DySPAN Spectrum Challenge: situational awareness and opportunistic spectrum access benchmarked

Felix Wunsch, Francisco Paisana, Sreeraj Rajendran, Ahmed Selim, Pedro Alvarez, Sebastian Müller, Sebastian Koslowski, Bertold Van den Bergh, and Sofie Pollin, *Senior Member, IEEE*

***Abstract*—In this paper, we describe the original problem statement and the two winning solutions to the IEEE DySPAN Challenge, organized in Baltimore in 2017. The idea of the challenge was to invite teams to propose as diverse as possible solutions to a well defined problem, and evaluate the performance of the proposed solutions in a realistic environment. The challenge is defined to enable benchmarking and comparison of multiple teams, possibly working on different parts of the system, in a real environment. The winning solutions represented a complete and working system, working robustly and adapting to both anticipated scenario changes, as well as random effects caused by the conference setting. The code for running the challenge along with the winning solutions is publicly available, so that interested teams can start from the code when designing or benchmarking solutions, as well as when setting up own challenges and competitions. As a result, the challenge can serve as a milestone towards the creation of a benchmarking series. This paper contains all the necessary details about the software repositories so that it becomes possible to rerun the challenge and start building novel solutions based on the winners in IEEE DySPAN 2017.**

***Keywords*—Spectrum challenge, Cognitive radio, Deep learning, FBMC.**

I. Introduction

Wireless data usage has increased tremendously over the past few years. New spectral resources are allocated to meet this ever increasing demand. These new allocations are limited as the wireless spectrum is scarce. To enable efficient utilization of this scarce resource, it is becoming crucial that wireless radios share the available spectrum. Efficient sharing can be enabled only when all the participating radios are well aware of their operating wireless environment. Thus wireless situation awareness is unavoidable in modern radios for enabling efficient spectrum sharing and reliable performance especially in highly interfering scenarios. For instance, multiple self-organized wireless networks can share the spectrum and work independently without explicit coordination.

To comprehend the benefits of learning and adaptation of wireless parameters and the importance of feedback information to facilitate this learning and adaptation, IEEE DySPAN organized a spectrum challenge in 2015 [1]. During this challenge it was concluded that the main challenges in realizing such situational awareness include limitations in observing the channel in a half duplex radio receiver, modeling limitations of the non-stationary PU transmissions and the inherent non-determinism of channel occupancy. Furthermore, benchmarking the importance of situational awareness was challenging mainly due to the versatility of the solutions.

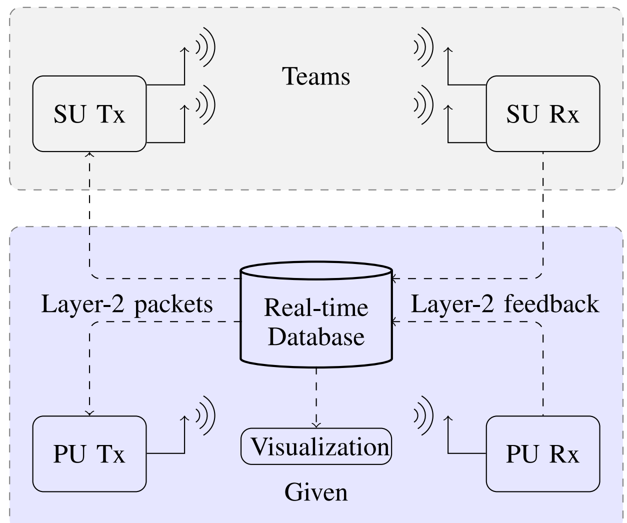


Fig. 1: Spectrum challenge setup: A real-time feedback from the database is enabled along with score logging. Primary User (PU) and Secondary User (SU) throughput and the spectrum usage is displayed by the visualization module.

For the IEEE DySPAN spectrum challenge 2017[1], a generic platform was developed to evaluate the state-of-the-art (SoA) wireless algorithm performance in terms of radio situation awareness and dynamic spectrum usage. First, a few fixed scenarios are designed to benchmark the detection performance of the SoA algorithms. Secondly, the algorithms' dynamic spectrum usage performance is tested and analyzed in these scenarios. The complete challenge framework along with the solutions is made public to the research community to enable future benchmarks [2].

A high level overview of the IEEE DySPAN 2017 challenge setup is given in Figure 1. The setup consists of two PU and two SU radios which are connected to a central database server. The rules of the challenge allowed each radio to use at most two antennas. Real-time packets and performance metrics were provided by the database. In addition, the database also logged statistics about the system performance which is forwarded to the visualization module for easy analysis during the course of the challenge.

The rest of the paper is organized as follows. Section II

[1]http://dyspan2017.ieee-dyspan.org/spectrum-challenge
[2]https://github.com/networkedsystems/dyspanchallenge_2017

Scenario	Description	Inter-packet delay
0	Single random channel	5 ms
1	Single random channel	10 ms
2	Two random channels, hopping	5 ms
3	Four random channels, hopping	10 ms
4	Four channels, synchronous	5 ms
5	Two random channels, synchronous	5 ms
6	Four channels, synchronous	2 ms
7	Four channels, Poisson-distributed delays	20 ms (mean)
8	Four channels, Poisson-distributed delays	10 ms (mean)
9	Four channels, Poisson-distributed delays	5 ms (mean)

TABLE I: Challenge scenarios: Different PU scenarios for the challenge. A visual representation of all the scenarios can be also found in Figure 3.

defines the challenge setup and the winning parameters. The winning solution for spectrum situation awareness is detailed in section III. Section IV details the solution that performed best in agile spectrum usage. Challenge results and conclusions are presented in sections V and VII, respectively.

II. Challenge Problem Definition

Some of the major criteria that were considered while designing the challenge included enabling breadth in the solutions in terms of hardware and software solutions and real-time feedback about the PU and SU throughput statistics. To comply with the criteria, an open GNU Radio based OFDM stack was used in the physical layer and made available to the participating teams for easy testing. Even though the PU design was made public, the challenge scenario parameters were randomized during the actual challenge. Challenge winning metrics were selected to answer the following questions

- Situation awareness: How good is SoA research in detecting wireless scenarios?
- Spectrum sharing: How well can the secondary users exploit the spectrum, once they detect the scenarios?
- Receiver feedback: What is the upper bound in performance that can be achieved if there is perfect PU and SU receiver feedback?

The PU simultaneously transmitted on four predefined frequency bands with a channel bandwidth of 2.5 MHz. The secondary user had to transmit over the same 10 MHz band which the PU was using. The spectrum usage was monitored in real-time and all out-of-band transmissions were heavily penalized. In order to avoid interferences from other wireless devices, a dedicated band with a center frequency of 3195 MHz was used during the challenge. More about the challenge setup, phases and scores is detailed in the following subsections.

A. *Challenge scenarios*

Selecting realistic wireless spectrum occupancy scenarios was important to benchmark the current capabilities of SoA research. Detailed models could be found in literature capturing the very nature of spectrum occupancies, in terms of probability distributions, that can model the busy and idle periods of real systems [2]–[5]. Various experimental studies assuming a constant packet length and Poisson distributed packet arrival times were conducted in [2]. Wireless traffic analysis based on more realistic sources were done in [3]. In most of these analyses the time spent on data transmission and acknowledgement states were found to be deterministic, while the idle periods were fitted to generalized Pareto distribution or a mixture of uniform distributions. In [5] authors analyzed the busy and idle periods with multiple wireless devices operating in 2.4GHz ISM band. They concluded that hyper-exponential distributions fits the busy and idle periods very well while generalized Pareto distributions can provide good approximations.

During the DySPAN spectrum challenge 2015 [1], it was noticed that participants found it difficult to adapt their algorithms to highly random scenarios. Furthermore, simple distributions were preferred over complex ones to understand the baseline performance of SoA algorithms as well as reduce the parameter estimation overhead for participants. Acknowledging all these factors, 10 scenarios were selected with varying levels of difficulty. The scenarios ranged from the PU occupying a single random channel with deterministic delay to independent transmissions on all channels with random delays drawn from a distribution as listed in Table I. The deterministic single channel scenarios modeled realistic pilot transmissions in real world wireless transmissions. For difficult scenarios, the inter-packet delays were sampled from Poisson distributions under the assumption that with a quite large number of users the packet arrival event is uncorrelated holding the memoryless property. A packet payload length of 64 bytes was selected and the inter-packet period for each scenario is also listed in the table. Including the header and CRC, the entire packet duration was about 200 microseconds. The scenario switching time, the time period in which the PU stays in a particular scenario, is selected randomly from a uniform distribution between 15 and 30 seconds.

