

aiOS: An Intelligence Layer for SD-WLANs

Estefanía Coronado*, Abin Thomas*, Suzan Bayhan[‡], and Roberto Riggio*

*Smart Networks and Services, Fondazione Bruno Kessler, Trento, Italy

Email: {e.coronado, athomas, rriggio}@fbk.eu

[‡]Design and Analysis of Communication Systems, University of Twente, Enschede, Netherlands

Email: s.bayhan@utwente.nl

Abstract—Software-Defined Networking promises to deliver a more manageable network whose behaviour could be easily changed using applications written in high-level declarative languages running on top of a logically centralized control plane resulting, on the one hand, in the mushrooming of complex point solutions to very specific problems and, on the other hand, in the creation of a multitude of network configuration options. This fact is especially true for 802.11-based Software-Defined WLANs (SD-WLANs). It is our standpoint that to tame this increase in complexity, future SD-WLANs must follow an Artificial Intelligence (AI) native approach. In this paper we present *aiOS*, an AI-based Operating System for SD-WLANs. Then, we use *aiOS* to implement several Machine Learning (ML) models for user-adaptive frame length selection in SD-WLANs. An extensive performance evaluation carried out on a real-world testbed shows that this approach improves the aggregated network throughput by up to 55%. Finally, we release the entire implementation including the controller, the ML models, and the programmable data-path under a permissive license for academic use.

Index Terms—Machine Learning, Software-Defined Networking, WLANs, IEEE 802.11, frame length selection, aggregation

I. INTRODUCTION

It is no doubt that Wi-Fi networks are one of the pillars of today's communications. With their pivotal role in our lives, the need for higher efficiency is essential. While recent amendments to the 802.11 standard, such as IEEE 802.11ax [1], increase the Physical Layer (PHY) rates using solutions like Multiple-Input Multiple-Output (MIMO), the achieved throughput is far from the theoretical capacity. In fact, with higher PHY rates, the transmission time is reduced, leading to huge inefficiencies since enhancements on top of the Medium Access Control (MAC) Layer are limited by fixed overheads, i.e., channel access and encapsulation [2].

The IEEE 802.11 standard has received several improvements across its lifetime, including the Enhanced DCF Channel Access (EDCA) as a Quality-of-Service (QoS) aware extension to the original Distributed Coordination Function (DCF), the support for frame aggregation, block acknowledgements, etc. Every revision of the standard, however, also introduced a new set of knobs into the 802.11 machinery. Each of these knobs (or combination of thereof) has been the focus of a humbling amount of scientific studies. Recently, Software-Defined Networking (SDN) and Network Function Virtualization (NFV) have attempted to move computer networks into the modern era by introducing several levels of softwarization and by separating mechanisms (the knobs) from

policies (how the knobs are turned), and by putting the latter into the hands of the so-called *network programmer*.

The promises of SDN and NFV go in the direction of delivering a much simpler network whose behaviour could be easily modified and adapted. This change of perspective resulted, on the one hand, in the mushrooming of convoluted solutions to very specific problems and, on the other hand, in the creation of a multitude of network configuration options. The expected complexity of future Wireless Local Area Networks (WLANs), and by extension also of 6G networks, is set to make such an approach impractical. Conversely, recent advances in Artificial Intelligence (AI), such as reinforcement learning and deep neural networks, are set to play an important role in the control and management of current and future wireless networks.

Future wireless networks, including WLANs, must follow an AI-native approach towards autonomous management, and become smart, agile, and able to *learn* from and *adapt* to the changing environment. If this transition from network *softwarization* to network *brainitisation* is to take place, AI cannot be treated as an afterthought but instead must be accounted for from the requirements phase. Similarly, each subsystem composing future wireless networks cannot be expected to employ distinct and separated AI tools and datasets. That approach would lead to AI-silos, thus preventing progress in one domain to be shared and leveraged for other aspects of network control and management.

In order to deal with these limitations, our contributions in this paper are threefold:

- We take a first step towards network *brainitisation* by introducing an AI-based Operating System for SD-WLANs. This Operating System, named *aiOS*, embeds state-of-the-art Machine Learning (ML) toolboxes with the aim of providing a global intelligence platform, which is at the same time driven by AI and designed to drive future AI-powered applications and services.
- We present a proof-of-concept implementation of *aiOS* and we validate it by implementing several low-complexity ML models for adaptive frame length selection in 802.11-based SD-WLANs. An extensive performance assessment carried out on a real-world testbed has shown that our approach can improve the aggregated network throughput by up to 55% over the standard A-MSDU aggregation with constant frame length.
- We share with the community the entire implementation including the controller, the ML models, and the programmable data-path.

The rest of the paper is outlined as follows. Section II discusses the related work. The problem statement and the system model are provided in Sec. III. The *aiOS* system architecture is introduced in Sec. IV. The ML models are presented in Sec. V. Section VI reports on the performance evaluation. Finally, Sec. VII draws the conclusions pointing out the future work.

II. RELATED WORK

A. Artificial Intelligence in Wireless Networks

Despite 5G is still in its early stages of deployment, there is already several attempts at sketching the roadmap of future 6G systems [3], [4]. While it is too early to clearly identify the characteristics of 6G and beyond networks, it is widely agreed that AI will play a pivotal role by providing the foundation upon which new services and applications will be built.

The application of ML has already been proved in tasks such as image and voice recognition. These and similar success stories motivated the use of ML techniques to address the challenges in networking and, in particular, in wireless communications [5]. This includes, for example, resource management at the MAC layer, mobility management at the network layer, and localization at the application layer. Several works can also be found on ML for securing WLANs [6]. Conversely, deep learning uses multi-layer neural networks to perform accurate pattern recognition. In the case of wireless networks it can be used to discover network dynamics (such as hotspots) starting from the analysis of a large number of network parameters [7], [8]. Similarly, cognitive networks leveraged AI concepts to implement optimal resource usage and management at the physical layer [9].

