



HAL
open science

Experiment-driven platform for link quality estimation in IEEE 802.11 WLANs

Thierry Arrabal, Marija Stojanova, Isabelle Guérin-Lassous, Joris Picot

► **To cite this version:**

Thierry Arrabal, Marija Stojanova, Isabelle Guérin-Lassous, Joris Picot. Experiment-driven platform for link quality estimation in IEEE 802.11 WLANs. HPSR 2023 - IEEE 24th International Conference on High-Performance Switching and Routing, Jun 2023, Albuquerque, NM, United States. pp.1-6. hal-04089954

HAL Id: hal-04089954

<https://hal.science/hal-04089954>

Submitted on 5 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Experiment-driven platform for link quality estimation in IEEE 802.11 WLANs

Thierry Arrabal*, Marija Stojanova[†], Isabelle Guerin-Lassous[‡] and Joris Picot[‡]

* Insa Lyon, Inria, Citi lab, France.

[†] ENS Paris, Laboratoire de Géologie, France.

[‡] Univ Lyon, UCB Lyon 1, ENS Lyon, Inria, CNRS, LIP UMR 5668, France.

Abstract—Experimental link quality assessment of recent Wi-Fi networks remains a challenge due to the rapid development of the Wi-Fi technology, the lack of availability of public datasets, and the difficulty to build such datasets. This paper addresses all three issues by first providing a publicly-available dataset using a custom-made Wi-Fi 5 experimental testbed. We then present an open-source framework for estimating the Frame Delivery Ratio (FDR) of a Wi-Fi link using the experimental data. The proposed solution relies on a small number of input features to build an estimation model of high accuracy, with an R2 coefficient of 0.89 and a mean absolute error of 0.06.

I. INTRODUCTION

The Wi-Fi technology, based on the IEEE 802.11 standard, is constantly undergoing many changes (including Wi-Fi 4, Wi-Fi 5, and Wi-Fi 6), and is now used in a variety of contexts ranging from local networks (home, universities, shopping malls, etc.) to fleets of UAVs (Unmanned Aerial Vehicles), vehicular networks, or IoT (Internet of Things). Wi-Fi performance is impacted by many factors including its environment and potential interferences (from Wi-Fi or other radio networks), or the distance and antenna orientation between the communicating devices. The performance can vary greatly, both on short and long time scales. Some applications rely on low-level performance metrics in order to properly operate, *a.k.a.*, the performance obtained at the physical and data link layers. In this study, we focus on the Frame Delivery Ratio (FDR) of a Wi-Fi link consisting of one source node and one destination node. The FDR is also an indicator of throughput and packet delay, as frame losses imply frame retransmissions and thus delays in the segment transmission times.

Monitoring the FDR is not an easy task. As far as we know, there are currently no simple tools available for the FDR monitoring during application run-time and it requires either to modify the Wi-Fi driver to collect statistics from the Wi-Fi interfaces or to deploy at least one extra Wi-Fi interface configured in monitor mode, in addition to the Wi-Fi interfaces of the source and

destination nodes. In monitor mode, a Wi-Fi interface can collect all the frames it detects, regardless of their destination, from which the FDR can be computed. Both methods have the disadvantage of requiring the modification of the system (at the hardware or software level).

Overcoming these difficulties calls for developing an FDR estimation tool that requires only on-line measurements available at the source and/or destination nodes. But, very few such tools exist for Wi-Fi devices and they are based on datasets collected using old Wi-Fi technologies not using all the features implemented since Wi-Fi 4 which can have an impact on the FDR. In addition, the authors are not aware of the existence of any publicly-available datasets that use recent Wi-Fi versions and provide the full physical and link layer headers. The lack of such datasets makes it difficult to design the needed estimation tools. To bridge this gap, the main contributions presented in this paper can be summarized as follows:

- Deployment of an experimental Wi-Fi platform, in various environments, for the collection of physical and data link layer headers as experimental data, and providing an open-source FDR computation method using that data (see Sec. III).
- Development of an open-source framework for FDR analysis and estimation using Machine Learning (ML) methods in Python (see Sec. IV).
- Publication of the obtained experimental datasets for any future uses [1].

We provide a state of the art on the problem studied in this paper in Sec. II and we conclude in Sec. V.

