Wi-Fi 6 Cross-Technology Interference Detection and Mitigation by OFDMA: an Experimental Study

Thijs Havinga, Xianjun Jiao, Wei Liu, Baiheng Chen, Adnan Shahid and Ingrid Moerman
IDLab, *Department of Information Technology*, Ghent University - imec
Ghent, Belgium
{Thijs.Havinga, Xianjun.Jiao, Wei.Liu, Colvin.Chen, Adnan.Shahid, Ingrid.Moerman}@UGent.be

***Abstract*—Cross-Technology Interference (CTI) poses challenges for the performance and robustness of wireless networks. There are opportunities for better cooperation if the spectral occupation and technology of the interference can be detected. Namely, this information can help the Orthogonal Frequency Division Multiple Access (OFDMA) scheduler in IEEE 802.11ax (Wi-Fi 6) to efficiently allocate resources to multiple users in the frequency domain. This work shows that a single Channel State Information (CSI) snapshot, which is used for packet demodulation in the receiver, is enough to detect and classify the type of CTI on low-cost Wi-Fi 6 hardware. We show the classification accuracy of a small Convolutional Neural Network (CNN) for different Signal-to-Noise Ratio (SNR) and Signal-to-Interference Ratio (SIR) with simulated data, as well as using a wired and over-the-air test with a professional wireless connectivity tester, while running the inference on the low-cost device. Furthermore, we use *openwifi*, a full-stack Wi-Fi transceiver running on software-defined radio (SDR) available in the w-iLab.t testbed, as Access Point (AP) to implement a CTI-aware multi-user OFDMA scheduler when the clients send CTI detection feedback to the AP. We show experimentally that it can fully mitigate the 35% throughput loss caused by CTI when the AP applies the appropriate scheduling.**

I. Introduction and related work

Partly thanks to the rise of the Internet-of-Things (IoT), there is an ongoing increase of wireless communication devices operating in unlicensed bands. Therefore, Cross-Technology Interference (CTI) becomes an increasingly important problem. Particularly in the 2.4GHz band, a multitude of technologies coexist, such as Wi-Fi (IEEE 802.11), Bluetooth/Bluetooth Low Energy (BLE), and Low-Rate Wireless Personal Area Network (LR-WPAN) devices based on the IEEE 802.15.4 standard like ZigBee, WirelessHART or Thread [1]. Although newer Wi-Fi standards can also operate in the 5GHz and 6GHz bands, the 2.4GHz band is still actively used due to its higher coverage range and compatibility with legacy devices. In our previous work [2], we demonstrated that even at a high Signal-to-Noise Ratio (SNR) of $23\,\mathrm{dB}$, CTI can significantly impact the Packet Error Rate (PER) of Wi-Fi. When employing a high Modulation and Coding Scheme (MCS) such as MCS 7, even relatively low interference power from ZigBee that results in an Signal-to-Interference Ratio (SIR) of approximately $15\,\mathrm{dB}$, can lead to 100% PER.

There are a few methods to manage spectrum access under CTI. Several technologies apply a Clear Channel Assessment (CCA), usually consisting of an energy detection and a signal detection step. Energy detection only works when the interference is significantly strong. When it is performed on different bandwidths with different thresholds, this may lead to asymmetric access opportunities. Moreover, energy detection at the transmitter may not be a good indicator of the CTI level perceived by the receiver, causing the hidden or exposed node problem. Lastly, since signal detection—usually performed by correlating the incoming signal with a known sequence—is technology-specific, it does not work for CTI [1].

Other CTI detection methods rely on high-level metrics like PER or receiver error codes [3]. However, since [3] relies on the temporal gap between errors, a fixed packet size and interval are assumed. Even if this assumption is met, the spectral location of CTI cannot be determined.

Technology recognition can be used to improve networks using domain-specific knowledge about the detected technology. For example, while ZigBee uses a fixed channel, BLE uses a frequency hopping scheme. To mitigate ZigBee, knowing the spectral location can be used to avoid future interference, while for BLE real-time detection is needed for each transmission to overcome an ongoing interference packet. While technology recognition can be valuable, it typically requires additional hardware, such as a separate software-defined radio acting as sensing engine [4]. Alternatively, communication devices may perform a low-resolution spectrum scan using existing hardware, but it requires an idle period, as demonstrated by [5]. These limitations make it impractical for low-cost deployments, which we will address in this work.

Channel State Information (CSI) is derived from the Fast Fourier Transform (FFT) of the training sequence in a Wi-Fi packet, which is required to demodulate the packet. Hence, it can also act as a spectrum sensing module without extra hardware cost. Several Wi-Fi chips expose the CSI to the user, including the low-cost ESP32-C6 [6], which we use in this work. However, there is little existing work on using CSI for interference detection, which we discuss hereafter.

