

Spectrum Sensing Using Optimized Deep Learning Techniques in Reconfigurable Embedded Systems

Priyesh Kumar* and Ponniyin Selvan

Mahendra College of Engineering, Salem, 636106, India

*Corresponding Author: Priyesh Kumar. Email: atpkec@gmail.com

Received: 23 March 2022; Accepted: 02 August 2022

Abstract: The exponential growth of Internet of Things (IoT) and 5G networks has resulted in maximum users, and the role of cognitive radio has become pivotal in handling the crowded users. In this scenario, cognitive radio techniques such as spectrum sensing, spectrum sharing and dynamic spectrum access will become essential components in Wireless IoT communication. IoT devices must learn adaptively to the environment and extract the spectrum knowledge and inferred spectrum knowledge by appropriately changing communication parameters such as modulation index, frequency bands, coding rate etc., to accommodate the above characteristics. Implementing the above learning methods on the embedded chip leads to high latency, high power consumption and more chip area utilisation. To overcome the problems mentioned above, we present DEEP HOLE Radio systems, the intelligent system enabling the spectrum knowledge extraction from the unprocessed samples by the optimized deep learning models directly from the Radio Frequency (RF) environment. DEEP HOLE Radio provides (i) an optimized deep learning framework with a good trade-off between latency, power and utilization. (ii) Complete Hardware-Software architecture where the SoC's coupled with radio transceivers for maximum performance. The experimentation has been carried out using GNURADIO software interfaced with Zynq-7000 devices mounting on ESP8266 radio transceivers with inbuilt Omni directional antennas. The whole spectrum of knowledge has been extracted using GNU radio. These extracted features are used to train the proposed optimized deep learning models, which run parallel on Zynq-SoC 7000, consuming less area, power, latency and less utilization area. The proposed framework has been evaluated and compared with the existing frameworks such as RFLearn, Long Term Short Memory (LSTM), Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN). The outcome shows that the proposed framework has outperformed the existing framework regarding the area, power and time. Moreover, the experimental results show that the proposed framework decreases the delay, power and area by 15%, 20% 25% concerning the existing RFLearn and other hardware constraint frameworks.

Keywords: Internet of things; cognitive radio; spectrum sharing; optimized deep learning framework; GNU radio; RF learn



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Internet of things (IoT) is often characterized by high mobility, high traffic demands, vulnerability to different security attacks, and jamming. Hence, new agile, intelligent solutions are required for efficient management among IoT wireless devices. Moreover, it is expected that by 2028, over 70 billion devices will be absorbed into the Internet, generating the crowded world of devices as “things” [1]. Integrating Cognitive Radio in IoT has enabled technology to efficiently use spectrum resources and support wireless communication for increasing traffic demand [2]. Cognitive Radio (CR) in IoT performs various tasks related to spectrum management for better communication. Moreover, with the few radio spectrum bands available, cognitive radio adopts several new technologies to adopt spectrum sensing, spectrum access and spectrum sharing [3–7]. To support these tasks, Machine Learning (ML) and Deep Learning (DL) provide automated means to learn from and adapt to spectrum dynamics [8,9]. Applying ML and DL techniques to the CR-IoT Domain faces many challenges that are substantially absent in conventional learning models. Another big issue absent in traditional DL models is the need to satisfy the hardware limitations such as power, area and delay. Indeed, models with a high number of neurons/layers in deep learning networks will require additional resources and energy consumption which are scarce in IoT devices. Hence, the deep learning model is required to effectively implement CR in IoT devices and achieve the best performance. However, prior works have investigated the opportunity of using hardware-centric deep learning models to perform different CR techniques such as spectrum sensing, spectrum sharing and spectrum management [10–14]. An intelligent deep learning algorithm that can provide a good tradeoff between the hardware and performance is badly needed. RFLearn [15] is one such Field Programmable Gate Array (FPGA) architecture which has ported the deep learning algorithm using hardware-software codesign. Though the RFLearn architecture implements real-time learning, improvisation is still required to deal with real-time IoT signals. This paper proposes the new technique of integrating the novel deep learning algorithm with higher performance and also exhibits the high resource constraint parameters such as Low latency, Low power consumption and low area utilization to overcome this challenge.

2 Related Works

Mukesh V.'s point is to limit delay and expand the throughput of decision-making engine. This framework has the decision-making engine based on a Genetic Algorithm (GA) and Field Programmable Gate Array (FPGA). The GA has a multi-objective handling capacity. The entire framework runs at 199 MHz, and the ideal arrangement is gotten in 9.38 ns over the plan. At first, the plan is tried on XILINX. This work's primary target is satisfied by equally preparing the engineering of psychological motor on programming to diminish delay up to 10% over past programming plans. Nonetheless, minimising force utilization mainly expands the Bit Error Rate (BER) [16].

Zhi-Ling Tang proposes a master-slave Automatic Modulation Recognition (AMR) architecture utilizing the reconfigurability of FPGAs. Initially, this work discusses AMR's building method, which uses a convolution autoencoder. This technique can get better outcomes under the state of a low Signal to Noise Ratio (SNR), yet it is vital to ascertain different component amounts as indicated by the numerical model of the tweak strategy ahead of time [17].