B. *Primary User Setup*

The PU radio used a four channel GNU Radio based OFDM stack which was connected to a USRP X310 front-end via Ethernet interface as shown in Figure 2. The packet controller block was responsible for controlling the inter-packet timing in the four channels for different scenarios. The packet controller selects a particular scenario from the ones shown in Table I, instructs the packet generator to request required packets from the database and sends the packets to the OFDM block for modulation. The modulated packets are then split into four independent channels which are passed to the channel combiner for further transmission over the air using the USRP transmitter. The packet controller and channel combiner together achieved the exact timing parameters for each scenario by following a sample count based design. On the receiver side the channel splitter splits the four channels and distributes them to four independent OFDM receiver blocks for demodulation. The demodulated packets are then sent to the database server for validation.

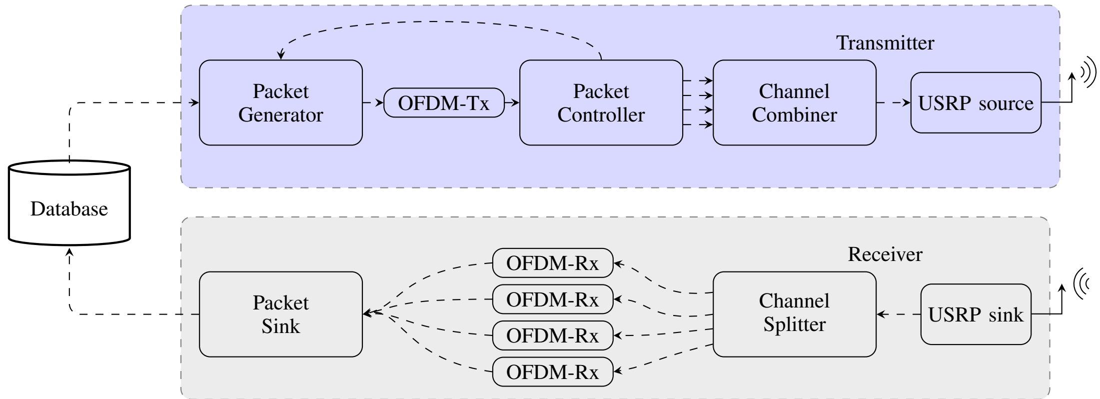


Fig. 2: Primary user setup

C. Database and Feedback

A central database server is implemented to provide PU and SU performance feedback. The server delivers random packets in requested number and size to the PU and SU transmitters. Packets received at the receiver end are sent back to the database where they are verified for correctness. The database also logs the PU and SU throughput along with the situation awareness status reported by the SU.

AES-128 is used in counter mode with a random key to generate packets with strong randomness at high throughput. The frames are sufficiently random thereby restricting further compression before transmission. A secure message authentication code (MAC) is calculated over the frame to detect if it has been corrupted. A sequence number is added to prevent packets from being counted multiple times. AES along with a secure MAC were used to prevent participating teams from predicting future packets or generating their own packets at the receiver end without transmitting over the air.

D. Challenge Phases

The challenge consisted of two phases: a situation awareness phase and an agile spectrum usage phase as given below.

- Situation awareness
  - SU should detect each scenario and report it to the database
  - PU feedback is provided to optimize the SU spectrum usage and radio parameters. SU is not penalized for interference which allows testing of spectrum sharing algorithms along with scenario detection.
  - Situation awareness score calculated in this phase
- Agile spectrum usage
  - SU should share the spectrum for data transmission
  - SU is penalized for interference
  - Spectrum usage metric calculated in this phase

Each phase had a duration of 10 minutes. The SU radio could learn about the environment, the PU transmission statistics and the exact PU transmitter and receiver RF properties impacting the interference sensitivity. This acquired knowledge could then be used to calibrate the SU parameters and algorithms to improve its performance.

E. Challenge metric

From the competing teams, two winners are selected, one metric scores the teams in function of the throughput and the other in function of their spectrum situation awareness. The spectrum agility metric was represented by the product of SU throughput (T_{SU}) and PU satisfaction. The PU satisfaction (S_{PU}) was calculated from the offered PU throughput ($\widehat{T}_{\mathrm{PU}}$) and the delivered PU throughput (T_{PU}) as given in Equation 1.

$$\begin{aligned} Score &= T_{\mathrm{SU}} \times S_{\mathrm{PU}} \\ S_{\mathrm{PU}} &= e^{-10\left(\dfrac{\widehat{T}_{\mathrm{PU}} - T_{\mathrm{PU}}}{T_{\mathrm{PU}}}\right)}. \end{aligned} \tag{1}$$

The situation awareness score is computed as the percentage of time the SU had perfect situation awareness. The score is calculated as given in Equation 2 where $\widehat{D}$ is the number of 1 millisecond time slots in which the SU radio has correct situation awareness. The total number of slots $D = 10 \times 60 \times 10^3$.

$$Score = \frac{\widehat{D}}{D} \tag{2}$$

III. Winning Situational Awareness Approach

A. Overview

The solution provided by Trinity College, Dublin (TCD) [6] comprises the following components:

- frequency-hopping system capable of overlay operation
- transmitter and receiver with OFDM waveform
- deep learning chain for PU's scenario classification
- energy detection for PU packet detection and positioning in frequency and time
- continuous adaptation of SU's frequency hopping pattern based on sensing results

- transmit power control for underlay operation, which relies on the database feedback

This solution is implemented in C++, using standalone libraries such as liquid-dsp for signal processing, and caffe for deep learning. The solution source code[3] and the samples dataset[4] used for training and testing the deep learning chain along with the trained models are made public for future research. The solution uses off-the-shelf USRP N210 hardware, where the reception and sensing is done using omnidirectional antennas, while the transmitter uses a directional waveguide antenna made of an open-ended metal can, for increased SNR and directionality.

This solution was built on top of TCD's DySPAN challenge 2015 objective winner solution [7].

B. Situational Awareness Considerations

Energy detection-based sensing algorithms generally display poor performances, when applied in the context of detection of short duration signals in low SNR scenarios. The reason behind this is that fast variations on the energy envelope of signals can be easily mistaken for the impulsiveness of AWGN [8]. This problem only gets exacerbated in real-world spectrum sharing scenarios, where multiple users' bursty transmissions and leakage contribute to more unpredictable changes in the energy envelope of an SU's received samples. One possible solution to overcome these issues is to use more advanced cyclostationary detection techniques tuned for the detection of specific PUs' waveform features. However, the lack of generality of these techniques is seen as impractical in many scenarios where multiple incumbent radio access technologies can operate in the same band.

TCD tackled both aforementioned issues by applying a more innovative machine learning-based approach. Leveraging recent breakthroughs in deep learning, TCD represented the problem of PU scenario classification, as a computer vision problem. This approach comprised two main stages: preprocessing, where the received raw IQ samples were converted to spectrogram images; and scenario classification, where the generated spectrograms were classified into one of the PU possible scenarios using a Convolutional Neural Network (CNN).

With this work, TCD extended the application of machine learning algorithms beyond modulation recognition [9], employing it in the identification of other PHY and MAC parameters, such as channel bandwidth, packet length, and inter-transmission delays.

1) Deep Learning for Scenario Classification

As shown in Table I, each PU's scenario is characterized by a unique set of characteristics. The traditional approach is to build single purpose estimators to estimate each parameter individually. Then, based on these estimates, the scenario is inferred using a decision tree or look-up table. However, here TCD introduced a strategy for recognizing the PU scenario that does not require estimation of individual parameters. Instead, TCD trained a deep CNN model to perform the classification directly from spectrograms. The fact that each PU scenario has unique characteristics is reflected as a unique set of spectrograms per class. Hence, the detection of PU scenarios becomes an image classification task that can be solved with CNNs. The choice of spectrograms for data representation was mainly determined by the nature of the PU scenario recognition task in the context of the challenge. The distinction between scenarios requires precise information of both the received signals' frequency and timing. The spectrogram provides a flexible tradeoff between time and frequency resolution, which is not as simple to achieve if raw time-series or frequency domain representations were utilized.

