Machine Learning Enabled Spectrum Sharing in Dense LTE-U/Wi-Fi Coexistence Scenarios

ADAM DZIEDZIC, VANLIN SATHYA [ID], MUHAMMAD IQBAL ROCHMAN, MONISHA GHOSH, AND SANJAY KRISHNAN

University of Chicago, Chicago, IL 60637 USA

CORRESPONDING AUTHOR: VANLIN SATHYA (e-mail: vanlin@uchicago.edu)

ABSTRACT The application of Machine Learning (ML) techniques to complex engineering problems has proved to be an attractive and efficient solution. ML has been successfully applied to several practical tasks like image recognition, automating industrial operations, etc. The promise of ML techniques in solving non-linear problems influenced this work which aims to apply known ML techniques and develop new ones for wireless spectrum sharing between Wi-Fi and LTE in the unlicensed spectrum. In this work, we focus on the LTE-Unlicensed (LTE-U) specification developed by the LTE-U Forum, which uses the duty-cycle approach for fair coexistence. The specification suggests reducing the duty cycle at the LTE-U base-station (BS) when the number of co-channel Wi-Fi basic service sets (BSSs) increases from one to two or more. However, without decoding the Wi-Fi packets, detecting the number of Wi-Fi BSSs operating on the channel in real-time is a challenging problem. In this work, we demonstrate a novel ML-based approach which solves this problem by using energy values observed during the LTE-U OFF duration. It is relatively straightforward to observe only the energy values during the LTE-U BS OFF time compared to decoding the entire Wi-Fi packet, which would require a full Wi-Fi receiver at the LTE-U base-station. We implement and validate the proposed ML-based approach by real-time experiments and demonstrate that there exist distinct patterns between the energy distributions between one and many Wi-Fi AP transmissions. The proposed ML-based approach results in a higher accuracy (close to 99% in all cases) as compared to the existing auto-correlation (AC) and energy detection (ED) approaches.

INDEX TERMS LTE, Unlicensed Spectrum, Wi-Fi, Machine Learning.

I. INTRODUCTION

The growing penetration of high-end consumer devices like smartphones and tablets running bandwidth hungry applications (e.g. mobile multimedia streaming) has led to a commensurate surge in demand for mobile data (pegged to soar up to 77 exabytes by 2022 [1]). An anticipated second wave will result from the emerging Augmented/Virtual Reality (AR/VR) industry [2] and more broadly, the Internet-of-Things that will connect an unprecedented number of intelligent devices to next-generation (5 G and beyond) mobile networks as shown in Fig. 1. Existing wireless networks, both cellular and Wi-Fi, must therefore greatly expand their aggregate *network* capacity to meet this challenge. This is being achieved by a combination of approaches including use of multi-input,

multi-output (MIMO) techniques [3], network densification (i.e. deploying small cells [4]) and more efficient traffic management and radio resource allocation.

Since licensed spectrum is a limited and expensive resource, its optimal utilization may require spectrum sharing between multiple network operators/providers of different types -increasingly licensed-unlicensed sharing is being contemplated to enhance network spectral efficiency, beyond the more traditional unlicensed-unlicensed sharing. As the most common unlicensed incumbent, Wi-Fi is now broadly deployed in the unlicensed 5 GHz band in North America where approximately 500 MHz of bandwidth is available. However, these 5 GHz unlicensed bands are also seeing increasing deployment of cellular services such as Long Term Evolution
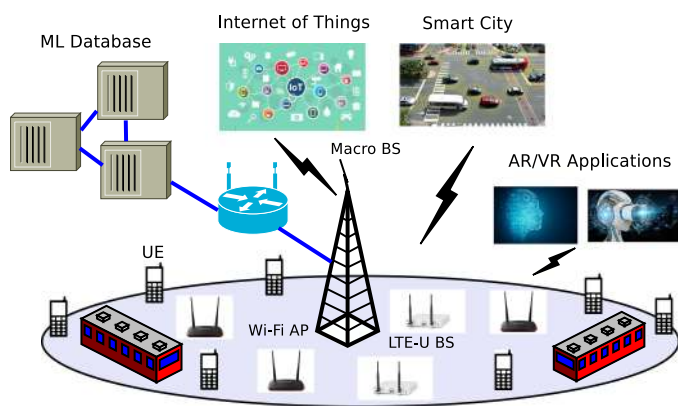
FIGURE 1. Future applications on unlicensed spectrum band.

(LTE) Licensed Assisted Access (LTE-LAA). Recently, the Federal Communications Commission (FCC) sought to open up 1.2 GHz of additional spectrum for unlicensed operation in the 6 GHz band through a Notice of Proposed Rule Making (NPRM) [5]. This allocation of spectrum for unlicensed operation will thus only accelerate the need for further coexistence solutions among heterogeneous systems.

However, the benefits of spectrum sharing are not devoid of challenges, the foremost being the search for effective coexistence solutions between cellular (LTE and 5 G) and Wi-Fi networks whose medium access control (MAC) protocols are very different. While cellular systems employ a Time Division Multiple Access (TDMA)/Frequency Division Multiple Access (FDMA) scheduling mechanism, Wi-Fi depends on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. The 5 GHz band being unlicensed and offering 500 MHz of available bandwidth has prompted several key players in the cellular industry to develop the LTE-LAA specification within the Third Generation Partnership Project (3GPP). Specification differences between LTE and the incumbent Wi-Fi will lead to many issues due to the incompatibility between the two standards. Therefore, to ensure fair coexistence, certain medium access protocols have been developed as an addition to the licensed LTE standard. In addition to LTE-LAA, there also exists LTE-U which was developed by an industry consortium called the LTE-U Forum and will be the main focus of this paper.

LTE-LAA was proposed by 3GPP [6], [7] and its working mechanism is similar to the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol used by Wi-Fi. In LTE-LAA, an LAA base station (BS) acts essentially similar to a Wi-Fi access point (AP) in terms of channel access, i.e., a BS needs to ensure that the channel is free before transmitting any data, otherwise it will perform an exponential back-off procedure similar to CSMA/CA in Wi-Fi. Therefore, there is no need to precisely determine the number of coexisting Wi-Fi APs, due to the channel sensing and back-off mechanism which is adaptable to varying channel occupancy. However, LTE-U which was developed by the LTE-U forum [8], uses a simple duty-cycling technique where the LTE-U BS

will periodically switch between ON and OFF states in an interval set according to the number of Wi-Fi APs present in the channel. In the ON state, the BS transmits data as a normal LTE transmission while in the OFF state, the BS does not transmit any data but passively senses the channel for the presence of Wi-Fi. The number of sensed Wi-Fi APs is then used to properly adjust the duty cycle interval, and this process is known as Carrier Sense Adaptive Transmission (CSAT). Therefore, accurately determining the number of coexisting Wi-Fi APs is important for optimum operation of the CSAT procedure.

Existing literature addresses the LTE-U and Wi-Fi coexistence in terms of optimizing the ON and OFF duty cycle [9], power control [10], hidden node problem [11], etc. On the other hand, the LTE-U specification does not specify, and there has been relatively less work on, how a LTE-U operator should detect the number of Wi-Fi APs on the channel to adjust the duty cycle appropriately. There are a number of candidate techniques to determine the number of Wi-Fi APs as follows:

- **Header-Based CSAT (HD):** Wi-Fi APs transmit beacon packets every 102.4 ms, containing important information about the AP, such as the Basic Service Set Identification (BSSID) which is unique to each AP. This is a straightforward way to identify the Wi-Fi AP, but it adds additional complexity since the LTE-U BS would require a full Wi-Fi decoder to obtain this information from the packet.
- **Energy-Based CSAT (ED):** Rather than a full decoding process, it is hypothesized that sensing the energy level of the channel is enough to detect the number of Wi-Fi APs on the channel. However, it is still a challenging problem since the energy level may not correctly correlate to the number of APs under varying conditions (e.g., different category of traffic, large number of Wi-Fi APs, variations in transmission powers, multipath, etc.).
- **Autocorrelation-Based CSAT (AC):** To detect the Wi-Fi signal at the LTE-U BS, one can develop an auto-correlation (AC) based detector where the LTE-U BS performs auto-correlation on the Wi-Fi preamble, without fully decoding the preamble. This is possible since all Wi-Fi preambles[1] contain the legacy short training field (L-STF) and legacy long training field (L-LTF) symbols which contain multiple repeats of a known sequence. However, the AC function can only determine whether a signal is a Wi-Fi signal and cannot derive any distinct information pertaining to each APs.

Table 1 lists the different types of CSAT approaches with their own pros and cons. We studied energy detection (ED) and AC based detection of Wi-Fi APs in our previous work [12], [13],[2] and proved that our algorithms performed reasonably well under various scenarios.

---

[1]All Wi-Fi frames, even those in newer specifications like 802.11ax, begin with the legacy short training field (L-STF) symbol.

[2]The latest version can be found here: http://bit.ly/2LDVWWo

TABLE 1. Different Types of LTE-U CSAT

| CSAT Types	Method	Pros	Cons
Header De-coding (HD)	Decodes the Wi-Fi MAC header at the LTE-U BS	100% accurate	Additional Complex-ity [15], high cost
Energy De-tection (ED)	Based on the change in the *energy level* of the air medium	Low-cost, low-complexity	Low-accuracy [12]
Auto-correlation (AC)	LTE-U BS per-forms correlation on the Wi-Fi L-STF symbol in the preamble	Low-cost, low-complexity	Medium accuracy (more accurate than ED) [13]
Machine Learning (ML)	Train the model based on energy values on the channel	Much more ac-curate than ED and AC meth-ods	Requires gath-ering data and training models
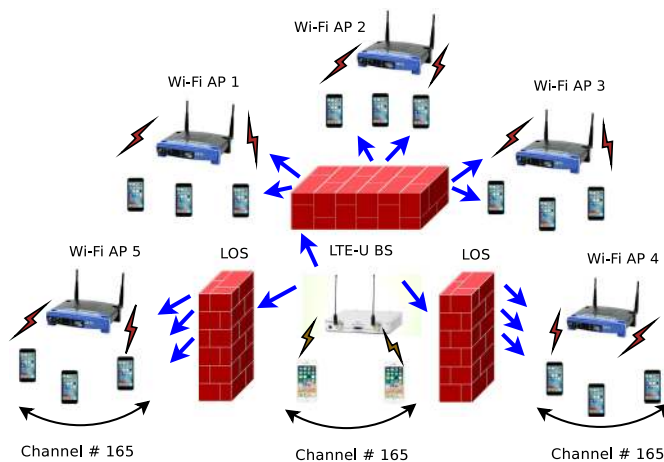
FIGURE 2. Dense LTE Wi-Fi Co-existence deployment setup.

Of late, Machine Learning (ML) approaches are beginning to be used in wireless networks to solve problems such as agile management of network resources using real-time analytics based on data. The advantage of ML is that it has the ability to learn useful information from input data, which can help improve network performance. ML models enable us to replace heuristics with more robust and general alternatives. In this paper, we propose observing the Wi-Fi AP energy values during LTE-U OFF duration and using the data to train different ML models [14]. We also apply the models in an online experiment to detect the number of Wi-Fi APs. Finally, we demonstrate significant improvement in the performance of the ML approach as compared to the ED and AC detectors.

