IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 1, March 2021, pp. 131~138
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i1.pp131-138

Implementation of decision tree algorithm on FPGA devices

Kritika Malhotra, Amit Prakash Singh
Computer Science and Engineering Department, University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University, Delhi, India

Article Info

Article history:

Received May 24, 2020
Revised Jan 5, 2021
Accepted Jan 29, 2021

Keywords:

Classification
Decision tree classification
FPGA
Hardware
Machine learning

ABSTRACT

Machine learning techniques are rapidly emerging in large number of fields from robotics to computer vision to finance and biology. One important step of machine learning is classification which is the process of finding out to which category a new encountered observation belongs based on predefined categories. There are various existing solutions to classification and one of them is decision tree classification (DTC) which can achieve high accuracy while handling the large datasets. But DTC is computationally intensive algorithm and as the size of the dataset increases its running time also increases which could be from some hours to days even. But thanks to field programmable gate arrays (FPGA) which could be used for large datasets to achieve high performance implementation with low energy consumption. Along with FPGA's, python is used for accelerating the application development and python is leveraged by using python productivity for zynq (PYNQ), a python development environment for application development. This paper provides the literature review of an implementation of DTC for FPGA devices along with future work that can be done.

Corresponding Author:

Kritika Malhotra
Computer Science and Engineering Department
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
Dwarka, Delhi, India
Email: kritikamalhotra8.96@gmail.com

Amit Prakash Singh
Computer Science and Engineering Department
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
Dwarka, Delhi, India
Email: aps.ipu@gmail.com

1. INTRODUCTION

Machine learning is application of artificial intelligence (AI). Through machine learning, the system gets the ability of learning automatically and improving from experience without being explicitly programmed. To do so classification is an important step in machine learning.

In classification, it is found out to which category a new encountered observation belongs to on the basis of dataset containing observations. There are two datasets in classification - training dataset and testing dataset. The training set consists of example records whose category is known beforehand and the testing set is used to validate the model created using training set.

The objective of a classification algorithm is to build a model which can be used to assign unclassified records to one of the defined classes using the training dataset. Many different predictive models (classifiers), including the artificial neural networks (ANN) [1], decision trees (DT) [2] and support vector machines (SVMs) [3] have been proposed for classification.

One of the widely used classification techniques is a method called decision tree classification (DTC). Despite its simplicity in terms of implementation, it is computationally too exhaustive. The reason is that it strives to build a statistical model of the underlying operating environment (reality). Such a methodology describes a class of techniques called model-based methods. Model-based methods can find solutions with high statistical accuracy, yet at high computational price [4-7]. DTC involves two steps - the first step involves the construction of the decision tree model with the help of the example records. In the second step this constructed decision tree is applied to other new records to know about their class. Decision trees have various application such as email filtering, cells categorization (in biology), galaxies classification [8]. They give better accuracy even on large datasets when compared to other models of classification [9]. But due to large datasets the DTC and in fact other classification algorithms cannot stand up to the mark of computational power.

Acceleration of Hardware for classification algorithms is the best way to cope up with the problem of computational power. In this context, many different hardware platforms have been proposed. Graphical processing unit (GPU) is the most commonly used solution. But field programmable arrays (FPGA) has significantly shown better performance than GPU in many applications such as image processing, pattern recognisition, digital signal processing, and others. In literature, many performance comparisons for GPU and FPGAs have been done and FPGAs have performed better in most of the cases. Although, GPU have lower cost and shorter development time but FPGA are superior in terms of power consumption [10]. Also, with complicated algorithms GPU cannot provide performance as needed due to their memory architecture that caused limitations to memory access [11].

Thus, for achieving improved throughput, reduced latency, and low energy consumption (the three most important constraints for computational systems) even better than GPU, field programmable gate arrays, FPGA's are one of the popular, powerful, and parallel processing reprogrammable devices for implementing hardware in machine learning or deep learning algorithms to achieve high computing power in low cost and energy. The reprogrammable property of FPGA makes it more flexible and adaptable to changes. FPGA's are used to achieve high performance, increased security and reduced latency.

