MACHINE LEARNING FOR PERFORMANCE PREDICTION OF CHANNEL BONDING IN NEXT-GENERATION IEEE 802.11 WLANS

Francesc Wilhelmi[1], David Góez[2], Paola Soto[3], Ramon Vallés[1], Mohammad Alfaifi[4], Abdulrahman Algunayah[4],
Jorge Martín-Pérez[5], Luigi Girletti[5], Rajasekar Mohan[6], K Venkat Ramnan[6], Boris Bellalta[1]
[1]Universitat Pompeu Fabra, Spain, [2]Universidad de Antioquia, Colombia, [3]University of Antwerp, Belgium,
[4]Saudi Telecom, Saudi Arabia, [5]Universidad Carlos III de Madrid, Spain, [6]PES University, India

NOTE: Corresponding author: Francesc Wilhelmi, fwilhelmi@cttc.cat

***Abstract** – With the advent of Artificial Intelligence (AI)-empowered communications, industry, academia, and standardization organizations are progressing on the definition of mechanisms and procedures to address the increasing complexity of future 5G and beyond communications. In this context, the International Telecommunication Union (ITU) organized the first AI for 5G Challenge to bring industry and academia together to introduce and solve representative problems related to the application of Machine Learning (ML) to networks. In this paper, we present the results gathered from Problem Statement 13 (PS-013), organized by Universitat Pompeu Fabra (UPF), whose primary goal was predicting the performance of next-generation Wireless Local Area Networks (WLANs) applying Channel Bonding (CB) techniques. In particular, we provide an overview of the ML models proposed by participants (including artificial neural networks, graph neural networks, random forest regression, and gradient boosting) and analyze their performance on an open data set generated using the IEEE 802.11ax-oriented Komondor network simulator. The accuracy achieved by the proposed methods demonstrates the suitability of ML for predicting the performance of WLANs. Moreover, we discuss the importance of abstracting WLAN interactions to achieve better results, and we argue that there is certainly room for improvement in throughput prediction through ML.*

***Keywords** – Channel bonding, IEEE 802.11 WLAN, ITU Challenge, machine learning, network simulator*

1. INTRODUCTION

The utilization of Artificial Intelligence (AI) and Machine Learning (ML) techniques is gaining momentum to address the challenges posed by next-generation wireless communications. The fact is that networks are nowadays facing unprecedented levels of complexity due to novel use cases including features such as spatial multiplexing, multi-array antenna technologies, or millimeter wave (mmWave) communications. While these features allow providing the promised performance requirements in terms of data rate, latency, or energy efficiency, their implementation entails additional complexity (especially for crowded and highly dynamic deployments), thus making hand-crafted solutions unfeasible.

IEEE 802.11 Wireless Local Area Networks (WLANs) are one of the most popular access solutions in the unlicensed band, and they represent a prominent example of increasing complexity in wireless networks. The optimization of WLANs underlines particular challenges due to the decentralized nature of these types of networks, which mostly operate using Listen-Before Talk (LBT) transmission procedures. If to this we add that WLAN deployments are typically unplanned, dense, and highly dynamic, the complexity is even increased.

To address the optimization of next-generation WLANs, the usage of AI/ML emerges as a compelling solution by leveraging useful information obtained across data, which allows deriving models from experience. Nevertheless, the adoption of AI/ML in networks is still in its initial phase, and a lot of work needs to be done. In this regard, standardization organizations are undertaking significant efforts towards fully intelligent networks. An outstanding example can be found in the International Telecommunication Union (ITU)'s ML-aware architecture [1], which lays the foundations of pervasive ML for networks.

Another aspect essential for the prosperity of AI/ML in communications is data availability and openness. In this context, the *ITU AI/ML in 5G Challenge* [2] was set in motion to encourage industry, academia, and other stakeholders to collaborate and exchange data for solving relevant problems in the field. This initiative entailed a big step forward in bringing open source closer to standards.

As a contribution to the ITU challenge, and aligned with WLANs optimization, in this paper, we present the results obtained from problem statement *"Improving the capacity of IEEE 802.11 Wireless Local Area Networks (WLANs) through Machine Learning"* (referred to as PS-013 in the context of the challenge), whereby participants were called to design ML models to predict the performance of next-generation Wi-Fi deployments. In this article, we gather all the work done in the context of the above-mentioned problem statement and provide a compilation of the proposed ML models used to address the problem of CB in WLANs.

To carry out the ITU AI for 5G challenge, an open data set with WLAN measurements obtained from a network simulator was made available. As later discussed, network simulators are gaining importance to enable future ML-aware communications by acting as ML sandboxes. In the context of the challenge, synthetic data was used for training ML models.

In summary, based on the proposed problem statement and the solutions provided by participants, we discuss the feasibility of predicting the throughput of complex WLAN deployments through ML. To the best of our knowledge, this is an under-researched subject with high potential. Accurate performance predictions may open the door to novel real-time self-adaptive mechanisms able to enhance the performance of wireless networks by leveraging spectrum resources dynamically. For instance, given a change in the network configuration, predictions about future performance values can be used as a heuristic to guide the choices made by algorithms that operate in decentralized self-configuring environments [3].

Table 1 briefly summarizes all the models proposed by the participants of the challenge and the main motivation behind them. For instance, the model proposed by the *ATARI* team aims to exploit the graph structure inherent in WLAN deployments. Alternatively, the solution provided by *Ramón Vallés* is focused on abstracting different categories of features (e.g., signal quality, bandwidth usage) and generate predictions based on them.

Table 1 – Summary of the ML models proposed by the participants of the challenge.

Team	Proposed Model	Motivation	Ref.
ATARI	Graph Neural Network	Exploit graph representation of WLANs	[29]
Ramon Vallés	Feed-forward Neural Network	Abstract problem characteristics by categories	[31]
STC	Gradient Boosting	High performance, flexibility, and ease of deployment	[35]
UC3M NETCOM	Feed-forward Neural Network	Learn throughput function exhaustively	[36]
NET INTELS	Random Forest Regression	Address problem's non-linearity and reduce dimensionality	[37]

The results presented in this paper showcase the feasibility of applying ML to predict the performance of next-generation WLANs. In particular, some of the proposed models have been shown to achieve high prediction accuracy in a set of test scenarios. Moreover, we have identified the main potential and pitfalls of the proposed models, thus opening the door to new contributions that improve the baseline results shown in this paper. The data set is available online [4], and we expect it can be used for benchmarking other ML methods in the future.

The remainder of this paper is structured as follows: first, Section 2 presents the CB problem for next-generation WLANs and describes the data set provided for throughput prediction in dense deployments. Then, Section 3 gives an overview of the ML-based solutions proposed by the ITU challenge participants, for which the results are provided in Section 4. Finally, Section 5 concludes the paper with final remarks.

2. CHANNEL BONDING IN NEXT-GENERATION WLANS

In this section, we first describe the CB problem in WLANs and underscore the need for ML models for performance prediction. Then, we introduce the data set provided for the ITU challenge, which is open to any researcher interested in this topic.

