



A Cognitive Radio Spectrum Sensing Implementation Based on Deep Learning and Real Signals

Mohamed Saber^{1,2(✉)}, Abdellah Chehri³, Abdessamad El Rharras¹, Rachid Saadane¹, and Mohammed Wahbi¹

¹ SIRC-LaGeS, Hassania School of Public Works, Casablanca, Morocco
saber.smhd@gmail.com, saadane@ehp.ac.ma

² Computer Science Research Lab, Ibn Tofail University, Kenitra, Morocco

³ Department of Applied Sciences, University of Quebec, Chicoutimi, QC G7H 2B1, Canada
achehri@uqac.ca

Abstract. In a cognitive radio environment, spectrum sensing is an essential phase for improving spectrum resources management. Based on a deep learning method and real signals, a new spectrum sensing implementation is proposed in this work. The real signals are artificially generated, using an ARDUINO UNO card and a 433 MHz wireless transmitter, in ASK and FSK modulation types. The reception interface is constructed using an RTL-SDR receiver connected to MATLAB software. The signals classification is carried out by a convolutional neural network (CNN) classifier. Our proposed model's main objective is to identify the spectrum state (free or occupied) by classifying the received signals into a licensed user (primary user) signals or noise signals. Our proposed model's performance evaluation is evaluated by two metrics: the probability of detection (Pd) and the false alarm probability (PFA). Finally, the proposed sensing method is compared with other used techniques for signal classification, such as energy detection, artificial neural network, and support vector machine. The experimental results show that CNN could classify the real signals better than traditional methods and machine learning methods.

Keywords: Cognitive radio network · Spectrum sensing · CNN · RTL-SDR · ASK-FSK signals

1 Introduction

In recent decades, the demand for spectrum resources has augmented rapidly with the fast development of wireless communication services and the propagation of the Internet of Things (IoT). The traditional static allocation of spectrum causes numerous challenges to the recent wireless industry. Since IoT enables many autonomous devices to share information, spectrum scarcity is one of the main challenges due to the limited natural radio spectrum resource that helps wireless communications [1]. The statistics of spectrum allocation around the world show that the radio spectrum has been almost fully

allocated, and compared to the demands of high data-rate devices, it was evident that the current fixed spectrum allocation leads to an inefficient spectrum utilization [2, 3]. Additionally, some frequencies bands are not used all the time by their owners, and others are highly utilized. The unused frequencies are termed as white spaces or spectrum holes. A spectrum hole is a spectrum band assigned to a licensed user, also known as primary user (PU), but it is momentarily unoccupied by this PU. Therefore, due to this static allocation and user-needs growth, the radio spectrum is inefficiently exploited while a secondary utilization is possible.

The underutilization of the allocated spectrum, a high-efficient spectrum sharing, and access are recommended to exploit the radio spectrum effectively. In this context, Cognitive Radio (CR) Networks is introduced in [4, 5] as an emerging paradigm that ensures efficient spectrum allocation to address spectrum scarcity. In CR networks, the unlicensed users, also known as secondary users (SUs), can access the spectrum of authorized users (PUs) if the primary spectrum is idle or simultaneously share PUs long as the PUs services can be completely protected. By doing this, the SUs can gain transmission opportunities without requiring a dedicated spectrum. Therefore, CR is considered as a critical enabling technology for dynamic spectrum access (DSA) [6]. To enable the DSA, a cognitive user (SU) needs to obtain the correct information of the spectrum (Free or Occupied), so that the quality of services (QoS) of the PUs can be guaranteed. Two methods can be adopted by SUs to detect spectrum holes. The first is the *Geolocation Database* that can be used when the PU's activity is regular and highly predictable. The second is *Spectrum Sensing*, in which the SU can carry out spectrum sensing, periodically or consistently, to observe the primary spectrum and detect the spectrum holes. Figure 1 illustrates a typical cognitive cycle for a CR, as introduced by Haykin in [7]. Although enabling DSA with CR is a technical issue that involves multidisciplinary efforts from various research communities, such as signal processing, information theory, communications, computer networking, and machine learning.

