Histograms to Quantify Dataset Shift for Spectrum Data Analytics: A SoC Based Device Perspective

Zaheer Khan*, Janne Lehtomäki*, Chanaka Ganewattha*, and Shahriar Shahabuddin†
University of Oulu*, Oulu, Finland, and Nokia†, Oulu, Finalnd
zaheer.khan@oulu.fi

***Abstract*—Cloud/software-based wireless resource controllers have been recently proposed to exploit radio frequency (RF) data analytics for a network control, configuration and management. For efficient resource controller design, tracking the right metrics in real-time (analytics) and making realistic predictions (deep learning) will play an important role to increase its efficiency. This factor becomes particularly critical as radio environments are generally dynamic, and the data sets collected may exhibit shift in distribution over time and/or space. When a trained model is deployed at the controller without taking into account dataset shift, a large amount of prediction errors may take place. This paper quantifies dataset shift in real wireless physical layer data by using a statistical distance method called earth mover's distance (EMD). It utilizes an FPGA to process in real-time the in-phase and quadrature (IQ) samples to obtain useful information, such as histograms of wireless channel utilization (CU). We have prototyped the data processing modules on a Xilinx System on Chip (SoC) board using Vivado, Vivado HLS, SDK and MATLAB tools. The histograms are sent as low-overhead analytics to the resource controller server where they are processed to evaluate dataset shift. The presented results provide insight into dataset shift in real wireless CU data collected over multiple weeks in the University of Oulu using the implemented modules on SoC devices. The results can be used to design approaches that can prevent failures due to datashift in deep learning models for wireless networks.**

***Index Terms*—System-on-Chip, Zynq-7000, dataset shift, deep learning models, spectrum analytics, FPGA, dataset shift, Xilinx, channel utilization, resource controller.**

I. Introduction

The evolution of wireless systems to the 5th generation (5G) and beyond is driven by low-latency demands, improved throughput requirements and additional use cases for wireless access, such as support for vehicular communications, and internet of things [1]. The next-generation (NG) of wireless networks is envisioned as a network of networks integrating multiple radio access technologies and providing spectrum access harmonization across licensed, and unlicensed shared spectrum bands. Due to the usage of more spectrum bands in future and diverse services, a natural research question to ask is: How to design an intelligent proactive resource provisioning solution for the NG of networks that can efficiently allocate various network resources for different services?

Proactive resource provision requires the use of deep learning based prediction techniques. However, most prediction techniques assume that training and target datasets have same distribution, and hence they produce erroneous predictions when target datasets have different distribution than the training dataset [2]. Wireless networks are dynamic and it is natural that real-wireless dataset distributions may change from time to time and also from one location to another location. This is called shift in a dataset and if a trained learning model is deployed at the cloud controller without adapting to the occurred dataset shift, a large amount of prediction errors may take place. Therefore, studying dataset shift is a crucial problem for deep learning models in wireless networks. In this paper, we study dataset shift in real wireless physical layer channel utilization (CU) in unlicensed 2.4GHz spectrum. CU is an important metric as use of deep learning based CU predictions can help facilitate proactive resource allocation in wireless networks. We use a statistical distance based technique called earth mover's distance (EMD) on histograms of real CU data to identify dataset shift. Simply put, we calculate histograms of CU data using our implemented modules on Xilinx's Zynq-7000 SoC devices.

Datasets in the form of histograms of real CU data can be obtained via IQ data processing. Radio frequency (RF) IQ data acquisition in a wireless system with MHz channel bandwidths can produce hundreds of millions of samples per second. Signal detection, channel utilization and histogram computation modules implemented on a device close to the network edge reduces overhead as transfer of several million raw IQ samples is no longer required and it also increases measurement speed, accuracy, and performance. For example, as gaps in IQ data acquisition can be lethal to valid inference in dataset shift analysis of CU data, the edge device allows gapless acquisition of streaming raw samples and their simultaneous processing. Gapless streaming also increases the number of samples used to compute histograms. To realize a complete solution, in this paper, we also present a low cost real-time CU histogram computation architecture. In our proposed solution, the histogram values are computed in real-time on a Zedboard which is Zynq-7000 based low cost SoC device and are sent to the cloud controller that utilizes the EMD to analyze dataset shift.

