Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 1, July 2026, pp. 78~92
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp78-92

FPGA based forest fires prediction system

Faroudja Abid[1], Nouma Izeboudjen[1], Fatiha Louiz[2]
[1]Division de Microélectronique et Nanotechnologie, Centre de Développement des Technologies Avancées, Algiers, Algeria
[2]DAS, Centre de Développement des Technologies Avancées, Algiers, Algeria

Article Info

Article history:

Received Jan 6, 2026
Revised Jun 15, 2026
Accepted Jun 27, 2026

Keywords:

Decision tree
Fire detection
Fire prediction
Machine learning
Smart sensor node
System on chip

ABSTRACT

This paper describes the idea of designing and implementing a field-programmable gate array (FPGA) based system on chip (SoC) for forest fires prediction (FFP). The FFP in the proposed system is based on the decision tree (DT) algorithm implemented as an intellectual property (IP) in the FFP Zynq-SoC architecture that constitutes the processing part of a smart sensor node. This latter processes the collected meteorological data and takes decision locally at sensor node level; without having to send massive data to the base station for decision. The performance of the decision-tree software classifier in terms of accuracy and recall are about 75% and 0.88, respectively. The hardware implementation results of the DT -based forest-fires prediction SoC show that the developed DT IP core is area and power-efficient.

Corresponding Author:

Faroudja Abid
Division de Microélectronique et Nanotechnologie, Centre de Développement des Technologies Avancées
Cité 20 août 1956 Baba Hassan, Algiers, Algeria
Email: fabid@cdta.dz

1. INTRODUCTION

Algeria is among countries affected by the forest fire disaster, mainly in summer [1]. Firefighting strategy and forest fires surveillance systems are mainly based on traditional methods such as the monitoring tower. The drawbacks of the conventional systems are inaccuracy of fire danger estimation and localization. Forest fires prediction (FPP) and detection are critical applications that require real-time processing and decision speed up. A system that operates automatically constitutes a solution for early forest-fires prediction and detection. Several technologies and systems have been proposed to detect fire, e.g., systems employing charge-coupled device cameras and infrared detectors, satellite systems and wireless sensor networks (WSNs). The introduction of communication technologies such as WSN [2]-[7] in fire monitoring systems is a promising direction that can be exploited for FPP. In fact, the approach that integrates the artificial intelligence to the WSNs gains interest in fire-detection automation field. In this context, we propose the integration of the decision tree (DT) hardware classifier as a part of a sensor node architecture to allow FPP locally in automated way. The binary tree is considered since the fire occurrence classification is two classes.

We discuss in this work the field-programmable gate array (FPGA)-based system on chip (SoC) for forest-fires prediction. The basic unit of the proposed FPP SoC architecture is the DT IP core (DT_IP) developed using the high level synthesis (HLS) approach. The proposed solution requires some processing power since the data processing and decision are performed locally at sensor node level. From this point of view, FPGA is a good alternative for implementing our model. Additionally the classification using FPGA is performed directly (there is no need to decode or fetch instructions), which reduce processing time and power consumption. The hardware implementation of the DT classifier can provides a required real-time processing while minimizing power consumption as well as decision time. The FFP SoC constitutes the processing part

of a smart sensor node. In addition to the meteorological data (temperature, relative humidity (RH), and wind speed (WS)) sensing by means of a combination of several sensors; the sensor node will be able to predict fire occurrence locally using the DT classifier, without sending the data to the sink for decision. In contrast to software-based fire detection systems [2], [3], [8], we propose a hardware implementation of the DT classifier based fire-occurrence prediction model. It is noteworthy to specify that this paper concerns the DT_IP core and the Zynq-SoC based FFP SoC (the processing part of the sensor node) design; the whole smart sensor node architecture (i.e., communication and sensing units) is not in the scope of this paper.

The main contribution of this work is the DT based forest-fires prediction model elaboration, which is further designed as a hardware accelerator and integrated as a reusable IP. In fact, to the best of the author's knowledge and according to the conducted related work investigation [9], there is no similar work at hardware level targeting fire prediction and detection applications, the available existing DT based forest-fire detection are software-based. We propose a hardware solution where the decision is taken locally at sensor node level, thus there is no need to send the collected data to the base station for decision. The aim being the speed up of the fire prediction process and the reduction of the communication activity and energy consumption; compared to the software-based solution.

This paper is organized as follows: In section 2, we review related work to the decision-tree hardware implementations beyond the targeted application. The adopted method is exposed in section 3. The proposed forest fire-prediction software-model developement is presented in section 4; we also expose in this section the performance evaluation of the DT based model. Section 5 deals with the FFP system design and section 6 exposes its hardware implementation. The implementation results analysis and discussion are presented in section 7. Finally, section 8 concludes this paper.

2. RELATED WORK

This section undertakes a review of existing works related to the decision-tree hardware implementations to situate this work in respect to the literature, beyond the application (i.e., not chiefly targeting fire prediction and detection issues). The DT-learning algorithm is widely applied for forest fire detection and prediction purpose; due to its simplicity and ease readability compared to other machine learning (ML) algorithms. The hardware acceleration of the DT classifier and its ensembles on FPGA platforms has been investigated in several studies [10]-[22]. The hardware implementation was performed for both training and classification (inference) processes to improve the classification speed by exploiting the parallelism of the FPGA platforms.

For instance, an optimized hardware implementation of the DT classifier on FPGA device was proposed in [10]. The authors exposed an application that automatically produces the very high-speed integrated circuit hardware description language (VHDL) code describing the predictor for FPGA synthesis. The konstanz information miner (KNIME) data-mining framework was used for the model (predictor) generation in the learner definition phase. Where a PMML file is generated, and then converted to VHDL model using the PMML2VHDL tool. The generated model includes two components, namely the decision nodes VHDL for decisions parallel computation and the Boolean Net VHDL for effective classification computation. The targeted FPGA was the Virtex-5 and sensors data includes temperature, humidity, pressure, battery level and GPS position. Sayed *et al.* [11], a framework to automate the electronic design and hardware implementation of ML based classifier models was proposed. The platform allows the low-cost, high-performance integrated circuit (IC) based classifier design and training. The proposed framework generates the tree-based classifier in two formats, namely extensible markup language (XML the hierarchy of the generated tree) and Verilog (the HDL code of the trained model). The generated verilog file is used in the application-specific integrated circuit (ASIC) synthesis and implementation for a customized classifier IC. The proposed design operates at 100 MHz, and achieved 80.76% accuracy using five different datasets. The reported results in terms of average total dynamic power and chip-area were about 0.22 mW and 5189.83µm2, respectively. Narayanan *et al.* [12] the authors developed a hardware architecture using the bitmapped data structure for the split attribute and split index calculation in order to reduce the running time in the Gini-score computation process and thus the decision-tree induction. According to this study, the Xilinx Virtex-II Pro FPGA platform-based implementation allows a speed-up factor up to 5.58 compared to software-based implementation. The work exposed in [13] discussed a binary axis-parallel DT and the principal component analysis (PCA) based gas identification system. The hardware implementation was performed on both FPGA and ASIC, the data acquisition was performed using a 16-array sensor. A detection rate about 91.36% was obtained with the proposed hardware axis-parallel DT.

A co-design approach has been considered in various works for the DT and its ensembles hardware design. For example, a fall detection system (FDS) based on the Zynq-SoC platform was exposed in [14]. The authors used a DT as falls classifier, combined with the PCA technique for dimensionality reduction.