For application development for FPGA devices various high-level languages like C/C++ can be used but to speed up the application development work to its best, python is used which is further improved by usage of python productivity for zynq (PYNQ). By combining the use of python, its tools and libraries, PYNQ, provides a platform to developers for application development for FPGA devices [12].

There exist many machine learning algorithms which can be implemented using FPGA devices for hardware acceleration. But this paper reviews the existing literature for implementation of decision tree classification for FPGA devices.

This paper is divided as follows. In section 2, machine learning is explained, in section 3, the decision tree classification (DTC) algorithm is explained, in section 4 FPGA devices are explained in detail, section 5 presents about the literature survey that has been done on FPGA implementation of DTC and finally the paper ends with last two sections having conclusions and references.

2. MACHINE LEARNING

Machine learning is one of the subset of artificial intelligence. Through machine learning, the machine can learn by gathering experience by doing a certain task and improve its performance by doing the similar tasks in future. When it gets difficult to handle data analytically, then machine learning proves out to be the best solution for those applications [13].

The machine learning process can be divided into two steps - training and inference [13]. In training the data is observed using the dataset provided. The result of training process is the trained network. Then the second step called as inference is performed. In inference, the trained network obtained is applied on the new data [13] to perform tasks like image recognisition, counting and tracking people within the room, speech recognisition, and others. In Figure 1 two main steps of Inference are shown: feature extraction and classification [14].

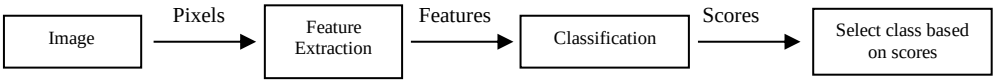

Figure 1. Inference process [8]

2.1. Feature extraction

Large datasets have large number of variables and thus large number of resources is required to process them. In feature extraction, to make large, initial, raw dataset more manageable for processing, dimensionality of dataset is reduced. So, feature extraction is a process in which variables are combined and selected into features which lead to creation of a new, smaller dataset that has reduced amount of data. This dataset has reduced data but it still accurately and completely describe the original dataset.

2.2. Classification

Classification is a process of assigning a category to test dataset based on category that is derived using training data set [15]. Classification has several applications over a wide variety of industries. For e.g. classification can be used for email filtering, speech recognisition, handwriting recognisition, biometric identification, document classification and much more. Figure 2 explains the process of classification that is the main steps involved in classification. There are various algorithms such as support vector machines [16], linear discriminant analysis, decision tree, naives bayes, k-nearest neighbor, and others that can be used for classification purposes but the algorithm studied in this paper is decision tree classification.

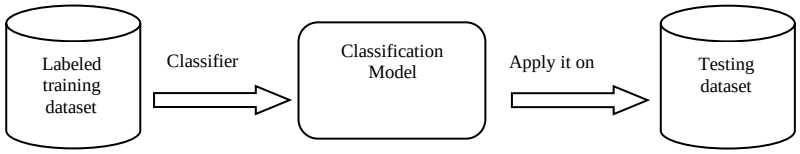

Figure 2. Classification

3. DECISION TREE CLASSIFICATION ALGORITHM

As stated above classification is defined as a process in which a dataset is given which is splitted into two parts - training dataset and testing dataset. The training dataset has several records with each record having unique record id. There are several fields in record known as attributes. There are two types of attributes - continuous and categorical. The attributes having continuous domain are the continuous attributes and the attributes which have finite set of discrete values are the discrete attributes. The attributes used for classification are categorical attribute. In DTC a model is built that allows prediction of class of a record in terms of its remaining attributes [8].

As understood by the name in decision tree learning a model is built in the form of tree structure [17]. Thus a decision tree model consists of internal nodes and leaves. There is a splitting decision and a splitting attribute associated with each internal node. The leaves have a class label assigned to them which represents a possible value for output variable. A decision tree is represented as follows in Figure 3.