Fig. 2 illustrates an example of a dense LTE-U/Wi-Fi coexistence, where a number of Wi-Fi APs and one LTE-U BS are operating on the same channel, with multiple clients associated with each AP and BS. In such a situation, it is crucial that LTE-U reduce its duty-cycle proportional to the number of Wi-Fi APs, else with a duty-cycle of 50% the Wi-Fi APs will be starved of air-time. As the number of Wi-Fi APs increase on the channel, it becomes increasingly important to detect

the number of APs accurately at the LTE-U BS with out any co-ordination *i.e.,* in a distributed manner. According to the LTE-U forum, it is expected that the LTE-U BS will adjust its duty cycle when one or more Wi-Fi APs turned off, and vice versa. With a large number of Wi-Fi APs, it becomes harder to detect the number accurately using either energy-based or correlation-based approaches. In this work, our goal is to infer the presence of one or more Wi-Fi APs accurately from the collected energy level data using ML algorithms that have been trained on real data. We accomplish this by creating a realistic open lab experimental scenarios using a National Instruments (NI) USRP RIO board with a LTE-U module, five Netgear Wi-Fi APs, and five Wi-Fi clients.

The rest of the paper is organized as follows. Section II presents a brief overview of existing studies on ML as applied to wireless networks and LTE/Wi-Fi coexistence in the unlicensed spectrum. Section III explains the channel access procedure in Wi-Fi using CSMA/CA and the LTE-U duty cycle mechanism. Section IV presents the coexistence system model and the impact of LTE-U and Wi-Fi transmissions on each other. Section VI explains the HD, ED and AC based LTE-U duty cycle adaptation algorithms. Section V describes the experimental set-up used to measure energy values and gather statistics of the energy level in the presence of one or more Wi-Fi APs. Section VII then evaluates various ML algorithms and chooses the most appropriate one for adjusting the duty cycle based on the collected data. Experimental results are presented in Section VIII. Section IX presents the performance (in terms of successful detection, delay and different ML methods) comparison between HD, ED, AC and ML for fixed and different configuration. Finally, Section X concludes the paper with the main contributions and future work in this area.

II. RELATED WORK

In this section, we briefly discuss (a) the existing work on LTE Wi-Fi coexistence without ML, (b) the use of ML in general wireless networks and (c) the application of ML to LTE Wi-Fi coexistence.

A. EXISTING WORK ON LTE AND WI-FI COEXISTENCE

There has been a significant amount of research, from both academia and industry, on the coexistence of LTE and Wi-Fi that discuss several key challenges such as: Wi-Fi client association, interference management, fair coexistence, resource allocation, carrier sensing, etc. Coexistence scenarios are well studied in simulations for both LAA/Wi-Fi and LTE-U/Wi-Fi deployments [15]–[17]. These papers examine coexistence fairness in varying combinations of detection threshold and duty-cycle. However, the auto-correlation based and energy based methods for spectrum sensing in this coexistence context have not been well studied. Recently, we proposed an energy-based CSAT for duty cycle adaptation in LTE-U [12], [18], [19], and studied this approach via rigorous theoretical and experimental analyses. The energy-based CSAT algorithm can infer the number of coexisting Wi-Fi APs by

detecting the energy level in the channel, which is then used to adjust the duty cycle accordingly. Using a threshold of −42 dBm, the algorithm is able to differentiate between one or two Wi-Fi APs, with a successful detection probability P_D of greater than 80% and false positive probability P_{FA} of less than 5%. Hence, this initial work proved the feasibility of stand-alone energy-based detection, without the need for packet decoding. In our succeeding work, we proposed a novel algorithm that utilizes auto-correlation function (AC) [13] to infer the number of active Wi-Fi APs operating in the channel. The AC function is performed on the preamble of a signal to determine if the signal is a Wi-Fi signal. This work further improved the performance of the energy-based approach, with P_D of 0.9 and P_{FA} of less than 0.02, when using an AC threshold N_E of 0.8. In both [12], [13], the maximum number of Wi-Fi APs considered on the channel was two. In realistic dense deployment scenarios, we can expect more than 2 APs on the same channel. Hence, in this paper we study the performance of ED and AC for more realistic dense deployment scenarios.

B. ML AS APPLIED TO WIRELESS NETWORKS

In [20], several state-of-the-art applications of ML in wireless communication and unresolved problems have been described. Resource management in the MAC layer, networking and mobility management in the network layer, and localization in the application layer are some topics that have been identified as being suitable fo ML approaches. Within each of these topics, the authors provide a survey of the diverse ML based approaches that have been proposed. In [21], [22], a comprehensive tutorial has been provided on the use of artificial neural networks-based machine learning for enabling a variety of applications in wireless networks. In particular, the authors presented an overview of a number of key types of neural networks such as recurrent, spiking, and deep neural networks. For each type, the basic architecture as well as the associated challenges and opportunities have been presented, followed by an overview of the variety of wireless communication problems that can be addressed using artificial neural networks (ANNs). This work further investigated many emerging applications including unmanned aerial vehicles, wireless virtual reality, mobile edge caching and computing, Internet of Things, and multi-Random Access Technology (RAT) wireless networks. For each application, the author provided the main motivation for using ANNs along with their associated challenges while also providing a detailed example for a use case scenario.

C. ML AS APPLIED TO LTE WI-FI COEXISTENCE

A learning-based coexistence mechanism for LTE unlicensed based heterogeneous networks (HetNets) was presented in [23]. The motivation was to maximize the normalized throughput of the unlicensed band while guaranteeing the Quality of Service (QoS) of users: the authors thus considered the joint resource allocation and network access problem. A two-level framework was developed to decompose the problem into two subproblems, which were then solved using learning-based approaches. The outcome of the proposed solution achieves near-optimal performance and is more efficient and adaptive due to the distributed and learning-based approach. Authors in [24] provide an overview of earning schemes that enable efficient spectrum sharing using a generic cognitive radio setting as well as LTE and Wi-Fi coexistence scenarios. Most LTE-U duty cycle solutions rely on static coexistence parameter configurations, which may not be applicable in real-life scenarios which are dynamic. Hence in [25], the author uses the Markov decision process modeling along with a solution based on a ML CSAT algorithm which adapts the LTE duty-cycle ratio to the transmitted data rate, with the aim of maximizing the Wi-Fi and LTE-U aggregated throughput. A ML based approach was proposed in [26] for a model-free decision-making implementation of opportunistic coexistence of LTE-U with Wi-Fi, which enabled the LTE-U BS to dynamically identify and further exploit white spaces in the Wi-Fi channel, without requiring detailed knowledge of the Wi-Fi system. By adaptively adjusting the LTE-U duty cycle to Wi-Fi activity, the proposed algorithm enabled maximal utilization of idle resources for LTE-U transmissions, while decreasing the latency imposed on Wi-Fi traffic. The proposed approach also provided a means to control the trade-off between LTE-U utilization and Wi-Fi latency in the coexisting networks.

In [27], the author analytically analyzes the LTE-U scheme when it coexists with Wi-Fi and introduces a ML technique that can be used by an LTE-U network to learn the wireless environment and autonomously select the transmission opportunity (TXOP) and muting period configurations that can provide fair coexistence with other co-located technologies. Simulation results show how ML can assist LTE-U in finding optimal configurations and adapt to changes of the wireless environment thus providing the desired fair coexistence. Authors in [28] propose a convolutional neural network (CNN) that is trained to perform identification of LTE and Wi-Fi transmissions which can also identify the hidden terminal effect caused by multiple LTE transmissions, multiple Wi-Fi transmissions, or concurrent LTE and Wi-Fi transmissions. The designed CNN has been trained and validated using commercial off-the-shelf LTE and Wi-Fi hardware equipment. The experimentation results show that the data representation affects the accuracy of CNN. The obtained information from CNN can be exploited by the LTE-U scheme in order to provide fair coexistence between the two wireless technologies.

The above papers on ML in wireless and unlicensed spectrum do not address the problem of accurately identifying the number of Wi-Fi APs which is a crucial first step in addressing fair coexistence for LTE-U/Wi-Fi coexistence. Hence, in this paper, we modify the classical ML approaches to develop algorithms that can identify the number of Wi-Fi APs on air faster and more reliably than existing methods. Our approach is based on collecting data in realistic coexistence environments for both training and testing. We also compare
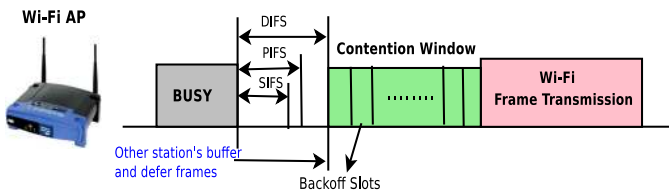
FIGURE 3. Wi-Fi CSMA/CA transmission.

FIGURE 4. LTE-U duty cycle transmission.

the performance of the ML-based approaches with the more conventional ED and AC methods described above.

III. CHANNEL ACCESS PROCEDURE FOR WI-FI AND LTE-U

In this section, we discuss the differences in the channel access procedures for Wi-Fi, using CSMA/CA and LTE-U, using the duty cycle mechanism.

A. WI-FI CSMA/CA

The Wi-Fi MAC distributed coordination function (DCF) employs CSMA/CA as illustrated in Fig. 3. Each node attempting transmission must first ensure that the medium has been idle for a duration of DCF Interframe Spacing (DIFS) using the ED and Carrier Sensing (CS) mechanism. If either ED or CS is true, the Clear Channel Assessment (CCA) is set to be busy. If the channel is idle and the station has not just completed a successful transmission, the station transmits. Otherwise, if the channel is sensed busy during the DIFS sensing period or the station is contending after a successful transmission, the station persists with monitoring the channel until it is measured idle for a DIFS period, then selects a random back-off duration (counted in units of slot time) and counts down. Specifically, a station selects a back-off counter uniformly at random in the range of $[0; 2^i W_0 - 1]$ where the value of i (the back-off stage) is initialized to 0 and W_0 is the minimum contention window chosen initially. Each failed transmission due to packet collision results in incrementing the back-off stage by 1 (binary exponential back-off or BEB) and the node counts down from the selected back-off value; *i.e.,* the node decrements the counter every $\sigma(\mu s)$ corresponding to a back-off slot as long as no other transmissions are detected. If during the countdown a transmission is detected, the counting is paused (freeze the back-off counter), and nodes continue to monitor the busy channel until it goes idle; thereafter the medium must remain idle for a further DIFS period before the back-off countdown is resumed for accessing the channel. Once the counter hits zero, the node transmits a packet. When a transmission has been completed successfully, the value of i is reset to 0. The maximum value of back-off stage i is m with the maximum contention window size of W_m and it stays in m-th stage for one more unsuccessful transmission with the same contention window size W_m, i.e. the retry limit is 1. The value of W_0 and m is determined in the standard. If the last transmission was unsuccessful, the node drops the packet and resets the backoff stage to i = 0. If a unicast transmission is successful the intended receiver will transmit
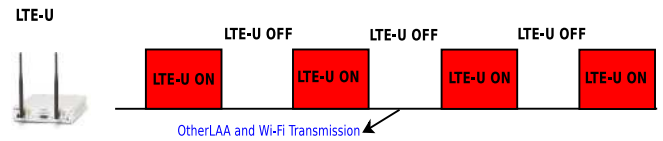
an Acknowledgment frame (ACK) after a Short Interframe Spacing (SIFS) duration post successful reception; the ACK frame structure which consists of preamble and MAC header. The ACK frame chooses the highest basic data rate (6 Mbps, 12 Mbps, or 24 Mbps) for transmitting the MAC header which is smaller than the data rate used for data transmission.

