POLITECNICO DI TORINO
Repository ISTITUZIONALE

Accurate and Efficient Prediction of Wi-Fi Link Quality Based on Machine Learning

Original
Accurate and Efficient Prediction of Wi-Fi Link Quality Based on Machine Learning / Formis, G., Cena, G., Wisniewski, L., Scanzio, S.. - In: IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. - ISSN 1551-3203. - 22:2(2026), pp. 763-774. [10.1109/tii.2025.3609224]

Availability:
This version is available at: 11583/3004399 since: 2025-10-23T13:29:31Z

Publisher:
Institute of Electrical and Electronics Engineers

Published
DOI:10.1109/tii.2025.3609224

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

Accurate and Efficient Prediction of Wi-Fi Link Quality Based on Machine Learning

Gabriele Formis, *Member, IEEE*, Gianluca Cena, *Senior Member, IEEE*, Lukasz Wisniewski, *Senior Member, IEEE*, and Stefano Scanzio, *Senior Member, IEEE*

***Abstract*—Wireless communications are characterized by their unpredictability, posing challenges for maintaining consistent communication quality. This article presents a comprehensive analysis of various prediction models, with a focus on achieving accurate and efficient Wi-Fi link quality forecasts using machine learning techniques. Specifically, the article evaluates the performance of data-driven models based on the linear combination of exponential moving averages, which are designed for low-complexity implementations and are then suitable for hardware platforms with limited processing resources. Accuracy of the proposed approaches was assessed using experimental data from a real-world Wi-Fi testbed, considering both channel-dependent and channel-independent training data. Remarkably, channel-independent models, which allow for generalized training by equipment manufacturers, demonstrated competitive performance. Overall, this study provides insights into the practical deployment of machine learning-based prediction models for enhancing Wi-Fi dependability in industrial environments.**

***Index Terms*—Channel quality prediction, data-driven models, exponential moving average (EMA), machine learning (ML), Wi-Fi.**

I. Introduction

WIRELESS communications suffer from disturbance due to the open nature of the transmission medium, including interference from nearby communication equipment operating in the same frequency range and electromagnetic noise generated by power equipment, which may abound in industrial plants [1]. These phenomena impact on frame transmission attempts tangibly, to the point that their outcomes (either success or failure) can be modeled as binary random variables, whose probability varies over time since spectrum conditions are typically nonstationary. This leads to unreliable behavior, which cannot be tolerated by applications. For this reason, automatic repeat request (ARQ) mechanisms are customarily included in the media access control (MAC) layer, which allow to cope with frame losses by exploiting confirmations and retransmissions, even though, doing so enlarges communication latency and jitter.

As a matter of fact, the lack of adequate determinism affects most of the existing wireless network technologies, slowing down their adoption in industrial environments [2]. Consequently, communication for automation at present still relies for the most part on cables, and is progressively migrating from legacy solutions (fieldbuses) to those based on IEEE 802.3 (industrial Ethernet), which according to the yearly hardware meets software (HMS) study have grown by 12% and now reach 71% of the installed base [3]. The same study also shows that wireless technologies have seen a steady growth in recent years and constitute 7% of the total share in 2024.

The increase in the usage of wireless network technologies in industrial environments is partially motivated by practicability, particularly in the Industry 4.0 context [4], whenever wiring is either impossible or simply inconvenient. This is the case, e.g., for automated guided vehicles (AGVs) and autonomous mobile robots (AMRs) [5], which must coordinate themselves in real-time and integrate their operations in the overall production process. The two most appealing solutions to provide wireless extensions of wired infrastructures in such scenarios [6] are probably IEEE 802.11 [7], also known as Wi-Fi, and mobile 5G networks. Other solutions exist for industry, e.g., IO-Link Wireless (based on the IEEE 802.15.1 PHY) and WirelessHART (based on IEEE 802.15.4), but they offer a noticeably lower throughput and are unsuitable for applications, such as industrial augmented reality [8]. Wi-Fi is the only technology that ensures direct interoperability with Ethernet at the data-link layer, as both rely on the same addressing space (EUI-48) and have similar maximum transmission units and speed. Moreover, the acquisition and maintenance costs for Wi-Fi are noticeably lower than private 5G networks, which are also likely to include licensing fees. Hence, Wi-Fi is preferable for indoor use (shop-floor and warehouse), as witnessed by commercial solutions based on modified Wi-Fi 4 technology, such as Siemens' industrial point coordination function (iPCF) and wireless networks for industrial automation - factory automation (WIA-FA) [9], whereas 5G ensures ubiquitous connectivity.

Making Wi-Fi behavior more dependable and deterministic has been one of the main research goals of the past two decades. The distributed coordination function (DCF) and the enhanced distributed channel access (EDCA) of the hybrid coordination function (HCF) are essentially random mechanisms. Centralized

Received 27 September 2024; revised 19 December 2024, 21 July 2025, 11 August 2025, and 28 August 2025; accepted 1 September 2025. Date of publication 15 October 2025; date of current version 5 February 2026. This work was supported in part by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" under Grant PE00000001 - program "RESTART". Paper no. TII-24-5051. *(Corresponding author: Stefano Scanzio.)*

Gabriele Formis is with the Politecnico di Torino, 10129 Torino, Italy, and also with the CNR-IEIIT, National Research Council of Italy, 00185 Roma, Italy (e-mail: gabriele.formis@polito.it).

Gianluca Cena and Stefano Scanzio are with the CNR-IEIIT, National Research Council of Italy, 00185 Roma, Italy (e-mail: gianluca.cena@cnr.it; stefano.scanzio@cnr.it).

Lukasz Wisniewski is with the Institute Industrial IT-inIT, Technische Hochschule OWL, 32657 Lemgo, Germany (e-mail: lukasz.wisniewski@th-owl.de).

Digital Object Identifier 10.1109/TII.2025.3609224

© 2025 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

solutions, such as the point coordination function (PCF), the HCF controlled channel access (HCCA), and more recently the much more promising trigger frames in Wi-Fi 6, although able to enhance determinism, are unsuitable to tackle sources of disturbance other than nearby Wi-Fi devices.

Improving Wi-Fi dependability constitutes the primary focus of the IEEE 802.11bn Ultra High Reliability Task Group, which is in charge to define Wi-Fi 8 [10]. Among the many mechanisms that can be exploited for this purpose, link-level seamless redundancy (Wi-Red) [11] and coordinated spatial reuse [12] are deemed promising options. In addition, machine learning (ML) techniques can be also exploited to deliver better features. In particular, the ability to foresee with acceptable confidence, the behavior of the wireless spectrum in the near future, may enable mechanisms aimed at improving link quality [13]. Such mechanisms can operate at the MAC layer, in which case the timeframe for forecasts is short (milliseconds up to a second), and at the application layer (including network management), where prediction intervals are larger (seconds to minutes). For example, both rate adaptation (e.g., Minstrel) and packet steering (in Wi-Fi 7) algorithms can benefit from accurate predictions.

Practical feasibility and cost are essential aspects that impact on the chance of any proposal to be considered for inclusion in commercial devices. For this reason, in this article, we seek for prediction models that feature adequate accuracy subject to the constraint that they must allow for efficient implementation on hardware platforms with limited processing resources (CPU and memory), e.g., access points (APs) and wireless adapters for end stations (STAs). In particular, we focused on methods that rely on the exponential moving average (EMA), which is known to enable fast and lightweight implementations, incorporating modifications aimed at maximizing the ability to accurately predict link quality by minimizing the mean prediction error.

The models we present here are data-driven, that is, their operation depends on parameters whose value is determined through a suitable training phase carried out on data acquired from the real world. As we show, this approach yields tangible improvements in prediction accuracy. We also note that results are not completely specific to the environment in which the link quality measurements were performed. Conversely, they are of broader validity, meaning that training was able to capture more general characteristics about the disturbance affecting real Wi-Fi communications, not directly related to the considered scenario.

The rest of the article is organized as follows. Section II provides a brief survey about the state of the art on the subject. Section III presents three simple models for predicting link quality. Section IV reports results of a postanalysis carried out on experimental data. Finally, Section V concludes this article.

II. State of the Art

Industrial wireless communication systems are continually evolving, driven by the ever increasing demand for high-speed, low-latency, and deterministic and reliable connectivity in several application fields, including AGVs, mobile human–machine interface devices, and Internet of Things (IoT) sensors, to cite a few. This evolution has led to a pressing need for more sophisticated techniques to make communication protocols able to adapt to the dynamic nature of the wireless spectrum [14], [15], [16]. One notable challenge is the unpredictability of wireless channel quality, which may fluctuate due to a multitude of factors, including environmental conditions, interference from other devices, and mobility of users and objects, which pose a significant obstacle in achieving consistent and efficient data transmission [17].

While traditional methods for predicting channel quality have served as the foundation for early wireless networks, they often fall short in capturing the intricacies of real-world scenarios. This is due to the fact that they typically rely on simplified statistical models, which may not adequately account for the actual nature of the wireless spectrum. As the spectrum becomes increasingly crowded, with numerous nodes in motion, the wireless environment becomes particularly challenging when it comes to coexistence issues. In contrast, the emergence of ML has revolutionized the approach to wireless channel prediction (WCP) [18], [19], [20], [21]. ML models, particularly artificial neural networks (ANN) and deep neural networks, have demonstrated remarkable capabilities in learning complex patterns and relationships from large datasets [22], [23], [24], [25]. By analyzing historical channel data, these models can extract valuable insights and make accurate predictions about future channel conditions.

Recent research has explored the application of deep learning (DL), and ML in general, to a number of contexts specifically related to industrial networks, such as security [26], network resource allocation [27], but also WCP techniques for both Wi-Fi [28] and 6G [29], [30], with promising results. In [31], a DL-based approach is proposed for channel prediction in millimeter-wave communication systems. Similarly, in [32], a convolutional neural network (CNN) model is developed for predicting channel quality in dynamic vehicular environments, showing the potential of DL in addressing the challenges of wireless communication.

The high heterogeneity of traffic, environment, nodes, and protocols is one of the reasons to exploit ML for WCP. ML can be profitably applied to Wi-Fi in a wide range of application contexts, to predict different relevant quantities and optimize a number of key performance indicators (KPIs). In [33], throughput is predicted by means of a data-driven approach based on several ML methods. Handover prediction and AP selection problems are the main goals addressed in [34]. Instead, [35] focused on the reduction of power consumption through reinforcement learning, a technique commonly used for optimization (e.g., rate adaptation in Wi-Fi) but hardly suitable for WCP. The problem of the coexistence among network protocols in the 2.4 GHz band (e.g., Wi-Fi and Bluetooth) is tackled in [36], with the goal of optimizing the spectrum usage. In this work, the prediction target is the future frame delivery ratio (FDR), but other link quality metrics exist as well, such as the received signal strength indicator (RSSI) or the channel occupancy. We opted for the FDR as its forecasts can be exploited by the protocol layers above the radio block, e.g., to try preventing latency-related constraints of industrial applications from being missed.

