

Automatic network recognition by feature extraction: a case study in the ISM band

Maria-Gabriella Di Benedetto, *Senior Member, IEEE*, Stefano Boldrini, Carmen Juana Martin Martin, and Jesus Roldan Diaz

Abstract— Automatic network recognition offers a promising framework for the integration of the cognitive concept at the network layer. This work addresses the problem of automatic classification of technologies operating in the ISM band, with particular focus on Wi-Fi vs. Bluetooth recognition. The proposed classifier is based on feature extraction related to time-varying patterns of packet sequences, i.e. MAC layer procedures, and adopts different linear classification algorithms. Results of classification confirmed the ability to reveal both technologies based on Mac layer feature identification.

Index Terms—Cognitive networking, network discovery, automatic network classification

I. INTRODUCTION

This work is framed under the umbrella of the AIR-AWARE Project, developed to achieve classification amongst technologies and interference entities operating over the ISM band. This project aims at creating a black box — the AIR-AWARE module — capable of classifying technologies, as well as different types of interference in play.

Such classification is important for cognitive mechanisms to be implemented in the network, given the numerous commercial technologies operate in this range of the spectrum, such as:

- IEEE 802.11 networks: 2.4 GHz and 5.8 GHz bands;
- Bluetooth: 2.4 GHz band, using Frequency Hopping and any of 79 available channels;
- HIPERLAN [High Performance Radio LAN]: European alternative to IEEE 802.11, operating in the 5.8 GHz band to avoid interference entities at 2.4 GHz;
- Closed-Circuit TV: Security cameras at 2.4 GHz;
- ZigBee IEEE 802.15.4: 2.4 GHz range band;
- Wireless Mouse and Keyboard: 2.4 GHz band.

Probably, the most common interference at 2.4 GHz comes from the microwave oven, followed by baby monitors and cordless Wi-Fi phones, and the interference produced by DECT standard cordless phones, operating in the 1.9 GHz band. When in use, these devices can compromise the quality of an IEEE 802.11 network.

The final goal is to propose a classification strategy based on information regarding protocol layers above the physical one (PHY). In particular, the objective is to identify MAC sublayer [1,2] specific features for each of the above

technologies. Previous work, as for example [3], has addressed a similar problem, by classifying Wi-Fi vs. Bluetooth, using a spectrum sensing procedure based on distributed detection theory. The present work extends beyond previous investigations by considering Wi-Fi real traffic captures, and by focusing feature extraction and classification on MAC sublayer characteristics, leading to simplicity and computational efficiency.

In this work, feature identification lays its foundation on the observation that packet exchange patterns are technology-specific. As such, by identifying patterns clues, network recognition can be achieved. In particular, the study focuses on the ISM 2.4 GHz band. A first step consists in providing the AIR-AWARE module with a device capable of sensing the spectrum with a good time resolution. Albeit this piece of hardware will not be in a position to demodulate and decode the distinct signals in the air, it will enable AIR-AWARE to statistically study temporization of presence or absence of energy - against pre-defined thresholds - and therefore decode the packet sequence structure, in real time. Figure 1 illustrates the schematic of the cognitive energy detector, as well as software modules for the recognition of each technology embedded onto the device.

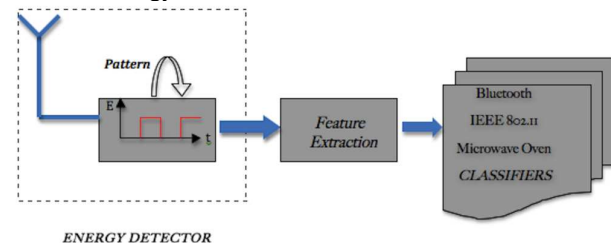


Figure 1 - The AIR-AWARE module

The study presented in this paper focuses on 802.11 (Wi-Fi) and 802.15.1 (Bluetooth) network recognition. To achieve recognition of both technologies, a set of features will be proposed. These features will serve as input for a linear classifier, that automatically performs classification among these two technologies.