The work in [7] detects interference of Bluetooth, ZigBee and microwave ovens using the CSI of Wi-Fi devices with 52 subcarriers. In their subsequent work [8], they also identify which subcarriers are interfered by estimating the distortion peak, which is assumed to be at the center of a fixed bandwidth. In static environments using 500 CSI snapshots, ZigBee and Bluetooth are correctly detected in more than 90% of the time with almost 100% true negative rate, while in dynamic environments it drops to about 86%. In both cases,

 DOI: 10.1109/EuCNC/6GSummit63408.2025.11037177

the interferer was at $1\,\mathrm{m}$ distance from the Wi-Fi receiver, while at $4\,\mathrm{m}$ the ZigBee detection rate dropped to 25%.

The authors of [9] use different machine learning methods to classify interference of Bluetooth, Wi-Fi and microwave ovens using a Wi-Fi chip which provides 30 CSI values per snapshot. By creating heavy Wi-Fi traffic such that during a sample period of $1\,\mathrm{s}$ 400 CSI snapshots are recorded, the classification accuracy of the best method reaches around 90% accuracy, but the SNR and SIR are not specified.

Orthogonal Frequency Division Multiple Access (OFDMA) is introduced to Wi-Fi in the IEEE 802.11ax standard to serve multiple users at the same time using different subcarrier sets, called Resource Units (RUs). This reduces the time spent for contention to access the channel and limits the overhead of the preamble, thereby lowering latency and increasing throughput. The standard defines the necessary guidelines to ensure interoperability, but the scheduling of the RU width (number of subcarriers), RU index (spectral location) and MCS to the different users, etc., is left to vendors to implement. Several works consider OFDMA scheduling while taking into account the channel variation across the occupied spectrum, mainly due to small-scale fading. The authors of [10] use a per-subcarrier channel gain to maximize the overall data rate by allocating RUs and MCSs appropriately. The work in [11] includes channel fading from CSI feedback into the scheduler for allocating RUs and MCSs. While this is a valid approach in interference-free channels, CTI usually constitutes as a boost in the magnitude of the CSI, while the packet reception performance is actually worse. In our previous work [2], we have shown the concept of puncturing an RU—meaning no data is sent on subcarriers belonging to a certain RU—when affected by CTI in a single-user (SU) scenario. There it is assumed that the interference is known to the AP. In this work, we realize the CTI detection feedback and we create a CTI-aware multi-user (MU) OFDMA scheduler as explained next.

Since Wi-Fi 6, packets are modulated on a maximum of 242 instead of 52 active OFDM subcarriers per 20MHz bandwidth for 802.11a/g, or 56 for 802.11n/ac. The CSI in Wi-Fi 6 is the FFT of the High Efficiency Long Training Field (HE-LTF) in the preamble, after compensating for the known sequence. The highest frequency resolution is achieved with a $12.8\,\mu\mathrm{s}$ long HE-LTF, which consists of 242 active subcarriers, which is more than four times the amount of earlier standards. This allows us to classify CTI with only one CSI snapshot obtained from regular Wi-Fi traffic on low-cost hardware, eliminating the need to transmit multiple CSI snapshots to a high-end AP for processing.

Since commercial Wi-Fi 6 APs lack control over low-level OFDMA features, we use openwifi [12] to implement a CTI-aware scheduler. We show that with spectral knowledge of the CTI, for MU packets, an affected RU can be assigned to a user that experiences less CTI due to its physical location, and this improves the network's throughput.

To the best of our knowledge, this work is the first to achieve the following combination of attractive properties:

1) Interference detection including spectral location and technology classification;
   - Requiring only a single CSI snapshot obtained from an existing FFT module in the Wi-Fi chipset;
   - Executable on very low-cost ($\approx$ \$8) hardware.
2) A CTI-aware OFDMA scheduler using CSI-based technology classification demonstrating to mitigate 35% loss in throughput due to CTI.