II. RELATED WORK

Network monitoring tools: Linux commands like *ip* and *iw*, provide networking statistics (such as the number of transmitted and received packets) on a network interface since the interface was set up. It is difficult to

use these commands to obtain statistics on a very short time scale. Wavemon is a Linux tool that provides low-level Wi-Fi parameters, *e.g.*, the MCS index (Modulation and Coding Scheme). However, these statistics are shown in an ncurses window, making it difficult to extract raw data. Monitoring tools, *e.g.*, Wireshark, can capture all the frames received and transmitted by a network interface and provide the data included in all the headers of the different TCP/IP layers. Unfortunately, to get data from the physical and link layers of Wi-Fi frames, the Wi-Fi interface has to be configured in monitor mode preventing it from communicating in the monitored network. This implies using a new interface solely for monitoring purposes, and not for communication. Finally, some Wi-Fi drivers may be modified in order to get statistics on the FDR, however this approach requires a modification of the driver, which is not always straightforward or desirable.

FDR estimation tools: The survey of Gregor et al. [2] lists the different Machine Learning (ML) solutions that estimate a wireless link’s quality. The survey includes only a few works relating to the Wi-Fi technology [3]–[7] and, among them, only two estimate the FDR [3], [7]. Moreover, these works are based on old Wi-Fi versions (the IEEE 802.11a/b/g standard amendments) which are significantly different from Wi-Fi 4/5/6. Scanzio et al. [8] study a Wi-Fi 4-based network, however their FDR estimation method requires active-probing packets and the addition of a specialized Software-Defined MAC for managing the Wi-Fi interface.

Publicly available datasets and testbeds: Gregor et al. [2] provide a list of open-source datasets that can be used to estimate link quality metrics. Several datasets use Wi-Fi networks [9]–[12], however none of them use recent Wi-Fi devices. We found datasets using recent Wi-Fi devices, but they do not provide the entire content of the physical and link headers of the collected frames¹²³ [13]. Concerning the publicly-available testbeds, some of them implement the Wi-Fi technology, *e.g.*, NICTA, w-iLabt, NITOS, NTUA, federated within the Fed4FIRE, FIT wireless, and ORBIT testbeds. Among them, only ORBIT implements Wi-Fi versions 4 or higher. Moreover, these testbeds offer little control in terms of node location and antenna orientation, and only a limited diversity of environments.

¹<https://zenodo.org/record/5616432#.Y34opTOZOV4>

²<https://zenodo.org/record/814898#.Y34saTOZOV4>

³<https://zenodo.org/record/3952557#.Y3ID-76ZOV4>

III. EXPERIMENTAL DATA COLLECTION

Our experiments are focused on estimating the quality of a Wi-Fi link between a source and a destination. The source emits a stream of data which the destination receives over an imperfect radio medium whose quality we wish to assess.

A. Platform

The experiments are performed using a homemade platform, allowing us to have full flexibility in its configuration and location. We use an IEEE 802.11ac (Wi-Fi 5) network operating in the 5 GHz frequency band with a maximum channel bandwidth of 80 MHz. All the technical details on the used Wi-Fi interfaces and the system can be found in [1].

In the interest of keeping a high degree of fidelity to an operating IEEE 802.11 WLAN, the choice of transmission parameters is left up to the default rate adaptation algorithm of the Wi-Fi interface. We use iPerf3 to generate UDP/CBR data flows, tshark to capture the exchanged frames, and Bash and awk scripts to automate experiments, and to extract and format data.

We opted for the approach using a Wi-Fi interface configured in monitor mode⁴ to collect the Wi-Fi physical and link layer headers. Therefore, our platform consists of three nodes: a source, a destination, and a monitor node. The monitor node acts as a sniffer that simply collects all the frames exchanged in its vicinity, but does not participate in any of the communications. The monitor is placed close to the destination in the effort to closely replicate its wireless environment and the frames it receives. Since the monitor is simply an intermediary for the destination, there is of course an inherent measurement error.

B. Experimental setup and environment

One of the main goals in our data collection campaign was to have a diverse set of measurements reflecting the impact of various environments on the FDR. In this section, we describe the different experimental setups, as well as some of the boundaries of those setups. A single experimental setup on our platform is described by:

- 1) the physical environment and Line of Sight (LoS);
- 2) the distance (in meters) between the source and the destination and the position of the nodes’ antennas.

Each experimental setup with given values for the different parameters (physical environment, LoS, distance, antennas’ position) is called a *scenario*. Every scenario is replicated in three independent runs of 120 seconds.