The paper is organized as follows. Section II contains the description of the experimental set-up, i.e. the environment and tools by which data were collected and generated. The packet data-base is described in Section III, and particular focus on the Wi-Fi data-base (Section III.A) vs. the Bluetooth data-base (Section III.B). The classification algorithms are all linear classifiers as analyzed in Section IV; four different approaches are taken into consideration, i.e. Perceptron, Pocket, Least Mean Squares (LMS), and Sum of Error Squares Estimation (SOE). Experimental results on automatic classification of Wi-Fi vs. Bluetooth are reported in Section V. These results are discussed in Section VI, which also contains the conclusions of this study.

Manuscript submitted March 21, 2010. This work was supported in part by the European Commission in the framework of COST Action IC0902: Cognitive Radio and Networking for Cooperative Coexistence of Heterogeneous Wireless Networks.

The authors are with the Info-Com Department, School of Engineering, University of Rome La Sapienza, Via Eudossiana 18, 00184, Rome, Italy. E-mail address: dibenedetto@newyork.ing.uniroma1.it.

II. EXPERIMENTAL SET-UP

This Section describes the environment and tools used for collecting the reference data and build-up the data-base.

As for Wi-Fi, real data packets were detected within experimental measurements, using a packet capturing device (“Sniffer Station”). These measurements were carried out in the ACTS laboratory, located on the 2nd floor of the Information and Communication Department (InfoCom Dept.), in the Faculty of Engineering of the University of Rome “La Sapienza”, Rome, Italy.

A long corridor, with offices and laboratories on both sides, composes the 2nd floor of the building, the ACTS lab being one of these laboratories. The (main) Access Point, to whose frequency channel the sniffer was tuned to, is located in this corridor, near the ceiling, at a height of about 2.5 - 3 meters from the ground, and at a distance of about 4 - 5 meters from the ACTS laboratory door.

The computers were placed on a fixed location on a table, at a height of about 1 - 1.5 meters from the ground, at a distance of about 1.5 meters from the nearest wall and of 2 meters from a door, that leads to the corridor mentioned above, i.e. the distance between the computers and the Access Point was about 5 - 6 meters. The distance between each computer was about 0.5 meters. On this same floor there are about 20 additional offices, containing about 2 - 3 computers each. Since we expected that any of these may be connected to the main Access Point during the measurements, and in the aim of capturing “clean data” for the training set, we performed measurements at night after verification that no other computer was active during captures.

There are also other Access Points in the same building, but tuned to other frequency channels. However, some packets of these other Access Points may have been captured. That occurs because of the frequency channels partial overlapping, that is expected in the IEEE 802.11 Standard. To this respect, collected data were carefully examined, in order to ensure again that only packets of the relevant network were captured.

Four computers were used to perform measurements. Three computers were used to generate traffic in heterogeneous traffic conditions. The fourth, used as the Sniffer Station, had the following technical specifications:

- Model: *HP Pavilion DV2320US*
- Processor: *AMD Turion 64 X2 TL 56*
- RAM Memory: *2GB*
- Network Wireless Adapter: *AirForce 54g 802.11 a/b/g PCI Express*, with a *BCM4311* chipset

Operating System was *Linux Ubuntu 9.10*, with the real-time kernel *2.6.31-9-rt*. The driver used for the Broadcom 4311 chipset was the *b43*. The main Access Point was a *Cisco Aironet 12xx 802.11 b*.

As for Bluetooth packets, in this first phase we decided to design a complete simulator by which Bluetooth packets were obtained. By doing so, one of the two data packet stream was fully controllable by software.