In Fig. 1, we illustrate the analytics system utilized for this paper. It shows the following components of the CU histogram computation modules on a Zynq SoC device: 1) An AD9361 RF frontend [3] attached to the SoC device for streaming reception of IQ samples from multiple active APs in an unlicensed 2.4GHz channel; 2) Various FPGA accelerated modules, such as noise floor estimation, signal detection, and CU state calculations; 3) Embedded processor modules, such as sample transfer via direct memory access (DMA) from FPGA, mean CU value computations, and CU histogram computations; and 4) a communication module between the SoC device and the server for streaming transfer of processed CU and histogram samples. The figure also shows a server where an EMD based datashift computations are performed

on collected CU histograms.

II. State of the art for Spectrum Data Analytics

3rd Generation Partnership Project (3GPP) has introduced an analytics function called NWDAF to incorporate data analytics functionality in the 5G and beyond network architecture, [1]. Virtualized cloud-based resource controllers that utilize dedicated measurement/data collection modules have been deployed to exploit useful information from such analytics functionality for a wireless network control, configuration and management [4]–[6]. For example, use of measurement capable devices (MCDs) and data analytics for 5G networks have been proposed in [7]. Moreover, regulatory bodies in [10] have decided to include new tools in which environment sensing capability (ESC) is an essential component for future shared spectrum operations in radar bands. Advances in software/hardware technologies and internet of things (IoTs) allow wireless operators to collect in real time network related data sets not only from their user equipments but also from their network elements, such as access points (APs) [8]. For example, Cisco System's Meraki Cloud Controller (MCC) utilizes dedicated WIPS (Air Marshal) radio modules in each of its Meraki APs to constantly monitor the behavior of the network [9]. However, MCC uses average channel utilization values to ensure channel arrangements for APs are made in a way that utilization is less than 50% on average.

Wireless networks operate in diverse environments and obtaining appropriate knowledge in real-time can be challenging [5]. In much of the research literature, there has been focus on non-real time spectrum analytics where IQ samples are collected using spectrum analyzers or SDR boards, such as WARP or USRP boards. The collected samples are then processed on laptops or PC servers to obtain CU values. These approaches can lead to performance limitations due to: 1) storing the samples in memory buffers and then processing them results in non-real time knowledge of the wireless environment. Storing samples is in general required due to slow transfer speed between IQ data collection module and the host processing them; and 2) due to huge sample quantity (hundreds of million samples per second) storing and then processing also leads to gaps in the collected IQ samples over time. The gaps in samples can degrade the accuracy of statistical analyses performed on raw data. To avoid these limitations, we implement CU histogram computations on an FPGA which can process values in real-time without any gaps.

CU values and CU histograms can be used to develop deep learning based CU predictions which in turn can be used in design of efficient resource allocation algorithms in both currently allocated licensed and new shared spectrum bands. Although deep learning has been currently intensively used in the context of wireless networks [8], however, to the best of knowledge no work has studied the problem of dataset shift in deep learning for real wireless CU data.

III. Spectrum Analytics and Device Design

A. Channel Utilization: Background

Wireless CU is a metric to represent the usage of a particular frequency or a channel within some measuring time interval t. Typically, CU indicates how much any transmissions the implemented CU computation device can "hear" on a channel, from all wireless sources. The CU is often given in a percentage between 0% to 100% and indicates the amount of time a device finds a channel to be busy. It includes all type of transmissions from all wireless sources operating in the channel. To measure CU, our implemented device directly processes IQ samples in real-time (with processing speed of several million samples per second). Signal is declared to be present when received I^2+Q^2 value exceeds a threshold, otherwise, it is declared to be absent. To accurately detect signals in real time, we have a noise floor estimation module which is used to set the detection threshold value appropriately. The closed-form expression for CU computation in a block i can be given as:

$$\psi_i = \frac{n_p}{N} \tag{1}$$

where n_p represents number of samples in block i in which signal is declared to be present, and N is the total number of samples in each block i. Note that depending on the channel bandwidth, the number of samples in each block i may vary from several hundred thousands to a few million.

B. CU Histogram Computation

Every received block of CU statistics samples over n time units contains a sequence of CU observations $\mathcal{A}_i = \{\psi_1, \psi_2, ..., \psi_n\}$ from an unknown distribution function F. Given n samples of CU in a block, an ith histogram $\mathcal{H}_i$ of CU is given by

$$\mathcal{H}_i = \{(I_1, \pi_1), (I_2, \pi_2), \cdots, (I_b, \pi_B)\} \tag{2}$$

where $I_1, I_2, \cdots, I_b$ are partitioning of CU into B contiguous intervals which are also known as bins. The count values for B bins are given by $\pi_1, \pi_2, \cdots, \pi_B$. Each bin has an interval $I_j = [\underline{I}_j; \bar{I}_j)$ with $\underline{I}_j$ as the minimum value and $\bar{I}_j$ as the maximum value. The implemented module computes histogram values as follows: When a sample of CU is within some bin I_j then the counter for that bin is incremented by one or else it remains the same.