The expected complexity of future wireless networks is set to make current network optimization approaches based on analytical models and on system-level simulations impractical. Likewise, while early works on SDN did attempt to provide *network programmers* with powerful abstractions to control their networks [10], [11], [12], they eventually fell short of providing a practical platform that can leverage, often low-level, primitives to implement complex optimization tasks. As opposed to the current efforts on SDN, which did not deliver anything fundamentally new, but rather proposed a different way of arranging network functionalities, the goal of *aiOS* is to provide a coherent, practical, and data-driven AI platform for SD-WLANs. It is our standpoint that such an approach is pivotal to enable re-utilization of best practises in AI within the networking domain.

B. Frame Length Selection in WLANs

Frame length selection in 802.11 networks has been so far widely investigated. Literature in this respect mainly relies on frame aggregation mechanisms defined in the standard, namely A-MSDU, A-MPDU, and a combination of the two.

Works on A-MSDU aggregation are mostly focused on QoS, real-time traffic, and small-sized frames. Maqhat et al. [13] propose a scheduler for delay-sensitive traffic. In their proposal, control bits are separately adjusted for each sub-frame to enable faster retransmissions. This work is later implemented

using NS-2 [14]. Similarly, in [15] the authors pursue improvements for error-prone channels by adding control bits to every subframe to enable per-subframe retransmissions. The work in [16] deals with adaptive frame size estimation based on Extended Kalman Filter for saturated networks. Saldana et al. [17] discuss the trade-off between throughput and latency of frame aggregation in a specific scenario accounting for mobile users. In [18], a dynamic scheme is introduced to calculate the optimal size based on the traffic class of the packets. This study shows a trade-off between throughput and delay caused by the time taken to form an aggregated frame.

Regarding A-MPDU mechanisms, the scheme introduced in [19] aims at dynamically adapting the A-MPDU length by observing how the mobility of the stations affects the quality of the channel. Likewise, the work in [20] uses a Proportional Integral Derivative controller to appropriately select the A-MPDU aggregation size based on QoS indicators. Complementary to [20], the objective of [21] is to find the optimal number of MPDUs based on delay requirements for 802.11ac WLANs. In particular, via simulation, the authors seek throughput improvements while satisfying delay requirements using RTS/CTS. In contrast to the previous works, the authors of [22] propose a QoS-aware adaptive A-MPDU aggregation scheduler for voice traffic. However, this approach is non-standard-compliant.

The combination of A-MSDU and A-MPDU aggregation is adopted by Kim et al. for achieving airtime fairness and improving overall network throughput [23]. The work in [24] studies the performance of A-MSDU and A-MPDU mechanisms in NS-2 under error-prone channel conditions. The authors propose an optimal frame aggregation scheme based on the results obtained from an analysis studying the relationship between frame length and Bit Error Rate (BER). Similar strategies are also applied in vehicular networks [25], [26]. Li et al. propose in [2] a scheme with fragment retransmission where multiple packets are aggregated and transmitted as a single frame. Instead of using MSDU or MPDU aggregation, this work discusses an algorithm where a fragmentation threshold is set and any packet longer than this threshold is fragmented before the aggregation process begins. The model is evaluated using NS-2 for TCP, HDTV and VoIP.

Kriara et al. [27] study the effect of PHY rate and frame aggregation on the performance of a 802.11 network, pointing out their higher relevance in comparison to other factors. In line with this, in [28] the authors deal with rate and frame size adaptation using A-MPDU aggregation. The network conditions are modelled in NS-2 using different BER values. Similarly, the work in [29] performs rate adaptation, frame aggregation, and MIMO mode selection based on Channel State Information (CSI) focusing on A-MPDU aggregation. Finally, authors in [30] propose a joint PHY-MAC link adaptation strategy with theoretical link quality analysis together with A-MSDU aggregation in error-prone channel conditions.

Despite the abundance of solutions for frame length adaptation to channel conditions, we observe that most of them overlook the question about how diverse factors determining such channel conditions affect each other. In this context, many

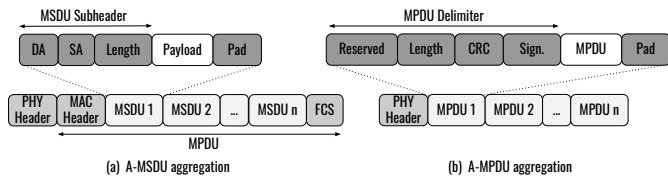


Fig. 1. Schematic diagram of A-MSDU and A-MPDU aggregation formats.

works have proved the greater role of the PHY rate over others when selecting a user specific frame length [30], [31]. In [30] the authors jointly select PHY rate and A-MSDU length. Link characteristics are estimated using an analytical model and are used to compute the optimal A-MSDU length and rate. The key assumption is that the network is saturated. Similarly, in [31] the signal strength of the ACKs is used to jointly set frame size and transmission rate.

Different than prior work, our solution relies on ML to compute user-specific frame length through A-MSDU aggregation based on the transmission rate. To set the parameters in a timely fashion, we adhere to a supervised learning approach in which a Software-Defined controller collects link state and packet delivery statistics from the rate control algorithm to estimate the best frame length for each user. Our ML-based approach obviates the need to derive the relation between Modulation and Coding Scheme (MCS) and frame size, which is hard without simplifying assumptions and perfect channel state information, making our solution suitable in realistic environments. We implement and validate our proposed approach on a real-world testbed under various network conditions.

III. PROBLEM STATEMENT

In this section we provide the technical background on frame aggregation and rate adaptation in 802.11. Then, we discuss in various contexts how frame length and MCS influence delivery probability and the need to find such a relationship. Finally, we introduce the system model and formulate our frame length adaptation problem in SD-WLANs.

A. Frame Aggregation in IEEE 802.11

IEEE 802.11 defines three levels of frame aggregation in the MAC layer, namely A-MSDU, A-MPDU and a combination of the two methods, as depicted in Fig. 1, to reduce the overhead caused by channel access, headers, and preambles [32], [33].

