Data-Driven Prediction Models for Wireless Network Configuration

Samir Si-Mohammed and Fabrice Theoleyre

Abstract Wireless networks are widely used for a wide range of applications, from best-effort object tracking solutions to robot control in smart factories. However, the performance of these networks is highly dependent on their configuration. Worse, the different *links* have heterogeneous characteristics, and a homogeneous configuration is often suboptimal. We believe that digital twins are an excellent tool for achieving autonomous networks that can automatically reconfigure themselves based on conditions. To this end, digital twins of networks must be able to incorporate this heterogeneity into their models and capture the impact of a configuration on the performance of a given radio link. We therefore propose here a link-oriented prediction model, able to predict the expected Packet Reception Rate for a given MAC configuration. Our experimental evaluation demonstrates the relevance of a data-driven prediction method to capture the links specificities.

1 Introduction

Wireless networks are vital to modern communication, facilitating connectivity among devices, systems and people. Their role extends beyond personal communication and entertainment, impacting crucial areas such as healthcare or smart cities. In Industry 4.0, wireless networks are transformative, enabling seamless connectivity for technologies like the Internet of Things (IoT) and real-time data analytics. This connectivity empowers a new generation of services, including predictive maintenance and improved supply chain management.

The growing complexity of wireless networks has highlighted the need for advanced tools to design, monitor, and optimize their performance. Among these tools, Digital Twins (DTs) are emerging as a key enabler, providing a powerful framework to replicate network behavior in a virtual environment. A Digital Twin is a digital representation of a physical entity or system continuously updating and evolving based on real-time data. By leveraging DTs for wireless networking, it becomes possible to explore the interplay of diverse configurations and environmental conditions without extensive real-world testing.

Samir Si-Mohammed
CNRS/University of Strasbourg, France, e-mail: simohammed@unistra.fr

Fabrice Theoleyre
CNRS/University of Strasbourg, France, e-mail: fabrice.theoleyre@cnrs.fr

A critical challenge in this context is to predict the behavior of an unknown configuration based on previously observed data. This challenge is particularly significant for wireless networks (e.g., IEEE 802.15.4) where varying configurations—such as MAC parameters—can result in highly heterogeneous performance outcomes. Generalization in DTs refers to the ability of a model to accurately predict outcomes across different unseen scenarios. In wireless networking, it means the prediction of Key Performance Indicators (KPIs) without having tested the corresponding configurations.

To address these challenges, experimental studies play a crucial role in advancing IoT research by bridging the gap between theoretical models and practical implementations. Several works have leveraged testbeds to evaluate network performance, including latency, throughput, and link heterogeneity [1, 2]. Additionally, extensive research has focused on optimizing MAC parameters in wireless standards like IEEE 802.15.4 and Wi-Fi, employing machine learning techniques to enhance throughput, minimize latency, and improve energy efficiency [3, 4]. These approaches span simulation and real-world deployments, addressing various metrics and configurations for network performance optimization.

In contrast to prior works, we adopt an innovative approach by considering a separate predictive model for each link constituting a wireless network. This granularity accounts for the unique characteristics of each link, resulting in high prediction accuracy and low error rates. Our approach is data-driven to create a building block of the future DT for wireless networks. Our contributions are as follows:

1. **A thorough analysis of an experimental deployment**, providing insights into:
   - The heterogeneity of wireless links.
   - The impact of configurations on network performance.
   - The link-specific nature of configuration effects.
2. **A data-driven prediction method** to estimate the performance impact of previously unseen configurations. We notably show that **a link-specific model reduces significantly prediction errors**: a unified model is insufficient.

2 Related Works

Experimental studies are vital for bridging the gap between theoretical models and real deployments. Some works have investigated the performance of IoT networks using different technologies and testbed environments. For instance, [1] explore RPL over IEEE 802.15.4 networks with a focus on latency. Using the FIT IoT-LAB testbed, they evaluate the delay and throughput of links under varying message sizes and transmission frequencies. Meanwhile, [2] delve into the heterogeneity of FIT IoT-LAB [5] deployments, studying the realism of testbeds by identifying phenomena such as external interference, multi-path fading, and dynamic connectivity. They showcase the links' heterogeneity in the Grenoble site of the platform. Fraile *et al.* [6] provide a detailed experimental study to quantify the impact of the MAC parameters on the network performance. In particular, they highlight that the MAC behavior of LoRa is strongly dependent on the Spreading Factor and the Bandwidth parameters.