The two algorithms were implemented in the programmable logic (PL) part of the Zynq-SoC platform. A discrete wavelet transform (DWT) was integrated in the processing system (PS) for preprocessing the data from the Shimmer sensors. The ARM processor in the PS part performed the decision on the data classified by the DT. An accuracy about 88.4% was reported. In another study, Ali *et al.* [15] proposed a Zynq-SoC platform based electronic nose (EN) implementation. The EN is based on two algorithms, namely the PCA for dimensionality reduction and the DT for gas identification. The system achieved accuracies that range from 90% to 100%. Saqib *et al.* [16] the authors investigated the hardware implementation of the axis-parallel binary DT classifier. They proposed a pipelined hardware accelerator architecture using double-buffered input and output (implemented using the Xilinx Logicore IP block memory generator). The MicroBlaze processor was used to supply and fetch data. The correctly classified percentage varied from 83.3% to 100% using the Iris and contact lenses datasets. The whole design used 62% of the FPGA resources. According to the authors, the architecture is 3.5 times faster than other hardware architectures.

Regarding the DT ensembles hardware implementations [17]-[22], the author in [17] developed and evaluated area-efficient hardware accelerator architectures for the DT ensembles classifiers, namely axis-parallel, oblique and non-linear DT ensembles classifier systems. The model's evaluation was performed using the UCI datasets. Both the MicroBlaze and ARM based systems were used for the software-models comparison against the hardware proposed accelerators. The same author exposed in [18] the hardware implementation of a pipelined evaluator architecture, which is composed of identical DT node evaluator modules organized in pipeline chain. A hardware implementation of the random forest (RF) algorithm that includes the k-means clustering was proposed in [19]. The k-mean clustering was included to reduce the hardware resources utilization by sharing the comparators of the branch nodes on the DT. The authors stated that compared to the CPU the FPGA realization is 8.4 times faster, and 7.8 times better regarding the power-consumption efficiency. Compared to the GPU the FPGA implementation is 62.8 times faster, and 385.9 times better regarding the power-consumption efficiency.

A comparison of FPGA-based RF hardware implementation against the GPU based implementation was also investigated in [20]. Essen *et al.* [20] considered the hardware acceleration models generated using the compact random forest (CRF) ML classifier and evaluated using three approaches. The authors discussed the effectiveness of three platforms: a multi-core chip multiprocessor (CMP), FPGA, and a GP-GPU. They evaluated performance in terms of power and cost-efficiency on both FPGA and GP-GPU solutions. According to this study, the CRF accelerates the classification process with an ensemble of DTs on FPGA, which pipelined the processing of each DT. Besides, the GP-GPU achieved a massive thread-level parallelism.

Hardware architecture of the ML algorithm for sensor data filtering and analysis was discussed in [21]. The authors made a comparison of the RF algorithm implementation in both FPGAs and CPUs. They targeted the First G-APD Cherenkov Telescope (FACT) application for the theoretical model evaluation on the Zedboard. They obtained 80% prediction accuracy for gamma-hadron separation using RF model of 50 trees. According to this study, the FPGAs are clearly the best approach in terms of energy consumption per element with acceptable throughput and represent a good alternative for implementing DT based models for applications with an emphasis on power consumption.

The hardware DT ensembles was also investigated in [22]; the authors proposed a hardware multiple classifiers system implementation. They presented an architecture based on the majority-voting rule as a combiner. They evaluated mainly the bagging DT, boosting DT and RF. The hardware version of the DT classifier introduced in [10] was used in this study. The model was evaluated on the Xilinx Virtex 5 FPGA platform targeting two applications, namely spam detection and IP traffic classification. According to this study the multiple classifiers system outperformed the single DT classifier for the spam detection application. Whereas a single classifier solution was most suitable for the IP traffic application classification (a multi-class problem where latency need to be minimized, while preserving high accuracy).

In this paper, we propose a hardware implementation of the DT classifier for forest-fire prediction purpose. Compared to works that predict forest fires, we propose a hardware FPGA-based DT classifier's accelerator for speed up the decision process by predicting and giving the decision locally at sensor node level. Hence, there is no need to communicate the collected data to the base station for decision as for the software-based solutions.

3. METHOD

Figure 1 illustrates the main phases of the adopted methodology for the hardware implementation of the DT classifier on FPGA platform using the Vivado HLS approach. The methodology involves the following steps: (i) the development of the DT-based forest fire prediction software model. The DT based predictive model is trained off-chip using MATLAB, prior to its hardware implementation. The k-fold cross-

validation method was applied for the DT classifier performance evaluation. (ii) The trained DT is coded in C/C++ as rule based model; the rules were established from the DT structure generated after the off-chip training. The C source code 'DT.c' of the fire prediction model describes, mainly the function called "fire_predict". Additional C source code (DT_test.c) is written for the HLS co-simulation phase. (iii) The DT-based fire prediction model IP development is performed using the high-level modeling approach. The HLS approach transforms a macro-architecture into micro-architecture with gain in the design development and verification time. A co-simulation is performed in both C and register transfer level (RTL), i.e., the same testbench used for checking the C code is also used for the automatic RTL simulation model generation. The RTL corresponding to the 'DT.c' is generated automatically after synthesis. (iv) To interconnect the DT_IP to other IPs included in the system, we have added an Advanced eXtensible Interface (AXI) to the developed HLS DT_IP core.

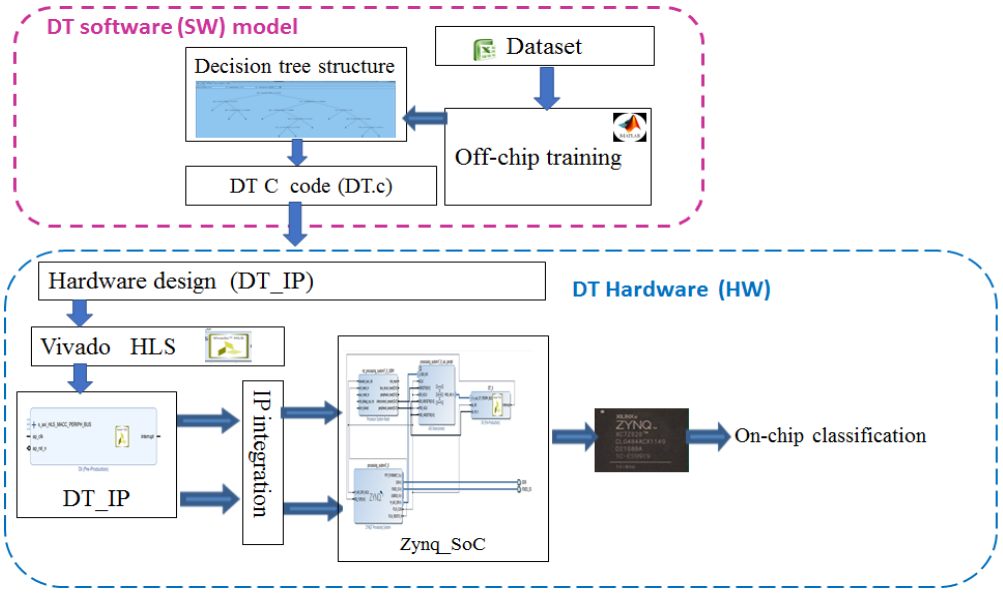

Figure 1. Methodology steps

4. PROPOSED FIRE PREDICTION MODEL SOFTWARE DEVELOPMENT

4.1. Dataset

We have used the Algerian forest fires dataset that regroups data of two regions of Algeria, namely the Sidi Bel-Abbas region located in the northwest of Algeria and the Bejaia region located in the northeast of Algeria. According to the statistics [1] these two regions are among the most affected regions by the forest fires disaster in Algeria, this is why we have considered in our study these two main fire hotspots. The constructed dataset includes 244 instances. The dataset is available in the UCI repository at: https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+dataset.