A-MSDU aggregation seeks higher efficiency by combining multiple MSDUs within a single PHY and MAC header, which is especially suitable for small payloads. Nevertheless, the subframe header of the MSDUs must have as Destination Address (DA) and Sender Address (SA) the same as Receiver Address (RA) and Transmitter Address (TA) in the MAC header to be aggregated. The A-MSDU is complete when the packet length reaches the maximum aggregation size (limited to 3839 or 7935 bytes) or if the oldest frame delay reaches a threshold. However, given that a unique Frame Check Sequence (FCS) is generated for the entire packet, A-MSDUs are vulnerable under error-prone channel conditions.

TABLE I
MINSTREL RETRY CHAIN CONFIGURATION.

Rate	Look-around		Normal transmission
	Random < Best	Random > Best	
r_0	Best rate	Random rate	Best rate
r_1	Random rate	Best rate	Second best rate
r_2	Best probability	Best probability	Best probability
r_3	Base rate	Base rate	Base rate

A-MPDU aggregation includes a single PHY header and aggregates multiple A-MSDUs or MPDUs encapsulating their own MAC headers. Contrary to A-MSDU aggregation, FCS is present in each subframe, which allows retransmitting only the affected MPDU in case of error. A-MPDU aggregation is solely dependent on the number of packets already in the queue. The maximum aggregation length is limited to 65536 bytes. All this points to the fact that A-MSDU is more suitable for smaller frames.

B. Rate Adaptation in IEEE 802.11

Among the rate adaptation algorithms, such as Onoe [34] and ARF [35], Minstrel [36] is one of the most advanced and widely used due to its implementation in the MadWifi driver and its ability to work even under noisy and/or fast faded environments. Due to these reasons, Minstrel is taken as a reference for our research. However, any other rate control method could be used instead. Minstrel is based on a retry chain composed of four rate-count pairs namely r_0/c_0 , r_1/c_1 , r_2/c_2 and r_3/c_3 . If a frame is successfully transmitted, the remaining part of the chain is ignored. Otherwise, the next pair is used until the frame is transmitted or finally dropped.

The ratio of transmission attempts to acknowledgments received is calculated using Exponential Weighted Moving Average (EWMA) to smoothen the probability estimation. Such a ratio is then stored for each rate, r , as shown in Table I. Minstrel uses the link delivery statistics to configure the retry chain during 90% of the time, while in the remaining 10%, a random rate is used to gather new statistics.

C. Problem Analysis

To understand how the optimal frame length changes according to MCS and channel conditions we have performed a set of simulations using NS-3. More precisely, we aim to address the following question: *what is the $\langle MCS, framelength \rangle$ combination that, given certain channel conditions, results in the highest goodput?*

The reference scenario consisted of a Wi-Fi AP and a Wi-Fi station, which has been positioned at diverse distances from the AP at each run of the simulations. The transmission power of the AP has been set to 18 dBm and the channel between the AP and the client was an ITU-indoor channel for an office setting [37]. For each distance in the coverage area of the AP, $2R$, we have used a specific MCS and we have measured the goodput at the station using different frame lengths.

In Fig. 2a we plot the goodput for an increasing frame length for three MCS values. We plot on purpose the goodput for different distances between AP and station to highlight how

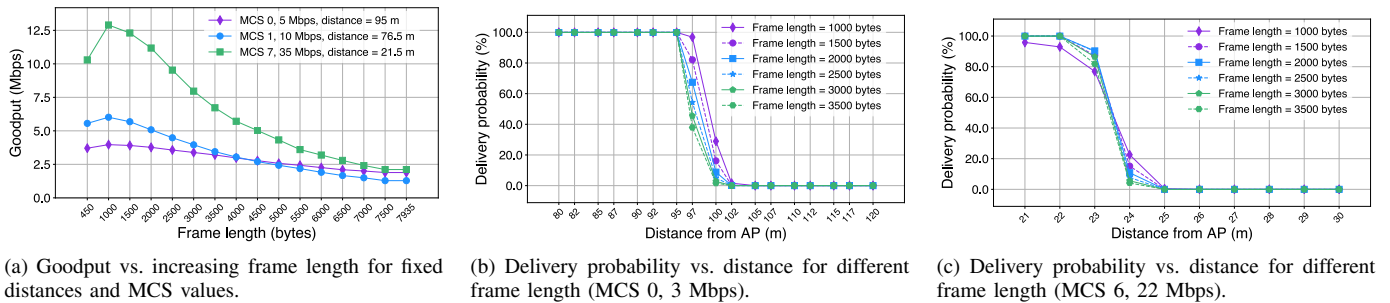


Fig. 2. Performance comparison for a single downlink stream according to different frames, distances and MCS values.

stations with different channel conditions do have a frame length that maximises the goodput at the receiver. Notice however how the particular peak of the curve could change according to both distance *and* MCS. To develop further insights, we also plot the change in the delivery probability (i.e., the percentage of packets successfully decoded) with increasing distance from the AP and for various frame lengths. Figure 2b depicts the results for a robust modulation and coding scheme (MCS 0) whereas Fig. 2c presents the results for a higher MCS (MCS 6). Both figures show that shorter packets with lower number of MSDUs sustain a longer communication range compared to the longer ones. Comparing Fig. 2b and Fig. 2c, we observe that higher MCS values, despite achieving higher goodput, as shown in Fig. 2a, are characterized by a smaller coverage (as expected). This confirms once more that the optimal aggregation length is not always the maximum supported by the link layer and thus the need for a more advanced strategy to effectively select the frame length.

D. System Model

The system model considered in this work is based on a Wi-Fi network, as in Fig.3, comprising M WiFi APs, N Wi-Fi stations (also referred to as clients), and a Software-Defined Radio Access Network (SD-RAN) controller. We assume both uplink and downlink traffic. Moreover, clients may have diverse traffic activity evolving over time. The SD-RAN controller collects network state information regularly from the APs. This information includes network-wide statistics, e.g., channel utilization, and per-station statistics, e.g., rate control statistics. Based on this information, the ML models presented in Sec. V compute the frame length for each AP/client pair and communicate it to the SD-RAN controller, who applies the new directives on the network elements.