2) Training and Testing

Table II shows the number of collected spectrograms for all classes. 80% of the collected data is used for training while the rest (20%) is used to test the performance of the trained model. The data for each scenario was collected with the PU-Tx gain ranging from 0 to 30 dB. Such procedure is needed to train the model to work properly at low and high SNR values. To take into account different propagation conditions as well as hardware impairments, the training and test samples are collected at two different centre frequencies (2.3 and 3.195 GHz), with two different SBX daughter boards. For perfect training, the TCD team manually checked all generated spectrograms and removed any mislabeled data (often happens at very low SNR) that may degrade the performance of our model. Figure 3 shows few samples of our training dataset for all PUs scenarios.

To build the CNN model, TCD started by training a CNN architecture similar to [10] with input resolution 227x227. The main drawback of this model is that it takes 0.2 seconds to classify one spectrogram on our machine (Intel Core i7-6820HK Processor), which is not feasible to run in real time during the competition. For that reason, TCD experimented with other simpler configurations, using lower input dimensions and fewer convolutional layer filters. For each configuration, a CNN model is trained, its classification accuracy is tested, and the CPU-feedforward time is calculated. After this process, TCD reached the proposed architecture shown in Figure 4, using gray-scale spectrograms with input size of 64×64, which gives high classification accuracy and CPU-feedforward time of 6 msec per spectrogram. At this point, TCD decided not to further tune/optimize the model parameters, and focused on data collection instead. Each horizontal line of the spectrogram image represents the average power of 120 successive FFT bins (0.768 ms), and each channel (2.5 MHz) is represented with 16 bins (i.e. one spectrogram covers a duration of $0.768 \times 64 = 49.2$ ms and a bandwidth of $2.5 \times 4 = 10$ MHz). TCD concluded that a window of 49.2 ms is enough to create distinctive sets of spectrograms for each scenario. The CNN network consists of 5 convolutional layers equipped with 48, 128, 192, 192, and 128 filters, respectively. Each convolutional layer is accompanied by a rectified linear unit (ReLU) and followed by a max-pooling layer. The output of these convolutional layers is then passed through 3 fully connected layers with 1024, 1024, and 10 neurons, respectively.

[3]https://github.com/alvasMan/dyspan_radio_2017
[4]http://dx.doi.org/10.7910/DVN/EBLENC

TABLE II: Number of spectrogram images used per scenario

scenario	training dataset	testing dataset
0	8000	2000
1	8000	2000
2	18000	4500
3	3200	800
4	18000	4500
5	3200	800
6	3200	800
7	3200	800
8	3200	800
9	3200	800
Total	71200	17800

The last 10 neurons are fed to a softmax classifier to compute the probability $P(y = k|x;\theta)$ for $k \in \{0, 1, .., 9\}$, where x denotes the input spectrogram, and θ denotes our model parameters.

The CNN model is trained on 71200 spectrograms and tested on 17800 spectrograms (see Table II). The model is trained for 40K iterations, with batch size of 50. The training is performed using the stochastic gradient descent algorithm in which the weights of the network are updated after each iteration to minimize classification errors. The learning rate starts with $\alpha = 0.001$ and is multiplied by $\gamma = 0.1$ every 10K iterations (α is the weight of the negative gradient). The momentum μ is set to 0.9 (μ is the weight of the previous update). L2 regularization is used with weight decay 0.0005 to prevent over-fitting. Dropout was not used during the training process. The model was trained using the Caffe framework [11] and a current high-end GPU (Tesla K40c). After training, the model was tested over 17800 spectrograms (test dataset) achieving classification accuracy of 99.53%.

Figure 6 illustrates the performance of TCD's CNN classifier at various SNR levels for different PU scenarios. These results were obtained using a dataset different than the one employed for training and testing. As can be observed, the error rates are quite low, in some cases, even at negative SNRs. These results show that deep learning-based algorithms are reliable solutions, despite not requiring prior noise calibration. It is also possible to observe that the classification performance varies with the PU scenario. In Figure 7, this phenomenon is illustrated with higher detail by virtue of confusion matrices at different SNR levels. The main reason behind the observed behavior is the distinct transmission rates (i.e. number of channels and inter-packet delays) utilized by the PU for each scenario. As the SNR decreases, it becomes harder to distinguish PU's packets from noise, and the classifier increases its bias towards scenarios of low transmissions rates (e.g. 1 and 3). For SNRs below -5 dB, in particular, the classifier is not able to detect transmissions and gets fully biased towards the scenario 1, which makes sense as this scenario has the lowest rates of transmissions. In Figure 7, it is also possible to observe that scenario 6 is the most distinguishable for the classifier, as a result of its low and unique inter-packet delay.

For the challenge, the model was used in CPU mode and it took about 6 ms to execute one spectrogram. This means that our model is able to detect the change in PU's scenario with a delay of ≈ 56 ms. However, to accommodate for hardware impairments and inaccuracies, our decision for scenario change used a majority vote of 5 successive spectrograms. This is translated to a delay of 156 ms each time the PU changes its scenario. During the DySPAN spectrum challenge, our model achieved classification accuracy of 99.3%. Most of the 0.7% error was mainly due to delays associated with scenario changes as each participant experienced 20+ changes during the course of the challenge (i.e.$\approx 80\%$ of the error).

Despite being trained and tested in a single lab room, the CNN model's performance seemed unaffected by the distinct geometry and size of the room where the challenge took place. This suggests that the model is relatively robust to changes in the multipath environment. The main reasons behind this robustness are related to the nature of the challenge and the choice of spectrograms for data representation. While frequency selective fading can alter the shape of transmitted signals, it has little impact on their position (time and frequency) within the spectrogram image, which is more relevant in the context of PU scenario recognition.

C. Agile SU Considerations

The solution provided by TCD for the second phase comprises a frequency-hopping SU that can continuously adjust its hopping pattern and transmit power based both on the captured sensing results and database feedback. To make this solution flexible enough to accommodate a wide range of PU SNRs, simultaneous in-band sensing and transmission was not considered a viable approach. Instead, the SU transmits in one single channel at a time, while sensing the remaining channels to detect the presence of the PU and search for other free channels to hop to.

During this phase of the challenge, TCD did not utilize the spectrogram-based CNN model presented in Section III-B. There were two main reasons behind this decision. The first is that the number of possible states or scenarios the CNN would need to be trained for in this phase is significantly higher than in the previous one. This is due to the fact that the SU's activity, in particular, its hopping pattern and transmit power, affect its the samples used for sensing. Training the CNN model for all possible combinations would require resources that were not available at the time. The second reason is, the way it is designed, the CNN model only guesses the current PU scenario, not providing information on the Time of Arrivals (TOAs) and channel of PU's packets that is essential for frequency hopping. As an alternative to the CNN approach, TCD designed an energy detector, capable of assessing channel availability, and perform coarse PU scenario estimation. Knowing the PU scenario is relevant for the SU to know what frequency hopping strategy to employ as will be explained below.