The optimization of MAC parameters in wireless standards has attracted much attention in the past. Aboubakar *et al*. [3] use multilayer perceptron, random forest, K-nearest Neighbors (k-NN) and decision trees to identify the optimal values of the MAC parameters, aiming to minimize end-to-end delay. Similarly, Alkaseem *et al*. [7] present an Artificial Neural Network-based approach for estimating optimal IEEE 802.15.4 MAC parameters. Using simulation, they predict configurations that best improve the end-to-end transmission delay. However, these optimization techniques use simulations for training. Unfortunantely, simulators' accuracy strongly depends on the physical layer models' accuracy [8]. PHY models often deviate significantly from real-world conditions [9].

Karmakar *et al*. [4] propose an online learning-based solution for Wi-Fi networks. They tune channel bandwidth, MCS values, and the number of MIMO antennas to achieve high throughput in Wi-Fi-based topologies. In contrast, Chen *et al*. [10] adopt a deep learning approach to adapt the Contention Window. Their data is generated via simulations covering various conditions. Furthermore, [11] introduce an algorithm to dynamically adjust the level of 802.11n frame aggregation by Wi-Fi stations, targeting both QoS optimization and improved energy efficiency, particularly under network congestion. Still, these approaches rely extensively on simulations.

Data-driven methods and digital twins have garnered significant attention for their ability to capture network characteristics and predict potential outcomes through "what-if" scenarios. For example, in [12], the authors introduce a Network Digital Twin designed to optimize the configuration of 802.15.4 networks using a heuristic and an online learning algorithm. In the same vein, Masaracchia et al. [13] highlight that Network Digital Twins (NDT) enable the integration of AI modules that generate insights from real-time data, allowing the system to evolve as deployment progresses by optimizing network operational parameters. They suggest that AI-enabled NDT can support network design by evaluating different scenarios in various contexts to identify the optimal configuration for maximizing QoS. In general, machine learning and statistical techniques have demonstrated their potential in providing relevant and actionable insights by leveraging empirical data. These approaches are especially crucial for generating reliable predictions within NDT frameworks.

3 Problem Statement

Wireless networks are known to be lossy, and retransmissions represent a way to make the communication more reliable. Unfortunately, increasing the number of retransmissions increases congestion and impacts negatively both the latency and the energy efficiency. Similarly, defining the backoff value has also an impact on the collision probability.

The network configuration is most of the time defined when the network is deployed. We propose here to use a digital twin that collects metrics of performance in real-time to predict the behavior of the different links, and thus their optimal configuration. Since the number of configurations increases exponentially with the number of parameters to tune, we need to predict the performance for unknown (i.e., untested) configurations.

We propose here an experiment-driven approach. We leverage a large-scale wireless testbed that runs an IEEE 802.15.4/6LoWPAN stack. IEEE 802.15.4 employs CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) prone to collisions. More specifically, we focus on the following parameters:

- **macMinBE and macMaxBE** (Backoff Exponent), which define the initial and the maximum backoff exponent value, from which is derived the backoff value;
- **macMaxCSMABackoffs**, which limits the number of backoff attempts before a transmission is abandoned. When the medium is still idle after this number of attemps, the packet is dropped;
- **aMaxFrameRetries** which sets the maximum number of retries for a frame in case of transmission failures.

Each of these parameters can take multiple values, creating a vast exploration space of possible configurations ($> 2,000$ configurations), each with potentially significant effects on network performance [12]. Thus, it is crucial to find optimized configurations for specific application needs and environments.

Suppose a network is deployed, and a node has experienced a series of configurations $C = [C_1, \dots, C_n]$, each associated with a corresponding resulting performance metric $KPI = [K_1, \dots, K_n]$. Our objective is to develop a model capable of accurately predicting the performance impact K_m of a configuration C_m that the node has not yet encountered (*i.e.,* $C_m \notin C$).

4 Deployment Analysis

We first characterize the heterogeneous impact of the network configuration on the performance. For this purpose, we use the large-scale FIT IoT-Lab wireless testbed (https://www.iot-lab.info/) to conduct our experiments.