II. System Model

Given a known transmitted OFDM signal in the frequency domain $\boldsymbol{x}$ (namely the HE-LTF in 802.11ax), a transmitted interference signal $\boldsymbol{i}$ and noise vector $\boldsymbol{n}$, the received OFDM symbol $\boldsymbol{y}$ in a frequency selective fading environment is then formulated as:

$$\boldsymbol{y} = \boldsymbol{h_w} \odot \boldsymbol{x} + \boldsymbol{h_i} \odot \boldsymbol{i} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{h_w}$ and $\boldsymbol{h_i}$ are the channel responses of the Wi-Fi and interference signal, respectively, and $\odot$ represents element-wise multiplication. All signals are vectors of complex values with dimension M as the number of active subcarriers of the Wi-Fi OFDM symbol. The observed CSI $\hat{\boldsymbol{h}}$ by a Wi-Fi receiver is then formulated as:

$$\hat{\boldsymbol{h}} = \boldsymbol{y} \oslash \boldsymbol{x} = (\boldsymbol{h_w} \odot \boldsymbol{x} + \boldsymbol{h_i} \odot \boldsymbol{i} + \boldsymbol{n}) \oslash \boldsymbol{x}, \tag{2}$$

where $\oslash$ represents element-wise division. From the characteristics of $\boldsymbol{i}$, such as its presence on different subcarriers, as well as its magnitude and phase change due to different modulation types, the spectral location and technology can be derived. Given a newly observed CSI $\hat{\boldsymbol{h}}$, the task is thus to estimate $\boldsymbol{i}$ and extract features from $\boldsymbol{i}$ to classify the type of CTI, if any. This is a typical task for a Convolutional Neural Network (CNN). By training it on different $\hat{\boldsymbol{h}}$ with and without interference, the model learns each of the individual components of the received signal in order to determine $\boldsymbol{i}$ from a new CSI $\hat{\boldsymbol{h}}$. The benefit of using machine learning over purely rule-based methods is that it does not require threshold tuning and it can easily be retrained with real-life data.

A typical scenario where this theory can be applied is for a Wi-Fi channel in the $2.4\,\mathrm{GHz}$ ISM band, which overlaps with multiple LR-WPAN and BLE channels, as shown in Fig. 1. Before Wi-Fi 6, the entire bandwidth (indicated by blue colour) is either occupied or free; with OFDMA enabled, Wi-Fi 6 allows dividing subcarriers into several RUs, the boundaries of the RUs comprised of 106 subcarriers (106-tone RUs) are highlighted by dashed blue lines.

Fig. 2 shows the magnitude and phase of three typical CSI snapshots captured by ESP32-C6 on Wi-Fi channel 1, either with no interference, or with CTI from LR-WPAN or BLE. In real-life scenarios, packets with different SNR and SIR levels will be received. The SNR represents the power of $\boldsymbol{h_w} \odot \boldsymbol{x}$ as compared to $\boldsymbol{n}$ and thus influences the quality of the CSI. Similarly, the SIR refers to the power ratio of $\boldsymbol{h_w} \odot \boldsymbol{x}$ with respect to $\boldsymbol{h_i} \odot \boldsymbol{i}$. The higher the SIR, the weaker the interference signal, the more difficult it is to detect and characterize CTI.

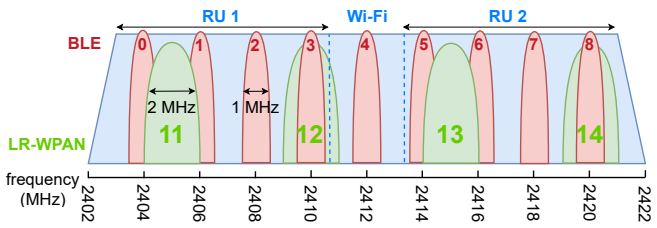

Fig. 1. Spectrum used by Wi-Fi channel 1 (dotted lines show borders between two 106-tone RUs), LR-WPAN channels 11-14 and BLE channels 0-8.

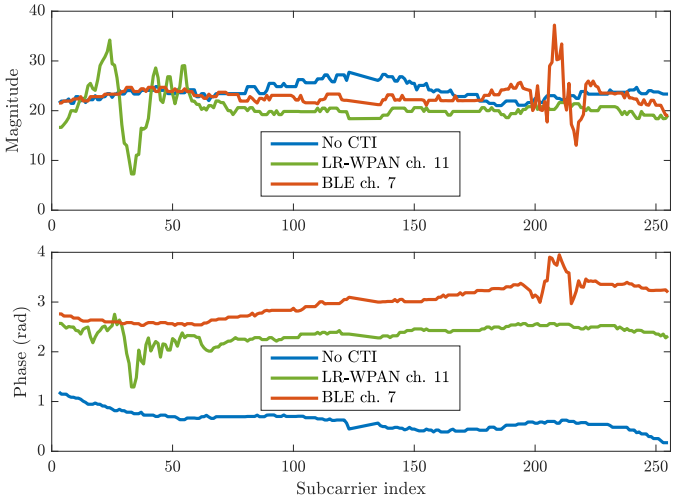

Fig. 2. Magnitude and phase of CSI snapshots captured at high SNR by the ESP32-C6 when not interfererd (blue), interfered with LR-WPAN channel 11 (green), or BLE channel 7 (red) with an SIR of 1 dB.