2.1 Channel bonding in IEEE 802.11 WLANs

Next-generation IEEE 802.11 WLANs are called to face the challenge of providing high performance under complex situations, e.g., to provide high throughput in massively crowded deployments where multiple devices co-exist within the same area. To fulfill the strict requirements derived from novel use cases, features such as Multiple-Input Multiple-Output (MIMO), Spatial Reuse (SR), or multi-Access Point (AP) coordination are being developed and incorporated into the newest amendments, namely IEEE 802.11ax (11ax) and IEEE 802.11be (11be) [5, 6].

One of the features that are receiving more attention is Channel Bonding (CB) [7, 8], whereby multiple frequency channels can be bonded with the aim of increasing the bandwidth of a given transmission, thus potentially improving the throughput. Since its introduction to the 802.11n amendment, where up to two basic channels of 20 MHz could be bonded to form a single one of 40 MHz, the specification on CB has evolved and currently allows for channel widths of 160 MHz (11ac/11ax). Moreover, CB is expected to support up to 320 MHz channels in the 11be amendment.
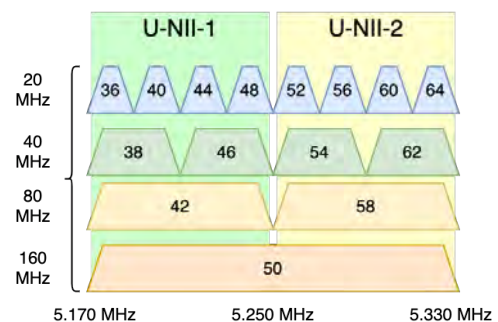

Fig. 1 – U-NII-1 and U-NII-2 sub-bands of the 5 GHz Wi-Fi channels.

Fig. 1 shows a snapshot of the 5 GHz band in Wi-Fi (particularly, U-NII-1 and U-NII-2 bands are shown), where basic 20 MHz channels can be bonded to form wider channels of up to 160 MHz. Each allowed channel is represented by an identifier. For instance, channel 36 is the first 20 MHz channel in the U-NII-1 band. As can be appreciated, the number of combinations for bonding channels is high, even for a small portion of the available spectrum[1] and under the constraint that only contiguous channels can be bonded. Moreover, novel bonding techniques combine Orthogonal Frequency Multiple Access (OFDMA) with preamble puncturing [9] to use non-contiguous channels. With all this, given the number of Basic Service Sets (BSSs) and devices in crowded deployments (see Fig. 2), we can say that CB is a problem with a combinatorial action space.
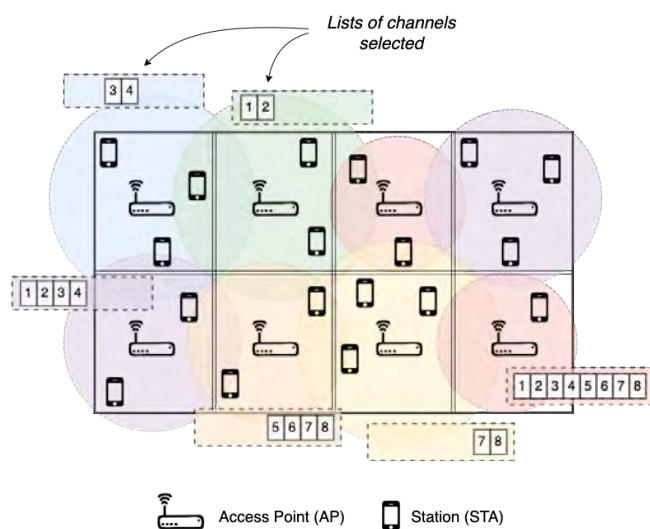

Fig. 2 – Dense WLAN deployment with multiple CB configurations. Each number in the list of channels represents one basic 20 MHz channel.

2.2 Policies for dynamic channel bonding

To harness the available spectrum within the CB operation, Dynamic CB (DCB) mechanisms [7, 8] are applied to decide the set of channels for transmitting on a per-packet basis, thus potentially improving performance. In [8], the following DCB policies were proposed and analyzed:

- **Static Channel Bonding (SCB):** a transmitter is allowed to use the entire set of channels only, thereby limiting the election of any subset of channels. While such a policy may optimize the performance in isolated deployments, it lacks the necessary flexibility to deal with inter-BSS interference.

- **Always-Max (AM):** in this case, the widest combination of channels is picked upon having sensed them free during the back-off procedure. While such a policy seems to properly harness the available spectrum, it has also been shown to generate starvation and other issues as a result of inter-BSS interactions.

- **Probabilistic Uniform (PU):** as an alternative to SCB and AM, PU is introduced to add some randomness in the process of picking free channels, so that any combination is chosen with the same probability. This policy has been shown to improve both SCB and AM in some scenarios in which flow starvation was present due to inter-BSS interactions.

To better illustrate the behavior of CB policies, Fig. 3 shows a simplification of the transmission procedure that a Station (STA) follows when implementing AM. In particular, CB may be applied over channels 1-4. Based on the AM policy, at time t_1, the STA can transmit only over channel 2, which is the only one that is sensed free at the moment of initiating a transmission. Similarly, in t_2, both channels 1 and 2 are found free, so the transmission is performed over those two channels. Finally, provided that the entire spectrum is free, a transmission over channels 1-4 is performed at t_3. Notice that, if applying SCB, the STA would have not been able to transmit until t_3. Alternatively, regarding PU, any combination of free channels could have been selected in t_2 and t_3.
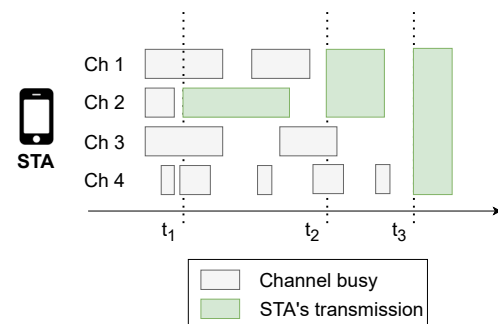

Fig. 3 – Dynamic channel selection for transmitting when applying DCB.

Through the analysis conducted in [8], it was shown that the right channel choice is not always trivial (i.e., selecting the widest channel does not necessarily entail achieving the highest performance). First of all, using wider channels entails spreading the same transmit power over the selected channel width, which can potentially affect the data rate used for the transmission, and therefore the capabilities of the receiver on decoding data successfully. Moreover, the potential gains of DCB in crowded deployments are hindered by the interactions between Wi-Fi devices, which may provoke contention or collisions. The fact is that WLAN deployments are unplanned and operate under Carrier Sense Multiple Access (CSMA). From the perspective of a given transmitter-receiver pair, such a lack of coordination leads to uncontrolled interference that can potentially degrade their performance.

Other DCB mechanisms were proposed in [10, 11, 12, 13], which include collision-detection, carrier sensing adaptation, or traffic load awareness. More recently, ML and game theory have been applied to address CB as an online decision-making problem involving multiple agents [14, 15].