4.1 Experimental Setup

We run a 20-hour experiment on the Grenoble site. We consider a single-hop (cellular) topology where a gateway collects packets from 8 M3 motes. We mimic a realistic smart building scenario, where the different motes are distributed in a corridor, at a different distance from the gateway (cf. Figure 1). The building also hosts other (e.g., Wi-Fi) networks that may create interference. We used the Contiki-NG operating system to run the 802.15.4 protocol stack.

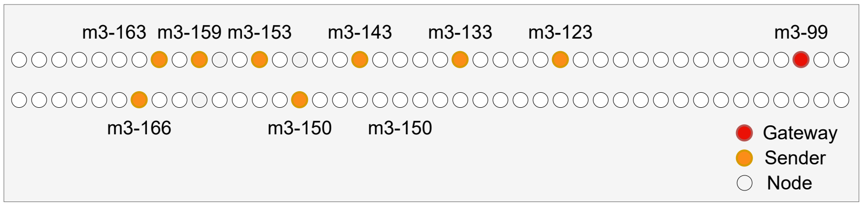

Fig. 1 Nodes placement in the FIT IoT-Lab indoor room in the Grenoble site.

Parameter	Value range
macMinBE	$3-8$
macMaxBE	$4-10$
macMaxCSMABackoffs	$1-8$
aMaxFrameRetries	$1-7$

Table 1 Medium Access Control parameters.

Each mote generates Constant Bit Rate traffic toward the gateway at a rate of one packet per second. We consider the MAC configuration is updated for all the nodes every 100 seconds. It is worth noting that each mote may have a different MAC configuration, but the change is triggered synchronously for all the motes. Table 1 shows the value range for each considered parameter.

As a Key Performance Indicator, we use the number of (re)transmissions required before a packet is delivered from a mote to the gateway. More precisely, we collect for each link ($mote \rightarrow gateway$) two time-series:

1. **Configurations:** $C_l = [C_{l,1}, \ldots, C_{l,k}]$, where $C_{l,i}$ denotes the configuration used by the link l for the i^{th} time interval. We denote by $\mathscr{C}$ the set of all the possible configurations;
2. **Packet Reception Rate:** $PRR_l = [PRR_{l,1}, \ldots, PRR_{l,k}]$, where $PRR_{l,i}$ denotes the average Packet Reception Rate achieved by the transmissions through the link l for the time interval i. More precisely, the PRR is computed with the ratio of the packets received by the gateway, and the packets transmitted through the link l.

4.2 Experimental Characterization of the Diversity

Figure 2 illustrates the reception rate achieved individually per mote for a specific configuration. A Packet Reception Rate close to 1 means that a unique transmission is sufficient to deliver the packet to the gateway. Inversely, red values of the PRR (close to 0) correspond to very unreliable links. For simplicity, each link is referred to by its transmitter (i.e., mote) in the rest of this section.

We can note that a given configuration (e.g., (6, 9, 3, 3)) may be very good for some links (e.g., m3-133) and very bad for other ones (e.g., m3-166). This result may be quite obvious since a bad link provides a high Packet Error Rate at the PHY layer, impacting negatively the reliability. However, it is also interesting to

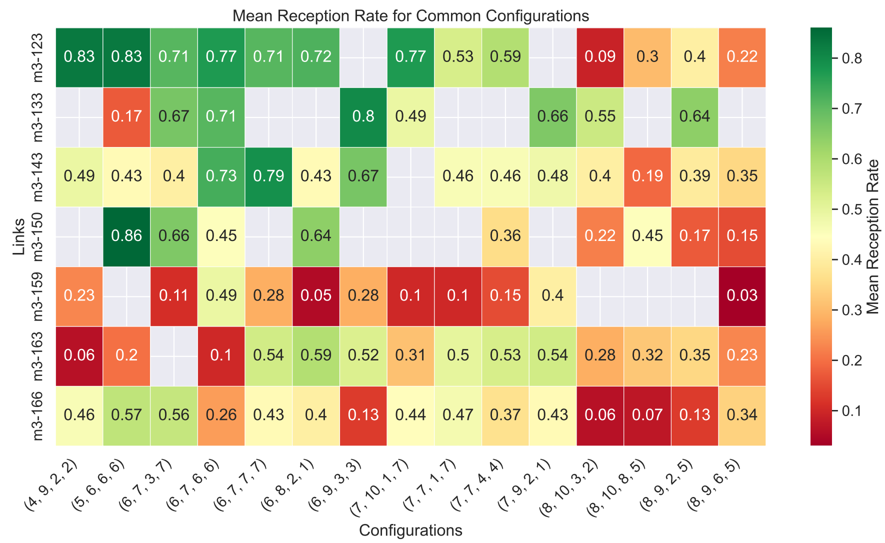

Fig. 2 Mean reception rate for common configurations to different links.

