Learning from Errors: Detecting Cross-Technology Interference in WiFi Networks

Daniele Croce[1,2], Domenico Garlisi[1,2], Fabrizio Giuliano[1,2], Nicola Inzerillo[1], Ilenia Tinnirello[1]
[1]DEIM, Università di Palermo, viale delle scienze ed. 9 - 90128 Palermo, Italy
[2]CNIT Consortium, Viale G.P. Usberti, 181/A - 43124 Parma, Italy
name.surname@unipa.it

***Abstract*—In this paper we show that inter-technology interference can be recognized using commodity WiFi devices by monitoring the statistics of receiver errors. Indeed, while for WiFi standard frames the error probability varies during the frame reception in different frame fields (PHY, MAC headers, payloads) protected with heterogeneous coding, errors may appear randomly at any point during the time the demodulator is trying to receive an exogenous interfering signal. We thus detect and identify cross-technology interference on off-the-shelf WiFi cards by monitoring the sequence of receiver errors (bad PLCP, bad FCS, invalid headers, etc.) and propose two methods to recognize the source of interference based on Artificial Neural Networks and hidden Markov chains. The result is quite impressive, reaching an average accuracy of over 95% in recognizing ZigBee, Microwave and LTE (in unlicensed spectrum) interference.**

***Index Terms*—Wireless LAN, Interference, Neural Networks, Hidden Markov models.**

I. Introduction

Nowadays, we are witnessing an impressive success of IEEE 802.11 technology, better known as WiFi, for supporting the growing demand of wireless broadband connectivity. Public WiFi networks are deployed worldwide, with more than 50% of the total mobile traffic carried by WiFi. The availability of WiFi networks is often considered as a commodity service driving immense economic value, and the unlicensed spectrum is becoming one of society's most valuable resources. Although WiFi is a dominant communication technology in this spectrum, many other low range technologies coexist in unlicensed ISM (Industrial, Scientific and Medical) bands for supporting several vertical applications, such as building automation, smart metering systems, health care monitoring, surveillance systems, game remote controllers and so on. Moreover, cellular technologies are trying to extend their operation to ISM bands for increasing their capacity. Two different solutions have been envisioned by 3GPP (Third Generation Partnership Project) in ISM bands, referred to as Long Term Evolution (LTE) with Licensed Assisted Access (LAA) [1] and LTE-Unlicensed (LTE-U) [2], which work respectively, with and without the listen-before-talk mechanism.

Although in WiFi carrier sense and adaptive modulation mechanisms have been included, it has been shown that serious performance impairments can arise in presence of exogenous interfering signals due to different technologies. For example, in [3] it is shown that the capacity of a good WiFi link can be reduced to zero in presence of analog phones, video cameras, or sensors based on IEEE 802.15.4 technology [4], [5], while other devices such as a Xbox controller and a microwave oven can half the throughput. About the interference with cellular technologies, several research studies are trying to characterize the impact on LTE transmissions on WiFi performance. Preliminary empirical and simulation results [6] show that WiFi performance can be critically affected even when LTE links operate at the minimum bandwidth of 1.4 MHz.

In this scenario, we argue that a critical aspect for WiFi networks is enabling the correct identification of coexistence problems with other technologies, which in turn can serve as basis for some cross-technology coordination mechanisms. While state-of-the-art solutions for detecting coexistence problems in WiFi networks have mainly worked on the characterization of RSSI samples observed at different frequencies and with varying temporal gaps, in this work we propose to simply monitor the reception errors of commodity WiFi cards, and then apply a classification mechanism devised to recognize typical error sequences due to heterogeneous interference sources. In other words, our mechanism is based on the analysis of the *error domain*, i.e. on the classification of error events and on the time intervals between their occurrence. Statistics of these errors are widely available on many WiFi *commodity* cards and can be easily exploited to improve interference detection and troubleshooting algorithms of wireless networks. Although in this work we focus on three interference sources, namely ZigBee, LTE and microwave ovens, our solution does not depend on the type of technology, but only requires a training phase based on the events generated in presence of a controlled source of interference.

Our contribution is twofold: we propose a new mechanism for extracting observable features to be used for interference classification, and we design two different classifiers, exploiting domain-specific information on the WiFi receiver behaviors. More into details, we implement and compare two different classification techniques: a hidden Markov chain (following the initial approach of [7]) and an Artificial Neural Network (ANN), extending the analysis to the emerging *LTE in unlicensed spectrum*. Experimental results show that our proposed solutions provide excellent results, up to an average 95% of accuracy.

After a brief review of some literature solutions for detecting

This work has been partially supported by EU funded H2020 research projects symbIoTe, Grant No. 688156, and WiSHFUL, Grant No. 645274.

and reacting to interference, which also motivate our work (section II), we provide necessary background information on the competing technologies (section III) and we analyze the theoretical and experimental error rates caused by this interference (section IV). The Hidden Markov Model (HMM) and the ANN model used for classifications are presented in section V and VI. Finally, we discuss the two solutions in section VII, while section VIII concludes the paper and proposes possible future extensions.

II. Related Work

Effects of cross-technology interference. Performance degradation of WiFi networks in presence of cross-technology interference has been widely studied in recent literature. Indeed, since each technology implements different mechanisms and protocols for reacting to interference, it is not obvious to predict WiFi performance in case of coexistence with other technologies. While several works have focused on ZigBee throughput degradation in presence of WiFi, performance reductions can happen also in WiFi networks [4], [5]. The possible reasons are that some WiFi implementations are unable to detect non-WiFi signals [8] or because of the different timings to perform CSMA/CA [9], [10].

LTE transmissions in unlicensed bands can have a deep impact on WiFi performance, even when the listen-before-talk mechanism is adopted [11]. Although most of the current studies are based on simulations (see for example [6]), preliminary empirical results show that WiFi performance can be critically affected even when LTE links operate at the minimum bandwidth of 1.4 MHz. This is due to the fact that WiFi nodes are generally able to sense LTE nodes operating in ISM bands and therefore are prevented from accessing the medium in case of LTE transmissions. Solutions based on duty-cycle muting or blank subframes [12] can mitigate WiFi throughput degradation, but they are unilaterally controlled by LTE nodes. Advanced PHY solutions can also be envisioned for improving coexistence. For example, in [13] a mechanism to decode WiFi MIMO transmissions under strong LTE interference is proposed using a GNU Radio testbed with USRP devices.