1) Channelizer

The input samples are converted to frequency domain through a 64-size FFT block and magnitude squared to obtain their power. The 64 FFT bin powers are grouped into 12 sections, each section defined by a channel index (1-4) and

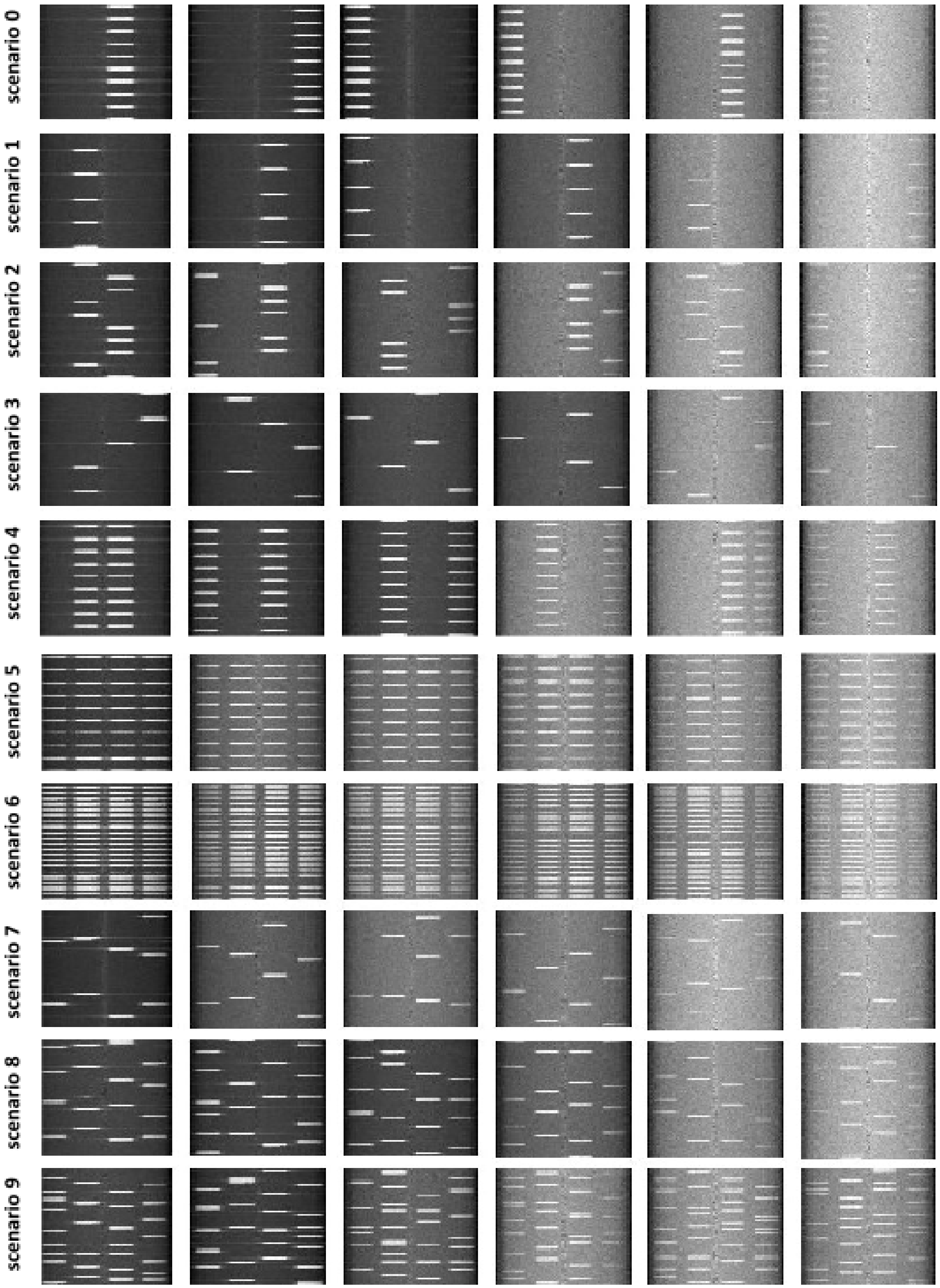


Fig. 3: Examples of training dataset for all PU's scenarios. The ith row shows six spectrograms for the $(i-1)$th scenario at different SNR values.

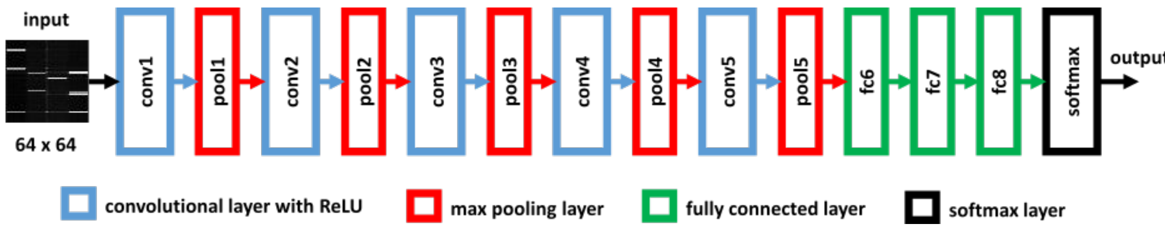


Fig. 4: Proposed CNN network architecture.

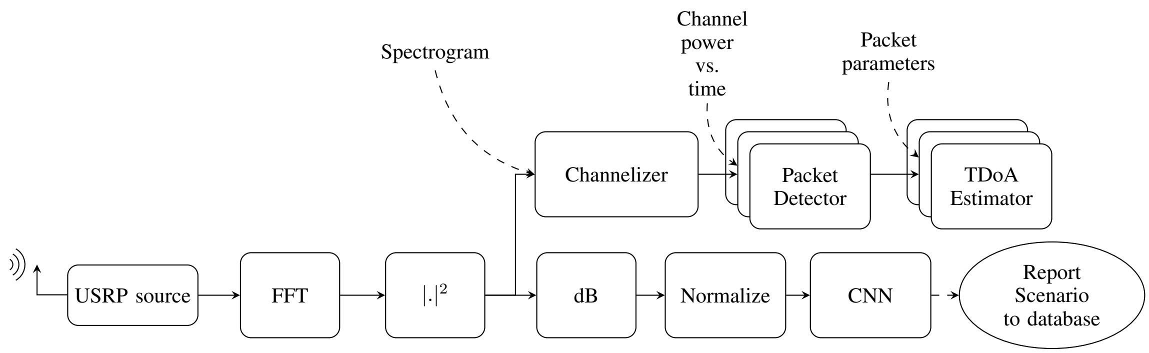


Fig. 5: RX chain

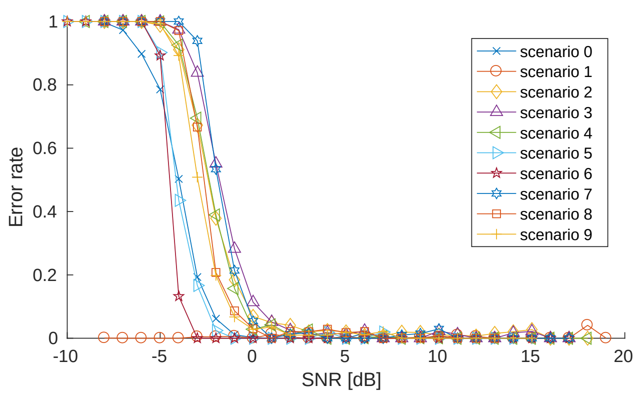


Fig. 6: Error rate variation of the proposed CNN model with SNR.

whether it belongs to the PU's useful, left guard, or right guard band of the channel.

The bins within the same section are then averaged to obtain a section's estimated received power. These powers then enter as input in the packet detector that will provide a list of the available channels.

2) Packet Detection

As the PU's transmission energy leaks to other channels, employing simple energy detection may lead to very high false alarm rates. For that reason, as a first stage, TCD's packet detector compensates the noise uncertainty in each channel by dividing the useful band section powers by the average of the powers obtained from the left and right guard band sections. The end result is an array of 4 relative received powers, which can be compared against a threshold to detect the arrival of a packet. To avoid the detection of the SU's own transmissions, packets detected in the current SU's channel were discarded.