Note that configuring the frame aggregation parameters at this granularity is essential as clients might have very diverse traffic patterns and link qualities. This is also a feature enabled by our SDN approach. As a matter of fact, in traditional Wi-Fi networks the maximum aggregation length is static and defined either at the AP or at the traffic class level. We assume that the frame length optimization is implemented only in the downlink direction as it does not require changes to the clients nor to the standard. On the contrary, while possible, uplink aggregation length optimization would require modifications to the Wi-Fi stations hence making it non standard-compliant.

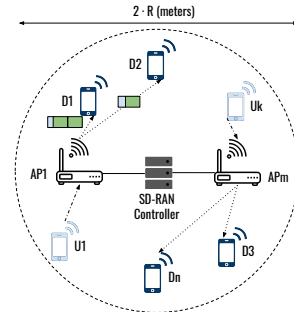


Fig. 3. System model considered for the ML-based frame length selection.

The SD-RAN controller uses the ML models presented in Sec. V to set the optimal downlink frame length for a given MCS. The reason for this is that, in Sec. III-C we claimed that the distance/Signal-To-Noise Ratio (SNR) clearly affects the selected MCS and the other way around. The output of the rate control algorithm (i.e., Minstrel in this work) is actually based on the delivery probability, which includes the state of the network and accounts for the distance. Therefore, given its more than proved effectiveness, we have decided to accept the MCS that the rate control algorithm chooses, and provide it as input for the ML-based frame length adaptation models.

IV. aiOS SYSTEM ARCHITECTURE

Figure 4 illustrates the high-level *aiOS* system architecture. Note that *aiOS* is 802.11 standard-compliant and does not require modifications to the 802.11 MAC/PHY layers nor to the clients. Based on SDN principles, where control and forwarding planes are decoupled, the architecture is divided into infrastructure, control and application layers.

A. Control Layer

It is based on a modular architecture, where the SD-RAN controller is in charge of building the global view of the network and of issuing management policies to the devices at the infrastructure layer. The SD-RAN controller defines a Python Application Programming Interface (API) that provides a set of programming abstractions to specify network directives while sheltering the *network programmer* from the complexities of the underlying wireless technology. However, once defined, such directives are unable to change by themselves, or even to be created from the network state since the system lacks

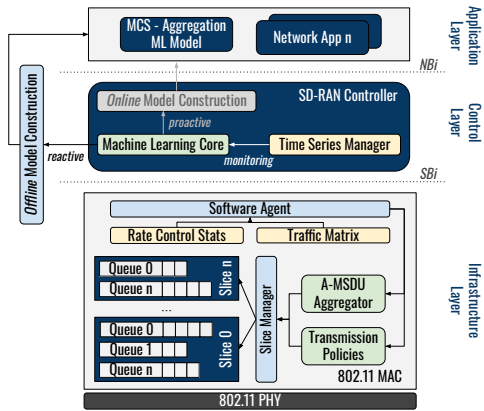


Fig. 4. High-level view of the system architecture.

this *ability to reason*. Building on this, as shown in Fig. 4, we introduce a *Machine Learning Core* that is able to drive and be driven by the changing network dynamics.

Strategically located, the *Machine Learning Core* supervises the network state information handled by the SD-RAN controller and filters it using the *Time Series Manager* module. The *Machine Learning Core* can act in two manners: proactively or reactively. In the former mode, depending on the nature of the data, which could be for example labeled or unlabeled, complete or incomplete due to partial network synchronization, real-time or cumulative, the *Machine Learning Core* is able to proactively select the most relevant features in a dataset, and propose and push to the application layer a new learning-based policy. If required, modifications can be performed afterwards. In addition, *network programmers* can reactively leverage this layer to build a model of interest, and deploy it as a network app at the application layer. It is worth stressing that in this work we focus on how specific networking problems can be tackled in the *reactive* mode, leaving the *proactive* capabilities as a future work. Conversely, the *Time Series Manager* is responsible for merging, cleaning and filtering network statistics from different sources (e.g., diverse network devices and performance metrics) and unifying them into a homogeneous data structure.

The offline model construction (i.e., reactive approach) is left to the choice of the implementer. The output provided by the *Machine Learning Core* is available as a Comma-Separated Values (CSV) file that can be processed by any Python-based ML framework. Some examples include Sklearn and TensorFlow. After the model is built, it can be easily loaded as a network application at the application layer.

B. Infrastructure Layer

It is composed of several independent modules in charge of applying the directives issued by the SD-RAN controller.

- **Software Agent.** It is responsible for the communication with the control layer via the southbound interface and for implementing the policies from the SD-RAN controller. Furthermore, it collects information about the network

state, including PHY/MAC statistics (e.g., rate control and CSI), and reports it to the SD-RAN controller.

- **Slice Manager.** It is responsible for partitioning radio resources into logical networks or slices attending, for example, to traffic types. Slices are identified by the tuple $\langle SSID, Slice ID \rangle$, where *SSID* is the network name and *Slice ID* is the portion of the flowspace of the incoming traffic that must be mapped to the slice. Slices are characterized by a set of parameters, including EDCA parameters, aggregation type (e.g., A-MSDU), and the fraction of airtime that can be assigned to the slice.¹
- **Transmission Policies.** They specify the parameters the APs can use for the communication with a wireless client. Such parameters include, the MCS values that can be used by the rate selection algorithm², the RTS/CTS threshold, and the multicast strategy. *Transmission Policies* are specified on a L2 destination address basis. As a result, for each destination address and for each network slice, a specific transmission policy can be created.
- **A-MSDU Aggregator.** It is responsible for assembling and encapsulating the A-MSDUs. Each slice contains m traffic queues identified by the tuple $\langle src, dst \rangle$, where *src* and *dst* are, respectively, the MAC source and destination addresses. A-MSDUs are generated from each of these queues according to the maximum length specified by the SD-RAN controller. It is noteworthy that, since incoming packets are classified by the tuple $\langle src, dst \rangle$, expensive search and post-processing is not required, thus reducing the computational complexity of the frame aggregation subsystem to $O(1)$.

C. Application Layer

It is made up of network applications which, taking advantage of the global view exposed by the SD-RAN controller, implement diverse network functionalities. In the next section, we will introduce a particular ML-oriented network application on which it is essential to improve the Wi-Fi network efficiency and realize the promised high transmission capacity.