4.2. Software DT-based prediction model performance evaluation

Table 1 summarizes the performance evaluation results of the proposed DT based forest-fires prediction model in terms of accuracy, precision, recall and area under RoC curve (AUC). The evaluation is done for three. In view of the size of our dataset, we have used the k-fold cross-validation method (with k=10). The dataset is randomly partitioned into *k* (10) equal sized subsets. The DT hyperparameters are: maximum depth = 4, split criterion = Gini's diversity index, maximum split = 10. We have considered the critical weather elements (temperature, RH, and WS) as attributes (features), the rain attribute is not used in the performance evaluation since it is an irrelevant attribute, i.e., it has no impact on the DT model prediction (result). The tabulated results show that the obtained performance is significant, mainly in terms of recall and accuracy. The obtained accuracy is about 75% and the recall value is about 0.88. The recall value indicates that among the '138' fire cases, '121' are correctly predicted, which is a significant prediction result. We give in Figure 2 the ROC curve for the DT model.

Table 1. Software based DT model-performance evaluation results

Datasets	Recall	Accuracy (%)	Precision (%)	AUC
Bejaia dataset	0.76	70.7	72.7	0.67
Sidi-Bel-Abess dataset	0.86	74.6	77.3	0.76
The two regions datasets	0.88	75	73.1	0.69

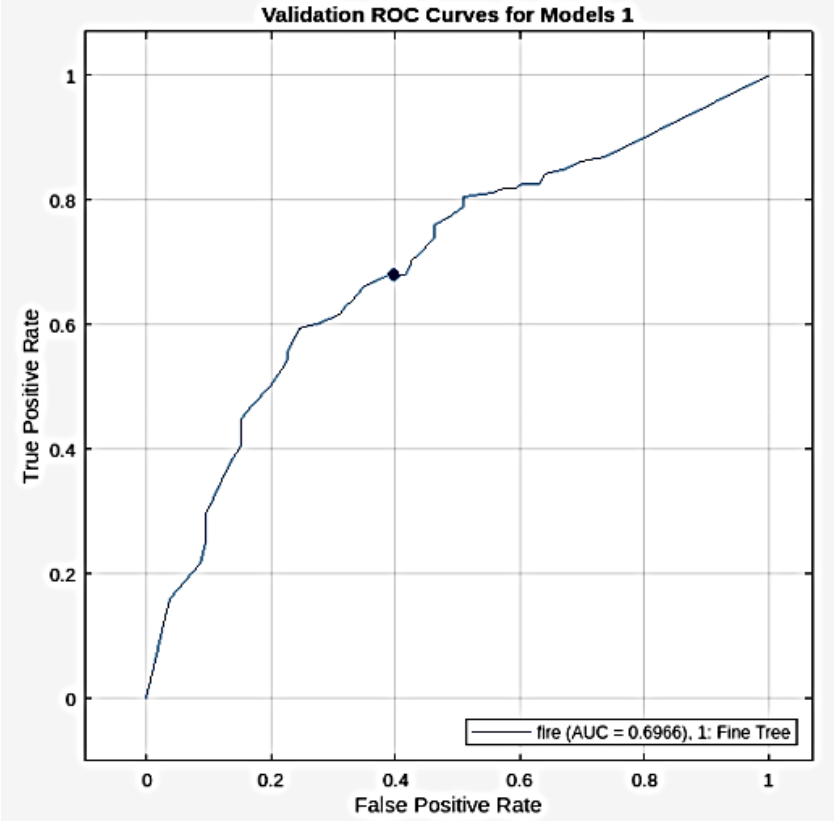

Figure 2. ROC curve for the DT model

4.2.1. Model performance stability

We have evaluated the performance stability across data segmentation ratios method. We have adopted model validation (performance evaluation) under changing training-to-testing splits and compared the metrics values. Four split configurations: 50%-50%, 70%-30%, 80%-20%, and 90%-10% were considered. The obtained results summarized in Table 2 demonstrate the model's performance stability across the four partitioning ratios. Indeed, we observed relatively low variation in all metrics' values (As shown in Table 2 the metrics values are not changing significantly). Besides, we have evaluated the performance stability using the k-fold cross-validation method, considering different values of k (k =5, 10, 7, 4, 15) to confirm the stability of the model's performance across various dataset's subsets. Results summarized in Table 3 show the model's performance stability across different k-flods evaluation.

Table 2. Performance stability across data segmentation ratios

Split ratio	Accuracy (%)	Recall	Precision (%)	AUC
50%-50%	75	88.4	73.1	0.61
70%-30%	72.6	80.5	73.3	0.76
80%-20%	77.1	88.9	75	0.75
90%-10%	79.2	78.6	84.6	0.84

Table 3. Performance stability evaluation using k-fold cross-validation

K value	Accuracy (%)	Recall	Precision (%)	AUC
5	74.6	84.8	74.1	0.76
10	75.0	88.0	73.1	0.69
7	70.9	78.3	72.5	0.72
4	70.9	79.9	71.7	0.71
15	74.2	83.3	74.2	0.73

The classification in our case is for fire prediction purpose, the important issue is to know how much the classifier predicts correctly the positive classes. The recall value indicates the sensitivity and gives this estimation of correctly predicted positive cases (fire occurrence). Thus we compared our work to the existing works, mainly in terms of recall. The second compared metric is the F-measure, which combines the recall and the precision values. The F-measure indicates the robustness of the classifier, i.e. its ability to correctly classify instances without missing a significant number of examples. We give in Table 4 a comparison between the existing works and this work (regarding the DT software-based implementation). As shown in Table 4, the obtained recall is better than the result reported in [3], [8] and comparable to the one reported in [2]. Regarding the F-measure value, the obtained result is comparable to the other reported results. It is noteworthy to specify that this comparison is indicative in view of the difference in the datasets and experiments setups used in each study; nevertheless, the obtained results are promising and comparable to exiting works.

Table 4. Comparison with existing works

References	Recall	F-Measure
This work	0.88	0.80
Stojanova *et al.* [8]	0.81	0.81
Maksimović *et al.* [2]	0.96	0.96
Giuntini *et al.* [3]	0.83	0.86

5. SYSTEM DESIGN

5.1. System overview

Figure 3 shows the main parts of the proposed FPP system, which consists of two main parts. The first part is for sensing data, it includes two types of meteorological data sensors, namely temperature and RH sensors and an anemometer (to measure the WS). As highlighted previously, the forest fire prediction is based on the meteorological data corresponding to the critical weather elements that influence the forest fire occurrence, namely temperature, RH, and WS.