Unfortunately, the adoption of complex ML models comes with its own set of challenges, particularly in terms of computation time and resource requirements. Deploying these models on resource-constrained devices, such as IoT sensors and low-power wireless nodes, may prove to be impractical due to limitations in processing power and memory. To address them, alternative approaches are being explored that strike a balance between prediction accuracy and computational effort. One promising direction are lightweight ML models that can run efficiently on embedded hardware platforms [37]. These models leverage techniques, such as quantization, pruning, and model compression, to reduce their memory and computational footprint, while maintaining satisfactory performance.

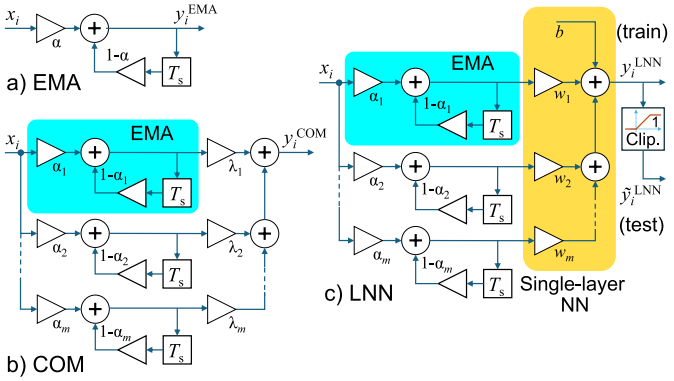


Fig. 1. Operation of prediction models (EMA, COM, and LNN).

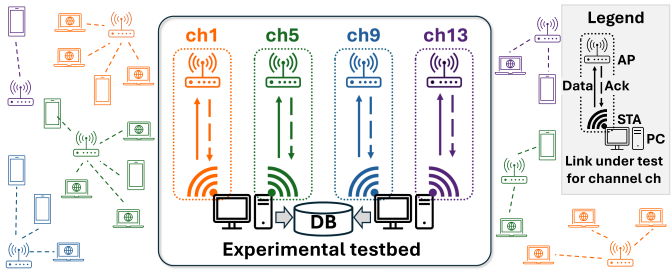


Fig. 2. Experimental testbed: every link under test (operating on channels 1, 5, 9, and 13) sees a different pattern of interferers.

In summary, the search for reliable and efficient wireless communication systems continues to drive innovation in WCP. While advanced ML techniques have unlocked new possibilities for achieving higher levels of accuracy and adaptability, their practical implementation requires careful consideration of both computational constraints and deployment efforts in real-world scenarios.

III. Prediction Models for Wi-Fi Link Quality

This work considers a wireless link between a pair of nodes, e.g., the AP and an associated STA in a Wi-Fi infrastructure network. However, the analysis can be easily applied to other wireless communication technologies and scenarios as well. To characterize link quality, a cyclic probing is carried out by means of confirmed one-shot frame transmissions (no retries allowed). The receiver (the AP, in this case) confirms the correct reception of the ith data frame by returning an acknowledgment (ACK) frame. Depending on whether the ACK frame correctly arrived to the sender or not, the outcome $x_i = 1$ (success) or $x_i = 0$ (failure) were logged, respectively. For every experiment, a dataset is created that consists of the ordered sequence of outcomes $\mathcal{D} = (x_1, \ldots, x_i, \ldots, x_{|\mathcal{D}|})$ obtained from the testbed on a time interval long enough to provide significant statistics (a few weeks), where $|\mathcal{D}|$ denotes the number of included outcomes. Two kinds of datasets are envisaged, namely, *training* ($\mathcal{D}_{\text{tr}}$), used to train prediction models by learning their parameters from real data, and *test* ($\mathcal{D}_{\text{te}}$), used to evaluate their prediction accuracy.

The main goal of this work is to find efficient and effective methods to predict the expected quality of the Wi-Fi link over a *reference* interval of a given duration in the near future. The metric we employ for quality is the FDR, defined as the fraction of transmission attempts the sender performs in said interval that it considers to be successful. In the following, a number of simple prediction models are considered, schematically described in Fig. 1, and their accuracy is compared. The simplest one relies directly on the EMA and can be taken as the baseline. Then, a linear combination of EMA models (COM) was taken into account. Finally, an additional method is introduced, named linear ANN (LNN), which exploits a simple multilayer perceptron with a single layer for combining EMA outcomes and enables cost-effective implementations. Prediction models based on the simple moving average (SMA), as well as on linear and polynomial regression, were not considered since they showed poorer accuracy than EMA [38]. For example, from Table I of that article, the mean squared error (MSE) of the predicted FDR is $0.987 \cdot 10^{-3}$ for SMA, whereas, it is $0.803 \cdot 10^{-3}$ for EMA, which means a 18.64% improvement on accuracy. Every prediction model M is characterized by a distinct set $\boldsymbol{\psi}_{\text{M}}$ of parameters, hence, it is fully described by the tuple $\langle \text{M}, \boldsymbol{\psi}_{\text{M}} \rangle$.

A. Experimental Setup

To shorten the time needed to acquire datasets, four nonoverlapping channels in the 2.4 GHz band were probed contextually by means of two pairs of TP-Link TL-WDN4800 network adapters, which comply with Wi-Fi 4, installed in two Linux PCs (see Fig. 2). This is not a limiting choice, as robustness is prioritized over throughput in industrial environments. Every one of these four STAs was associated to a distinct AP, located about 3 m apart from it (APs were tuned on channels 1, 5, 9, and 13). STAs were instructed to repeatedly send in co-ordered way data frames with period $T_{\text{s}} = 0.5$ s and size 50 bytes (which is typical of industrial networks). Having attempts fairly spaced in time is highly desirable, since our prediction models resemble linear digital filters operating on transmission outcomes. For this purpose, frame aggregation was disabled on the links under test.

Acquisition of datasets relied on a modified version of the `ath9k` device driver, which permitted to detect the arrival of the ACK frame (or the expiration of the ACK timeout) for every transmitted data frame, as well as to disable retransmissions (by setting the retry limit to 0) and backoff (by setting the contention window to 0). By doing so, randomness due to the collision avoidance mechanism of DCF is prevented, and only the effect of disturbance on single transmission attempts was evaluated. Practically, what we did was to "sample" the conditions of every channel in terms of successes and failures of probing attempts on the related link. Sampling is slow enough (2Hz) so that short error bursts, e.g., due to repeated collisions on air, affect results in a limited way, making outcomes of subsequent attempts reasonably independent: in fact, every interfering STA terminates any ongoing transmission process after `MaxTransmitMsduLifetime` (default value 0.5 s). Besides, by doing so, our testbed perturbs the surrounding environment negligibly. If the sampling rate is increased tangibly, then its contribution to channel occupation (which is known) and the related growth of collisions (not easy to evaluate) should be considered as well.

Two experimental campaigns were performed at two distinct times, spaced by a few months, which yielded two sets of four datasets: the former, which covered about 30 days each, were employed for training models; while the latter, which covered about 20 days each, served for testing. They are denoted as $\mathcal{D}_{\text{tr}}^{\text{ch}}$ and $\mathcal{D}_{\text{te}}^{\text{ch}}$, where ch $\in \{1, 5, 9, 13\}$, and every single dataset

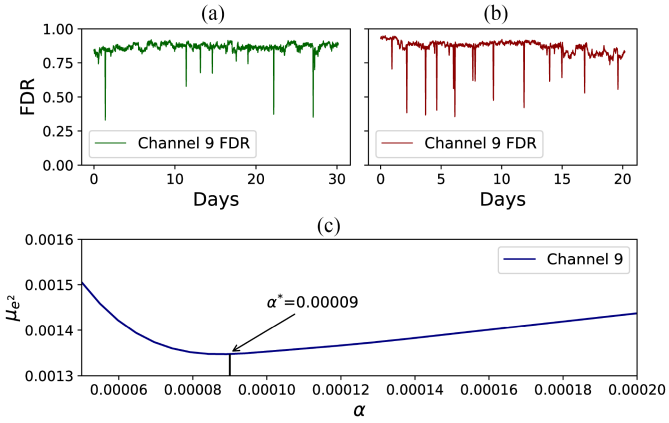


Fig. 3. Timing diagrams for the FDR (target) on channel 9 used for training (a) and testing (b) models, and MSE versus pole placement diagram for EMA that highlights the optimal configuration α^* (c). (a) Train (ch9) (b) Test (ch9) (c) α^* (ch9).

includes about 5 and 3.5 millions of samples, respectively. For example, the two diagrams in the upper part of Fig. 3 refer to channel 9 and report the FDR of the two related datasets, computed over a 30-min moving window centered around the current time i, which can be used as an estimate for the success probability ς_i of the link. Several disturbance phenomena can be intuitively observed, characterized by different dynamics, which cause the FDR to fluctuate to some extent (variations are typically less than $\pm 10\%$). However, occasionally events occur that cause the link quality to drop abruptly. The two plots visually bear some resemblance, which highlights the fact that the interference patterns seen on the same channel in the two distinct timeframes were similar.

Datasets acquired on the different links are actually representative of distinct spectrum conditions, as: a) the spatial distribution of nearby interferers (APs and STAs, including Wi-Fi 6 ones) observed on the related channel is different, as well as the traffic they send on air (amount and pattern); b) the spatial orientation (angle) of any direct STA $\rightarrow$ AP link with respect to walls and objects differed by at least 20°, which made effects due to multipath fading differ. Hence, we expect that claims about prediction accuracy derived from them are broadly valid. An informal proof of this can be found in Fig. 3 of [18], which considers datasets acquired with a very similar testbed: as can be seen there, the patterns about the FDR for the four channels differ noticeably, this reflecting the different amount and type of observed interference.

We purposely did not inject any additional interfering traffic, as we are interested in checking to what extent prediction models are able to provide reliable forecasts about the effects of the traffic generated over the air by real applications running on nearby nodes (beacons, downloads, multimedia, etc.). Moreover, in this article, we did not consider the effects of node mobility. While extremely relevant, this kind of analysis must also consider roaming, and is left as future work together with Wi-Fi network digital twins [39] (it is worth noting that WCP is a basic component of network digital twins, and combined with other elements and methods, it may concur in creating a federated digital twin). In the absence of mobility, signal attenuation is a lesser cause of unpredictability, hence, we set the distance between STA and AP similar for all the links.