V. MACHINE LEARNING MODELS DESIGN

A. Design Decisions

The ML models utilized in this work target the 802.11n version of the standard. This release defines four basic modulation schemes (BPSK, QPSK, 16-QAM, and 64-QAM), each of them with different coding schemes. This results in a total of eight basic MCS values (from MCS 0 to MCS 7). Albeit the standard supports up to four MIMO streams, most APs typically support only two MIMO streams. MCS values higher than 7 are essentially the same of the lower ones but with an increasing number of streams, e.g., MCS 15 has two streams, MCS 23 has 3 streams, and MCS 31 has four streams. For this work we have focused purely on effects of MCS on the optimal aggregation length, leaving the analysis of the impact

¹Slice management and definition are out of the scope of this paper. Further details about this system can be found in [38].

²While the transmission policies are set by SD-RAN controller, the frame-by-frame MCS selection is implemented at the AP.

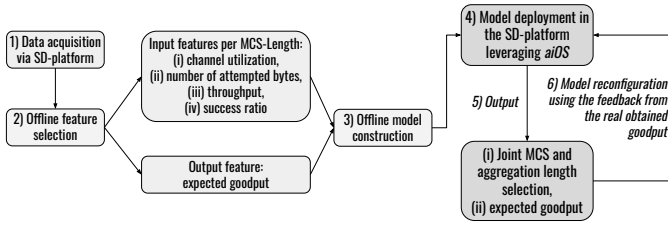


Fig. 5. Flowchart of the construction and deployment of the ML models.

of MIMO for a future work. As a result, the training of the ML models has been done using only MCS values from 0 to 7. Notice also how the 802.11n standard defines two maximum aggregation lengths for an A-MSDU: 3839 and 7935 bytes. Wi-Fi clients can support either of the two values. In this work we have decided to use 3839 bytes as maximum length. The reason for this choice is that this is the most common value found in Wi-Fi clients for 802.11n interfaces.

Concerning the design of the ML models, in this work we have leveraged supervised learning techniques. There are three key reasons for this. First, we aimed to guide the algorithms to predict a specific output variable, i.e., *the expected goodput when selecting a specific MCS and frame length*. Second, the performance of wireless networks can be influenced by many factors. For that reason, we considered controlled scenarios that facilitate the knowledge acquisition. Lastly, in addition to the accuracy, the interpretability and understanding of the models is an essential requirement since, once loaded in the system presented in Fig. 4, they must be able to *learn and adjust* the predictions based on the network outputs.

Supervised learning refers to the process of defining a model, $h_{\Theta}(x)$, from supervised data, which is characterized by n input features, $X = (X_1, \dots, X_n)$, and an output variable, Y . Thus, data must be previously acquired and represented as a pair, (X, Y) . Based on this, the models must predict the output of other unlabeled data, y , from its input features, x . Depending on the output class, two types of supervised learning can be distinguished: classification (for binary/categorical classes) and regression (for numerical classes). Since the output class in this problem is a numerical variable, i.e., the expected goodput for a specific MCS and aggregation length, we have therefore focused on regression models. In particular, we have used *M5P* [39] and *Random Forest Regressor (RFR)* [40] with the aim of comparing their performance and adaptability to the problem. These observable ML techniques are characterized by low computational complexity and simple decision branches, facilitating their comprehension and amendment. Moreover, *RFR* is able to tackle problems with high variance and high bias. Hence, it is suitable for wireless networks problems where channel conditions can vary along time. The process followed for building and deploying the ML models is depicted in Fig. 5 and will be described in detail in the following subsections.

TABLE II
PARAMETERS USED IN THE DATA GENERATION PROCESS.

Parameter	Value
MCS	[0-7]
Aggregation length	[512, 1024, 2048, 3839] bytes
Payload size	[200, 500, 1000, 1470] bytes
Aggregated bitrate	[5, 10, 15, 20, 25, 30, 35, 40] Mbps
Distance from AP	[15, 35, 50, 70] meters
Number of stations	[1, 2, 4, 10]
Background traffic	2 uplink transmissions (1Mbps)

B. Data Acquisition

The objective of data acquisition is to obtain the dataset needed to train the ML models. However, complexity and privacy aspects are an important issue for collecting data from operational networks. For that reason, in this work we have chosen a dataset generation approach using an experimental WiFi testbed (*Step 1* in Fig. 5) based on a network setup similar to the one in Fig. 3 but comprising of a single AP. The AP has been deployed using a PCEngines ALIX 2D (x86) board mounted with an Atheros AR9220 Wi-Fi interface running OpenWRT 18.06.04. The AP has been set on channel 36 isolated from other external noise. The SDN controller has been built using the 5G-EmPOWER SD-RAN controller [41]. Both the controller and the stations have been deployed on Dell laptops with an Intel i7 CPU running Ubuntu 18.04.02.

In this context we have carried out a wide test distribution covering different traffic scenarios, as shown in Table II. Each combination of these parameters are run using the setup described above for a duration of 30 seconds. Among the parameters involved, we shall highlight the aggregation length, for which we have proposed a range of values in addition to the standard one. It should be noted that the ML models seek to independently select the frame length for every particular station instead of imposing a network-wide configuration. For each parameter combination, we have collected the statistics of the rate control algorithm for each station, which have been extended to account not only in terms of packets but also in terms of bytes. The rate control algorithm in this case is Minstrel. Moreover, we have measured other Key Performance Indicators (KPIs) such as goodput, throughput, success ratio, delivery probability, and channel utilization, among others. As a result of this process, approximately 60000 instances were generated to build the training dataset.

C. Model Construction and Learning Process

Once the dataset is acquired, it has been processed offline using the *Sklearn* library, which has been deployed on a *a1.medium* instance at the Amazon EC2 platform. This process includes three main subtasks, namely data cleaning, variable selection and model building (*Step 2* and *Step 3* in Fig. 5).