III. PACKET DATA-BASE

A. Capturing Wi-Fi packets

In order to capture Wi-Fi Packets, a packet capturing application was developed using the Java library *jpcap* [4]. The Wi-Fi standard foresees a logic unit, the MAC PDU,

while the unit sent over the air interface, called PPDU, includes beyond the MAC PDU, two additional fields (preamble and header). The driver used for the 802.11 network adapter enabled to intercept data of every PPDU [1] within the Sniffer range by means of its monitor mode [5]. This driver was also compatible with the radiotap header [6], which provides information such as preamble type, or time of arrival of first bit, for the captured MAC PDU.

Experiments were made in three different conditions: one, two, and three computers (nodes) associated to the Access Point. In all conditions, each computer was downloading at least two files or processing a video call; this way, we could ensure a high traffic scenario. Two 1000-packet captures were run for each condition.

B. Generation of Bluetooth packets

Bluetooth simulated packets were generated using MATLAB. The reference standard for this simulation is the IEEE Standard 802.15.1 – 2005 [2], i.e. bitrate of 1 *Mbit/s*. Piconets of two devices in connection state (one master and one slave) were considered. The two devices send their packets alternately: one device (the master, for example) sends its data packets, and for every received packet, the other device (the slave) sends back an acknowledgement. Data packets sent by the master can occupy 1, 3 or 5 time slots (where the time slot is 625 μ s), according to their length, whereas acknowledgement packets (NULL packets, with a fixed length of 126 *bits*) occupy 1 time slot.

Two different scenarios were considered:

- Scenario 1: data packets occupy only 1 Time Slot
- Scenario 2:
 - 80% Data Packets occupy 1 Time Slot
 - 15% Data Packets occupy 3 Time Slots
 - 5% Data Packets occupy 5 Time Slots

In every scenario, 70 % of the data packets have a duration that is fixed by the 802.15.1 standard specification to the values shown in Table I. The duration of the remaining 30% is uniformly distributed between minimum and maximum values (see Table I).

According to the standard, for every packet arrival time a jitter of $\pm 10\mu$ s has been set, to consider imperfect synchronization between the two devices. The jitter was modeled by a Gaussian distribution with zero mean and standard deviation $\sigma=10/3\mu$ s; given the model, 99% of jitter values fell within a $\pm 10\mu$ s interval, while the remaining 1% exceeded this interval and were readjusted in order to meet the standard specifications.

TABLE I
BLUETOOTH STANDARD SPECIFICATION

	Fixed duration	Min. Duration	Max. Duration
Time slot	625 μ s		
1-time-slot-packet		126 μ s	366 μ s
3-time-slot-packet		1250 μ s	1622 μ s
5-time-slot-packet		2500 μ s	2870 μ s
NULL packet	126 μ s		

IV. AUTOMATIC CLASSIFICATION

We started by designing a generic linear classifier that was able to distinguish amongst C classes, each class being characterized by M features. In general, a linear classifier divides the feature space into C regions; this division comes about through calculation of discrimination functions [7] that characterize the region of the space where each class is located:

$$g_j(x) = w_{0,j} + \sum_{i=1}^M w_{i,j} x_i, \quad j = 1, 2, \dots, C, \quad (1)$$

where $w = [w_0, w_1, \dots, w_M]$ is known as the weight vector, and $X = [x_1, x_2, \dots, x_M]$ is a point on the decision hyperplane.

The classifier objective is to find each discrimination function, and generate a decision using the major score criterion. This score represents a measure of similarity between the object and each class. The classification module was implemented on MATLAB, based on four classification methods selected because of their simplicity and computational appeal:

- Perceptron. One of the oldest methods, its convergence depends on the separability of the classes. The idea is to calculate the weight vector through an iterative method, formulated in this way [8]:

$$w(t+1) = w(t) - \rho_t \frac{\partial J(w)}{\partial w} \Big|_{w=w(t)}, \quad (2)$$

where, as displayed in the equation, there is a learning coefficient ρ_t and the minimization of a cost function is required, defined in this case as :

$$J(w) = \sum_{X \in Y} \delta_X w^T X, \quad (3)$$

where Y is the subset of training vectors which are misclassified by the weight vector w , and δX is a coefficient that takes value equal to -1 if $X \in \text{Class1}$, and equal to 1 if $X \in \text{Class2}$ or vice-versa. In the multiclass case, δX equal to -1 for all classes.