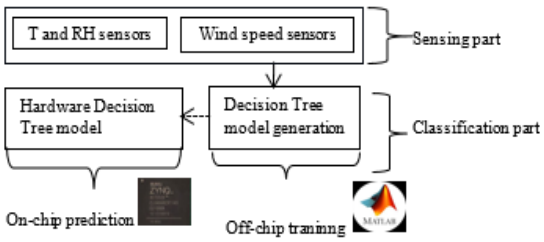

Figure 3. The main parts of the proposed forest-fires prediction (FFP) system

The second part is the classification part where a DT is used to classify and predict the fire occurrence cases (‘fire’ or ‘not fire’). The DT classifier is developed and trained off-chip for the tree generation prior to its hardware development using the HLS approach. The structure of the generated DT forms off-chip training (depth =4 and number of nodes = 6) is illustrated in Figure 4 and the internal DT_IP block diagram is depicted in Figure 5. A low-cost development board for the Xilinx Zynq®-7000 SoC, i.e., the Zedboard is targeted for the system prototyping.

The six comparators represent the six tree’s nodes: three comparators for the RH feature (thresholds are 44.5, 45 and 70), one threshold for the temperature (threshold = 30.5) and two thresholds for the WS feature (thresholds are 14.5 and 10.5).

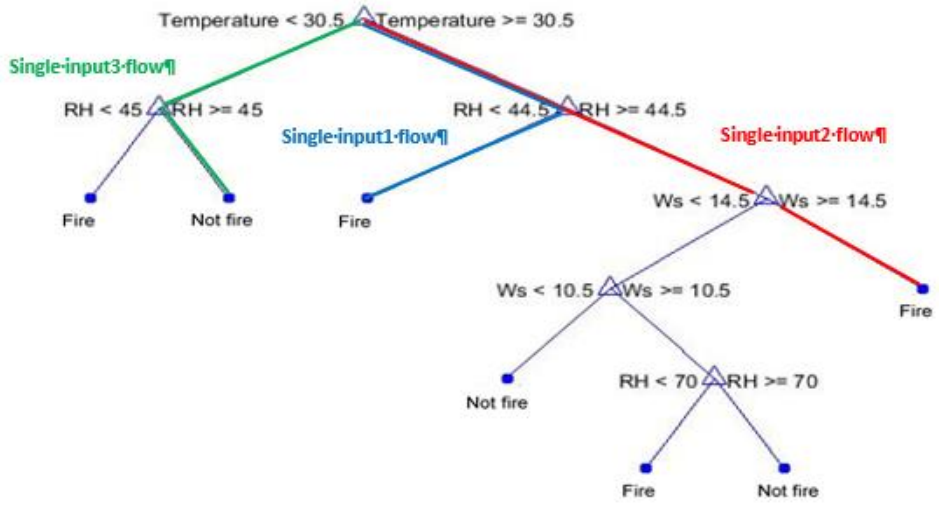

Figure 4. Structure of the generated DT

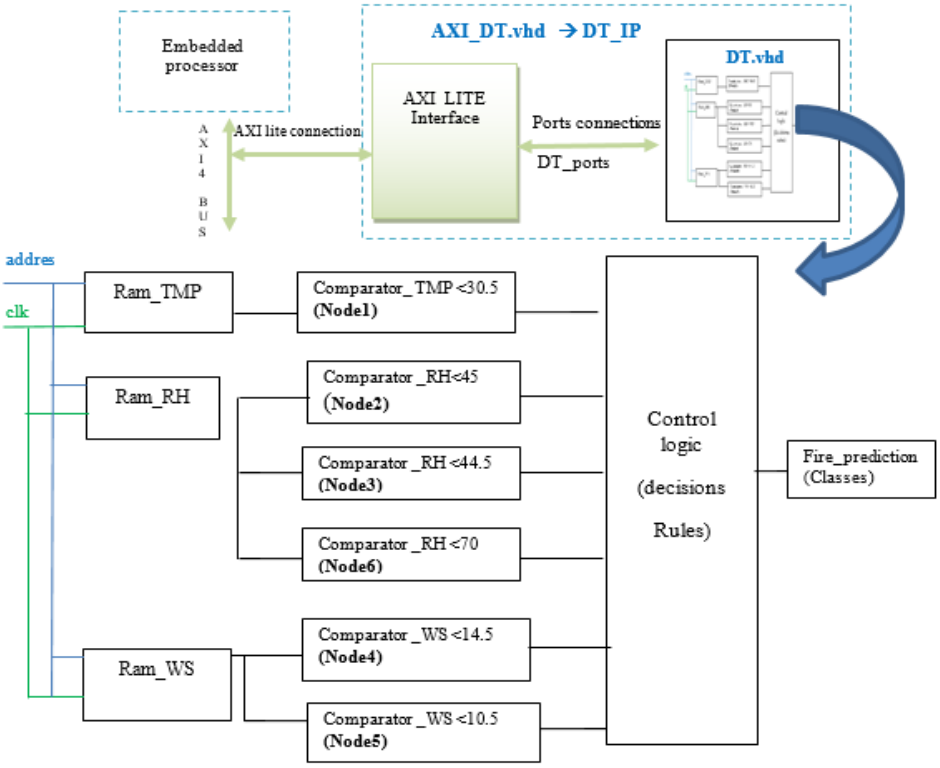

Figure 5. Internal DT_IP block diagram

5.1.1. Single data input inference results

Figure 6 shows the simulation results of a single data input inference. A single data input represents a combination of weather data (TEMP, WS, and RH), while the resulting output (fire_prediction) denotes the 'fire' or 'not_fire' classes. We have considered the two cases: 'fire' and 'not fire'.

a) Exemple of a single data input 1 → output1 'fire class'

The model receives a single input1: [TEMP, WS, RH] = [44, 0, 21] → TEMP = 44, Ws =0, RH = 21. In this case the TEMP> 30.5, RH <44.5, the inference result output1 → fire_prediction = '1' (fire class). The data flow is shown in blue in the Figure 4.

b) Exemple of a single data input 2 → output2 'fire class'

The model receives a single input2: [TEMP, WS, RH] = [44, 18, 80] → TEMP = 44, WS =18, RH =80. In this case the TEMP> 30.5, RH >44.5 and WS> 14.5, the inference result output2 → fire_prediction = '1' (fire class). The data flow is shown in red in Figure 4.

c) Exemple of a single data input3 → output3 'not fire' class

The model receives a single input3: [TEMP, WS, RH] = [23, 0, 65] → TEMP = 23, WS =0, RH = 65. In this case the TEMP<30.5, RH >45, the inference result output3 → fire_prediction = '0' (not fire class). The single data flow in this case is shown in green in Figure 4.

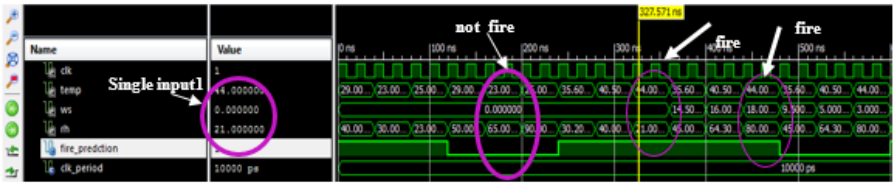

Figure 6. Single data input inference results

5.3. System architecture

The data required for the evaluation of the proposed FFP system will be collected by the sensing part. In the proposed system the meteorological data collection is performed by a set of sensors namely, temperature and RH sensors and anemometers for WS measurement. The second part is the processing part based on the Zynq PS, which performs the data acquisition and decisions visualization (the fire prediction results). It is noteworthy to precise that the data acquisition unit is not yet implemented (the sensing part is still not realized). The data is stored in a memory as an alternative for the FFP system evaluation. The DT is added as a separated HLS IP (DT_IP) core, which represents the fire classifier. The hardware DT classifier is executed on the PL part of the system. The AXI interface is used for the PS and PL parts communication. The FFP system whole architecture implementation diagram is depicted in Figure 7.