3) TDoA Estimator

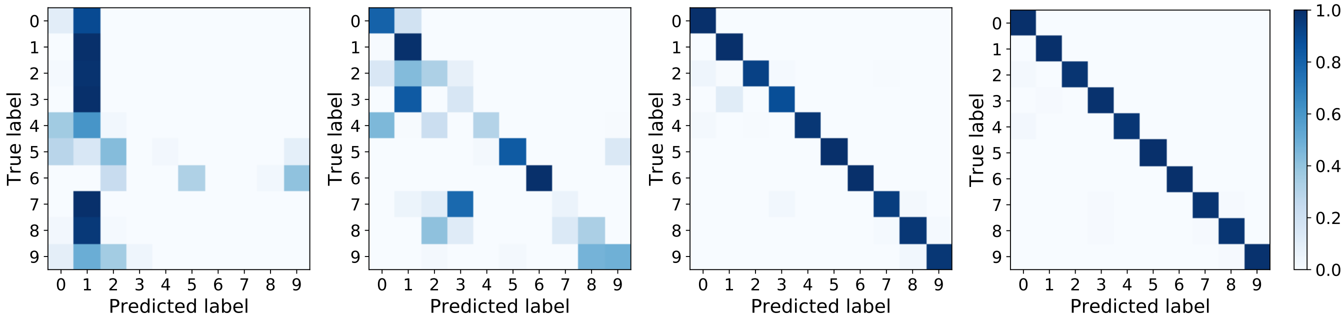
The packet detector outputs the packets' TOA to the Time Difference of Arrival (TDOA) estimator, which will then compute the mean and standard deviation of the TDOA, also known as the difference between consecutive packets' TOAs. The obtained means and deviations are then compared to those of each scenario, previously stored in a look-up table. The end result is a rough estimation of the PU's scenario.

D. Transmit Chain

1) Frequency Hopping Controller

The SU follows two different hopping strategies based on the current PU's scenario. In the case of scenarios 1, 2, 3, and 5 where only one or two channels are occupied by the PU, the SU selects one of the empty channels for operation. There is, however, one event this simple strategy does not account for. As the SU is not able sense its own channel, it may ambiguously classify scenario 5 as scenario 1, if it is operating in one of the PU channels. To avoid this issue, a maximum channel occupancy time of 500 ms was stipulated, after which the SU has to hop to another empty channel.

For scenarios 4, 6, 7, 8, 9 and 10, the SU is set to hop to the channel of the packet that was most recently detected. This minimizes the likelihood of causing interference, as the SU will mainly occupy the intervals between PUs' packet transmissions.



(a) SNR = -6 dB (b) SNR = -3 dB (c) SNR = 0 dB (d) SNR = 3 dB

Fig. 7: Confusion Matrices of the CNN model's guesses at different SNR levels.

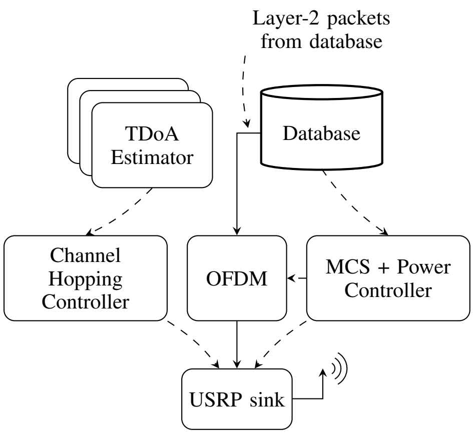


Fig. 8: TX chain

2) Power Controller

In 6 of the 10 possible scenarios faced during the challenge, all the 4 channels are occupied by the PU. As a frequency hopping strategy may be not be sufficient to effectively avoid causing interference in some of these cases, TCD also implemented a power control mechanism that allows the SU to operate underlay. To select an optimal transmit power, the SU continuously queries the database for feedback on its and the PU's throughput and total score.

In particular, the transmitter will increase its transmit gain, if it is observed that increasing the gain led to an increase in the score in the last 500 ms. If this is not observed, the transmitter will start decreasing the gain, until a decrease in the overall score is observed. Besides these mechanisms, an extra guard mechanism exists to quickly react to interference caused to the PU. To do this, the SU monitors the percentage of successfully received throughput by the PU, and if it is below a pre-configured threshold of 90%, the SU quickly decreases its transmit gain.

3) MCS Controller

To calibrate the modulation for the different gains used, in an initial stage, the transmitter will continuously increase its gain and look at the received throughput from the database at various modulations. From these observations, the best modulation for each transmit power will be saved into a table and this table will then be referred to whenever the gain is changed. This will ensure that the best possible modulation order is used for each gain.

E. *Future Directions*

The TCD authors envision, as a possible future direction, a closer integration between the CNN and the frequency hopping system used in the first and second challenge stages, respectively. Despite the overall superior performance of the CNN at scenario classification in comparison to a traditional packet detector, it was not evident how the CNN's outputs could be used for timing the SU's transmissions. Performing such task could be accomplished in two possible ways. One would be to train a machine learning algorithm (e.g. CNN) for the regression task of estimating the TOA and channel of the PU' detected packets. The estimated values would then be used by the frequency hopping system to schedule its channel transitions. The second alternative approach would be to employ reinforcement learning, in particular deep reinforcement learning networks which based on the spectrogram images assesses what are the best actions the SU can take at any instant of time to avoid causing interference.

IV. Winning Agile SU Approach

A. *Overview*

The team from KIT presented a cognitive overlay system based on a Filter Bank Multicarrier (FBMC) physical layer (PHY) [12]. The solution is based on the system that was first demonstrated in Stockholm at DySPAN 2015 [13]. In order to demonstrate a competitive system for the challenge, however, the PHY as well as the cognitive component were improved in several respects.

FBMC is a suitable choice for overlay systems due to its high selectivity and high spectral efficiency. With respect to the previous challenge, the synchronization and equalization algorithms is adapted to reflect current state-of-the-art research. More reliable synchronization and equalization also allows the utilization of higher order modulation schemes and therefore a higher achievable throughput.

To enable a cognitive SU behavior, a classification of the PU scenario based on the extraction of multiple features of the PU's behavior is performed. This knowledge is then used to predict transmission opportunities and thereby completely avoids interference with the PU. Additionally, the current scenario estimation is continuously validated in order to react to unexpected changes in the environment in a timely manner.

For the implementation, the GNU Radio Software Defined Radio (SDR) framework [14] was chosen due to its good performance in real-time applications. The employed hardware comprised two Universal Software Radio Peripheral (USRP) B210 SDRs and two commercial off-the-shelf notebook computers.

A block diagram of the setup is depicted in Fig. 9. The key components as well as the challenges involved are discussed in the following sections.

B. The FBMC PHY Layer

Fig. 10 shows that Filter Bank Multicarrier (FMBC) waveforms offer superior selectivity compared to OFDM since every subcarrier is filtered with a long prototype filter. This allows smaller guard bands to adjacent channels as well as an improved spectral efficiency, while out-of-band radiation and therefore interference with the PU is still considerably lower [15].

Furthermore, an Orthogonal Quadrature Amplitude Modulation (OQAM) with a modulation order of up to 64 is used, enabling only real data symbols to be transmitted. This technique allows adjacent subcarriers to overlap in frequency direction, since the quadrature component can be used to create orthogonality by adding a phase shift of $\frac{\pi}{2}$ between them. This method creates a chess pattern in the time-frequency plane which can easily be reverted in the SU receiver. Interference from surrounding symbols only affects the imaginary part of each data symbol, which is discarded in the receiver. The achievable data rate does not suffer in comparison to a traditional QAM modulation because of the doubled subcarrier density.

For the time synchronization in the SU receiver, a Zadoff-Chu (ZC) preamble is inserted at the beginning of each frame. In order to efficiently insert such a preamble into an FBMC system, the approach in [16] was used, avoiding guard symbols between preamble and payload and therefore reducing overhead. Due to the zero-autocorrelation property of the ZC sequence, the cross-correlation between the preamble sequence and the receive signal exhibits a very sharp peak and therefore enables a very accurate estimation of each frame start.

For fine frequency and phase synchronization as well as channel tracking and equalization, scattered pilots are used in a rectangular arrangement in the time/frequency plane, based on the work of [17]. Since the pilot symbols, in contrast to the data symbols, need to be complex in order to estimate a complex channel transfer function, the imaginary component needs to be kept free from interference. This is done by inserting auxiliary pilots on the same subcarrier in the following symbol. The expected interference on each pilot symbol is calculated in the transmitter yielding the value for each corresponding auxiliary pilot. A simple zero forcing approach is used in the receiver to perform equalization after estimation and linear interpolation of the channel transfer function with help of the pilot symbols. The performance of the equalization is sufficient to abdicate a more extensive frequency synchronization with the employed hardware, which was equipped with GPS disciplined oscillators (GPSDOs) in order to achieve local oscillator frequency offsets smaller than the subcarrier spacing. The existing frame structure, however, allows the estimation of the integer frequency offset, if need be.