As shown in the above figure, the CR cycle's essential phase is spectrum sensing operation (radio-scene analysis). Its objective is to sense the spectrum holes to obtain the band state (free/occupied), and based on this result. Data transmission is adapted to optimize the performance by maximizing the throughput and minimizing the interference. Different spectrum sensing techniques have been discussed in the literature [8, 9], such as the cyclostationarity feature detection technique [10], Energy detection techniques [11], and Matched filter detection method [10]. Efficient spectrum sensing can be achieved by adopting strategies that can classify signals, with high accuracy, to PU signals or noise signals. Deep learning has recently gained popularity in signal processing into applications involving signals/time-series data such as voice assistants, digital health, radar, and wireless communications. In our context, we propose a deep learning model for spectrum sensing operation, in which we use real-world signals that represent the PU signals. The evaluated performance of our proposed model is compared with our previous proposed classifiers in [12].

The remainder of this paper is organized as follows. In Sect. 2, we describe the proposed system model for spectrum sensing, and we present the generation and acquisition of the used database. Section 3 gives more detailed descriptions of the system model

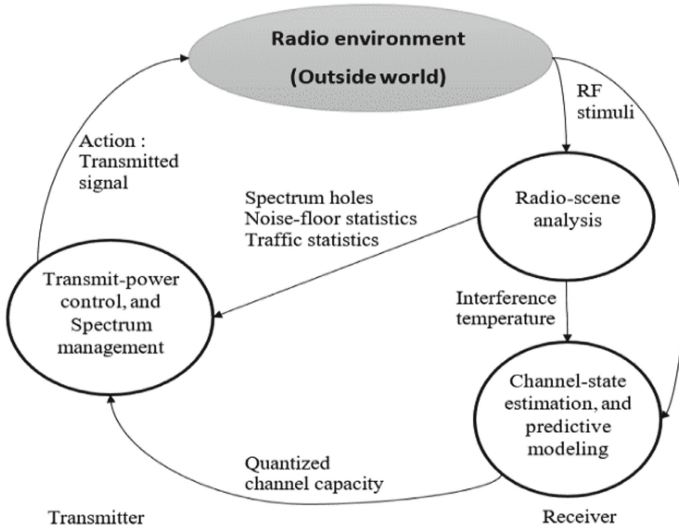


Fig. 1. Basic cognitive cycle (The figure focuses on three fundamental cognitive tasks).

implementation and presents the results and discussions. Finally, Sect. 5 concludes the paper.

2 System Model

In this paper, we consider a spectrum sensing model that classifies the received signals from a single antenna (RTL-SDR device) to PU signals or Noise signals. Our proposed platform can operate in a dynamic spectrum access environment to identify the spectrum holes to attain higher spectrum utilization. With Matlab and Python 3 language, we can develop a deep learning model and build real-world smart signal processing systems. In general, there are four steps involved. The first step in building a deep learning model is to access and manage the data. Using Matlab, we can acquire signals from a variety of hardware devices. We can also generate synthetic data via simulation or use data augmentation techniques. We proposed a model that transmits real ASK/FSK signals by using the Arduino Uno card and the 433 MHz transmitter. The transmitted signals are collected by an RTL-SDR device connected to the Matlab environment. The received signals are processed as features and fed into a CNN classifier developed in Python language to decide the PU availability.

Once the data is collected and ready, we interpret the signal data and label it. We visualize and analyze the received signals with attribute regions. We use the domain-specific tools to label all our signals to prepare our data for training. There are two main approaches to performing deep learning on signals. The first approach involves converting signals into time-frequency representations and training custom convolutional neural networks to extract patterns directly from these representations. A time-frequency presentation is a view of a signal represented over both time and frequency. More specifically, it describes how spectral components in signals evolve as a function of time [13].

This approach enhances the patterns that may not be visible in the original signal. There are various techniques generating time-frequency representation from signals and saving it as images, including spectrogram, continuous wavelet transforms (or scalograms), and constant Q transforms.