⁴<https://wiki.wireshark.org/CaptureSetup/WLAN>

1) *Physical environment and LoS*: The small size of our experimental platform makes it easy to transport in different environments. As a result, we performed experiments in three different floors of our laboratory. In the three environments, we ran experiments with and without the presence of an obstacle obstructing the LoS between the source and destination. Changing the LoS creates diversity in the experiments at a minimal cost.

It should be noted that during all the experiments, traffic from surrounding networks may exist. The 5GHz band is seldom used in our laboratory, however a small amount of traffic may come from other access points in the building and it impacts the collected measurements. In the interest of system fidelity, we decided to keep the environment in its existing conditions and not try to limit this traffic or the other perturbations created by the movement of our colleagues.

The experiments are always conducted in a corridor, however each floor has different characteristics:

- 1st floor: The corridor is slightly curved, and the nodes are placed in a way that guarantees an LoS between them when an obstacle is not present.
- 2nd floor: The corridor is straight and empty.
- 3rd floor: The corridor is also straight, but different elements are present (e.g., tables along the walls).

2) *Distance and antenna position*: The distance between the source and the destination varies between zero and 26 meters. The maximum distance that permits successful communication depends on the chosen environment and the position of the antennas. The antennas can be in one of the four positions depicted in Figure 1. It was our experience that different floors of our building result in different propagation environments due to their shape and construction.

The segment length is set to 1470 Bytes and the CBR datarate is 1 Gbps, corresponding to a saturated flow.

C. Data extraction

At the end of every experimental run, the monitor has collected a trace containing the traffic exchanged between the source and destination nodes⁵. We extract the following transmission parameters from every data frame captured by the monitor:

- Modulation and Coding Scheme (MCS) index: Higher MCS indexes offer higher datarates, however, they are less robust to a lossy channel.

⁵We recall that the monitor’s trace is merely an approximation of the real stream of traffic exchanged between the source and destination.

- Number of Spatial Streams (NSS) used for the frame transmission: Our nodes can use one, two, or three antennas during a transmission.
- The channel Bandwidth (BW) chosen for the frame transmission: Our nodes can use channels of 20MHz, 40MHz, or 80MHz.
- Datarate: The physical transmission rate at which the frame payload is sent. The datarate is dependent on the chosen MCS, BW, NSS, and several other IEEE 802.11 parameters.
- Received Signal Strength Indicator (RSSI): Indicates the quality of the signal (in dBm) with which the frame was received.

We then also compute several metrics from the trace:

- Number of frames captured by the monitor, denoted *frames_monitor*.
- Number of frames not captured, but expected by the monitor, denoted *missing_monitor*.
- Number of frames acknowledged in the Block Acknowledgements (BACKs), denoted *frames_ack*.
- Number of frames reported as missing in the BACKs, denoted *missing_ack*.
- The link layer throughput computed based on the average frame payload length, denoted as *throughput_monitor*.

These measured parameters and computed metrics constitute our database. In our final database used for building the FDR estimation model, each observation is the average of 100ms of experimental data. Typically, over 600 frames are exchanged during each interval, ensuring that this averaged value is representative of the behaviour during the interval.

D. Link quality metric

The performance metrics of interest in this paper is the Frame Delivery Rate (FDR) defined as:

$$\text{FDR} = \frac{\text{Number of well received frames}}{\text{Number of sent frames}}. \quad (1)$$

However, calculating the FDR from experimental data is far from a straightforward task as both the number of received and sent frames are difficult to accurately compute from traces. Since the monitor acts as an intermediary for the destination, we regularly observed several undesired events:

- Frames are well received by the monitor, however marked as missing in the BACKs sent by the destination.
- Frames are not at all received by the monitor, but they are acknowledged by the destination in the BACKs.

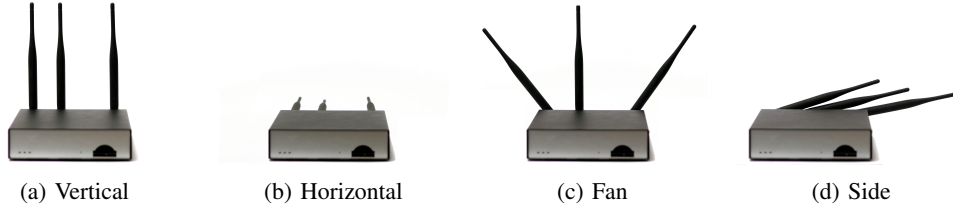


Fig. 1: Possible antenna positions.