B. Prediction Accuracy

Since link quality probing occurs at a periodic pace, time proceeds in discrete steps of equal duration T_s. At any given time i, let x_i denote the most recent outcome and $y_i^{\langle \mathrm{M},\boldsymbol{\psi}_\mathrm{M}\rangle}$ the FDR *forecast* provided by prediction model M with parameters $\boldsymbol{\psi}_\mathrm{M}$. Clearly, only outcomes $\{x_l\}_{l\in[1,i]}$ can be used by M to compute $y_i^{\langle \mathrm{M},\boldsymbol{\psi}_\mathrm{M}\rangle}$. Whatever the model, no predictions are made available until at least N_p outcomes have been processed, so that enough information about the link is available to make forecasts accurate. In this way, the initial transient is skipped and FDR predictions can settle down.

Prediction accuracy at any time $N_\mathrm{p} \le i \le |\mathcal{D}| - N_\mathrm{f}$ can be assessed a posteriori using a *target* z_i that coincides with the FDR in a reference interval in the immediate future whose width is $T_\mathrm{f} = N_\mathrm{f} \cdot T_\mathrm{s}$, computed as the SMA of the following N_f outcomes $(x_{i+1}, \ldots, x_{i+N_\mathrm{f}})$

$$z_i = \frac{1}{N_\mathrm{f}} \sum_{l=i+1}^{i+N_\mathrm{f}} x_l \tag{1}$$

which provides an unbiased estimate of the mean success probability $\overline{\varsigma}_i$ in said interval (we selected $N_\mathrm{f} = 3600$, which implies $T_\mathrm{f} = 30$ min). As an example, apart from a 15 min time shift, the target for channel 9 coincides with the plots in Fig. 3(a)/(b). Shrinking N_f excessively should be avoided, since fluctuations of the (x_i) random process may impair the precision of target z_i and render it unreliable: in fact, it is evaluated as the mean of binary random quantities and its variance depends linearly on $1/N_\mathrm{f}$. Previous experiments on other datasets showed that the future interval T_f can be shrunk down to 3 min without impacting on accuracy appreciably, but decreasing it below 1 min is not advisable. Therefore, our prediction models meet the needs of adaptive distributed applications (where, e.g., the rate of nontime-sensitive data exchanges is lowered when spectrum conditions are expected to worsen) but they can hardly be exploited at the MAC layer. The target is used both in the training phase (where $\mathcal{D} = \mathcal{D}_\mathrm{tr}^\mathrm{ch}$) and in the test phase (by setting $\mathcal{D} = \mathcal{D}_\mathrm{te}^\mathrm{ch}$). In the latter case, it permits to reliably compare the accuracy of different models in the same operating conditions.

Generally speaking, the closer $y_i^{\langle \mathrm{M},\boldsymbol{\psi}_\mathrm{M}\rangle}$ and z_i, the better the prediction ability of model M parameterized according to $\boldsymbol{\psi}_\mathrm{M}$. Besides the instantaneous error $e_i = z_i - y_i$, the absolute error $|e_i|$ and the squared error e_i^2 were additionally considered. Starting from them, statistical indices can be evaluated for any given dataset $\mathcal{D}$. Relevant quantities include average (μ_e, $\mu_{|e|}$, and μ_{e^2}), standard deviation (σ_e, $\sigma_{|e|}$, and σ_{e^2}), minimum (e_min), high percentiles (e_{p90}, e_{p95}, e_{p99}, $|e|_{p90}$, $|e|_{p95}$, $|e|_{p99}$, e^2_{p90}, e^2_{p95}, and e^2_{p99}), and maximum (e_max, $|e|_\mathrm{max}$, and e^2_max), every one computed on $|\mathcal{D}| - N_\mathrm{p} - N_\mathrm{f} + 1$ predictions, which constitute suitable metrics for assessing prediction accuracy. For example, the MSE μ_{e^2} is customarily employed as the objective function in model training.

For every model we consider, a suitable training phase is preliminarily carried out to determine its optimal parameters. In particular, we denote $\boldsymbol{\psi}^*{}_\mathrm{M}(\mathcal{D}_\mathrm{tr}^\mathrm{ch})$ the set of parameter values that minimizes μ_{e^2} for dataset $\mathcal{D}_\mathrm{tr}^\mathrm{ch}$

$$\boldsymbol{\psi}^*{}_\mathrm{M}(\mathcal{D}_\mathrm{tr}^\mathrm{ch}) = \arg\min_{\boldsymbol{\psi}_\mathrm{M}} \sum_{i=N_\mathrm{p}}^{|\mathcal{D}_\mathrm{tr}^\mathrm{ch}|-N_\mathrm{f}} \left(z_i - y_i^{\langle \mathrm{M},\boldsymbol{\psi}_\mathrm{M}\rangle}\right)^2. \tag{2}$$

To achieve a fair comparison, the same training datasets $\mathcal{D}_{\text{tr}}^{\text{ch}}$ were used for all models, and above accuracy metrics were evaluated for every optimized model $\langle \text{M}, \boldsymbol{\psi}^*{}_{\text{M}}(\mathcal{D}_{\text{tr}}^{\text{ch}})\rangle$ using the very same test datasets $\mathcal{D}_{\text{te}}^{\text{ch}}$. From now on, superscript $\langle \text{M},\boldsymbol{\psi}_{\text{M}}\rangle$ will be omitted when it can be clearly identified by the context, in particular, when referring to optimized models.

C. Exponential Moving Average

Every sequence (x_i) of outcomes can be seen as an instance of a binary random process, which must be filtered suitably to obtain reliable estimates of the instantaneous link quality. As shown in [40], one of the most basic (and popular) ways to perform predictions is computing their EMA, described by

$$y_i^{\langle \text{EMA},\alpha\rangle} = \alpha \cdot x_i + (1-\alpha)\cdot y_{i-1}^{\langle \text{EMA},\alpha\rangle} \tag{3}$$

where y_{i-1} denotes the previous prediction and α is a coefficient, also known as smoothing factor, for balancing present and past ($0 < \alpha < 1$). Higher values of α make the model more reactive in tracking sudden changes of the link quality, while those close to 0 make it less susceptible to statistical fluctuations. Value y_0 was initialized to 0.5, but more fitting estimates should be used if available. Equation (3) corresponds to the simplest form of an infinite impulse response (IIR) low-pass filter [see Fig. 1(a)], described in the z-domain by the transfer function

$$H_\alpha(z) = \frac{Y(z)}{X(z)} = \frac{\alpha}{1-(1-\alpha)z^{-1}} = \frac{1-\beta}{1-\beta z^{-1}} \tag{4}$$

where $\beta = 1-\alpha$ represents the pole. As can be seen, the EMA model is completely parameterized by the placement of the pole, which implies that $\boldsymbol{\psi}_{\text{EMA}} = \{\alpha\}$.

Let α^* be the optimal value of α that minimizes the objective function μ_{e^2} for $\mathcal{D}_{\text{tr}}^{\text{ch}}$ according to (2). This means that $\boldsymbol{\psi}^*{}_{\text{EMA}} = \{\alpha^*\}$. Determining α^* in real operating conditions is what data-driven training is meant to achieve. For example, Fig. 3(c) (lower part) depicts μ_{e^2} versus α for channel 9. Determining the position of the minimum can be done easily for the EMA model, since the above function is typically convex. In fact, lower values of α decrease jitters (and hence the MSE), making results more stable, but reducing the cutoff frequency too much prevents the EMA from promptly track variations of the link quality (the MSE increases again, making predictions worse in those nonstationary scenarios we are interested in). In the case of channel 9, $\alpha^* = 0.00009$.

As well known from the literature, the precision (MSE) with which the instantaneous FDR can be evaluated for binary random processes in stationary conditions is $\varsigma(1-\varsigma)\cdot\alpha/(2-\alpha)$ for y_i (EMA) and $\varsigma(1-\varsigma)/N_{\text{f}}$ for z_i (SMA). By setting $\varsigma = 0.8652$ (the mean value of outcomes x_i in $\mathcal{D}_{\text{tr}}^{9}$), we obtain that the overall MSE is $3.76\cdot 10^{-5}$ (the sets of samples used by EMA and SMA are disjoint, and their MSEs can be summed), much smaller than the minimum μ_{e^2}, as observed in Fig. 3(c) for real channel 9 (about $1.35\cdot 10^{-3}$). This means that the prediction errors mostly depend on sudden variations of the spectrum conditions, which cannot be foreseen by moving averages, and not on uncertainties due to an insufficient number of samples.

D. Linear Combination of EMAs

To improve the prediction accuracy, a second model (COM) can be exploited, which relies on a linear combination of the output values produced by several concurrently operating EMA models, each one characterized by its smoothing factor α_j (i.e., by the related pole $\beta_j = 1-\alpha_j$)

$$y_i^{\text{COM}} = \sum_{j=1}^{m} \lambda_j \cdot y_i^{\langle \text{EMA},\alpha_j\rangle} \tag{5}$$

with weights $0 \le \lambda_j \le 1$ chosen in such a way that

$$\sum_{j=1}^{m} \lambda_j = 1. \tag{6}$$

The COM model given by (5) is fully described by the tuple $\boldsymbol{\psi}_{\text{COM}} = \langle \boldsymbol{\alpha}, \boldsymbol{\lambda}\rangle$, where $\boldsymbol{\alpha} = (\alpha_j)_{j\in[1,m]}$ and $\boldsymbol{\lambda} = (\lambda_j)_{j\in[1,m]}$. It coincides with a multipole low-pass IIR filter [Fig. 1(b)] characterized by the transfer function

$$H_{\langle \boldsymbol{\alpha},\boldsymbol{\lambda}\rangle} = \sum_{j=1}^{m} \lambda_j \cdot H_{\alpha_j}(z) \tag{7}$$

whose poles coincide with vector $\boldsymbol{\beta} = \mathbf{1} - \boldsymbol{\alpha}$. Equations (4) and (7) in the z-domain are not used for calculations (that are carried out directly on samples in the time domain by means of (3) and (5), for EMA and COM, respectively), but are only meant as compact model descriptions.

Unlike the EMA model, finding the optimal configuration $\boldsymbol{\psi}^*{}_{\text{COM}}$ that minimizes the MSE is not completely trivial. In the following, a suitable training procedure is described that consists of three steps and provides good results.