- Pocket. A version of Perceptron, that provides a better behavior when separability of the classes is not totally guaranteed. The key [9] is to run perceptron learning, while keeping an extra set of weights in the pocket. By this way, if a new iteration provides a weight vector that classifies a greater number of training vectors than its predecessor, then that last is chosen and used in the next step; otherwise, the vector obtained in the preceding iteration is maintained as the weight vector.
- Least Mean Squares Method (LMS). Similar to Perceptron, but the cost function, that is minimized corresponds to the error. Here, the weight vector is computed so as to minimize the Mean Square Error (MSE) between true and desired output (y) [8]:

$$J(w) = E \left[|y - X^T w|^2 \right], \quad (4)$$

where the mean value (that cannot be generated due to the lack of statistical data), is replaced by samples obtained during experimentation. The weight vector is therefore obtained according to the following rule:

$$w(k) = w(k-1) + \rho_k X_k (y_k - X_k^T w(k-1)), \quad (5)$$

where ρ_k is a learning coefficient.

- Sum of Errors Squares Estimation (SOE). Similar to LMS, here the cost function takes the form of the sum of quadratic error for each of the N training vectors X :

$$J(w) = \sum_{k=1}^N (y_k - X_k^T w)^2 = \sum_{k=1}^N e_k^2, \quad (6)$$

where the calculation of weight vectors is the simple matrix operation [8] shown below:

$$\sum_{k=1}^N X_k (y_k - X_k^T w) = 0 \Rightarrow \left(\sum_{k=1}^N X_k X_k^T \right) w = \sum_{k=1}^N (X_k y_k) \quad (7)$$

In order to compute both optimal w and desired values y , the Ho-Kashyap algorithm was used [10].

V. EXPERIMENTATION

A. Training set and feature extraction

As described in Section III, in the Wi-Fi case, six (6) 1000-packet captures formed the training set. In the Bluetooth case, simulated MATLAB captures consisted in two 6000-packet sequences corresponding to Scenarios 1 and 2.

The first proposed feature is the time interval between PPDU, defined in [1] as Short Inter Frame Space (SIFS) corresponding to silence gaps on the medium when DATA-ACK procedures are in play. Most IFS timings are fixed and independent of the bitrate at the PHY [1]. Of all existing IFS types, SIFS has a nominal value of $10\mu s$ for the ISM 2.4GHz band, and is the most likely to occur in a scenario with medium to high traffic; it is usually used by a node responding to any polling, and always prior to: a) transmission of an ACK frame; b) a CTS frame; c) a second or subsequent PPDU of a fragment burst. For automatically extracting the SIFS and estimating its statistical behavior, SIFS was differentiated from a non-SIFS when two consecutive PPDU durations were such that: $0.6 * PPDU_{ith} > PPDU_{ith+1}$. The second proposed feature is the duration of the longest packet considering all the packets between two consecutive silence gaps, previously identified as SIFS. Note that both proposed features are extremely simple and easy to extract thanks to simplest hardware such as an energy detector. Figure 2 illustrates the feature plane for both Wi-Fi (real traffic and Bluetooth (simulated traffic) training set. Bluetooth data correspond to Scenario 1 (single-slot case).