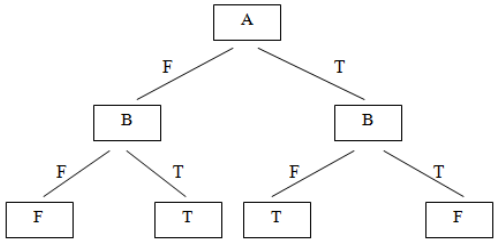

Figure 3. Decision tree

Each internal node/decision node (represented by boxes) tests an attribute represented as A/B. Each branch has an attribute value represented as T/F. Each leaf node/terminal node assigns a classification. The

first node of the tree is called as "root node." The nodes connected through root node are called as Branch nodes and the nodes at last level at the end of tree are called as Leaf nodes [17].

A decision tree model can be built from training dataset by following an approach of recursive partitioning [17]. Two phases are there to build the decision tree model. The process starts from the root node and a path to a leaf is traced by using the splitting decision at each internal node. In first phase, a splitting attribute and a split index are chosen. The second phase involves splitting the records among the child nodes base on the decision made in the first phase. This process is recursively continued until a stopping criterion is met. The usual stopping criteria are [17]:

- All or most of the data at a particular node have the same class.
- All attributes have been used up in the partitioning.
- The tree has grown to a pre-defined limit.

Now at this point, the decision tree can be used to predict the class of an incoming record, whose class ID is unknown.

In decision Tree the major challenge is attribute selection [8]. Attribute selection is a procedure to determine the splitting criterion that best partitions the data attributes into individual classes. We have two popular attribute selection measures:

- Information gain
- Gini index

When a node in a decision tree is used to partition the training instances into smaller subsets the entropy (entropy is the measure of uncertainty of a random variable) changes. Information gain is measure of this change in entropy [17]. This criterion will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e., the attribute with a high value is placed at the root. The formula of information gain is:

$$Information\ Gain\ (T, X) = Entropy\ (T) - Entropy\ (T, X) \quad (1)$$

$$E\ (T) = \sum - p_i \log_2 p_i \quad (2)$$

Where i = 1 to c

$$E\ (T, X) = \sum P(c)\ E(c) \quad (3)$$

Where c Є X, **T** → Current state, and P_i → Probability of an event i of state T or Percentage of class i in a node of state T, and X → Selected attribute.
In a much simpler way, it can be concluded that:

$$Information\ Gain = Entropy\ (before) - \sum Entropy\ (j, after) \quad (4)$$

Where j=1 to K, "before" is the dataset before the split, K is the number of subsets generated by the split, (j, after) is subset j after the split.
Gini Index is metric to measure how often and randomly chosen element would be incorrectly identified [8]. It can be understood as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. The formula of gini index is:

$$Gini = 1 - \sum (p_i)^2 \quad (5)$$

Where i = 1 to c

3.1. Algorithm

Input - Training dataset, test dataset
Output - Decision tree
Steps -
Do for all attributes
Calculate the Entropy E_i of the attribute F_i to calculate the value of Information gain or gini index.
End do
Split the dataset into subsets using the attribute.
Draw a decision tree node containing this best attribute and split the dataset into subsets.
Repeat the above step until the following stop criteria is met:

All attributes have been used up in partitioning.
The three has grown to a pre-defined length.

3.2. Strengths

The strengths of decision tree methods are [17-18]:

- Decision tree are competent to generate understandable rules.
- Decision tree are able to handle both continuous and categorical variables.
- Decision tree provide a clear indication of which fields are most important for prediction or classification.
- Help determine worst, best, and expected values for diverse scenarios.
- Decision trees are trouble-free to use and explain.

3.3. Challenges

The challenges in implementing decision tree classification in embedded system are energy consumption and speed as it can be seen from the description of DTC that DTC is highly computationally intensive algorithm and therefore it gets highly expensive and complex to train it when there are many class labels. Thus to accelerate the performance, to reduce the energy consumption and cost and to get better throughput, FPGAs are used about which it is explained in next section.

4. FIELD PROGRAMMABLE GATE ARRAYS

Field programmable gate arrays (FPGA) is a semi-conductor integrated circuit (IC) designed in such a way that user can configure it after manufacturing [19]. FPGA vendors such as Xilinx provide not just the physical circuits but also the development tools that can be used to develop design for FPGA and ultimately program them. They even provide security solutions such as encryption and authentication to secure the FPGAs.