note that the optimal configuration in the considered set is different for each link: m3-123 should use (4, 9, 2, 2) while link m3-166 should use (6, 8, 2, 1). Thus, **the configuration has to be adapted to each link**, which is seldom the case in the literature.

Even more interesting, the PRR varies non-uniformly for all the links. For instance, m3-159 has a much higher PRR for configuration (6, 7, 6, 6), and a bad PRR for contiguous configuration, which is the inverse for m3-159. Thus, **we must implement a per-link strategy**: a model should map the expected PRR to each link and configuration, and this is the objective we tackle in the next section with a data-driven model.

5 Data-driven Link Quality Prediction

In this section, we detail a solution for building a data-driven model to predict configuration performance. We begin by outlining the data preparation process and how it is split into training and testing sets, followed by a description of the model training process.

5.1 Data Preparation

To predict configuration performance accurately, we prepare data as follows:

1. **Mean Reception Rate** ($\overline{PRR}_{l,c}$): the mean *Packet Reception Rate* for the link l and for a specific configuration c. Indeed, a configuration is randomly chosen and may be tested several times (or never) for a specific link:

$$\overline{PRR}_{l,c} = \frac{1}{|\{i \in \mathbb{N}|\ C_{l,i} = c\}|} \sum_{i \in \mathbb{N}| c = C_{l,i}} PRR_{l,i} \tag{1}$$

2. **Input of the Prediction model** consists of the configuration parameters ($c \in \mathscr{C}$) for the link l;
3. **Output of the Prediction model** is the mean reception rate ($\overline{PRR}_{l,c}$) corresponding to the input configuration c.

5.2 Model Training

We target lightweight prediction models: we must minimize the computational complexity of the prediction model. Indeed, one model should be executed continuously for each link. To be sustainable, the Digital Twin must present a reasonable computational complexity. Thus, we use classical regression models to predict the mean reception ratio for a given configuration:

1. **Support Vector Regression** (SVR): it aims to find a hyperplane in a high-dimensional feature space that has the maximum margin from the training data points [14].
2. **Gradient Boosting**: it builds a set of prediction models in a sequential manner. It combines these models to create a more through the reduction from each model of the errors made by the previous models [15],
3. **Decision Trees**: it splits the data into subsets based on the value of input features, creating a tree-like structure where each node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome [16].

These models are particularly appealing due to their low computational complexity, although more complex models. Instead of selecting one model, our Digital Twin acts rather as a meta-model, capable of selecting the most suitable model for each link. Indeed, we are convinced that a trade-off between accuracy and computational complexity exists. Thus, our solution switches models on-the-fly when necessary (see Figure 3).

We explore three distinct approaches to train our prediction models:

1. **Global Regression**: to maximize the training dataset, we consider all the links together. That's a classical assumption in the literature: a generic model is derived to estimate the Key Performance Indicator of the MAC layer from the MAC parameters. A single, generalized model aimed at predicting the mean reception rates for any given link assuming they present all the same behavior. While there are unique characteristics for each link, training on the aggregated data can help capture broad network patterns that apply to all links. Thus, we propose here to exploit training data containing all the links combined in the same dataset. This global dataset serves as the training data for our regression model to predict the expected Packet Reception Rate from the input configuration.