III. Methodology and Proposed CNN

We create a CNN that has a matrix input $\hat{\mathbf{H}} \in \mathbb{R}^{2\times 242}$, representing the real and imaginary part of the 242 active subcarriers of the CSI. Note that the CSI is thus not converted to magnitude and phase components to avoid this computational step on-device. The output is 14 classes $\{C_1, C_2, \ldots, C_{14}\}$, where C_1 is no interference, C_2 to C_5 are the interference of the LR-WPAN channels and the remaining classes, C_6 to C_{14}, pertain to the interference of the BLE channels (see Fig. 1). In order to keep the size of the CNN small, and since there are less features to be extracted compared to e.g. image recognition, it consists of only two convolutional layers. Afterwards, in order to understand the complex relationships between features like the shape of the overall CSI and the distortions due to interference, three fully-interconnecting layers with generalized matrix multiplication (Gemm) are applied. Each layer uses Rectified Linear Unit (ReLU) activation and at the final stage, a LogSoftmax function generates the probability distribution across the classes. See Fig. 3 for the full model.

In order to create a general model that is not dependent on specific transmitter, receiver or environmental characteristics, we opt to use artificial data to train the model. The data is generated based on the physical layer specifications and multipath channel models of the relevant standards. Namely,

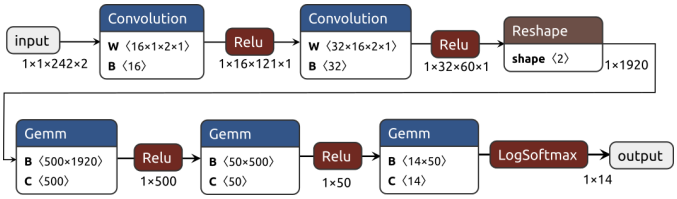

Fig. 3. Diagram of the CNN with number of **W**eigths, **B**iases and **C**hannels.

we prepare a 242-tone HE-LTF field in MATLAB and generate several LR-WPAN and BLE waveforms using the Communications and Bluetooth Toolbox, respectively. We shift the LR-WPAN and BLE signals in the frequency domain to one of the overlapping channels within Wi-Fi channel 1 as shown in Fig. 1. The HE-LTF and the interfering signals are convoluted with a different realization of the IEEE 802.11ax indoor spatial channel models B or C [13], suitable for small to medium indoor environments. No carrier and sampling frequency offset are added to the HE-LTF symbol, because the receiver will compensate for this already using the legacy preamble training fields. Afterwards, we scale the IQ samples of the interference according to a certain SIR of the full 20 MHz and superpose them with the HE-LTF. Then we add white Gaussian noise according to the SNR, which is based on the signal power without channel response and interference. Following, we perform the 256-point FFT on it, and compensate it with the known HE-LTF sequence to get the CSI at the active subcarriers.

The output of CSI of the ESP32-C6 are two 8-bit signed integers per subcarrier, one for the real part and one for the imaginary part. The typical range of the magnitude of the CSI is experimentally determined by receiving packets from the highest to lowest decodable RSSI range. The CSI output from the generated data is scaled such that the average magnitude is in this range. For each SIR and SNR value (as determined below), we generate 2000 CSI snapshots with random noise and realization of the channel model, and let a random part of the CTI signals overlap with the HE-LTF.

The CNN is implemented in Python using the PyTorch library. Models are initially trained based on CSI obtained at specific SNR and SIR to determine the range at which sufficient accuracy is achieved. The general model is trained on data with an SNR range of 14 dB to 24 dB and an SIR range of 1 dB to 15 dB, varied in 1 dB steps. The generated data is divided into 80% training and 20% validation data. We use the Adam optimizer with a learning rate of 0.001 and a batch size of 256 for 200 epochs with negative log likelihood loss function. Training was performed on an 8-core Intel Xeon CPU E5-2620 v4 at 2.10GHz with 32GB RAM, which took about 10 hours. We convert the PyTorch model to the Open Neural Network Exchange (ONNX) format. It is then pre-processed and optimized without quantization for the ESP32.

The firmware is open source[1] and consumes 4.77MB of the

[1]Link to the source code for training the model and the firmware: `https://github.com/HavingaThijs/CSI-Based_CTI_Detection`.

available 8MB flash memory on the ESP32, of which 0.91MB is occupied by the ESP-IDF framework including Wi-Fi driver. The model runs with a clock frequency of $160\,\mathrm{MHz}$ on the ESP32-C6 single-core RISC-V RV32IMAC CPU, which lacks dedicated floating-point operations. The result is a considerable inference time (i.e. $680\,\mathrm{ms}$). Techniques such as removing packets at too low SNR, applying integer quantization, or decreasing the model size when only the spectral location needs to be detected, can be used to reduce the inference time.