In view of the complexity of selecting the best configuration of channels, the proposed problem statement had the goal of shedding light on the potential role of ML in DCB. In particular, it served to gather participants' proposals of ML models able to predict the performance of different CB configurations. This information can be used by a decision-making agent to choose the best configuration of channels before initiating a transmission. Throughput prediction in WLANs has been widely adopted for performance analysis through mathematical models, including the well-known Bianchi model [16], Continuous-Time Markov Networks (CTMNs) [17], or stochastic geometry [18]. However, these well-known models lack applicability for online decision-making because they fail to capture important phenomena either on the PHY or the MAC, or they entail a high computational cost. Thus, modeling high-density complex deployments through these models may be highly inaccurate or simply intractable.

In this regard, we envision ML models to assist the CB decision-making procedure in real time. The fact is that ML can exploit complex characteristics from data, thus allowing to solve problems that are hard to solve by hand-programming (see, for instance, its success in image recognition). Moreover, ML models can be trained offline and then improve their accuracy with measurements acquired online. To the best of our knowledge, this is an under-researched subject. While ML has been applied for predicting aspects related to Wi-Fi networks, such as traffic and location prediction [19, 20], it has been barely applied for explicitly predicting their performance. In this context, the work in [21] provided an ML-based framework for Wi-Fi operation, which includes the application of Deep Learning (DL) for waveforms classification, so that WLAN devices can identify the medium as idle, busy, or jamming. Closer in spirit to our work, [22] proposed an ML-based framework for WLANs' performance prediction.

2.3 Introduction to the data set

To motivate the usage of ML for predicting WLANs' performance, we provide an open data set[2] obtained with the Komondor simulator.[3] Komondor is an open-source IEEE 802.11ax-oriented simulator, whose fundamental operation has been validated against ns-3 in [23]. Komondor was conceived to cost-effectively simulate complex next-generation deployments implementing features such as channel bonding or spatial reuse [24]. Furthermore, it includes ML agents, which allows simulating the behavior of online learning mechanisms to optimize the operation of WLANs during the simulation.

The data set generated with Komondor has been used for training and validating ML models in the context of the ITU AI for 5G Challenge. The assets provided comprise both training and test data sets corresponding to multiple random WLAN deployments at which different CB configurations are applied. As for training, two separate enterprise-like scenarios, namely, *training1* and *training2*, have been characterized. In each case, a different fixed number of BSSs coexist in the same area, according to users' density.

In *training1*, there are 12 APs, each one with 10 to 20 associated STAs. Regarding *training2*, it contains 8 APs with 5 to 10 STAs associated with each one. For both training scenarios, three different map sizes have been considered (*a*, *b*, and *c*), where STAs are placed randomly. Similarly to training scenarios, the test data set includes a set of random deployments depicting multiple CB configurations and network densities. In this case, four different scenarios have been considered according to the number of APs (4, 6, 8, and 10 APs). Note, as well, that, for each type of scenario, 100 and 50 random deployments have been generated for training and testing, respectively. In all the cases, downlink UDP traffic was generated in a full-buffer manner (i.e., each transmitter always has packets to be delivered). Table 2 summarizes the entire data set in terms of the simulated deployments.

Table 2 – Summary of the simulated deployments used for generating both training and test data sets.

	Scenario id	Map width	# APs	# STAs
Training	training1a	80 x 60 m	12	10-20
	training1b	70 x 50 m		
	training1c	60 x 40 m		
	training2a	60 x 40 m	8	5-10
	training2b	50 x 30 m		
	training2c	40 x 20 m		
Test	test1	80 x 60 m	4	2-10
	test2		6	
	test3		8	
	test4		10	

With respect to input features, these are included in the files used for simulating each random deployment. In particular, the most relevant information to be used for training ML models is:

1. **Type of node:** indicates whether the node is an AP or an STA.

2. **BSS id:** identifier of the BSS to which the node belongs.

3. **Node location:** {x,y,z} coordinates indicating the position of the node in the map.

4. **Primary channel:** channel at which carrier sensing is performed.

---

5. **Channels range:** minimum and maximum channels allowed for bonding.

6. **Transmit power:** power used for transmitting frames.

7. **Sensitivity threshold:** used for detecting other transmissions and assess the channels' availability.

8. **Received Signal Strength Indicator (RSSI):** power detected at receivers from their corresponding transmitters.

9. **Inter-BSS interference:** power sensed from other ongoing transmissions.

10. **Signal-to-Interference-plus-Noise Ratio (SINR):** average SINR experienced during packet receptions.

Regarding output labels, we provide the throughput obtained by each device during the simulation, being the APs' throughput the aggregate throughput of each BSS (i.e., the sum of all the individual STAs' throughput in a given BSS). Moreover, the airtime per AP is provided, which indicates the percentage of time each BSS has occupied each of its assigned channels.

3. MACHINE LEARNING SOLUTIONS FOR THROUGHPUT PREDICTION

In this section, we give an overview of the solutions proposed by the participating teams of PS-013 in ITU AI for 5G Challenge: *ATARI* (University of Antwerp and Universidad de Antioquia), *Ramon Vallés* (Universitat Pompeu Fabra), *STC* (Saudi Telecom), *UC3M NETCOM* (Universidad Carlos III de Madrid), and *Net Intels* (PES University). From these teams, ATARI, Ramon Vallés, and STC succeeded to advance to the Grand Finale, where teams from all the problem statements competed for winning the global challenge [25].

3.1 ATARI

Wireless networks can be represented by graphs G=(V, E), where V is the set of nodes, i.e., STAs and APs, and E represent wireless links. Typically, DL approaches deal with graph-structured data by processing the data into simpler structures, e.g., vectors. However, nodes and links in high-density WLAN deployments are characterized by a set of high-dimensional features, thus complicating the graph-like structure of the problem and therefore hindering the application of deep learning.

To overcome the problem of data representation, Graph Neural Networks (GNNs) have been proposed as neural networks that operate on graphs intending to achieve relational reasoning and combinatorial generalization [26]. Accordingly, the wireless interactions between STAs and APs (connectivity, interference, among others) can be easily captured via a graph representation.

Therefore, we select a GNN approach to predict the throughput of the devices in a WLAN. In particular, each deployment is considered as a directed graph where STAs and APs are the graph's nodes. Additionally, we define two types of nodes present in the data set, each one having generic or specific features. For instance, parameters like channel configuration are related to both types of nodes, while SINR is only related to STAs, and airtime is exclusive for APs. Furthermore, the edges are defined based on the type of wireless interaction derived from the data set. We consider two types of interactions, namely AP-AP interactions (represented by the interference map), and AP-STA interactions (represented by the RSSI values). For completeness, we define an additional edge feature based on the distance of every transmitter-receiver pair. The features considered for training the proposed GNN are summarized in Table 3.

Table 3 – Features used by ATARI team to train a GNN.