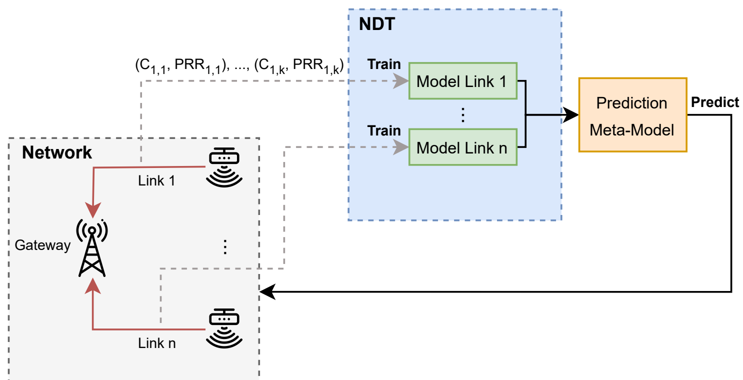

Fig. 3 Prediction models are trained with data coming from the network and can be used to predict the outcome (i.e., Packet Reception Rate) of unexplored configurations.

2. **Single-Link Regression**: we propose to consider each link separately. Thus, each link has its own specialized model for making predictions. This approach offers a more personalized and targeted prediction for each link's performance. The trade-off, however, is the need for more training data and computational resources, as each model must be created and maintained separately. This approach may increase the accuracy of the prediction, but may lead also to over-fitting. We use one dataset per link to predict the Packet Reception Rate.

3. **Single-link k-NN**: we use similar configurations to make our predictions for a specific (untested) configuration. We assume here that similar configurations result in similar performance metrics, making it a straightforward but possibly less precise approach compared to more sophisticated models. More precisely, we test the k-means algorithm, which clusters the data points (of the training set) based on their similarity by minimizing the within-cluster variance, and uses the mean reception rate of the closest cluster as the prediction.

Each approach offers unique advantages depending on the specific scenario. While global regression ensures generalization across links, the single-link regression emphasizes link-specific behavior, potentially improving prediction accuracy for unique environmental conditions. Sink-link k-NN exploits only the closest configurations, which may lead to suboptimal predictions if the tested configurations are old, or with significantly different parameters.

5.3 Data Splitting

We must consider the temporality: recent measurements should be preferred to the old ones. Indeed, conditions are time-variant (e.g., external interference, obstacles). Thus, we split the dataset chronologically: the first *S*% of configurations (based on temporal order) are used as training data, and the rest of the configurations are used for testing.

$$\underbrace{(C_{l,1}, PRR_{l,1}), \ldots, (C_{l,k}, PRR_{l,k})}_{\text{Training data}}, \underbrace{(C_{l,k+1}, PRR_{l,k+1}), \ldots, (C_{l,n}, PRR_{l,n})}_{\text{Testing data}}$$

This approach ensures that the model reflects the temporal evolution of configurations, making the testing phase closer to real-world scenarios.

6 Evaluation

We compare the three approaches described in the previous section. We ran our 20-hour experiments 6 times to make our evaluation more robust and averaged the corresponding results. The configuration stays unchanged but competing experiments may run on the testbed, or Wi-Fi traffic may vary. We measure the prediction accuracy of each method with the Mean Squared Error (MSE), defined as follows:

$$MSE(l) = \frac{1}{n}\sum_{i=1}^{n}\left(PRR_{l,i} - \widehat{PRR}_{l,i}\right)^2 \tag{2}$$

with $PRR_{l,i}$ is the observed PRR value for the link l and the interval i while $\widehat{PRR}(l,i)$ is the predicted one.

6.1 Global Comparison

Figure 4 illustrates the Mean Squared Error (MSE) on the testing data, with the first 70% of the data used for training and the remaining 30% for testing. As observed, the accuracy is higher for all the links with the *single-link regression* model: only the measurements of the links are used to construct the model. This aligns with our earlier observation: the links are highly heterogeneous, and merging them into a single model leads to the loss of unique features that characterize each link. The *global regression* model may use the data from very different links, leading to poor prediction values for the *unusual* links (e.g., m3-123). The *single-link k-NN* approach yields comparatively poor results, performing worse than the other approaches. This is primarily due to the complexity of capturing the intricate relationships and high variability across links using a distance-based method. As it relies heavily on the