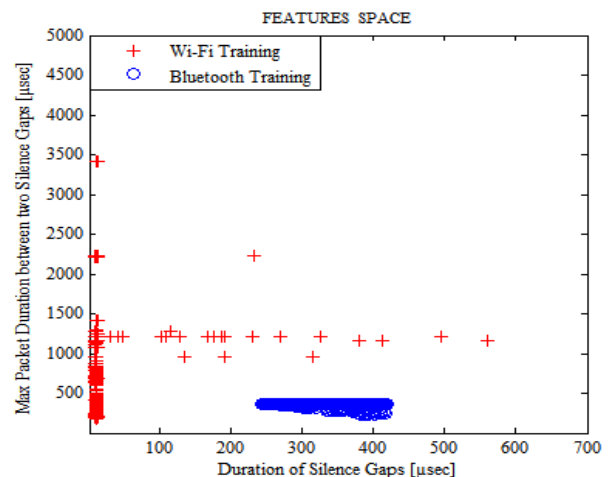


Figure 2 - Feature Plane for Wi-Fi and Bluetooth single-slot
Figure 3 shows the feature plane in the multi-slot Bluetooth Communication scenario (Scenario 2).

Note the presence of a few Wi-Fi points invading the Bluetooth "zone". Capture file revision indicated that these

corresponded to non-SIFS, i.e. erroneously estimated SIFS. These points were however less than 1% of total.

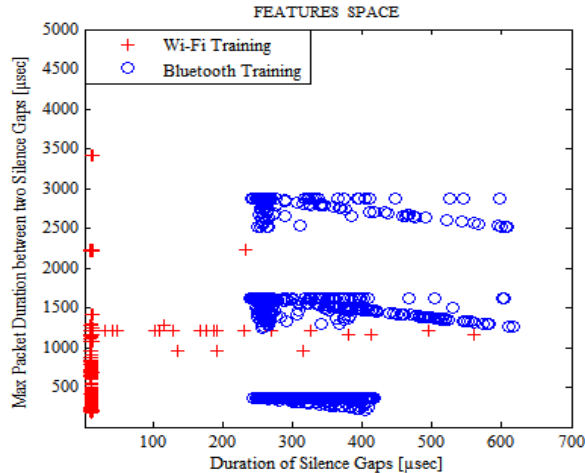


Figure 3 - Feature Plane for Wi-Fi and Bluetooth multi-slot

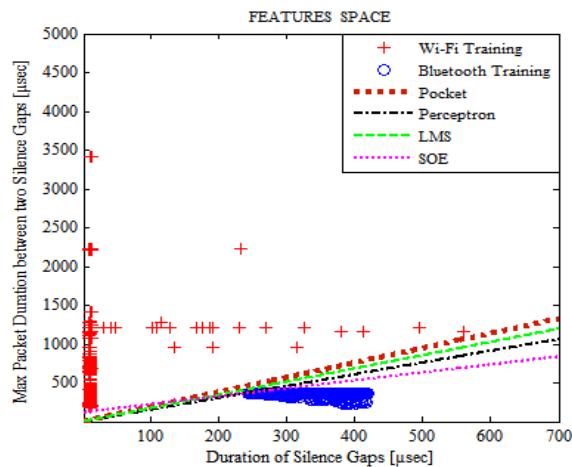


Figure 4 - Automatic classification of Wi-Fi vs. Bluetooth single-slot

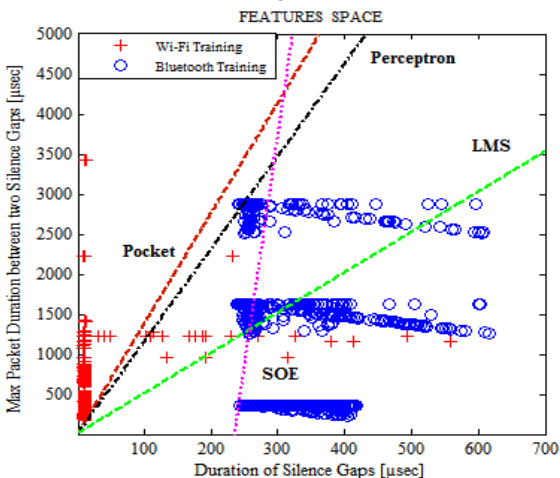


Figure 5 - Automatic classification of Wi-Fi vs. Bluetooth multi-slot

B. Results of automatic classification

The four classification algorithms were run over the training sets of Figs. 2 and 3. Results are reported on Figs. 4 and 5, for the single-slot vs. multi-slot Bluetooth cases. The classifiers were then applied to data not belonging to the training sets, i.e. a new 1000-packet Wi-Fi capture (1.4