1) Initial Pole Selection: A number of distinct EMA models are initially considered, characterized by different α' coefficients taken from an *initial sequence* defined as

$$\boldsymbol{\alpha}_{\mathbf{s}} = (\alpha'_j \mid \alpha'_j = \alpha^* \cdot r^{j-N_{\text{l}}-1})_{j\in[1,N_{\text{l}}+N_{\text{u}}+1]} \tag{8}$$

which is a finite-size *geometric progression* with common ratio $r > 1$ that includes $|\boldsymbol{\alpha}_{\mathbf{s}}| = N_{\text{s}} = N_{\text{l}} + N_{\text{u}} + 1$ elements in strictly increasing order (see block 1 in Fig. 4). N_{l} and N_{u} specify the lower and upper bounds for the pool of available α' coefficients, respectively, and are chosen in such a way that the most significant poles of the filter we are looking for fit in the range $[\alpha^*\cdot r^{-N_{\text{l}}}, \alpha^*\cdot r^{N_{\text{u}}}]$. Instead, r affects granularity and permits to adjust the tolerance with which such poles will be approximated. For example, when $r = \sqrt{2}$, $N_{\text{l}} = 2$, and $N_{\text{u}} = 4$, then $\boldsymbol{\alpha}_{\mathbf{s}} = (\frac{\alpha^*}{2}, \frac{\alpha^*}{\sqrt{2}}, \alpha^*, \sqrt{2}\alpha^*, 2\alpha^*, 2\sqrt{2}\alpha^*, 4\alpha^*)$, and $N_{\text{s}} = 7$ poles are included.

2) MSE Minimization: Optimal weights for the above EMA models are then evaluated. This has been addressed as a minimization problem for the MSE, as formally described by (2), of the COM model specified by (5) over the training dataset $\mathcal{D}_{\text{tr}}^{\text{ch}}$, constrained by both (6) and the given boundaries on every λ_j. We relied on the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS-B) [41], a quasi-Newton gradient-based optimization method that iteratively updates an estimate of the solution using gradient and curvature information, while also handling boundary constraints. It is designed for high-dimensional problems and can find the minimum of a function efficiently. This algorithm is implemented in the `minimize` function included in the `Python` library `scipy.optimize`. The outcome is a sequence $\boldsymbol{\lambda}_{\mathbf{s}} = (\lambda'_1, \lambda'_2, \ldots, \lambda'_{N_{\text{s}}})$ of optimal weights for the linear combination in (5), each one associated to the corresponding smoothing factor in $\boldsymbol{\alpha}_{\mathbf{s}}$ (by definition

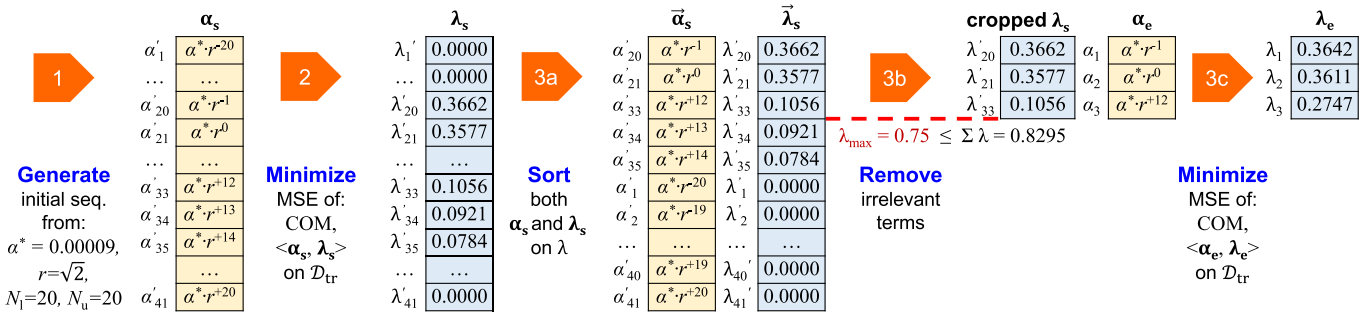


Fig. 4. Training procedure for the COM model (channel 9 is shown): removing irrelevant terms lowers the number of poles from 41 down to 3.

$|\boldsymbol{\alpha}_\mathbf{s}| = |\boldsymbol{\lambda}_\mathbf{s}| = N_\mathrm{s}$), which minimizes the MSE for the COM model (block 2). Tuple $\langle \boldsymbol{\alpha}_\mathbf{s}, \boldsymbol{\lambda}_\mathbf{s} \rangle$ constitutes the initial, optimal selection for parameter configuration.

3) Final Pole Selection: We observed that a fair number of the weights in $\boldsymbol{\lambda}_\mathbf{s}$ evaluated in the previous step were typically rather small, hence the contribution of the related EMA filter to the prediction is irrelevant. This is even more true when test is performed on a dataset other than the one used for minimization, e.g., $\mathcal{D}_{\mathrm{te}}^{\mathrm{ch}}$. If $\boldsymbol{\alpha}_\mathbf{s}$ and $\boldsymbol{\lambda}_\mathbf{s}$ are shortened by removing such elements, then the algorithm becomes faster and more suitable for implementation in embedded devices with scarce computational power, with a negligible impact on accuracy. The *final sequence* $\boldsymbol{\alpha}_\mathbf{e}$ is obtained from $\boldsymbol{\alpha}_\mathbf{s}$ by selecting the elements α'_j for which the corresponding weights λ'_j are larger. Practically, let $\vec{\boldsymbol{\lambda}}_\mathbf{s} = (\lambda'_{\pi_1}, \lambda'_{\pi_2}, \ldots, \lambda'_{\pi_{N_\mathrm{s}}})$ be the same as $\boldsymbol{\lambda}_\mathbf{s}$ but sorted in decreasing order, that is, $\iota < \kappa \Rightarrow \lambda'_{\pi_\iota} \geq \lambda'_{\pi_\kappa}$. Sequence $(\pi_j)_{j\in[1,N_\mathrm{s}]}$ is a suitable permutation of $(1, 2, \ldots, N_\mathrm{s})$ that enforces such ordering. Similarly, $\vec{\boldsymbol{\alpha}}_\mathbf{s} = (\alpha'_{\pi_1}, \alpha'_{\pi_2}, \ldots, \alpha'_{\pi_{N_\mathrm{s}}})$ contains the elements in $\boldsymbol{\alpha}_\mathbf{s}$ reordered according to (π_j) (block 3a). Then, $\boldsymbol{\alpha}_\mathbf{e}$ is a prefix of $\vec{\boldsymbol{\alpha}}_\mathbf{s}$

$$\boldsymbol{\alpha}_\mathbf{e} = \left(\alpha_j \mid \alpha_j = \alpha'_{\pi_j} \in \boldsymbol{\alpha}_\mathbf{s}\right)_{j\in[1,N_\mathrm{e}]}. \tag{9}$$

The number $N_\mathrm{e} = |\boldsymbol{\alpha}_\mathbf{e}| \leq N_\mathrm{s}$ of EMA filters that are retained for testing is not fixed. Instead, it is evaluated as the minimum number of them for which the sum of the related weights exceeds a given threshold $\lambda_{\max}$ (block 3b)

$$N_\mathrm{e} = \min\left\{ n \in [1, N_\mathrm{s}] \,\middle|\, \sum_{j=1}^{n} \lambda'_{\pi_j} \geq \lambda_{\max} \right\}. \tag{10}$$

For example, $\lambda_{\max} = 0.75$ means that only the elements strictly needed, so that their weights contribute cumulatively for at least 75%, are retained. Setting $\lambda_{\max} = 1.0$ implies that no EMA filters are removed, that is, $\boldsymbol{\alpha}_\mathbf{e} = \boldsymbol{\alpha}_\mathbf{s}$ and $\boldsymbol{\lambda}_\mathbf{e} = \boldsymbol{\lambda}_\mathbf{s}$.

Since (6) must still hold, MSE minimization on $\mathcal{D}_{\mathrm{tr}}^{\mathrm{ch}}$ is applied a second time to $\boldsymbol{\alpha}_\mathbf{e}$ to obtain the final sequence of weights $\boldsymbol{\lambda}_\mathbf{e}$ (block 3c). Tuple $\boldsymbol{\psi}^*{}_{\mathrm{COM}} = \langle \boldsymbol{\alpha}_\mathbf{e}, \boldsymbol{\lambda}_\mathbf{e} \rangle$ satisfactorily approximates optimality, and constitutes the configuration of parameters we used for testing.

The above optimization procedure was applied to the training datasets collected from the testbed using $r = \sqrt{2}$, $N_\mathrm{l} = 20$, and $N_\mathrm{u} = 20$. As depicted in the example of Fig. 4, which refers to channel 9, the initial selection $\boldsymbol{\alpha}_\mathbf{s}$ includes 41 smoothing factors in the range from $8.789 \cdot 10^{-8}$ to $9.216 \cdot 10^{-2}$. A first minimization phase on $\mathcal{D}_{\mathrm{tr}}^{\mathrm{ch9}}$ provides $\boldsymbol{\lambda}_\mathbf{s}$, which is subsequently rearranged in descending order. By setting $\lambda_{\max} = 0.75$, the number of relevant terms in $\boldsymbol{\alpha}_\mathbf{e}$ (i.e., poles of the transfer function) shrinks to just three. Applying minimization a second time yields $\boldsymbol{\lambda}_\mathbf{e}$.

E. Linear Combination of EMAs with an ANN

In this section, we explore the integration of EMA and simple neural networks consisting of a single linear layer. This model (LNN) consists of a number of concurrent EMA filters, each one described by (3), which are fed in parallel with the same outcomes x_i (as in the COM case). At any discrete time i, EMA predictions can be collectively described as a vector $\mathbf{y}_i^{\mathrm{EMA}} = (y_i^{\langle \mathrm{EMA}, \alpha_j \rangle})_{j\in[1,N_\mathrm{s}]}$, where α_j is the jth element in $\boldsymbol{\alpha}_\mathbf{s}$ that describes the smoothing factor of the related EMA. This vector summarizes the information derived from the history of outcomes with a focus on recent observations, but considering different dynamics of disturbance. Its elements are fed as input features to a single-layer neural network [see Fig. 1(c)], whose actual operation is a scalar (dot) product with a *weight* vector $\mathbf{w}$, after which a (scalar) *bias* b is added

$$y_i^{\mathrm{LNN}} = \mathbf{y}_i^{\mathrm{EMA}} \cdot \mathbf{w} + b. \tag{11}$$

This model, for which $\boldsymbol{\psi}_{\mathrm{LNN}} = \langle \boldsymbol{\alpha}_\mathbf{s}, \mathbf{w}, b \rangle$, uses an identity activation function for training, which makes it completely linear like the COM one. Unlike the COM model, however, the optimal configuration $\boldsymbol{\psi}^*_{\mathrm{LNN}}$ of the LNN is determined using the conventional ANN training methods based on the Adam optimizer, which was chosen for its ability to dynamically adapt to loss gradients during training, facilitating effective and rapid learning. The MSE was selected as the loss function, to assess the discrepancy between model predictions and the target. To promote convergence toward a global minimum, a gradual reduction of the learning rate was implemented during training. In addition, the model's performance was continuously monitored using metrics, such as MSE and the mean absolute error $\mu_{|e|}$, ensuring effective convergence during training. More in detail, the model was trained with the `Keras` module of `TensorFlow` in 15 epochs, with batchsize equal to 64, and the learning rate initialized to 0.01 and halved at each epoch. Weights were randomly initialized with the `Glorot normal` initializer.