B. LTE-U DUTY CYCLE

LTE-U uses a duty-cycling approach (i.e. alternating the ON and OFF period, where the LTE BS is allowed to transmit only during the ON duration) where the duty cycle (ratio of ON duration to one cycle period) is determined by perceived Wi-Fi usage at the LTE-U BS, using carrier sensing. During the ON period, the LTE-U BS schedules DL transmissions to UEs, unlike Wi-Fi in which transmissions are governed by the CSMA/CA process. Fig. 4 shows the LTE-U transmission for the duty cycle of 0.5. LTE-U uses the basic LTE subframe structure, *i.e.,* the subframe length of 1 ms; each sub-frame consists of two 0.5 ms slots. Each subframe consists of 14 OFDM symbols of which 1 to 3 are Physical Downlink Control Channel (PDCCH) symbols and the rest are Physical Downlink Shared Channel (PDSCH) data. LTE-U BSs start downlink transmissions synchronized with slot boundaries, for (at least) one subframe (2 LTE slots) duration. After transmission, the intended receiver (or receivers) transmits the ACK on the uplink via the licensed band if the decoding is successful.

In LTE, a Resource Block (RB) is the smallest unit of radio resource which can be allocated to a user equipment (UE), equal to 180 kHz bandwidth over a Transmission Time Interval (TTI) of one subframe (1 ms). Each RB of 180 kHz bandwidth contains 12 subcarriers, each with 14 OFDM symbols, equaling 168 Resource Elements (REs). Depending upon the modulation and coding schemes (QPSK, 16-QAM, 64-QAM), each symbol or resource element in the RB carries 2, 4 or 6 bits per symbol, respectively. In the LTE system with 20 MHz bandwidth, there are 100 RBs available.

IV. SYSTEM MODEL AND IMPACT OF LTE-U AND WI-FI ON EACH OTHER

In this section, we describe the coexistence system model assumed in the paper followed by the mutual impact of LTE-U and Wi-Fi on each other.

A. COEXISTENCE SYSTEM MODEL

We assume a deployment where LTE-U and Wi-Fi are operating on the same unlicensed 20 MHz channel in the 5 GHz band. The LTE-U BS transmits only downlink packets on the
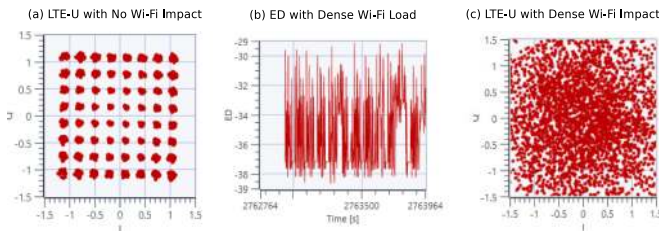
FIGURE 5. Wi-Fi impact on LTE-U on transmission.

FIGURE 6. LTE-U impact on Wi-Fi transmission.

FIGURE 7. LTE-U duty cycle mechanism.

unlicensed spectrum, while all uplink transmissions are on the licensed spectrum. Control and data packets are transmitted using PDCCH and PDSCH respectively. The LTE-U BS operates at maximum transmit power using all possible resource blocks and the highest modulation coding scheme (i.e., 64-QAM). We assume that the Wi-Fi APs also operate at maximum transmission power, transmitting a full buffer video traffic. CSMA/CA and duty-cycle adaptation mechanism are used for channel access for Wi-Fi and LTE-U, respectively. Both Wi-Fi and LTE-U follow their respective retransmission schemes such that when a packet transmission is unsuccessful (packet or acknowledgement lost), the packet will be retransmitted. Finally, we assume that the Wi-Fi APs support both active and passive scanning mode, *i.e.,* both beacon and probe response packets are transmitted by the AP during the association process.

B. IMPACT OF WI-FI ON LTE-U DURING THE ON PERIOD
In order to observe the impact of Wi-Fi on LTE-U during the ON period (*i.e.,* LTE-U is ON without appropriate sensing of a Wi-Fi transmission), we deploy a NI based LTE-U BS (Section V describes the experiment set-up in detail) on channel 165 which is a 20 MHz channel and five Wi-Fi APs on the same channel. Each client is associated with one Wi-Fi AP with full buffer video transmission. Fig. 5(a) shows the constellation of received signals when there is no Wi-Fi AP on the channel, that is, LTE-U BS can transmit the data with high modulation coding scheme of 64-QAM. Similarly, Fig. 5(b) shows the energy value observed when there are 5 Wi-Fi APs on the same channel, where X-axis represents time and Y-axis represents energy values. Fig. 5(c) shows the effect of Wi-Fi transmissions on LTE-U during the ON period, when Wi-Fi APs are unaware of the sudden LTE-U ON cycle starting in the middle of an ongoing Wi-Fi transmission: clearly the constellation is distorted. This clearly points to the inefficient use of the spectrum and the need for the LTE-U BS to sense or learn the medium to identify the number of Wi-Fi APs on the air and scale back its duty cycle accordingly.

C. IMPACT OF LTE-U ON TRANSMISSION ON WI-FI DATA
In case of Wi-Fi/Wi-Fi coexistence where 5 Wi-Fi APs are deployed at the distance of 6F, we observe successful transmission of packets as shown in Fig. 6(a) and (b). We see that the CSMA mechanism works well for Wi-Fi/Wi-Fi coexistence, since the number of packets in error with no LTE-U is
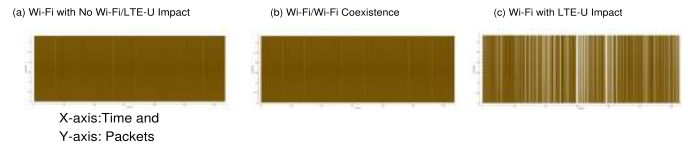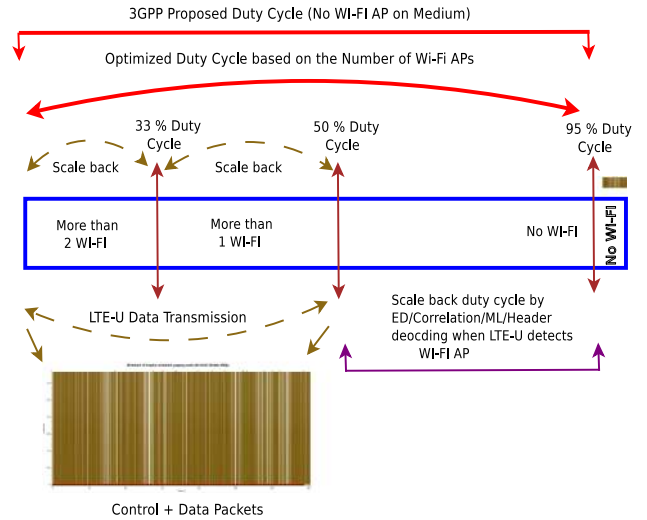
similar that when Wi-Fi coexists with Wi-Fi. Fig. 6(c) shows the packet transmission errors when Wi-Fi coexists with a fixed, LTE-U duty cycle: the number of Wi-Fi packets in error increase. To solve the above problem, the LTE-U forum proposed the dynamic CSAT approach [8], [12], [13] based on the number of Wi-Fi APs on the same channel. Fig. 7 shows the LTE-U duty cycle adaptation process when detecting a varying number of Wi-Fi APs. When no AP is detected on the channel, an LTE-U BS will operate at the maximum 95% duty cycle [8] (*i.e.,* minimum of 1 ms OFF duration). When one AP is detected (assumed using a predetermined sensing technique), the BS will scale back to 50% duty cycle (i.e., 20 ms ON time and 20 ms OFF time). If a new Wi-Fi AP starts transmitting, it will contend with the existing AP only during the OFF time which is 50% of the available medium. Since this is unfair to the Wi-Fi APs, the LTE-U specification recommends scaling the duty cycle back to 33% when more than one Wi-Fi AP is using the channel. However, there is no specific mechanism proposed to detect the number of coexisting Wi-Fi APs in both sparse and dense deployment scenarios.

V. EXPERIMENTAL SETUP FOR MACHINE LEARNING BASED DETECTION
Our experimental set-up consists of one LTE-U BS and a maximum of five Wi-Fi APs. To emulate the LTE-U BS, we use the National Instruments USRP 2953-R software defined radio (SDR) which is equipped with the LTE-U radio framework. There are five Netgear Wi-Fi APs and five Wi-Fi clients deployed in a static configuration. The Wi-Fi clients are combination of laptops and smartphones capable of Wi-Fi

TABLE 2. Experimental Set-Up Parameters

Parameter	Value
Available Spectrum and Frequency	20 MHz and 5.825 GHz
Maximum Tx power for both LTE and Wi-Fi	23 dBm
Wi-Fi sensing protocol	CSMA/CA
Traffic	Full Buffer (Saturation Case)
Wi-Fi & LTE-U Antenna Type	MIMO & SISO
LTE-U data and control channel	PDSCH and PDCCH
Type of Wi-Fi Clients	2 Google Pixel, 1 Samsung, 1 Redmi, and 1 Apple Laptop

FIGURE 8. LTE-U duty cycle adaptation algorithm.

802.11 ac connection. As soon as the client connects to the Wi-Fi AP, it starts a live video streaming application to simulate a full-buffer transmission. The experimental setup is shown in Fig. 9 and the complete experimental parameters are described in Table 2.

We set the BS and APs to be active in the same 20 MHz channel in the 5 GHz band (i.e., Wi-Fi channel 165 and LTE band 46 EARFCN 53540). We separated the APs and BS into six cells, with five cells (Cell A, C, D, E, and F) as Wi-Fi cells and one cell (Cell B) as the LTE-U cell. Each Wi-Fi cell consists of one AP and one client, while the LTE-U BS and UE are contained within the same USRP board.

The BS transmits full buffer data at maximum power by enabling all of its resource blocks with the highest modulation coding scheme (i.e., 64-QAM). It operates at a 50% duty cycle during the experiment, and listens to the configured unlicensed channel during the OFF period for *RF power* and AC measurement. The *RF power* measurement is configured in the LTE block control module of the NI LTE application framework, which outputs energy value as defined in VI-B. The AC function is also configured in the LTE block control module of the same framework and outputs the AC events as defined in VI-C. The energy values observed from Algorithm 2 are given as input to the ML algorithm (explained in detail in Section VII) to classify the number of Wi-Fi APs on the channel. Each Wi-Fi AP transmits full buffer downlink data and beacon frames, with occasional probe responses if it receives probe requests for clients in the vicinity. We also ensure that there is no extra interference in the channel from other Wi-Fi APs.