The ML models are meant to find the frame length that provides the highest goodput for each client. Given the high accuracy of Minstrel, we rely on the MCS that it chooses at each moment and, based on this MCS, the models provide a prediction on the expected goodput for each possible frame size. Despite the several features collected per scenario, not all

of them have clear impact on the prediction. For this reason, after cleaning the dataset (i.e., addressing missing values, duplicates, etc.) we have carried out a variable selection process to reduce the variance of the prediction. For this task we have leveraged Random Forest techniques, which are able to rank input features in such a manner that the purity of the nodes is maximized [42]. As a result, the input features selected are: (i) channel utilization, (ii) number of attempted bytes in the last window of the rate control algorithm, (iii) throughput, and (iv) success ratio of the selected MCS.

Considering these input features, we have built an *M5P* and an *RFR* model for each MCS, limiting the depth of the trees to 3 levels to reduce over-fitting. The models have undertaken a 10-fold cross validation to guarantee that the training and the testing datasets are independent. This process reports on a mean absolute error of 1.73% and 9.80%, for *M5P* and *RFR*, respectively, which shows the accuracy of the models and the relationship between the parameters involved. Notice how the error of *M5P* is much lower than the one offered by *RFR* given that regression trees tend to provide more overfitted models than random forest techniques.

Finally, the models have been deployed at the application layer of the platform presented in Fig. 4, where they can be loaded, and modified in real-time thanks to the *Machine Learning Core* present at the control layer (*Step 4* in Fig. 5). The models are run once per second for every station, producing as output the $\langle MCS, framelength \rangle$ combination and the corresponding predicted goodput (*Step 5* in Fig. 5). However, the models are not static: in the next run, the real goodput obtained is compared with the predicted one, thus correcting the next predictions with a factor, f , that represents the prediction error (*Step 6* in Fig. 5).

VI. PERFORMANCE EVALUATION

A. Methodology

The effectiveness of the ML models has been assessed on the real-world testbed described in the previous section and compared with the performance offered by transmissions performed without frame aggregation and by using the fixed A-MSDU aggregation mechanism defined in the IEEE 802.11 standard. In this regard, we have measured different metrics, namely goodput improvement, channel utilization and retransmission attempts. Moreover, we have analysed the distribution of the aggregation length selected by each ML model.

The layout of the scenario is based on the system model shown in Fig. 3, comprising a single AP, 2 stations transmitting traffic to the AP, and an increasing number of stations (from 2 to 4) receiving UDP traffic from the AP. The configuration of both the AP and the SD-RAN controller is the same as the one previously described in Sec. V-B. Regardless of the number of stations, the aggregated bitrate transmitted by the AP has been set to 20 Mbps, while the payload has been set to 200 bytes. The uplink transmissions have used the same payload size but limiting the aggregated bitrate of the 2 stations to 1 Mbps in order to decrease the transmission opportunities of the AP and account for more realistic scenarios. The traffic has been generated using Iperf.

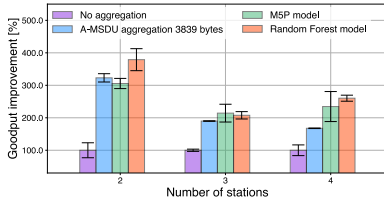
To perform the evaluation under a controlled environment, we have previously placed N stations at distances from 30 to 50 meters from the AP in order to analyse the status of the channel. In this sense, we have observed two main behaviours of the rate control algorithm. At closer distances, i.e., around 30 m, Minstrel tended to select MCS 3 and MCS 4, while at longer distances, i.e., around 50 m, it usually selected MCS 0. Considering that the ML models depend on the MCS chosen, and given the difficulty to replicate the behaviour of Minstrel in real environments, we have differentiated 2 main scenarios with the aim of drawing a comparison in the fairest possible manner. In *Scenario 1*, MCS 0 and MCS 4 have been configured as follows. For 2 stations, one has been set with MCS 0 and the other with MCS 4; for 3 stations, two of them have been set MCS 0 and the other MCS 4; and for 4 stations, the MCS values have been set equally in pairs. Conversely, in *Scenario 2*, MCS 3 and MCS 4 have been configured following the same approach. All the scenarios have been repeated 5 times, and the results shown below are represented with a confidence interval of 95%.

B. Experimental Results

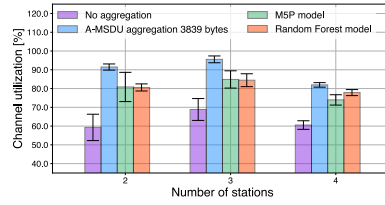
Figure 6 reports on the results of the evaluation performed in *Scenario 1*. In particular, Fig. 6a shows the goodput improvement with respect to the single frame delivery, i.e., without frame aggregation, of the different mechanisms for 2, 3 and 4 stations. In this regard, we can observe that, although the standard A-MSDU policy (fixed to 3839 bytes) enhances the performance of the single frame delivery, the length is not adequate for all the stations. In fact, for the stations under worse channel conditions, the use of longer frames leads to a higher number of transmission errors (as shown in Fig. 6c). By contrast, the *M5P* and the *RFR* models adapt such an aggregation length to the conditions of each station, thus outperforming the results of the standard mechanisms. Notice how the higher tolerance to variance of the *RFR* model allows it to reach the highest goodput improvement by up to 278% with respect to not using frame aggregation and by up to 55% with respect to the standard A-MSDU mechanism.

Moreover, it should be noted that since the channel differences are more significant for 2 stations given that they are using very different MCS values, i.e., MCS 0 and MCS 4, the improvement achieved by performing frame aggregation and, specifically, by the ML models is more significant in this scenario. Again, the standard aggregation mechanism fails when setting the same aggregation length for stations that are clearly under completely diverse conditions. Finally, in Fig. 6b it can be seen how as a result of the improvements in the transmission, the proposed models are able to decrease the channel utilization in all the evaluated cases.

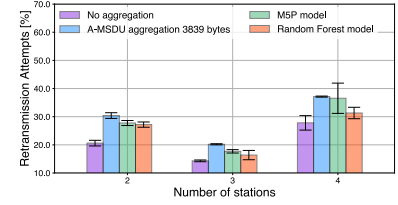
Similarly, Fig. 7 sketches the results obtained in *Scenario 2*, where the stations experience better channel conditions with respect to the previous experiments. This fact can be seen in Fig. 7b, where the channel utilization is significantly lower than in *Scenario 1*. Having similar conditions entails three consequences. First, the enhancement in the simplest setup, i.e., composed of 2 stations, is less significant than



(a) Goodput improvement of the aggregation policies with respect to single frame transmissions.