IV. Performance evaluation of CTI detection

Three different tests are done to evaluate the performance of the classification model, namely an over-the-air (OTA) test, a test using coaxial cable, and a test with artificial data on a host PC. The OTA test includes hardware impairments as well as effects from unseen channel responses. The cable test eliminates the channel response, while the artificial data shows the performance without any hardware impairments on a non-constraint device.

In order ensure 100% interference ratio, we utilize the R&S®CMW270 wireless connectivity tester as an IEEE 802.11ax AP, which simultaneously replays clean IQ samples of the waveforms of LR-WPAN channels 11-14 or BLE channels 0-8 via the general purpose RF output. The AP is set to use channel 1 and create HE-SU packets using MCS 0. The Wi-Fi tester is set to its maximum transmit power of $-3\,\mathrm{dBm}$ with antennas connected, while the ESP32-C6's antenna port is connected to an RF terminator. In this way, the ESP32-C6 is only able to receive the strong signal from the nearby Wi-Fi tester. By moving the ESP32-C6 around in approximately a meter range, we control the SNR relative to the AP's power. Still in some cases a clear unintentional interference was visible in the CSI. These measurements were manually discarded. The SNR is estimated based on RSSI reports during the measurement and the receiver sensitivity of the ESP32-C6. SIR is determined by the power difference between the RF ports for Wi-Fi and the interference signal. For the cable test, the ESP32-C6's antenna port is connected to the output of a 2-way power combiner, one input is connected by coaxial cable to the tester's RF port for Wi-Fi and one to the RF port for interference. The different RSSI values are obtained by changing the tester's output power. When deploying the model on the ESP32-C6, it could be seen that at lower SNR, the model often judges interference at BLE channel 4, which is exactly in the middle of Wi-Fi channel 1. A possible reason for this is that leakage of the transmitter's or receiver's Local Oscillator (LO) can create distortion in this region. From here on, we use *filtered* accuracy to refer to results without misdetection at BLE channel 4.

We evaluate the performance by running the model on 100 filtered CSI snapshots per class (i.e., no interference, and LR-WPAN or BLE present on one of the overlapping channels with Wi-Fi), at different SNR and SIR on the ESP32-C6. In total, 16,800 filtered snapshots for both the OTA and cable test are collected. The average accuracy is determined by ratio of correctly classified interference source out of all snapshots.

TABLE I
Confusion matrix per technology

Actual \ Predicted		No CTI	LR-WPAN	BLE
No CTI	**OTA**	94.75	3.0	2.25
	Cable	91.5	3.75	4.75
	Artificial	85.53	5.03	9.44
LR-WPAN	**OTA**	15.17	81.46	3.38
	Cable	7.54	87.5	4.91
	Artificial	2.27	95.82	1.91
BLE	**OTA**	9.79	1.8	88.42
	Cable	2.42	3.68	93.9
	Artificial	2.16	0.80	97.03

Finally, we ran the model on a PC with the artificial test set, using the same SNR and SIR values.

Fig. 4 shows the accuracy of CSI-based technology classification for each combination of SNR and SIR of the OTA, cable and artificial test. For the OTA and cable test we show both the original accuracy (bars with borders) and filtered accuracy (bars without borders). In general, the lower the SNR and the weaker the interference (higher SIR), the lower the accuracy, as expected from Equation 2. Two interesting outliers from this general rule can be identified. The accuracy of the OTA test at SIR $15\,\mathrm{dB}$ peaked at $20\,\mathrm{dB}$ SNR; there is a downward trend at higher SNR. This may be because the model has learned that for a smooth CSI (often happens at high SNR), there needs to be a clear distortion visible to classify it as interference. However, in the dynamic environment with unseen channel responses, weak CTI might be easily confused as part of the fading channel. Furthermore, in the cable test at $20\,\mathrm{dB}$ SNR and above, detecting weaker interference (SIR of $8\,\mathrm{dB}$) is better than detecting stronger interference (SIR of $1\,\mathrm{dB}$). A reason for this might be that due to the very strong CTI, the synchronization, frequency or sampling offset estimation of the Wi-Fi receiver is more likely to be wrong, which will hurt the quality of the CSI, hence affecting the accuracy.