Soltani planned and executed a Deep Learning (DL) put together Radio Frequency (RF) signal classifier for the FPGA of an installed programming characterized radio stage, DeepRadioTM, that groups the signs through the RF front finish to various regulation sorts progressively and with low force. This classifier execution effectively catches complex attributes of remote signs to serve basic applications in remote security and correspondence frameworks. With Cognitive Radio (CR) transmissions through the air, this show that the classifier executed on DeepRadioTM achieves high precision with low inertness and low

energy utilization, and this exhibition is not coordinated by other implanted stages, for example, installed Graphics Processing Unit (GPU). In any case, highlight amounts are not finished [18].

Maclellan expect to quantise a current Convolutional Neural Network (CNN) for radio balance grouping to 2-digit loads and enactments for sending on a Zynq System on Chip (SoC). The PYNQ system is introduced as a down-to-earth implies for getting to the usefulness of the CNN. The actual outcomes show a high order exactness even with 2-bit loads, actuators and furnish better with lower intricacy in more limited perception time. The weakness of this kind of calculation is that the presumption that the images are autonomous and indistinguishably dispersed is usually false in all actuality, and the intricacy increments with an increment in the number of obscure boundaries [19].

Mendis presented an Automatic Modulation Classification (AMC) technique for Cognitive Radio (CR). The proposed system comprises one Spectral Coherence based Function (SCF)-based element portrayal component and Dynamic Bayesian Network (DBN)-based ID conspire. This proposed SCF-based component portrayal system creates SCF designs that describe the highlights of the related tweak procedures. DBN-based ID conspire utilizes DBN to characterize the balance strategies by taking in their highlights from the related SCF designs. With the commotion, versatile SCF designs this strategy can accomplish high precision of order even within sight of climate clamour. Moreover, the DBN procedure empowers us to portray the discernable highlights of the adjustment methods having related SCF designs. Reproduction results show proposed strategies can accomplish precision above 90% in grouping the regulation methods when SNR is > -2 dB. This kind of technique is more touchy to the confound of the model; for instance, when a period move happens, the acknowledgement execution disintegrates significantly [20].

Gao presented a Convolutional Long Short-Term Deep Neural Networks (CLDNN) to defend against noise while sensing the spectrum. This framework does not require prior knowledge about primary signals or noise for the defending mechanism. Also provides good results in low SNR scenarios. However, this framework struggles when handled with real-time datasets [21].

Zheng proposed a deep learning-based solution for the spectrum sensing problem. This framework optimized the signal power to tolerate the noise power uncertainty issues. This framework also adopted transfer learning-based methods for the real-time analysis of spectrum signals. This framework provides a better result in spectrum detection, but a significant drawback of this framework is that system performance is degraded when the data size is increased [22].

Sarikhani presented Deep Reinforcement Learning (DRL) based solution for Cooperative Spectrum Sensing (CSS) issues. This framework reduced signalling in the Secondary User's (SUs) Network. This framework improved local signal sensing results and increased the spectrum sensing quality. However, the main downside of this framework is this framework struggles under low SNR scenarios [23].

Xie presented Unsupervised Deep Spectrum Sensing (UDSS) for spectrum detection. This framework's main advantage is that it does not require prior knowledge about noise power and statistical Covariance Matrix (CM). This framework provides better results under Gaussian Noise and Laplace Noise. Nevertheless, this framework requires improvisation to handle some external noise factors [24].

Zheng presented a deep learning-based solution for Primary User Adversarial Attack (PUAA). This framework utilized an auto-encoder as a defence mechanism. It can extract both temporal and local features at the same time to achieve improved system performance. Results show that this framework effectively reduced the probability of detection and defended PUAA without any degradation in its performance. However, it requires additional resources to maintain same speed of the network [25].

Xing presented a Deep Neural Network (DNN) based framework for effective spectrum sensing in CR. This framework utilized local and global features from time series data to handle low SNR scenarios. This

framework showed better results regarding false alarm rates and missed detection probabilities. However, a drawback of this framework is that it requires improvisation to handle other noises without any degradation in its performance [26].

3 Proposed Architecture

3.1 System Overview

Fig. 1 shows the system setup. There are two primary components such as transmitter and receiver. On the Transmitter side, IoT users are configured using NodeMCU that the users control. The IoT users transmit signals at 2.4 GHz frequency at various Signal to Noise Ratio (SNR) effects. The transmission completed over the air by considering the restrictions of spectrum management.

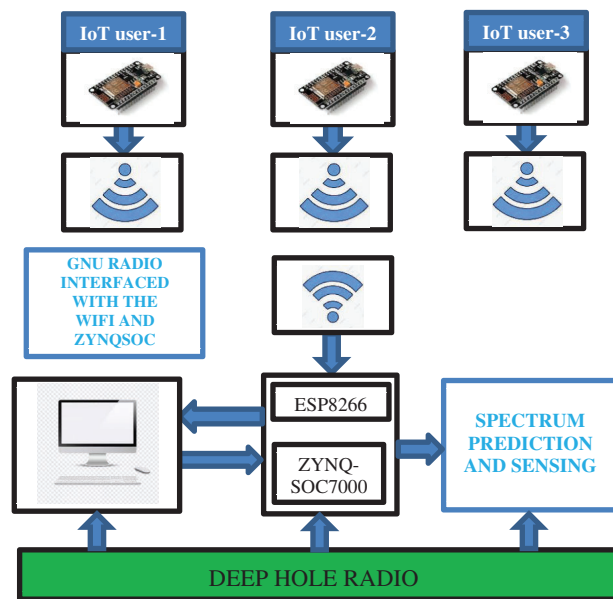


Figure 1: System setup for the proposed architecture

On receiver side, the proposed DEEP HOLE Application Peripheral Interface (API) radio runs the deep learning model interfaced with GNU Radio by taking the received signals as input and determining the spectrum prediction and detection. The novelty in the system setup is running optimized deep learning models on the FPGA at the receiver end. The received signals are preprocessed by the GNURadio, which calculates the signal features and sends them to the FPGA boards for a real-time decision.