We measure the energy, AC value and ML (same energy value as input to ML) at the LTE-U BS for the following scenarios:

- **Scenario 0:** No Wi-Fi APs are deployed and only one LTE-U cell (i.e., Cell B) is deployed.
- **Scenario 1:** One Wi-Fi AP (*i.e.,* Cell A) and one LTE-U (i.e., Cell B) is deployed.
- **Scenario 2:** Two Wi-Fi APs (i.e., Cell A & C) and one LTE-U (i.e., Cell B) are deployed.
- **Scenario 3:** Three Wi-Fi APs (i.e., Cell D, E, & F) and one LTE-U (i.e., Cell B) are deployed.
- **Scenario 4:** Four Wi-Fi APs (i.e., Scenario 1: Cell A, Scenario 3: Cell D, E, & F) and one LTE-U (i.e., Cell B) are deployed.
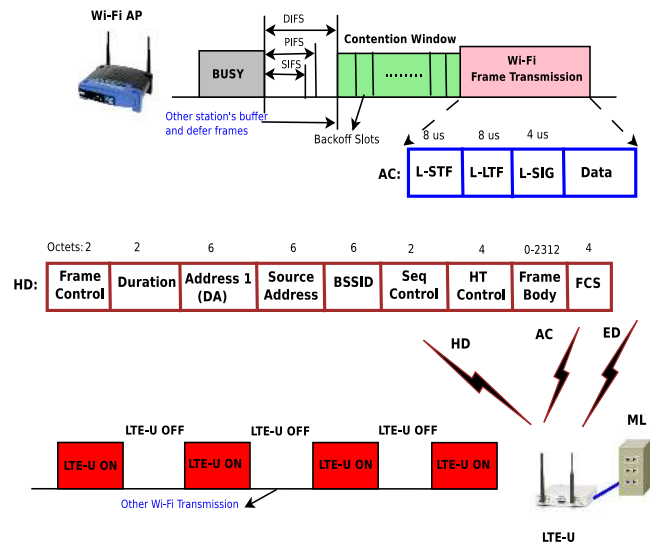
- **Scenario 5:** Five Wi-Fi APs (i.e., Cell A, C, D, E, & F) and LTE-U (i.e., Cell B) are deployed.

In all scenarios, Cell B measures the energy and AC values during the LTE-U OFF period, while the rest of the Wi-Fi cells are transmitting full buffer downlink transmission. We also vary the distances and the LOS and NLOS environment of each cell. In NLOS setup, the wall act as a obstruction between the LTE-U and Wi-Fi APs. We measure the received Wi-Fi AP signals at the LTE-U BS for different 6 feet (For example in Scenario 5, where all the 5 Wi-Fi APs placed at 6 feet from the LTE-U BS), 10 feet and 15 feet distances. Our previous work focused only on detecting Scenarios 1 and 2 (i.e., 1 and 2 Wi-Fi APs coexisting with LTE-U) [12], [13]. Also, we demonstrated that Scenario 0 can be easily distinguished from other scenarios [19].

VI. LTE-U DUTY CYCLE ADAPTATION ALGORITHMS
In order to solve the problems identified in the previous section, we propose header (HD), energy (ED) and autocorrelation (AC) based detection algorithms for a dense deployment scenario to identify the number of Wi-Fi APs on the channel. Fig. 8 explains how different sensing algorithms work based on the known Wi-Fi packet structure.

A. HEADER-DECODING BASED LTE-U DUTY CYCLE ADAPTATION ALGORITHM
We assume that there is either a common preamble [29], [30] between the LTE-U and Wi-Fi systems or the LTE-U BS has a full Wi-Fi decoder that will allow it to decode the Wi-Fi MAC header and hence obtain the BSSID. Doing so, one can accurately detect the number of Wi-Fi APs on the channel and hence header-based decoding is the most accurate method compared to energy, auto-correlation, and ML. However, the decision algorithm to adapt the duty cycle needs to be designed carefully to avoid misclassification. We define a simple algorithm shown in Algorithm 1, to classify the number of

---

Algorithm 1 : Header-decoding based LTE-U Scale Back.

Initialization: (i) $Beacon_i = 0$
(ii) $Count.detect_i = 0$, $Count.falsealarm_i = 0$
(iii) $LastTime = 0$, $TimeSlot = 0.512s$, $Threshold = 4$

---

while *true* **do**
   /* A Wi-Fi beacon with BSSID i is detected at time $CurrentTime$ */
   $Beacon_i$ ++;
   if $CurrentTime - LastTime \geq TimeSlot$ **then**
      $NumberOfAp = 0$;
      for i in $Beacon$ **do**
         if $Beacon_i \geq Threshold$ **then**
            | $NumberOfAp$ ++;
         end
         $Beacon_i = 0$;
      end
      $LastTime = CurrentTime$;

      for $i = 1$ to 5 **do**
         if i Wi-Fi is ON **then**
            if $i == NumberOfAp$ **then**
               | $Count.detect_i$ ++;
            end
            else
               | $Count.falsealarm_i$ ++;
            end
         end
      end
   end
end

Algorithm 2: Energy Based LTE-U Scale Back.

Input: $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$
Initialization: (i) $\alpha_6 = \infty$
(ii) $Count.detect_i = 0$, $Count.falsealarm_i = 0$

---

while *true* **do**
   /* Received $Avg(EnergyLevel)$ over one second */
   for $i = 1$ to 5 **do**
      if i Wi-Fi is ON **then**
         if $\alpha_i \leq Avg(Energy\ Level) \leq \alpha_{i+1}$ **then**
            | $Count.detect_i$ ++;
         else
            | $Count.false.alarm_i$ ++;
         end
      end
   end
end

---

active Wi-Fi APs at each time slot. In brief, the algorithm counts the number of beacon of each uniquely identifiable BSSID, for a defined time slot. Since we can expect that an AP in a real deployment may hop between channels frequently, it is important to collect beacons for a longer period of time rather than deciding based on just one beacon. We initially set a time slot of 10 beacons (i.e., 1.024 s) and count the number of beacons for each BSSID in the time slot. We set a threshold of 9 beacons for an AP to be considered as active, this means that there is 90% confidence that the AP is actually active. The length of the time slot determines the inference delay, hence one would like this delay to be as small as possible. We reduced the time slot to 5 beacons (0.512 s), but to get the same accuracy we need to set the threshold to 4 beacons which means that the confidence rate is at a lower 80%. Thus, with a slightly lower confidence rate, we can reduce the inference time to half without compromising the detection accuracy.

B. ENERGY BASED LTE-U DUTY CYCLE ADAPTATION ALGORITHM

The experiment setup is shown in Fig. 9. We measure the received energy at the LTE-U BS for different distances between the LTE-U BS and Wi-fi APs and obtain histograms of the measured signal when one or more Wi-Fi APs are transmitting at 6, 10 and 15 feet from the LTE-U BS. We then fit the measured histograms to probability distribution functions as described in [12] to develop a classification algorithm. In Algorithm 2, an energy-based detection listens to the energy level in the channel and according to a set threshold [12], decides whether to scale back the duty cycle or not. Since the measured energy threshold depends on the the number of detected Wi-Fi APs, the choice of threshold is important to the algorithm. Finally, we implement the algorithm in the LTE-U BS NI hardware and validate it experimentally.

First, we modify the NI LTE application framework to measure *RF power* during the LTE-U OFF period. The collected energy values are then averaged over one second time duration and used for algorithm input. If the averaged energy value is greater than the specified threshold α_1, i.e., if energy value $\geq \alpha_1$ then there is a possibility of Wi-Fi packets (beacon, probe request, probe response, data, or ACK) transmitted in the channel. The BS then can declare whether one, two, three, four, or five AP is present, based on the other thresholds: α_2, α_3, α_4, α_5 (e.g., if $\alpha_3 \leq$ energy value $\leq \alpha_4$ then there are 4 APs in the channel). By keeping count of correct and incorrect decisions made by the algorithm, we calculate the probability of correct detection and false positive on predicting the number of Wi-Fi APs in the unlicensed spectrum. These probability values are used as a metric to determine the performance of the threshold, such that we pick a set of threshold with high probability of correct detection and low probability of false positive.

C. AC BASED LTE-U DUTY CYCLE ADAPTATION ALGORITHM

In the same experiment setup as shown in Fig. 9, we count the total number of AC events that are above a threshold for every one second over the duration of 90 seconds. We measure the total number of events above the AC threshold at the
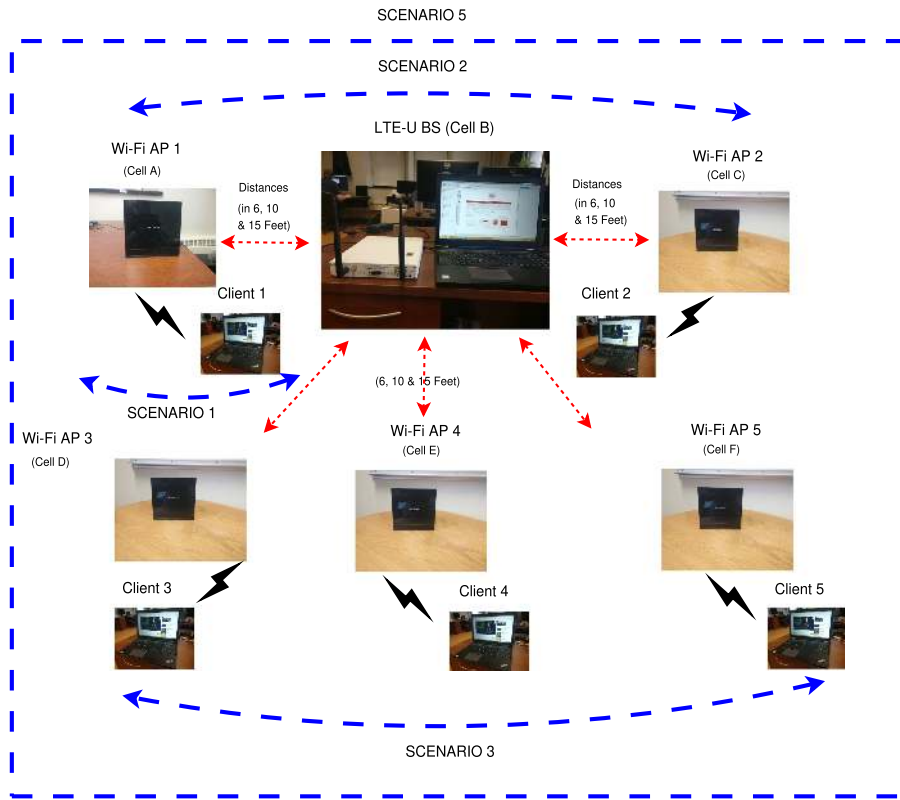
FIGURE 9. LTE Wi-Fi Co-existence experimental setup.