- Frames in retransmission do not consistently have the "retry" flag set to True.

In an effort to minimize the effect of these events, we decided to calculate two different FDRs: a first one based on the frames that the monitor received from the source node, and a second one based on the BACKs received from the destination node.

Both FDR versions rely on a mechanism called the Sequence Number (SN)⁶. Every data frame has a single SN that is constant during all retransmissions. A stream of frames from the same flow will have increasing SNs. The SN enables the destination to 1) properly reassemble a stream of frames in case they arrive out of order and 2) acknowledge only the well-received frames and indicate the missing frames in the BACKs sent back to the source. The source can then consult each individual BACK to know which frames were not properly received and only retransmit those frames.

Thus, we compute the FDR using two different versions of Eq. 1, one based on the BACK information:

$$FDR_{ack} = 1 - \frac{\text{Nb. missing frames in BACKs}}{\text{Nb. appeared frames in BACKs}}, \quad (2)$$

and a second one based on data frames' Sequence Numbers (SN):

$$FDR_{seq} = 1 - \frac{\text{Nb. missing SNs for monitor}}{\text{Max received SN} - \text{Min received SN}}. \quad (3)$$

Equation 3 calculates the FDR based on the number of missing sequence numbers in a series of data frames captured by the monitor and the difference between the maximum and minimum SNs received in that series. Equation 2 calculated the FDR using the number of frames missing and the total number of frames appearing in the BACKs captured by the monitor.

IV. DATA ANALYSIS AND PROPOSED SOLUTION

This section introduces a novel framework for a trace-driven performance evaluation of IEEE 802.11 networks,

⁶We only present a high-level explanation of this mechanism, for a detailed review we refer the reader to the IEEE 802.11 standard [14]

developed in Python, making it fully open-source and available online [15].

We focus on 1) identifying the highest-impact *features*, *i.e.*, transmission parameters and computed metrics, 2) building an FDR estimation model based on these features, 3) studying its accuracy, and 4) discussing a posteriori insight we can obtain using the model. The collected traces can be divided into six datasets: experiments with LoS named First, Second, and Third floor, and experiments with no LoS named Obstacle 1st, Obstacle 2nd, and Obstacle 3rd.

A. Correlation study

We first study the Pearson's correlation coefficient between the linear and quadratic features and the FDR. Fig. 2 shows these correlations for both versions of FDR for the full dataset containing all six scenarios. FDR_{ack} does not correlate strongly with any of the other features, and this is true even when considering the different scenarios separately. In the interest of brevity, for the remainder of the paper we focus solely on FDR_{seq} .

FDR_{seq} is strongly correlated with *throughput_monitor* and *frames_monitor*. All the Pearson's coefficients higher than 0.4 are a combination of at least one of these features. Surprisingly, the FDR_{seq} is not as highly affected by the distance between the nodes and the RSSI as one would expect. Many existing works have considered the FDR to be a function of these two features, and it is the authors' opinion that the complexity of MIMO has made the relationship between these features highly unpredictable. We remove the BW and NSS features, as they have almost constant values and weak correlations to the FDR.

B. FDR estimation model

The estimation model we aim to develop needs to be computationally simple so that it can be executed on small Wi-Fi-enabled devices. To this effort, we first limit the number of input features used to build the estimation model. We then choose a model that reflects