C. Main hardware/software components for CU computations

The FPGA accelerated modules act directly on IQ samples and perform streaming real-time wireless CU measurements to compute the CU values and also perform histogram computations on obtained CU values. In particular, the FPGA accelerated modules perform three main tasks in parallel: a) it performs noise floor estimation and signal detection; b) it performs statistical computations on blocks of measured data and outputs their descriptive statistics, such as mean values, and histograms-based probability density functions; c) and it sends the low-overhead histogram outputs to the resource controller server.

The use of an FPGA and an embedded processing unit to obtain CU values and also CU histogram values allows partitioning of time critical signal processing tasks to be done on FPGA, letting the processor do less critical processing. In Fig. 2, we illustrate this partitioning of tasks between the FPGA and the processor by providing a simplified high level circuit diagram and also embedded processor functions that are implemented on the Zedboard. Fig. 2 shows FPGA implemented modules: 1) adaptive noise floor estimator, 2) detection

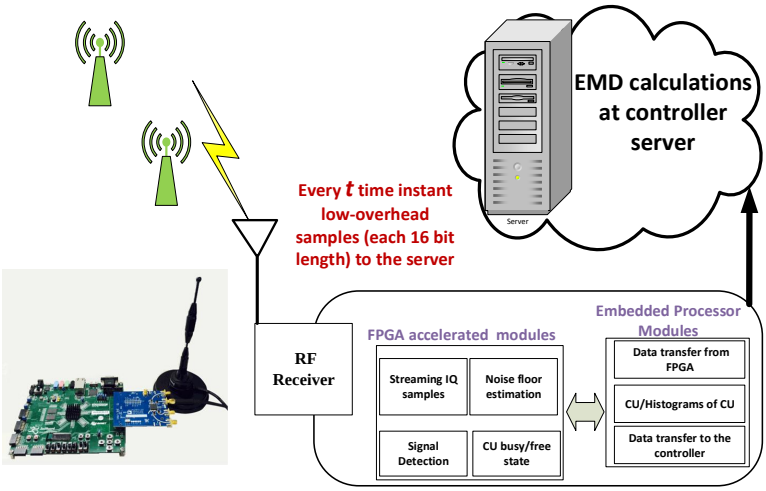


Fig. 1. Various components of our prototyped analytics device and analytics system.

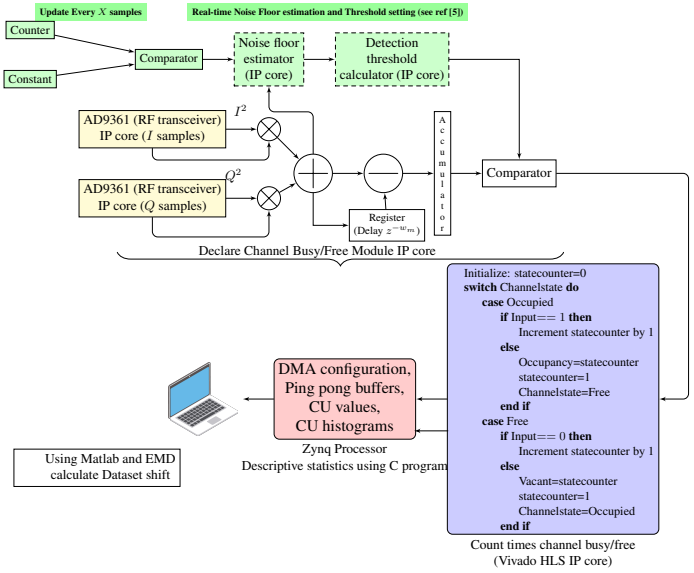
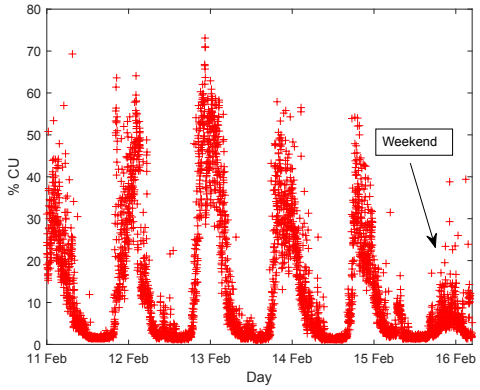
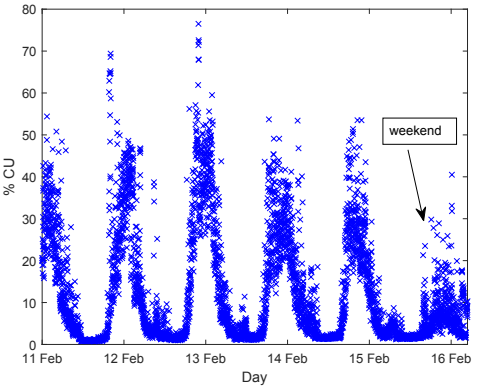