3.2 Dataset Collection

Datasets contain WIFI signals with noises and noise-only samples. Since deep learning algorithms are supervised learning techniques, data labelling is required to distinguish the different categories of users. The user transmits the signals by adjusting the signal power, distance and SNR at every iteration presented in Fig. 2. Dataset generation is presented in Tab. 1.

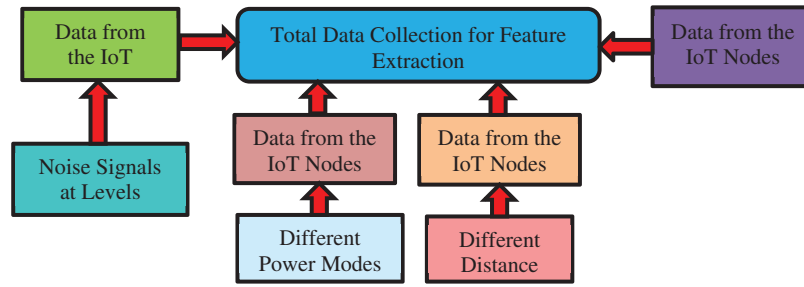


Figure 2: Data set collection used as the inputs for the deep hole radio systems

Table 1: Algorithm for dataset collection and generation

Algorithm 1 for dataset generation	
01	No of iterations N = 1000
02	SNR = -5 to 20 db
03	For n = 0: N-1
04	// Transmitting the WIFI signals At QAM modulations//
05	For SNR = 5 to 20
06	Add the Noise Level
07	// Transmitting the WIFI signals with Noise Signals without user ID At QAM Modulations//
07	End go to step 03
08	End
09	End

3.3 Feature Vectors

After receiving signals from the different users at different SNR iterations, features such as Energy, distance and User ID are calculated by GNURADIO using the python API. Generally, two substantial clients will be named primary and secondary for spectrum usage and management. During e-health data transmission, both clients were initiated for spectrum usage [27]. The total energy expended for every transmission by all users is given in Eq. (1). We presume the initial energy consumption of each operator is maximum, and the energy utilization of the operator varies and depends on the number of data transmitted.

$$E_x^{i,j} = J_{j,i} / D_{j,i} (\text{Sec}) \tag{1}$$

where “ $J_{j,i}$ ” and “ $D_{j,i}$ ” represents the throughput and channel capacity. The total energy mentioned in Eq. (1) utilized during each data broadcast corresponds to the transfer rate and assigned power. Therefore, energy expenditure of the user for each iteration is calculated by

$$E^X(\text{data}) = N_{max} X E_x^{i,j} \tag{2}$$

$$TE^X(\text{total}) = ET^X(\text{data}) + E_x^{i,j} + E_{idle} \tag{3}$$

Eq. (3) estimates the total energy utilized during transmission. In this paper, the adjustable energy consumption model is created based on the combination of idle and dynamic utilization under different Signal to noise ratios (SNR) and distances.

$$TET^X(total) = [ET^X(data) + E_x^{i,j} + E_{idle}] * dist \quad (4)$$

After calculating the energy vectors from the received signals, the distance between the IoT users and DeepHole Radio is calculated. As mentioned in [28,29], Received Signal Strength (RSSI) is taken as the main parameter to calculate the distance. The mathematical expressions are given as follows as

$$Dist_{(N, Bs)} = 10 \left[\frac{\rho_o - \sigma_m - \rho_r - 10n \log(f) + 30n - 32.44}{10n} \right] \quad (5)$$

where “ ρ_o ” is the power of the signal (dBm) in the zero distance, “ ρ_r ” is the Signal power (dBm) in the distance d, “ f ” is the signal frequency in MHz, “ σ_m ” is the Fade margin, and “ n ” is the path-loss exponent.

3.4 Optimized Deep Learning Model

This section discusses the motivation behind the proposed optimized algorithm for hardware porting in FPGA.

3.4.1 Motivation

As discussed in [30], Long Term Short Memory (LSTM) has several drawbacks when using large datasets. It leads to the usage of many memory cells, which makes it unsuitable for the resource constraint hardware. Also, there might be a chance of an overfitting issue directly connected to the network stability. There is a need to have a computationally efficient model that can predict the different categories of features used for detecting the spectrum according to the users to overcome this drawback. A more advanced, simple, efficient and highly accurate model is required to satisfy the above criteria. The current study aims at developing the new hybrid algorithm by integrating the Firefly algorithms [25] in LSTM networks.

3.4.2 Optimized LSTM Models

The simple firefly algorithm is used to optimize the weights of LSTM networks. Initially, a random number of weights and biases are passed to the LSTM cells. The accuracy of the proposed model is coined as the fitness function. For each iteration, input bias and weights are calculated which are then fed to the LSTM network and then fitness functions are calculated. The iteration is continued until the fitness function reaches the maximum accuracy or else it will be stopped. In this method, firefly provides a lower convergence speed, improved detection time and requires less time for optimization when compared with the other meta-heuristic algorithms. This optimized learning model consumes less training time and bias weights, making them suitable for the embedded systems’ hardware constraints.