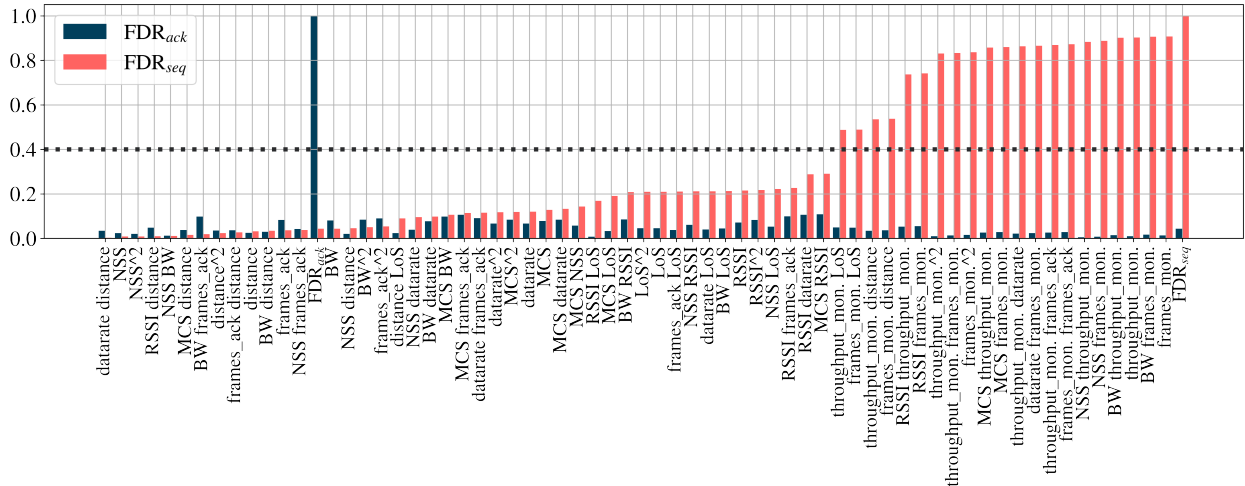


Fig. 2: Pearson’s feature correlations for the FDR_{ack} and FDR_{seq} .

the configuration of our dataset, *i.e.*, a small number of features and a large number of samples.

The model presented has the following characteristics:

- We minimize the number of input features by selecting only the features that have a Pearson’s correlation coefficient of at least 0.4 in at least one of the datasets. These are the MCS index, RSSI, *throughput_monitor*, *frames_monitor*, *datarate*, and *distance*⁷.
- We use Ridge linear regression that has a low complexity in the number of samples, and (potentially) high complexity in the number of features. Ridge regressions also allow us to circumvent any multicollinearity issues that are potentially present in experimental Wi-Fi traces.
- We use five-fold cross validation on 80% of the data, and use the remaining 20% as a test set.
- We use grid search to find the optimal α parameter for the Ridge linear regression. We tested all values in the interval $[0, 1]$ with a step of 0.01 and found an optimal $\alpha = 0.04$.
- We use polynomial features, *i.e.*, our input features are a collection of linear features and all their possible quadratic combinations.
- We make sure that each scenario is represented with an equal number of samples in the complete dataset⁸.

⁷We refer the reader to our Jupyter notebook for a complete correlations heatmap [15].

⁸Our complete dataset contains 91494 samples with 13082 samples per scenario.

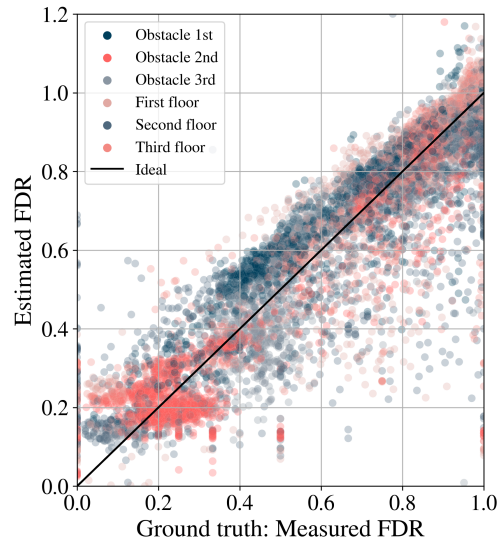


Fig. 3: Estimated FDR_{seq} vs. ground truth FDR_{seq} .

A scatter-plot of the estimation vs. ground truth is shown in Fig. 3. We notice that most of the points lie closely to the ideal estimation, with a Mean Absolute Error (MAE) of 0.06 across the datasets, and an R2 score of 0.89. Given the simplicity of the proposed model and its input features, we believe this accuracy level makes it relevant for many Wi-Fi devices and applications.

C. *A posteriori* data knowledge

We look at the knowledge we can extract from the models built using the different datasets separately. We

train one model per dataset and we only keep the features that have a Pearson correlation coefficient larger than 0.4 in that particular dataset⁹. The MAE is shown in Fig. 4, where the y-axis is the name of the dataset used in training, and the x-axis is the name of the test dataset.

There are several key observations we can make:

- 1) The obstacle datasets are both harder to estimate and less suited for the estimation of other datasets.
- 2) The datasets *First floor* and *Third floor* are the best choice when it comes to estimating the FDR of other datasets. Interestingly, these are also the two simplest models, as they only contain only *frames_monitor* and *throughput_monitor*.
- 3) As expected, the best choice when it comes to overall estimation is to use the complete dataset named here *All Data*.