To always provide meaningful results, a clipping function $\tilde{y}_i^{\mathrm{LNN}} = \min(1, \max(0, y_i^{\mathrm{LNN}}))$ was also defined, only used in the test phase, which enforces FDR forecasts to stay in [0, 1].

IV. Results

Experiments were carried out to assess the accuracy of the three proposed prediction models (EMA, COM, and LNN). Two distinct cases were taken into account for any one of them, which differed for training and, as a consequence, the related optimal configuration parameters ${\psi^*}_{\mathrm{M}}(\mathcal{D}_{\mathrm{tr}})$. The EMA model was taken as the reference, because it offers a good compromise between prediction quality and computational complexity [38] (sort of a state-of-the-art).

In the first experimental campaign, *channel-dependent* training was employed for models, i.e., the training and test datasets referred to exactly the same channel (we say they are coherent). While *specialized* training should offer, in theory, the best performance, applying it in real-world scenarios could be cumbersome. In fact, the training dataset must be acquired from the actual hardware deployed in the intended environment. Even worse, training should be periodically repeated to adapt the prediction model to changes in the spectrum conditions. This implies that suitable procedures need to be incorporated in Wi-Fi equipment that operate continuously and autonomously, which increase implementation complexity.

Consequently, a second experimental campaign was carried out where above models were treated as *channel-independent* ones, this means that the training and test datasets did not refer to the same channel. This condition permits to assess models' accuracy against previously unseen interference patterns: in fact, different channels are characterized by different interfering nodes, of different kinds and placed in distinct relative positions, and generating different traffic patterns belonging to various protocols (including Bluetooth and wireless sensor networks based on IEEE 802.15.4). Also, their behavior over time can be assumed to vary independently. The aim is to mimic the behavior of *generalized* training, which can be performed once and for all in the design phase. Using static, pretrained models has the advantage that they can be integrated by the manufacturer inside Wi-Fi equipment, which leads to dramatically lower implementation complexity. Whether or not the accuracy of channel-independent models resembles channel-dependent ones is what we wish to determine through experimentation on real data. Acquisitions of the test and training datasets were spaced by a few months. In this time lapse, the configuration of most of the nearby APs (not under our control) remained the same: this means that their operating channels and relative positions (distance and angle) with respect to our testbed did not vary. The same applies to the environment topology, including walls, furniture, and fixed obstacles. Conversely, we expect that the number of STAs associated to any AP, as well as their position and traffic, varies consistently over time, especially for smartphones and notebooks of students in nearby premises. This explains why the spectrum conditions we observed on any of the four considered channels differed noticeably from one another, and kept changing over time.

We computed the Pearson correlation coefficients between the outcomes of the transmissions performed at the same time on the different channels and discovered that they were mostly uncorrelated. Only channels 9 and 13 showed some marginal correlation, which was likely due to nearby APs and STAs tuned on "canonical" channel 11, which overlaps with both of them causing contextual interference. Hence, we deem that: a) the spectrum conditions described by datasets obtained on distinct channels are different enough and, when they are used to train models, different optimal parameter configurations are expected; b) such datasets are also uncorrelated and can be exploited to perform generalized training as well.

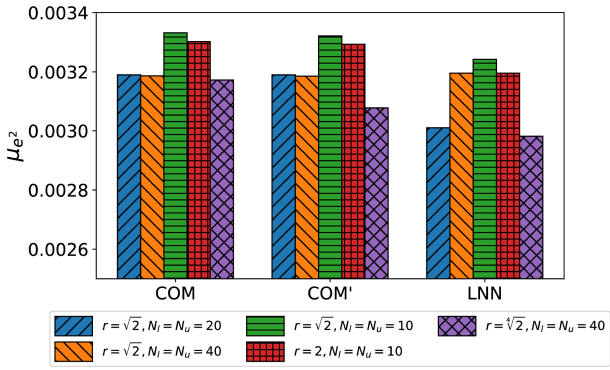


Fig. 5. Parameter sensitivity analysis (MSE versus smoothing factors).

A. Channel-Dependent Models

Results about prediction accuracy when channel-dependent training is exploited are reported in Table I. For every channel, the metrics about accuracy related to the EMA, COM, and LNN models are shown in separate rows. Two rows are included for the linear combination of EMAs, denoted COM', where all N_{s} poles in the initial selection are included (${\psi^*}_{\mathrm{COM'}} = \langle \boldsymbol{\alpha_{\mathrm{s}}}, \boldsymbol{\lambda_{\mathrm{s}}} \rangle$), and COM, which just considers the N_{e} most significant ones in the final selection (${\psi^*}_{\mathrm{COM}} = \langle \boldsymbol{\alpha_{\mathrm{e}}}, \boldsymbol{\lambda_{\mathrm{e}}} \rangle$). The most important metric we consider for accuracy is the MSE. However, high order percentiles for the absolute error ($|e|_{\mathrm{p90}}$, $|e|_{\mathrm{p95}}$, and $|e|_{\mathrm{p99}}$) are also relevant, as they specify an upper bound on the error incurred by 90%, 95%, and 99% of all the performed predictions, respectively. Best and worst cases in the table have been highlighted in green and red, respectively.

As can be seen, COM and LNN always behave better than EMA, with LNN often offering the best accuracy. The two variants of COM practically provided the same accuracy, which means that retaining all the poles in the low-pass filter described by $\boldsymbol{\alpha_{\mathrm{s}}}$ is useless. For this reason, only significant terms described by $\boldsymbol{\alpha_{\mathrm{e}}}$ will be retained in the following. Conversely, the LNN has proved able to exploit all terms in $\boldsymbol{\alpha_{\mathrm{s}}}$. This is probably due to the training procedure for neural networks, which outperformed the L-BFGS-B optimization method. In this case, better accuracy could justify higher implementation complexity, which is a nonnegligible result.

B. Sensitivity Analysis of Parameter Selection

All the presented models are parameterized by poles (described by $\boldsymbol{\alpha}$), whose placement depends on r, N_{l}, N_{u}, and λ_{max}. Basically, the latter parameter λ_{max} was selected in order not to worsen COM performance noticeably. In fact, results about COM in Table I are mostly the same as COM'.

To analyze the sensitivity of training with respect to the model configuration procedure (in terms of the prediction accuracy for the FDR), four additional experiments were carried out on channel 9 by varying the former three parameters. In the first two experiments, we kept the variation step of the smoothing factors α_j fixed ($r = \sqrt{2}$) and doubled/halved the width of their range by selecting $N_l = N_u = 40$ and $N_l = N_u = 10$, respectively. Results about μ_{e^2} are sketched in the histogram of Fig. 5, where the first bar of every set of bars (the blue ones) coincides with the value in Table I ($r = \sqrt{2}$, $N_l = N_u = 20$, taken as baseline),

TABLE I
ACCURACY OF PREDICTION MODELS (EMA, COM, AND LNN) WITH CHANNEL-DEPENDENT TRAINING ON CHANNELS 1, 5, 9, AND 13

Training / Tests	Prediction models (M)	Model param. (ψ^*_M)	μ_{e^2}	e^2_{p95} [$\cdot 10^{-3}$]	e^2_{max}	$\mu_{\lvert e\rvert}$	$\sigma_{\lvert e\rvert}$	$\lvert e\rvert_{p90}$ [%]	$\lvert e\rvert_{p95}$ [%]	$\lvert e\rvert_{p99}$ [%]	$\lvert e\rvert_{max}$ [%]	e_{min} [%]	e_{p5} [%]	e_{p95} [%]	e_{max} [%]
ch1 / ch1	EMA	$\alpha^* = 0.000900$	2.03	10.58	111.53	2.64	3.65	6.92	10.29	17.57	33.40	-31.35	-6.73	7.16	33.40
	COM	$m = N_e = 6$	1.87	9.42	117.92	2.65	3.41	6.82	9.70	15.81	34.33	-32.64	-7.07	6.57	34.34
	COM'	$m = N_s = 41$	1.87	9.45	117.15	2.65	3.41	6.82	9.72	15.80	34.23	-32.57	-7.11	6.56	34.22
	LNN	–	1.86	9.62	112.30	2.67	3.38	6.78	9.81	15.56	33.51	-32.61	-7.34	6.34	33.51
ch5 / ch5	EMA	$\alpha^* = 0.000085$	1.15	2.43	190.43	1.89	2.81	3.90	4.93	12.22	43.64	-43.64	-3.61	4.16	15.64
	COM	$m = N_e = 3$	0.96	1.91	195.45	1.73	2.57	3.41	4.38	10.28	44.21	-44.21	-3.40	3.41	31.42
	COM'	$m = N_s = 41$	0.96	1.91	195.45	1.74	2.57	3.40	4.37	10.29	44.21	-44.21	-3.40	3.40	31.44
	LNN	–	0.91	1.74	189.75	1.74	2.47	3.29	4.17	9.64	43.56	-43.56	-3.89	2.77	25.28
ch9 / ch9	EMA	$\alpha^* = 0.000090$	3.68	10.03	248.20	2.65	5.46	5.54	10.01	31.44	49.82	-49.82	-4.56	6.01	21.43
	COM	$m = N_e = 3$	3.19	5.43	258.17	2.39	5.11	4.39	7.37	29.24	50.81	-50.81	-3.74	4.72	36.18
	COM'	$m = N_s = 41$	3.19	5.40	257.83	2.39	5.12	4.37	7.35	29.28	50.78	-50.78	-3.73	4.71	36.27
	LNN	–	3.01	3.56	257.70	2.45	4.91	4.02	5.97	29.20	50.76	-50.76	-4.49	3.68	28.40
ch13 / ch13	EMA	$\alpha^* = 0.000500$	1.22	5.82	41.55	2.42	2.53	5.51	7.63	12.29	20.38	-18.45	-5.31	5.76	20.38
	COM	$m = N_e = 6$	1.09	4.86	40.08	2.37	2.30	5.19	6.97	11.10	20.02	-16.27	-5.01	5.39	20.02
	COM'	$m = N_s = 41$	1.09	4.84	40.63	2.38	2.29	5.18	6.96	11.07	20.16	-16.32	-5.01	5.39	20.16
	LNN	–	1.11	4.55	35.52	2.51	2.18	5.13	6.74	10.55	18.85	-17.02	-5.17	5.07	18.85