	Feature	Preprocessing
Node	Node type	AP=0, STA=1
	Position (x,y)	None
	Primary channel	Combined into a
	Min. channel	categorical variable
	Max. channel	using one-hot encoding
	SINR	None
	Airtime	Mean
Edge	Edge type	AP-AP=0, AP-STA=1
	Distance	Computed from (x,y)
	RSSI	None
	Interference	None

Concerning the GNN model, we have used an implementation of the Graph Network Block (GNB), as proposed in [27]. A GNB contains three update functions and three aggregation functions where the computation is done from the edge to nodes, and then to global parameters. So first, the edge's features are updated and aggregated into the node features, then the node features are updated having taken into account the vicinity within depth/range defined by the number of GNBs, and lastly, the global parameters are updated according to the state of the nodes. Our model follows a layered approach, similar to DL, where each layer is a GNB. The input of the model is a graph representing the deployment, and the output is the predicted throughput of the devices in that deployment. A general overview of our model's architecture is shown in Fig. 4.

The implementation of the GNB is referenced as a meta-layer in PyTorch Geometric [28], a geometric deep learning extension library for PyTorch. We defined an edge model that uses two dense layers using a Rectified Linear Unit (ReLU) as an activation function in a typical Multi-Layer Perceptron (MLP) configuration. A node model is also defined by two MLPs, one for aggregating the edge features into the node features and the second to update
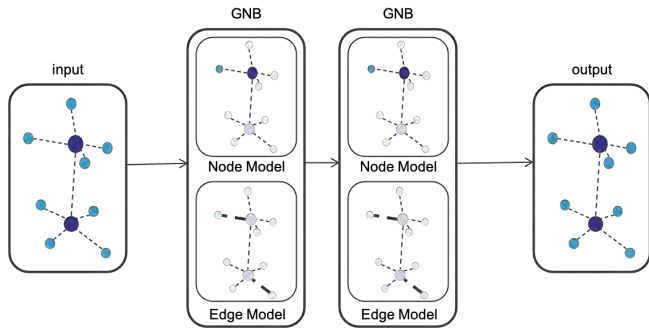
Fig. 4 – GNN model proposed by ATARI.

nodes' states based on their neighbors. Note that, following the formal definition of a GNB in [27], global parameters were not used. Our implementation is available in GitHub [29].

To train the model, we have considered splitting the provided data set (80% for training and 20% for validation). As each deployment is considered to be represented by a graph, 480 graphs have been used for training, and 120 for validation purposes. The loss function considered to assess the performance of our model is the Root Mean Squared Error (RMSE). The error obtained across all the predictions is used to compare the accuracy of our model's predictions to the actual results. However, initial results showed that the model mostly focused on predicting the throughput of the APs, given that the error is minimized on large values. Therefore, we proposed a masked loss provided that AP's throughput should be equal to the sum of the associated STAs' throughput. The RMSE is calculated using the STAs' predicted and the APs' computed throughput.

3.2 Ramon Vallés

To address the throughput prediction problem in CB-compliant WLANs, we propose a deep neural network where the information of each BSS is processed independently, thus following the idea of Multi-Layer Perceptron (MLP) [30]. More specifically, the proposed model is a feed-forward deep learning algorithm implemented in Python with the support of the PyTorch libraries.[4]

Our model aims to predict the aggregate throughput of each BSS, rather than the individual throughput at STAs. The fact is that predicting the throughput per STA is very challenging because of the dynamic channel bonding policy used in complex scenarios, which contributes to generating multiple interactions among nodes that cannot be captured at a glance. Accordingly, to derive an overall representation of each BSS, the features from individual STAs are preprocessed so that we consider only their global distribution (mean, and standard deviation). To do so, our model is divided into three main blocks performing different tasks:

---

[4]The code used to implement the method proposed by Ramon Vallés is open access [31].

1. **Signal quality:** The first block is meant to abstract the interactions among APs and STAs in the RF domain. To achieve this, we consider two separate layers, which process the RSSI, SINR, distance among nodes, and SINR. A Parametric Rectified Linear Unit (PReLU) activation function is used together with 1-dimensional batch normalization.

2. **AP bandwidth:** The second block analyzes the available bandwidth of the APs, and it consists of a single linear layer, which receives as input a vector with the corresponding airtime for all the available channels. This layer outputs a 3-dimensional array and is activated with a PReLU function.

3. **Output:** Finally, the last block takes the output from both the signal quality and AP bandwidth blocks and computes the final prediction value. For this final layer, we employ a simple ReLU (instead of a PReLU) to avoid negative throughput predictions.

With this structure, we have built a much more efficient model than if we had used a fully-connected NN with all the STA features. Notice that the proposed model needs far fewer neurons (and thus, less computational force) to capture the most relevant information of each scenario. In our opinion, an excess of neurons in such a complex scenario would result in overfitting, thus making the model less accurate for predicting the performance of new deployments.

As for training the MLP model, we have considered the following key features: the type of device (AP or STA), its location, the minimum, and maximum channel through which it is allowed to transmit, the RSSI, the SINR, and the airtime. The training was performed using 80% of the data set (keeping the 20% left for validation), following an evaluation criterion based on the RMSE of the predicted throughput with respect to the real value. The Adam optimizer has been used to optimize the training process, which is straightforward to implement, computationally efficient, and memory-efficient. For the training phase, several experiments had been made by modulating the hyper-parameters. The best results were achieved with a learning rate of 0,025 and a total of 700 epochs.

3.3 STC

Our proposal includes popular ML regression algorithms such as MLP, Support Vector Machine (SVM), Random Forest, and eXtreme Gradient Boosting (XGboost). These algorithms are backed by rich research, known to do well on regression problems, ease of implementation and deployment, which are important characteristics from the business perspective.

From all the above-mentioned methods, XGBoost [32] was selected for competing in the challenge because it offered the highest performance on both the training and validation stages compared with the other models. XGboost is a gradient boosting framework available for multiple platforms, thus providing high portability.

To train our model, we have analyzed the variance of the variables in the data set, and discarded the features with low or moderate variability. Some of the considered features are the position of nodes, the primary channel, the distance among nodes, or the power received by neighboring devices. To preprocess the selected features, we have applied Yeo-Johnson transformation [33] and normalization. These steps were done for predictors to improve their utility in the models.

The training data set was split into training and validation, so that we could train the model on the whole training data set using a 10-fold cross-validation procedure. Further, we used 100 and 300 combinations, respectively, with 10-fold cross-validation.

As for the hyper-parameter setting (e.g., max depth or minimum child weight), we tuned hyper-parameters using a grid search. More specifically, the hyper-parameter values were set using Latin Hypercube Sampling, while their maximum and lower range value for each hyper-parameter were mostly predetermined using default values from tidymodels [34].

Finally, significant efforts have been put to deploy our model. In particular, we have used docker to make our model easy to (re)train and deploy. All the code and documentation has been made publicly available [35].

3.4 UC3M NETCOM

We formulate the throughput forecasting in WLANs as a linear regression problem, which assumes a linear relationship between the features and label(s) of a given data set $\{y_i, \ x_{i,1}, \ x_{i,2}, \ ..., x_{i,N}\}$, and a set of unknown parameters w to be learned (being w_0 the bias).