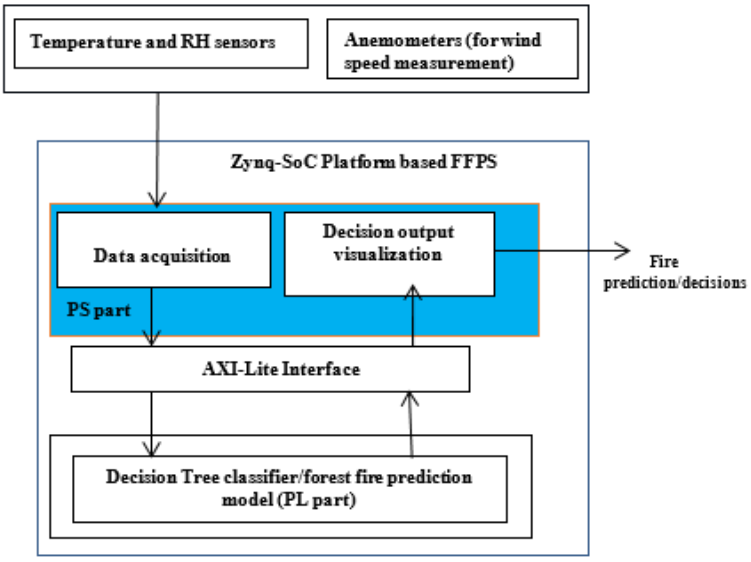

Figure 7. The FFP system whole architecture implementation diagram

6. SYSTEM HARDWARE IMPLEMENTATION

6.1. The FFP SoC architecture design

6.1.1. The FFP SoC hardware design

We have adopted the high-level modeling approach for the DT_IP core design for the fire-prediction process hardware acceleration. The HLS approach transforms a macro-architecture into micro-architecture with gain in the development and verification time. The HLS approach allows fast development and prototyping on the hardware model compared to the classical implementation approaches using hardware

description languages (HDL). The HLS IP design allows the execution of the code initially executed by the CPU into the FPGA target. Examples of HLS commercial tools are the Vivado HLS from Xilinx [23] and the Stratus HLS from Cadence.

In this work, we have adopted the Vivado HLS approach, where the IP is created and simulated using C/C++ code then exported to the Vivado IP Catalog. Note that the Vivado HLS tool generates drivers related to the developed IP; these drivers are included in the IP package and will be used by the Zynq processing part. An AXI interface is used to interconnect the HLS DT_IP core to other components. This makes the designed IP reusable and allows its integration into several architectures based on various processors (e.g., the MicroBlaze and ARM processors). The developed DT_IP core exported to the IP Catalog is integrated into the FFP SoC implemented in the Zynq-SoC platform.

The block design implementation of the proposed FFP Zynq-SoC is depicted in Figure 8. The designed DT based fire prediction IP with an AXI interface core is called 'DT_0', this latter constitutes the chief block in the FFP SoC architecture, which includes the Zynq PS (for configuration purpose), the developed DT_IP core, the AXI interconnect IP used to connect the designed HLS DT_IP with the Zynq PS and the PS reset (rst_processing_system). Additional IP cores can be integrated to the proposed SoC for further functionalities, e.g., the GPIO for on board led utilization.

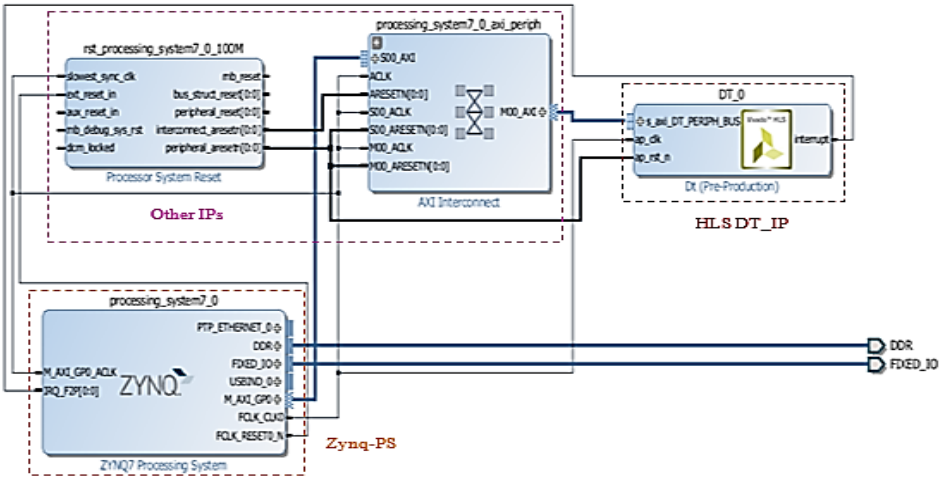

Figure 8. The FFP Zynq-SoC block design implementation

6.1.2. FFP SoC software design

We have performed the DT based forest fire-prediction software model validation using the co-design approach. The validation of the proposed HW/SW system running the FFP application, which executes the developed 'DT.c' classifier was performed on the Nexys4 DDR board (the first FPGA board used in the project). The DT based forest-fire classifier software model validation results using the co-design approach and the experimental test results visualization; meaning alarms using LED (on the FPGA board) and text message have been reported in our previous work [24]. The designed hardware model (including the FPGA programming file "bitstream") is exported to the Software Development Kit (SDK), where the FFP application code is written and will be executed by the PS part. The FFP application is created to test the HLS DT_IP based prediction model. It includes mainly, the source code read/write data from/to the HLS DT_IP core. The driver files created and exported from the HLS tool include the data read and write function, which allows communication with the hardware implemented in the PL part. The test application includes, mainly the function dedicated to the insertion of the three meteorological variables (temperature, RH, and WS), and the function to retrieve the decision (prediction) variable ('fire'/'not fire'), which indicates the result provided by the designed DT classifier's accelerator (the DT_IP).

7. RESULTS AND DISCUSSION

We discuss in this section the hardware implementation results including the Zynq-SoC based FFP SoC and the HLS DT_IP hardware implementation results. We also give the primary results in terms of power consumption.

7.1. Zynq-SoC based FFP SoC implementation results

The Zynq-SoC platform, which combines the PS and the PL parts inside a single chip is chosen for the FFP system implementation. The Zynq-SoC platform gives tradeoff between power consumption and resources utilization compared to the MicroBlaze based FFP SoC [25]. The Xilinx Zynq®-7000 SoC in the Zedboard has been targeted for the FFP Zynq-SoC implementation. Table 5 tabulates the implementation results in terms of area occupancy for the FFP Zynq-SoC. The percentage of the total resources utilization is about 6.84%; with 2.4% of LUTs, 0.92% of FFs and about 0.40% for memory. The primary implementation results in terms of hardware resources utilization for the DT IP core (within the FFP SoC architecture) are summarized in Table 6. Percentages of about 0.84% (445) of LUTs and 0.31% (325) of FFs are used by the designed DT_IP. The obtained latency (clock cycles) values are about '3' and '2', for default setting (before pipelining) and with optimization using pipeline directive (after pipelining), respectively. We observe that there is an improvement in terms of latency, i.e., a decrease in latency (clock cycles) when using the pipeline directive. We have also schematized in Figure 9 the resources utilization of the DT_IP core in respect to the whole FFP Zynq-SoC. The results represented in Figure 9 show that the developed DT_IP core is area-efficient since it requires few resources when integrated in the whole FFP SoC.