TABLE II
ACCURACY OF PREDICTION MODELS (EMA, COM, AND LNN) WITH CHANNEL-INDEPENDENT TRAINING ON CHANNELS 1, 5, 9, AND 13

Training / Tests	Prediction models (M)	Model param. (ψ^*_M)	μ_{e^2}	e^2_{p95} [$\cdot 10^{-3}$]	e^2_{max}	$\mu_{\lvert e\rvert}$	$\sigma_{\lvert e\rvert}$	$\lvert e\rvert_{p90}$ [%]	$\lvert e\rvert_{p95}$ [%]	$\lvert e\rvert_{p99}$ [%]	$\lvert e\rvert_{max}$ [%]	e_{min} [%]	e_{p5} [%]	e_{p95} [%]	e_{max} [%]
ch1 / all	EMA	$\alpha^* = 0.000325$	2.10	10.74	114.37	2.68	3.71	6.97	10.36	18.14	33.82	-32.11	-6.73	7.25	33.82
	COM	$m = N_e = 4$	1.99	10.46	111.03	2.71	3.55	7.31	10.23	16.76	33.32	-32.95	-7.76	6.91	33.32
	LNN	–	1.97	10.25	108.93	2.72	3.51	7.28	10.12	16.48	33.00	-32.99	-7.88	6.77	33.00
ch1 / $\overline{\text{ch1}}$	EMA	$\alpha^* = 0.000150$	2.56	13.54	115.81	2.96	4.11	8.02	11.64	20.70	34.03	-33.64	-8.28	7.71	34.03
	COM	$m = N_e = 4$	2.14	11.45	111.29	2.77	3.70	7.52	10.70	17.47	33.61	-33.61	-8.12	7.09	33.42
	LNN	–	2.13	11.39	111.49	2.80	3.67	7.51	10.67	17.18	33.39	-3339	-8.32	6.97	32.86
ch5 / all	EMA	$\alpha^* = 0.000325$	1.24	2.63	200.70	1.81	3.03	3.62	5.14	14.65	44.80	-44.80	-3.49	3.74	36.84
	COM	$N_{\alpha^*} = 4$	1.00	1.91	193.97	1.70	2.66	3.32	4.38	11.11	44.04	-44.04	-3.30	3.33	38.37
	LNN	–	0.99	1.93	192.69	1.71	2.63	3.31	4.39	10.67	43.90	-43.90	-3.07	3.46	37.90
ch5 / $\overline{\text{ch5}}$	EMA	$\alpha^* = 0.000475$	1.31	2.62	212.68	1.80	3.14	3.56	5.11	15.26	46.12	-44.95	-3.50	3.61	46.12
	COM	$m = N_e = 4$	1.06	2.22	202.16	1.80	2.71	3.54	4.72	11.06	44.96	-44.96	-3.43	3.63	42.01
	LNN	–	1.02	1.99	192.73	1.73	2.69	3.33	4.46	11.37	43.90	-43.90	-2.96	3.56	41.33
ch9 / all	EMA	$\alpha^* = 0.000325$	4.43	18.88	251.93	2.60	6.12	4.44	13.74	34.45	50.19	-50.19	-3.62	5.03	45.90
	COM	$m = N_e = 4$	3.43	4.96	257.83	2.39	5.35	4.03	7.04	31.99	50.78	-50.78	-3.48	4.40	47.17
	LNN	$\alpha^* = 0.000325$	3.39	5.92	259.84	2.37	5.32	3.90	7.70	31.58	50.97	-50.97	-3.50	4.21	45.64
ch9 / $\overline{\text{ch9}}$	EMA	$\alpha^* = 0.00045$	4.74	18.64	297.14	2.61	6.37	3.95	13.65	37.61	54.51	-50.43	-3.47	4.38	54.51
	COM	$m = N_e = 5$	3.59	5.73	256.26	2.39	5.49	3.87	7.57	32.58	50.62	-50.62	-3.44	4.22	50.11
	LNN	$\alpha^* = 0.00045$	3.50	6.17	260.32	2.39	5.41	3.82	7.86	32.26	51.02	-51.02	-3.48	4.08	48.52
ch13 / all	EMA	$\alpha^* = 0.000325$	1.22	5.82	40.55	2.42	2.52	5.55	7.63	12.16	20.14	-17.44	-5.27	5.85	20.14
	COM	$m = N_e = 4$	1.11	4.83	42.32	2.45	2.27	5.30	6.95	10.70	20.57	-16.16	-5.07	5.58	20.57
	LNN	$\alpha^* = 0.000325$	1.13	4.71	39.35	2.52	2.23	5.33	6.86	10.61	19.84	-16.66	-5.49	5.08	10.84
ch13 / $\overline{\text{ch13}}$	EMA	$\alpha^* = 0.000275$	1.23	5.76	40.72	2.43	2.52	5.60	7.59	12.11	20.18	-16.90	-5.31	5.92	20.18
	COM	$m = N_e = 2$	1.16	4.92	38.01	2.51	2.29	5.43	7.01	10.75	19.50	-16.32	-5.27	5.63	19.50
	LNN	$\alpha^* = 0.000275$	1.31	5.02	36.54	2.83	2.26	5.75	7.08	10.54	19.12	-17.60	-6.34	4.36	19.11

while the second and third bars represent two new experimental conditions. As can be seen, enlarging the pole range does not lead to better results, while narrowing it is typically pejorative. In the LNN case, setting $N_l = N_u = 40$ leads to a slightly worse accuracy. Likely, this is because the newly added poles (α_j values) are not useful for prediction, and so they only have the effect of making convergence of the ANN-based model more difficult.

The latter two experiments are characterized by parameters $r = 2, N_l = N_u = 10$, and $r = \sqrt[4]{2}, N_l = N_u = 40$, respectively. They are aimed at investigating the effect of the granularity of smoothing factors α_j, leaving their range width unchanged with respect to the baseline. As expected, lowering the number of poles leads to a degradation of the ability of the model to predict the FDR (last but one bar in every set of bars in Fig. 5). Conversely, an increase in their number achieves a slight increase in accuracy (last bars of every set). This improvement is, however, irrelevant compared to the additional model complexity, and is inconvenient in many application contexts. In conclusion, prediction accuracy loosely depends on the parameters used to determine model configuration, as long as they are selected near their optimum values.

Regarding LNN, parameters, such as the learning rate, batch size, and epochs, were selected as the result of extensive simulation campaigns aimed at finding the best settings for the training of this kind of models.

C. Channel-Independent Models and Generalization

Results about prediction accuracy when channel-independent training is exploited are reported in Table II. For any test dataset (one per channel), two distinct training datasets were considered. The first, we denote "all", is obtained by merging every training dataset, i.e., $\mathcal{D}_{\text{tr}}^{\text{all}} = \mathcal{D}_{\text{tr}}^{\text{ch1}} \cup \mathcal{D}_{\text{tr}}^{\text{ch5}} \cup \mathcal{D}_{\text{tr}}^{\text{ch9}} \cup \mathcal{D}_{\text{tr}}^{\text{ch13}}$. This single dataset, which summarizes the behavior of all the considered channels, is employed to train all models, reasonably mimicking a *generalized* channel-independent training.

However, some residual correlation exists between $\mathcal{D}_{\text{tr}}^{\text{all}}$ and every test dataset $\mathcal{D}_{\text{te}}^{\text{ch}}$, as the former includes the training dataset $\mathcal{D}_{\text{tr}}^{\text{ch}}$ acquired on the same channel ch. To remove the effects of such correlation, we performed an additional set of experiments where the training dataset, we denote $\mathcal{D}_{\text{tr}}^{\overline{\text{ch}}}$, explicitly omits the samples of the channel ch on which test is made, in formulas, $\mathcal{D}_{\text{tr}}^{\overline{\text{ch}}} = \mathcal{D}_{\text{tr}}^{\text{all}} \setminus \mathcal{D}_{\text{tr}}^{\text{ch}}$. On the one hand, $\mathcal{D}_{\text{tr}}^{\overline{\text{ch}}}$ is completely uncorrelated from $\mathcal{D}_{\text{tr}}^{\text{ch}}$, which makes training *truly generalized*. On the other hand, training coverage is worse, as the behavior of one channel was ignored.

Again, the best accuracy is achieved by COM and LNN, the latter showing a slight advantage. This confirms that, once properly trained, multipole IIR low-pass filters manage to improve prediction accuracy tangibly, by tackling the different dynamics of interference over the air.

When comparing models by varying the training procedure (specialized, generalized, or truly generalized), one can see that, as expected, the first provides the best accuracy, as optimization is carried out in conditions that mostly resemble those under which the model is then operated. It is also clear that removing the channel under test from training negatively impacts accuracy. Therefore, if accuracy is relevant, channel-dependent models are always preferable, although they (repeatedly) require custom training on site.

Interestingly, in the same conditions (same test dataset), both COM and LNN with generalized training behaved slightly better than EMA with a specific training. This is a very important result, because channel-independent models are way simpler and cheaper than channel-dependent ones, and can be easily included in commercial products. For example, the generalized COM model we used to test all channels in Table II has only four poles and is described by attenuations $\boldsymbol{\alpha} = (8.125 \cdot 10^{-5}, 5.792 \cdot 10^{-5}, 1.1483 \cdot 10^{-4}, 5.201 \cdot 10^{-3})$ and weights $\boldsymbol{\lambda} = (0.2759, 0.0857, 0.2022, 0.4362)$. It should be noted that the forecasting error of EMA roughly resembles an SMA evaluated on $N_{\text{p}} = 2/\alpha - 1$ samples. Hence, the above COM filter approximately behaves as a linear combination of averages computed over four past periods, corresponding to 24 615, 34 532, 17 412, and 383 samples, with durations of 3 h : 25 min, 4 h : 48 min, 2 h : 25 min, and 3 min : 11 s, respectively.