The second approach involves feeding signals directly into deep networks, such as Long Short-Term Memory (LSTM) networks. To make the deep network learn the patterns more quickly, a reduction of the signal dimensionality and variability is needed in this approach. We have two options, we can manually identify and extract features from signals, or we can automatically extract features using invariant scattering convolutional networks, which provide low variance representations without losing critical information. We adopt the first approach for our signals in our work, and we follow the described steps in Fig. 2 for training and testing the proposed deep network.

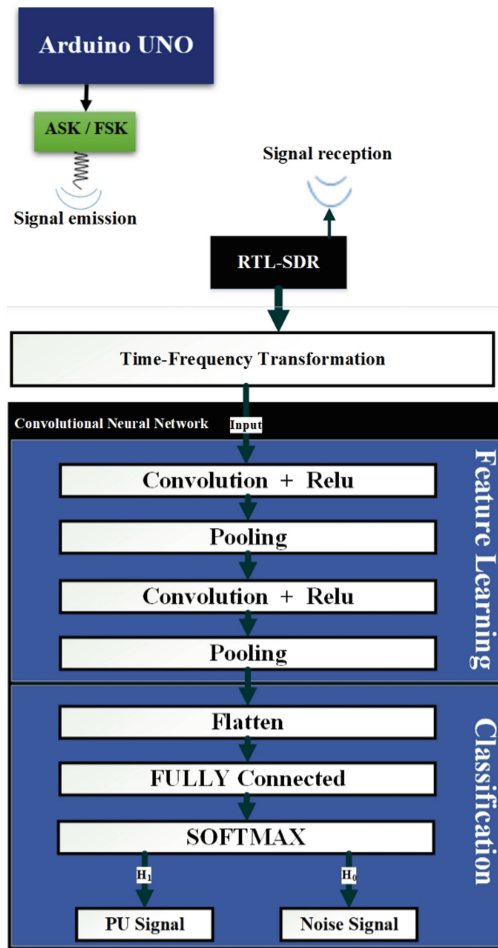


Fig. 2. The proposed model.

2.1 Database Generation

Deep learning is a part of machine learning methods that, based on artificial neural networks, enables computers to learn from experience and understand the world in terms of a hierarchy of concepts. Generally, learning can be supervised, semi-supervised, or unsupervised [14, 15]. In our context, we have supervised learning. Supervised learning methods are generally composed of two main phases: training/learning and classification. To make our work more independent and realistic, we will create our signals data. For these reasons, we have constructed a real signal-emitter by using an Arduino Uno card and a 433 MHz wireless device that transmits signals in ASK/FSK modulation type.

The *Arduino Uno* is a microcontroller board [16] used to construct and program electronics projects. It is flexible, inexpensive, and can present a diversity of digital and analog inputs. Arduino Uno, enabled by a software program (Simplified C ++), can send and receive information to most devices, in our case, the 433 MHz ASK/FSK, and even though the internet to control the specific electronic devices. The 433 MHz ASK/FSK is an electronic module that operates at 434 MHz to transmit radio signals. It is commonly used in wireless data transfer applications such as mobile robots and remote control. It receives programmed serial data from a specific platform and transmits it wirelessly over the radio frequency via its antenna connected at pin4 of the Arduino Uno.

2.2 Database Acquisition

Spectrum sensing is based on a famous technique called signal detection, which allow us to determine whether a given frequency band is being used. In our work, we consider an SU spectrum sensor with 1 antenna (represented by the RTL-SDR in Fig. 2). Mathematically, signal detection can be reduced to a simple identification problem, formalized as a hypothesis test [17, 18]. There are two hypotheses: H_0 , the PU is inactive; and H_1 , the PU is active. The received signal at the antenna is given by:

$$y(k) = \begin{cases} n(k) & H_0 \\ s(k) + n(k) & H_1 \end{cases} \quad (1)$$

Where $y(k)$ is the sample, the detected signal in the licensed channel, to be analyzed at each instant k , and $n(k)$ is the received noise plus possible interference (not necessarily white Gaussian noise) of variance σ^2 . At hypothesis H_1 , $s(k)$ represents the signal that the network wants to detect, which is the transmitted signal through the wireless channel. Generally, the receiver reconstructs the message sent from the captured signal by the inverted processing operations done on transmission. Those processes can be performed using a single RTL-SDR hardware and MATLAB software. RTL-SDR or Software Defined Radio is a radio communication system, wherein the traditional hardware components (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on an embedded system [19–27]. RTL-SDR is capable of receiving any signal in its frequency range. This range varies depending on the type of device used. In this work, the used dongle has a frequency capability of approximately 25 MHz–1750 MHz. The RTL-SDR software provides the same functions of a traditional receiver: Selection of the desired frequency and bandwidth, Demodulation of the received signal and Interference suppression (Noise blanker).

Figure 3 shows the block diagram of the different processing stages of an RTL-SDR dongle.

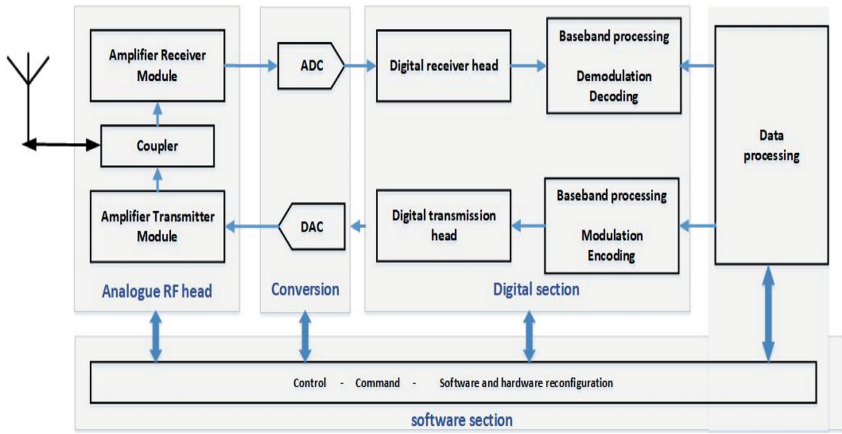


Fig. 3. Synoptic diagram of SDR card.

3 Implementation and Results

3.1 Description of the Used Data

In the proposed sensing model, the used database is collected in the SIRC/LAGES laboratory of HASSANIA school of public works, for different distances between the transmitter and the receiver. It consists of 2000 ASK/FSK signals. Table 1 shows how we have organized the used database in the learning phase and test phase.

Table 1. The used Data-base in learning and testing phases

Signal	Learning phase	Test phase
Primary signal	600	500
Noise signal	400	500

This data is generated for several distances between the sender and the receiver because SUs which are close to the PU (e.g., SU1 and SU2 in Fig. 4) are probably can detect the PU presence more reliably than SUs which are far away from the PU (e.g., SU3 in Fig. 4). Moreover, SUs close to each other (SU1 and SU2 in Fig. 4) are likely to report the same sensing results.

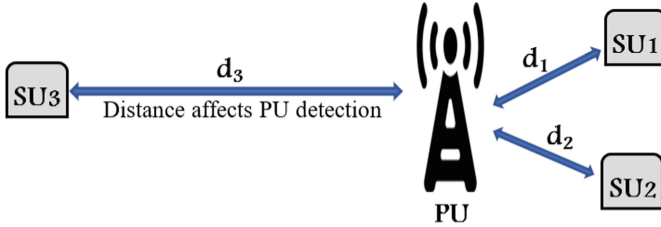


Fig. 4. The effect of distance on spectrum sensing results

3.2 Performance Evaluation

As previously mentioned, H_0 and H_1 are the sensed states for absence and presence of signal, respectively. Therefore, as presented in Fig. 5, we can define four possible scenarios in detecting a signal. The first is declaring H_0 when H_0 is true (H_0/H_0), the second is declaring H_1 when H_1 is true (H_1/H_1), the third is declaring H_0 when H_1 is true (H_0/H_1), and the fourth is declaring H_1 when H_0 is true (H_1/H_0).