Table 5. Zynq-SoC based FFP SoC hardware resources utilization

Hardware resources	Available	Utilization	Percentage (%)
LUTs	53200	1275	2.4
FFs	106400	983	0.92
Memory	17400	70	0.40
BUFG	32	1	3.12

Table 6. The DT_IP hardware resources utilization

Hardware resources	Available	Utilization	Percentage (%)
LUTs	53200	445	0.84
FFs	106400	325	0.31
Memory	17400	0	0

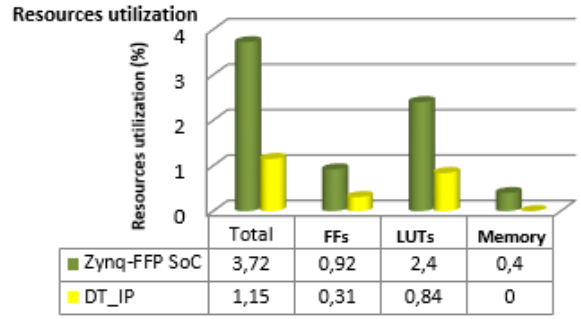

Figure 9. Resources utilization (DT_IP core in respect to the whole FFP Zynq-SoC)

7.2. Power consumption

Table 7 summarizes the power consumption results for the proposed FFP Zynq-SoC. The power consumption is about 1.698W for the whole SoC, with 1.540W (91%) for the dynamic power and about 0.158W (9%) for the static power consumption. The processing part (PS7) consumes about 1.529W dynamic power, this value represents 97% of the whole SoC power consumption. The logic part (PL), which includes the DT_IP consumes about 0.003W (<1%). The dynamic power consumption results show that the part that includes the DT_IP core (the PL part) consumes less power than the PS part (which includes the ARM processor running at 667 MHz). The results demonstrate that the DT_IP based fire prediction classifier is power-efficient.

Table 7. Power consumption for the FFP SoC

Power (W)	FFP SoC	PS	PL (that includes the DT_IP core)
Total	1.698W		0.84
Dynamic power	1.540W (91%)	1.529W (97%)	0.003W (1%)
Static power	0.158 (9%)		0

7.3. Comparison with FPGA-based implementations

We present in this section the outcomes of the conducted comparison with exiting FPGA-based implementations of the DT classifier, in terms of resource utilization and power consumption. For fair comparison, we have considered, mainly studies exposed in section 2 (related work) that used FPGA platform instead of the ASIC platform. It is noteworthy to specify that we did not compare with studies that considered the DT-ensemble models hardware implementation ([17], [19]-[22]), because these architectures include several trees instead of a single tree. Besides, we point-out that the comparison is beyond the targeted application. As we have previously highlighted, to our knowledge there is no hardware DT-implementations targeting forest fire prediction in the available literature. The outcomes of the conducted comparison are summarized in Table 8. The conducted comparison reveled that:

a) Regarding the hardware single DT implementations (DT_IP): comparison with [10], [11], [13], [16]. We observe that, in terms of FPGA resources utilization (LUTs and FFs), the DT_IP architectures proposed in [10] (44 LUTs, 45 FFs) and [11] (49 LUTs, 22 FFs) are more optimized than our DT_IP implementation (445 LUTs, 325 FFs). While DT_IP architectures proposed in [13] (160 LUTs, 233 FFs) and [16] (240 LUTs, 320 FFs) resources utilization are somehow comparable to our implementation (445 LUTs, 325 FFs), mainly in terms of FFs utilization.
b) Regarding the implementations as a whole system that integrates the DT_IP: comparison with [14]-[16]. For fair comparison, we have compared our results against the results reported in [14]-[16], which used the same approach, i.e., the HLS modeling and targeted similar platform, i.e., the Zynq SoC platform. The results in terms of FPGA resources utilization considering the whole system that integrates the DT_IP show that our architecture resources utilizations (1275 LUTs, 983 FFs) are comparable to the results reported in [14] (2799 LUTs, 895 FFs) in terms of FFs and less in terms of LUTs. Compared to [15] (3925/3310 LUTs, 2089/1793 FFs) and [16] (6442 LUTs, 5336 FFs) our whole system architecture is more optimized (uses less resources in terms of LUTs and FFs).
c) In terms of power consumption considering the whole system that integrates the DT_IP (1.698W total power, 1.529W (PS) dynamic power, 0.003W (PL) dynamic power, 0.003W HLS DT_IP dynamic power), our results are comparable to results reported in [15] in terms of power consumption of the PS part (1.564W) and better regarding the PL power consumption (0.13W max, 0.041W min for [15] against 0.003W for our architecture). Regarding the HLS DT_IP dynamic power consumption (0.003W) our result is comparable to the result reported in [15] (0.001W). Note that the studies reported in [16] and [14] did not provide results in terms of power consumption.
d) In terms of frequency, our whole system's frequency (128 MHz) is better than [14] (121 MHz) and comparable to [15] (141 MHz).

7.4. Limitations

Even though the concept's functional-validation, i.e., the DT accelerator-based forest fire prediction was reached, and despite the designed FFP SoC hardware efficiency (low-cost FPGA implementation), we acknowledge limitations regarding: the used dataset limitations in terms of size and generalizability: in fact, an extended Algerian forest fire dataset was recently created to address these limitations. To increase temporal coverage and geographical scope, the extended dataset includes multi-year data (2012, 2017 and 2021) of nine regions of Algeria. The extended dataset comprises 1215 instances against 244 instances for the initial dataset. It is publicly available at: https://ieee-dataport.org/documents/extended-algerian-forest-fires-dataset.

We point-out that the performance evaluation results using the extended dataset (1215 instances) are about 87.8%, 87.3, 88.3%, 87.79 and 91 for accuracy, recall, precision, F-Score and AUC, respectively. It is noteworthy to specify that in this paper we present the first release (DT_0) of the hardware implemented model (DT_IP) using the DT structure generated with the initial dataset (244 instances). Indeed, the hardware implementation of the DT classifier (second release of the DT_IP) structure issued from the evaluation using the extended dataset will be considered in eventual further work.

The DT classifier robustness analysis: the robustness analysis of the DT classifier under noisy or incomplete inputs is an important issue that is not yet tackled. It is noteworthy to specify that we have targeted as first objective the functionality validation (we focus on the concept feasibility) of the proposed FPGA-based forest fire prediction system that integrates the DT classifier as part of the smart sensor node processing unit architecture. The refinement of the DT classifier and its robustness analysis will be addressed in eventual future work.

The predictors (features) variety limitation: improvement in terms of the DT classifier performance can be reached by considering additional predictors (features). Indeed, in addition to the meteorological predictors, it will be practical to consider additional predictors (features), such as topographical and anthropological (human factors) features.