D. Comparison between COM and LNN

To better analyze the differences between multipole filters (COM and LNN), the cumulative distribution function (CDF) of the absolute prediction error $|e|$ has been computed for the different kinds of training (channel-dependent/-independent/-truly independent). We considered channels 9 and 5: the former was the most problematic one from the point of view of predictability, as witnessed by the high percentiles on the absolute error ($|e|_{\text{p99}}$), which were by far the worst

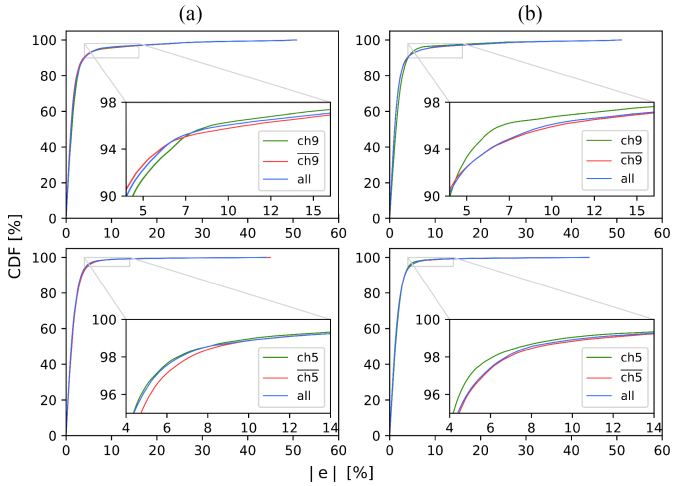


Fig. 6. CDFs of the absolute prediction error $|e|$ on channels 9 and 5 for COM and LNN (channel-dependent/-independent/-truly independent training). (a) COM. (b) LNN.

across all experiments. By comparison, also channel 1 often resulted in gross inaccuracies, but not as large as channel 9. Channel 5 showed instead a generally foreseeable behavior concerning most metrics, with the exception of the maximum.

Results are reported in the four plots of Fig. 6 (channel 9 in the upper part and channel 5 below), where a specific portion of interest has been zoomed-in to provide more details. The 90, 95, and 99 percentiles correspond to the abscissa of the intersection points between plots and the relevant percentages (in the ordinate). As can be seen, thanks to the effectiveness of the ANN training technique, which relies on gradient descent and backpropagation, LNN was able to get the most benefits from specialized training performed in the intended operating conditions (on the same channel as test, see the green line), by lowering the likelihood to make large errors. COM was seemingly less affected by training, and its accuracy in the channel-dependent case was not remarkably better than in the channel-independent one. Differences are however small.

It is important to point out that in harsh conditions, such as those encountered on channel 9, both COM and LNN provide tangible improvements over EMA: see, e.g., the 95 percentile in Tables I and II, where the absolute prediction error is always consistently lower, and practically cut in half when solutions based on channel-independent training are exploited.

E. Impact of the Future Interval

To provide some insights on the effects of the width T_{f} of the reference interval on which the FDR is computed, prediction errors were evaluated on the same datasets by choosing N_{f} in $\{60, 120, 360, 600, 1200, 3600\}$, which correspond to durations in the range from 30 s to 30 min. We considered models with channel-dependent training as they offer the best accuracy. Results for two channels with quite different behavior (1 and 5), which include the optimal attenuation α^* for EMA as well as the MSE μ_{e^2} for both EMA and COM, are reported in Table III (LNN is quite similar to COM).

TABLE III
ACCURACY OF EMA AND COM VERSUS REFERENCE INTERVAL WIDTH

N_f	T_f	ch1 α^*	ch1 μ_{e^2} [$\cdot 10^{-3}$] EMA	ch1 μ_{e^2} [$\cdot 10^{-3}$] COM	ch5 α^*	ch5 μ_{e^2} [$\cdot 10^{-3}$] EMA	ch5 μ_{e^2} [$\cdot 10^{-3}$] COM
60	30 s	0.008120	2.404	2.400	0.007860	1.687	1.684
120	1 min	0.007480	1.627	1.622	0.007580	1.112	1.111
360	3 min	0.006280	1.308	1.239	0.007040	0.848	0.800
600	5 min	0.005600	1.425	1.310	0.006720	0.890	0.806
1200	10 min	0.004640	1.863	1.602	0.006280	1.123	0.932
3600	30 min	0.000900	2.030	1.870	0.000085	1.150	0.960

TABLE IV
COMPUTATIONAL COMPLEXITY OF MODELS ($N_e = 6$ AND $N_s = 41$)

Models	Training time	Mean response time	Memory footprint
EMA	-	5.6 ns	8 B
COM	435 s	28 ns	72 B
COM'	388 s	179 ns	492 B
LNN	601 s	179 ns	496 B

Accuracy of EMA on channel 1 improves when the reference interval for FDR is shrunk from 30 min to 3 min, since optimization leads to more reactive filters that can track channel quality variations better. However, it worsen tangibly when the interval reaches 30 s, as there are not enough samples and the variability of the random binary process described by outcomes becomes the predominant error source. Despite the different optima obtained for α^* (especially on large windows), the same behavior is also observed for channel 5. Similar results are found for COM, which generally shows better accuracy than EMA but, as expected, fails to reduce errors when the number of samples is too small ($T_f = 30$ s).

F. Computational Complexity

The computational complexity of the proposed methods was analyzed in terms of the *training time* for parameterizing the model, the *response time* for its use, and the *memory footprint*. Experiments reported below were performed on a Linux PC with kernel version 6.8.0-40-generic, equipped with an Intel Core i3-10105 CPU running at 3.70 GHz and 8 GB of DDR4 RAM. Results are reported in Table IV. The training time (not including the evaluation of α^*, which is the same for all models) is zero for EMA. Training of COM' is faster than COM, as the L-BFGS-B optimizer is invoked just once (and not twice). By contrast, LNN is the slowest one, with a training time of 601 s. To achieve short response times, low computational complexity is a prerequisite for the adoption of the proposed methods in small embedded systems. An implementation based on the C programming language and compiled with GCC 11.4.0 without any specific optimization shows that the mean time taken by COM to perform a single prediction (28 ns) is extremely short, especially if compared to what is needed by COM' and LNN (179 ns).

Regarding memory footprint, the EMA model only requires two floating points (8 B) to store its two parameters (previous prediction y_{i-1} and smoothing factor α). For the COM and COM' models, memory occupation is $m \cdot 8\text{B} + \text{m} \cdot 4\text{B}$, where $m = N_e$ for COM and $m = N_s$ for COM' (the second term refers to the space to store weights λ_j). Finally, the memory footprint for LNN is $m \cdot 8\text{B} + (\text{m} + 1) \cdot 4\text{B}$, where $m = N_s$ (the quantity $m + 1$ is the number of weights plus the bias of the output neuron).

G. Comparison with DL and Autoregressive Integrated Moving Average (ARIMA) Models

One may wonder how COM and LNN compare to advanced DL models, such as CNN, long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM). These models were evaluated in [42] using the same datasets as those we used in this study, performing both channel-dependent and channel-independent training (not the truly generalized one). By checking results in that article against those presented here, it can be seen that the prediction accuracy is similar, with EMA-based models often behaving better than DL ones. This is particularly apparent for channel 13, characterized by high variability over time, and is probably due to the fact that individual (Boolean) transmission outcomes x_i were directly fed as input features in CNN (3600 samples) as well as in LSTM and Bi-LSTM (1200 samples). As a consequence, performing training once and for all may result in suboptimal performance. Conversely, both COM and LNN rely on multiple EMA filters, which offer greater adaptability to channel variations.

The key distinction between EMA-based and DL models lies in their computational complexity. While response time in COM and LNN was always less than 180 ns, a single-test operation with DL was many orders of magnitude higher (it took up to 5.2 ms for CNN and up to 78 ms for the more complex Bi-LSTM). Also, memory footprint differs significantly, with CNN requiring 30 kB and Bi-LSTM about 700 kB. This makes it clear that unlike COM and LNN, DL modes can be hardly embedded in inexpensive Wi-Fi equipment.

The classical ARIMA model was finally evaluated on the same datasets, but results have not been reported because they are sensibly worse than those obtained by the proposed methods.

V. CONCLUSION

Data-driven techniques for predicting spectrum conditions in the immediate future with adequate confidence are among the primary keys for improving dependability and determinism of wireless communication technologies in general, and Wi-Fi in particular. In fact, information about the current trend of the disturbance affecting a given link (as described by the related FDR) can be exploited, at the MAC and application levels, to counteract its negative effects on communication quality.

Several proposals have appeared on this topic in the past years. In this article, we explicitly focus on solutions that enable very simple implementations, and can be thus incorporated inexpensively in commercial-off-the-shelf equipment. In particular, we analyzed three low-pass filters that are fed with the outcomes of transmission attempts. The first, based on EMA, is characterized by an extremely simple single-pole transfer function, whose cutoff frequency is selected by means of a preliminary data-driven training phase. The latter two, we term COM and LNN, rely instead on multipole transfer functions. They differ in the way the training is carried out. In the COM model, a gradient-based general-purpose technique is applied to minimize the MSE of predictions. Instead, the LNN model relies on the same training technique used for neural networks. Experimental results highlight that, most of the times, both COM and LNN provided better accuracy than EMA (LNN behaving slightly better than COM), still remaining simple enough to keep implementation cost low.

Another goal of this research work was to determine how much a specialized training, performed in the expected target operating conditions, improves accuracy over a generalized training. Results confirmed that the latter approach was slightly worse, but remained nonetheless accurate enough. This shows that simple prediction models, pretrained by the producer of Wi-Fi equipment, may offer valuable performance improvements, yet keeping implementation costs low at the same time. Future work will aim to improve accuracy without significantly increasing complexity, e.g., by including additional input features, such as timings about ACK frame reception and ACK timeout expiration. In addition, the ability of the models to adapt to the variability of the channel conditions (e.g., by using incremental learning and feedback-based model refinement), or their use on datasets acquired with node mobility, could be analyzed as well. Finally, we plan to consider also other classic predictors, such as Kalman filtering.

REFERENCES

[1] T. M. Chiwewe, C. F. Mbuya, and G. P. Hancke, "Using cognitive radio for interference-resistant industrial wireless sensor networks: An overview," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1466–1481, Dec. 2015.

[2] R. Candell, M.Kashef J.Hany Perez-Ramirez, and J. Conchas, "An IEEE standard for industrial wireless performance evaluation," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 82–88, Mar. 2023.