3.5 DEEP HOLE RADIO Architecture

Fig. 3 depicts the overview of the architecture of the proposed Deep Hole Radio system. Together with the GNURDIO interface (Software) and the Accelerators (Hardware), the proposed system reforms the full-fledged reconfigurable software-defined radio architecture where the proposed system entirely does learning according to the environment. We briefly introduce the System on Chip Architecture in the preceding section and then describe each component of the proposed system.

3.5.1 DEEP HOLE RADIO SoC Architecture

For the proposed architecture we adopted the SoCs because of their low power consumption and customized hardware designs. SoC offers flexibility to the proposed architecture as Programmable Logic (PL) portions. All the PL can be controlled and managed by the processing system [31].

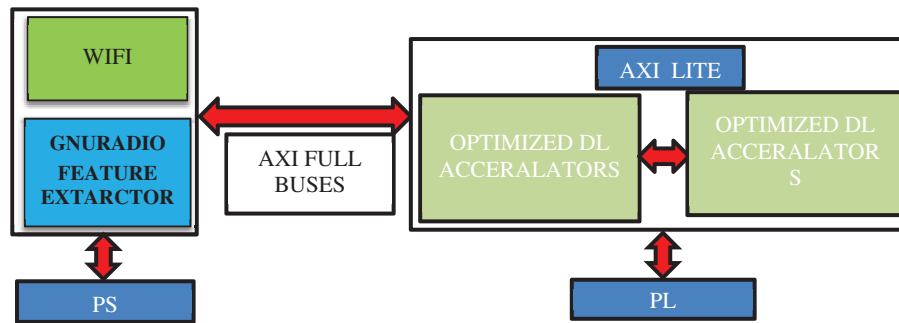


Figure 3: Overall hardware architecture for implementing deep hole radio systems

The proposed DeepHole Radio uses the three categories of Advanced Extensible Interface (AXI) bus specifications such as AXI-Lite, AXI-Stream, and AXI-Full, for the following purposes

1. To exchange the data between the PL's blocks (AXI-Lite)
2. Interchange of data between the PL and PS (AXI-Full)
3. Exchange of data between the PS. (AXI Streaml)

The proposed system uses Direct Memory Access (DMA) along with the AXI-Full buses to allow PL circuits for the read/write operations via AXI-Stream to the RAM residing in the PS. DMA is highly essential since the CPU is responsible for read/write operation and thus unavailable to perform other work. Since GNURADIO resides on the CPU (PS), parallel feature extraction and feeding to the PL take place simultaneously using DMA. This feature helps maintain a stable data transfer rate between PL and PS. The proposed system architecture resides both in the PS and PL. The ultimate issue addressed by the PS of the proposed system is to drive the components in PL without any latency. Hence the PS can run either on the operating system or in bare metal mode. In the proposed architecture, the Feature extractor and GNURadio have been installed on the operating system's top layer. The extracted features are stored in the separate memory and then fed to the PS, which runs the optimized deep learning models. Fig. 3 depicts the architecture of PL for the proposed feature extractor with GNUradio interfaced.

3.5.2 Hardware Architecture of PL Cores

The PL core aim is to deliver the architecture of the proposed optimized learning architecture, which continuously receives the features from the unit. The circuit used for constructing the proposed deep learning architecture is illustrated in Fig. 4. The inputs to the core are ((i) Features collected from the PS cores and (ii) Parameter tuning (weights and layers) of optimized deep learning models. Since the core needs to access these parameters in various time instants, both are stored in the Pipelined Distributed Look Up Tables (PDLUT) for area reduction and high speed. Thus, the FIFO buffer is triggered by the core so that it can send the features to PDLUTS via AXI-Stream. AXI-Full is utilized for the transactions between the core and PDLUTS. Each layer presents the following structure: (i) receives the input from a PDLUTS, (ii) process the input based on optimized layers (iii) writing respective results on the PDLUTS of the following layer. The significant advantages offered by this architecture are (i) Scalability, (ii) configurability and (iii) modularity.

3.5.3 DEEP HOLE RADIO–A Design Framework

The fundamental challenge addressed by the proposed architecture in PL is transitioning from a software-based deep learning implementation to a hardware-based one. High-Level Synthesis (HLS) for the core design has been employed to implement the deep learning architecture to have an effective

mechanism. The HLS approach which is used in the design of the proposed architecture is explained as follows