seconds capture duration), and two new 1000-packets Bluetooth simulations (Scenarios 1 and 2) were generated (each around 0.7 seconds long). Results of classification percentage of Wi-Fi vs. Bluetooth (single-slot case), when the input to the classifier is formed by either Wi-Fi captures or Bluetooth sequences of packets are reported in Tables II and III, for the single vs. multi-slot Bluetooth, respectively.

TABLE II
CLASSIFICATION RESULTS

	Classifier Input Network	Classification into Wi-Fi	Classification into single-slot Bluetooth
Pocket	Bluetooth	0% [0/456]	100% [456/456]
Pocket	Wi-Fi	100% [352/352]	0% [0/352]
Perceptron	Bluetooth	0% [0/456]	100% [456/456]
Perceptron	Wi-Fi	100% [352/352]	0% [0/352]
LMS	Bluetooth	0% [0/456]	100% [456/456]
LMS	Wi-Fi	100% [352/352]	0% [0/352]
SOE	Bluetooth	0% [0/456]	100% [456/456]
SOE	Wi-Fi	100% [352/352]	0% [0/352]

A mixed input to the classifiers (multi-network environment) was then considered. Given that the Wi-Fi capture is on real traffic, while the Bluetooth streams were simulated, the mixture could be controlled by software. In particular, three different mixes were generated: a) predominant Wi-Fi (1000 Wi-Fi packets vs. 200 Bluetooth packets); b) balanced (1000 Wi-Fi packets vs. 1000 bluetooth packets); c) Bluetooth predominant (1000 Wi-Fi vs. 2000 Bluetooth packets). All Bluetooth sequences were multi-slot. Wi-Fi captures were 1.4 seconds long, while Bluetooth simulations lasted 0.16, 0.75 and 1.6 seconds. Due to differences in the duration of captures only partial overlapping in the combined packet sequences was achieved. Results for this test are displayed on Table IV.

TABLE III
CLASSIFICATION RESULTS

	Classifier Input Network	Classification into Wi-Fi	Classification into multi-slot Bluetooth
Pocket	Bluetooth	0% [0/462]	100% [462/462]
Pocket	Wi-Fi	98.86% [348/352]	1.14% [4/352]
Perceptron	Bluetooth	0.43% [2/462]	99.57% [460/462]
Perceptron	Wi-Fi	98.86% [348/352]	1.14% [4/352]
LMS	Bluetooth	34.85% [161/462]	65.15% [301/462]
LMS	Wi-Fi	99.43% [350/352]	0.57% [2/352]
SOE	Bluetooth	29.87% [138/462]	70.13% [324/462]
SOE	Wi-Fi	99.72% [351/352]	0.28% [1/352]

TABLE IV
CLASSIFICATION RESULTS
MULTI-NETWORK ENVIRONMENT

Classifier	Input Network	Classification into Wi-Fi	Classification into multi-slot Bluetooth
Pocket	Bluetooth predominant	17.10% [133/778]	82.90% [645/778]
Pocket	Wi-Fi predominant	86.07% [315/366]	13.93% [51/366]
Pocket	Balanced	41.34% [210/508]	58.66% [298/508]
Perceptron	Bluetooth predominant	17.22% [134/778]	82.78% [644/778]
Perceptron	Wi-Fi predominant	86.07% [315/366]	13.93% [51/366]
Perceptron	Balanced	41.53% [211/508]	58.47% [297/508]
LMS	Bluetooth predominant	37.79% [294/778]	62.21% [484/778]
LMS	Wi-Fi predominant	90.16% [330/366]	9.84% [36/366]
LMS	Balanced	56.89% [289/508]	43.11% [219/508]
SOE	Bluetooth predominant	36.89% [287/778]	63.11% [491/778]
SOE	Wi-Fi predominant	90.71% [332/366]	9.29% [34/366]
SOE	Balanced	56.10% [285/508]	43.90% [223/508]

The construction of the decision hyperplane is a one-time occurrence, and therefore execution time required by the different algorithms was not deemed relevant.