Coordination strategies. A simple solution for improving coexistence is introducing some forms of coordination mechanisms among technologies. Early solutions which detect interference and simply choose a better channel to transmit are becoming not viable because of the increasing number of technologies and applications in the market [14]. Other solutions rely on complex and expensive radio transceivers to communicate with multiple protocols and different technologies [15], or increase the robustness of the transmission with use of error correction codes or multiple antennas [16]. Different approaches have considered the possibility to introduce beacon transmissions and dynamic quiet periods [17] or, more specific to WiFi and ZigBee, several solutions have been proposed based on TDMA-like schemes [18], indirect forms of coordination opportunistically exploiting WiFi temporal spaces [8], channel reservations based on an additional ZigBee channel for making the channel busy for WiFi stations [9], or by means of simple forms of adaptive redundancy [10]. Regarding LTE, it has been proposed to improve coexistence with WiFi by introducing a centralized controller and tune LTE parameters based on WiFi traffic conditions [19], [20]. However, this requires a global authority which is difficult to implement in practice.

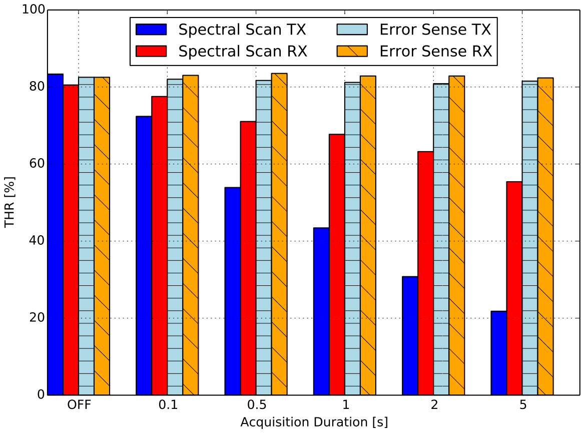


Fig. 1. Comparison between Spectral Scan and Error-Sense

Detection of cross-technology interference. The correct identification of the interference problem is an essential task to enable any form of coexistence mechanism. Some solutions exploit dedicated hardware, for example, to simultaneously demodulate the signal according to different PHY specifications [21]. Other approaches which do not implement a complete per-technology demodulator are based on cyclostationary signal analysis and blind signal detection [22] or other spectrum sensing techniques [23]. Although these approaches are very effective, they require specialized hardware (basically, a spectrum and signal analyzer). The possibility to identify WiFi signals by using commodity ZigBee nodes have been explored in [24] and [25]. The approach proposed in [24] is based on the analysis of temporal samples of link quality indicators and RSSI values, as well as on the identification of the portions of ZigBee corrupted packets to be compared with the typical WiFi transmission times. A similar temporal analysis is carried out in [25] with the aim to find periodic interference signatures caused by WiFi beacons and enabling the detection of WiFi networks by using a low-power monitoring interface. In [26], network level information (such as packet loss, transmission retires and FCS errors) together with peer collaboration are exploited to identify the root cause of WiFi performance degradation. However, the solution is intrusive and complex to deploy because it requires the use of several nodes and the injection of packets as active probes in the network.

In [3], a commodity 802.11n card by Atheros is used to perform a spectral analysis of the channel signals, by collecting RSSI samples at different sub-carriers. The results of the spectral scan are used for recognizing different frequency and temporal signatures of the signals and detecting the interfering technologies which most likely produced those signatures. The approach is very effective and generalizable, although the extraction of some features requires to monitor the channel for time intervals of a few seconds or more. At the end of the monitoring interval, the scheme is able to identify the technologies that have been active, but interference detection is not performed by classifying each interference event alone – as considered in this work. Moreover, running the spectral scan function degrades the throughput achievable by the card. This is due to the hardware reconfigurations needed for scanning the spectrum or transmitting and receiving a frame, which introduce some latencies. Figure 1 quantifies the throughput degradation due to the spectral scan, when this function is periodically activated for a variable time (from 0.1s to 5s) and suspended for 1s before the subsequent call. For comparison, we also plot the throughput results obtained when the monitoring WiFi card tracks the receiver errors rather than the RSSI samples. The figure clearly shows that, differently from running the spectral scan function, monitoring the receiver errors does not have an impact on the throughput.

Summarizing, in this paper we propose an interference recognition mechanism based on error monitoring: rather than characterizing the frequency and time signatures of *external* interfering signals, we propose a classifier that is able to identify the interfering transmissions from the error events detected by a monitoring WiFi card. The approach is much less intrusive than scanning the spectrum, because it does not require specific hardware configurations.

III. Background

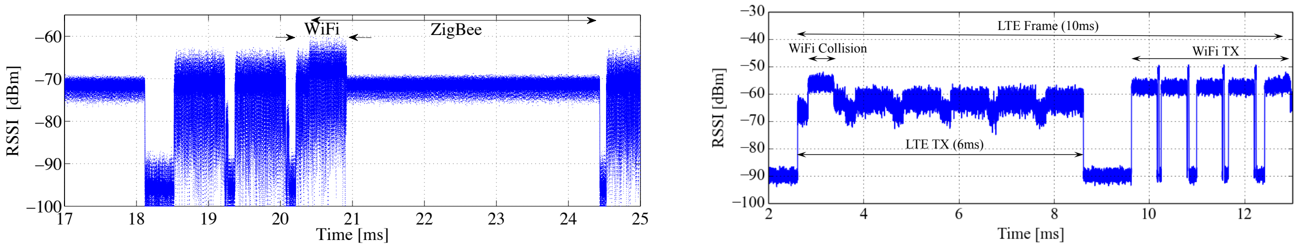
In this section we briefly recall some key aspects of the MAC/PHY layers in WiFi, ZigBee and LTE that affect the power of cross-technology interference and the typical timings of transmissions and channel idle intervals.

Interference power. WiFi and LTE transmissions are typically performed at a maximum power of 15 or $20dBm$, while ZigBee transmissions can span in the range $[-25, 0]dBm$. LTE transmission power is modulated because of power control mechanisms, which are usually not implemented in WiFi and ZigBee. Additionally, each WiFi channel is 20 MHz wide and is spaced of 5 MHz from the adjacent ones. ZigBee channels have only 2 MHz of bandwidth with 3 MHz of inter-channel gap bands (i.e. the center frequencies maintain the spacing of 5 MHz from the adjacent channels). It follows that four ZigBee channels are entirely included in a WiFi channel. LTE center frequencies in ISM bands coincide with WiFi ones, with bandwidth typically of 5 MHz and spanning from 1.4 up to 20 MHz.