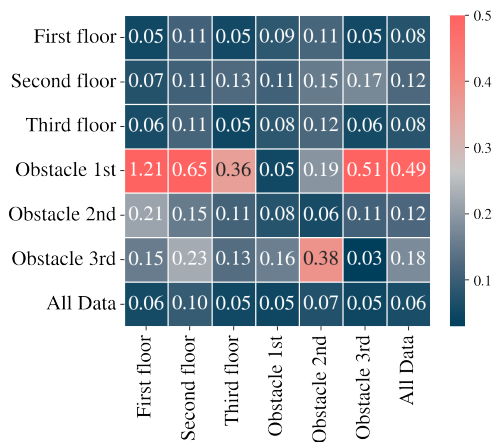


Fig. 4: Mean Absolute Error (MAE) when using the y-axis train dataset to estimate the x-axis test dataset.

V. CONCLUSION

In this work, we have set up a custom-built Wi-Fi 5 experimental platform that collects frames exchanged between a source node and a destination node, with all their headers including the physical and link headers. We have carried out experiments in various environments, distances between nodes, and antennas' positions. We made available a database of all the parameters accessible in the physical and link layer headers, as well as several computed metrics. Using this database, we have provided two methods for computing the FDR and a separate correlation study, from which we have built a simple FDR estimation model. We have shown that

⁹Separate training and testing datasets are always used even when working on a single scenario.

using the complete dataset and the proposed estimation model, we can estimate the FDR obtained in different environments with high accuracy.

As a future work, we plan to conduct additional experiments in other environments (anechoic chamber, in a fleet of UAVs) and with other traffic profiles (in terms of stream datarate and segment size, and also with TCP) in order to complement the database.

VI. ACKNOWLEDGEMENTS

The authors would like to thank the ANR and the AID for their funding support via the CONCERTO project (ANR-20-ASTR-0003) and the CNRS for funding a research engineer position.

REFERENCES

- [1] T. Arrabal, M. Stojanova, I. Guérin Lassous, and J. Picot, "Wi-Fi 5 packets capture," 2022. [Online]. Available: <https://entrepot.recherche.data.gouv.fr/dataset.xhtml?persistentId=doi:10.57745/T5YEMM>
- [2] G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "Machine learning for wireless link quality estimation: A survey," *IEEE Comm. Surveys & Tutorials*, 2021.
- [3] K. Srinivasan, M. A. Kazandjieva, M. Jain, and P. Levis, "PRR is not enough," 2008.
- [4] P. Millan, C. Molina, E. Medina, D. Vega, R. Meseguer, B. Braem, and C. Blondia, "Time series analysis to predict link quality of wireless community networks," *Computer Networks*.
- [5] M. L. Bote-Lorenzo, E. Gómez-Sánchez, C. Mediavilla-Pastor, and J. I. Asensio-Pérez, "Online ML algorithms to predict link quality in community wireless mesh netw.," *Computer Networks*.
- [6] G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "On designing a machine learning based wireless link quality classifier," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020.
- [7] G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "Learning to fairly classify the quality of wireless links," in *2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*, 2021.
- [8] S. Scanzio, F. Xia, G. Cena, and A. Valenzano, "Predicting Wi-Fi link quality through artificial neural networks," *Internet Technology Letters*, 2022.
- [9] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-Level Measurements from an 802.11b Mesh Network," ser. SIGCOMM '04.
- [10] S. K. Kaul, M. Gruteser, and I. Seskar, "Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection," in *TRIDENTCOM 2006*.
- [11] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker, "Physical layer attacks on unlinkability in wireless LANs," in *Int. Symp. on Privacy Enhancing Technologies*. Springer.
- [12] E. Anderson, G. Yee, C. Phillips, D. Sicker, and D. Grunwald, "The impact of directional antenna models on simulation accuracy," in *WiOpt 2009*. IEEE.
- [13] S. Barrachina-Muñoz, B. Bellalta, and E. Knightly, "Wi-Fi All-Channel Analyzer," in *WiNTECH 2020*. ACM.
- [14] "IEEE Standard for IT Part 11: Wireless LAN MAC and PHY Specifications," *IEEE Std 802.11-2016*, 2016.
- [15] M. Stojanova, T. Arrabal, I. Guérin Lassous, and J. Picot, "marija-stojanova/hpsr-wifi-link-quality- estimation: HPSR 2023 Link Quality Estimation," 2023. [Online]. Available: <https://doi.org/10.5281/zenodo.7863819>