Fig. 2. High level circuit diagram and embedded processor modules.



(a) CU values obtained via prototyped device over five days.



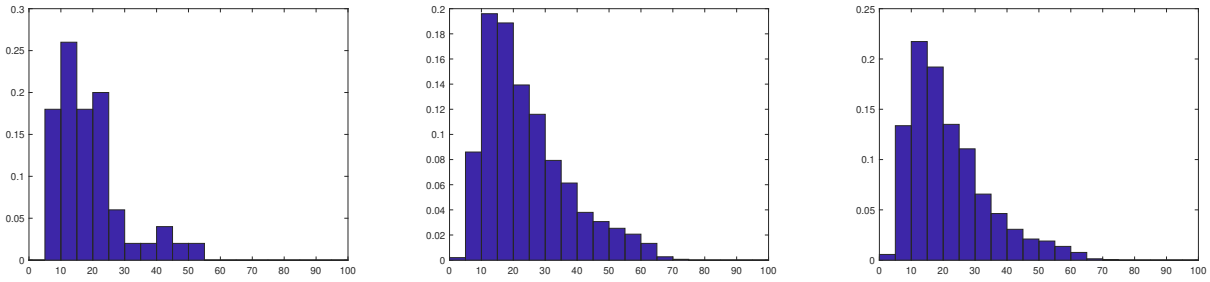
(b) CU values obtained via second prototyped device placed 20 metres away from the first device.

Fig. 3. Comparing sample CU values obtained via two prototyped devices.

threshold calculator, 3) channel busy/free state calculator, and 4) channel state dwell time (how long channel remains in busy/free state). Fig. 2 also presents various software modules implemented on the embedded processing system of the Zedboard: 1) configuring the DMA for streaming data transfers using interrupts; 2) Ping pong buffers based processing 3) block mean CU values and histogram computation of CU values. In our work in [5], we provide detailed explanation regarding the implementation of various modules in Fig. 2.

D. EMD

To convert the obtained CU histograms to probability distributions data one can simply divide the count values in each bin by the total number of count values in the histogram. We can then define EMD as the minimum amount of effort needed



(a) Normalized CU histogram obtained over 20 seconds interval.

(b) Normalized CU histogram obtained over 10 minutes interval.

(c) Normalized CU histogram obtained over 30 minutes interval.

Fig. 4. Normalized CU histograms over various time interval lengths.

to transform a probability distribution α (which represents the CU probability distribution at time t) towards probability distribution β (which represents the CU probability distribution at time $\acute{t}$). The effort can be defined in simple words as: effort $=$ (number of normalized CU count values moved) $\times$ (number of bins over which they are moved). Simply put, the idea of EMD is to imagine two probability distributions as piles of dirt and calculate the minimum amount of effort needed to reshape the first pile so that it has the same shape as the second pile. The important feature of EMD is that it takes into account distance. With increasing dissimilarity of two CU distributions the EMD increases because the probabilities need to be moved over larger number of bins (distances). Two exactly matching CU probability distributions will have zero EMD, while the maximum value for the EMD is $B-1$ bins as for the histogram case $B-1$ represents the thresholded value for EMD. For the maximum EMD value $B-1$ case, both CU distributions are completely separated and further apart. Moreover, one can obtain normalized EMD values between 0 and 1 if one divides the EMD with $B-1$.