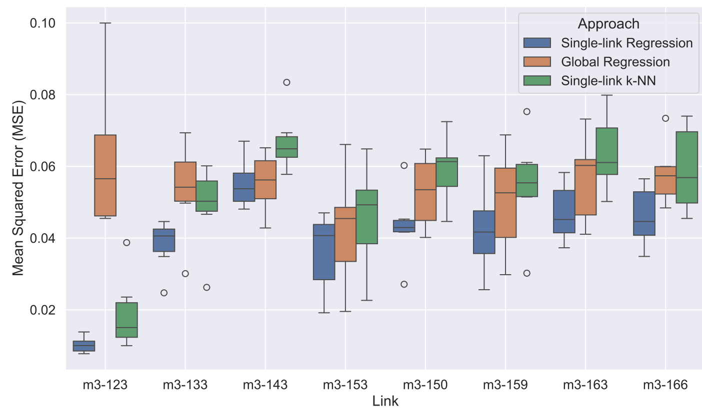

Fig. 4 Comparison of the three approaches on the given deployment.

similarity between data points, it performs poorly in heterogeneous environments where links exhibit highly diverse behaviors.

Interestingly, the performance gap is even more pronounced for links that exhibit distinctly different characteristics from others—for example, $m3$-123, which has the best link quality in the network. Lastly, the obtained MSE values are sufficiently low (≤ 0.08), demonstrating that the proposed approach can accurately predict the mean reception ratio of an unknown configuration with minimal average error.

6.2 Impact of the training length

Next, we examine the impact of the dataset's training length. A shorter training period might enhance responsiveness to time-variant relationships. However, it could also reduce prediction accuracy due to the smaller amount of data available for model training.

Figure 5 shows the distribution of prediction errors for the three approaches across various split ratios, ranging from 0.1 to 0.7 (i.e., with 10% to 70% of the dataset used for training). To minimize bias, the testing set remains fixed at the next 30% of the dataset, regardless of the split ratio.

As shown in the figure, *single-link regression* consistently outperforms the other methods across all split ratios. For certain links (e.g., $m3$-123), increasing the training dataset size noticeably reduces prediction errors. However, for most links, the error remains relatively stable. This stability likely stems from the fact that link behavior, on average and over extended periods, tends to be relatively stationary. Consequently, adding more data does not always yield significant performance gains. Nevertheless, links experiencing temporary disturbances may benefit more from larger training datasets. Finally, the results indicate that *global regression* can deliver satisfactory performance when the target link shares similar characteristics with the rest of the network.

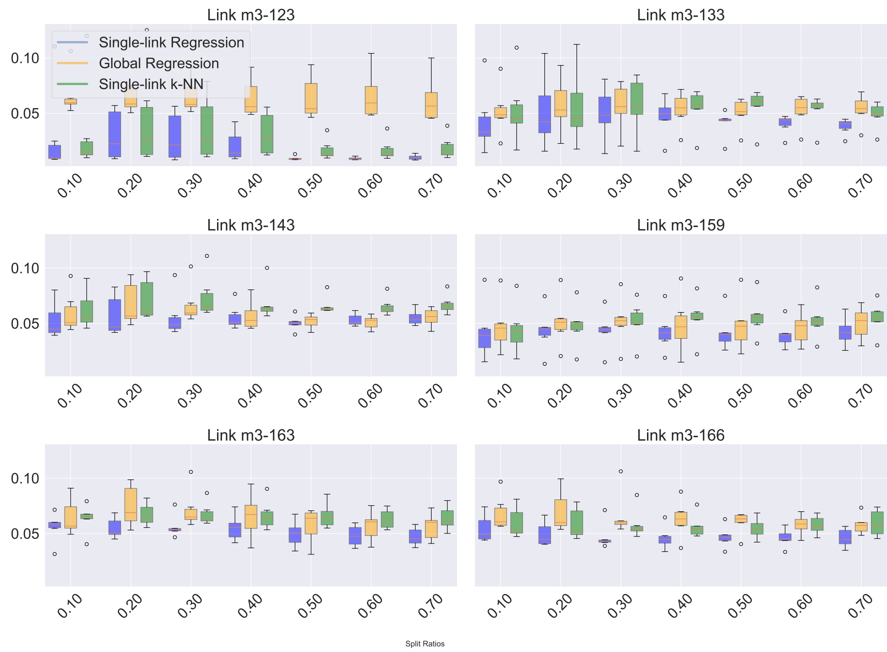

Fig. 5 Comparison of the three approaches on the given deployment, according to the train/test split.