Our solution (named *Gossip*) is based on a linear regression method, and it aims to predict the throughput of an STA in a given WLAN deployment where CB is applied [36]. Based on STAs' individual throughput, we derive the performance of each AP by aggregating the values of their associated STAs. In particular, Gossip derives the unknown bias and weight parameters w by (i) processing the WLAN data set, and (ii) applying a neural network to perform regression.

As for the processing part, Gossip takes the input features generated by the Komondor simulator, and selects/generates the most relevant ones: the position of the STA, the AP to which the STA is associated, the RSSI, the SINR, the set of nodes using the same primary channel, and the set of allowed channels. After processing the

features, each STA of every deployment is characterized by a feature vector $(x_i, ..., x_{21+3k})$, with k denoting the number of wireless channels. Note, as well, that the entire data set is considered for training, thus combining STAs from different deployments. The rationale is that features such as the number of neighbors in the primary channel, the SINR, and the interference should differentiate STAs from different deployments.

When it comes to the regression problem, Gossip uses a feed-forward neural network with four layers. The input layers pass the input features to two fully connected layers of neurons with a ReLU activation unit. Finally, a single neuron receives the output of the hidden layers and generates the prediction of the throughput. As a remark, the last neuron has a linear activation. It is important to remark that the proposed neural network is mostly meant to tackle linear regression problems. Nevertheless, even if the throughput prediction problem for WLANs is not linear, we expect our model to properly identify local minimum/maximum points that allow providing reasonable prediction results.

To train the proposed neural network, we have used the RMSprop gradient descend method, considering the Mean Squared Error (MSE) as a loss function. Moreover, 50 training episodes and a batch size of 50 STAs have been considered. Thanks to Gossip design, the training data set is populated with every STA of every deployment present among all scenarios.

3.5 Net Intels

To address the objective of predicting the throughput of APs and STAs in typical dense environments, we explore a set of popular regression techniques. With the help of these techniques, we aim to build complex mathematical relationships among features and labels from the data set, so that performance of WLANs can be predicted at unseen deployments. In particular, we propose using the following techniques:[5]

1. **Artificial Neural Network (ANN):** The ANN method is selected chiefly due to its potential and versatility to model nonlinear and complex relationships in OBSS data elegantly. The proposed ANN is built using Tensorflow and Keras libraries in Python [38]. The NN model is designed with one input layer, 7 hidden layers, and 1 output layer (see Fig. 5). The ReLU function is employed to activate hidden layers. In each of the first six hidden layers, there are 1024 nodes. For the seventh hidden layer, there are 512 nodes. The model is trained using an Adam optimizer. The batch size and number of epochs for training, after multiple trials, were set to 250 and 1000 respectively.

---

[5]The code used to implement all the proposed methods by Net Intels is available in Github [37].

2. **K-Nearest Neighbor (KNN) regression:** For OB-SSs involving several AP-STA combinations, the dynamics of interrelations between entities of OBSS rely predominantly on the relative positioning of AP/STAs. To abstract such complexity in a cost-effective manner, KNN is selected, which is characterized by its simplicity, speed, and protection against high variance and bias. The KNN model is built using the Scikit Learn library in Python [39]. The inbuilt *KNearestRegressor* function is directly used, where neighbor number is fixed to 10. The algorithm for structuring the k-dimensional space of the data set (Ball Tree, KDTree, or Brute Force) is automatically selected based on input values.

3. **Random forest regression:** Motivated by the fact that the interrelationship between the features is non-linear, we propose dividing the data set dimensional space into smaller subspaces. To generalize the data and for better feature importance, an ensemble of trees forming a random forest is used. Random forest mechanisms are useful to reduce the spread and diversion of predictions. The proposed random forest regression is built using Scikit Learn, an ensemble module of the Sklearn library. The default number of trees was set to 100, which split is performed according to the mean squared error function. The maximum depth of the tree is set to 10.
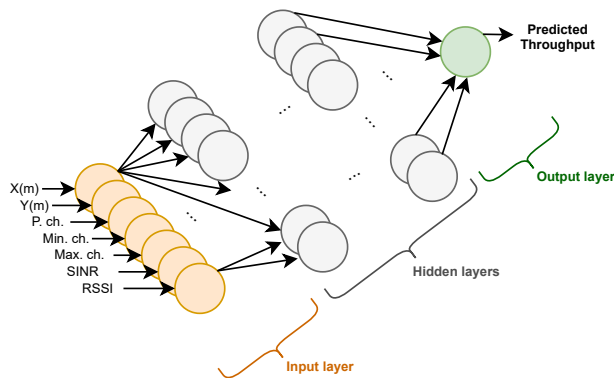

Fig. 5 – Net Intels' ANN architecture.

For all the proposed methods, we have first preprocessed the data set comprising six hundred different random deployments. In particular, static features such as the Contention Window (CW) were not included for training purposes. As for the rest of the features, we noticed a low correlation degree (see Fig. 6), so we have used all the features with higher variability from one simulation to another, including the node type (used when considering both APs and STAs during training), X and Y coordinates, primary channel, minimum and maximum channel allowed, SINR, and RSSI values.

The data was normalized before being fed into the regression models. Only data for STAs (stations) are considered for training and the throughput values of STAs are predicted using the models. The sum of the throughputs of
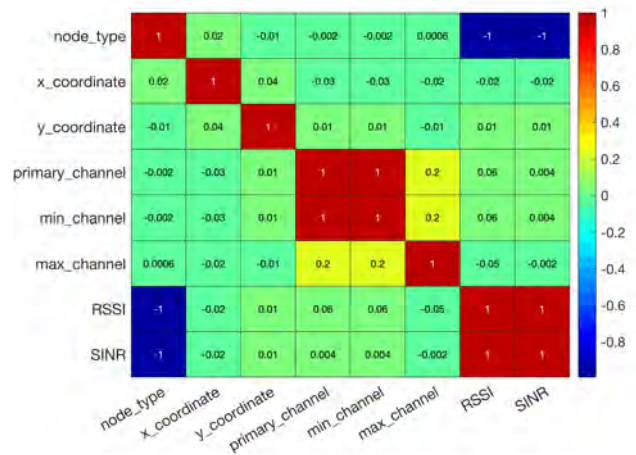

Fig. 6 – Correlation among input features.

STAs gives the throughput of the respective AP. For training purposes of all the three methods, the data is split (80% for training and 20% for validation).

4. PERFORMANCE EVALUATION

In this section, we show the results obtained by the participants' models presented in Section 3. In the context of the ITU AI for 5G Challenge, a test data set was released to assess the performance of each model, without revealing the actual throughput obtained through simulations. Participants were asked to predict the performance in Mbps of each BSS in the test scenarios.

The test data set consists of random deployments with different characteristics than the ones provided in the training data set, ranging from low to high density in terms of the number of BSSs and users. In total, test scenarios consist of 200 random deployments containing 1.400 BSSs and up to 8.431 STAs (randomly generated). To assess the participants' model accuracy, we focused on the throughput of the BSSs in each deployment (i.e., the throughput of each AP). Specifically, we used both the RMSE and the Mean Absolute Error (MAE) as reference performance metrics. Accordingly, Fig. 7 shows the MAE in Mbps obtained by each team in each type of test scenario.