[3] HMS-Networks, "Industrial network market shares 2024," 2024. [Online]. Available: https://www.hms-networks.com/news/news-details/17-06-2024-annual-analysis-reveals-steady-growth-in-industrial-network-market

[4] R. Kunst, L. Avila, A. Binotto, E. Pignaton, S. Bampi, and J. Rochol, "Improving devices communication in industry 4.0 wireless networks," *Eng. Appl. Artif. Intell.*, vol. 83, pp. 1–12, 2019.

[5] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Ann. Operations Res.*, vol. 308, no. 1, pp. 125–143, Jan. 2022.

[6] S. Scanzio et al., "IEEE access special section: Advances on high performance wireless networks for automation and IIoT," *IEEE Access*, vol. 11, pp. 142748–142753, 2023.

[7] *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2024 (Revision of IEEE Std 802.11-2020), 2025.

[8] J. M. Batalla, "On analyzing video transmission over wireless WiFi and 5G C-band in harsh IIoT environments," *IEEE Access*, vol. 8, pp. 118534–118541, 2020.

[9] "Industrial networks - Wireless communication network and communication profiles - WIA-FA," IEC 62948, 2017.

[10] L. Galati-Giordano, G. Geraci, M. Carrascosa, and B. Bellalta, "What will Wi-Fi 8 Be? A Primer on IEEE 802.11bn Ultra High Reliability," *IEEE Commun. Mag.*, vol. 62, no. 8, pp. 126–132, 2024.

[11] G. Cena, S. Scanzio, and A. Valenzano, "Seamless link-level redundancy to improve reliability of industrial Wi-Fi networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 608–620, Apr. 2016.

[12] F. Wilhelmi, L. Galati-Giordano, G. Geraci, B. Bellalta, G. Fontanesi, and D. Nuñez, "Throughput analysis of IEEE 802.11bn coordinated spatial reuse," in *Proc. IEEE Conf. Standards Commun. Netw.*, 2023, pp. 401–407.

[13] J. Perez-Ramirez, O. Seijo, and I. Val, "Time-critical IoT applications enabled by Wi-Fi 6 and beyond," *IEEE Internet Things Mag.*, vol. 5, no. 3, pp. 44–49, Sep. 2022.

[14] A. Al-Tahmeesschi, K. Umebayashi, H. Iwata, J. Lehtomäki, and M. López-Benıtéz, "Feature-based deep neural networks for short-term prediction of WiFi channel occupancy rate," *IEEE Access*, vol. 9, pp. 85645–85660, 2021.

[15] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "DeepChannel: Wireless channel quality prediction using deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 443–456, Jan. 2020.

[16] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118112–118124, 2019.

[17] A. Palaios et al., "Machine learning for QoS prediction in vehicular communication: Challenges and solution approaches," *IEEE Access*, vol. 11, pp. 92459–92477, 2023.

[18] A. S. Colletto, S. Scanzio, G. Formis, and G. Cena, "On the use of artificial neural networks to predict the quality of Wi-Fi links," *IEEE Access*, vol. 11, pp. 120082–120094, 2023.

[19] S. Scanzio, F. Xia, G. Cena, and A. Valenzano, "Predicting Wi-Fi link quality through artificial neural networks," *Internet Technol. Lett.*, vol. 5, no. 2, 2022, Art. no. e326.

[20] D. Li, Y. Xu, M. Zhao, J. Zhu, and S. Zhang, "Knowledge-driven machine learning and applications in wireless communications," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 454–467, Jun. 2022.

[21] W. Tashan, I. Shayea, S. Aldirmaz-Çolak, O. A. Aziz, A. Alhammadi, and Y. I. Daradkeh, "Advanced mobility robustness optimization models in future mobile networks based on machine learning solutions," *IEEE Access*, vol. 10, pp. 111134–111152, 2022.

[22] J. D. Herath, A. Seetharam, and A. Ramesh, "A deep learning model for wireless channel quality prediction," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[23] Q. Cui, Z. Zhang, Y. Shi, W. Ni, M. Zeng, and M. Zhou, "Dynamic multichannel access based on deep reinforcement learning in distributed wireless networks," *IEEE Syst. J.*, vol. 16, no. 4, pp. 5831–5834, Dec. 2022.

[24] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using RF data: A review," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 77–100, First Quarter 2023.

[25] O. Stenhammar, G. Fodor, and C. Fischione, "A comparison of neural networks for wireless channel prediction," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 235–241, Jun. 2024.

[26] W. Li, G. Chen, X. Zhang, N. Wang, S. Lv, and J. Huang, "When industrial radio security meets AI: Opportunities and challenges," *IEEE Trans. Ind. Informat.*, vol. 20, no. 10, pp. 11854–11865, Oct. 2024.

[27] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation for D2D-Aided digital twin edge networks in 6G industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 7228–7236, May 2023.

[28] S. Szott et al., "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1843–1893, Third Quarter 2022.

[29] C. Nguyen, T. M. Hoang, and A. A. Cheema, "Channel estimation using CNN-LSTM in RIS-NOMA assisted 6G network," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 43–60, May 2023.

[30] N. H. Mahmood, G. Berardinelli, E. J. Khatib, R. Hashemi, C. De Lima, and M. Latva-aho, "A functional architecture for 6G special-purpose industrial IoT networks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 2530–2540, Mar. 2023.

[31] K. Hassan, M. Masarra, M. Zwingelstein, and I. Dayoub, "Channel estimation techniques for millimeter-wave communication systems: Achievements and challenges," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1336–1363, Aug. 2020.

[32] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, First Quarter 2020.

[33] M. A. Khan, R. Hamila, N. A. Al-Emadi, S. Kiranyaz, and M. Gabbouj, "Real-time throughput prediction for cognitive Wi-Fi networks," *J. Netw. Comput. Appl.*, vol. 150, 2020, Art. no. 102499.

[34] M. A. Khan, R. Hamila, A. Gastli, S. Kiranyaz, and N. A. Al-Emadi, "ML-Based handover prediction and AP selection in cognitive Wi-Fi networks," *J. Netw. Syst. Manage.*, vol. 30, no. 4, Aug. 2022, Art. no. 72.

[35] G. Famitafreshi, M. S. Afaqui, and J. Melià-Seguı, "Introducing reinforcement learning in the Wi-Fi MAC layer to support sustainable communications in e-health scenarios," *IEEE Access*, vol. 11, pp. 126705–126723, 2023.

[36] A. Nikoukar, Y. Shah, A. Memariani, M. Güneş, and B. Dezfouli, "Predictive interference management for wireless channels in the Internet of Things," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2020, pp. 1–7.

[37] B. Sliwa, N. Piatkowski, and C. Wietfeld, "LIMITS: Lightweight machine learning for IoT systems with resource limitations," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–7.

[38] G. Formis, S. Scanzio, G. Cena, and A. Valenzano, "Predicting wireless channel quality by means of moving averages and regression models," in *Proc. IEEE 19th Int. Conf. Factory Commun. Syst.*, 2023, pp. 1–8.
[39] S. Scanzio et al., "Multi-link operation and wireless digital twin to support enhanced roaming in next-gen Wi-Fi," in *Proc. IEEE 20th Int. Conf. Factory Commun. Syst.*, 2024, pp. 1–4.
[40] G. Formis, S. Scanzio, G. Cena, and A. Valenzano, "Linear combination of exponential moving averages for wireless channel prediction," in *Proc. IEEE 21st Int. Conf. Ind. Informat.*, 2023, pp. 1–6.
[41] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Prog.*, vol. 45, pp. 503–528, 1989.
[42] G. Formis, A. Ericson, S. Forsstrom, K. Thar, G. Cena, and S. Scanzio, "Improving Wi-Fi network performance prediction with deep learning models," in *Proc. IEEE 34th Int. Symp. Ind. Electron.*, 2025, pp. 1–8.



Gabriele Formis (Member, IEEE) received the B.Sc. degree in mechanical engineering and the M.Sc. degree in automation and control engineering from the Politecnico di Milano, Milan, Italy, in 2018 and 2020, respectively. He is currently working toward the Ph.D. degree in artificial intelligence with the Politecnico di Torino, Torino, Italy.

He is currently a Research Associate with the Institute of Electronics, Computer, and Telecommunication Engineering, National Research Council of Italy, Turin, Italy. His research interests include artificial intelligence, wireless networks, and autonomous driving.



Gianluca Cena (Senior Member, IEEE) received the M.S. degree in electronic engineering and the Ph.D. degree in information and system engineering from the Politecnico di Torino, Torino, Italy, in 1991 and 1996, respectively.

Since 2005, he has been a Director of Research with the National Research Council of Italy, Turin, Italy. He has coauthored approximately 170 papers and one international patent on real-time protocols, automotive networks, and wired/wireless industrial communication systems.

Dr. Cena was the recipient of the Best Paper Award of IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS in 2017 and IEEE Workshop on Factory Communication Systems in 2004, 2010, 2017, 2019, and 2020. He was a Program Co-Chair of IEEE Workshop on Factory Communication Systems in 2006 and 2008. Since 2009, he has been an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.



Lukasz Wisniewski (Senior Member, IEEE) received the M.Sc. degree in informatics from the Technical University of Opole, Opole, Poland, in 2007, and the Ph.D. (Dr.-Ing.) degree in communication engineering from the Faculty of Integrated Automation, Otto von Guericke University, Magdeburg, Germany, in 2016.

Since 2008, he has been with the Institute Industrial IT (inIT), Technische Hochschule OWL (TH-OWL), Lemgo, Germany. Since 2019, he has been an Executive Board Member of inIT. In 2022, he became a Full Professor at the TH-OWL. In 2024, he became a Deputy Director of inIT. He has coauthored more than 100 articles.

Dr. Wisniewski currently chairs the area of technologies of digital transformation. He is regularly involved in the organization bodies of several conferences of the IEEE Industrial Electronics Society, such as WFCS, ETFA, and INDIN.



Stefano Scanzio (Senior Member, IEEE) received the Laurea and Ph.D. degrees in computer science from the Politecnico di Torino, Turin, Italy, in 2004 and 2008, respectively.

Since 2009, he has been with the National Research Council of Italy, where he is currently a Senior Researcher with the institute CNR-IEIIT, Milan, Italy. He is a Teacher of several courses on computer science. He has authored or coauthored more than 110 papers in international journals and conferences in the areas of industrial communication systems, real-time networks, wireless networks, and artificial intelligence. He took part in the program and organizing committees of many international conferences of primary importance in his research areas.

Dr. Scanzio was the recipient of the 2017 Best Paper Award from IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and four Best Paper Awards in conferences of the IEEE Industrial Electronics Society. He is an Associate Editor for IEEE ACCESS, *Ad Hoc Networks* (Elsevier), and *Electronics* (MDPI).