Transmission times. Since the three technologies have been defined for different applications, the frame size, the data rates and the channel access units considered by the standards are quite different. For WiFi and ZigBee, channel access is performed on a per-packet basis, i.e. transmission times correspond to the time required for completing the transmission of a packet (or an aggregation/fragmentation of packets). ZigBee packets are small, with a maximum payload of only 128 bytes. Bytes are organized into 4-bit symbols that are mapped into 16 pseudo-random sequences of 32-chip transmitted at 2 Mchip/s (i.e. 250 Kbps), which correspond to a frame transmission interval of about $4.5ms$ for the maximum frame size. WiFi frames are much longer, with a maximum frame size of 2358 bytes and multiple OFDM modulations and coding schemes available (in 802.11g from 6 Mbps up to 54 Mbps, which lead respectively to a maximum transmission time of about 3.2 ms and 0.37 ms). For LTE, the channel access is performed on the basis of resource block allocations, which are organized into sub-intervals lasting a fixed time of 1 ms within a frame of $10ms$. Packet transmissions are achieved by scheduling a given set of resource blocks in one or multiple consecutive frames. Although the total number of resource blocks used for each packet depends on the employed data rate and multiple rates are available (up to 25.2 Mbps for 5 MHz of bandwidth with 300 sub-carriers, 64-QAM modulation, and a symbol time of 71.4 μs), the channel occupancy time in each channel access is fixed according to the LTE frame structure.

Intervals between transmissions. Different channel access schemes are employed in WiFi, ZigBee and LTE for unlicensed bands. WiFi and ZigBee are mostly based on random access although channel sensing is performed with different granularity: ZigBee spends 128 μs for detecting the channel activity and 192 μs to switch from reception to transmission mode. Since WiFi slots are much shorter (9 μs), if a WiFi transmission is originated during this switching time, it cannot be detected by the ZigBee node. Figure 2-a shows a channel occupancy trace acquired by means of a USRP node in a network in which a WiFi node coexist with a ZigBee one. In the figure we clearly observe that each transmitter is characterized by a specific RSSI value and frame transmission time: WiFi frames occupy the channel for less than 1 ms with a RSSI value of -65 dBm, while ZigBee frames last 4 ms with a RSSI value of -72 dBm. The figure also shows a ZigBee transmission colliding with WiFi, because the WiFi frame is transmitted during the time spend by ZigBee for switching from sensing to transmission mode.

LTE transmissions in licensed bands are organized into frames of 10 ms that start at regular time intervals. For operating in unlicensed bands, two different adaptations have been envisioned: employing duty cycles for periodically suspending frame transmissions, while keeping the synchronization of time instants at which frame transmissions can start (LTE-U); employing listen-before-talk before transmitting each frame (LTE-LAA). In this second case, when the medium is sensed as busy, the deferral time is given by a fixed time of 10 ms for maintaining the synchronization of frame starting times (with the so called FBE mechanism) or it is given by a random slotted deferral time compensated by a varying channel occupancy time (with the so called LBE mechanism). In our work, we emulate both the LTE-U and LTE-LAA approach, by assuming that LTE frame transmissions can start only at regular time intervals. Figure 2-b gives an example of the interaction between an LTE-U transmission with 6 active and 4 silent subframes (i.e. 6 ms on and 4 ms off) and a WiFi



(a)

(b)

Fig. 2. Interference between technologies: temporal trace (RSSI samples) of WiFi-ZigBee (a) and WiFi-LTE collisions (b).

TABLE I
RECEIVER EVENTS REPORTED BY BCM4318 CARDS.

Receiver Event	Description
Too Long	Frame longer than 2346 bytes
Too Short	Frame shorter than 16 bytes
Invalid MAC Header	Protocol Version is not 0
Bad FCS	Checksum Failure on frame payload
Bad PLCP	Parity Check Failure on PLCP Header
Good PLCP	PLCP headers and Parity Check OK
Good FCS and RA match	Correct FCS matching the Receiver Address
Good FCS and not RA match	Correct FCS not matching the Receiver Address

station which tries to access the same channel: the figure shows that WiFi packets can collide with LTE and that part of the channel time is wasted due to the consequent backoff.

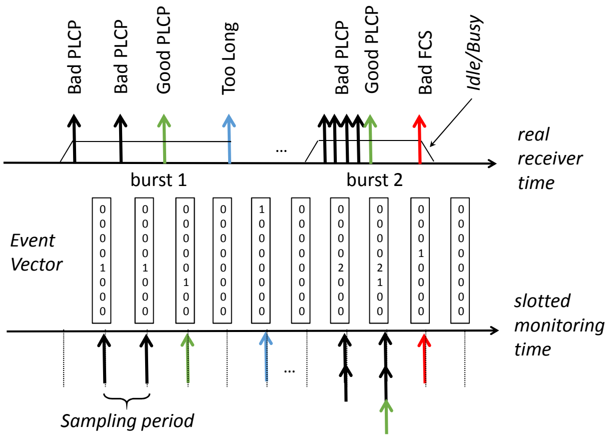


Fig. 3. Mapping between a real trace of receiver events and the time-slotted vectors generated by the monitoring process.

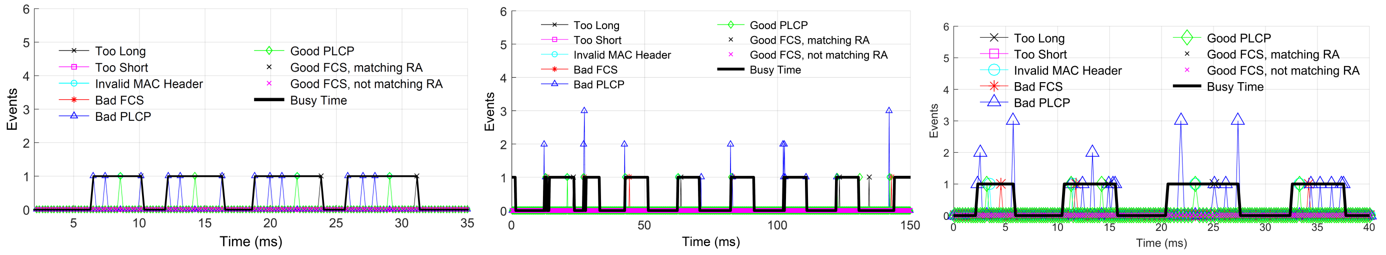
IV. ERROR ANALYSIS IN WIFI RECEIVERS

A. Monitoring Receiver Errors

In [7], we have shown that WiFi cards receiving non-WiFi modulated signals generate error patterns significantly different, in terms of occurrence probability and time intervals between consecutive errors, from the ones generated by collisions with other WiFi transmissions. In presence of wide-band noise and exogenous interference signals, WiFi receivers demodulate a sequence of completely random bits and try to interpret these bits according to the format of WiFi frames. Being all the bits random, the probability of having a specific error heavily depends on the format of the expected frame.