Table 8. FPGA-based implementation results comparison with exiting works

References	Platform/Hardware support	Results (resources utilization and power consumption)	Targeted application	Implementation type (single DT_IP or whole system)
		Hardware single DT implementations		
Amato *et al*. [10]	FPGA Virtex 5	LUTs: 44 FFs: 45	Not specified	Single DT_IP implementation
Sayed *et al*. [11]	FPGA VCU108 ASIC umc180nm (IC)	FPGA LUTs: 49 FFs: 22 IOs: 28 ASIC Area: 5189µm2 Power: 0.22mw	Real time applications. Not specified application	Single DT_IP implementation
Li and Bermak [13]	FPGA Virtex 2 ASIC 0.18µm	FPGA LUTs: 160 FFs: 233 ASIC Area: 8.9µm2 Power:3.1 mw	Gas identification	Single DT_IP implementation
Saqib *et al*. [16]	FPGA Nexys2 saprtan3E	Single DT 4 stage: LUTs: 240 FFs: 320 Whole system: LUTs: 6442 FFs: 5336 BRAMs: 22	Not specified	Single DT_IP implementation and the whole system that integrates the DT_IP
This work	FPGA Zynq SoC	Whole system: LUTs: 1275 FFs: 983 Frequency:128 MHz MEM:70 Single DT_IP LUTs :445 FFs: 325 Total Power: 1.698W PS dynamic power: 1.529W PL dynamic power: 0.003W HLS DT_IP dynamic power: 0.003W	Forest fire prediction	Whole system that integrates the HLS DT_IP,
Ali *et al*. [14]	FPGA Zynq SoC	LUTs: 2799 FFs: 895 Frequency: 121 MHz	Falls detection	Whole system that integrates HLS DT_IP
Ali *et al*. [15]	FPGA Zynq SoC	Whole system LUTs: 3925/3310 FFs: 2089/1793 Frequency :142 MHz MIN FFs: 303 MIN LUTs: 764 Dynamic power PS: 1.564W PL: 0.13W (max) 0.041W (min) HLS IP min power: 0.001W HLS IP: max power 0.135W	Electronic noise (Gas identification)	Whole system that integrates HLS DT IP
		Hardware ensembles DT implementations		
Essen *et al*. [20]	FPGA: Hitech Global HTG-V6-PCIE-L240-1 (Virtex 6)	LUTs: 76,731 to 139,348 Slice registers: 48,642 to 122,370	Identifying suspicious URLs	Ensemble DT: RF
Buschjager and Morik [21]	FPGA Zedboard	If-else implementation LUTs: 10244 FFs: 1317 Naïve tree LUTs: 31 FFs: 64 DNF (disjunctive normal form) tree LUTs: 9609 FFs: 3183	FACT application	Ensemble DT: RF

7.5. Scalability issues and challenges of the FPGA-based implementation in real environment

In this section, we list scalability issues in terms of cost, power consumption, and communication. It is noteworthy to precise that we did not address the scalability issues in details, since they are beyond the scope of this paper, as we focus on the concept feasibility i.e., on accelerating the classification phase using hardware DT for local decision at sensor node level. In fact, extending the use of the proposed FPGA-based sensor node architecture (evaluated at single-node level) to real-world forest monitoring deployments with hundreds or thousands of nodes introduces several key scalability challenges that require careful attention. While FPGAs offer high computational performance, parallel processing capabilities, high integration density, and flexibility, their use as wireless sensor nodes in real environmental monitoring networks can raise concerns in terms of cost, energy consumption, and communication bottlenecks.

First, the cost per node remains a critical issue. Compared to microcontroller-based solutions, using FPGA platforms can significantly increases deployment costs when scaling to hundreds or thousands of nodes. In large forest environments, where a high density of sensors is required for reliable prediction, this cost becomes a major barrier to large-scale adoption. This issue can be addressed by targeting low-cost FPGA devices such as the Xilinx Artix 7 FPGA device, which gives tradeoff between costs and processing power and decrease the overall deployment costs.

Second, high power consumption limits scalability. In real-world forest deployments, sensor nodes perform continuous sensing, local processing, and prediction. We recognize that FPGA-based nodes inherently consume more power than low-power MCU-based systems, making long-term autonomous operation difficult in energy-constrained environments. However, MCUs are not suitable for applications that require real-time processing and decision speed up, where FPGAs are alternatives. Solution such as energy harvesting techniques can address the power consumption issue for FPGA-based nodes using for exemple, solar based battery as part of the power unit for power saving.

Third, scaling continuous sensing, local processing, and prediction tasks to hundreds or thousands of nodes leads to network congestion, which causes communication bottlenecks. It is noteworthy to specify that, the proposed local data processing and decision at sensor node level is one key strategy that can handle constraints related to communication and network congestion by reducing the number of direct data transmitted to the central station (which receives the decision only, while the data collected by the sensors is processed locally) reducing, hence energy consumption, network bandwidth and power. As we highlighted previously, scalability issue is beyond the scope of this paper, future work will address this concern.

8. CONCLUSION

In this paper, we have exposed a SoC for forest fire prediction. Our first aim is the development of a predictive model based on the DT classifier for FPP in Algeria. The second objective is its hardware implementation as an IP core that constitutes the main part of the proposed FFP SoC. We propose a local data processing and decision at sensor node level, which is one key strategy that can handle constraints related to communication and network congestion by reducing the number of direct data transmitted to the central station reducing, hence energy consumption, network bandwidth and power.

The DT classifier offers optimal trade-off between predictive accuracy and hardware efficiency, positioning it as an appropriate candidate for the hardware acceleration of the fire prediction process. Indeed, tow aspects have been considered for the ML model selection: First, the model should be ease to convert to hardware (has a less complex structure) and hardware-efficient (low-cost FPGA-based implementation), since the generated model will be implemented as an IP core, which will be integrated to the smart sensor node architecture (processing unit). Secondly, the ML classifier should give acceptable performance and accurate prediction.

The developed DT_IP for hardware acceleration of the forest fire classification (prediction) task has been integrated into the FFP Zynq-SoC. Synthesis results show that the DT-based prediction model is area-efficient when implemented as an IP core; this makes its suitable for our purpose since it is a part of the whole SoC, which includes other IPs. Besides, the DT classifier gives acceptable performance compared to existing works (software-based), mainly in terms of recall. Promising results are obtained with the proposed DT classifier in terms of recall and accuracy, with about 0.88 and 75% for recall and accuracy, respectively. We point-out that the system is still in prototype stage.

We aim further to develop the other parts of the sensor node, namely the communication and sensing units, which integrates multi-sensors for weather data collection, such as temperature, RH, and WS; to construct an automatic FPP system. In addition to the weather data sensing, the sensor node will be able to predict fire occurrence locally using the DT_IP classifier, thus decreasing the communication cost and the energy consumption.

ACKNOWLEDGMENT

We would like to acknowledge the editors and reviewers for their constructive comments and useful recommendations and suggestions, which contribute to this manuscript improvement.

FUNDING INFORMATION

This work has been done under the socio-economic project no14/cdta/dgrsdt.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contribution Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Faroudja Abid	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		
Nouma Izeboudjen										✓			✓	✓
Fatiha Louiz			✓	✓										

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that no conflicts of interest pertaining to the publication of this manuscript.