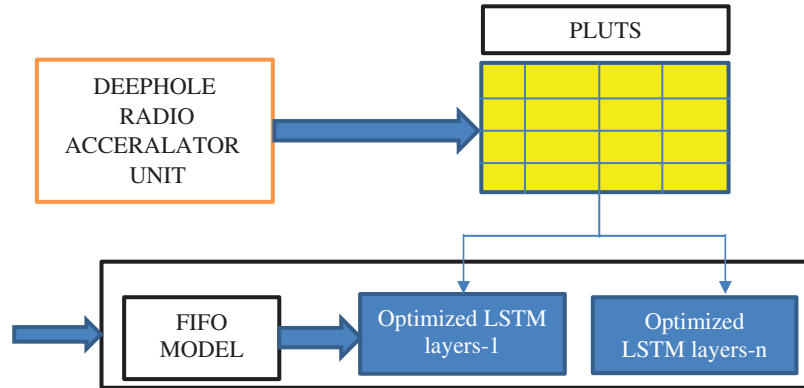


Figure 4: PDLUTS Implementation for the deephole radio systems

High-Level Synthesis

The digital circuits used for designing the proposed architecture have several benefits over the traditional methods. In HLS, C/C++/python languages can be adopted for the implementation. This paper adopts python for designing the proposed architectures. This approach will reduce the developer's designing time and increase efficiency since the HLS program optimizes the circuit according to the system specification. Moreover, parallelizing the circuit operation is more crucial for a faster response. Hence the HLS toolchain can tell how many cycles are needed for a circuit to generate all the outputs for a given input size, given a target parallelization level. It helps the proposed architecture to make the best trade-off between hardware complexity and latency. Also, the loop pipelining is incorporated to allow the operations in a loop to be implemented parallelly.

4 Experimentation

Spectrums of the WIFI signals are created in Python, which is then connected to the Zynq-SOC 7000 series using the UART (Universal Asynchronous Receiver Transmitter) to evaluate the performance of the proposed architecture on the real-world IoT devices. The specification of the test bed used for the experimentation is tabulated in [Tab. 2](#).

Table 2: Specification of the parameters used for experimentation

Sl.no	Experimentation parameters	Specifications
01	Soc type	Xilinx Zynq-7000 XC7Z045-system-on-chip (SoC)
02	PS type	Dual ARM-9 Core
03	PL	Artix-7 FPGA
04	WIFI Transceivers used	ESP8266
05	No of users used	5-10

We have used the NODEMCU mounted with MCP3008 ADC for the transmitter side and interfaced with the four different sensors. The NodeMCU uses the 16-QAM type of modulation type for transmitted

signals. The receiver test bed is designed based on the above specifications mentioned in Tab. 2. Fig. 6 shows the validation curves of the proposed deep learning under different no. of epochs.

4.1 Proposed Deep Learning Model

4.1.1 Dataset Collection

To evaluate the proposed architecture, we have collected nearly different signals at different Signal to Noise ratios (SNR) at various time intervals. Nearly the number of users is increased from 5 to 50, and signals are captured using the testbed which runs the proposed architecture. Nearly 2,40,567 feature data were used for training the proposed architecture. The performance metrics used to evaluate the proposed architecture are given as follows

$$\text{Accuracy of Sensing} = \frac{\text{DR}}{\text{TNI}} \times 100 \quad (6)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{TN}} \times 100 \quad (7)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TP} + \text{TN}} \times 100 \quad (8)$$

where “TP and TN Represent True Positive and True Negative values and

DR & TNI Represents Number of Detected Results and the Total number of Iterations”. For evaluating the performance of the proposed algorithm, nearly 70% of data were taken for training, and 30% were taken for testing.

4.1.2 Results and Discussion

It is found that the best results in the tuning process were optimized to 115 epochs, with 0.0001 learning rate and output batch size set to 80. Fig. 5 shows the detection exactness and error rate in training with a different scenario of testing data used for validation. Also performance of the algorithm has been tested with the different users with IoT devices.

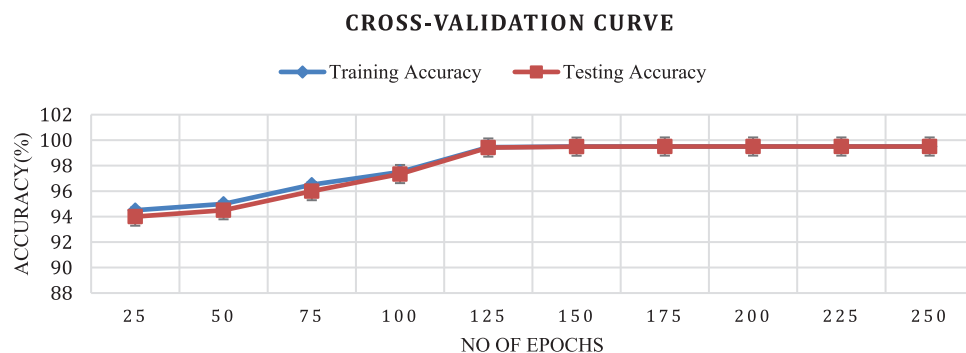


Figure 5: Validation curve for the proposed deep learning model with the No of epochs optimized using different users

Fig. 5 shows the validation curve for the proposed resource constraint deep learning models. It is found that no of epochs is fixed to 126, and accuracy is maintained high and constant at 99.5%. The performance of the proposed algorithm has been validated for the 50 users. Moreover, the confusion matrix for the optimized and non-optimized deep learning network has been shown in Figs. 6 and 7. It is clearly shown that the optimized learning model has produced the constituent accuracy of 99.5% even though the number of

users increases. However, Fig. 7 shows the decrease in performance of the non-optimized learning models as the number of users increases. From the above Figures, it is clear that the stability of the network is maintained in the optimized learning model even though the number of users increases.