Most commercial WiFi cards track the occurrence of different *receiver events*, such as the start of a synchronization trial, the detection of wrong PLCP, the end of a frame transmission, etc., by means of specific counters implemented in internal registers. As a reference WiFi receiver, we considered a WiFi card (namely, Broadcom bcm4318) for which the card internal registers are documented and an interface for reading the register values is available [27]. Table I summarizes the receiver events tracked by this card. For producing a temporal trace of the receiver events, storing the ordered sequence of event type and occurrence time, we implemented a monitoring process devised to sample at regular intervals the receiver registers. Indeed, the event occurrence cannot be detected by the card host as an interrupt signal, but needs to be indirectly identified by comparing the state of the receiver registers in consecutive sampling times. We set a sampling interval equal to 250μs as a trade-off between detection delay and tracking complexity, while avoiding the overloading of the card to host interface. Because of the periodic sampling, multiple receiver events can occur in the same monitoring interval. Event samples are represented by a vector of eight components, whose value represents the counter of each different event type. We also sampled another card register, called busy time register, which does not track the occurrence of receiver events but rather the cumulative time during which the receiver remains active. The differences among consecutive values of the busy time register can be mapped into a logical idle/busy state of the channel as observed by the receiver.

Figure 3 shows the operation of our monitoring process: a real trace of receiver errors is mapped into a time series of event vectors, in which we can easily recognize consecutive *error bursts* due to the same interfering transmission. Error bursts can be originated for many different reasons: for example, a checksum failure can follow the detection of a good PLCP, or multiple (failed or not) synchronization trials are performed after a bad PLCP event. The total number of receiver events in a burst depends on the duration of the interfering transmission and on the receiver implementation, i.e. on the reset time required by the demodulator for performing consecutive synchronization trials. Finally, each burst can be



(a) (b) (c)

Fig. 4. Bursts of receiver events corresponding to the reception of ZigBee, Microwave and LTE-U interference respectively.

delimited by observing the time interval elapsed from the previous and next events and/or by considering the channel transitions from idle to busy and from busy to idle in the busy time register.

B. Temporal Analysis

Testbed. For our experiments, we set up a testbed at the University of Palermo and placed a monitoring WiFi node together with heterogeneous interfering sources. Four different interfering sources have been considered: a ZigBee transmitter, a LTE-U transmitter, a WiFi transmitter and a microwave oven. All nodes have been set to a few meters distance between each other and the transmitting nodes are programmed to work on different interfering and non-interfering channels. For ZigBee, the nodes used in our testbed are based on Microchip MRF24J40 transceiver and the frames are transmitted at 250kbps with a length of 128 bytes. WiFi transmitter has been implemented by using the same Broadcom card used by the WiFi monitoring node, with a frame length of 1500 bytes transmitted at 24 or 36 Mbps. The LTE-U transmitter, instead, was implemented on a SDR platform based on USRP B-210 and the srsLTE framework [28]. We considered a downlink interfering stream with 5 MHz of bandwidth and 300 subcarriers, centered on channel 11. Following the standard, the whole frame allocation time is $10ms$ composed of 10 subframes, which can be optionally empty.

Results. Figure 4 shows three traces of receiver events when receiving ZigBee, Microwave and LTE-U interference. Figure 4-a, for example, shows four ZigBee packets, with error events spaced approximately $1ms$ from each other. Figure 4-b shows the error events caused by a Microwave oven. From the figure, it can be clearly recognized the periodical radiation pattern of the oven, with $10ms$ of activity and $10ms$ of inactivity. During radiation, channel is sensed as busy by the WiFi node, but error events are pretty different from the ones caused by ZigBee transmissions, since they are concentrated at the beginning and at the end of the radiation interval (rather than being continuously repeated). This can be due to the power-on and power-down ramp of the Microwave, being the demodulator unable to work when the radiation power is stable. Finally, Figure 4-c shows the receiver events in presence of LTE frames, with only 5 sub-frames. Under this interference source, the WiFi receiver behavior resembles the ZigBee interference. However, the synchronization trials are generally closer to each other, in comparison with ZigBee, and the occurrence of the first synchronization trial is not always synchronized with the activation of the channel busy register. For example, at time $20ms$ the busy channel state switches to 1, while the first event vector with non-null components (namely, three Bad PLCP events) is revealed after $2ms$.

C. Classification Features

The experimental results presented in the previous section show that, although all non-WiFi interfering signals generate the same type of errors with similar statistics, their temporal analysis can be exploited for discriminating among different interfering sources. From the qualitative description of Figure 4, it clearly emerges that several features can be exploited for such a discrimination, such as:

1) *the error rate* generated by the monitoring process during the same interfering burst, which depends on the interfering power, with an higher number of synchronization trials performed in case of LTE-U signals;
2) the specific *sequence of error vectors*, which is affected by the variability of the interfering power during the same transmission (as in the case of Microwave ovens and LTE frames) and exhibits completely different occurrence probabilities in case of WiFi modulated signals;
3) *the temporal gap between consecutive error events* within the same burst, which depend on the receiver reset time required after a synchronization failure, which in turns is affected by the signal type and power (e.g. the ramp effects for the Microwave oven).

These features can be easily exploited for developing automatic classification schemes. We consider two different approaches for modeling the receiver behavior under different interfering sources: an HMM, capturing the main features of the receiver operations, as evident from our error analysis; an ANN, able to aggregate the distinguishing features of the error patterns emerged in our analysis into a classification decision.

V. Hidden Markov Model

We propose to model the receiver behavior by means of a HMM, whose discrete evolution times correspond to the regular sampling intervals of our monitoring process. Although at a given time it is not possible to directly know which operations are performed by the receiver, such as a synchronization trial, the demodulation of a frame field, the gain adjustment, etc.,

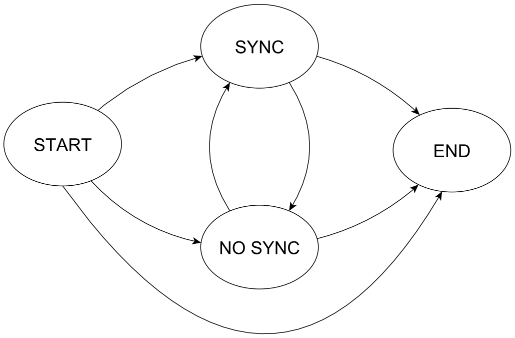


Fig. 5. Generalized state model of the receiver behavior: transition probabilities depend on the interference source.

the error vectors generated by our monitoring process can be considered as indirect observations of the receiver state. Being observations generated at discrete times, we assume that model evolutions are performed at the same time instants.

The adoption of a Markov chain is motivated by the need of modeling the memory effects described in the temporal analysis of the error vectors. Indeed, for some interfering sources such as the microwave oven, non-null error vectors are generated only at the beginning and at the end of the interfering signal, while for some other sources, such as LTE, the first synchronization trial can be performed after a delay from the detection of a channel busy state. Moreover, some specific error events related to the non-valid frame formats (such as too long or too short frames) are triggered only after the detection of a valid preamble. Figure 5 shows our receiver model with four possible states: the START and END states identifies the initial and final stage of the receiver activation; the SYNC state identifies the receiver operation after the synchronization of a valid preamble; the NO SYNC state characterizes the multiple synchronization trials performed when a valid preamble is not detected.