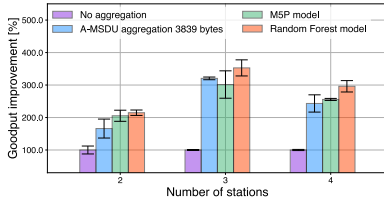


(b) Channel utilization for single frame delivery versus different aggregation policies.

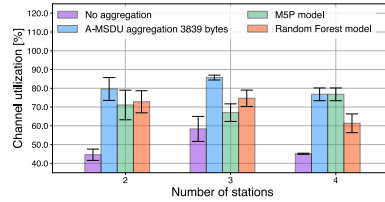


(c) Retransmission attempts for single frame delivery versus different aggregation policies.

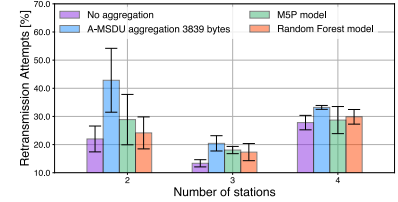
Fig. 6. Performance evaluation for an increasing number of stations in *Scenario 1* (MCS 0 and MCS 4).



(a) Goodput improvement of the aggregation policies with respect to single frame transmissions.

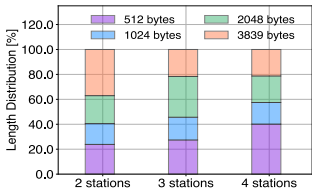


(b) Channel utilization for single frame delivery versus different aggregation policies.

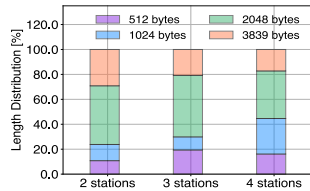


(c) Retransmission attempts for single frame delivery versus different aggregation policies.

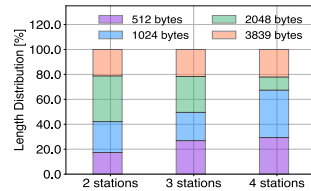
Fig. 7. Performance evaluation for an increasing number of stations in *Scenario 2* (MCS 3 and MCS 8).



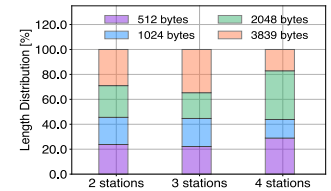
(a) MSP model in *Scenario 1*.



(b) RFR model in *Scenario 1*.



(c) MSP model in *Scenario 2*.



(d) RFR model in *Scenario 2*.

Fig. 8. Distribution of the aggregation length selected by the MSP and RFR models in *Scenario 1* and *Scenario 2*.

in *Scenario 1* for all the schemes. Second, although the ML models outperform the standard A-MSDU aggregation policy, the improvement is slightly reduced due to the good reception conditions. Finally, when increasing the number of stations the improvement ratio stabilises at about 200% with respect to single frame transmission and at about 25% with respect to the standard A-MSDU mechanism using fixed aggregation length. In line with this, it can be also observed that this improvement results in a reduction in both the channel utilization (Fig 7b) and the retransmission attempts (Fig 7c).

Lastly, Fig. 8 reports the distribution of the aggregation length used by the ML models in each scenario. The single frame transmission and the standard A-MSDU aggregation mechanisms are omitted since their frame length does not vary. As can be seen, both ML models select smaller lengths under heterogeneous channel conditions in *Scenario 1* (Fig. 8a and Fig. 8b). Conversely, Fig. 8d shows that *RFR* chooses frame lengths greater than 2048 bytes in 60% of the cases under good channel conditions. However, in both scenarios and regardless of the number of stations, *MSP* is more conservative than *RFR* and tends to select frame lengths below 1024 bytes. This analysis demonstrates again how the *RFR* model is more adequate than the *MSP* tree for being applied in problems with high variance as it is the case of wireless networks.

VII. CONCLUSIONS

This paper aims at lowering the barrier for deploying AI-based control applications in SD-WLANs. To this end, we have proposed *aiOS* as an intelligence plane that is at the same time driven by ML and capable of driving ML applications. *aiOS* embeds several ML functionalities that are exposed for automated network management towards *self-driven* networks. The capabilities of *aiOS* in practical settings have been proved by implementing several ML models for user-adaptive frame length selection in SD-WLANs. Experimental results have shown considerable improvements over standard frame aggregation mechanisms, especially when user channel conditions are substantially diverse. Our future work aims at extending *aiOS* to support a wider range of ML models (including reinforced learning) and network configurations. Moreover, we plan to extend *aiOS* to account for 4G and 5G networks.

ACKNOWLEDGEMENTS

This work has been performed in the framework of the European Union's Horizon 2020 project 5GZORRO co-funded by the EU under grant agreement No 871533.