For the prior work relying on multiple CSI snapshots, [9] does not specify any signal quality or interference power, and [8] only specifies distances between the Wi-Fi devices and interferer. Assuming $20\,\mathrm{dBm}$ Wi-Fi transmit power with the largest distance between Wi-Fi devices of $4\,\mathrm{m}$ in [8]'s experiment, it would lead to an SNR much higher than $25\,\mathrm{dB}$. Then they achieve an accuracy higher than 90% when the ZigBee interferer is located at $1\,\mathrm{m}$. Our OTA test achieves similar performance under more difficult conditions (see Fig. 4 at SNR $25\,\mathrm{dB}$ and SIR $8\,\mathrm{dB}$). However, when the ZigBee interferer is placed $4\,\mathrm{m}$ away from Wi-Fi receiver, their true positive rate already drops to 25%. This corresponds to roughly $20\,\mathrm{dB}$ SIR assuming ZigBee's transmit power is $0\,\mathrm{dBm}$, whereas in our OTA test, with $25\,\mathrm{dB}$ SNR and $15\,\mathrm{dB}$ SIR, the model still maintains 70% recognition precision.

Next, we analyze the confusion matrix per technology in Table I. It shows the percentage of predicting either no CTI, LR-WPAN or BLE in case of each actual class, obtained for all SNR and SIR using the filtered results. In general it can be seen that from the OTA, to the cable and artificial test, the

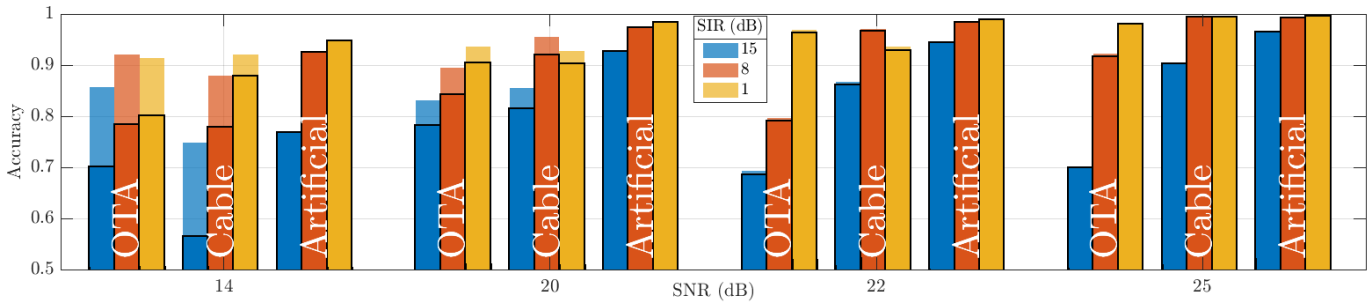

Fig. 4. Classification accuracy of the OTA, cable and artificial tests for different SNR and SIR (note that the accuracy starts at 0.5 for better visibility).

model went from low to high false positive rate (predicting LR-WPAN or BLE when there was no CTI), and from low to high true positive rate. Also, the presence of LR-WPAN is more often classified as "no CTI" than BLE, which can be explained by the fact that the power of BLE is concentrated on a smaller bandwidth, meaning that at the same SIR, BLE has a higher power spectrum density than LR-WPAN, hence its distortion on the CSI is clearer.

To assess the accuracy of the spectral location, we divided the spectrum into four 52-tone RUs, each overlapping with two BLE channels and one LR-WPAN channel. When interference is detected in the OTA test, the correct overlapping RU is detected in 99.8% of the cases.

V. CTI-aware OFDMA Scheduling Results

In order to show the benefit of CTI-aware OFDMA scheduling, we implement a scheduler on an openwifi AP and let the ESP32-C6 perform CTI detection. Openwifi [12] is an open-source implementation of the IEEE 802.11 standard running on a System-on-Chip. The baseband processing and low-MAC is realized on the Field Programmable Gate Array (FPGA); the driver and high-MAC run as Linux kernel modules on the on-chip ARM processor. The baseband processing for downlink OFDMA support following the IEEE 802.11ax standard has been implemented in our previous work [14]. In this work, we make the necessary addition to the driver to control the RU allocation and corresponding parameters like MCS. The STAs perform the CTI classification and send the result to the AP using Wi-Fi frames. Since the two-user OFDMA frames consisting of 2x106-tone RUs have a different CSI dimension than the 242-tone RU used previously, the model has been retrained to incorporate the new CSI dimension.

For this specific OFDMA scheduler we are only interested in whether CTI is detected and its spectral location. Therefore, the AP holds a historical record of 64 interference detection results per RU and per STA within the driver. First, the AP decides upon an RU allocation based on the number of users for which it has data available. Then, users will be assigned to RUs where they experienced the least interference during the record. By using a single CSI snapshot combined with a historical record, the CTI detection time is relatively low, hence the scheduler can capture transient interference and quickly adapt to it.

