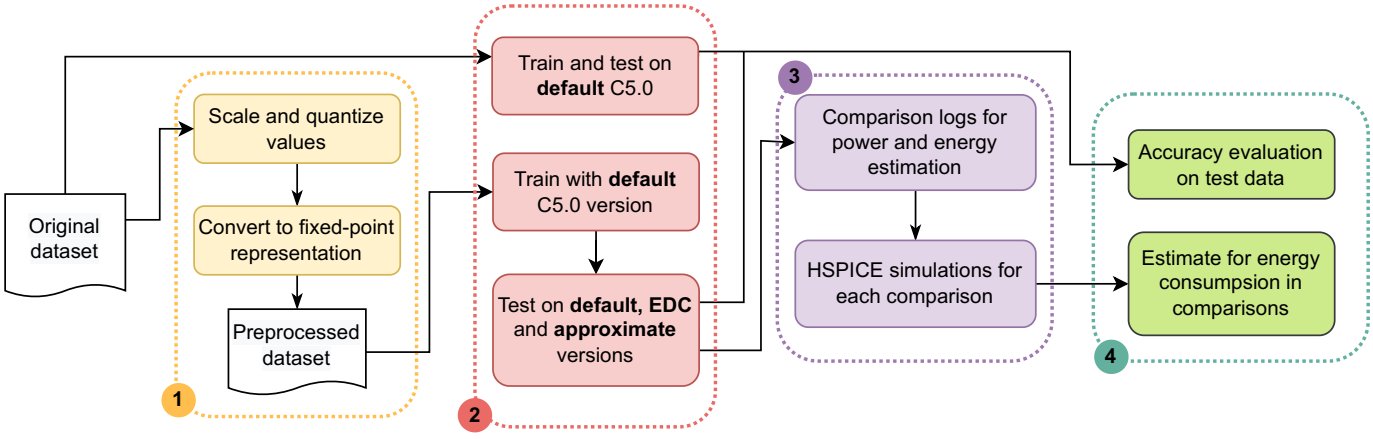


Fig. 4: AxC comparators evaluation in C5.0 workflow

the C5.0 source code in stage 2. A log of all continuous attribute test comparisons performed in the second stage is recorded. These logs saved during classification were used to generate sources for electrical simulation using the circuit descriptions generated for electrical characterization. All the operations were then simulated with HSPICE, with duration of 2.5 ns under nominal voltage (0.7 V). The energy data was collected from the simulations, which were summed to provide an estimate of the total energy consumption for each comparator in the entire test set classification.

The last stage (4) was the evaluation of all produced metrics, to provide clearer insight into the characteristics and behavior of each approximation in a given dataset. We analyze accuracy, total energy consumption for all comparison operations executed, and the number of operations performed. With these values, we can also calculate the average power consumption of the comparators per inference.

V. Evaluation on Decision Trees

Table II summarizes the results obtained for each dataset employed, presenting the number of test samples and attributes in each case. Furthermore, the table includes the accuracy (Acc) achieved in classification, the energy estimate calculated and the number of comparison operations (#Ops) done in continuous attributes. Note that a larger number of continuous attributes will incur in greater energy consumption.

In terms of prediction performance, we can observe that the adoption of approximated comparators on the classification of continuous datasets has varying effects on the accuracy compared to the exact version. The EDC presents a very similar behavior to the default C5.0 version, with an average deviation of only 0.01%, caused by the quantization error. All approximated comparators were thus analyzed against the EDC, to account for both accuracy and energy performance. For the approximated dedicated comparators, the AxDC1 showed the closest accuracy to the EDC, with a reduction on accuracy of only 0.12% on average for evaluated datasets. This is mostly due to the nearly negligible error rate of this comparator.

The AxDC2 reaches the higher energy reduction with more than 51% of savings compared to the EDC circuit for all datasets evaluated. The highest impact on energy is observed on the Iris dataset (86.64%) despite the fact that this approximate circuit increases the number of comparison operations executed in 81.8%. Despite reductions in resource consumption, the accuracy for this circuit was significantly worse than most other evaluated comparators, mostly due to its large error metrics.

The variable number of operations seen in Tab. II is explained by the different paths taken in the traversal of the trees when performing classification with each approximation. This variation is directly related to the energy consumption of each inference performed. In fact, the AxDC1 shows an average reduction of 21.2% in the number of comparisons, while the AxDC2 has an increase of 51.1%. This explains the similar energy reductions obtained in both AxDC1 and 2, of 51% and 46.3% respectively, especially relevant considering the much more accurate results of the AxDC1. Notably, this effect could be minimized by also applying the approximations the construction of the DTs, in which the threshold for continuous tests would be tuned for each different circuit.