LTE-U BS for 6, 10 and 15 feet distances. Then, we observe the PDF distribution of the number of AC events above the threshold [13] for Scenario 0 to 5 described above. We make use of this key observation to develop a classification algorithm (*i.e.,* Algorithm 3) for both LOS and NLOS scenarios. The algorithm uses AC functions and optimal thresholds to determine the number of Wi-Fi APs in the channel, therefore the selection of threshold is also important and will be shown in this section. We implement the algorithm in the LTE-U BS hardware and validate it experimentally. The AC function is performed at LTE-U BS to sense the spectrum for Wi-Fi preamble signals (*i.e.,* L-STF). The output of the function is an AC value which determine the likelihood that the signal is a Wi-Fi preamble. We observed on many experiments, that the threshold th_ρ of 0.25 is sufficient to determine that the captured signal is a Wi-Fi signal (beacon, probe request, probe response, data, or ACK). Using the threshold, we predicted the number of Wi-Fi signals in every one second period. Next, we calculate the ratio [13] and then compared to R_i which is a threshold determined during a preliminary experiment with i Wi-Fi AP and no LTE-U on the channel. The R_i is determined such that the true positive rate is as high as possible and false positive rate is as low as possible during the preliminary experiment. Since it is not possible for the observed ratio to be higher than R_i, we set a correct prediction that i Wi-Fi AP is present in the channel if the ratio is less than or equal to the threshold R_i, and false prediction otherwise.

VII. ML ALGORITHMS FOR LTE-U DUTY CYCLE ADAPTION

ML models enable us to replace heuristics with more robust and general alternatives. For the problem of distinguishing between different numbers of Wi-Fi APs, we train a model to detect a pattern in the signals instead of finding a specific energy threshold in a heuristic manner. The state-of-the-art ML models leverage the unprecedented performance of neural network models that are able to surpass human performance on many tasks, for example, image recognition [31], and help us answer complex queries on videos [32]. This efficiency is a result of large amounts of data that can be collected and labeled as well as usage of highly parallel hardware such as GPUs or TPUs [33], [34]. In the work described in this paper, we train our neural network models on NVidia GPUs and collect enough data samples that enable our models to achieve high accuracy. Our major task is a classification problem to distinguish between zero, one, two, three, four, or five Wi-Fi BSSs.

We consider machine learning models that take time-series data of width w as input, giving an example space of $\mathcal{X} \in \mathcal{R}^w$, where $\mathcal{R}$ denotes the real numbers. Our discrete label space of k classes is represented as $\mathcal{Y} \in \{0, 1\}^k$. For example, $k = 3$ classes, enables us to distinguish between 0, 1, and 2 Wi-Fi APs. Machine learning models represent parametrized functions (by a weight vector θ) between the example and label spaces $f(x; \theta) : \mathcal{X} \mapsto \mathcal{Y}$. The weight vector θ is iteratively updated during the training process until the convergence of

---

Algorithm 3 : Auto-correlation Based LTE-U Scale Back.

Input: th_ρ, R
Initialization: $Count.detect_i = 0$, $Count.falsealarm_i = 0$

---

while *true* **do**
    /* Received T number of AC values over one second */
    for $i = 1$ *to* 5 **do**
        if i *Wi-Fi is ON* **then**
            $Signal = 0$;
            for $t = 1$ *to* T **do**
                if $AC_t \geq th_\rho$ **then**
                    $Signal$ ++;
                end
            end
            $ratio = \frac{Signal}{T}$;

            if $ratio \leq R_i$ **then**
                $Count.detect_i$ ++;
            else
                $Count.falsealarm_i$ ++;
            end
        end
    end
end

the training accuracy or training loss (usually determined by very small changes to the values despite further training), and then the final state of θ is used for testing and real-time inference.

A. DATA PREPARATION

The training and testing data is collected over an extended period of time with a single scenario taking about 8 hours. For ease of exposition, we consider the case with one and two Wi-Fi APs. We collect data for each Wi-Fi AP independently and store the two datasets in separate files. Each file contains more than 2.5 million values and the total raw data size in CSV format is of about 60 MB. Each file is treated as time-series data with a sequence of values that are first divided into chunks. We overlap the time-series chunks arbitrarily by three-fourths of their widths w. For example, for chunks width $w = 128$, the first chunk starts at index 0, the second chunk is formed starting from index 32, the third chunk starts at index 64, and so on. This is part of our data augmentation and a soft guarantee that much fewer patterns are broken on the boundary of chunks. The width w of the (time-series data) chunk acts as a parameter for our ML model. It denotes the number of samples that have to be provided to the model to perform the classification. The longer the time-series width w, the more data samples have to be collected during inference. The result is higher latency of the system, however, the more samples are gathered, the more accurate the predictions of the model. On the other hand, with smaller number of samples per chunk, the time to collect the samples is shorter, the inference

is faster but of lower accuracy. We elaborate more on this topic in Section VIII.

The collection of chunks are shuffled randomly. We divide the input data into training and test sets, each 50% of the overall data size. The aforementioned shuffling ensures that we evenly distribute different types of patterns through the training and test sets so that the classification accuracy of both sets is comparable. Each of the training and test sets contain roughly the same number of chunks that represent one or two Wi-Fi APs. We enumerate classes from 0. For the case of 2 classes (either one or two Wi-Fis), we denote by 0 the class that represents a single Wi-Fi AP and by 1 the class that represents 2 Wi-Fi APs. Next, we compute the mean μ and standard deviation σ only on the training set. We check for outliers and replace the values that are larger than 4σ with the μ value (e.g., there are only 4 such values in class 1).

The data for the two classes have different ranges (from about -45.46 to -26.93 dBm for class 0, and from about -52.02 to about -22.28 dBm for class 1). Thus, we normalize the data D in the standard way: $ND = \frac{(D-\mu)}{\sigma}$, where ND is the normalized data output, μ and σ are the mean and standard deviation computed on the training data. We attach the appropriate label to each chunk of the data. The overall size of the data after the preparation to detect one or two Wi-Fi APs is about 382 MB, where the Wi-Fi APs are on opposite sides of the LTE-U BSS and placed at 6 feet distance from the LTE-U BSS). We collect data for many more scenarios and present them in Section VIII. The final size of the collected data is 3.4 GB.

For training, we do not insert values from different numbers of Wi-Fi APs into a single chunk. The received signal in the LTE-U BSS has higher energy on average for more Wi-Fi APs, thus there are differences in the mean values for each dataset. Our data preparation script handles many possible numbers of Wi-Fi APs and generates the data in the format that can be used for model training and inference (we follow the format for datasets from the UCR archive). In the future, we plan on gathering additional data samples for more Wi-Fi APs and making the dataset more challenging for classification.

B. NEURAL NETWORK MODELS: FC, VGG AND FCN

Our data is treated as a uni-variate time-series for each chunk. There are many different models proposed for the standard time-series benchmark [35].

First, we test *fully connected (FC)* neural networks. For simple architectures with two linear layers followed by the ReLU non-linearity the maximum accuracy achieved is about 90%. More linear layers, or using other non-linearities (e.g. sigmoid) and weight decays do not help to increase the accuracy of the model significantly. Thus, next we extract more patterns from the data using the convolutional layers.

Second, we adapt the *VGG* network [36] to the one dimensional classification task. We changed the number of weight layers to 6 (we also tested 7, 5, and 4 layers, but found that 6 gives the highest test accuracy of about 99.52%). However,

the drawback is that with fewer convolutional layers, the fully connected layers at the end of *VGG* net become bigger to the point that it hurts the performance (for 4 weight layers it drops to about 95.75%). This architecture gives us higher accuracy but is rather difficult to adjust to small data.[3]

Finally, one of the strongest and flexible models called *FCN* is based on convolutional neural networks that find general patterns in time-series sequences [37]. The advantages of the model are: simplicity (no data-specific hyper-parameters), no additional data pre-processing required, no feature crafting required, and significant academic and industrial effort into improving the accuracy of convolutional neural networks [38], [39].

The architecture of the FCN network contains three blocks, where each of them consists of a convolutional layer, followed by batch normalization $f(x) = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}$ (where ϵ is a small constant added for numerical stability) and ReLU activation function $y(x) = \max(0, x)$. There are 128, 256, and 128 filter banks in each of the consecutive 3 layer blocks, where the sizes of the filters are: 8, 5, and 3, respectively. We follow the standard convention for Convolutional Neural Networks (CNNs) and refer to the discrete cross-correlation operation as convolution. The input x to the first convolution is the time-series data chunk with a single channel c. After its convolution with f filters, with filters denoted as y, the output feature map o has f channels. For training, we insert $s = 32$ time-series data chunks into a mini-batch. We have $j \in f$ and the discrete convolution [40] that can be expressed as:

$$o = x * y \qquad (1)$$

and in the Einstein notation:

$$o_{(s,j)} = \sum_{i \in c} x_{(s,i)} \cdot y_{(j,i)} \qquad (2)$$

C. ML MODELS FROM SCIKIT-LEARN

To diversify the machine learning models used in our comparison, we select the most popular models from the scikit-learn (also denoted as *sklearn*) library.[4] The library exposes classical machine learning algorithms implemented in Python. This is a common tool used for science and engineering. We run our experiments using *sklearn* version 0.19.1 with Python 3.6. We analyze how the following models perform on our WiFi data and report their test accuracy. The decision tree is a simple classifier that learns decision rules inferred from the data features. The deeper the tree, the more complex the decision rules and the fitter the model. The decision tree classifier achieves accuracy of 79.46% for the task of distinguishing between one or two Wi-Fi APs. The AdaBoost [41] classifier is one of the best out-of-the-box models in the *sklearn* library that creates an ensemble of classifiers. In our experiments, AdaBoost begins by fitting a decision tree classifier on the

original dataset and then fits additional decision tree classifiers on the same dataset but where the weights of incorrectly classified instances are modified such that subsequent classifiers focus more on difficult cases. It is tuned by adjusting the maximum number of the decision tree classifiers used. AdaBoost achieves accuracy of 94.57%. Random Forest is an averaging algorithm based on randomized decision trees. Its test accuracy is 79.87%. We find that the best tested model from the *sklearn* library is AdaBoost. The highest test accuracy achieved for AdaBoost for the standard case with two Wi-Fi APs is worse by about 5% when compared to the overall best FCN model (described in Section VII-B), which achieves accuracy of 99.38% for the same configuration (with 2 Wi-Fi APs, 512 chunk size, NLOS, and 6 feet distance). For more than 5 classes, Random Forest model achieves higher accuracy than AdaBoost.

D. TIME-SERIES SPECIFIC MODELS

BOSS in Vector Space (BOSS VS) model [42] is a time-series classification algorithm, whose properties make it suitable for our task. This algorithm is characterized by fast inference, tolerance to noise that enable us to achieve high test accuracy, moderate training time, which allows for periodic model updates. Moreover, BOSS VS achieves best test accuracy for repetitive and long time-series data. Within the time-series specific models, we also compared to WEASEL [43], which yielded lower test accuracy despite much longer training time.

We run the BOSS VS time-series specific model for the NLOS 6 feet case. Other time-series models train much longer (in the order of days) on our large (a few GBs) time-series data or do not fit even into 128 GB of RAM memory provided. We observe that from 2 to up to 4 WiFi APs, the performance of the BOSS VS model is on-par with the performance of FCN model. However, for the scenario where we have to distinguish between 0 to 5 WiFi APs, the accuracy of the FCN model is higher by about 7%. One concern with the BOSS VS model is that we have to use a machine with 128 GB of RAM to train the model and for larger data sizes, the out of memory exception is thrown as well (the model is implemented in Java). For the FCN, we are able to scale to arbitrary amount of data. Based on the thorough experimental analysis, we see the FCN model and other neural network based models as the most accurate and scalable models that can be used to predict the number of Wi-Fi APs.