As shown, for the aggregate BSS performance, most of the models offer low accuracy for the less dense scenarios (namely, *test1* and *test2*), whereas higher accuracy is achieved for the densest deployments (namely, *test3* and *test4*). The fact is that denser deployments are much more similar to the training scenarios than the sparser ones. As a result, models behave pessimistically in low-density deployments by assuming lower performance even if interference is low. As an exception, we find the model provided by Ramon Vallés, a feed-forward neural network with three blocks. The main difference of this model with respect to the others is that it separates the features related to signal quality and interference, and processes
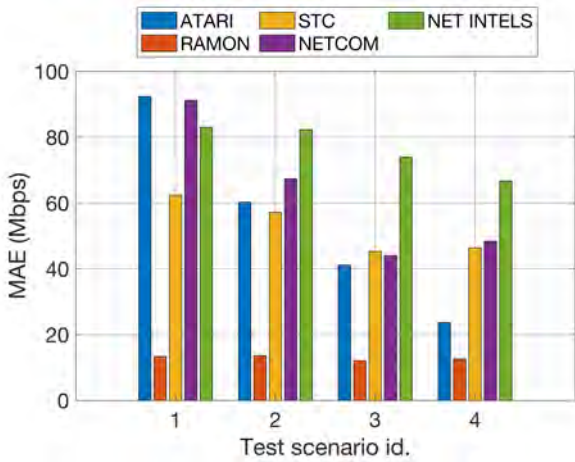
Fig. 7 – Mean absolute error obtained by each team, for each of the test scenarios of the data set. The aggregate throughput of all the BSSs is considered.

them apart from the rest. As a result, it is able to generalize well, even for new deployments with characteristics unseen in the training phase.

Although the prediction error is high for some test scenarios, it is important to remark that the performance of WLANs applying CB can be up to a few hundreds of Mbps (especially in sparse scenarios with low competition). To better illustrate the accuracy of the proposed models, we now show the prediction results obtained on a per-STA basis. Notice that the following results correspond to the solutions provided by three teams (*ATARI*, *STC*, and *Net Intels*), whose solution was based on predicting the throughput of STAs, and providing the aggregate performance afterward. Note, as well, that the target of the challenge was predicting the aggregate throughput in each BSS. In particular, Fig. 8 shows the histogram of the individual throughput predictions at STAs obtained across all the random test deployments.
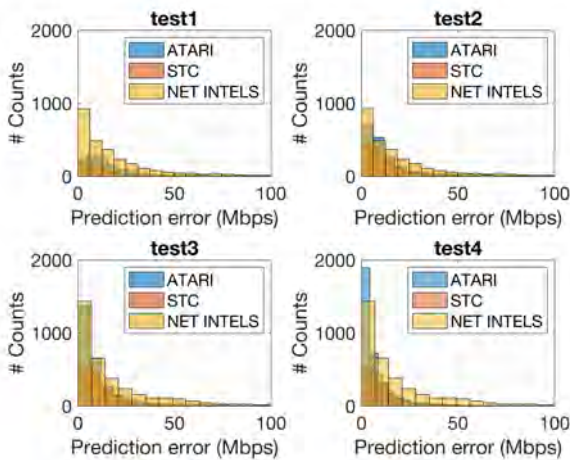

Fig. 8 – Histogram of the per-STA prediction error achieved by ATARI, STC, and NET INTELS.

As shown, the proposed ML mechanisms provide general accurate predictions; most of the error values are in the range of 0 to 10 Mbps. This means that, even in the presence of outliers, the predictions provided by the ML models are suitable for a significant percentage of the deployments. The accuracy of the different models proposed by ATARI, STC, and NET INTELS can be further observed in Table 4, which shows the percentage of the throughput predictions for STAs achieving an error below 10 Mbps.

Table 4 – Percentage of per-STA predictions achieving <10 Mbps error. Information is provided for ATARI, STC, and NET INTELS results.

	test1	test2	test3	test4
ATARI	36.97%	55.81%	67.01%	77.40%
STC	55.97%	56.27%	56.74%	60.67%
NET INTELS	38.09%	42.15%	44.01%	49.77%

For completeness, Fig. 9 shows the actual throughput achieved by STAs in all the test scenarios. As shown, the median is around 20 Mbps, but maximum values of up to 40 Mbps are also likely. Furthermore, several outliers were noticed, leading to up to 50 Mbps in some STAs.
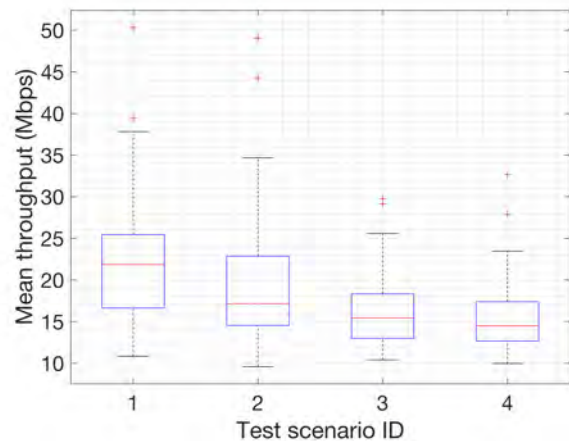

Fig. 9 – Boxplot of the mean throughput achieved by STAs for each test scenario.

Finally, to provide some insight on the computational needs required by the types of ML methods discussed in this paper, Table 5 contains the time required for training each model used by the team *Net Intels*, as well as the amount of computational resources employed. As shown, the training times are acceptable for providing near-real-time solutions.

Table 5 – Training time and computational resources used by the ML models proposed by *Net Intels*.

	Training time	RAM/GPU used
ANN	349 s	8 Gb RAM / 1.3 Gb GPU
Random forest	69 s	55 Gb RAM/ No GPU usage
KNN	122 s	5.3 Gb RAM/ No GPU usage

5. DISCUSSION

5.1 Contributions

With contributions from participants around the globe, the ITU AI for 5G Challenge has unprecedentedly established a platform for addressing important problems in communications through ML. As for the performance prediction problem in CB WLANs (referred to as PS-013), the challenge has allowed us to glimpse the potential of ML models for addressing it.

This paper provides a compendium of ML models proposed for throughput prediction in WLANs, including popular models such as neural networks, linear regression, or random forests. In particular, we have provided an overview of the proposed models and analyzed their performance in the context of the challenge. By opening the data set used during the competition, we encourage the development of mechanisms that improve the baseline performance shown by the ML models presented in this work.

5.2 Lessons learned

From the performance evaluation done in this paper, we have drawn the following conclusions:

1. First, even if the data set was not particularly big,[6] some of the proposed ML models achieved good results. This is quite a positive result since it opens the door to ML models that can be (re)trained fast, thus becoming suitable for (near)real-time solutions.