No of users	10	20	30	40	50
10	99.5	0.5			
20	0.5	99.5			
30		0.5	99.5		
40	0.5			99.5	
50			0.5		99.5

Figure 6: Confusion matrix for the proposed deep learning model using different users

No of users	10	20	30	40	50
10	99.5	0.5			
20	1	99.0			
30		2.5	97.5		
40	4.5			95.5	
50			5.75		94.25

Figure 7: Confusion matrix for the deep learning model (without optimization) using different users

Fig. 8 shows that the validation curve for the proposed algorithm with testing and training accuracy is close to 99.5%. Moreover, different performance metrics such as accuracy, sensitivity, and specificity have been calculated and compared with other existing deep learning models such as Long Short Term Memory (LSTM) [32], Convolutional neural network (CNN) [33], and Deep Neural Networks (DNN) [34] used of the spectrum depends on the number of users. Fig. 8 shows the sensing accuracy for the different deep learning models in which the proposed optimized deep learning model has shown a high sensing accuracy of 99.5%. The accuracy of other algorithms, LSTM, CNN and DNN, has 94.5%, 92.0% and 89%, respectively. However, the existing algorithms have shown a dip in their sensing accuracy, ranging from 10%–15% as the number of users increases. However, the usage of the optimization algorithm over the deep learning models maintains the stable and constant accuracy of 99.5%, though the number of users increases. A similar fashion is observed in sensitivity and specificity performance depicted in Figs. 9 and 10, respectively.

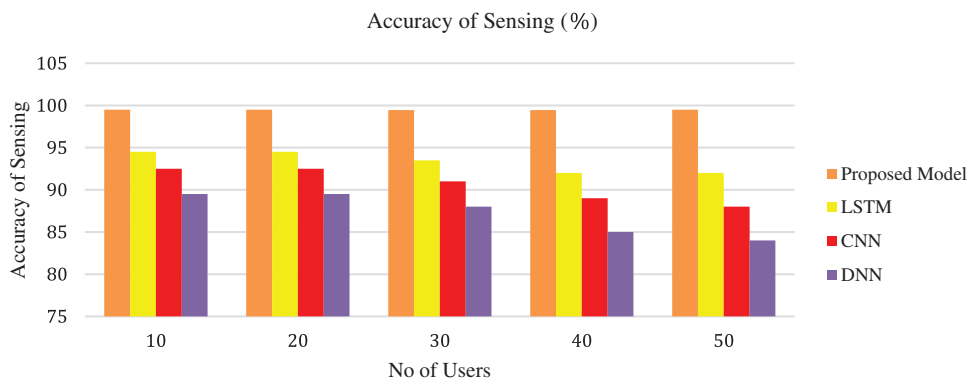


Figure 8: Comparative analysis for the accuracy of sensing using the different deep learning models

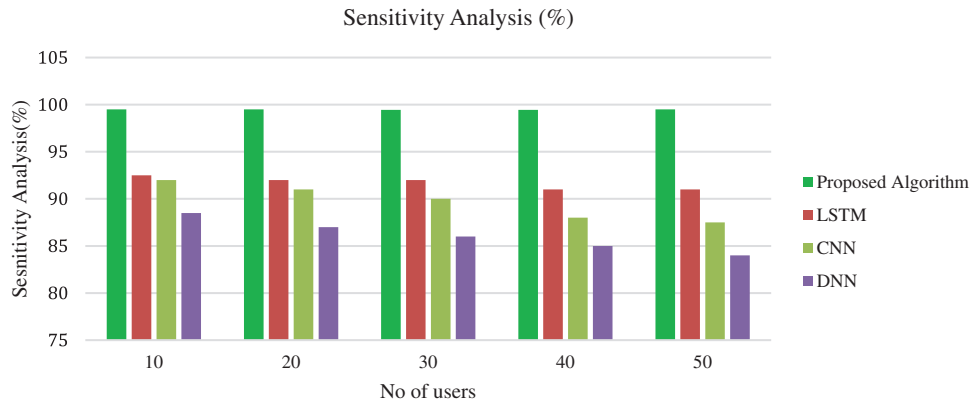


Figure 9: Comparative analysis of the sensitivity using the different deep learning models

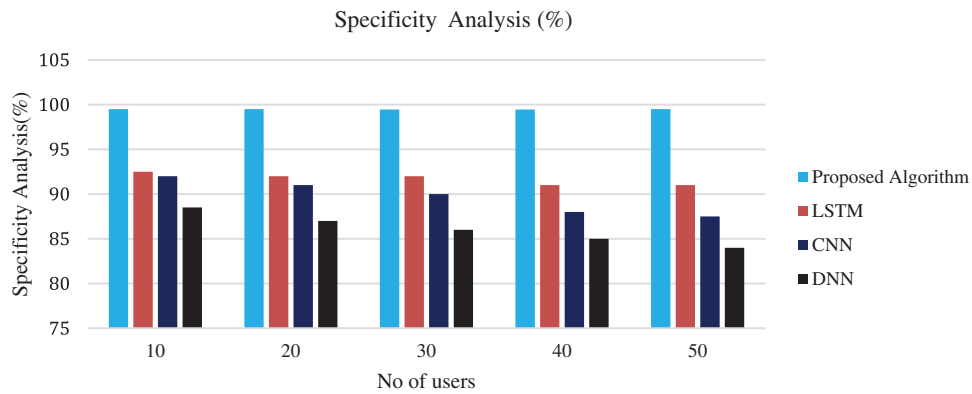


Figure 10: Comparative analysis for the specificity using the different deep learning models

To further investigate the impact of the proposed algorithm on the hardware, we have synthesized the proposed architecture on the Xilinx VIVADO, and the following parameters, such as area of utilization, power and latency, have been calculated represented in [Tabs. 3 and 4](#).