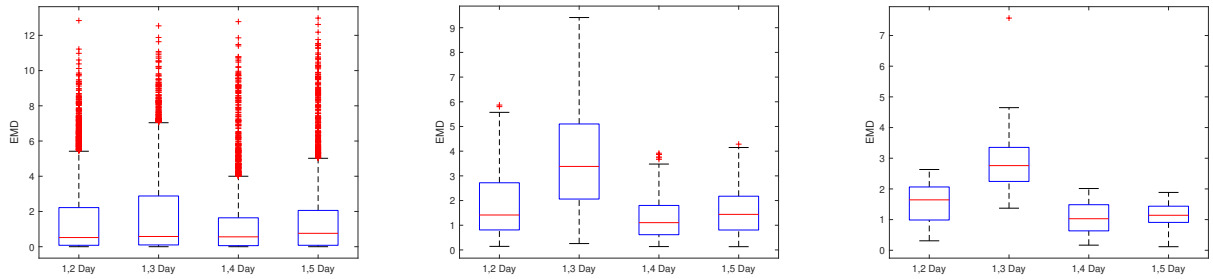
IV. Results from real over-the-air Data

In this section, we present detailed results relating to CU values and CU histograms which are collected using our implemented device. To obtain real wireless data, three Zedboard devices with implemented modules running on them were placed for over a period of two weeks in one of the busiest part (Tellus conference area) of the University of Oulu. The device collected mean CU, and CU histograms values in a 2.4GHz WLAN channel where multiple APs were active. The device was configured to output every 20 seconds a mean CU, and an entire CU histogram. The width of each histogram bin during measurements was set to 5 which means the collected histograms had a total of $B = 20$ bins. Further details regarding the utilized measurement devices can be found in [5].

In Figs. 3a and 3b we present measured mean CU values using two Zedboards which were placed around 20 metres apart. The two figures show the measured CU values for a period of 5 days including a weekend day. It can be seen that for day time the mean CU goes high and then for night time it goes low. Moreover, overall low mean CU can be also seen during the weekend as compared to the week day. In Figs. 4a, 4b, and 4c, we present obtained CU histograms over time interval lengths of 20 seconds, 10 minutes and 30 minutes, respectively. It can be seen from the figures that CU distribution is right skewed with a long right tail.

In a real wireless network, such as deployed in the University of Oulu, CU distribution can change according to the change in time. Moreover, within a given area, CU distribution can also change across locations which are not too much apart from each other. This is due to the reason that one location may receive different amount/strength of wireless signals than the other. To observe CU distribution changes across time in the collected real CU dataset, we have splitted the obtained CU histogram dataset into five parts. Each part represents a full day of a working week. For example, Part 1 represents Day 1 (Monday) and and Part 5 represents Day 5 (Friday). To see datashift across time we compute EMD for the same time between Day 1 and each of four other days of the week. In Figs. 5a, 5b, and 5c, we present box plots of EMD values for normalized histograms obtained over 20 seconds, 10 minutes, and 30 minutes, respectively. It can be seen from Fig. 5a that for the 20 second histogram interval case for the same time of the day the median (red line in the box) EMD values between Day 1 and each of four other days are no more than 1. For the same figure, the 75th percentiles (top edges of the boxes) are no more than 3. The whiskers show the most extreme point not considered as outliers can go as high as 7. Finally, it can be also seen that the outlier data points (+ symbols) can be as high as 13. Figs. 5b and 5c show that increasing the time interval of obtained CU histograms can slightly increase the median and the 75th percentile EMD values, however, the increase is not very high and the outlier EMD values have significantly decreased. The EMD results across time show that while the CU distribution across same time of different days are similar but they are exactly not the same. Hence, a deep learning model that uses certain week days CU data as training data to predict CU for other week days can only predict with limited accuracy.

To observe CU distribution changes across space in the collected real CU dataset, we have compared the obtained CU histogram dataset from two different Zedboards that were computing histograms almost 20 metres apart. In Figs. 6a, 6b, and 6c, we present box plots of EMD values for normalized histograms obtained over 20 seconds, 10 minutes, and 30 minutes, respectively. It can be seen from the three figures