DATA AVAILABILITY

The initial dataset (used in this study) is available in the UCI repository at: https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+datasetThe extended dataset is publicly available at: https://ieee-dataport.org/documents/extended-algerian-forest-fires-dataset

REFERENCES

[1] A. Benkheira, "Forest fires in Algeria analysis and perspectives," *Technical Report, Ministry of Agriculture and Rural Development and Fisheries, Forest Department, Algeria*, 2018.
[2] M. Maksimović and V. Vujović, "Comparative analysis of data mining techniques applied to wireless sensor network data for fire detection," *JITA - Journal of Information Technology and Applications (Banja Luka) - APEIRON*, vol. 6, no. 2, Dec. 2013, doi: 10.7251/jit1302065m.
[3] F. T. Giuntini, D. M. Beder, and J. Ueyama, "Exploiting self-organization and fault tolerance in wireless sensor networks: A case study on wildfire detection application," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 155014771770412, Apr. 2017, doi: 10.1177/1550147717704120.
[4] M. Bahrepour, B. J. van der Zwaag, N. Meratnia, and P. Havinga, "Fire data analysis and feature reduction using computational intelligence methods," in *Advances in Intelligent Decision Technologies*, Springer Berlin Heidelberg, 2010, pp. 289–298.
[5] M. Saoudi, A. Bounceur, R. Euler, and T. Kechadi, "Data mining techniques applied to wireless sensor networks for early forest fire detection," in *Proceedings of the International Conference on Internet of things and Cloud Computing*, Mar. 2016, pp. 1–7, doi: 10.1145/2896387.2900323.
[6] M. Hefeeda and M. Bagheri, "Wireless sensor networks for early detection of forest fires," in *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, Oct. 2007, pp. 1–6, doi: 10.1109/mobhoc.2007.4428702.
[7] Y. Liu, Y. Gu, G. Chen, Y. Ji, and J. Li, "A novel accurate forest fire detection system using wireless sensor networks," in *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*, Dec. 2011, pp. 52–59, doi: 10.1109/msn.2011.8.
[8] D. Stojanova, P. Panov, A. Kobler, S. Džeroski, and K. Taškova, "Learning to predict forest fires with different data mining techniques," *Proc. 9th International Multi Conference Information Society, Jubljana, Slovinia*, 2006.
[9] F. Abid, "A survey of machine learning algorithms based forest fires prediction and detection systems," *Fire Technology*, vol. 57, no. 2, pp. 559–590, Nov. 2020, doi: 10.1007/s10694-020-01056-z.
[10] F. Amato, M. Barbareschi, V. Casola, and A. Mazzeo, "An FPGA-based smart classifier for decision support systems," in *Intelligent Distributed Computing VII*, Springer International Publishing, 2014, pp. 289–299.
[11] R. Sayed, H. Azmi, A. M. Nassar, and H. Shawkey, "Design automation and implementation of machine learning classifier chips," *IEEE Access*, vol. 8, pp. 192155–192164, 2020, doi: 10.1109/access.2020.3032658.
[12] R. Narayanan, D. Honbo, G. Memik, A. Choudhary, and J. Zambreno, "An FPGA implementation of decision tree classification," in *2007 Design, Automation & Test in Europe Conference & Exhibition*, Apr. 2007, pp. 1–6, doi: 10.1109/date.2007.364589.
[13] 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, no. 1, pp. 45–58, Mar. 2011, doi: 10.3390/jlpea1010045.

[14] A. A. S. Ali, M. Siupik, A. Amira, F. Bensaali, and P. Casaseca-de-la-Higuera, "HLS based hardware acceleration on the zynq SoC: A case study for fall detection system," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, Nov. 2014, pp. 685–690, doi: 10.1109/aiccsa.2014.7073266.
[15] A. A. S. Ali, H. Djelouat, A. Amira, F. Bensaali, M. Benammar, and A. Bermak, "Electronic nose system on the Zynq SoC platform," *Microprocessors and Microsystems*, vol. 53, pp. 145–156, Aug. 2017, doi: 10.1016/j.micpro.2017.07.012.
[16] F. Saqib, A. Dutta, J. Plusquellic, P. Ortiz, and M. S. Pattichis, "Pipelined decision tree classification accelerator implementation in FPGA (DT-CAIF)," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 280–285, Jan. 2015, doi: 10.1109/tc.2013.204.
[17] R. Struharik, "Decision tree ensemble hardware accelerators for embedded applications," in *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, 2015, pp. 101–106, doi: 10.1109/sisy.2015.7325359.
[18] J. R. Struharik, "Implementing decision trees in hardware," in *2011 IEEE 9th International Symposium on Intelligent Systems and Informatics*, 2011, pp. 41–46, doi: 10.1109/sisy.2011.6034358.
[19] A. Jinguji, S. Sato, and H. Nakahara, "An FPGA realization of a random forest with k-means clustering using a high-level synthesis design," *IEICE Transactions on Information and Systems*, vol. E101.D, no. 2, pp. 354–362, 2018, doi: 10.1587/transinf.2017rcp0006.
[20] B. V. Essen, C. Macaraeg, M. Gokhale, and R. Prenger, "Accelerating a random forest classifier: multi-core, GP-GPU, or FPGA?," in *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, Apr. 2012, pp. 232–239, doi: 10.1109/fccm.2012.47.
[21] S. Buschjager and K. Morik, "Decision tree and random forest implementations for fast filtering of sensor data," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 209–222, Jan. 2018, doi: 10.1109/tcsi.2017.2710627.
[22] M. Barbareschi, S. Del Prete, F. Gargiulo, A. Mazzeo, and C. Sansone, "Decision tree-based multiple classifier systems: an FPGA perspective," in *Multiple Classifier Systems*, Springer International Publishing, 2015, pp. 194–205.
[23] "Vivado design suite user guide high-level synthesis," *UG902 (v2016.1) April 6*, 2016, [Online]. Available: www.xilinx.com.
[24] F. Abid, N. Izeboudjen, and F. Louiz, "Hardware/software development of a machine learning based forest fires prediction system," in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, Dec. 2021, pp. 1–5, doi: 10.1109/icisat54145.2021.9678486.
[25] F. Abid and N. Izeboudjen, "Zynq-SoC vs.microblaze based soc for forest fires prediction: comparison study," in *12th International Conference on Information Systems and Advanced Technologies "ICISAT 2022,"* Springer International Publishing, 2023, pp. 279–289.

BIOGRAPHIES OF AUTHORS

Dr. Faroudja Abid received the Engineering degree in Electronics Engineering from the University of Tizi-Ouzou, Algeria, the M.Sc. from Ecole Nationale Polytechnique (ENP) of Algiers, and a Ph.D. degree from Ecole Nationale Polytechnique (ENP) of Algiers. Currently, she is a researcher at CDTA center. Her research interests include embedded system, FPGA and ASIC design, SoC, hardware machine learning (ML) and FPGA-ML accelerators. She can be contacted at email: fabid@cdta.dz.

Dr. Nouma Izeboudjen received the diploma of Electronic Engineering degree from the University of Science and Technology Houari Boumedienne (USTHB), Algiers; the diploma of magister degree in electronic-telecommunication from the Ecole nationale polytechnique (ENP), Algiers; and also, the Doctorate in science degree in electronic engineering from the ENP. She is currently affiliated with the division of microlectronic and nanotechnology at the (CDTA) research center. Her current research area interest includes implementation of intelligent systems, hardware software co-design, low power and energy harvesting techniques. She can be contacted at email: nizeboudjen@cdta.dz.

Fatiha Louiz received the Engineering degree in Computer Sciences from the University of Science and Technology Houari Boumedienne (USTHB), Algiers. Currently, she is an engineer at CDTA center. Her research interests include optimization, and Computer Arithmetic for VLSI Applications. She can be contacted at email: flouiz@cdta.dz.