REFERENCES

- [1] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11 ax high efficiency WLANs," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2018.
- [2] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation with fragment retransmission for very high-speed WLANs," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 591–604, 2009.
- [3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, August 2019.
- [4] K. David and H. Berndt, "6g vision and requirements: Is there any need for beyond 5g?" *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, Sep. 2018.
- [5] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.
- [6] C. Koliass, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 184–208, Firstquarter 2016.
- [7] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2595–2621, Fourthquarter 2018.
- [8] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.
- [9] L. Gavrilovska, V. Atanasovski, I. Macaluso, and L. A. DaSilva, "Learning and reasoning in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1761–1777, Fourth 2013.
- [10] N. Foster, A. Guha, M. Reitblatt, A. Story, M. Freedman, N. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison, "Languages for software-defined networks," *Communications Magazine*, *IEEE*, vol. 51, no. 2, pp. 128–134, February 2013.
- [11] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," *SIGPLAN Not.*, vol. 46, no. 9, pp. 279–291, Sep. 2011.
- [12] X. Jin, L. Li, L. Vanbever, and J. Rexford, "SoftCell: scalable and flexible cellular core network architecture," in *Proc. of ACM CoNEXT*, 2013.
- [13] B. Maqhat, M. D. Baba, and R. A. Rahman, "A-MSDU real time traffic scheduler for IEEE802.11n WLANs," in *Proc. of IEEE Symposium on Wireless Technology and Applications*, Bandung, Indonesia, 2012.
- [14] B. Maqhat, M. Dani Baba, R. A. Rahman, and A. Saif, "Performance analysis of fair scheduler for A-MSDU aggregation in IEEE802.11n wireless networks," in *2014 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE)*, Kuala Lumpur, Malaysia, 2014.
- [15] A. Saif and M. Othman, "A Reliable A-MSDU Frame Aggregation Scheme in 802.11n Wireless Networks," *Procedia Computer Science*, vol. 21, pp. 191–198, 12 2013.
- [16] G. Kim, C. Shin, and H. Park, "Adaptive frame size estimation using extended Kalman filter for high-stressed WLANs," in *proc. of IEEE PIMRC*, Sydney, Australia, 2012.
- [17] J. Saldana, J. Ruiz-Mas, and J. Almodvar, "Frame Aggregation in Central Controlled 802.11 WLANs: The Latency Versus Throughput Tradeoff," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2500–2503, 2017.
- [18] C. Emna, C. Gueguen, L. Fourati, B. Cousin, and L. Kamoun, "Dynamic Frame Aggregation Scheduler for Multimedia applications in IEEE 802.11n Networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, pp. 1–17, 04 2015.
- [19] S. Byeon, K. Yoon, O. Lee, W. Cho, S. Oh, and S. Choi, "MoFA: Mobility-aware Frame Aggregation in Wi-Fi," in *Proc. of ACM CoNEXT*, Sydney, Australia, 2014.
- [20] S. V. Azhari, O. Gurbuz, and O. Ercetin, "QoS based aggregation in high speed IEEE802.11 wireless networks," in *Proc. of Mediterranean Ad Hoc Networking Workshop*, Barcelona, Spain, 2016.
- [21] W. Hyoung Lee and H. Young Hwang, "A-MPDU aggregation with optimal number of MPDUs for delay requirements in IEEE 802.11ac," *PLOS ONE*, vol. 14, pp. 1–17, 03 2019.
- [22] S. Seytnazarov and Y. Kim, "QoS-aware adaptive A-MPDU aggregation scheduler for enhanced VoIP capacity over aggregation-enabled WLANs," in *Proc. of IEEE NOMS*, Taipei, Taiwan, 2018.
- [23] M. Kim, E.-C. Park, and C.-H. Choi, "Adaptive two-level frame aggregation for fairness and efficiency in IEEE 802.11n wireless lans," *Mobile Information Systems*, vol. 2015, pp. 1–14, 01 2015.
- [24] Y. Lin and V. W. S. Wong, "WSN01-1: Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs," in *Proc. of IEEE Globecom*, San Francisco, CA, USA, 2006.
- [25] K. Liu and C. Li, "An efficient adaptive frame aggregation scheme in vehicular Ad Hoc networks," in *Proc. of IEEE WCSP*, Nanjing, Jiangsu, China, 2017.
- [26] X. Zhou and A. Boukerche, "AFLAS: An Adaptive Frame Length Aggregation Scheme for Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 855–867, 2017.
- [27] L. Kriara, M. K. Marina, and A. Farshad, "Characterization of 802.11n wireless LAN performance via testbed measurements and statistical analysis," in *Proc. of IEEE SECON*, New Orleans, USA, 2013.
- [28] X. He, F. Y. Li, and J. Lin, "Link adaptation with combined optimal frame size and rate selection in error-prone 802.11n networks," in *Proc. of IEEE International Symposium on Wireless Communication Systems*, Reykjavik, Iceland, 2008.
- [29] S. Abdallah and S. D. Blostein, "Joint rate adaptation, frame aggregation and MIMO mode selection for IEEE 802.11ac," in *Proc. of IEEE Wireless Communications and Networking Conference*, Doha, Qatar, 2016.
- [30] C. Shin, H. Park, and H. M. Kwon, "PHY-supported frame aggregation for wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2369–2381, 2014.
- [31] D. Kim and S. An, "Throughput enhancement by Dynamic Frame Aggregation in multi-rate WLANs," in *Proc. of IEEE SCVT*, Eindhoven, Netherlands, 2012.
- [32] B. Gautam, "AMSDU vs AMPDU: A Brief Tutorial on WiFi Aggregation Support," Tech. Rep., 2017.
- [33] D. Skordoulis, Q. Ni, H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour, "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40–47, 2008.
- [34] MadWifi, "Implementation of the Onoe Algorithm [online]," http://madwifi-project.org/browser/madwifi/branches/madwifi-0.9.4/ath_rate/onoe/onoe.c, Accessed on 2019-08-20.
- [35] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proc. of ACM MSWiM*, Venice, Italy, 2004.
- [36] D. Xia, J. Hart, and Q. Fu, "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *Proc. of IEEE ICC*, Budapest, Hungary, 2013.
- [37] ITU-R, "Propagation data and prediction methods for the planning of indoor radio communication systems and radio local area networks in the frequency range 900 MHz to 100 GHz," in *Recommendation ITU-R P.1238-7*, 2012.
- [38] E. Coronado, R. Riggio, J. Villalón, and A. Garrido, "Lasagna: Programming Abstractions for End-to-End Slicing in Software-Defined WLANs," in *Proc. of IEEE WoWMoM*, Chania, Greece, 2018.
- [39] J. R. Quinlan, "Improved Use of Continuous Attributes in C4.5," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 77–90, 1996.
- [40] G. Biau, "Analysis of a Random Forests Model," *Journal of Machine Learning Research*, vol. 13, pp. 1063–1095, 2012.
- [41] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming Abstractions for Software-Defined Wireless Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.
- [42] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable Selection Using Random Forests," *Pattern Recognition Letters*, vol. 31, no. 14, 2010.