7 Conclusion

In this work, we tackled the challenge of developing predictive models to assess the impact of unknown configurations, focusing specifically on MAC parameters in IEEE 802.15.4 networks and their effect on the number of (re)transmissions. We proposed and compared three modeling approaches: i) a global regression model, ii) a single-link regression model, iii) a single-link k-NN model. Using a real-world testbed, we deployed a wireless network and conducted an in-depth analysis that revealed the heterogeneity of link behaviors, and thus the interest of the single-link models. The single-link regression model predicts with a very good accuracy the link quality for unknown configurations. Our models constitute a solid piece of work to be integrated into a Digital Twin for wireless networks.

In future work, we plan to expand this study by incorporating additional performance metrics, such as latency and jitter, to provide a more comprehensive analysis. We need also to go one step further to integrate these models in the future Digital Twins for wireless networks. We have also to reduce the volume of measurements to use for training, to reduce both the overhead and the computational complexity.

Code sharing: For the sake of reproducibility, we provide the source code/dataset in the following link: https://github.com/SamirSim/NDT-Generalization

Acknowledgements This research was funded by the region Grand Est through the project ENGLAB.

References

1. Santi, N., Grünblatt, R., Foubert, B., Hameed, A., Violos, J., Leivadeas, A., & Mitton, N. (2021, November). Automated and reproducible application traces generation for IoT applications. In Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks (pp. 17-24).
2. Brun-Laguna, K., Minet, P., Watteyne, T., & Gomes, P. H. (2019). Moving beyond testbeds? Lessons (we) learned about connectivity. IEEE Pervasive Computing, 17(4), 15-27.
3. Aboubakar, M., Roux, P., Kellil, M., & Bouabdallah, A. (2020, October). An efficient and adaptive configuration of IEEE 802.15. 4 MAC for communication delay optimisation. In 2020 11th international conference on network of the future (NoF) (pp. 1-7). IEEE.
4. Karmakar, R., Chattopadhyay, S., & Chakraborty, S. (2020). An online learning approach for auto link-Configuration in IEEE 802.11 ac wireless networks. Computer Networks, 181, 107426.
5. Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., ... & Watteyne, T. (2015, December). FIT IoT-LAB: A large scale open experimental IoT testbed. In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT) (pp. 459-464). IEEE.
6. Fraile, L. P., Tsampas, S., Mylonas, G., & Amaxilatis, D. (2020). A comparative study of LoRa and IEEE 802.15. 4-based IoT deployments inside school buildings. IEEE Access, 8, 160957-160981.
7. Al-Kaseem, B. R., Al-Raweshidy, H. S., Al-Dunainawi, Y., & Banitsas, K. (2017). A new intelligent approach for optimizing 6lowpan mac layer parameters. IEEE Access, 5, 16229-16240.
8. Laura Marie Feeney. (2012). Towards trustworthy simulation of wireless MAC/PHY layers: a comparison framework. In ACM MSWiM, 295–304. https://doi.org/10.1145/2387238.2387288
9. Fontes, H., Campos, R., & Ricardo, M. (2017, June). A trace-based ns-3 simulation approach for perpetuating real-world experiments. In Proceedings of the 2017 Workshop on ns-3 (pp. 118-124).
10. Chen, C., Li, J., Balasubramaniam, V., Wu, Y., Zhang, Y., & Wan, S. (2020). Contention resolution in Wi-Fi 6-enabled Internet of Things based on deep learning. IEEE Internet of Things Journal, 8(7), 5309-5320.
11. Camps-Mur, D., Gomony, M. D., Pérez-Costa, X., & Sallent-Ribes, S. (2012). Leveraging 802.11 n frame aggregation to enhance QoS and power consumption in Wi-Fi networks. Computer Networks, 56(12), 2896-2911.
12. Si-Mohammed, S., Bardou, A., Begin, T., Lassous, I. G., & Vicat-Blanc, P. (2024). NS+NDT: Smart Integration of Network Simulation in Network Digital Twin, Application to IoT Networks. Future Generation Computer Systems, 157, 124-144.
13. Masaracchia, A., Sharma, V., Canberk, B., Dobre, O. A., & Duong, T. Q. (2022). Digital twin for 6G: Taxonomy, research challenges, and the road ahead. IEEE Open Journal of the Communications Society, 3, 2137-2150.
14. Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. Advances in neural information processing systems, 9.
15. Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis, 38(4), 367-378.
16. Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1, 81-106.