The probability of switching from one state to another depends on the errors detected by the receiver and on their typical timings. For example, the transition probability from START to SYNC is almost 1 for WiFi signals and 1/4 for non-WiFi signals [7]. The probability to observe a given error vector, also called *emission probability*, mostly depends on the receiver internal state. For example, the probability to generate an error due to a too long packet is non-null only in the SYNC state, when the receiver is trying to demodulate the interfering signal as a valid WiFi frame. The interfering source affects the emission probability, because the latency required for resetting the receiver and performing multiple synchronization trials depends on the specific interfering signal. It follows that a different receiver model, specified in terms of transition and emission probabilities, can be used for characterizing the receiver behavior under specific interference conditions. Interfering signals which do not trigger the activation of the WiFi receiver are not detected by our scheme.

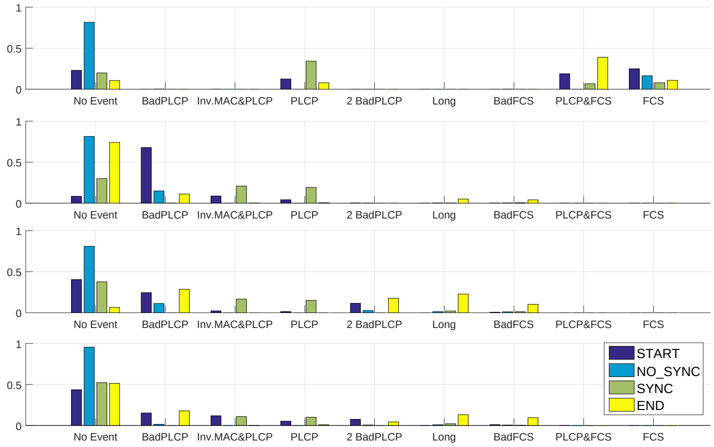


Fig. 6. Emission Probabilities of most significant observations for different experiments (from top to bottom: WiFi, ZigBee, LTE-U and Microwave).

A. Model training

For tuning the emission and transition probability from each state as a function of a specific source of interference, we implemented a *training phase* of the hidden Markov chain, based on a trace of error vectors acquired in presence of controlled interference. While the number of possible events summarized in table I is eight, the overall number of possible error vectors is higher because multiple events can be triggered during the sampling interval of the card registers. However, in most cases error vectors have a single non-null component and can be directly mapped into an event.

For deriving a known state path, we implemented the following approach. On the basis of the busy channel register, we organized the error vector trace into bursts of consecutive errors due to the same interfering transmission. For example, in Figure 4-a there are four error bursts, with a last interval equal to the event sequence {Bad PLCP, Bad PLCP, Bad PLCP, Good PLCP, Too Long}. The state path corresponding to each activity interval can be easily derived by considering that the first and last observations are always performed from the START and END state, while all the others depend on the last preamble synchronization. In case of observations including both a Bad PLCP and Good PLCP event, the last receiver event can be estimated by considering the occurrence of Too Long, Invalid MAC or Bad FCS events, which always follows a Good PLCP event.

We collected three different event traces of $10s$ under WiFi traffic, ZigBee, LTE-U and Microwave interference. By using each trace and corresponding state path, we obtained the maximum likelihood estimates of the emission and transition probabilities from each state, devised to characterize the receiver behavior in presence of different signals. The derivation is based on the Baum-Welch algorithm. Figure 6 visualizes the emission probabilities of the most significant observations for different interference models. It is interesting to observe how the figure quantifies our previous qualitative considerations.

For the WiFi model, most observations result in a synchronized preamble followed by a correct checksum (that can be sampled into the same observation interval or into two consecutive observation intervals due to the short duration of WiFi frames). Packet duration is equal to about 350 μs, because we used frames with 1500 bytes transmitted at 36 Mbps. For the ZigBee model, bad preambles are generated very often: about 70% of error bursts start with such an event, while the other bad preambles are revealed during the intermediate model states. Checksum failures, too long frames or invalid MAC occur at the edge states or when the receiver is synchronized. For the Microwave oven, bad preambles are generated in the START and END states and the no event probability is higher than the previous ones (being the interference interval equal to 10 ms and the demodulator active only during the power ramp). Finally, the LTE-U model falls somehow in between the ZigBee and the microwave model, with a slightly higher number of error events triggered.

Although the specific emission probabilities may depend on the receiver implementation, and in particular on the reaction times to synchronization errors and sensitivity to narrow-band signals, the approach for training the hidden Markov chain is general and can be applied to different receiver implementations (provided that they can track the internal error events).

B. Classification scheme

As a result of the training phase, we define four different HMM models characterizing the receiver behavior in presence of WiFi, ZigBee, Microwave and LTE-U interference. A HMM model is specified by the definition of the transition probability matrix, governing the state evolution process, and by the emission probability matrix, characterizing the probability to observe different error vectors from each state. The number of hidden states in the general receiver model, as depicted in Figure 5, is equal to 4. The number of possible error vectors is higher than the total number of possible events (which in our implementation is equal to 8), because multiple events can be triggered during the same sampling interval. However, being such a maximum number limited, the total number of possible configurations is limited too (in our experiments we found a maximum number of 40 different vectors). Let n be the generic number of states and m be the total number of error vectors with non-null occurrence probability. The receiver model in presence of the k-th interference source is given by the transition probability matrix $P_k^{n \times n}$ and emission probability matrix $E_k^{n \times m}$ found by the training mechanism described in the previous section.

The classification is based on the receiver behavior during a given error burst delimited by the channel busy register (i.e. a single frame or a single microwave radiation period). Thus, idle times between consecutive error bursts are not considered for the classification. Let $\mathbf{e} = e_1, e_2, \cdots e_L$ be a sequence of error vectors and L the length of a single error burst, delimited by using the busy channel register. Our classification scheme works by selecting the interference model which maximizes the probability of obtaining the sequence $\mathbf{e}$, i.e. the interfering source is $k = argmax_k \; Pr\{\mathbf{e}|P_k, E_k\}$. The probability $Pr\{\mathbf{e}|P_k, E_k\}$ can be obtained by deriving the state probability at each sampling interval $i = 1, \cdots L$ of the sequence, and by weighting accordingly the emission probability of each observation e_i from each state. Note that the state path is partially known because it is delimited by the START and END states, whose occurrence probability are equal to 1, respectively, at time 1 and time L of the burst (regardless of the transition probabilities derived in the training phase).