E. FFT COMPRESSION

We use the FFT-based convolution with compression proposed in [38] and here describe its essential component. We express input x and filter y as discrete functions that map tensor index positions n to values $x[n]$. Their corresponding Fourier representation re-indexes tensors in the spectral domain:

$$F_x[\omega] = F(x[n]) \quad F_y[\omega] = F(y[n])$$

---

[3]The dimensionality of the data is reduced slowly because of the small filter of size 3.

[4]https://scikit-learn.org/stable/index.html

This mapping is invertible $x = F^{-1}(F(x))$. Convolutions in the spectral domain correspond to element-wise multiplications:

$$x * y = F^{-1}(F_x[\omega] \cdot F_y[\omega])$$

For natural data, such as time-series data, a substantial portion of the high-frequency domain is close to 0. This observation allows us to compress the data.

Let $M_c[\omega]$ be a discrete indicator function defined as:

$$M_c[\omega] = \begin{cases} 1, & \omega \leq c \\ 0, & \omega > c \end{cases}$$

$M_c[\omega]$ is a mask that limits the input data and filters to a certain *band* of frequencies. The FFT-based convolution with compression is defined as follows:

$$x *_c y = F^{-1}\{(F_x[\omega] \cdot M_c[\omega]) \cdot (F_y[\omega] \cdot M_c[\omega])\} \quad (3)$$

The mask $M_c[\omega]$ is applied to both the signal $F_x[\omega]$ and filter $F_y[\omega]$ (in equation 3) to indicate the compression of both arguments.

VIII. EXPERIMENTAL RESULTS

In this section we discuss the model training, inference and transition between different classes. The code for our project can be found on github: http://bit.ly/2Ob5kAr.

A. TRAINING AND INFERENCE

Each model is trained for at least 100 epochs. We experiment with different gradient descent optimization algorithms, e.g. Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam).[5] For the SGD algorithm, we grid search for the best initial learning rate and primarily use 0.0001. The learning rate is reduced on plateau by 2X after 50 consecutive iterations (scheduled patience). SGD is used with momentum value 0.9. We use standard parameters for the Adam optimization algorithm. The batch size is set to $s = 32$ to provide high statistical efficiency. The weight decay is set to 0.0001. For our neural network models, the dataset is relatively simple. The Wi-Fi data can be compared in its size and complexity to the MNIST dataset [44] or to the GunPoint series from the UCR archive [35].

B. TIME-SERIES WIDTH

The number of samples collected per second by the LTE-U BS is about 192. The inference of a neural network is executed in milliseconds and can be further optimized by compressing the network. The final width of the time-series chunk imposes a major bottleneck in terms of the system latency. The smaller the time-series chunk width w, the lower the latency of the system. However, the neural network has to remain highly accurate despite the small amount of data provided for its inference. Thus, we train many models and systematically vary the chunk width w from 1 to 2048 (see Fig. 10). In
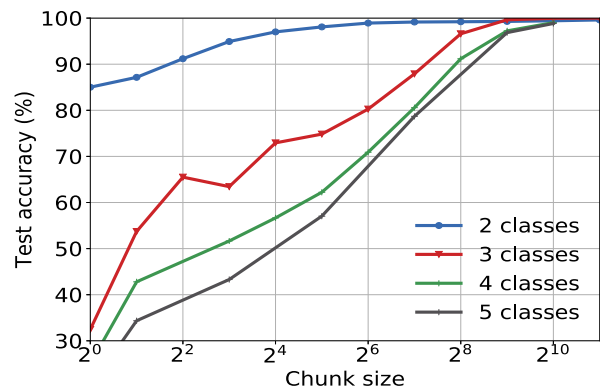

FIGURE 10. The test accuracy (%) for a model trained and tested for a given chunk size (ranging from 1 to 2048) to distinguish between 2 classes (either 1 or 2 Wi-Fi APs), 3 classes (distinguish between 0, 1, or 2 Wi-Fi APs), 4 classes (distinguish between 0, 1, 2, or 3 Wi-Fi APs), and 5 classes (distinguish between 0, 1, 2, 3 or 4 Wi-Fi APs).
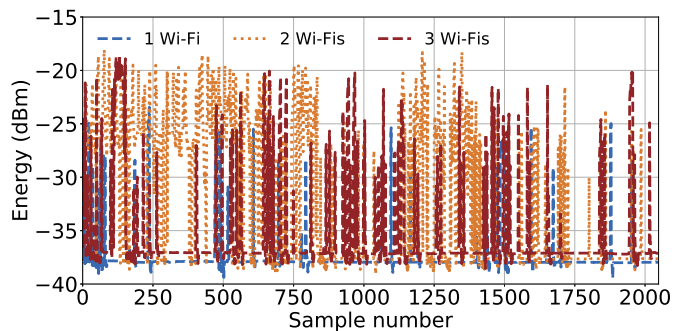

FIGURE 11. Number of Wi-Fi APs. The values of the energy (in dBm) captured for 2048 samples in LTE-U BS while there are 1 Wi-Fi, 2, and 3 Wi-Fis scenarios at 6 Feet, NLOS. The more Wi-Fi APs active, the more energy picks we observe.

this case, each model is trained only for the single scenario (placement of the Wi-Fi APs) and with zero, one, two, or three active Wi-Fi APs. When we decrease the chunk sizes to the smaller chunk consisting of a single sample, the test accuracy deteriorates steadily down to the random choice out of the 3 or 4 classes (accuracy of about 33% and 25%, respectively) and for the 2 classes, its performance is very close to the ED (Energy-based Detection) method.

We present the energy of the signals captured in different configurations: (1) Fig. 11 shows the values of energy captured for different number of Wi-Fi APs (one, two and three), (2) Fig. 12 demonstrates the scenario with different distances of Wi-Fi APs from the LTE-U, and (3) Fig. 13 gives insight into energy of the signal in NLOS and LOS scenarios.

We consider in detail the signal from about 1500th sample to 2000th sample in Fig. 11. It is challenging to distinguish between two or three Wi-Fis.[6] The visual inspection of the signals suggests that width of the time-series chunk

---

[5]A very good explanation can be found here: http://bit.ly/2Y9XaQ8

[6]The Energy values for 4 and 5 Wi-Fi APs are more dense and challenging. In order to better visualize we plotted only 1, 2 and 3 Wi-Fi APs
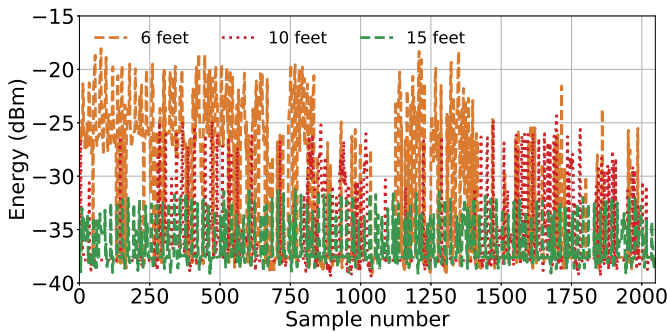
FIGURE 12. Distances from LTE-U. The values of the energy (in dBm) captured for 2048 samples in LTE-U BS while there are 2 Wi-Fi APs at 6, 10, and 15 Feet, NLOS. The closer the Wi-Fi APs are to the LTE-U, the higher energy is captured.
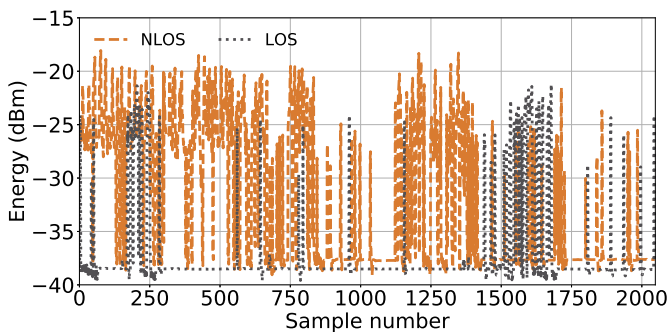

FIGURE 13. NLOS vs LOS. The values of the energy (in dBm) captured for 2048 samples in LTE-U BS while there are 2 Wi-Fi APs at 6 Feet, in NLOS and LOS scenarios. The fewer obstructions, the higher energy is captured.

should be longer than 500 samples. Signals with width of 384 achieve test accuracy below 99% and signals with width 512 can be trained to obtain 99.68% of test accuracy. Based on the experiments in Figs. 10 and 11, we find that the best trade-off between accuracy and inference time is achieved for chunk of size 512.

C. TRANSITIONS BETWEEN CLASSES
When we switch to another class (change the state of the system in terms of the number of Wi-Fis), we account for the transition period. If in a given window of 1 second a new Wi-Fi is added, the samples from this first second with new Wi-Fi (or without one of the existing Wi-Fis - when it is removed), the chunk is containing values from n and $n + 1$ (or $n - 1$) number of Wi-Fis. An easy workaround for the *contaminated* chunk is to change the state of the system to new number of Wi-Fis only after the same class is returned by the model in two consecutive inferences (classifications).

D. REAL-TIME INFERENCE
We deploy the model in real-time, which is similar to the energy data collection experiment setup, and is shown in Fig. 14. We prepare the model only for the inference task in the following steps. Python scripts load and deploy the
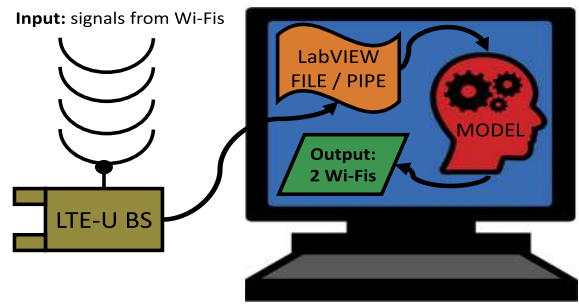

FIGURE 14. The schema of the inference process, where the input received by the LTE-U BS is signals from Wi-Fis and the output is the predicted number of Wi-Fis.

trained PyTorch model. We set up the Wi-Fi devices and generate some network load for each device. The LTE-U BS is connected to a computer with the hardware requirements of at least 8 GB RAM (Installed Memory), 64-bit operating system, x64-based processor, Intel(R) Core i7, CPU clock 2.60 GHz. The energy of the Wi-Fi transmission signal in a given moment in time is capture using NI LabVIEW. From the program, we generate an output file or write the data to a pipe. The ML model reads the new values from the file until it reaches the time-series chunk length. Next, the chunk is normalized and passed through the model that gives a categorical output that indicates the predicted number of Wi-Fis in the real-time environment.

IX. PERFORMANCE COMPARISON BETWEEN HD, ED, AC AND ML METHODS
We analyze and study the performance differences between HD, ED, AC and ML methods for different configuration setups and discus the inference delay. In ML method, we validate the performance on ML real-time inference data. For the final evaluation, we train a single Machine Learning model that is based on the FCN network and used for all the following experiments. The model is trained on the whole dataset of size 3.4 GB, where the train and test sets are of the same size of about 1.7 GB.