Observing the experiment results by the metric of the power per inference, we highlight the fact that the power consumption calculated in this workflow is impacted directly by the number of continuous attributes which are present in the target dataset. For example, the Arrhythmia dataset shows the highest power per inference due to the presence of a total of 279 attributes, of which 206 were continuous. With the SMA-based comparator each inference in this dataset consumes 10.08 μW, while the adoption of the AxDC2 significantly reduces the power per inference to 1.77 μW. This difference reinforces how the approximate comparator proposed in this work can be applied in the DT classification, achieving high power efficiency with little to no interference on the accuracy of the application. We can also note that the AMA1-based comparator also proves to be a good choice, with an average reduction of 32% in power per inference when compared to the EDC. The AxDC1 provided reduction of 12.5% up to 84.95% in power per inference, with an average of 46.25% on all datasets.

VI. Conclusions

This work proposed two architectures for dedicated approximate comparators, evaluating their usage in the classification stage of Decision Trees. Along with this, we also evaluated another approach utilizing subtractors and approximated FAs. All circuits were described and studied in 8-bit versions, applied to the classification stage of the a Decision Tree Classifier. The proposed dedicated circuits circuits showed good overall results in electrical characteristics, with reduction of up to 31% in power consumption in relation to the exact baseline comparator. However, the AxDC2 provided significantly worse accuracy, compared to the remaining comparators. We also

TABLE II: Evaluation of the Comparison Circuits on Decision Trees Classification Model

	Heart disease Test samples = 60 Attributes = 13 (6 continuous)			**Iris** Test samples = 30 Attributes = 5 (4 continuous)			**Arrhythmia** Test samples = 90 Attributes = 279 (206 continuous)			**Adult** Test samples = 16281 Attributes = 15 (6 continuous)			**Forest fires** Test samples = 56 Attributes = 13 (10 continuous)		
Comparator circuit	Acc (%)	Energy (fJ)	# Ops	Acc (%)	Energy (fJ)	# Ops	Acc (%)	Energy (fJ)	# Ops	Acc (%)	Energy (pJ)	# Ops	Acc (%)	Energy (fJ)	# Ops
EDC	63.3	160.7	120	93.3	35.8	88	63.3	1,003.1	1,195	86.4	73.7	115,704	98.2	94.4	184
AxDC1	63.3	140.6	127	93.3	5.4	75	63.3	725.8	949	85.9	21.2	107,532	98.2	61.4	56
AxDC2	40.0	76.8	138	30.0	4.8	160	7.8	398.8	1,734	23.5	58.5	176,441	39.3	58.4	296
SMA	55.0	601.7	141	73.3	95.4	121	16.7	2,267.5	1,653	77.6	185.7	211,991	78.6	389.5	129
AMA1	55.0	174.2	21	73.3	16.1	30	16.7	334.2	456	77.6	29.5	96,287	78.6	106.1	102
AMA2	46.7	253.5	138	33.3	105.9	160	6.7	1,610.5	1,740	24.2	344.5	192,574	60.7	390.6	296

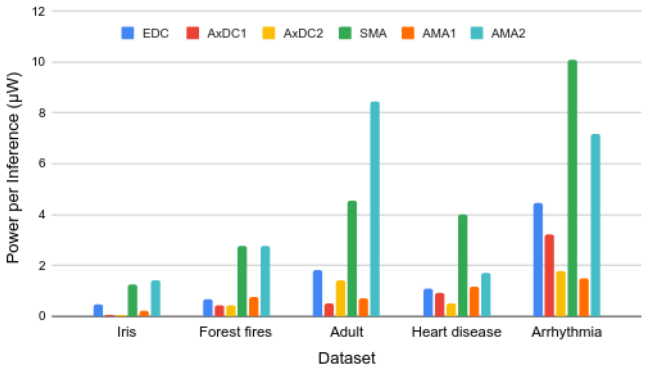


Fig. 5: Evaluation of the Power Consumption per Inference

investigated the effects of using subtractor-based approximated comparators, which provide greater design usability in applications where AxC arithmetic blocks are present, with the caveat of worse performance when compared to the dedicated approach. Overall, the AxDC1 showed the best results in error metrics, just of 0.59% error rate.

In general, the dedicated circuits show more advantages in accuracy and energy savings compared to the versions based on subtractors exploring approximations based on Mirror Adder. From the FA-based comparators, the AMA1 was especially competitive, providing gains of 32% in energy and at the cost of 27.7% in accuracy, by decreasing the number of operations by more than 54%, the largest reduction in number of comparisons between evaluated options. The advantage of this approach could be in an approximate environment, where there is the demand for an approximate adder/subtractor and the hardware can be reused. However, for dedicated hardware design, the AxDC1 was still considered the best option regarding the power-accuracy trade-off.