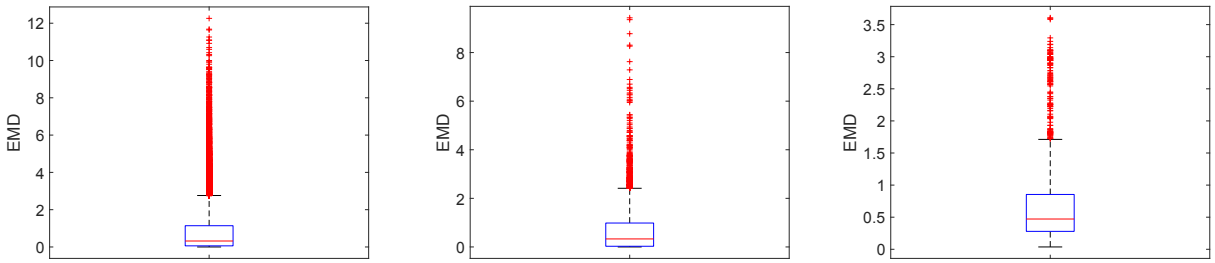


(a) EMD comparison for normalized histograms obtained over 20 seconds interval.

(b) EMD comparison for normalized histograms obtained over 10 minutes interval.

(c) EMD comparison for normalized histograms obtained over 30 minutes interval.

Fig. 5. EMD comparison across time, i.e., between a day and four other week days.



(a) EMD comparison for normalized histograms obtained over 20 seconds interval.

(b) EMD comparison for normalized histograms obtained over 10 minutes interval.

(c) EMD comparison for normalized histograms obtained over 30 minutes interval.

Fig. 6. EMD comparison across space, i.e., between histograms measured by one device and with histograms measured by another device which was placed 20 metres away from the first device.

that median values are no more than 0.5. Moreover, the 75th percentile is no more than 1 for all the three figures. The three figures also show that increasing the time interval of obtained CU histograms can slightly decrease the 75th percentile and also the EMD values for extreme points. The EMD results across space show that the CU distribution across different locations within a given area show small differences.

V. Concluding Remarks

It is challenging to get a publicly available real wireless CU dataset that can be used to study dataset shift in wireless CU. The main contributions of this paper is two-fold. First, we present our SoC based implemented devices that are used to obtain real CU values and CU histogram values. Second, we utilize the collected CU data to study dataset shift across time and space in wireless CU distributions. We use a statistical distance based technique called EMD to quantify the shift in dataset. Our results show that CU distributions across time are similar but not the same which means there is some dataset shift across time and it can affect the prediction performance of any deep learning models developed for CU predictions.

References

[1] K. Ganesan, P. B. Mallick, J. Löhr, D. Karampatsis, and A. Kunz, "5G V2X architecture and radio aspects," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2019, pp. 1–6.

[2] A. Subbaswamy, P. Schulam, and S. Saria, "Preventing failures due to dataset shift: Learning predictive models that transport," in *Proceedings of Machine Learning Research*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 3118–3127. [Online]. Available: http://proceedings.mlr.press/v89/subbaswamy19a.html

[3] Z. Yang, W. Xiong, and Y. Zhao, "Software defined radio hardware design on ZYNQ for signal processing system," in *2019 8th International Symposium on Next Generation Electronics (ISNE)*, Oct 2019, pp. 1–3.

[4] Q. Qin, K. Poularakis, G. Iosifidis, S. Kompella, and L. Tassiulas, "SDN controller placement with delay-overhead balancing in wireless edge networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1446–1459, Dec 2018.

[5] Z. Khan and J. J. Lehtomäki, "FPGA-assisted real-time RF wireless data analytics system: Design, implementation, and statistical analyses," *IEEE Access*, vol. early access, pp. 1–13, Dec 2019.

[6] T. Cooklev, V. Poulkov, D. Bennett, and K. Tonchev, "Enabling RF data analytics services and applications via cloudification," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 5-6, pp. 44–55, May 2018.

[7] L. Laughlin, F. Boccardi, C. Gamlath, E. Arabi, K. C. Balram, K. A. Morris, and M. A. Beach, "Emerging hardware enablers for more efficient use of the spectrum," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Nov 2019, pp. 1–9.

[8] S. Gupta, M. Mittal, and A. Padha, "Predictive analytics of sensor data based on supervised machine learning algorithms," in *International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, Dec 2017, pp. 171–176.

[9] K. Sui, M. Zhou, D. Liu, M. Ma, D. Pei, Y. Zhao, Z. Li, and T. Moscibroda, "Characterizing and improving WiFi latency in large-scale operational networks," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16. New York, NY, USA: ACM, 2016, pp. 347–360. [Online]. Available: http://doi.acm.org/10.1145/2906388.2906393