During development, an important focus was on the efficient implementation of the signal processing algorithms in order to facilitate the usage of standard consumer laptop computers while being able to support wide bandwidths in real-time. To this end, KIT used the Vector-Optimized Library of Kernels (VOLK) , which provides fast implementations of common mathematical operations in digital signal processing by leveraging the Single Instruction Multiple Data (SIMD) capability of current processors [18].

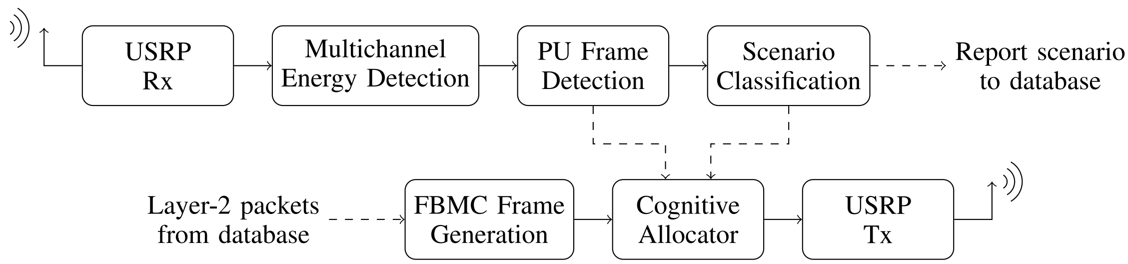
For the challenge, the PHY layer was configured to use 53 of 64 subcarriers per subchannel, leaving DC and the outer 6 subcarriers empty to avoid adjacent channel interference. Considering the employed 16-QAM modulation as well as synchronization and tracking symbols and the inter-frame gaps required for the classification, the maximum throughput of the system in absence of the PU was about 13.8 Mbit/s.

C. Classification of PU behavior

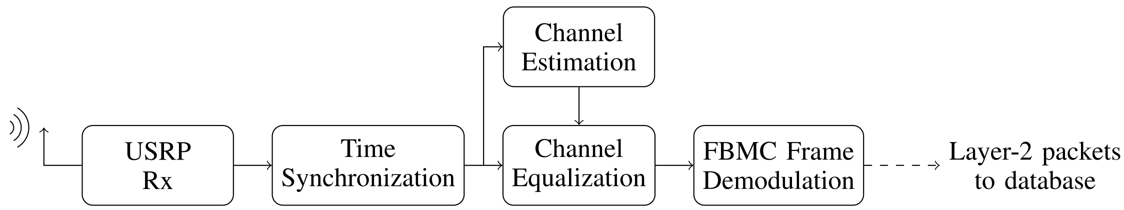
In a first step, the energy in each of the 2.5 MHz wide subchannels is detected by looking at the magnitude-squared output of an N-point FFT. It must be noted that the choice of N directly affects the delay but also the accuracy of the estimation, so there is a trade-off to be considered. Practically, a relatively short FFT with $N = 512$ points (i.e., 128 points per subchannel) was used for the challenge.

In order to reduce the influence of out-of-band radiation of neighboring channels and a possible DC offset, only the $N/2$ bins in the center of each subchannel are integrated. The binary decision whether the channel is occupied or not is then made by comparing this energy estimate with a threshold. To calculate the value for the threshold, the noise floor is estimated continuously between the channels using a low-bandwidth single-pole IIR filter. The detection threshold is then set to the noise floor plus a fixed but user-configurable offset. During the challenge, an offset of 7 dB provided a high detection rate with only very sporadic false alarms. To enable this continuous estimation even during SU transmissions, the high selectivity of the FBMC PHY is crucial and would probably not be possible in the same manner with an OFDM PHY.

At this point, it also showed that transmit amplifier non-linearities impact the detection and classification performance



(a) SU transmitter



(b) SU receiver

Fig. 9: Block diagram of KIT's solution for the SU transmitter and receiver.

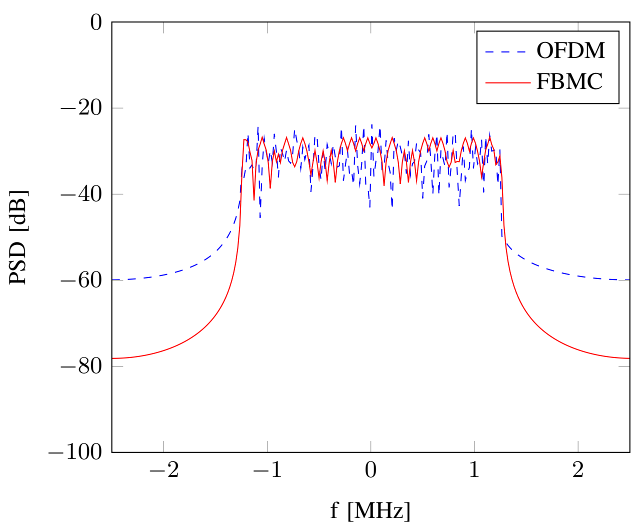


Fig. 10: Comparison between PSDs of OFDM and FBMC when configured to use same number of subcarriers. It can be seen that FBMC provides much higher out-of-band attenuation, facilitating a reduction of the number of guard carriers.

severely. Due to the high power difference between PU and SU packets, even seemingly negligible nonlinearities can lead to interference in neighboring channels, causing false detections as a result.

Based on the binary output of the Multichannel Energy Detection block, three features are extracted: set of used channels, average inter packet delay, and inter packet delay variance. For convenient handling, so-called "frame events" are introduced to abstract from the continuous binary sequence on each channel. To generate them, the PU Frame Detection block looks for PU packets and, at the same time, SU packets are filtered out based on their (different) length.

In order to minimize the probability of missed PU packets, a conservative approach is employed that classifies any detected signal that differs in length from a SU packet as a PU packet.

Every packet detection is then tagged with the channel number it is detected in and a frame start time. Based on this information, it is possible to calculate estimates for the mentioned features by choosing a certain observation window. The number of frame events in this window also represents an important trade-off between classification delay and classification accuracy. During the challenge, the observation window based its decision on 50 consecutive frame events, which amounts to a maximum delay of 500 ms in the second scenario, where only one channel is used every 10 ms. In scenario 6, however, with frames being sent only 2 ms apart on all four subchannels, the classification delay is much smaller.

Finally, a decision tree is used to estimate the current PU scenario. While traversing the tree, the channel occupation is evaluated first, followed by the average inter packet delay and the inter packet delay variance due to their increasing error-proneness. Furthermore, consecutive and identical estimations are used to increase a confidence value in order to reduce the impact of single estimation errors. This confidence value, even though being bounded, of course also has to be chosen carefully as it affects the reaction time in case of a scenario transition. Among the scenarios in Table I, scenario 4 and 9 showed the most potential for estimation errors because they only differ in the variance of the inter packet delay. Generally, the stochastic scenarios (7-9) require a relatively long observation interval for the expectation and variance estimations of the inter packet delay to converge sufficiently.

During the challenge, KIT's classification algorithm proved to be very robust, providing the correct classification results for over 98% of the time. This probably could have been optimized by shortening the observation window as errors only occurred in the transition period between two scenarios when the observation window contained frames that followed different patterns.

D. The Cognitive Allocator

The Cognitive Allocator represents the center of system, where PHY, sensing and classification are combined to predict and make use of transmission opportunities.

Basically, the estimated PU scenario provides the pattern of PU transmissions, but the alignment of this pattern to the current time has still to be done. To achieve this, the extracted frame events are used. Based on the real-time information on the PU frame arrival time, the characteristics of the current scenario can be used to predict which time-frequency resources can be used.

As soon as the time-frequency resources are allocated, FBMC frames are assigned to them. Since FBMC is not efficient for very short packets due to its long pulse shaping filter and the Cognitive Allocator exactly knows how much time can be safely allocated, the frame length is optimized for a good overhead-to-payload ratio. In the challenge, every frame carried aggregated 12 layer-2 packets from the database with 64 bytes each, totalling 768 bytes of payload data and a total length of about 1.7 milliseconds.