The proposed workflow has demonstrated to be able to deal with mixed datasets, and data larger than the 8-bit representation limits used. Next steps include to include approximation in the decision tree construction, and also combine other energy efficient approaches together with the proposed comparator circuits in a configurable environment.

Acknowledgment

This work was financed in part by National Council for Scientific and Technological Development – CNPq and the Propesq/UFSC.

References

[1] S. Singh, "Green computing strategies & challenges," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE, 10 2015, pp. 758–760. [Online]. Available: http://ieeexplore.ieee.org/document/7380564/

[2] H. B. Barua and K. Chandra Mondal, "Green data mining using approximate computing: An experimental analysis with rule mining," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018, pp. 115–120.

[3] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Research*, vol. 2, pp. 87–93, 9 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214579615000271

[4] J. Han, "Introduction to approximate computing," in *Proceedings of the IEEE VLSI Test Symposium*, vol. 2016-May. IEEE, apr 2016, pp. 1–1. [Online]. Available: http://ieeexplore.ieee.org/document/7477305/

[5] B. A. Abreu, M. Grellert, and S. Bampi, "Vlsi design of tree-based inference for low-power learning applications," in *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2020-Octob. IEEE, 10 2020.

[6] H. B. Barua and K. C. Mondal, "Approximate computing: A survey of recent trends—bringing greenness to computing and communication," *Journal of The Institution of Engineers (India): Series B*, vol. 100, pp. 619–626, 12 2019. [Online]. Available: http://link.springer.com/10.1007/s40031-019-00418-8

[7] T. Moreau, A. Sampson, and L. Ceze, "Approximate Computing: Making mobile systems more efficient," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 9–13, apr 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7093019/

[8] A. G. Strollo and D. Esposito, "Approximate computing in the nanoscale era," in *ICICDT 2018 - International Conference on IC Design and Technology, Proceedings*. IEEE, 6 2018, pp. 21–24. [Online]. Available: https://ieeexplore.ieee.org/document/8399746/

[9] D. Marwaha and A. Sharma, "A review on approximate computing and some of the associated techniques for energy reduction in iot," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. IEEE, 1 2018, pp. 319–323. [Online]. Available: https://ieeexplore.ieee.org/document/8399087/

[10] B. Zhang, A. Davoodi, and Y. H. Hu, "Exploring energy and accuracy tradeoff in structure simplification of trained deep neural networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, pp. 836–848, 12 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8354792/

[11] A. Goel, C. Tung, Y. H. Lu, and G. K. Thiruvathukal, "A Survey of Methods for Low-Power Deep Learning and Computer Vision," in *IEEE World Forum on Internet of Things, WF-IoT 2020 - Symposium Proceedings*, mar 2020. [Online]. Available: http://arxiv.org/abs/2003.11066

[12] Q. Li and A. Bermak, "A low-power hardware-friendly binary decision tree classifier for gas identification," *Journal of Low Power Electronics and Applications*, vol. 1, pp. 45–58, 3 2011. [Online]. Available: http://www.mdpi.com/2079-9268/1/1/45

[13] A. Kumar, S. Goyal, and M. Varma, "Resource-efficient machine learning in 2 kb ram for the internet of things," in *34th International Conference on Machine Learning, ICML 2017*, vol. 4. Springer New York, 2017, pp. 3062–3071. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7138-7

[14] E. García-Martín, N. Lavesson, H. Grahn, E. Casalicchio, and V. Boeva, "Energy-aware very fast decision tree," *International Journal of Data Science and Analytics*, vol. 11, pp. 105–126, 3 2021. [Online]. Available: https://link.springer.com/10.1007/s41060-021-00246-4

[15] M. Osta, A. Ibrahim, H. Chible, and M. Valle, "Approximate multipliers based on inexact adders for energy efficient data processing," in *2017 New Generation of CAS (NGCAS)*, Sep. 2017, pp. 125–128.

[16] Y. Zhou, J. Lin, J. Wang, and Z. Wang, "Approximate comparator: Design and analysis," in *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, vol. 2018-Octob. IEEE, 10 2018, pp. 129–133. [Online]. Available: https://ieeexplore.ieee.org/document/8598366/

[17] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: Imprecise adders for low-power approximate computing," in *Proceedings of the International Symposium on Low Power Electronics and Design*. IEEE, 8 2011, pp. 409–414. [Online]. Available: http://ieeexplore.ieee.org/document/5993675/

[18] P. A. Silva and C. Meinhardt, "Energy-efficient design of approximated full adders," in *ICECS 2020 - 27th IEEE International Conference on Electronics, Circuits and Systems, Proceedings*. IEEE, 11 2020.

[19] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "Asap7: A 7-nm finfet predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 7 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002626921630026X

[20] J. R. Quinlan, *C4.5 - Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[21] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml