

Cooperative Spectrum Sensing in Cognitive Radio Networks: A Survey on Machine Learning-based Methods

Sundous Khamayseh¹ and Alaa Halawani^{1,2}

¹ College of Information Technology and Computer Engineering, Palestine Polytechnic University

² Intelligent Systems Research Center, Palestine Polytechnic University

<https://doi.org/10.26636/jtit.2020.137219>

Abstract—The continuous growth of demand experienced by wireless networks creates a spectrum availability challenge. Cognitive radio (CR) is a promising solution capable of overcoming spectrum scarcity. It is an intelligent radio technology that may be programmed and dynamically configured to avoid interference and congestion in cognitive radio networks (CRN). Spectrum sensing (SS) is a cognitive radio life cycle task aiming to detect spectrum holes. A number of innovative approaches are devised to monitor the spectrum and to determine when these holes are present. The purpose of this survey is to investigate some of these schemes which are constructed based on machine learning concepts and principles. In addition, this review aims to present a general classification of these machine learning-based schemes.

Keywords—cognitive radio, cooperative spectrum sensing, IEEE 802.22, machine learning, spectrum sensing.

1. Introduction

Cognitive radio (CR) is an intelligent radio technology capable of determining the frequencies that are in use, detecting the available spectrum holes, and then reconfiguring transceiver parameters based on the radio environment information obtained. Spectrum sensing (SS) is the first task in cognitive radio life cycle. It has been gaining in significance, since it allows to detect spectrum holes.

Several authors have attempted to come up with different classifications of spectrum sensing methodologies utilized in cognitive radio networks (CRNs) [1]–[3]. The authors in [1] categorized SS schemes based on the mode that was relied upon in SS decision making, namely local or cooperative sensing. In [2], [3], the authors categorized SS schemes based on the radio communication types adopted, mainly into narrow- and wide-band sensing. Although cognitive radio is based on artificial intelligence (AI) concepts and principles [4], only a few works were devoted to machine learning schemes relied upon in cooperative spectrum sensing. In contrast, this survey examines different machine

learning-based schemes proposed for cooperative spectrum sensing (ML-based CSS) in CRN.

This paper concludes that three main types of ML-based CSS schemes exist. The first one is known as unsupervised-based CSS. Here, features are extracted using a suitable feature extraction algorithm. These features are then fed into a machine learning (ML) model that tends to divide data into different clusters that are characterized by their corresponding centroids. Consequently, it makes a decision about the channel's availability status by comparing each centroid with a predefined threshold value. This approach offers reasonable performance levels but its accuracy is rather low compared to other methods [5].

The second approach is referred to as supervised-based cooperative spectrum sensing (CSS). In contrast to the previous method, specific features are fed to a classifier along with their labels that are assigned to them for training purposes. The training process then aims to find the best decision boundary that is capable of separating this labeled data into classes. Therefore, the decision is taken based on that boundary. This approach is more accurate than others, but the labeling process generates some overhead during the training phase [6]. The third method is known as reinforcement learning-based CSS. Here, the algorithm focuses on a certain problem, like throughput and energy consumption. Then, it tries to solve that problem by finding the optimal CSS policy which returns the highest cumulative reward. However, used in CSS, this approach creates several challenges, such as high computational complexity and the requirement for a policy that is needed to make decisions in real time [7].

The remaining part of this paper is organized as follows: Section 2 presents the background of SS tasks performed in CRNs. The general model of CSS, as well as the classification of ML-based CSS schemes, are presented in Section 3. In Section 4, feature extraction methods are introduced, while ML-based CSS schemes are explained in Section 5. Section 6 discusses the problem of evaluating performance of ML-based CSS schemes and, finally, Section 7 contains some concluding remarks.

2. Spectrum Sensing Task in CRNs

2.1. Cognitive Radio Technology: An Overview

CR is defined as intelligent radio being aware of the surrounding environment and knowing the frequencies that are in use. One of the most notable features of cognitive radio networks (CRN) is their ability to switch between different radio access methods, as well as ability to transmit within different portions of the radio spectrum [8]. CR is the primary component of any CRN structure and there are two main types of users operating CRNs. They are classified into [9] two categories.

Primary users (PUs) are licensed users who have a legal right to use a part of the spectrum. However, PUs are not granted the exclusive use of that part of the spectrum, they are merely granted a higher priority than other users, and enjoy additional interference protection guarantees.

Secondary users (SUs) are the remaining users who have the opportunity to use a part of the spectrum alongside the PUs. However, their right is conditioned on the activities of the PUs and they can use their part of the spectrum if it is temporarily not occupied by the PUs, or if they are able to share the spectrum. Small scale CRNs consist of one PU and a few SUs [5], [6], [10]–[16] with each one being able to use multiple antennas [12]. In contrast, large scale CRNs consist of more than one PU [4], [7], [17]–[21]. However, in modeling CRNs, multiple PUs may operate within the same CRN area and each one may occupy several sub-bands [7], [21].

2.2. Spectrum Sensing

Spectrum sensing (SS) is a task that is of key importance in CR. CR learns and is conscious of the surrounding environment, trying to detect any existing spectrum holes. SS was first formulated as a binary hypothesis test for radar signal detection [22]. Later, the same test was used to describe the general SS problem:

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

where $y(k)$ is the received signal that was received by SU under ambient noise $w(k)$ (usually $w(k)$ is the additive white Gaussian noise (AWGN) with 0 mean and variance of 1), $s(k)$ is the PU's signal and $k = 1, \dots, K$ are signal samples received by the SU. Parameter α denotes the channel's fading coefficient between the PU and the SU. The SU checks whether PU's signals are present and the channel is considered **idle** under the null hypothesis (H_0 condition) and **busy** under the alternative hypothesis (H_1 condition). Two types of errors are faced in connection with detection of the PU's signals. The first one, type 1, is a false alarm (FA). The decision made indicates that the PU exists, while it is not present, in fact (the decision is $H_1|H_0$). The other error, type 2, is known as missed detection (MD) and occurs when a busy channel is identified as being idle (the decision

is $H_0|H_1$). The two types of errors referred to above are described by the probability of a false alarm (P_{FA}) and of a missed detection (P_{MD}), respectively. The IEEE 802.22 work group recommended that P_{FA} should not exceed 10%, while the probability of detection $P_D = 1 - P_{FA}$ being higher than 90% [23]. If type 1 errors occur frequently, i.e. P_{FA} increases, the SU will lose numerous opportunities to use the communication channel concerned. In contrast, if type 2 errors occur frequently, i.e. P_{MD} increases, then interference between PU and SU is inevitable [22].

The task of SS may be performed in non-cooperative or cooperative modes. In the non-cooperative mode, each SU determines the channel's state individually. This mode is suitable when the nodes are unable to share their SS information. In cooperative spectrum sensing (CSS), SUs work together to determine the channel's status. CSS offers a high accuracy level, as all SUs take part in making a shared decision about the channel's condition.

3. ML-based Cooperative Spectrum Sensing

3.1. General Model of ML-based CSS

In conventional CSS schemes, an SU investigates whether the PU's signal exists and makes a decision about the channel's availability status. Such an approach to CSS is based on ML schemes (ML-based CSS) and encompasses several steps that are shown in Fig. 1. After aggregating the sensing information from all SUs, useful features that differentiate the PUs' signal samples are extracted. Then, these features are fed to the ML model used for decision making. Lastly, the final sensing decision is made upon the result obtained with the use of the ML model. Several ML algorithms relied upon in numerous research projects will be explained next.

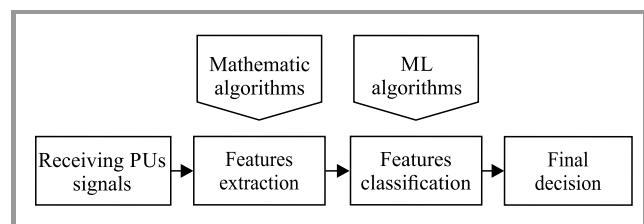


Fig. 1. General model of ML-based CSS schemes.

3.2. Classification of ML-based CSS

ML-based CSS schemes, as shown in Fig. 2, are generally classified based on the ML algorithm used. The three main types are unsupervised ML-based CSS, supervised ML-based CSS, and reinforcement ML-based CSS. In unsupervised ML-based CSS, the features are fed to the model without any labels during the training phase [10]–[13], [15], [17], [19], [21], [24], [25]. Many unsupervised machine learning algorithms may be found in the literature, some of which are mentioned in Fig. 2. Clustering paradigms are

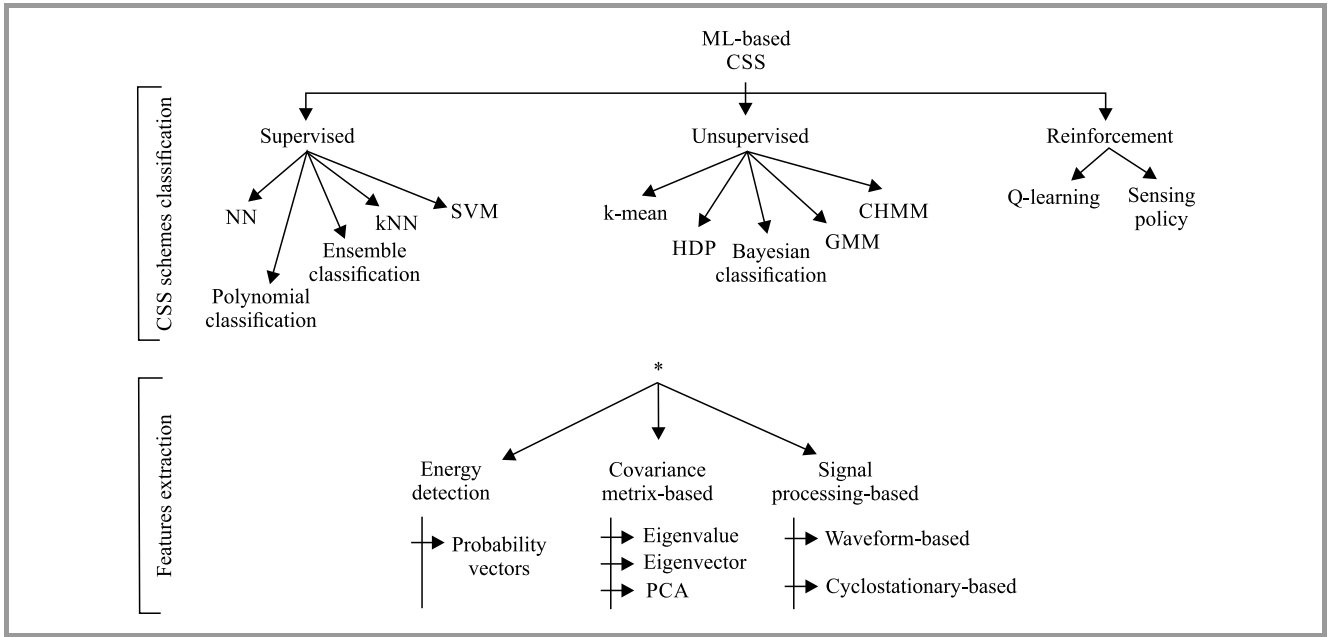


Fig. 2. Classification of ML-based CSS.

the most famous ones, e.g. the very well-known k-means clustering algorithm that will be discussed later in this paper (alongside its variant, the Gaussian mixture model (GMM)).

On the contrary, the training of supervised ML-based methods requires that features be fed to the classifier along with their labels [4]–[6], [14], [16], [20], [26]–[28]. Artificial neural networks (ANN) are the most famous supervised machine learning model. They come in two variants, with traditional architecture and deep neural networks architecture that serves as a basis for modern deep learning schemes. Other examples of the supervised model include support vector machines (SVM), k-nearest neighbors (KNN), and ensemble classifiers. This paper will focus primarily on supervised-based CSS contributions that utilize both neural networks and support vector machine architectures, since they are the most famous and the most commonly used architectures.

Reinforcement ML-based methods consider certain characteristics of the signal in SS [7], [18] such as power consumption, throughput, energy efficiency, etc. The agent is provided with rewards in order to evolve its behavior. These rewards are constructed depending on which problems need to be solved. Q-learning is a famous technique for reinforcing learning algorithms.

Regardless of which ML algorithms (k-mean, GMM, SVM, ANN) are used, several mathematical algorithms have been innovated in order to extract useful features from the signal. As shown in the bottom part of Fig. 2, these algorithms are divided into three categories. The first category covers energy detection-based (ED) algorithms [4]–[6], [10], [14], [16], [17], [20], [21], [26], [27]. Here, energy samples of the PU’s signal are collected and are then transformed into energy vectors. The second category covers signal processing-based solutions [14], [16],

[19], [25]. Here, the focus is placed on finding circularly characteristics that help differentiate between transmitted and noise signals. The third category is known as the covariance matrix [11]–[13], [15], [24] and relies on constructing feature matrices from the sensing samples, and on performing specific computations. These methods are presented in detail in Section 4.

4. Feature Extraction Methods

Feature extraction is the second phase of the general ML-based CSS model. Various algorithms have been designed in order to perform the feature extraction task. This section will explain some of these algorithms.

4.1. ED-based Feature Extraction

A general CSS model encompasses P PUs and N SUs. In numerous research projects, a single PU is used, since it reduces the degree of complexity. This paper will also focus on a single PU (unless mentioned otherwise) in order to facilitate the explanation. When the n -th SU receives K samples of the transmitted PU’s signal, it computes the average energy (E) level of the signal:

$$E_n = \frac{1}{K} \sum_{k=1}^K |Z_n(k)|^2, \quad (2)$$

where $|Z_n(k)|^2$ is the energy value corresponding to k -th sample and estimated by SU n . After computing E value by all N SUs, the column vector of the energy values $\{E_1, E_2, \dots, E_n, \dots, E_N\}^T$ is fed to the ML model as a feature vector to make the final sensing decision. **Probability vectors method** is one of ED-based feature extraction scenarios. This method reduces multidimensional energy vec-

tors into two-dimensional probability vectors [11]. In this method, new probability vectors of the two binary hypotheses (H_0 and H_1) are defined under the probability density function (PDF) of the multivariate Gaussian distribution, taking the mean and the covariance matrix of each hypothesis as input parameters and constructing the probability vector based thereon.

4.2. Covariance Matrix-based Feature Extraction

The S_i sensing matrix is constructed based on the PU's signal received. Thanks to the random matrix theory, covariance matrix R is reconstructed from that sensing matrix:

$$R = \frac{1}{K} \sum_{k=1}^K S_i \cdot S_i^T . \quad (3)$$

Several proposals stem from this approach, including **eigenvalue/eigenvector methods**. Let us suppose that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq \dots \geq \lambda_N$ are the eigenvalues of covariance matrix R . Then, the eigenvector (\vec{v}_n) of R corresponding to the eigenvalue may be computed as:

$$R\vec{v} = \lambda \vec{v} . \quad (4)$$

Numerous schemes relied on the eigenvalue/eigenvector in several ways, as a feature of the ML model. For example, the ratio of the maximum and minimum eigenvalues (MME) and the difference between the maximum eigenvalue and the average eigenvalue (MSE) are used in [11], [24], whereas the improved MME (IMME) and the improved MSE (MSE) are used in [12] after computing the principal component of the eigenvector of matrix S_i .

4.3. Signal Processing-based Feature Extraction

Signal processing is based on transforming the received PU's signal from the time domain to the frequency domain, using the Fourier transform. **Waveform-based detection (WFD)** or **coherent-based detection (CD)** and **cyclostationarity-based feature detection (CFD)** are the most common techniques of this type. CD assumes that patterns of the control signal, such as the pilot, preambles, spreading sequence, midambles, etc. are known and can be efficiently utilized [14]. The notion of cyclostationarity, as used in CFD, indicates the periodic characteristic of a certain signal. Usually, cyclostationarity of the received PU signal is described in terms of its mean and auto-correlation [29], [14]. In contrast, noise signal does not have such periodic characteristics, so the difference between the transmitted signals and noise may be relied upon. The presence of a cycle-stationary pattern of the PU signal received may be determined using the so-called spectral correlation density function (SCD) [14]. In order to determine whether a PU signal is present, one may rewrite Eq. (1) using the SCD function, as:

$$S_{y(k)}^\omega[k] = \begin{cases} S_{w(k)}^\omega[k], & H_0 \\ |\alpha|^2 S_{s(k)}^\omega[k] + S_{w(k)}^\omega[k], & H_1 \end{cases} , \quad (5)$$

where $S_{y(k)}^\omega[k]$ is the SCD of the transmitted PU signal at some cyclic frequency ω and α is the channel gain coefficient.

5. ML-based Cooperative Spectrum Sensing Schemes

A large number of researchers considered applying ML algorithms in constructing CSS systems. Table 1 summarizes some of these algorithms and the feature extraction methods used. The classification of these research projects was explained in Subsection 3.2, where they are categorized into supervised, unsupervised and reinforcement ML-based approaches. This section explains how the problem of CSS may be tackled using these algorithms.

5.1. Unsupervised-based CSS

After collecting a sufficiently large number of training feature vectors created by any type of feature extraction algorithm, the ML model is trained to make a decision about the channel's availability state. In unsupervised methods, these features are fed to the ML model to produce the global decision about PUs' state. Several released proposals [21], [15], [25] suggested applying different types of unsupervised learning algorithms to resolve the CSS problem in CRNs. This part explores some of these most prominent algorithms.

K-means is an algorithm that maps a collection of data samples into non-overlapping clusters [15]. The assignment of a certain point to a given cluster is determined based on a distance measure (usually Euclidean distance). The point will be assigned to the cluster whose center is closest to that point. Let us assume that the collection of feature vectors constructs a set called Ψ . Each Ψ represents a cluster and is indexed by j ($j \in \{0,1\}$ under the binary hypotheses condition). Each cluster has its own centroid C_j representing the arithmetic mean of that cluster. Therefore, the distortion function Θ computes, for all samples, the overall square distance to the corresponding centroid. Here, the k-means algorithm aims at minimizing the distortion function Θ .

After training the k-means model, the members and the centroid of each cluster are determined. In the testing phase, the model is able to predict the suitable decision, i.e. the channel's availability state. Let two clusters i and j represent the channel's availability and unavailability states, respectively. l' denotes a new test vector (data point) that needs to be assigned to a cluster. Then, the model decides about the membership based on a predefined threshold ζ :

$$\frac{\|l' - C_i\|}{\|l' - C_j\|} \leq \zeta . \quad (6)$$

This means that vector l' is classified into the cluster C_i (i.e. the channel is available) if the ratio of the distance between that vector and C_i and the distance between that vector and C_j is lower than threshold ζ .

Table 1
Classification table

Reference	ML algorithms types	ML algorithms, number of clusters	Features extraction
[10]	Unsupervised	k-means, $k = 2$	ED
[11]	Unsupervised	GMM and k-mean, $k = 2$	Eigenvalue/eigenvector
[12]	Unsupervised	k-medoids, $k = 2$	IMSE/IMME
[13]	Unsupervised	Fuzzy c-means, $c = 2$	Geodesic distance
[17]	Unsupervised	Kernel fuzzy c-means, $c = 3$	E _{max}
[24]	Unsupervised	WEMD (with k-means), $k = 2$	DMEAE/DMM
[19]	Unsupervised	HDP, cls = 9, 18, 36	Fourier coefficients (CFD)
[21]	Unsupervised	Bayesian learning model (BP-SHMM)	ED
[15]	Unsupervised	k-means (multi-bands based), $k = 2$	Eigenvalue
[25]	Unsupervised	Blind-CHMM (k-means-based)	CWT (singularity detection)
[5]	Supervised	Lin/poly-SVM, cls = 2	Probability vectors
[4]	Supervised	2-phase SVM, cls = 2	ED
[6]	Supervised	DNN (CNN) ED (as inputs)	
[20]	Supervised	CSVM (multi-class SVM)	ED
[14]	Supervised	Lin/second-order PC (polynomial classifier)	ED/CFD/CD
[16]	Supervised	Ensemble classifier (based on decision trees and AdaBoost algorithm)	CFD
[26]	Supervised	Back propagation neural network ED and LRS-G ² (as inputs)	
[28]	Supervised	SVM	Beamformer-aided
[27]	Supervised	CSVM (multi-class SVM)	ED (for the received signal and the residual energy of SU)
[18]	Reinforcement	Sensing policy (based on ϵ -greedy policy) Q-value (throughput)	
[7]	Reinforcement	POMDP (Q-learning-based) Q-value (number of idle channels)	

The Gaussian mixture model (GMM) provides a smoother membership function than the k-means algorithm. The distribution function of GMM may be defined under the Gaussian density function $\mathcal{N}(l|\mu_j, R_j)$ which indicates that training vector l belongs to cluster j with the mean μ_j and the covariance matrix R_j being its parameters [11]. The training phase in GMM involves estimating parameters μ_j , R_j and π_j for each class, where π_j is a mixing probability coefficient defining how big/small the Gaussian function will be. The estimation process is performed using an expectation-maximization algorithm, with the details thereof described in [11]. Finally, the testing vectors are assigned to each cluster based on this classifier rule (assuming the binary hypotheses test is adopted):

$$\ln \frac{\pi_1 \mathcal{N}(l|\mu_1, R_1)}{\pi_2 \mathcal{N}(l|\mu_2, R_2)} \geq \zeta . \quad (7)$$

Then, the test vector is assigned to the “channel available” cluster when the previous equation is satisfied, and to the “channel unavailable” cluster otherwise.

As stated above and as may be seen in Table 1, unsupervised CSS methods are widely adopted in many research projects. K-means with ED-based feature extraction is adopted in [10], whereas geodesic distance as feature vectors with the fuzzy c-mean is proposed in [13]. In addition, paper [17] proposes filtering the energy vector collected in order to get maximum one vector as a cleaned feature of the kernel fuzzy c-means. The eigenvalue/eigenvector features with k-means were adopted in [12], while they are used with GMM in [11]. Non-parametric Bayesian learning model hierarchical Dirichlet process and beta process sticky hidden Markov model (BP-SHMM) are proposed [19] and [21], respectively. The hierarchical Dirichlet process converts the received signal into the frequency domain in order to find the Fourier coefficients. Then, it uses these coefficients as feature vectors, whereas the beta process sticky hidden Markov model is ED-based.

Additionally, article [24] introduces a signal processing scheme called wavelet empirical mode decomposition (WEMD). This scheme combines the empirical mode decomposition algorithm and the wavelet threshold algo-

rithm in order to remove noise components, hence reducing noise effects. Then, after filtering noise from the transmitted signal, features are extracted using the difference between the maximum eigenvalue and the average energy, and the difference between the maximum and minimum eigenvalue methods. Finally, k-means is adopted as an ML model, relying on these two types of features extraction methods. Last but not least, a blind continuous hidden Markov model (blind-CHMM) scheme is proposed in [25]. This algorithm is capable of recognizing the transmitted power level of the PU. It uses the k-means algorithm to detect the presence of a PU, as well as the continuous wavelet transform method for feature extraction. Moreover, in that scheme, two strategies are proposed. The first applies the ED method to build the observation sequence before computing the continuous wavelet transform, while the other uses the minimum-maximum eigenvalues as feature vectors.

5.2. Supervised-based CSS

To train a supervised-based CSS system, the features are extracted from the received signal and are then assigned with a certain label corresponding to their class affiliation. Due to the binary hypothesis, binary labels may be represented in the following manner. Label 1 indicates that PU is absent and the channel is available, whereas label 0 indicates that PU is present and the channel is not available. However, multi-class hypotheses are more complex, thereby resulting in the need to form label codes corresponding to each class [20], [27], [28].

Support vector machines (SVM) and neural networks (NN) are the most famous examples of supervised-based CSS. SVM is an algorithm that aims to find the optimal hyperplane that leaves the maximum margin from all potential classes. Due to the binary hypothesis test, two potential classes reflect the channel's availability state. In more complicated scenarios, the multi-class SVM with one-versus-all (OVA) strategy is used. OVA is a commonly used approach in which a certain class is marked as the positive class, and the remaining classes are marked as negative [20], [27].

Many research contributions proposed using SVM as a machine learning model. Probability vectors were proposed as feature vectors for SVM, with linear and polynomial kernels, in [5], whereas [4] proposed applying SVM in two phases, along with the ED-based method, for feature extraction. In the first phase, the data samples are randomly split into two sets. The first set contains training samples, whereas the other is used for testing (the true labels are omitted). In the second phase, the samples with their labels work as a training set. The reason for applying the second phase is to neutralize the misclassification of the first phase. Finally, the multi-class SVM ED-based approach with multiple PUs is adopted in [20], [27], and the beamformer-based approach is adopted in [28].

Neural networks are a collection of algorithms encompassing a set of interconnected virtual neurons intended to

work in a manner closely resembling a biological neuron network or electronic structures. A few CSS schemes based on neural networks were proposed in the literature. One is deep CSS (DCSS), based on convolutional neural networks (CNNs) from [6].

Generally, a CNN architecture is a layered architecture with each type of layers having a specific task. Three types of layers may be identified: convolutional layers, max pooling layers, and fully connected layers. The general architecture may be mainly divided into two parts: the convolution part and the fully connected part (FC), as shown in Fig. 3. The convolution part contains the convolutional layers and the maximum pooling layers. The convolutional layers contain a set of filters (kernels) that are convolved with the input matrix. The main task of these filters is to extract the most important features of the input matrix. Unlike in the case of manually-defined filters, the contents of these filters are determined during the training of the network. This means that the process of feature extraction becomes a part of the network learning process, and the traditional stage of manual feature engineering disappears. The size of the kernel (e.g. 3×3) is a hyperparameter that should be decided in advance. Usually, the size of 3×3 is a good choice that works very well.

The objective of the max pooling layers is to downsample the input matrix to reduce data dimensionality, leading to lower computational costs. Maximum pooling helps also in reducing overfitting in the network. Max pooling divides the input matrix into distinct blocks and the maximum value from each block is kept only. So, if the input is 4×4 matrix and the block size is set to equal 2×2 , the output of the max pooling layer will be a 2×2 matrix. Rectified linear unit (ReLU) layers are also found in the CNN architecture. The purpose of the ReLU unit is to increase nonlinearity of data.

Fully-connected layers are needed for the classification process. They resemble a traditional feed-forward neural network. The output of the convolution part is flattened into a column vector and fed to this feed-forward neural network for classification purposes.

The model proposed in [6] uses the CNN architecture described above. In this model, a small-scale network with one PU is adopted that may operate several sub-bands. Each SU examines the existence of PU within a given sub-band and produces a two-dimensional array from the labeled sensing data, as illustrated in Fig. 3 (the label assumes the value of 0 or 1, indicating whether the channel is available or not). Then, the 2D array is fed to the convolution part which encompasses the 3×3 convolution layer, the rectifier linear unit layer and the max pooling layer. The first layer extracts spatial correlation of the sensing data, whereas the remaining layers deal with the non-linear behavior and reduce the size of the sensing data, respectively. The FC layer multiplies the weights and adds the biases to the result produced by the previous layer. Finally, the softmax operator is used at the end of the FC layer for making decisions regarding the presence of PUs.

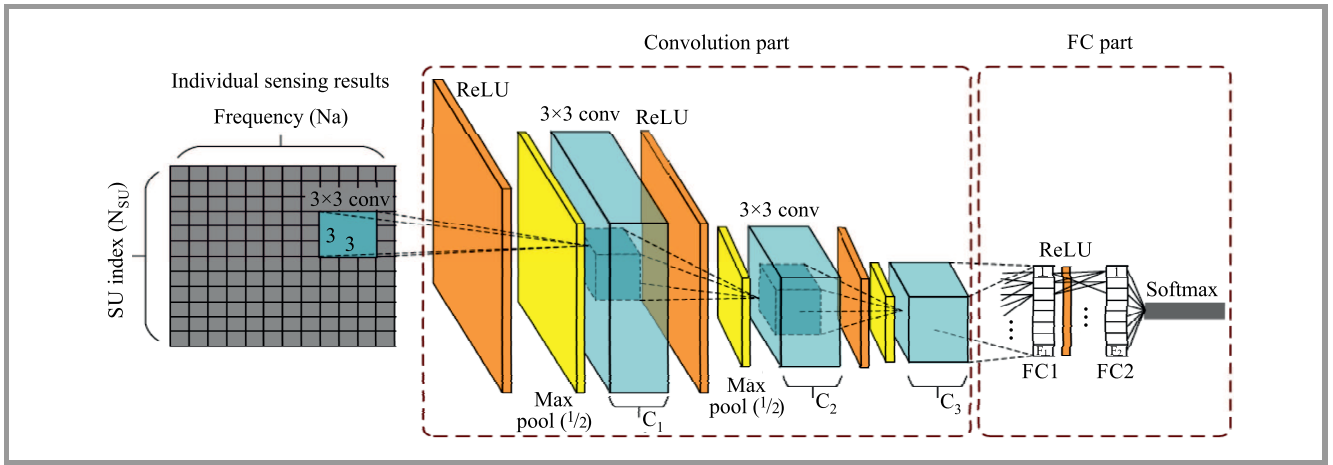


Fig. 3. CNN model for DCSS [6].

Another scheme that is based on traditional (non-deep) artificial neural networks (ANN) was proposed in [26]. This scheme utilizes the ED and Zhang statistic from likelihood statistic test to train the ANN for decision making purposes. Moreover, another supervised scheme, i.e. the ensemble-based classifier, is introduced in [16]. The design of this scheme is based on the decision tree and on AdaBoost algorithms, while the features extraction method that is relied upon is CFD-based.

5.3. Reinforcement CCS

Reinforcement learning means that the action is taken which brings the maximum reward. Q -value $Q(a)$ denotes the reward granted, corresponding to action a taken. In a reinforcement ML-based paradigm, the agent initially estimates the action that should be taken next time, based on a certain policy [18], [7]. Then, it gets a reward $r(a)$ that reflects the validity of the action taken. There is a frequently used term when speaking of reinforcement ML-based algorithms – namely balance between exploration and exploitation. Exploitation indicates that the agent takes an action upon the highest estimated Q -value, whereas exploration indicates the random selection of a certain action regardless of action-value estimation [18].

Sensing based on the ϵ -greedy policy is proposed in [18], whereas an efficient sub-band selection policy based on replicated Q-learning is proposed in [7]. In the sensing policy, the reward represents the instantaneous throughput for the sub-band selection. Assuming that the sub-band is denoted with sd , in this policy the SU which had the permission to access the sd feeds back information about the achieved throughput towards the FC. Herein, FC updates the Q -value for a particular SU by comparing its decision with the SU's decision. Then, the obtained reward is updated based on the following rule:

$$r_{t+1} \sim \begin{cases} dn_{t+1}(SU, sd), & dn_{t+1}(FC, sd) = 1 \\ Q_t(SU, sd), & dn_{t+1}(FC, sd) = 0 \end{cases}, \quad (8)$$

where $dn_{t+1}(SU, sd)$ indicates the local decision taken by the SU on sd while $dn_{t+1}(FC, sd)$ is the global decision initiated by the FC. After updating all Q -values based on that rule, FC exploits its knowledge and informs SUs to sense the sub-band that has the maximum Q -values. Finally, [7] proposed an efficient sub-band selection policy which is modeled based on the partially observable Markov decision process (POMDP) in which the highest rewards are granted to the action that identifies the largest number of idle channels, and the lowest rewards are granted based on the lowest number of idle channels identified.

6. Comparison and Discussion

This section analyzes the performance of several ML-based CSS approaches at low signal-to-noise ratio (SNR) levels. The performance of ML-based CSS schemes was measured based on the receiver operating characteristic (ROC) curve, i.e. a plot showing the probability of detection PD versus that of a false alarm PFA. According to the ROC curve, performance of the proposed schemes is somewhat convergent, regardless of which ML type (unsupervised or supervised) is used for decision making. Moreover, considering the IEEE 802.22 standard, one may observe that the majority of schemes offer superior performance even at low SNR levels, as may be seen in Table 2a-b. Performance of the reinforcement-based CSS scheme is usually measured based on the throughput obtained, as shown in Table 2c.

6.1. ROC for Performance Evaluation

Unsupervised-based CSS schemes showed good performance – according to the ROC curve presented in Table 2a. The unsupervised wavelet empirical mode decomposition scheme [24], with the difference between maximum eigenvalue and the average energy, and the difference between maximum and minimum eigenvalue features, showed the best performance exceeding the IEEE

Table 2
Performance evaluation of ML-based CSS

A: ROC for performance evaluation, unsupervised-based CSS						
Reference	P_D (P_{FA})	Number of PUs (P)	Number of SUs (N)	SNR	Number of training (M)/ testing samples	Performance improvement
[10]	0.98 (0.1)	1	3	-12 dB	500/500	SNR+, N+, M+, k-, depends on channel type
[11]	0.81 (0.1) GMM_E-val, 0.65 (0.1) GMM_E-vec 0.78 (0.1) k-mean_E-val, 0.73 (0.1) k-mean_E-vec	1		-10 dB	200 samples	SNR+, N+, M+ (≥ 4), E-val does better
[12]	0.97 (0.1) IMSE, 0.94 (0.1) IMME	1	4	-15 dB	1000/1000	SRN+, N+, IMSE does better
[13]	0.98 (0.1)	1	10	-15 dB	1000/1000	SNR+, N+, M+
[17]	0.89 (0.3)	4	30	-15 dBW (PU TX power)	2000 samples	SNR+, N+, M+, P-
[24]	0.96 (0.1) DMEAE, 0.94 (0.1) DMM	1	4	-18 dB	500/500	SNR+, N+, DMEAE does better
[19]	0.96 (0.1) clts = 9	8 (2 active)	30	0 dB (-90 dBm noise floor)	100 simulation scenarios	SNR+, cluster size (number of SUs)+, number of clusters-
[21]	0.98 (0.1)	8 (2 active)	9	50 mW (PU TX power)		N+, P-, PU TX power-
[15]	0.99 (0.025)	10 sub-bands 4 occupied		-13 dB	300 samples (5000 simulation trails)	SNR+, a little affected by noise uncertainly, affected considerably by unequal noise variances, affected considerably by the number of occupied sub-bands
[25]	0.90 (0.1) strategy 1 0.75 (0.01) strategy 2	1		-10 dB	5000 (observation size)	SNR+, M+
B: ROC for performance evaluation, supervised-based CSS						
Reference	P_D (P_{FA})	Number of PUs (P)	Number of SUs (N)	SNR	Number of training (M)/ testing samples	Performance improvement
[5]	0.96 lin, 0.94 poly	1	9	300 mW (PU TX power)	800/600	SNR+, N+, M+, affected a little by the kernel function, $P_{FA}=0.05$
[4]	0.84 (0.1)	2	25	200 mW (PU TX power)	560 samples	SNR+
[6]	0.91 (0.1)	1	32	-164 dBm/Hz (noise power density)	200 sample	SNR+, N+, M+
[20]	0.89 (0.1)	2	5	-15 dB	400/600	SNR+, N+, M+
[14]	0.92 (0.1) lin/ second-order-ED 0.90 (0.1) lin-CFD, 0.86 (0.1) second-order CFD 0.94 (0.1) lin-CD, 0.90 (0.1) second-order CD		5	-10 dB -18 dB -14 dB	200 bits (observation window size)	SNR+, N+, window size+
[16]	0.99 (0.1)			-12 dB	2000 samples (1000 of them with noise only)	SNR+, almost same with different OFDM signals
[26]	0.90 (0.022) FM broadcast 0.82 (0.02) GSM-900 DL 0.96 (0.013) DCS-1800 DL 0.97 (0.019) UHF television			-14 dB	500	SNR+, M+, affected by different types of radio technology
[28]	0.97 (0.02)	1	8 number of antennas	-18 dB	2000 samples	SNR+, M+, P-
C: Throughput for performance evaluation						
Reference	Throughput	Training/testing samples or number of iterations	Throughput improvement	The reference	Notes	
[27]	37% (SNR = -5 dB) 33% (SNR = -15 dB)	500/2000	Fixed when the battery capacity ≥ 1500 packets, SNR+	Max/min of PU TX power	2/2 (number of PU/number of SU)	
[18]	83% ($\epsilon = 0.1$) 60% ($\epsilon = 0.5$)	2000	Convergence at 200 iterations	The ideal policy		
[7]	~ 1.7 normalized accumulative rewards	5000	Convergence at 5000	Other schemes	The discount factor $\gamma = 0.4$ $\epsilon = 0.8$	
Legend: PU TX power is an abbreviation of the PU's transmitted power E-val and E-vec are the eigenvalue and the eigenvector, respectively + sign means increased by an increase - sign means increased by an decrease						

802.22 standard ($\geq 90\%$) at the lowest SNR (-18 dB). K-medoids with an improved ratio of the maximum and minimum eigenvalues and the difference between the maximum eigenvalue and the average eigenvalue [12], fuzzy c-means with geodesic distance [13], and kernel fuzzy c-means with max energy [17] showed the second lowest SNR result (-15 dB). However, kernel fuzzy c-means with max energy failed to meet the requirement of the IEEE.802.22 standard related to P_D and P_{FA} (0.89 and 0.3 respectively). This is caused by the fact that the system model of the scheme in question is more complicated, as there are four PUs that are active in the CRN.

Strategy 1 of the blind continuous hidden Markov model with the continuous wavelet transform [25] achieved the requirement of the IEEE 802.22 standard related to P_D percentage, but failed to achieve it with regard to low SNR. However, its performance may be improved by increasing the PU's transmit power or by increasing the number of collaborating SUs. GMM, k-means with the eigenvalue and eigenvector [11] and the strategy 2 of the blind continuous hidden Markov model [25] failed to achieve the requirements of the IEEE 802.22 standard concerning P_D percentage ($\ll 0.9$) or SNR (-10 dB).

Supervised-based CSS schemes also showed good performance related to ROC curve, as shown in Table 2b. Supervised SVM with beamformer-aided scheme and linear classifier with the CFD method exhibited the best performance, exceeding the IEEE 802.22 standard ($\geq 90\%$) at the lowest SNR (-18 dB). On the contrary, 2-order polynomial classifier with the CFD method, as well as linear and 2-order polynomial classifier with the ED method failed to achieve the requirements of the IEEE 802.22 standard (0.86 in the case of P_D percentage and -10 dB in the case of SNR, respectively). However, performance of 2-order polynomial classifier with the CFD method may be improved by increasing the PU's transmitting power. 2-phase SVM [4] and multi-class SVM [20] with the ED method failed to achieve the requirements of the IEEE 802.22 standard concerning P_D percentage (< 0.90), despite multi-class SVM with the ED method achieving the second lowest SNR. Linear and 2-order polynomial classifier with the CD method [14], as well as the back propagation neural network (with ED and likelihood ratio test statistic as input samples) that is employed in different radio technologies, achieved the third lowest SNR (-14 dB), showing superior performance in terms of P_D percentage as well (≥ 0.90 despite the use of GSM-1800 DL radio technology).

In general, the superior performance of several unsupervised- and supervised-based CSS approaches is usually related to the small scale of CRNs. Small scale networks are less complex and less prone to signal interference. So, they are expected to offer higher performance compared to large-scale networks. In principle, overall performance of unsupervised- and supervised-based CSS is mainly affected by four different factors. First, the PU's transmission power (the higher the better). Second, the number of collabo-

rating SUs (the higher the better). Third, the number of PUs present (the lower the better), and finally, the number of training samples. A higher number of samples will lead to better training results in general.

6.2. Throughput for Performance Evaluation

As stated above, performance was also measured using throughput [27], [18], [7] – as shown in Table 2c. The supervised multi-class SVM with ED methods [27] achieved 37% and 33% of the maximum throughput at -5 dB SNR and -15 dB SNR, respectively. The maximum throughput may be achieved by ensuring that the PU is sending with the highest possible transmission power. The sensing approach based on the ϵ -greedy policy [18] achieved the result of 83%, compared to the result of an ideal policy equaling $\epsilon = 0.1$. Ideal policy assumes that the sub-bands with the highest instantaneous throughput may be selected, and the highest instantaneous throughputs may then be found. Finally, replicated Q-learning-based sub-band selection [7] achieved a normalized accumulated reward of approximately 1.7, representing a throughput that is comparable with that of several sub-band selection policies, such as Bellman-optimality equation and Markov decision process. However, in reinforcement-based CSS schemes, throughput converges as the system trains more, meaning that these schemes cannot be applied in real time.

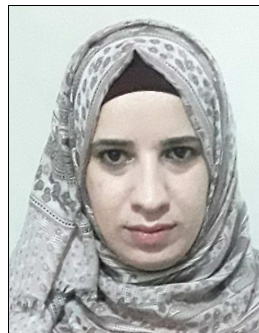
In general, unsupervised-based CSS schemes offer reasonable performance, but their accuracy is slightly lower compared to other approaches [5]. Supervised-based CSS schemes are more accurate than others, but generate some overhead during the training phase, because of the labeling process [6]. Reinforcement-based CSS schemes lead to several challenges, such as high computational complexity and the policy needed to operate in real-time, so this type is rarely used when proposing new ML-based CSS schemes [7].

7. Conclusion

This paper considered the main types of machine learning algorithms and categorized the proposed ML-based CSS schemes into three classes, namely unsupervised-based CSS, supervised-based CSS, and reinforcement-based CSS. It also categorized the mathematical algorithms used for feature extraction, dividing them into three types: energy detection-based feature extraction, cyclostationary-based feature extraction, and signal processing-based feature extraction. Finally, analysis of the results of several proposed schemes showed that unsupervised-based CSS is less complex but a bit less accurate than other approaches. On the other hand, supervised-based CSS offered excellent accuracy, but caused a slight overhead during the training phase. Reinforcement learning offers reasonable accuracy and improves throughput, but may be hard to apply in real time and requires high computational complexity.

References

- [1] V. Ramani and S. K. Sharma, "Cognitive radios: A survey on spectrum sensing, security and spectrum handoff", *China Commun.*, vol. 14, no. 11, pp. 185–208, 2017 (DOI: 10.1109/CC.2017.8233660).
- [2] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications", *IEEE Commun. Surveys & Tutor.*, vol. 19, no. 2, pp. 1277–1304, 2017 (DOI: 10.1109/COMST.2016.2631080).
- [3] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions", *Sensors*, vol. 19, no. 1, pp. 1277–1304, 2019 (DOI: 10.3390/s19010126).
- [4] E. Ghazizadeh, D. Abbasi-moghadam, and H. Nezamabadi-pour, "An enhanced two-phase SVM algorithm for cooperative spectrum sensing in cognitive radio networks", *Int. J. of Commun. Syst.*, vol. 32, no. 2, 2019 (DOI: 10.1002/dac.3856).
- [5] Y. Lu, P. Zhu, D. Wang, and M. Fattouche, "Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks", in *Proc. 2016 IEEE Wirel. Commun. and Network. Conf.*, Doha, Qatar, 2016 (DOI: 10.1109/WCNC.2016.7564840).
- [6] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks", *IEEE Trans. on Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, 2019 (DOI: 10.1109/TVT.2019.2891291).
- [7] M. A. Aref, S. Machuzak, S. K. Jayaweera, and S. Lane, "Replicated q-learning based sub-band selection for wideband spectrum sensing in cognitive radios", in *Proc. 2016 IEEE/CIC Int. Conf. on Commun. in China ICC China 2016*, Chengdu, China, 2016 (DOI: 10.1109/ICCChina.2016.7636732).
- [8] Y. Zhang, J. Zheng, and H.-H. Chen, *Cognitive Radio Networks: Architectures, Protocols, and Standards*. Boca Raton: CRC Press, 2016 (ISBN: 9781420077759).
- [9] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview", *IEEE Trans. on Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, 2011 (DOI: 10.1109/TVT.2011.2158673).
- [10] V. Kumar, D. C. Kandpal, M. Jain, R. Gangopadhyay, and S. Debnath, "K-mean clustering based cooperative spectrum sensing in generalized κ - μ fading channels", in *Proc. 2016 22nd National Conf. on Commun. NCC 2016*, Guwahati, India, 2016 (DOI: 10.1109/NCC.2016.7561130).
- [11] G. C. Sobabe, Y. Song, X. Bai, and B. Guo, "A cooperative spectrum sensing algorithm based on unsupervised learning", in *Proc. 10th Int. Congr. on Image and Sig. Process., BioMedical Engin. and Inform. CISP-BMEI 2017*, Shanghai, China, 2017 (DOI: 10.1109/CISP-BMEI.2017.8302156).
- [12] C. Sun, Y. Wang, P. Wan, and Y. Du, "A cooperative spectrum sensing algorithm based on principal component analysis and k-medoids clustering", in *Proc. 33rd Youth Academic Ann. Conf. of Chinese Assoc. of Autom. YAC 2018*, Nanjing, China, 2018, pp. 835–839 (DOI: 10.1109/YAC.2018.8406487).
- [13] S. Zhang, Y. Wang, J. Li, P. Wan, Y. Zhang, and N. Li, "A cooperative spectrum sensing method based on information geometry and fuzzy c-means clustering algorithm", *EURASIP J. on Wirel. Commun. and Network.*, vol. 2019, no. 1, 2019 (DOI: 10.1186/s13638-019-1338-z).
- [14] Y. Hassan, M. El-Tarhuni, and K. Assaleh, "Learning-based spectrum sensing for cognitive radio systems", *J. of Comp. Netw. and Commun.*, vol. 2012, 2012 (DOI: 10.1155/2012/259824).
- [15] K.-j. Lei, Y.-h. Tan, X. Yang, and H.-r. Wang, "A k-means clustering based blind multiband spectrum sensing algorithm for cognitive radio", *J. of Central South Univer.*, vol. 25, no. 10, pp. 2451–2461, 2018 (DOI: 10.1007/s11771-018-3928-z).
- [16] H. B. Ahmad, "Ensemble classifier based spectrum sensing in cognitive radio networks", *Wirel. Commun. and Mob. Comput.*, vol. 2019, Article ID 9250562, 2019 (DOI: 10.1155/2019/9250562).
- [17] A. Paul and S. P. Maity, "Kernel fuzzy c-means clustering on energy detection based cooperative spectrum sensing", *Digit. Commun. and Netw.*, vol. 2, no. 4, pp. 196–205, 2016 (DOI: 10.1016/j.dcan.2016.09.002).
- [18] J. Oksanen, J. Lundén, and V. Koivunen, "Reinforcement learning based sensing policy optimization for energy efficient cognitive radio networks", *Neurocomput.*, vol. 80, pp. 102–110, 2012 (DOI: 10.1016/j.neucom.2011.07.027).
- [19] X.-L. Huang *et al.*, "Intelligent cooperative spectrum sensing via hierarchical dirichlet process in cognitive radio networks", *IEEE J. on Selec. Areas in Commun.*, vol. 33, no. 5, pp. 771–787, 2015 (DOI: 10.1109/JSAC.2014.2361075).
- [20] O. P. Awe and S. Lambotharan, "Cooperative spectrum sensing in cognitive radio networks using multi-class support vector machine algorithms", in *Proc. 9th Int. Conf. on Sig. Process. and Commun. Syst. ICSPCS 2015*, Cairns, QLD, Australia, 2015 (DOI: 10.1109/ICSPCS.2015.7391780).
- [21] Y. Xu, P. Cheng, Z. Chen, Y. Li, and B. Vucetic, "Mobile collaborative spectrum sensing for heterogeneous networks: A Bayesian machine learning approach", *IEEE Trans. on Sig. Process.*, vol. 66, no. 21, pp. 5634–5647, 2018 (DOI: 10.1109/TSP.2018.2870379).
- [22] A. M. Wyglinski, M. Nekovee, and T. Hou, *Cognitive Radio Communications and Networks: Principles and Practice*. Academic Press, 2009 (ISBN: 9780123747150).
- [23] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios", in *Proc. 1st IEEE Int. Symp. on New Front. in Dynam. Spec. Access Netw. DySPAN 2005*, Baltimore, MD, USA, 2005, pp. 328–337 (DOI: 10.1109/DYSPAN.2005.1542649).
- [24] Y. Wang, Y. Zhang, P. Wan, S. Zhang, and J. Yang, "A spectrum sensing method based on empirical mode decomposition and k-means clustering algorithm", *Wirel. Commun. and Mob. Comput.*, vol. 2018, Article ID 6104502, 2018 (DOI: 10.1155/2018/6104502).
- [25] B. Liu, Z. Li, J. Si, and F. Zhou, "Blind continuous hidden Markov model-based spectrum sensing and recognition for primary user with multiple power levels", *IET Commun.*, vol. 9, no. 11, pp. 1396–1403, 2015 (DOI: 10.1049/iet-com.2015.0090).
- [26] M. R. Vyas, D. Patel, and M. Lopez-Benitez, "Artificial neural network based hybrid spectrum sensing scheme for cognitive radio", in *Proc. IEEE 28th Ann. Int. Symp. on Pers., Indoor, and Mob. Radio Commun. PIMRC 2017*, Montreal, QC, Canada, 2017 (DOI: 10.1109/PIMRC.2017.8292449).
- [27] S. Jan, V.-H. Vu, and I. Koo, "Throughput maximization using an SVM for multi-class hypothesis-based spectrum sensing in cognitive radio", *Appl. Sci.*, vol. 8, no. 3, 2018 (DOI: 10.3390/app8030421).
- [28] O. P. Awe, A. Deligiannis, and S. Lambotharan, "Spatio-temporal spectrum sensing in cognitive radio networks using beamformer-aided SVM algorithms", *IEEE Access*, vol. 6, pp. 25377–25388, 2018 (DOI: 10.1109/ACCESS.2018.2825603).
- [29] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications", *IEEE Commun. Surv. & Tutor.*, vol. 11, no. 1, pp. 116–130, 2009 (DOI: 10.1109/SURV.2009.090109).



Sundous Khamayseh is a graduate student at Palestine Polytechnic University. She received her B.Sc. degree in Information Technology and Communications from Al-Quds Open University, Hebron, Palestine, in 2014. In 2016 she joined the M.Sc. program in Informatics at Palestine Polytechnic University. She is currently fi-

nalizing her M.Sc. degree.

E-mail: 166007@ppu.edu
College of IT and Computer Engineering
Palestine Polytechnic University
Hebron, Palestine



Alaa Halawani received the B.Sc. degree in Computer Systems Engineering from Palestine Polytechnic University, Hebron, Palestine, in 1996, the M.Sc. degree in Electrical Engineering (computer major) from the Jordan University of Science and Technology, Irbid, Jordan, in 2000, and the Ph.D. degree from the Chair of Pattern Recognition and Image Processing, University of Freiburg, Freiburg, Germany, in 2006. From 2000 to 2001, he was a lecturer at the Electrical and Computer

Engineering Department, Palestine Polytechnic University, where he rejoined as an assistant professor in 2006. In 2009, he started a two-year Postdoctoral Fellowship with the Department of Applied Physics and Electronics, Umeå University, Umeå, Sweden. He continued in the same department as an assistant professor until 2016. Between 2017 and 2018 he worked as an Algorithm Development Engineer at Fingerprint Cards AB, Gothenburg, Sweden. He returned to Palestine in August 2018 and rejoined Palestine Polytechnic University as an assistant professor at the department of computer engineering. His current research interests include artificial intelligence, machine learning, biometrics, feature extraction, and object detection and recognition.

E-mail: ahalawani@ppu.edu
Intelligent Systems Research Center
College of IT and Computer Engineering
Palestine Polytechnic University
Hebron, Palestine