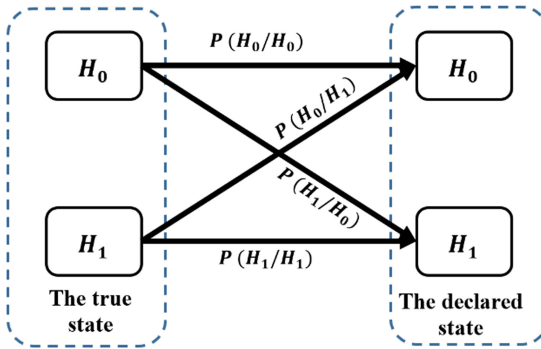


Fig. 5. Hypothesis test and possible scenarios with their corresponding probabilities.

The first and the second scenarios are known as a correct detection of a PU signal and a spectrum hole, respectively. Whereas scenarios 3 and 4 are known as a missed detection and a false alarm, respectively. The performances of our proposed spectrum sensing model are characterized by: the probability of detection P_d and the false alarm probability P_{fa} .

P_d : The probability of detecting the PU as being present when the PU is truly present (the band is occupied H_1). It is desirable to keep the detection rate as high as possible for spectrum sensing, since low P_d value (Failed detection) causes interference with the PU. We calculated P_d by (2):

$$P_d = P(\text{decision}H_1/H_1) = \frac{N_c}{N} \times 100 \tag{2}$$

P_{fa} : the probability of detecting the PU as being present (the band is occupied H_1) when the PU is actually absent (the band is not occupied H_0). It is desirable to keep the

false alarm rate as low as possible for spectrum sensing, since high P_{fa} value (Failed detection) reduces the efficiency of spectrum use. We calculated P_{fa} by (3):

$$P_{fa} = P(\text{decision}H_1/H_0) = \frac{N_e}{N} \times 100 \quad (3)$$

Where:

N_e : The number of times in which the PU presence is declared, while hypothesis H_1 ;

N_d : The number of times in which the PU signal is detected, while hypothesis H_0 ;

N : The number of all captured signals.

3.3 Results

The data acquirement part is implemented in MATLAB 9.4.0 (R2018a) for a 64-bit computer with core i5 processor, clock speed 2.4 GHz, and 8 GB RAM. The used energy detector is described in our previous work [12], whereas, for ANN, SVM and DNN detectors are implemented in python 3. For SVM detector, the used parameters are: $C = 1.0$, $\text{break_ties} = \text{False}$, $\text{cache_size} = 200$, $\text{class_weight} = \text{None}$, $\text{coef0} = 0.0$, $\text{decision_function_shape} = \text{'ovr'}$, $\text{degree} = 3$, $\text{gamma} = \text{'scale'}$, $\text{kernel} = \text{'rbf'}$, $\text{max_iter} = -1$, $\text{probability} = \text{False}$, $\text{random_state} = \text{None}$, $\text{shrinking} = \text{True}$, $\text{tol} = 0.001$, $\text{verbose} = \text{False}$. While, the used parameters for the ANN detector are: $\text{activation} = \text{'relu'}$, $\text{alpha} = 1e-05$, $\text{batch_size} = \text{'auto'}$, $\text{beta}_1 = 0.9$, $\text{beta}_2 = 0.999$, $\text{early_stopping} = \text{False}$, $\text{epsilon} = 1e-08$, $\text{hidden_layer_sizes} = (6,2)$, $\text{learning_rate} = \text{'constant'}$, $\text{learning_rate_init} = 0.001$, $\text{max_fun} = 15000$, $\text{max_iter} = 200$, $\text{momentum} = 0.9$, $\text{n_iter_no_change} = 10$, $\text{nesterovs_momentum} = \text{True}$, $\text{power_t} = 0.5$, $\text{random_state} = 1$, $\text{shuffle} = \text{True}$, $\text{solver} = \text{'lbfgs'}$, $\text{tol} = 0.0001$, $\text{validation_fraction} = 0.1$, $\text{verbose} = \text{False}$, $\text{warm_start} = \text{False}$.