While the usage of unoccupied channels is straight-forward, interweaving packets between PU transmissions is a more challenging task. As the inter packet delay is on the order of a few milliseconds for many scenarios, timing and delay management is critical. In the end, the total delay between sensing and transmission at the SU could be brought down to 1 millisecond, which is sufficient as it allows the usage of short inter packet gaps with a length of only 5 milliseconds.

In the stochastic scenarios, where the inter packet delays are sampled from a Poisson distribution, the time occupied by the SU frame is chosen such that the interference probability does not exceed 5%.

This approach works well for a static scenario but can lead to interference when the PU changes its behavior. To cope with this, a conservative and a more aggressive mode were implemented. In the aggressive mode, the Cognitive Allocator completely relies on the accurate classification of the scenario, which can lead to interference until the classification stabilizes again. The conservative approach implements an "override" for the classification results, temporarily blacklisting channels where PU frames have been registered that do not match the current scenario. This of course comes at the cost of SU data rate but can considerably reduce the interference, especially when the PU changes its behavior frequently.

During the challenge, the first round was done in conservative mode, while the second round employed the more aggressive mode. Fig. 11a indicates that, in the second round, the increase in interference was negligible while the (successfully received) throughput of layer-2 data was considerably improved, reaching about 12 Mbit/s and therefore almost the theoretical maximum data rate in certain scenarios.

E. Implementation Aspects and Possible Improvements

Even though the solution works well in practice, there is still room for improvement. The presented system does not make use of multiple antennas for the transmit or receive process for the sake of simplicity. Straight-forward improvements could therefore include beamforming and combining techniques such as Maximum Ratio Combining.

The main issue, however, that team KIT encountered during design and testing is the delay caused by the various buffers of GNU Radio, the USB connection, and the USRP. Shortening the "critical path" between receive and transmit antenna and adapting operating system settings helps a lot but delays on the order of a few microseconds still are not achievable.

One possible remedy besides the optimization of the different parameters discussed in the previous sections is the implementation of critical features on the FPGA, e.g., by using the RF Network on Chip (RFNoC) framework [19].

With very short delays, mechanisms could be implemented that detect a PU OFDM frame while it is still in its cyclic prefix and immediately shut down any ongoing SU transmissions, therefore providing a high level of safety against interference. FPGA implementations could also be used to offload computation-heavy tasks as they are usually encountered in synchronization and equalization algorithms.

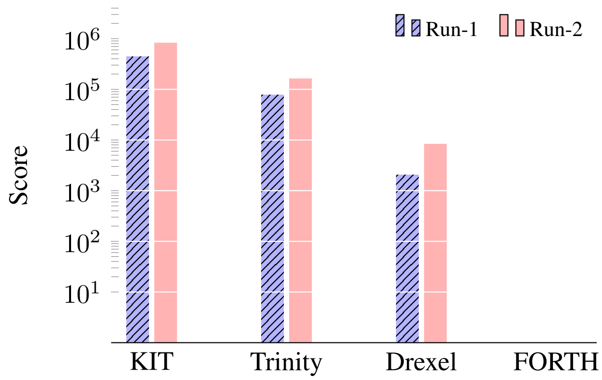
V. Challenge Results and Discussion

Two runs of the challenge, each of 20 minutes with two phases, were carried out to give the teams a second opportunity to fine tune their algorithms. The winner for each of the two phases was selected based on the best score out of the two runs. The availability of the dedicated spectrum and reproducible PU spectral occupancy pattern made the performance results repeatable.

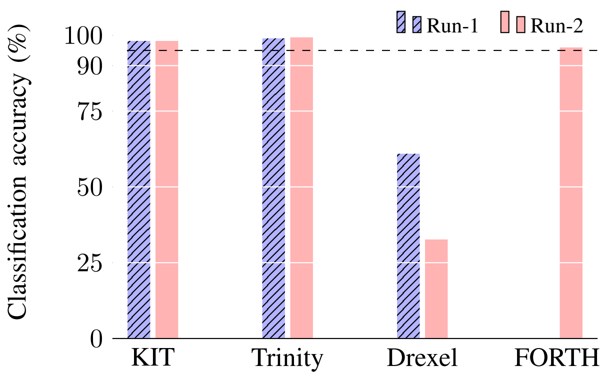
The final agile spectrum usage score is summarized in Figure 11a. Team KIT achieved the highest spectrum usage score during both runs with their advanced FBMC physical layer and cognitive allocator. As shown in Figure 11b, team Trinity achieved the highest situational awareness score of 99.31% using the non-linear deep learning model which had rich internal state representations. Team FORTH also achieved a good score of 99.04 during their second run. Team KIT's expert feature based classifier performed quite well with an overall accuracy of 98.14% which turned out to be one of the best solutions for both phases.

VI. Future benchmarking

This second IEEE DySPAN spectrum challenge was an evolution of the first challenge organized at IEEE DySPAN 2015 [1]. To strengthen the benefits of learning and situational awareness, it was decided to predefine specific patterns for the PU. The PU in 2015 was too irregular and impossible to predict. In addition, the PU PHY was replaced by an OFDM PHY, instead of the IEEE 802.15.4 PHY used in the first challenge. A part from these changes, the core logic and setup of the challenge (implemented by a central database that controls SU and PU packet flows) was reused. We made the source code of the challenge core database, the metric



(a) Agile spectrum usage score



(b) Situation awareness score

Fig. 11: Final scores

and PU scenarios of the IEEE DySPAN challenge 2017, and the two winning solutions as described in this paper public through the github repository mentioned in the introduction (footnote 2). Below, we describe two example scenarios for future experiments building on the current results and code.

First, it is possible to rerun the challenge exactly as it was defined, by testing both winning solutions in a specific and new environment. While rerunning the winning solutions, it is also possible to test a novel improved SU design, that can then then benchmarked against the current best solutions. If a better score is obtained, a new record can then be established. If the source code of the new winner is made available, future solutions can then again be benchmarked against the new winner as well. As the source code for the challenge database, and the two winners as described in this paper are shared in a joint github repository, this avenue is already possible.

Alternatively, it is possible to start from the challenge core database, and change the rules, metrics or PU characteristics. By doing so, a new challenge is defined, emphasizing novel research problems or aiming to steer research effort towards a well defined problem. This requires somebody to take the lead in reshaping the problem statement, and altering the metric or database. By doing so, a new series of winners can be established, targeting a slightly different version of the challenge. We can compare this to the Guiness book of records, where there are numerous winners, for numerous types of world records.

While sharing the challenge source code, and the code for the winner on GitHub, is a good starting point for organizing such benchmarking series, we should realise that the performance of each run is expected to differ a lot. Depending on the size or the room, the level of interference, or the exact hardware used to run the code, a different performance is expected. To minimize the impact of the environment or hardware on the performance, it would be better if there would be a test facility where competing solutions could be tested, continuously using the same environment and hardware. While the establishment of such facility is at the moment future work, it is for sure something the DySPAN community should consider.

VII. Conclusions

For the second time, the DySPAN spectrum challenge brought together researchers to demonstrate and compare their solutions for cognitive SU systems in a scenario with a highly dynamic, OFDM-based PU. The challenge offered an opportunity for the exchange of experience, ideas, and visions for the development of practical DSA systems.

The first phase of the challenge showed how recent advances in the research on machine learning can be leveraged and applied to a cognitive radio context. The team from Trinity College, Dublin, achieved the highest situation awareness score with over 99% classification accuracy by successfully employing deep learning techniques to classify PU behavioral patterns. Their classification approach was both highly accurate and fast, which resulted in this near-perfect score.

In the second phase of the challenge, SU throughput and PU interference were rated. The winning solution for this task was presented by the team from the Karlsruhe Institute of Technology. Their solution combined a spectrally efficient FBMC waveform with the prediction of transmission opportunities through continuous classification of the PU, therefore completely avoiding interference. By the effective use of the available time-frequency resources, a peak throughput of about 12 Mbit/s was achieved.