FPGAs are off the shelf programmable devices that have no processor to run the software [20]. They can be configured as simple as an AND gate and as complex as multi core processor. For implementing custom hardware functionality, they provide a flexible platform at low development costs. The FPGA architecture consists of three main components [20]:

- Programmable logic blocks - The programmable logic block provide basic computation and storage elements used in digital systems.
- Programmable routing (interconnects) - The programmable routing establishes a connection between logic blocks and Input/output blocks to complete a user-defined design unit.
- I/O blocks - The programmable I/O pads are used to interface the logic blocks and routing architecture to the external components.

Along with these three components FPGAs have a set of embedded components such as digital signal processing blocks to perform arithmetic intensive operations such as multiply and accumulate, block RAM, look up tables, flip flops, and others [20]. Modern FPGAs also contain specialized memory, arithmetic and communication blocks through which digital systems can be implemented efficiently [19].

FPGAs are used to accelerate performance of hardware for computation intensive application such as computer vision, communications, industrial embedded systems, IoT and many more as they provide flexibility, scalability, and parallelism. Through FPGAs performance can be maximized per watt of power consumption, thus reducing costs for large scale operations [20]. To make FPGAs more user friendly python is used along. Python along with Python productivity for ZYNQ (PYNQ) reduces the application development time by providing various packages, libraries, and tools. PYNQ is explained in [21] and can be read from there as it is out of the scope of this paper.

Thus, concluding the benefits of FPGAs, FPGAs provide flexibility due to their reconfigurability and reprogrammability feature; they provide acceleration to hardware by sharing the computations with processor, and they are secured [22]. Therefore, they can be used along with DTC for better throughput and to accelerate the performance of DTC in low power consumption and cost.

5. LITERATURE SURVEY

Existing literature for hardware implementation of decision tree algorithm on FPGA was reviewed and various publications were uncovered.

J.R. Struharik [2] presented four different architectures for hardware implementation of decision trees. One of them was single module per level (SMPL) and SMPL-P architecture and the other one was universal node (UN) and UN-P architecture. The SMPL architecture was a pipelined architecture with

number of pipeline stages equal to the depth of decision tree. Each stage had three major modules: attribute memory, decision tree node, and the memory that stored information about the nodes at same DT level. The parallelized version of the SMPL architecture which had more complex hardware was known as SMPL-P architecture. The UN architecture comprised of four modules out which three modules had same function as SMPL architecture with two differences and the fourth module added was a control unit that was responsible for correct operation of the whole system. The parallelized version of UN architecture with more number of hardware parts was named as UN-P architecture. The four architectures were applied on 23 different datasets with target device Virtex5 family FPGA and Xilinx ISE Foundation 12.1.03i software and the results calculated in the research showed that the four architectures differ from each other in terms of hardware complexity, throughput, or classification speed and speedup. SMPL-P proved out to be best in terms of throughput, speedup, and hardware demand. SMPL and UN-P architectures gave comparable results and the UN architecture gave worst results.

Narayanan *et al.* [8] have presented decision tree classification using FPGA for binary classification. Compute intensive kernel called the gini score computation was implemented in the learning process of the decision tree classification and a highly efficient, reconfigurable architecture was developed to implement gini score calculation. The architecture was further optimized by using bitmapped data structure and by reordering the calculations to minimize the bandwidth requirements of DTC. The proposed design was implemented on Xilinx vertex -II Pro FPGA platform with 100MHz clock rate. Fixed point integer computations were used to perform the calculations. With maximum of 16 gini units working in parallel, the algorithm was run on PowerPC CPU (which was embedded in FPGA device) and the results were obtained which showed that the designed architecture achieved 5.58x speedup in comparison to software implementation executed on the same platform.

Saqib *et al.* [15] designed a pipelined architecture to implement parallel decision tree binary classification to improve the execution time of the algorithm and thus accelerate the hardware in minimal resource utilization and low power consumption by processing the data in pipelined fashion. The proposed architecture was highly scalable and was implemented on Digilent Nexys2 Spartan 3E FPGA board with 100MHz clock rate. The design had a streaming architecture that used double-buffered input and output memories to simultaneously receive and process the data. The proposed system was configured with high-speed communication unit that enabled data processing and data transfer faster. The results obtained showed that as the number of clock cycles that were required to process the data and generate results reduced, the throughput increases. The designed system was 3.5 times faster than existing hardware implementations and its performance was linearly dependent on number of records in a dataset.