For the deep learning-based detector, we have used a fully-connected network structure with three layers. Due to the high performance that can achieve, we have used the rectified linear unit activation function, referred to as ReLU, on the first two layers. While, the Sigmoid function in the output layer to ensure our network output is between 0 and 1, where 0 for H_0 and 1 for H_1 . The first hidden layer has 1024 nodes, uses the relu activation function, and expects rows of data with 1024 variables (the $\text{input_dim} = 1024$ argument). The second hidden layer has 8 nodes and uses the relu activation function. The output layer has one node and uses the sigmoid activation function. Table 2 shows the comparison between the obtained results by the proposed detection techniques.

In a radio channel, the received signal at CR devices is processed as a feature vector and fed into a classifier to decide the channel availability. The classifier categorizes each processed signal into either one of the two classes, namely, H_0 : the “channel available class” and H_1 the “channel unavailable class”. The obtained values of P_d and P_{fa} are shown in Table 2.

3.4 Discussion

The obtained results manifest the capability of the proposed classifiers in detecting the PU presence. As we can see, the artificial intelligent-based techniques are significantly

Table 2. Performance evaluation of the proposed sensing techniques.

Detection technique	P_d	P_{fa}	The classifier accuracy (%)
ED	0,961	0,0120	–
ANN	0.986	0.0011	96,7
SVM	0.977	0.0011	95,7
CNN	0.994	0.0023	100

outperforming the traditional methods (ED) on spectrum sensing. Therefore, we can obtain the spectrum state with high efficiency with machine learning and deep learning techniques. Compared to all other classifiers, the supervised CNN classifier performs well in terms of the detection probability. Its higher detection efficiency compensates for the computational complexity of this classifier. Moreover, the proposed deep learning classifier shows the most elevated and perfect sensing accuracy. This sensing accuracy is crucial in the protection of primary users and the detection of spectrum holes.

The plot of our proposed deep learning classifier’s accuracy history is represented in Fig. 6. From the classification accuracy plot, we can see that the model achieves high accuracy in a low number of epochs. On the other hand, Fig. 7 shows the classification loss curve for the training and the testing. The obtained results show that the proposed classifier improves radio signal classification accuracy and efficiency compared with other methods, including energy detection, ANN, and SVM.

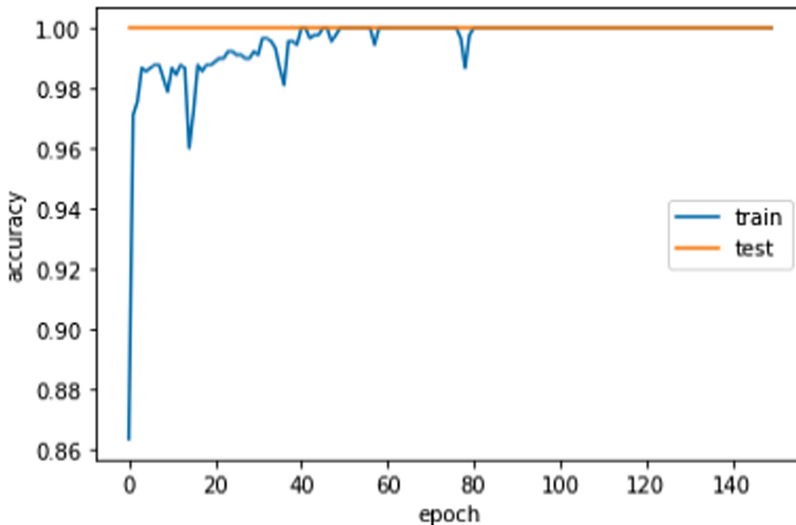


Fig. 6. Model accuracy in the training and testing phases.