A. COMPARISON BETWEEN ML METHODS
We present comparison between ML methods in Fig. 15. The time-series specific neural network models, such as FCN (VII-B) as well as BOSS VS (VII-D), perform much better than the general purpose models from scikit-learn library (described in Section VII-C). The middle-ground between the two options is a simple two-layer convolutional network called *LeNet*. The main benefit of using FCN (MEDIUM) or BOSS VS is greater model learning capacity than *LeNet* or scikit-learn models. There is a negligible difference in terms of test accuracy between the FCN and BOSS VS models. However, the FCN models can scale to much bigger data sizes and we observe that the BOSS VS model often goes out of memory for more than a few GBs of input data. Thus, we select FCN as our main Machine Learning (ML) model for all the remaining experiments.
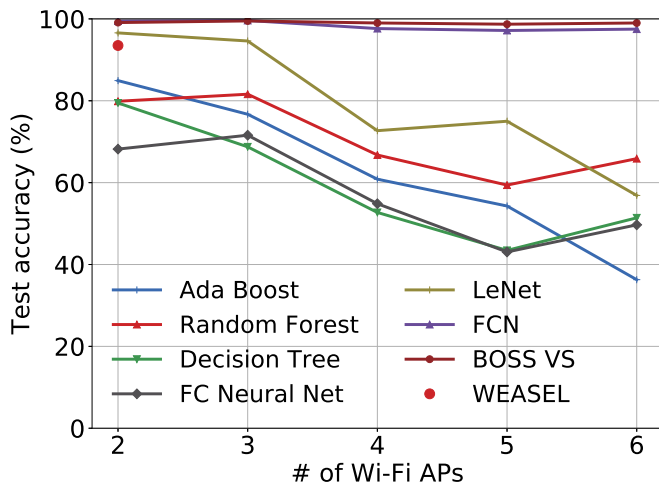
FIGURE 15. Comparison of test accuracy for different ML methods. Number of Wi-Fi APs equals to 2 denotes the Case D configuration (NLOS, 6 feet). Thus, 2 on the x axis corresponds to distinguishing between 1 and 2 Wi-Fi APs, whereas 3 denotes distinguishing between 0, 1, or 2 Wi-Fi APs. Similarly, the values on the x axis (4,5) denote distinguishing from 0 to (x-1) WiFi APs.
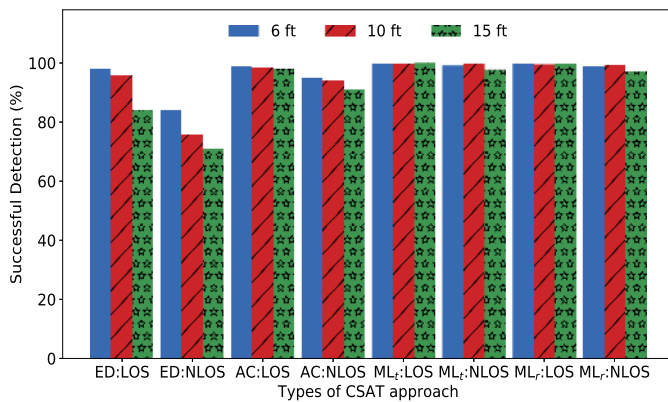

FIGURE 16. Comparison of results for successful detection between ED, AC and ML methods. ML results are presented for the test data (denoted as ML_t:) and for the real time inference (denoted as ML_r:).

B. SUCCESSFUL DETECTION AT FIXED DISTANCE

We compare the ML performance with HD,[7] ED and AC approaches using the NI USRP platform as shown in Fig. 9. Similarly we compare the performance of HD by analyzing the Wi-Fi BSSID through wireshark capture. In the experiment, Wi-Fi APs are transmitting full buffer data, along with beacon and probe response frames following the 802.11 CSMA specification. We performed different experiments with 6 ft, 10 ft and 15 ft for LOS and NLOS scenarios. Fig. 16 shows the performance of detection for LOS and NLOS scenarios. In ED and AC based approach the proposed detection algorithm achieves the successful detection on average at 93% and 95% for LOS scenario. Similarly, the algorithm achieves 80% and 90% for the NLOS scenario. In this work, we show that ML approach can achieve close to 100% successful detection rate

---

[7]The successful Wi-Fi detection in HD for LOS and NLOS scenario is 100%. Hence we have not included in the Fig. 9.

for both LOS and NLOS, and different distance scenarios (6 ft, 10 ft & 15 ft). We observe the ML approach works close to the performance of HD.

Table 3 shows the performance of detection for fixed distance configuration setup. From, this table the number of Wi-Fis columns represents the number of Wi-Fi APs deployed in the coexistence setup. The number of Wi-Fi AP 2 corresponds to distinguishing between 1 and 2 Wi-Fi APs, whereas 3 denotes distinguishing between 0, 1, or 2 Wi-Fi APs and so on. In all cases the performance of ML is close to 100%.

C. SUCCESSFUL DETECTION AT DIFFERENT CONFIGURATIONS

We verify how the detection works in different configurations. We placed more than two Wi-Fi APs on the same side of the LTE-U BS, unlike the above configuration (i.e., 6 ft, 10 ft and 15 ft) where they were on opposite sides. Wi-Fi AP 1, Wi-Fi AP 2, Wi-Fi AP 3, Wi-Fi AP 4 and Wi-Fi AP 5 are placed at distances of 6 feet, 10 feet and 15 feet from the LTE-U BS respectively. We measured the performance of detection with LOS and NLOS configurations. The goal in this section is to observe the performance of detection in the ML compared with HD, ED, and AC. Some of the possible cases are listed below.

- **Case A:** Only the Wi-Fi AP 1 at 6 feet is ON.
- **Case B:** Only the Wi-Fi AP 2 at 10 feet is ON.
- **Case C:** Only the Wi-Fi AP 3 at 15 feet is ON.
- **Case D:** Wi-Fi AP 1 at 6 feet is ON and Wi-Fi AP 2 at 6 feet is ON.
- **Case E:** The Wi-Fi AP 1 at 6 feet and Wi-Fi AP 3 at 15 feet is ON.
- **Case F:** The Wi-Fi AP 1 at 10 feet and Wi-Fi AP 3 at 15 feet is ON.
- **Case G:** Wi-Fi AP 1 and Wi-Fi AP 2 at 6 feet is ON and Wi-Fi AP 3 at 15 feet is ON.
- **Case H:** Wi-Fi AP 1 at 6 feet is ON, Wi-Fi AP 2 at 10 feet is ON, and Wi-Fi AP 3 at 15 feet is ON.
- **Case I:** Wi-Fi AP 1 and Wi-Fi AP 2 at 6 feet is ON, Wi-Fi AP 3 at 10 feet is ON and Wi-Fi AP 4 at 15 feet is ON.
- **Case J:** Wi-Fi AP 1 at 6 feet is ON, Wi-Fi AP 2 and Wi-Fi AP 3 at 10 feet is ON and Wi-Fi AP 4 at 15 feet is ON.
- **Case K:** Wi-Fi AP 1 and Wi-Fi AP 2 at 6 feet is ON, Wi-Fi AP 3 and Wi-Fi AP 4 at 10 feet is ON and Wi-Fi AP 5 at 15 feet is ON.
- **Case L:** Wi-Fi AP 1 at 6 feet is ON, Wi-Fi AP 2, Wi-Fi AP 3 and Wi-Fi AP 4 at 10 feet is ON and Wi-Fi AP 5 at 15 feet is ON.
- **Case M:** Wi-Fi AP 1 at 6 feet is ON, Wi-Fi AP 2 at 10 feet is ON, Wi-Fi AP 3, Wi-Fi AP 4 and Wi-Fi AP 5 at 15 feet is ON.
- **Case N:** Wi-Fi AP 1 and Wi-Fi AP 2 at 6 feet is ON, Wi-Fi AP 3 at 10 feet is ON, Wi-Fi AP 4 and Wi-Fi AP 5 at 15 feet is ON.

TABLE 3. Performance of Detection for Fixed Distance Configuration Setup

Configuration Distance	Classes # of Wi-Fis	HD (%)		ED (%)		AC (%)		ML (%)	
		LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS
6F	2	100	100	96	91	98	96	98.60	99.10
	3	100	100	88	85	95	90	99.10	99.50
	4	100	100	80	74	87	81	99.40	99.00
	5	100	100	74	62	76	65	99.20	98.70
	6	100	100	62	51	70	59	99.30	99.0
10F	2	100	100	94	89	97	94	99.80	99.98
	3	100	100	86	82	91	88	99.80	99.98
	4	100	100	78	72	85	79	99.80	99.90
	5	100	100	72	60	75	63	99.50	99.85
	6	100	100	64	54	68	57	99.80	99.84
15F	2	100	100	92	87	95	90	99.80	99.80
	3	100	100	84	80	85	81	99.90	99.60
	4	100	100	75	70	79	71	99.90	99.60
	5	100	100	70	58	71	64	99.60	99.50
	6	100	100	63	53	66	55	99.50	99.40

TABLE 4. Performance of Detection for Different Configuration Setup (From Case A to G)

CSAT Types	CASE A (%)		CASE B (%)		CASE C (%)		CASE D (%)		CASE E (%)		CASE F (%)		CASE G (%)	
	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS
HD	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ED	91	82	90	79	85	78	82	77	80	74	81	72	80	69
AC	95	91	94	91	92	90	91	90	88	85	88	83	86	77
ML	98.80	97.96	99.94	99.37	99.96	97.74	99.46	97.80	99.21	99.14	99.32	99.10	99.56	98.44

TABLE 5. Performance of Detection for Different Configuration Setup (From Case H to N)

CSAT Types	CASE H (%)		CASE I (%)		CASE J (%)		CASE K (%)		CASE L (%)		CASE M (%)		CASE N (%)	
	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS
HD	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ED	80	69	77	67	76	65	68	61	67	56	68	57	66	52
AC	84	74	79	68	77	67	75	65	74	64	72	63	71	59
ML	98.70	97.36	99.24	98.26	99.76	98.24	98.83	98.06	99.11	99.04	99.02	98.05	99.96	97.74

The different configurations are for LTE-U when it coexists with different number of Wi-Fi APs (from 1 to 5). Table 4 shows the better performance for ED and AC compared to the Table 5. This is due to fewer number of Wi-Fi AP deployments from Case A to G compared to Case H to N. Hence, the the ED and AC methods can detect the number of Wi-Fi APs close to 80% for ED and up to 90% for AC. As the number of Wi-Fi APs increases from 3 to 5 Wi-Fi APs (*i.e.,* Case H to N), we observe substantial degradation in ED performance (to 56%) and AC performance (to 63%). Tables 4 and 5 show that there is no such degradation in the performance of ML as compared to ED and AC. Hence, we believe that the ML approach is the preferred method for a LTE-U BS in a dense environment to detect the number of Wi-Fi APs and scale back the duty cycle efficiently.