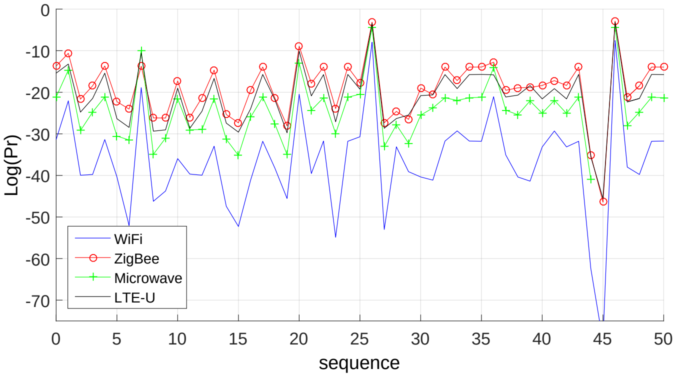


Fig. 7. Comparison between the burst-based receiver models for a sequence of error bursts due to ZigBee transmissions.

TABLE II
CONFUSION MATRIX OBTAINED WITH THE HMM-BASED CLASSIFIER.

	WiFi	ZigBee	Microwave	LTE-U
WiFi	**100.0**	0.0	0.0	0.0
ZigBee	0.0	**90.0**	4.6	5.4
Microwave	0.2	1.7	**89.6**	8.5
LTE-U	6.1	0.2	4.1	**89.5**
LTE-U 5ms	0.0	4.9	13.0	**82.2**

For assessing the performance of our classification schemes, we considered the case when a single interference source is active. Figure 7 visualizes the classification results obtained by the HMM-based classification in presence of ZigBee interference, with full packet size (128 Bytes) and average burst duration is approximately 4.5 ms (i.e. 18 error vectors). The figure shows the logarithm of the occurrence probability of each sequence $\mathbf{e}$ computed according to the four interfering models. From the figure it is evident that the highest probability corresponds to the ZigBee interference source in almost all the cases. Moreover, the results provided by the WiFi model are very far from the other models. Similar results were obtained also with the other interference sources.

The classification accuracy, defined as the ratio between the number of correct decisions and the total number of error bursts generated by each interfering source, is quantified in table II: the accuracy is on average close to 90%. For demonstrating the robustness of our approach in recognizing error bursts whose duration is different from the one used

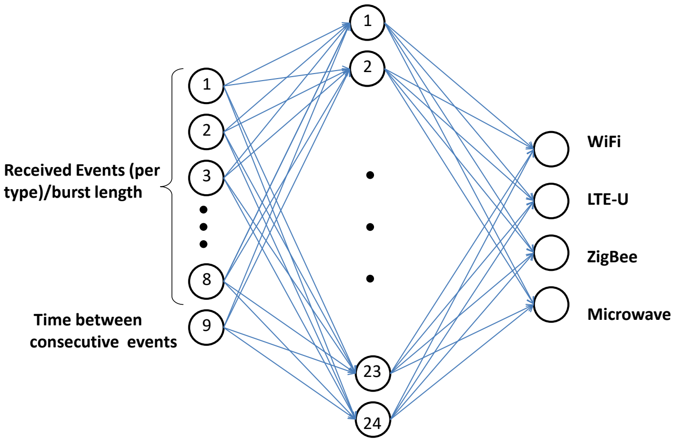

Received Events (per type)/burst length
Time between consecutive events
1
2
3
8
9
1
2
23
24
WiFi
LTE-U
ZigBee
Microwave

Fig. 8. Structure of the MLP neural network used in our experiments.

TABLE III
INPUT RECORD RELATIVE TO THE LAST ERROR BURST OF FIG. 4-C.

Features	value
Too Long	0
Too Short	0
Invalid Mac Header	0
Bad FCS	1/5370
Bad PLCP	4/5370
Good PLCP	1/5370
Good FCS, matching RA	0
Good FCS, not matching RA	0
Max time between events	1100

during the training, we also tried to classify error bursts due to LTE frame with blank sub-frames, lasting 5 ms only (rather than 10 ms, as considered for the HMM training). Despite the fact that in this case the error bust duration is very close to the ZigBee one, the last row in table II shows that the classification accuracy only slightly degrades by 7.5% and is still higher than 80%. Classification of independent bursts (generated by different technologies) should work as in the case of single interference sources, apart from the case when the burst is generated by collisions between multiple interference sources. This type of combined interference, in principle, can be modeled for introducing more advanced interference detection schemes (able for example to recognize WiFi/ZigBee collisions). However, the identification of such events is of little interest and is out of the scope of this paper.

VI. Neural Network Model

We consider a classification model based on Multi-Layer Perceptron (MLP) neural networks, that are widely used ANNs based on the so called *feedforward* architectures. As depicted in Figure 8, the architecture is based on one input layer, one hidden layer and one output layer. Features in the input layer are organized in 9 neurons: 8 neurons represent the ratio between the counter of 8 types of reception errors [7] and the duration of the burst in μs, while one neuron represents the maximum distance between two consecutive events in the same burst in μs. Table III shows an exemplary vector of input features generated by the last error burst generated by LTE-U in Figure 4-c. The choice of these features is motivated by the temporal analysis of the error patterns presented in section IV, where we clearly observed that the error generation rate and the distance between errors represent distinguishing elements of each interfering technology. Note also that these features do not directly depend on the specific duration of the interference event, although longer interference events allow to better estimate the error rate due to a given interference source. Since in this work we are considering 4 sources of interference (namely WiFi, ZigBee, Microwave ovens and LTE-U), the output layer of the MLP network is given by 4 neurons only, each one mapping a given technology available for detection.

The MLP network was implemented in Python using the *scikit-learn* machine learning library [29], trained by means of a back propagation algorithm with a Cross-Entropy loss function. Since MLP is sensitive to work on normalized data, i.e. on features of Gaussian distribution with zero mean and unit variance, we preprocessed our data by removing the average value and dividing the values by the feature's standard deviation.

The dataset was randomly divided into two parts (using the *train_test_split* function of scikit-learn), the training set and the test set: the first one is used for training and validating the neural network, the second one is used for evaluating the classification accuracy. We considered a training set of 6078 samples (equally distributed between LTE-U, WiFi, ZigBee and Microwave oven), while the test set was composed of 2606 samples. Each sample is constituted by a vector of nine features associated to the interference source that caused it. Finally, the hyper-parameters of the network, i.e. the number of neurons in the hidden layer, the activation function and the regularization factor have been studied in the Model Selection phase, as discussed in the following subsection.

A. Model selection

For the design of a neural network architecture, the model selection phase consists in comparing the performance obtained by changing different hyper-parameters, and choosing accordingly the ones that maximize the classification accuracy. To avoid the overfitting problem, we carried out a "k-fold" cross-validation with k = 10: we divided the training set into 10 equal parts and, at each step, one sub-sequence of the data set was used to evaluate the accuracy of the model trained with the remaining nine sub-sequences. We used the GridSearchCV function of *scikit-learn* to carry out an exhaustive "grid" search over the space of hyper-parameters considered in our analysis, and performed a k-fold cross-validation for each obtained model. Specifically, the space of hyper-parameters was configured by considering the following factors:

1) *solvers:* L-BFGS, adam, SGD with constant learning rate, SGD with adaptive learning rate;
2) *number of neurons in the hidden layer:* from 1 to 30;
3) *regularization factor "alpha" (L2 penalty):* 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7};
4) *activation function:* identity, logistic, tanh, ReLU.