VI. DISCUSSION OF RESULTS AND FUTURE DIRECTIONS

As described in the above Section, network classification of Wi-Fi vs. Bluetooth was attempted based on the definition of two features: the maximum packet duration between two silence gaps, and duration of silence gaps. Four different classification algorithms were used: Pocket, Perceptron, LMS, and SOE

Results of classification showed that:

1) For the Wi-Fi vs. single-slot Bluetooth case (Table II), all proposed classifiers achieved perfect classification into the two classes, when one traffic stream (either Wi-Fi or Bluetooth) was given as input to the classifier. This result shows that the selected features were appropriate since they completely identify these two classes.

2) For the Wi-Fi vs. multi-slot Bluetooth case (Table III), classification is not as perfect as in the previous case, and depends upon classification algorithm as well as input data to the classifier. Among all the proposed classification strategies, Pocket and Perceptron emerge as the most successful and reliable, leading to a correct classification rate greater than 98%.

3) Data in Table IV speak to the adequacy of the classifiers in environments with heavy predominance of one technology, by their ability to reveal both technologies in each case. This ability is shown by comparing results of Pocket reported by Tables III and IV. As shown by tables, only Pocket and Perceptron are capable of performing a reliable classification. Note that these classifiers were always capable of providing as output, the pre-dominant network, and moreover, the rate of classification follows the

trend in the proportion between both technologies packets in the observation sequence. When the traffic flows are balanced, the classifier seems to follow a “50-50” “win-win” rule, by outputting balanced classification decisions.

Future work will focus on investigating whether the selected features extend beyond the present case of two technologies in the ISM band. In particular, the AIR-AWARE project will proceed by incorporating the IEEE 802.15.4 technology (ZigBee) [11] into the set of possible classes. Preliminary investigations, based on the analysis of the 802.15.4 standard specifications, show that SIFS is also defined for ZigBee networks, with a nominal value of $192\mu s$ [11] in the ISM 2.4 GHz band. This value compared to extracted features on this paper experiments, should allow the classification algorithms to obtain good separation for all three classes (Wi-Fi vs. Bluetooth vs. ZigBee).

ACKNOWLEDGEMENT

This work was supported in part by COST Action IC0902 "Cognitive Radio and Networking for Cooperative Coexistence of Heterogeneous Wireless Networks".

REFERENCES

- [1] IEEE Std 802.11 – 2007, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 12 June 2007
- [2] IEEE Std 802.15.1 – 2005, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs), 14 June 2005.
- [3] Gandetto M. and Regazzoni C., “Spectrum Sensing: A Distributed Approach for Cognitive Terminals,” *IEEE Journal on selected areas in communications*, Vol.25 (3), 2007.
- [4] <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/>
- [5] http://en.wikipedia.org/wiki/Monitor_mode/
- [6] <http://www.radiotap.org/>
- [7] Duda R.O., Hart P. E., and Stork D.G., *Pattern classification*, 2^o Ed., Wiley-Interscience, 2004.
- [8] Theodoridis S. and Koutroumbas K., *Pattern recognition*, 4^o Ed., Elsevier Inc., 2009.
- [9] Gallant S. I., Perceptron-Based Learning Algorithms, *IEEE Transactions on neural networks*, Vol. 1(2), 1990.
- [10] Ho Y.H. and Kashyap R.L. “An algorithm for linear inequalities and its applications,” *IEEE Transactions on Electronic Computers*, Vol.14(5), 1965.
- [11] IEEE Std 802.15.4 – 2006, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 8 September 2006.