D. ADDITIONAL DELAY TO DETECT THE WI-FI AP
To study the additional delay to detect a Wi-FI AP, we consider a 5 Wi-Fi AP deployment scenario, where, Wi-Fi AP 1 and Wi-Fi AP 2 at 6 feet are ON, Wi-Fi AP 3 and Wi-Fi AP 4 at 10 feet are ON and Wi-Fi AP 5 at 15 feet is ON. We observe a large number of Wi-Fi packets on the air and moreover the LTE-U ON cycle interference impacts the delay in Wi-Fi transmissions. In HD, the total time for the LTE-U

TABLE 6. Other Additional Delay to Detect the Wi-Fi AP Due to the NI Hardware

CSAT Types	NI HW Delay
Header Decoding (HD)	1.4 S
Energy Detection (ED)	5.9 S
Auto-correlation (AC)	4.8 S
Machine Learning (ML)	3.1 S

BS to decode the BSSID is 1.4 seconds (i.e., Wi-Fi 1st BSSID beacon packet + LTE-U detects K beacon + Additional layer complexity + NI USRP RIO hardware processing time). In ED, the total time for the energy based CSAT algorithm to adopt or change the duty cycle from 50% to 33% is 5.9 seconds (i.e., Wi-Fi 1st beacon transmission time + LTE-U detects K beacon (or) data packets time + NI USRP RIO hardware processing time) as shown in Table 6. In AC, the total time for the AC based CSAT algorithm to change the duty cycle from 50% to 33% is 4.8 seconds (i.e., Wi-Fi 1st L-STF packet frame + LTE-U detects L-STF frame time + NI USRP RIO hardware processing time). In ML, the total time for the CSAT algorithm to adopt the duty cycle from 50% to 33% is about 3.1 seconds. This approach is dependent on the chunk size (in this case set to 512).
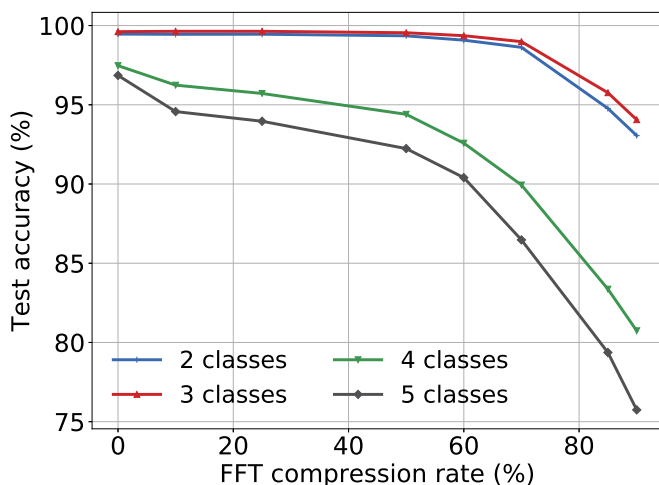
FIGURE 17. Effect of FFT compression embedded into the convolutional layers of the FCN model on test accuracy. We use the Case D configuration for 2 classes and the same configuration with NLOS and 6 feet for the remaining classes.

E. FFT COMPRESSION

We test the FCN model using the FFT based convolutional layers with compression [38]. The results are presented in Fig. 17. We observe that for 2 and 3 classes the data is highly compressible and we can allow up to even 60% compression with the test accuracy preserved on the level of above 99%. As we increase the number of classes, the accuracy of the model gracefully degrades and the 60% compression rate allows us to retain the test accuracy of about 90% for 5 classes.

We do not observe a significant difference between the cases with 2 and 3 classes. For 2 classes, we have 1 or 2 Wi-Fi APs and for 3 classes, we distinguish between 0, 1, or 2 Wi-Fi APs. The signal for no Wi-Fi APs is very different and hence easier to classify, than for the remaining signals with active Wi-Fi APs.

X. CONCLUSIONS AND FUTURE WORK

We have presented a comprehensive experimental study of different kinds of ML algorithms that could be used to address the problem of identifying the number of active Wi-Fi APs on the air to aid in setting the LTE-U duty cycle appropriately. Additionally, we have compared the performance of the optimum ML algorithm to conventional methods using energy detection and auto-correlation detection and demonstrated superior performance in multiple configurations. We believe that this is the first result that demonstrates the feasibility of using ML on energy values in real-time, instead of packet decoding [15], to reliably distinguish between the presence of different number of Wi-Fi APs. Such a result can have applications beyond LTE-U duty-cycle adaptation, for example in better Wi-Fi frequency management.

We aim to extend this work in the future by distinguishing between LTE-LAA BS and Wi-Fi APs for the coexistence scenario between Wi-Fi, LTE-U and LTE-LAA, thus enabling even finer duty cycle adjustments of a LTE-U BS

and improved coexistence with Wi-Fi. Also, we are interested in developing a ML framework that predicts the type of Wi-Fi traffic *i.e.,* voice, video, or data which in turn can further ensure fair access to the unlicensed spectrum since each traffic-type requires different transmission opportunity times (TXOPs) and per-traffic fairness is more important than per node (Wi-Fi AP) fairness. Similar concepts can also be applied to LTE-LAA/Wi-Fi coexistence deployments and future NR-U/Wi-Fi coexistence in the 6 GHz band.

REFERENCES

[1] V. CISCO, "CISCO visual networking index: Forecast and trends, 2017–2022," White Paper, vol. 1, 2018.
[2] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.
[3] G. Gampala and C. Reddy, "Massive MIMO beyond 4G and a basis for 5G," in *Proc. IEEE Int. Appl. Comput. Electromagn. Soc. Symp.*, 2018, pp. 1–2.
[4] V. Sathya, A. Ramamurthy, and B. R. Tamma, "On placement and dynamic power control of femtocells in LTE HetNets," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 4394–4399.
[5] F. C. Commission, "Notice of proposed rulemaking on unlicensed use of the 6 GHz band," 2018. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-18-147A1.pdf
[6] G. P. Project, "3GPP release 13 specification," [Online]. Available: http://www.3gpp.org/release-13/, 2015, Accessed: Dec. 12, 2019.
[7] M. Mehrnoush, S. Roy, V. Sathya, and M. Ghosh, "On the fairness of Wi-Fi and LTE-LAA coexistence," *IEEE Trans. Cogn. Commun. Netw.*, Dec. 2018. [Online]. Available: http://dx.doi.org/10.1109/TCCN.2018.2867032
[8] L. Forum, LTE-U CSAT Procedure TS V1.0, 2015.
[9] M. K. Singh, A. M. Baswade, and B. R. Tamma, "Wi-Fi user's video QoE in the presence of duty cycled LTE-U," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 720–722.
[10] F. S. Chaves *et al.*, "LTE UL power control for the improvement of LTE/Wi-Fi coexistence," in *Proc. IEEE 78th Veh. Technol. Conf.*, 2013, pp. 1–6.
[11] T. A. Atif, A. M. Baswade, B. R. Tamma, and A. A. Franklin, "A complete solution to LTE-U and Wi-Fi hidden terminal problem," *IEEE Trans. Cogn. Commun. Netw.*, 2019, pp. 920–934.
[12] V. Sathya, M. Merhnoush, M. Ghosh, and S. Roy, "Energy detection based sensing of multiple Wi-Fi BSSs for LTE-U CSAT," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–7.
[13] V. Sathya, M. Mehrnoush, M. Ghosh, and S. Roy, "Auto-correlation based sensing of multiple Wi-Fi BSSs for LTE-U CSAT," in *Proc. IEEE 90th Veh. Technol. Conf.*, 2019, pp. 1–7.
[14] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. Tut.*, vol. 21, no. 3, pp. 2224–2287, Jul.-Sep. 2019.
[15] E. Chai, K. Sundaresan, M. A. Khojastepour, and S. Rangarajan, "LTE in unlicensed spectrum: Are we there yet?" in *Proc. ACM 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 135–148.
[16] C. Cano and D. J. Leith, "Unlicensed LTE/WiFi coexistence: Is LBT inherently fairer than CSAT?" in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.
[17] Q. Chen, G. Yu, and Z. Ding, "Optimizing unlicensed spectrum sharing for LTE-U and Wi-Fi network coexistence," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2562–2574, Oct. 2016.
[18] V. Sathya, M. Mehrnoush, M. Ghosh, and S. Roy, "Association fairness in Wi-Fi and LTE-U coexistence," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2018, pp. 1–6.
[19] V. Sathya, M. Mehrnoush, M. Ghosh, and S. Roy, "Analysis of CSAT performance in Wi-Fi and LTE-U coexistence," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2018. [Online]. Available: http://dx.doi.org/10.1109/ICCW.2018.8403510
[20] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surv. Tut.*, vol. 21, no. 4, pp. 3072–3108, Oct.-Dec. 2019.

[21] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surv. Tut.*, vol. 21, no. 4, pp. 3039–3071, Oct.-Dec. 2019.

[22] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" 2019, *arXiv:1902.02647*.

[23] J. Tan, S. Xiao, S. Han, and Y.-C. Liang, "A learning-based coexistence mechanism for LAA-LTE based HetNets," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[24] S. Bayhan and G. Gür, "IEEE tutorial on machine learning for spectrum sharing in wireless networks."

[25] P. M. de Santana, V. A. de Sousa, F. M. Abinader, and J. M. d. C. Neto, "DM-CSAT: A LTE-U/Wi-Fi coexistence solution based on reinforcement learning," *Telecommun. Syst.*, pp. 1–12, 2019.

[26] N. Rastegardoost and B. Jabbari, "A machine learning algorithm for unlicensed LTE and Wi-Fi spectrum sharing," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.*, 2018, pp. 1–6.

[27] V. Maglogiannis, D. Naudts, A. Shahid, and I. Moerman, "A Q-learning scheme for fair coexistence between LTE and Wi-Fi in unlicensed spectrum," *IEEE Access*, vol. 6, pp. 27 278–27 293, 2018.

[28] V. Maglogiannis, A. Shahid, D. Naudts, E. De Poorter, and I. Moerman, "Enhancing the coexistence of LTE and Wi-Fi in unlicensed spectrum through convolutional neural networks," *IEEE Access*, vol. 7, pp. 28 464–28 477, 2019.

[29] I. L. (Quantenna), "Efficient and fair medium sharing enabled by a common preamble," [Online]. Available: http://grouper.ieee.org/groups/802/11/Workshops/2019-July-Coex/Quantenna_Contribution.pdf, 2019.

[30] R. AT&T, "Common preamble design in the 6 GHz band merits and challenges," AT&T Labs, Austin, TX, USA, 2019, [Online]. Available: http://grouper.ieee.org/groups/802/11/Workshops/2019-July-Coex/att_coex_ws_final.pdf

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[32] S. Krishnan, A. Dziedzic, and A. J. Elmore, "Deeplens: Towards a visual data management system," 2018, *arXiv:1812.07607*.

[33] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Architecture*, 2017, pp. 1–12.

[34] S. Chetlur *et al.*, "CUDNN: Efficient primitives for deep learning," 2014, *arXiv:1410.0759*.

[35] Y. Chen *et al.*, "The UCR time series classification archive," 2018. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data_2018

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[37] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2017, pp. 1578–1585.

[38] A. Dziedzic, J. Paparrizos, S. Krishnan, A. Elmore, and M. Franklin, "Band-limited training and inference for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1745–1754.

[39] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4013–4021.

[40] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun, "Fast convolutional nets with FBFFT: A GPU performance evaluation," 2014, *arXiv:1412.7580*.

[41] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statist. Its Interface*, pp. 349–360, 2009.

[42] P. Schfer, "Scalable time series classification," *Data Mining Knowl. Discovery*, vol. 30, 2015.

[43] P. Schäfer and U. Leser, "Fast and accurate time series classification with weasel," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017. [Online]. Available: https://doi.org/10.1145/3132847.3132980

[44] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.