TABLE IV
OPTIMIZATION SOLVERS WITH RELATIVE TRAINING TIMES.

Solver	Accuracy	Training time	Iterations
SGD with constant learning	88.5%	33 s	882
SGD with adaptive learning	88.5%	34 s	898
L-BFGS	94.5%	23 s	1002
Adam	92.9%	13 s	289

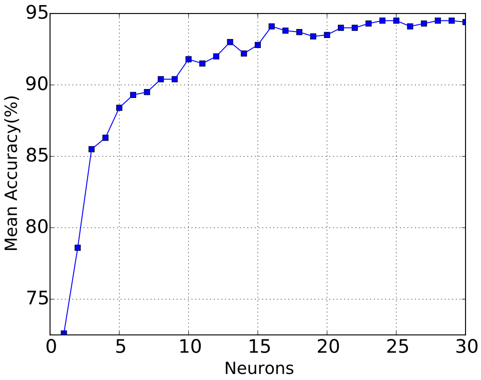


Fig. 9. Average accuracy versus the number of neurons in the hidden layer.

TABLE V
AVERAGE ACCURACY OBTAINED BY DIFFERENT ACTIVATION FUNCTIONS.

Function	Accuracy
Identity	58.5%
Logistic	93,9%
ReLU	93,7%
tanh	94.5%

TABLE VI
AVERAGE ACCURACY OBTAINED BY VARYING THE REGULARIZATION FACTOR.

Alpha:	0.1	0.01	0.001	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Accuracy:	94.5%	93.8 %	93.9%	94.4%	94.2%	94.2%	94.3%

For solvers adam and SGD the initial learning rate was set to 10^{-3} (default value in *scikit-learn*), which controls the step-size in updating the weights. SGD was set with a nestorovs momentum of 0.9, while in adam the exponential decay rate for estimates of first and second moment were set to β_1 = 0.9 and β_2 = 0.999. All solvers have tolerance *tol* = 10^{-4}. The solvers iterate until convergence (determined by *tol*) or up to a maximum number of iterations (never reached in our experiments).

In Table IV it is shown the average accuracy, the time required for training the weights of the optimization algorithms and the number of iterations until convergence. The optimization process has been run on a laptop PC with dual core 1.8 GHz CPUs and 8 GB of RAM. It is clear that L-BFGS method converges with higher accuracy. Figure 9 shows that, for a given configuration of the other hyper-parameters, increasing the number of neurons in the hidden layer improves the accuracy until a limit value of about 94.5%.

Tables V and VI show the performance achieved with different activation functions and regularization factors. In particular, the tanh function reaches a higher accuracy compared to other activation functions, while the optimal regularization factor alpha was 10^{-1}. The final hyper-parameters derived by the model selection phase result in the MLP architecture shown in Figure 8 (where we omit the bias node for the sake of simplicity) with 24 neurons in the hidden layer.

B. Classification performance

After identifying the best hyper-parameters, we trained the ANN network on the entire training set and evaluated the classification accuracy on the test set. To this purpose, we used a test set of 2606 burst samples (equally distributed per class) representative of the four categories WiFi, ZigBee, Microwave and LTE-U. Table VII shows the confusion matrix of the classifier, which obtains an average accuracy over 95%. The few errors are between ZigBee and LTE-U, because of the similarity of the error burst, as shown in figure 4. Finally, to verify the robustness of the model, we evaluated the classifier on the entire dataset composed of 9362 elements. Table VIII shows that the classification performance is maintained even considering such a larger dataset, confirming the excellent results shown on the test set. In this last experiment, we also considered the classification performance when the neural network model, trained with a LTE source transmitting a typical frame of 10 ms, is utilized for recognizing LTE frames with blank sub-frames lasting 5 ms only. The last row in table VIII demonstrates that the ANN classification scheme is not sensitive to the duration of the error bursts used for training, being the model based on the error rates and on the maximum latency between consecutive error events.

VII. DISCUSSION

The results in sections V and VI show that the ANN performs better than the HMM, despite the fact that the HMM classifier models the effects of the receiver memory after the sychronization of a good preamble. We justified such a result by observing that the interference class separation is more robust working on *aggregated burst features*, such as the maximum gap between consecutive errors or the occurrence rate of a given event, rather than *ordered sequences* of error events.

We also compared the two classifiers in terms of training and classification complexity. Regarding the first aspect, different training schemes for defining the ANN parameters have been compared in Table IV in terms of running times. The schemes have been executed on the same laptop, with a dual core 1.8 GHz CPU. Despite of the limited computational resources, the maximum training time resulted equal to 33s. Furthermore, training for the L-BFGS method (which performed best and was thus used in our tests) was 23s and this can be easily reduced by using a more powerful PC. The training time results lower than 1s for the training of our simple HMM structure. In any case, we remark that the

TABLE VII
CONFUSION MATRIX OF THE ANN ON THE TEST SET.

	WiFi	ZigBee	Microwave	LTE-U
WiFi	**100.0**	0.0	0.0	0.0
ZigBee	0.0	**94.0**	0.4	5.6
Microwave	0.0	0.0	**99.8**	0.2
LTE-U	0.0	6.0	0.7	**93.3**

TABLE VIII
CONFUSION MATRIX OF THE ANN ON THE ENTIRE DATASET.

	WiFi	ZigBee	Microwave	LTE-U
WiFi	**100.0**	0.0	0.0	0.0
ZigBee	0.0	**95.2**	0.1	4.7
Microwave	0.0	0.2	**99.2**	0.6
LTE-U	0.0	4.7	0.6	**94.7**
LTE-U 5ms	0.0	6.1	0.5	**93.4**

training time is required only once, during the configuration phase of the classifiers.

Regarding the classification complexity, it is easy to formalize the two classifier behaviors as a function of their design parameters. Being n the number of inputs, m the number of hidden neurons and p the number of outputs, the ANN classifier has complexity of $O(n \cdot m \cdot p)$. Being N the number of states and T the burst lenght, the HMM classifier has a complexity equal at most to $O(N^2 \cdot T)$. Since in our design we have $n = 10$, $m = 24$ and $p = 4$, and $N = 4$, while the average T value is equal to 5, it turns out that the HMM classification is much less expensive than the ANN one.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented two novel classification schemes for detecting ZigBee, LTE-U or microwave oven interference using commodity WiFi cards. The idea is to exploit the error events caused by cross-technology interference on the WiFi node. Based on such error traces, we defined two schemes able to characterize the receiver behavior in presence of different interference sources: a Hidden Markov Model and a Artificial Neural Network. After selecting the most appropriate structures and training the models, we tested the two classifiers on a large dataset. Our experimental results show that the accuracy is on average over 90% with HMM and over 95% with the ANN. This result suggests that classification is more robust by considering aggregated per-burst features rather than ordered sequences of error events.