M. Barbareschi *et al.* [23] presented a special Von Neumann architecture called tree visiting processing unit (TVPU) to implement a decision tree classifier. For better performance pipelining for TVPU is done and a separate branch predictor unit is used to leverage the pipeline. TVPU was implemented as a co-process on XilinxVirtex XC5VLX110T FPGA and its performance was measured in terms of classification and misprediction delay. The results showed that lower the computations delay faster is the classification and higher the misprediction delay better the performance.

Kulaga *et al.* [24] have proposed FPGA implementation of decision tree and its ensemble for letter and digit recognisition using Vivado High Level Synthesis tool. It is explained about Vivado HLS that it is one of the tool available for Xilinx FPGAs and Zynq SoC devices for synthesisation. Verilog and VHDL can be synthesized from C, C++ code. The function used for synthesisation consists of a pipelined loop that iterates over the lines and columns of a 1920 x 1080 grayscale image and had and initiation interval of four cycles. Four pixels were processed in each iteration. OpenCv was used to implement both the decision tree and its ensemble. The proposed architecture used Floating Point arithmetic for computations and had a clock frequency of 100MHz and 66.67MHz. Two different datasets was used at both training and testing stages and several optimizations tree code and node layout in memory were made. The three parameters classification accuracy, throughput, and resource usage for different tree depths, ensemble sizes, and algorithms was obtained for both shared attribute memory and separate attribute memory and module's functioning was verified for correctness using C/RTL co simulations and Zynq-7000 SoC devices. The results showed that when wide sample input and fixed point arithmetic is used highest classification throughput and resource utilization is obtained.

Amato *et al.* [25] proposed architecture for FPGA based decision tree classifier to speed up the classifier. The system consisted of three modules namely - transformation and integration module, semantic classifier module and post reasoner module and the data were collected from sensors nodes of intrusion detection system. The transformation module was used to gather and integrate the data. The semantic classifier implemented the rule based classifier. The post reasoner modules analyzed the causes and reasons of the events and then refine the procedures of classification. The proposed architecture had decision boxes and a Boolean Net. The decision boxes had a pipelined architecture with depth 2 that compute all the

decision in parallel while the Boolean net assigned the class to given input. The output from the two is VHDL code produced by PMML2VHDL tool used for FPGA synthesis. The results obtained showed that the decision box created 560 millions of decisions per second and delay time of 10ms for Boolean net which resulted in great performance.

M. Barbareschi *et al.* [26] illustrated the FPGA accelerator architecture design for decision tree classifier and its ensemble using majority voting. The architecture is based on pipelined technique and the decision tree design was followed as in Amato *et al.* The proposed architecture provides 114 times speedup and require only 0.03% energy compared with software approach. To improve the system performance, further, they have designed a multiple classifier system where the decision tree is used as base classifier.

In Another paper M. Barbareschi [27] proposed a hardware implementation of decision tree using Xilinx Virtex-5 XC5VLX110T FPGA device to determine the scalability in terms of required resources. The proposed architecture as compared to other memory-based architectures having static parameters didn't required additional memory banks follows pipelined mechanism and has dynamic parameters. The results showed that the delay and latency decrease with the number of pipeline stages. Also the clock frequency linearly decreases with the number of nodes, and power consumption increases with the area overhead. Further it was concluded that the number of gates needed to synthesize the decision tree algorithms is proportional to the product of number of internal node and number of features.

Another pipelined architecture for implementation of decision tree on FPGA was proposed by D. Tong *et al.* in [28]. They have used C4.5 algorithm of decision tree for classification purposes and have proposed two architectures to store the classifier - one with distributed RAM and other with block RAM. For acceleration of both the architectures FPGA was used. To measure the performance throughput and resource efficiency was observed and compared. The distributed RAM architecture achieved a clock rate of 180MHz, throughput of 460Gbps and resource efficiency of 26 slices/Gbps, while the block RAM architecture showed clock rate of 334MHz, throughput of 6Gbps, and resource efficiency of 21 slices/Gbps for 1024 nodes of tree. As the number of nodes was decreased the throughput and clock rate increased and resource efficiency decreased. Table 1 shows the comparison of all the above architectures in terms of throughput and clock rate.