Table 3: Area of utilization for the proposed deephole radio model

No of algorithm	No of specifications							
	No of LUTS		BRAM		DSP		Flip flops	
Proposed model	Available	Used	Available	Used	Available	Used	Available	Used
	21378	210	241534	190	2200	0	245450	24500

Table 4: Power and latency of the proposed model with no of users

No of algorithms	Power (mW)	Latency (ms)
Proposed model	1.100	5.4

Moreover, to validate the proposed model, we have compared the hardware performance of the proposed deep learning architecture with the other existing architecture, such as RFlearn and other software-based mechanisms presented in Fig. 11, Tabs. 5 and 6.

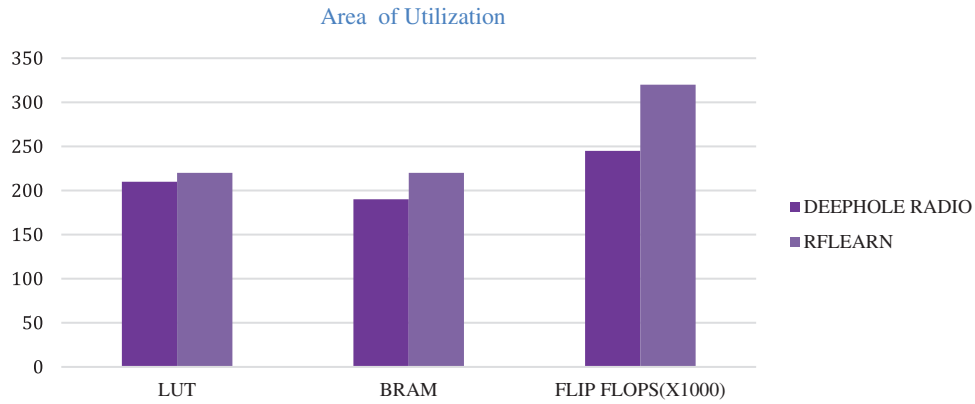


Figure 11: Comparative analysis of area utilization between the deephole radio and FPGA architectures

Table 5: Power analysis between the proposed deephole radio and RFlearn architecture

No of algorithms	Power (mW)
RFlearn	1.200
Deep hole radio	1.100

Table 6: Latency analysis between the proposed deephole radio and Rflearn architecture

No of algorithms	Latency (ms)
RFlearn	6.589
Deep hole radio	5.400

5 Conclusion and Future Challenges

In this paper, we have proposed a DEEP HOLE radio system. This novel approach integrates the optimized deep learning models for spectrum-driven decisions through real-time hardware-based IoT devices. The complete hardware design for the DEEP HOLE Radio framework is presented with the new Pipelined Distributed Look Up Tables (PDLUT) architecture in the place of traditional BRAMS. Finally, experimental results show that the proposed firefly optimized LSTM network produced an accuracy of 99.5% even though users get increased. Moreover, we have compared the existing learning models used for spectrum-enabled decisions. The proposed model has edged the other learning models in terms of a 10%–15% increase in performance. RFLEARN type architecture is used for the comparison in which the introduction of PDLUTS in the architecture has produced a 15% reduction in delay, 20% less power consumption and 25% area reduction than RFLEARN architectures to prove the hardware efficiency. In the Future scope, architecture needs improvisation in terms of implementing the hardware-based deep learning models along with the spectrum sensing and management system.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