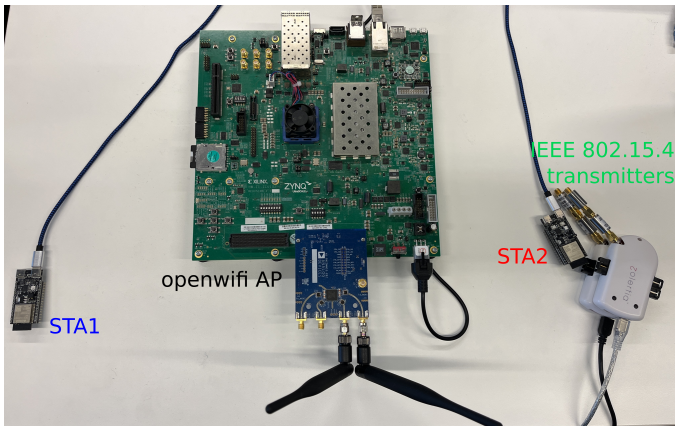

Fig. 5. Experimental setup to validate the CTI-aware OFDMA scheduler on the openwifi AP, when sending downlink packets to STA1 and STA2; only STA2 is interfered by two IEEE 802.15.4 transmitters on channel 14.

To validate the CTI-aware OFDMA scheduler, the experimental setup as shown in Figure 5 is used. Namely, STA1 and STA2 (ESP32-C6 boards) are located on opposite sides of the openwifi AP, which runs on a Xilinx Zynq UltraScale+ MPSoC ZCU102 with an Analog Devices FMCOMMS3 RF front-end. Next to STA2, the two IEEE 802.15.4 transmitters[2] are placed on top of each other. The openwifi AP has only limited output power, thus the STAs need to be close to the AP to stay connected. In order to create the topology where only STA2 experiences interference, the RF outputs of the LR-WPAN transmitters are connected to two $20\,\mathrm{dB}$ attenuators and their output power is set to $-24\,\mathrm{dBm}$. The LR-WPAN devices continuously send request and reply packets of 80 bytes to each other, which equals $3.83\,\mathrm{ms}$ of airtime per packet. Meaning that if each request is received correctly and answered with an equally long reply packet, around 51% duty cycle is achieved. Due to packet losses, in reality the duty cycle of the interference is only around 35%.

The Wi-Fi network runs on Wi-Fi channel 1, and the IEEE 802.15.4 transmitters use channel 14, thus overlapping with 106-tone RU 2 as shown in Figure 1. We compare the CTI-aware scheduler to a naive MU scheduler using a fixed RU

[2]The IEEE 802.15.4 radios are the Zolertia RE-Mote revison A available in the w-iLab.t testbed: https://doc.ilabt.imec.be/ilabt/wilab/hardware.html.

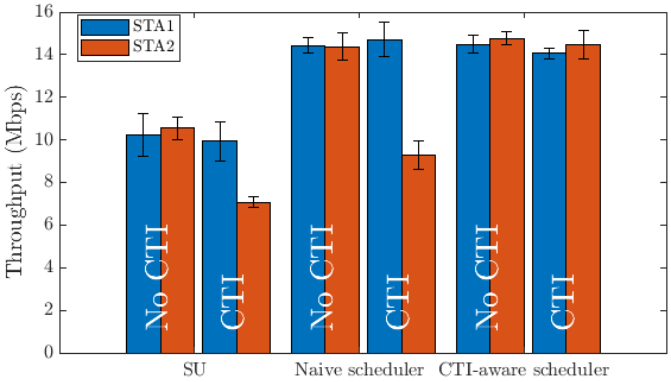

Fig. 6. Downlink throughput of two STAs using SU packets only, MU packets using a naive RU allocation, and MU packets by CTI-aware RU allocation.

allocation, which happens to always allocate a user to the RU where it experiences the worst interference. Furthermore, we evaluate the performance when using SU packets. This packet format always occupies the full bandwidth, which relies on the standard Carrier Sense Multiple Access with Collison Avoidance (CSMA/CA) for CTI mitigation. iPerf2 is used to start UDP traffic to the two STAs simultaneously, with the default payload length of 1470 bytes. The openwifi AP is set to use a fixed MCS 7. The high MCS is chosen to clearly show the impact of CTI. For each scheduler, five tests of one minute are executed and the average throughput reported by the STAs is recorded. In addition, the measurements are repeated also when the CTI is disabled, to obtain a performance baseline.