Table 1. Comparison of surveys

SNo.	Paper Ref. No.	Max. Clock Rate	Throughput (Average)
1.	[2]	100MHz	SMPL-P >SMPL=UN-P > UN
2.	[8]	100MHz	3.24Gbps
3.	[15]	100MHz	96.26%
4.	[23]	360MHz	26.91Gbps
5.	[24]	100MHz	1.3508MSa/s
6.	[25]	(not specified)	562113546,9 (float/s)
7.	[26]	(not specified)	112.8796MS/s
8.	[27]	450MHz	Explained in graphs given in paper
9.	[28]	180MHz & 334MHz	460Gbps&6Gbps

6. CONCLUSIONS

With machine learning algorithms, FPGA platform can be used as an accelerator to accelerate the speed and improve the performance of the algorithms. The flexibility and reprogrammability of FPGA allowed creation of architecture that gives optimal results. All the techniques proposed make use of parallel capability of the devices to design parallel architectures for efficient results and efficient memory utilization. It is even concluded that all the architectures implemented for FPGA are pipelined that allowed faster execution of algorithm and faster clock rate. Generally floating-point arithmetic is used in most of the architectures but its use is always tried to be avoided and fixed point arithmetic is forced to be used as floating point are more complex and utilize more resources which decreased the use of parallelism. The results from the literature review also showed some issues that prevent the use of FPGA widely. First issue seen is lack of development tools that are mature as it is very difficult to develop hardware accelerators without proper knowledge. Second issue is cost of advanced FPGA devices. The cost of advanced FPGA devices used for machine learning algorithms is quite high. Although these two issues currently prevents the use of FPGA devices, but all of the above issues the available high level languages and synthesis tools designing of FPGA have become much easier. With advancements in FPGA technology, soon FPGA platform is expected to be the most used platform for computations as well as machine learning algorithms due to its feature of low resource utilization, flexibility, and high performance. For the future work we will use FPGA in implementation of random forest and compare the results with decision tree implementation to choose the best one out.

REFERENCES

[1] C. Bishop, "Neural networks for pattern recognisition," Clarendon Press, Oxford University Press, 1995.

[2] J.R. Struharik, "Implementing Decision Trees in Hardware," *Proc. IEEE 9thInt'l Symp. Intelligent Systems and Informatics (SISY)*, pp. 41-46, Sept. 8-10, 2011, doi: 10.1109/SISY.2011.6034358.

[3] V. Vapnik, "Statistical learning theory," New York, Wiley, 1998.

[4] Hamid O.H., Braun J., "Reinforcement Learning and Attractor Neural Network Models of Associative Learning," *In: Joint Conference on Computational Intelligence. IJCCI 2017*, vol 829, pp 327-349, 2019.

[5] Daw, N. D., Niv, Y., & Dayan, P., "Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control," *Nature neuroscience*, vol. 8, no. 12, pp. 1704-1711, 2005, doi: 10.1038/nn1560.

[6] O. H. Hamid, F. H. Alaiwy and I. O. Hussien, "Uncovering cognitive influences on individualized learning using a hidden Markov models framework," *2015 Global Summit on Computer & Information Technology (GSCIT)*, Sousse, pp. 1-6, 2015, doi: 10.1109/GSCIT.2015.7353337.

[7] Glüge, S., Hamid, O.H., Braun, J. and Wendemuth, A., "May. A markov model of conditional associative learning in a cognitive behavioural scenario," *In International Work-Conference on the Interplay between Natural and Artificial Computation*, Springer, Berlin, Heidelberg, pp. 10-19, 2011.

[8] R. Narayanan, D. Honbo, G. Memik, A. Choudhary and J. Zambreno, "An FPGA Implementation of Decision Tree Classification," *2007 Design, Automation & Test in Europe Conference & Exhibition*, Nice, 2007, pp. 1-6. doi: 10.1109/DATE.2007.364589.

[9] J. Catlett, "Megainduction: Machine learning on very large databases," *Ph.D Thesis*, University of Sydney, 1991.