Finally, we can say that the spectrum challenge showed that the practical implementation of DSA systems is still a challenging task. In order to create a lasting impact and to enable reproducibility, the code for the PU as well as the winning SU solutions were published online. This way, new solutions can be benchmarked and compared with the existing systems, possibly establishing a series of records and continually improving the system's design and algorithms.

In order to allow better control over the environment, future spectrum challenges could also be held in a test facility, providing exactly identical conditions for all participants. Alternatively, the freely available code for the PU and database server could be used to adapt the challenge to focus on different problems encountered in the research on spectrum sharing.

Acknowledgments

The authors are grateful to National Instruments and Mathworks for the hardware support during the challenge and the final prizes. The authors would also like to thank all challenge participants for their enthusiasm and co-operation.

The team from KIT would like to thank Nicolas Cuervo for his valuable help in developing and setting up the testbed as well as Holger Jäkel and Friedrich Jondral for their support.

The TCD team would like to acknowledge Andre Puschmann, Justin Tallon, Maicon Kist, and Christian Bluemm for their help in development of our solution. Also, TCD would like to acknowledge that this work was partially funded by the European Union's Horizon 2020 under grant agreements no. 645274 (WiSHFUL project) and no. 732174 (ORCA project), and by the Science Foundation Ireland (SFI) under grant no. 13/RC/2077.

The KU Leuven team would like to acknowledge Tom Vermeulen for his enormous help during the definition of the challenge, making the challenge website, testing the PU code, taking care of the hardware and finally running the challenge during IEEE DySPAN 2017. Also, KUL would like to acknowledge that this work was partially funded by the IWT SBO project SAMURAI, and by the European Union's Horizon 2020 under grant agreement no. 732174 (ORCA project).

References

[1] S. Rajendran, B. V. den Bergh, T. Vermeulen, and S. Pollin, "IEEE 5G Spectrum Sharing Challenge: A Practical Evaluation of Learning and Feedback," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 210–216, November 2016.

[2] S. Geirhofer, L. Tong, and B. M. Sadler, "A measurement-based model for dynamic spectrum access in wlan channels," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, Oct 2006, pp. 1–7.

[3] ——, "Cognitive radios for dynamic spectrum access - dynamic spectrum access in the time domain: Modeling and exploiting white space," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 66–72, May 2007.

[4] M. López-Benítez and F. Casadevall, "Spectrum usage models for the analysis, design and simulation of cognitive radio networks," in *Cognitive radio and its application for next generation cellular and wireless networks*. Springer, 2012, pp. 27–73.

[5] L. Stabellini, "Quantifying and modeling spectrum opportunities in a real wireless environment," in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[6] F. Paisana, A. Selim, M. Kist, P. Alvarez, J. Tallon, C. Bluemm, A. Puschmann, and L. DaSilva, "Context-aware cognitive radio using deep learning," in *DySPAN*, 2017.

[7] J. Tallon, C. Blümm, A. Puschmann, F. Paisana, J. van de Belt, P. Di Francesco, N. J. Kaminski, and H. Ahmadi, "Coexistence through adaptive sensing and markov chains." in *DySPAN*, 2015, pp. 7–8.

[8] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[9] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[12] F. Wunsch, S. Koslowski, S. Müller, N. Cuervo, H. Jäkel, and F. K. Jondral, "A Cognitive Overlay System Based on FBMC," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Mar. 2017, pp. 171–172.

[13] A. Kaushik, F. Wunsch, A. Sagainov, N. Cuervo, J. Demel, S. Koslowski, H. Jäkel, and F. Jondral, "Spectrum sharing for 5G wireless systems (Spectrum sharing challenge)," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Sept 2015, pp. 1–2.

[14] T. W. Rondeau, "On the GNU Radio Ecosystem," in *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*. Wiley, 2015.

[15] B. Farhang-Boroujeny, "OFDM Versus Filter Bank Multicarrier," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 92–112, May 2011.

[16] W. Chung, C. Kim, S. Choi, and D. Hong, "Synchronization Sequence Design for FBMC/OQAM Systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 7199–7211, 2016.

[17] T. H. Stitz, T. Ihalainen, A. Viholainen, and M. Renfors, "Pilot-based synchronization and equalization in filter bank multicarrier communications," *Eurasip Journal on Advances in Signal Processing*, 2010.

[18] T. W. Rondeau, N. McCarthy, and T. OShea, "SIMD programming in GNURadio: Maintainable and user-friendly algorithm optimization with VOLK." *Conference on Communications Technologies and Software Defined Radio (SDR 12)*, 2012.

[19] J. Malsbury and M. Ettus, "Simplifying FPGA Design with a Novel Network-on-chip Architecture," in *Proceedings of the Second Workshop on Software Radio Implementation Forum (SRIF '13)*, 2013, pp. 45–52.

Felix Wunsch received his B.Sc. and M.Sc. Degree in electrical engineering from the Karlsruhe Institute of Technology (KIT), Karlsruhe, in 2012 and 2015, respectively. At KIT, he is now working as a research associate at the Communications Engineering Lab (CEL) and pursuing the PhD degree. His research focuses on MIMO systems and waveforms for low power wide area networks.

Francisco Paisana Francisco Paisana, a native of Lisbon, received his Master's Degree in electrical engineering and computers from Instituto Superior Tecnico, Portugal in 2012. He subsequently completed his PhD in wireless communications at Trinity College Dublin in 2016. He is currently a PostDoc at CONNECT Research Centre, Trinity College Dublin. His research interests include software-defined radio, spectrum sharing, machine learning, signal processing and cognitive radio.

Sreeraj Rajendran received his Masters degree in communication and signal processing from the Indian Institute of Technology, Bombay, in 2013. He is currently pursuing the PhD degree in the Department of Electrical Engineering, KU Leuven, Belgium. Before joining KU Leuven, he worked as a senior design engineer in the baseband team of Cadence and as an ASIC verification engineer in Wipro Technologies. His main research interests include machine learning algorithms for wireless and low power wireless sensor networks.

Ahmed Selim received his PhD degree in wireless communications at Trinity College Dublin in 2012. He worked for Alcatel-Lucent Bell-Labs Ireland for a year. Since 2014, he works as a research fellow in Trinity College Dublin. His research interests include digital signal processing, machine learning techniques for wireless communications, networks management and computer graphics.

Pedro Alvarez received is PhD in Electronic and Electrical Engineering from Trinity College Dublin in 2017. He has worked since in Trinity College Dublin as a Research Fellow. His research focus on the integration of optical and wireless networks and on Dynamic Bandwidth Allocation algorithms for Passive Optical Networks.

Sebastian Müller received his B.Sc. degree in electrical engineering and information technology from the Karlsruhe Institute of Technology (KIT), Karlsruhe, in 2014. He is currently pursuing his M.Sc. degree at Communications Engineering Lab (CEL), working on wireless technologies with software defined radio focus.

Sebastian Koslowski received his graduate degree in electrical engineering from the Karlsruhe Institute of Technology (KIT) in 2011. He is working as a research associate at the Communications Engineering Lab (CEL) and pursuing the PhD degree. His research focuses include filter bank multicarrier waveforms and software radio.

Bertold Van den Bergh is a PhD student at the KU Leuven Electrical Engineering Department. His research is focused on reliable communication systems for UAVs and other applications. Before starting his PhD he obtained his masters degree in Microelectronics at KU Leuven in 2013 and did an internship at Flanders Make on wireless localization and tracking of mechatronic systems. He also worked at ETH Zurich on the Swarmix project.

Sofie Pollin obtained her PhD degree at KU Leuven with honors in 2006. From 2006-2008 she continued her research on wireless communication, energy-efficient networks, cross-layer design, coexistence and cognitive radio at UC Berkeley. In November 2008 she returned to imec to become a principal scientist in the green radio team. Since 2012, she is tenure track assistant professor at the electrical engineering department at KU Leuven. Her research centers around Networked Systems that require networks that are ever more dense, heterogeneous, battery powered and spectrum constrained. Prof. Pollin is BAEF and Marie Curie fellow, and IEEE senior member.