- [1] P. Cai and Y. Zhang, "Intelligent cognitive spectrum collaboration: Convergence of spectrum sensing, spectrum access, and coding technology," *Intelligent and Converged Networks*, vol. 1, no. 1, pp. 79–98, 2020.
- [2] H. Ye, G. Y. Li and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.
- [3] H. Shokri-Ghadikolaei, F. Boccardi, C. Fischione, G. Fodor and M. Zorzi, "Spectrum sharing in mmwave cellular networks via cell association, coordination, and beamforming," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 11, pp. 2902–2917, 2016.
- [4] M. A. Vazquez, L. Blanco and A. I. Perez-Neira, "Hybrid analog–digital transmit beamforming for spectrum sharing backhaul networks," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2273–2285, 2018.
- [5] L. Lv, J. Chen, Q. Ni, Z. Ding and H. Jiang, "Cognitive non-orthogonal multiple access with cooperative relaying: A new wireless frontier for 5 g spectrum sharing," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 188–195, 2018.
- [6] X. Jin, J. Sun, R. Zhang, Y. Zhang and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2925–2938, 2018. <https://doi.org/10.1109/TMC.2018.2823314>.
- [7] T. M. Chiwewe and G. P. Hancke, "Fast convergence cooperative dynamic spectrum access for cognitive radio networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3386–3394, 2018.
- [8] R. Liu, Y. Ma, X. Zhang and Y. Gao, "Deep learning-based spectrum sensing in space-air-ground integrated networks," *Journal of Communications and Information Networks*, vol. 6, no. 1, pp. 82–90, 2021.
- [9] S. Agarwal and S. De, "Edsa: Energy-efficient dynamic spectrum access protocols for cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 3057–3071, 2016.
- [10] L. Zhang, F. Restuccia, T. Melodia and S. Pudlewski, "Learning to detect and mitigate cross-layer attacks in wireless networks: Framework and applications," in *IEEE Conf. on Communications and Network Security*, Las Vegas, NV, USA, pp. 1–9, 2017. <https://doi.org/10.1109/CNS.2017.8228631>.
- [11] J. Huang, G. Chang and J. Huang, "Anti-jamming rendezvous scheme for cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 648–661, 2017.
- [12] G. Chang, S. Wang and Y. Liu, "A Jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6712–6725, 2017.
- [13] M. Chen, U. Challita, W. Saad, C. Yin and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019. <https://doi.org/10.1109/COMST.2019.2926625>.
- [14] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu *et al.*, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *IEEE Int. Conf. on Communications Workshops*, Kansas City, MO, USA, pp. 1–6, 2018.
- [15] J. Lotze, S. A. Fahmy, J. Noguera, L. Doyle and R. Esser, "An fpga-based cognitive radio framework," in *IET Irish Signals and Systems Conf.*, Galway, pp. 138–143, 2008.
- [16] M. V. Hatwar and S. R. Bichwe, "Fpga implementation of cognitive engine using parallel processing," *International Journal of Scientific Engineering and Applied Science*, vol. 1, no. 3, pp. 1–6, 2015.
- [17] T. ZhiLing, S. Min, M. Yu and L. Juan, "Implementation of deep learning-based automatic modulation classifier on fpga sdr platform," *Electronics Journal*, vol. 7, no. 122, pp. 1–16, 2018. <https://doi.org/10.3390/electronics7070122>.
- [18] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng *et al.*, "Real-time experimentation of deep learning-based rf signal classifier on fpga," in *IEEE Int. Symp. on Dynamic Spectrum Access Networks*, Newark, NJ, USA, pp. 1–2, 2019. <https://doi.org/10.1109/DySPAN.2019.8935784>.

- [19] A. Maclellan, L. McLaughlin, L. Crockett and R. Stewart, "Fpga accelerated deep learning radio modulation classification using matlab system objects & pynq," in *Int. Conf. on Field Programmable Logic and Applications (FPL)*, Barcelona, Spain, pp. 246–247, 2019, <https://doi.org/10.1109/FPL.2019.00045>.
- [20] G. J. Mendis, J. Wei and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *IEEE Int. Conf. on Communication Systems (ICCS)*, Shenzhen, China, pp. 1–6, 2016. <https://doi.org/10.1109/ICCS.2016.7833571>.
- [21] J. Gao, X. Yi, C. Zhong, X. Chen and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727–1730, 2019.
- [22] S. Zheng, S. Chen, P. Qi, H. Zhou and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Communications*, vol. 17, no. 2, pp. 138–148, 2020.
- [23] R. Sarikhani and F. Keynia, "Cooperative spectrum sensing meets machine learning: Deep reinforcement learning approach," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1459–1462, 2020.
- [24] J. Xie, J. Fang, C. Liu and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5307–5319, 2020.
- [25] S. Zheng, L. Ye, X. Wang, J. Chen, H. Zhou *et al.*, "Primary user adversarial attacks on deep learning-based spectrum sensing and the defense method," *China Communications*, vol. 18, no. 12, pp. 94–107, 2021.
- [26] H. Xing, H. Qin, S. Luo, P. Dai, L. Xu *et al.*, "Spectrum sensing in cognitive radio: A deep learning based model," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 1, pp. e4388, 2022. <https://doi.org/10.1002/ett.4388>.
- [27] Y. Ren, P. Dmochowski and P. Komisarczuk, "Analysis and implementation of reinforcement learning on a gnu radio cognitive radio platform," in *Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications*, Cannes, France, pp. 1–6, 2010.
- [28] S. Tamizhelakkiya, U. Kumar, D. V. Simha and P. P. Sahu, "Design and implementation of fpga based configurable ai architecture with deep learning algorithm," *Journal of Physics*, vol. 1362, pp. 1–6, 2019. <https://doi.org/10.1088/1742-6596/1362/1/012046>.
- [29] M. E. Tarhuni, K. Assaleh and F. Kiftaro, "Implementation of machine learning spectrum sensing for cognitive radio applications," *Computer Science & Information Technology (CS & IT)*, vol. 9, no. 4, pp. 43–52, 2019. <https://doi.org/10.5121/csit.2019.90405>.
- [30] J. Xie, J. Fang, C. Liu and X. Li, "Deep learning-based spectrum sensing in cognitive radio: acnn-lstm approach," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2196–2200, 2020.
- [31] H. Sadeeq and A. M. Abdulazeez, "Hardware implementation of firefly optimization algorithm using fpgas," in *Int. Conf. on Advanced Science and Engineering*, Duhok, Iraq, pp. 30–35, 2018. <https://doi.org/10.1109/ICOASE.2018.8548822>.
- [32] B. Soni, D. K. Patel and M. López-Benítez, "Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics," *IEEE Access*, vol. 8, pp. 97437–97451, 2020. <https://doi.org/10.1109/ACCESS.2020.2995633>.
- [33] J. Xie, J. Fang, C. Liu and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2196–2200, 2020.
- [34] Y. Zhang and Z. Zhao, "Limited data spectrum sensing based on semi-supervised deep neural network," *IEEE Access*, vol. 9, pp. 166423–166435, 2021.