Although in this paper, the focus was to identify the interference caused by ZigBee, WiFi, microwave and LTE-U, the proposed approach could be easily extended to additional interfering technologies operating in the ISM band, e.g. Bluetooth or cordless phones. Moreover, we are currently considering the possibility of performing *time-based* decisions rather than *burst-based* decisions, by monitoring the channel for a longer observation interval which include multiple error bursts. In this case, it could be relevant to model correlation effects between consecutive error bursts, which in case of similarity are likely due to the same interference source. A possible design solution for addressing this scenario is the utilization of recurrent neural networks, such as the LSTM (Long Short-Term Memory) networks. For these structures, the final decision is taken at the end of multiple stages, in which the output of the previous stage is used as an additional input for classification. However, the output of this classifier should also be extended for taking into account that multiple interference sources can contend on the channel at the same time, for example by providing a percentage of activation for each possible interference class. More investigations are thus needed to study such a generalization.

Finally, we are planning different exploitations of our interference detection scheme. For example, we are implementing some forms of inter-technology communication protocols by opportunistically exploiting the generation and identification of error patterns with different durations. Inter-technology communications would allow to easily manage spectrum sharing and channel reservations among overlapping networks.

REFERENCES

[1] http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-13_description_20150917.zip

[2] http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-10_description_20140630.zip

[3] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: detecting non-WiFi RF devices using commodity wifi hardware. In Proc. of IMC 2011.

[4] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai. Harmful Coexistence Between 802.15.4 and 802.11: A Measurement-based Study. In Proc. of CrownCom, 2008.

[5] R. Gummadi, D. Wetherall, B. Greenstein, S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 Networks. In Proc. of ACM SIGCOMM '07, Pages 385-396.

[6] A. M. Cavalcante et al., Performance Evaluation of LTE and Wi-Fi Coexistence in Unlicensed Bands, 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), Dresden, 2013, pp. 1-6.

[7] D. Croce, D. Garlisi, F. Giuliano and I. Tinnirello, Learning from Errors: Detecting ZigBee Interference in WiFi networks, in Proc. 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET) 2014.

[8] J. Huang; G. Xing; G. Zhou; R. Zhou. Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance. ICNP, 2010.

[9] X. Zhang, K. G. Shin. Enabling Coexistence of Heterogeneous Wireless Systems: Case for ZigBee and WiFi. In Proc. of ACM MobiHoc '11.

[10] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In Proc. of SenSys 10, pages 309-322, 2010.

[11] Yubing Jian, Chao-Fang Shih, Bhuvana Krishnaswamy, Raghupathy Sivakumar. Coexistence of Wi-Fi and LAA-LTE: Experimental Evaluation, Analysis and Insights. IEEE International Conference Communication Workshop (ICCW), 2015

[12] E. Almeida et al., "Enabling LTE/WiFi coexistence by LTE blank subframe allocation," 2013 IEEE International Conference on Communications (ICC), Budapest, 2013, pp. 5083-5088.

[13] S. Yun and L. Qiu. Supporting WiFi and LTE co-existence. IEEE Conference on Computer Communications (INFOCOM), Kowloon, 2015.

[14] S. Bhattarai, J. M. J. Park, B. Gao, K. Bian and W. Lehr, "An Overview of Dynamic Spectrum Sharing: Ongoing Initiatives, Challenges, and a Roadmap for Future Research," in IEEE Transactions on Cognitive Communications and Networking, vol. 2, no. 2, pp. 110-128, June 2016.

[15] R. Gummadi, H. Balakrishnan, and S. Seshan. Metronome: Coordinating Spectrum Sharing in Heterogeneous Wireless Networks. 1st Int. Workshop on Communication Systems and Networks (COMSNETS), 2009.

[16] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the RF smog: making 802.11n robust to cross-technology interference. In Proc. of ACM SIGCOMM 11, pages 170-181, 2011

[17] K. Bian, J. M. Park, L. Chen and X. Li, "Addressing the Hidden Terminal Problem for Heterogeneous Coexistence Between TDM and CSMA Networks in White Space," in IEEE Transactions on Vehicular Technology, vol. 63, no. 9, pp. 4450-4463, Nov. 2014.

[18] P. De Valck, I. Moerman, D. Croce, F. Giuliano, I. Tinnirello, D. Garlisi, E. De Poorter, B. Jooris, Exploiting programmable architectures for WiFi/ZigBee inter-technology cooperation, EURASIP Journal on Wireless Communications and Networking, Vol. 1, pp. 1–13, 2014.

[19] S. Sagari, S. Baysting, D. Saha, I. Seskar, W. Trappe and D. Raychaudhuri, ”Coordinated dynamic spectrum management of LTE-U and Wi-Fi networks,” 2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Stockholm, 2015.

[20] Q. Chen, G. Yu and Z. Ding, ”Optimizing Unlicensed Spectrum Sharing for LTE-U and WiFi Network Coexistence,” in IEEE Journal on Selected Areas in Communications, vol. 34, no. 10, pp. 2562-2574, Oct. 2016.

[21] K. Lakshminarayanan, S. Sapra, S. Seshan, and P. Steenkiste. RF-Dump: An Architecture for Monitoring the Wireless Ether. In Procs. of CoNEXT 09, Dec. 2009.

[22] O. Zakaria. Blind signal detection and identification over the 2.4 GHz ISM band for cognitive radio. In MS Thesis USF09

[23] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. Comput. Netw., 2006.

[24] F. Hermans, L. Larzon, O. Rensfelt, P. Gunningberg. A Lightweight Approach to Online Detection and Classification of Interference in 802.15.4-based Sensor Networks. In ACM SIGBED Review - CONET 2012, Vol. 9, Issue 3, July 2012, Pages 11-20.

[25] R. Zhou, Y. Xiong, G. Xing. ZiFi: Wireless LAN Discovery via ZigBee Interference Signatures. In Proc. of ACM Mobicom 2010.

[26] K. H. Kim, H. Nam and H. Schulzrinne, ”WiSlow: A Wi-Fi network performance troubleshooting tool for end users,” IEEE INFOCOM 2014, Toronto, Canada, 2014, pp. 862–870.

[27] http://bcm-v4.sipsolutions.net/802.11/Registers/

[28] https://github.com/srsLTE/srsLTE

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau. ”Scikit-learn: Machine Learning in Python”. Journal of Machine Learning Research Vol. 12, pp. 2825-2830, 2011.