2. Second, most of the proposed DL-based models have shown higher accuracy for the denser and more complex deployments (which more closely match the training scenarios) than for the sparser ones. While capturing complex situations is quite a positive result, the pitfalls observed in simpler deployments also suggest that out-of-the-box DL methods may fail at capturing the relationship between interference and performance of WLANs. In this regard, well-known models characterizing WLANs (e.g., SINR-based models [40]) can potentially be incorporated into the ML operation for the sake of improving accuracy, thus leading to hybrid model-based and data-driven mechanisms.

3. Third, and related to the previous point, GNNs have been shown to be particularly useful to capture the complex interactions among devices in WLANs, both in terms of interference and neighboring activity. In particular, we have realized the importance of pre-processing the data set in order to obtain accurate prediction results. Deriving information specific to the problem (i.e., signal quality, interference) has turned out to be essential for the sake of generalization.

4. Finally, we remark the importance of cost-effectively predicting the performance in WLANs, which may open the door to novel mechanisms using these predictions as heuristics for online optimization. The incorporation of these kinds of models to WLANs is expected to be enabled by ML-aware architectural solutions [41].

5.3 The role of network simulators in ML-aware communications

The availability of data for training is key for the success of ML application to future 5G/6G networks. Given the current limitations in acquiring data from real networks, simulators emerge as a practical solution to generate complementary synthetic data for training ML models. The fact is that data may be scarce because, among other reasons, measurement campaigns are costly, data from networks involves privacy concerns, or data tenants are not willing to share their data.

Through the problem statement discussed in this paper, we have contributed to showcase how synthetic data can be used to train ML models for networks. A notorious advantage is that network simulators allow characterizing complex deployments, sometimes representing unknown situations, so they help to train and validate ML models.

Beyond generating data for training, network simulators are envisioned to serve as secure platforms for testing, training, and evaluating ML models before being applied to operative networks [42]. In consequence, we foresee the adoption of simulators into future ML-aware networks as a key milestone for enhancing both reliability and trustworthiness in ML mechanisms.

REFERENCES

[1] ITU-T, Recommendation Y.3172 (2019). "Architectural framework for machine learning in future networks including IMT-2020".

---

[6]Youtube-8M Data set (http://research.google.com/youtube8m/) consists of 350.000 hours of video, while only six hundred different random deployments have been used in this paper for training purposes.

[2] ITU-T. ITU AI/ML in 5G Challenge (2019). Available at: https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx

[3] Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Neu, G. (2019). "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs". Journal of Network and Computer Applications, 127, 26-42.

[4] Francesc Wilhelmi. (2020). [ITU-T AI Challenge] Input/Output of project "Improving the capacity of IEEE 802.11 WLANs through Machine Learning" [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4106127

[5] Bellalta, B. (2016). IEEE 802.11 ax: "High-efficiency WLANs". IEEE Wireless Communications, 23(1), 38-46.

[6] López-Pérez, D., Garcia-Rodriguez, A., Galati-Giordano, L., Kasslin, M., & Doppler, K. (2019). "IEEE 802.11 be extremely high throughput: The next generation of Wi-Fi technology beyond 802.11 ax". IEEE Communications Magazine, 57(9), 113-119.

[7] Barrachina-Muñoz, S., Wilhelmi, F., & Bellalta, B. (2019). "Dynamic channel bonding in spatially distributed high-density WLANs". IEEE Transactions on Mobile Computing.

[8] Barrachina-Muñoz, S., Wilhelmi, F., & Bellalta, B. (2019). "To overlap or not to overlap: Enabling channel bonding in high-density WLANs". Computer Networks, 152, 40-53.

[9] Cao, R., & Zhang, H. (2019). U.S. Patent Application No. 16/435,899.

[10] Wang, W., Chen, Y., Wang, Z., Zhang, J., Wu, K., & Zhang, Q. (2016). "Wideband spectrum adaptation without coordination". IEEE Transactions on Mobile Computing, 16(1), 243-256.

[11] Huang, P., Yang, X., & Xiao, L. (2016). "Dynamic channel bonding: Enabling flexible spectrum aggregation". IEEE Transactions on Mobile Computing, 15(12), 3042-3056.

[12] Chen, Y. D., Wu, D. R., Sung, T. C., & Shih, K. P. (2018, April). "DBS: A dynamic bandwidth selection MAC protocol for channel bonding in IEEE 802.11 ac WLANs". In 2018 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.

[13] Nabil, A., Abdel-Rahman, M. J., & MacKenzie, A. B. (2017, October). "Adaptive channel bonding in wireless LANs under demand uncertainty". In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) (pp. 1-7). IEEE.

[14] Qi, H., Huang, H., Hu, Z., Wen, X., & Lu, Z. (2020). "On-demand channel bonding in heterogeneous WLANs: A multi-agent deep reinforcement learning approach". Sensors, 20(10), 2789.

[15] Pan, C., Cheng, Y., Yang, Z., & Zhang, Y. (2018, July). "Dynamic opportunistic spectrum access with channel bonding in mesh networks: A game-theoretic approach". In International Conference on Machine Learning and Intelligent Communications (pp. 381-390). Springer, Cham.

[16] Bianchi, G. (2000). "Performance analysis of the IEEE 802. 11 distributed coordination function". IEEE Journal on selected areas in communications, 18(3), 535-547.

[17] Bellalta, B., Zocca, A., Cano, C., Checco, A., Barcelo, J., & Vinel, A. (2014). "Throughput analysis in CSMA/CA networks using continuous time Markov networks: A tutorial". Wireless Networking for Moving Objects, 115-133.

[18] Baccelli, F., & Błaszczyszyn, B. (2009). "Stochastic geometry and wireless networks (Vol. 1)". Now Publishers Inc.

[19] Feng, H., Shu, Y., Wang, S., & Ma, M. (2006, June). "SVM-based models for predicting WLAN traffic". In 2006 IEEE International Conference on Communications (Vol. 2, pp. 597-602). IEEE.

[20] Abbas, M., Elhamshary, M., Rizk, H., Torki, M., & Youssef, M. (2019, March). "WiDeep: WiFi-based accurate and robust indoor localization system using deep learning". In 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom (pp. 1-10). IEEE.

[21] Davaslioglu, K., Soltani, S., Erpek, T., & Sagduyu, Y. (2019). "DeepWiFi: Cognitive WiFi with deep learning". IEEE Transactions on Mobile Computing.

[22] Khan, M. A., Hamila, R., Al-Emadi, N. A., Kiranyaz, S., & Gabbouj, M. (2020). "Real-time throughput prediction for cognitive Wi-Fi networks". Journal of Network and Computer Applications, 150, 102499.

[23] Barrachina-Muñoz, S., Wilhelmi, F., Selinis, I., & Bellalta, B. (2019, April). "Komondor: a wireless network simulator for next-generation high-density WLANs". In 2019 Wireless Days (WD) (pp. 1-8). IEEE.