[10] S.M. Afifi, H. GholamHosseini, R. Sinha, Hardware Implementations of SVM on FPGA: A State-of-the-Art Review of Current Practice, *Int. J. Innov. Sci. Eng. Technol (IJISET)*, vol. 2, no. 11, pp. 733-752, 2015.

[11] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance Comparison of FPGA, GPU, and CPU in Image Processing," *in International Conference on Field Programmable Logic and Applications*, pp. 126-131, 2009, doi: 10.1109/FPL.2009.5272532.

[12] A. G. Schmidt, G. Weisz, M. French, "Evaluating Rapid Application Development with Python for Heterogeneous Processor based FPGAs*," Int. Symp. on Field-Programmable Custom Computing Machine*, pp. 121-124, 2017, doi: 10.1109/FCCM.2017.45.

[13] Mark Nadeski, "Bringing machine learning to embedded systems," Texas Instruments, pp. 1-5, 2019.

[14] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, Z. Zhang, "Hardware for machine learning: Challenges and opportunities", *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 1-8, 2017, doi: 10.1109/CICC.2017.7993626.

[15] F. Saqib, A. Dutta, J. Plusquellic, P. Ortiz and M. S. Pattichis, "Pipelined Decision Tree Classification Accelerator Implementation in FPGA (DT-CAIF)," in *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 280-285, Jan. 2015, doi: 10.1109/TC.2013.204.

[16] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods," *Cambridge university press*, pp. 165-172, 2000, DOI:10.1017/CBO9780511801389.012.

[17] S. Dutt, S.Chandramouli, A.K. Das, "Machine Learning," Pearson, 2019.

[18] Mahdieh Najafy, "What are pros and cons of decision tree versus other classifier as KNN SVM NN?," 2012, Available: https://www.researchgate.net/post/What-are-pros-and-cons-of-decision-tree-versus-other-classifier-as-KNN-SVM-NN.

[19] P. Škoda, B. Medved Rogina and V. Sruk, "FPGA implementations of data mining algorithms," *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, pp. 362-367, 2012.

[20] A. Shawahna, S. M. Sait and A. El-Maleh, "FPGA-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823-7859, 2019, doi: 10.1109/ACCESS.2018.2890150.

[21] PYNQ, "PYNQ: Python productivity," Available: http://www.pynq.io/.

[22] L. Stornaiuolo, M. Santambrogio and D. Sciuto, "On How to Efficiently Implement Deep Learning Algorithms on PYNQ Platform," *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Hong Kong, pp. 587-590, 2018, doi: 10.1109/ISVLSI.2018.00112.

[23] M. Barbareschi, E. Battista, N. Mazzocca, and S. Venkatesan, "A hardware accelerator for data classification within the sensing infras-tructure," in Information Reuse and Integration (IRI), *2014 IEEE 15thInternational Conference on. IEEE*, pp. 400-405, 2014, doi: 10.1109/IRI.2014.7051917.

[24] R. Kulaga and M. Gorgon, "FPGA implementation of decision trees and tree ensembles for character recognition in VIVADO HLS," *Image Processing and Communication*, vol. 19, no. 2-3, pp. 71-82, 2015, DOI: 10.1515/ipc-2015-0012.

[25] F. Amato, M. Barbareschi, V. Casola, & A. Mazzeo, "An FPGA-Based Smart Classifier for Decision Support Systems," *Intelligent Distributed Computing VII*, pp. 289-299, 2014, DOI: 10.1007/978-3-319-01571-2_34.

[26] M. Barbareschi, S. Del Prete, F. Gargiulo, A. Mazzeo, and C. Sansone, "Decision tree-based multiple classifier systems: An fpga perspective," in *Conference: Multiple Classifier Systems-12th International Workshop, MCS 2015*, pp. 194-205, 2015, DOI: 10.1007/978-3-319-20248-8.

[27] M. Barbareschi, "Implementing Hardware Decision Tree Prediction: A Scalable Approach," *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Crans-Montana, pp. 87-92, 2016, doi: 10.1109/WAINA.2016.171.

[28] D. Tong, L. Sun, K. Matam, and V. Prasanna, "High throughput and programmable online traffic classifier on FPGA," *In Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 255-264, 2013.