Figure 6 displays the average throughput for each STA with error bars showing the standard deviation. In general, using SU packets results in lower total throughput, which can be partially explained by the fact that it has more overhead due to the preamble and contention time. For SU packets, it can be seen that when CTI is applied, the throughput towards STA1 (the blue bars) remains roughly stable, while for the STA2, it decreases by about 33% (indicated by the red bar that drops from $10.5\,\mathrm{Mbps}$ to $7\,\mathrm{Mbps}$). Roughly the same observation for the naive MU scheduler can be made, where the throughput of STA1 stays above $14\,\mathrm{Mbps}$ but STA2's throughput dropped from $14\,\mathrm{Mbps}$ to $9\,\mathrm{Mbps}$ ($\approx$35% drop). On the other hand, the throughput reduction for STA2 is negligible when using the CTI-aware scheduler, because the data is now sent on a portion of the spectrum that does not overlap with the CTI.

VI. Conclusions

To combat CTI, this work shows that one snapshot of Wi-Fi 6 CSI can provide sufficient detection accuracy for on-device machine learning-based classification of LR-WPAN and BLE interference, including their spectral location. We have shown that the CTI classification accuracy for different SNR and SIR can go up to 99% in a scenario with high SNR and low SIR, but it drops significantly below $14\,\mathrm{dB}$ SNR due to the decrease of CSI quality. The overall classification accuracy is comparable or better than prior arts that use multiple CSI snapshots. Furthermore, due to the artificially generated training data, we observe that non-idealities such as local oscillator leakage can mislead the model, which should be addressed in future work.

To show that CSI-based CTI classification can be effectively applied for CTI mitigation, we implement a CTI-aware OFDMA scheduler on a software-defined radio. By collecting real-time CTI detection results on the STAs, the AP correctly assigns a clean RU in the frequency domain to a user that originally experiences narrowband CTI. In this way, we prove experimentally that CTI-aware OFDMA scheduling can fully mitigate the 35% throughput loss caused by CTI.

Acknowledgment

This work is funded by EU Horizon projects under Grant Agreement No. 101095738 (6G-SHINE) and No. 101139176 (6G-MUSICAL), and FWO SBO S003921N VERI-END.com.

References

[1] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli, "Low-Power Wireless for the Internet of Things: Standards and Applications," *IEEE Access*, vol. 6, pp. 67 893–67 926, 2018.

[2] T. Havinga, X. Jiao, W. Liu, and I. Moerman, "Experimental Study Towards Efficient Interference Avoidance Using Wi-Fi 6 OFDMA on SDR," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1–6.

[3] D. Croce, D. Garlisi, F. Giuliano, N. Inzerillo, and I. Tinnirello, "Learning From Errors: Detecting Cross-Technology Interference in WiFi Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 347–356, 2018.

[4] J. Fontaine, A. Shahid, R. Elsas, A. Seferagic, I. Moerman, and E. De Poorter, "Multi-band sub-GHz technology recognition on NVIDIA's Jetson Nano," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–7.

[5] M. Chwalisz and A. Wolisz, "Towards efficient coexistence of IEEE 802.15.4e TSCH and IEEE 802.11," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–7.

[6] "ESP32-C6 Wi-Fi 6 & BLE 5 & Thread/ZigBee SoC," https://www.espressif.com/en/products/socs/esp32-c6, Espressif Systems, 2024.

[7] Y. Zheng, C. Wu, K. Qian, Z. Yang, and Y. Liu, "Detecting Radio Frequency Interference for CSI Measurements on COTS WiFi Devices," *2017 IEEE International Conference on Communications (ICC)*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3970225

[8] Y. Zheng, Z. Yang, J. Yin, C. Wu, K. Qian, F. Xiao, and Y. Liu, "Combating Cross-Technology Interference for Robust Wireless Sensing with COTS WiFi," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, 2018, pp. 1–9.

[9] Z. Yang, Y. Wang, L. Zhang, and Y. Shen, "Indoor Interference Classification Based on WiFi Channel State Information," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, G. Wang, J. Chen, and L. T. Yang, Eds. Cham: Springer International Publishing, 2018, pp. 136–145.

[10] K. Wang and K. Psounis, "Efficient scheduling and resource allocation in 802.11ax multi-user transmissions," *Computer Communications*, vol. 152, pp. 171–186, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419305729

[11] S. Tutelian, D. Bankov, D. Shmelkin, and E. Khorov, "IEEE 802.11ax OFDMA Resource Allocation with Frequency-Selective Fading," *Sensors*, vol. 21, p. 6099, 09 2021.

[12] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source IEEE802. 11 SDR implementation on SoC," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–2.

[13] J. Liu, R. Porat *et al.*, "TGax Channel Model Document," IEEE 802.11-14/0882r4, September 2014.

[14] M. Aslam, X. Jiao, W. Liu, M. Mehari, T. Havinga, and I. Moerman, "A novel hardware efficient design for IEEE 802.11ax compliant OFDMA transceiver," *Computer Communications*, vol. 219, pp. 173–181, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366424000926