[24] Wilhelmi, F., Barrachina-Muñoz, S., Cano, C., Selinis, I., & Bellalta, B. (2021). "Spatial reuse in IEEE 802.11 ax WLANs". Computer Communications.

[25] ITU-T, ML5G-I-237 (2020). "A compilation of problem statements and resources for ITU Global Challenge on AI/ML in 5G networks". Available online: https://www.itu.int/en/ITU-T/AI/challenge/2020/Documents/ML5G-I-237-R5_v10.docx

[26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The graph neural network model," IEEE Transactions on Neural Networks, pp. 61–80, 2008.

[27] P. Battaglia, J. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro and R. Faulkner, "Relational inductive biases, deep learning, and graph networks". arXiv preprint arXiv:1806.01261, 2018.

[28] M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[29] P. Soto, D. Góez, M. Camelo, N. Gaviria and K. Mets, "ITU AI/ML in 5G Challenge 2020: Team ATARI (Github repository)", 12 Nov 2020. Online: https://github.com/ITU-AI-ML-in-5G-Challenge/ITU-ML5G-PS-013-ATARI. [Accessed 20 Feb 2021].

[30] Murtagh, F. (1991). "Multilayer perceptrons for classification and regression". Neurocomputing, 2(5-6), 183-197.

[31] R. Vallés (2020), "MLP Throughput prediction (Github repository)". Online: https://github.com/VPRamon/MLP-Throughput-prediction. [Accessed 20 Feb 2021].

[32] Chen, T., & Guestrin, C. (2016, August). "Xgboost: A scalable tree boosting system". In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[33] Yeo, I-K, and R Johnson. 2000. "A New Family of Power Transformations to Improve Normality or Symmetry." Biometrika 87 (4): 954–59.

[34] Kuhn M, Wickham H (2020). "Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles". Available online: https: //www.tidymodels.org.

[35] M Alfaifi, A Algunayah, A Aloshan, M Abid & K Sahari (2020). "ITU 2020: Problem 13 - STC Team 2". Github repository. Available online: https://github.com/ITU-AI-ML-in-5G-Challenge/p13_stc_team2. [Accessed on 21 February 2021]

[36] Jorge Martin Pérez & Luigi Girletti (2020). "ITU-ML5G-PS-013", Github repository. Available online: https://github.com/MartinPJorge/ITU-ML5G-PS-013. [Accessed on 21 February 2021]

[37] Rajasekar Mohan, K Venkat Ramnan, Megha G Kulkarni, and Vishalsagar Udupi (2020), "Net Intels solution for the ITU-T AI Challenge (Github repository)". Online: https://github.com/venkatramnank/UPF_ITU_T_5G_ML_AI_Challenge_NetIntels. [Accessed 20 Feb 2021].

[38] Brownlee, J. (2016). "Deep learning with Python: develop deep learning models on Theano and Tensor-Flow using Keras". Machine Learning Mastery.

[39] Pedregosa, F., et al. (2011). "Scikit-learn: Machine learning in Python". The Journal of Machine Learning Research, 12, 2825-2830.

[40] Yang, X., & Vaidya, N. (2005, March). "On physical carrier sensing in wireless ad hoc networks". In Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. (Vol. 4, pp. 2525-2535). IEEE.

[41] Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). "A flexible machine-learning-aware architecture for future WLANs". IEEE Communications Magazine, 58(3), 25-31.

[42] Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ram, V., & Bellalta, B. (2020). "Usage of Network Simulators in Machine-Learning-Assisted 5G/6G Networks". IEEE Wireless Communications Magazine, 2021.

AUTHORS
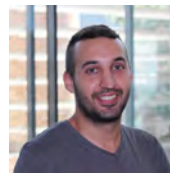
Francesc Wilhelmi holds a Ph.D. in information and communication technologies (2020) from Universitat Pompeu Fabra (UPF). He is currently a postdoctoral researcher in the Mobile Networks department at Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and a teaching assistant at UPF and at Universitat Oberta de Catalunya (UOC).

David Góez is a telecommunications engineer from the ITM Metropolitan Technological Institute, a Master in Automation and Industrial Control from the ITM, a PhD student in electronic and computer engineering at the University of Antioquia. His research focuses on radio-communication systems whose signal processing blocks can be built with machine learning.

Paola Soto is currently pursuing her Ph.D. degree at University of Antwerp - imec, Belgium. Her doctoral research investigates machine learning techniques applied to network management. She received a B.Sc. degree in Electronics and an M.Sc. degree in telecommunications engineering from the University of Antioquia, Medellín, Colombia, in 2014 and 2018, respectively. Her main research interests include artificial intelligence, machine learning, network management, and resource allocation algorithms.

Ramon Vallés is currently pursuing a B.Sc. in computer science at Universitat Pompeu Fabra (UPF). His research interests are artificial intelligence and machine learning, networks and network security.

Mohammad Alfaifi holds an M.Sc. degree in management studies from the University of Waikato (New Zealand). He is currently working as Data Mining Supervisor at Saudi Telecom (STC).

Abdulrahman Algunayah holds a B.Sc. degree in electrical engineering from King Saud University (KSU). He is an infrastructure design engineer at Saudi Telecom (STC).
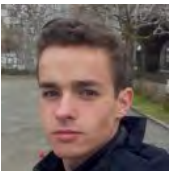
Jorge Martín-Pérez is a telematics engineering Ph.D. student at Universidad Carlos III de Madrid (UC3M). He obtained a B.Sc. in mathematics and computer science, both in 2016 from Universidad Autónoma de Madrid. In 2017, he obtained an M.Sc. in telematics engineering from UC3M. His research focuses on optimal resource allocation in 5G networks, and since 2016 he participates in EU-funded research projects.

Luigi Girletti is a telematics engineering Ph.D. student at Universidad Carlos III de Madrid (UC3M). He obtained both a B.Sc. and an M.Sc in computer science engineering in 2014 and 2018 from Universitá degli Studi Federico II di Napoli. His research focuses on SDN/NFV, artificial intelligence applied to next-generation networks, and 5G-based EU-funded research projects.

Rajasekar Mohan, alumnus of Indian Institute of Technology, Madras, is pursuing his research in the domain of wireless communication at PES University. His research interests include embedded systems, IoT, robotics and application of ML in communication. He has served in the Indian Air Force for over 23 years in various technology roles in the field of communication. Currently, he is an associate professor in the department of ECE at PES University, Bangalore.

K Venkat Ramnan is an undergraduate student in the Electronics and Communication Department at PES University, India. He is a beginner in machine learning, deep learning and image processing. He also works with the Internet of things, networking and Linux. He aims to apply machine learning and deep learning in various fields including wireless communications, healthcare and various other domains of electronics and communications.
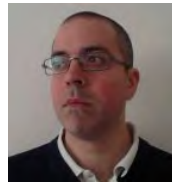
Boris Bellalta is an associate professor in the Department of Information and Communication Technologies (DTIC) at Universitat Pompeu Fabra (UPF). He is the head of the Wireless Networking research group at DTIC/UPF. His research interests are in the area of wireless networks, with emphasis on the design and performance evaluation of new architectures and protocols.