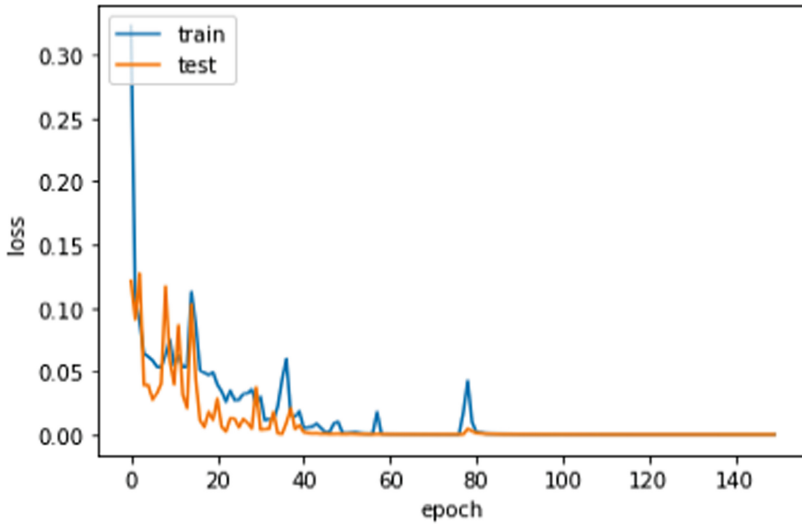


Fig. 7. Model loss in the training and testing phases

In the future we will evaluate the cognitive radio for Vehicular Ad hoc Networks (CRVs or CR-VANETS). The CR-VANETS is a new trend in the automotive market. Recent and future vehicles will offer functionalities for the transmission of intra-vehicular commands and dynamic access to wireless services, while the car is in transit [28, 29].

4 Conclusions

In this paper, we used a deep learning model for spectrum sensing in cognitive radio networks. The proposed model allows us to identify the spectrum state by classifying the captured signals into PU signals or Noise signals. Therefore, a simplified spectrum sensing implementation has been proposed based on real signals generated by an Arduino Uno card and 433 MHz Wireless transmitter. Then, the transmitted signals are detected in MATLAB software by RTL-SDR dongle using the deep learning model. The system performance is evaluated by two metrics, the probability of detection and the probability of false alarm. Then, the obtained results are compared with three other techniques, such as energy detection (ED), artificial neural networks (ANN), and support vector machine (SVM). The experimental results showed that the deep learning model using a convolutional neural network (CNN) possessed better classification accuracy and faster training speed than other methods, including ED, ANN, and SVM.

References

1. Identification and quantification of key socio-economic data to support strategic planning for the introduction of 5G in Europe-SMART, 2014/0008. Technical report, European Union (2016)

2. FCC Spectrum Policy Task Force, "Report of the spectrum efficiency working group," (2002) <http://www.fcc.gov/sptf/reports.html>
3. Kolodzy, P., et al.: Next generation communications: Kickoff meeting. In: Proceedings of DARPA (2001)
4. Mitola, J., Maguire, G.Q.: Cognitive radio: making software radios more personal. *IEEE Pers. Commun.* **6**(4), 13–18 (1999)
5. Mitola Iii, J.: Cognitive radio for flexible mobile multimedia communications. *Mobile Netw. Appl.* **6**(5), 435–441, 2001
6. Akyildiz, I.F., Won-Yeol, L., Vuran, M.C., et al.: A survey on spectrum management in cognitive radio networks. *IEEE Comm. Magaz.* **46**(4), 40–48 (2008)
7. Haykin, S.: Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.* **23**(2), 201–220 (2005)
8. Yucek, T., Arslan, H.: A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Commun. Surv. Tut.* **11**(1), 116–130 (2009)
9. Elrharras, A., Saadane, R., Wahbi, M., et al.: Signal detection and automatic modulation classification-based spectrum sensing using PCA-ANN with real word signals. *Appl. Math. Sci.* **8**(160), 7959–7977 (2014)
10. Cabric, D., Mishra, S.M., Brodersen, R.W.: Implementation issues in spectrum sensing for cognitive radios. In: *Signals, Systems and Computers* (2004)
11. Digham, F.F., Alouini, M.-S., Simon, M. K.: On the energy detection of unknown signals over fading channels. In: *Communications, ICC 2003* (2003)
12. Saber, M., El Rharras, A., Saadane, R., et al.: Artificial neural networks, support vector machine and energy detection for spectrum sensing based on real signals. *Int. J. Commun. Netw. Inf. Secur.* **11**(1), 52–60 (2019)
13. Sejdíć, E., Djurović, I., Et Jiang, J.: Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing* **19**(1), 153–183 (2009)
14. Bengio, Y., Courville, A., Et Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013)
15. Schmidhuber, Jürgen: Deep learning in neural networks: an overview. *Neural Networks* **61**, 85–117 (2015)
16. Banzi, M.: *Getting Started with arduino*. O'Reilly Media, Inc (2009)
17. Vantrees, H.L.: *Detection, Estimation and Modulation Theory*. Wiley, vol. 1 (1968)
18. Poor, V.: *An introduction to signal detection and estimation*. Springer, Berlin (1994)
19. Dillinger, M., Madani, K., Alonistioti, N.: *Software defined radio: Architectures, systems and functions*. John Wiley & Sons (2005)
20. Chehri, A., Mouftah, H.: An empirical link-quality analysis for wireless sensor networks. In: *2012 International Conference on Computing, Networking and Communications (ICNC)*, Maui, HI, pp. 164–169 (2019) <https://doi.org/10.1109/icnc.2012.6167403>
21. Farjow, W., Chehri, A., Hussein, M., Fernando, X.: Support Vector Machines for indoor sensor localization. In: *2011 IEEE Wireless Communications and Networking Conference, Cancun, Quintana Roo*, pp. 779–783 (2011) <https://doi.org/10.1109/WCNC.2011.5779231>
22. Chehri, A., Mouftah, H., Farjow, W.: Indoor cooperative positioning based on fingerprinting and support vector machines. In: Sénac, P., Ott, M., Seneviratne, A., (eds) *Mobile and Ubiquitous Systems: Computing, Networking, and Services. MobiQuitous 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 73. Springer, Berlin, Heidelberg (2012)
23. Chehri, H., Chehri, A., Saadane, R.: Traffic signs detection and recognition system in snowy environment using deep learning. In: *Fifth International Conference on Smart City Applications, Safranbolu, Turkey* (2020)

24. Saber, M., El Rharras, A., Saadane, R., Chehri, A., Hakem, N., Kharraz, H.A.: Spectrum sensing for smart embedded devices in cognitive networks using machine learning algorithms. In: 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Verona, Italy, to appear 16–18 September (2020)
25. Chehri, A.: Non-cooperative spectrum allocation based on game theory in IoT-oriented narrowband plc networks. In: IEEE 91st Vehicular Technology Conference, Antwerp, Belgique (2020)
26. Saber, M., Saadane, R., Aroussi, H., Chehri, A.: An optimized spectrum sensing implementation based on SVM, KNN and TREE algorithms. In: IEEE 15th International Conference on Signal Image Technology & Internet Based Systems, Sorrento (NA), Italy (2019)
27. Slalmi, A., Saadane, R., Chehri, A.: Energy efficiency proposal for Iot call admission control in 5G Network. In: IEEE 15th International Conference on Signal Image Technology Internet Based Systems, Sorrento (NA), Italy (2019)
28. de Carvalho, F.B., Lopes, W.T., Alencar, M.S., José Filho, V.S.: Cognitive vehicular networks: an overview. *Procedia Computer Science*, **65**, 107–114 (2015)
29. Chehri, A., Mouftah, H.T.: Autonomous vehicles in the sustainable cities the beginning of a green adventure. *Sustain. Cities Soc.* **51**, 101751 (2019)