Group-Based Link Modeling for Wireless Digital Twins: Towards Accurate Network Performance Prediction

Samir Si-Mohammed[a,*], Fabrice Theoleyre[b]

[a]*CRAN Laboratory; University of Lorraine; CNRS; 54000 Nancy; France*
[b]*ICube Laboratory; CNRS ; University of Strasbourg; 67412 Illkirch Cedex France*

Abstract

Wireless networks are increasingly deployed in diverse domains, from best-effort object tracking to real-time control in smart factories. Yet, their performance strongly depends on configuration choices, especially at the MAC level. Thus, a single homogeneous configuration is often suboptimal due to the heterogeneous nature of individual links. We argue that Digital Twins (DTs) are a promising enabler for autonomous networks, capable of adapting configurations dynamically to prevailing conditions. However, global modeling approaches in DTs make it difficult to capture link-level variability and to accurately model the impact of configuration changes on performance. In this work, we propose a link-oriented prediction model designed to serve as a cornerstone for future wireless Digital Twins. Our model estimates the Packet Reception Rate (Packet Reception Rate (PRR)) under different MAC configurations, capturing the unique characteristics of each communication link. To improve scalability in large deployments, we explore a clustering-based approach, where predictive models are trained per group of similar links rather than per individual link. Our experimental evaluation shows that these data-driven methods effectively capture link heterogeneity while offering robust prediction accuracy and enhanced generalization capabilities.

Keywords: Wireless Networks, Internet of Things, Performance Evaluation, Experimentation, Machine Learning.

---

1. Introduction

Wireless networks are a foundation of modern communication, seamlessly connecting devices, systems, and people across diverse contexts. Their impact extends far beyond personal communication and entertainment, underpinning critical domains such as healthcare, transportation, and smart cities [1]. In the era of Industry 4.0, they have become a transformative force, driving the integration of the Internet of Things (IoT), real-time data analytics, and intelligent automation. This pervasive connectivity is fueling the emergence of new generations of services across multiple sectors.

As wireless infrastructures grow in scale and complexity, there is an increasing need for advanced tools to design, monitor, and optimize their performance. Among these, Digital Twins (DTs) are emerging as a key enabler. A DT is a digital replica of a physical entity or system that continuously updates based on real-world data. DTs have been initially designed for Industry 4.0 to help decision-making: assumptions can be validated through the DT instead of disturbing the real-world facility. In the context of wireless networking, DTs provide a powerful means of reproducing and experimenting with network behavior in virtual environments. They allow researchers and practitioners to evaluate diverse configurations and environmental conditions, without the prohibitive cost and effort of extensive physical deployments.

Wireless networks are particularly challenging to model since wireless transmissions exhibit time-varying characteristics, are lossy, and may lead to asymmetrical links [2]. Thus, radio propagation models have been proposed to predict the link quality. However, they require an extensive knowledge of the environment (obstacles, etc.) Recently propagation models exploit machine learning (ML) algorithms to make predictions from measurements [3]. Unfortunately, the computation complexity of these models remains high and the system needs accurate low-level PHY measurements for calibration.

Generalization is particularly challenging: Digital Twins should be able to predict the performance of unseen configurations. This is particularly relevant in wireless networks operating in IoT where variations in configuration

*Corresponding author
Email address: samir.si-mohammed@univ-lorraine.fr (Samir Si-Mohammed)

parameters (e.g., at the Medium Access Control (MAC) layer) can lead to widely different performance outcomes [4]. Generalization within DTs thus refers to predicting Key Performance Indicators (KPIs) such as latency or throughput for configurations that the system has not directly encountered. Achieving this capability is essential, since real-world networks continuously evolve and are subject to conditions that may deviate from those explicitly tested.

To confront these challenges, experimental studies remain indispensable. They bridge the gap between theoretical modeling and real-world implementations, providing insights into performance and reliability under heterogeneous conditions. Several works have relied on testbeds to evaluate network metrics including latency, throughput, and link variability [5, 6]. In parallel, significant research has been devoted to optimizing MAC parameters in wireless standards like IEEE 802.15.4 and Wi-Fi, frequently leveraging machine learning to improve throughput, minimize latency, and enhance energy efficiency [7, 8]. Collectively, these studies cover both simulation and experimental perspectives, offering valuable baselines for performance optimization.

Accurately predicting network performance for a given topology or configuration is of primary importance. For example, RouteNet [9] leverages Graph Neural Network (GNN) to estimate global KPIs such as delay, jitter, and loss. By combining deep learning with graph-based interdependencies, it enables complex performance predictions. However, such approaches often incur high computational costs and rely on homogeneous models that do not capture the heterogeneity of wireless environments. In practice, local sources of interference or physical obstacles may affect different network areas in highly diverse ways.

Unlike prior approaches, we introduce per-link data-driven models, trained individually to capture the unique characteristics of each communication path. This fine-grained modeling delivers high prediction accuracy with low error rates. To ensure scalability in large-scale deployments, we further propose a clustering-based strategy, grouping links with similar behavior and training a single predictive model per cluster. Together, these two levels of modeling combine precision with scalability, while enhancing generalization to unseen scenariospaving the way for robust Digital Twins in future wireless networks.

Our contributions are as follows:

1. **A thorough analysis of an experimental deployment**, providing insights into i) the heterogeneity of wireless links, ii) the impact of configurations on network performance, iii) 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.

3. **A cluster-based method**, where links are aggregated according to several criteria (topology, link quality, etc.) and a model is trained per cluster, to ensure better scalability.

2. Related Work

Real-world deployments and testbeds play a central role in advancing the Internet of Things (IoT). Specifically, experiments help to identify pitfalls that may arise in practice and complete simulations relying on theoretical models.

FIT IoT-LAB [10] and SmartSantander [11] are two popular testbeds and real-world deployments, respectively. Testbeds enable repeatable evaluations in controlled settings, while real-world deployments uncover the challenges of environmental variability, long-term reliability, and user behavior. They allow researchers to test their propositions in a large-scale environment. Experimental studies are vital for bridging the gap between theoretical models and real deployments.

2.1. Digital Twin for wireless networks

Originally introduced in the context of Industry 4.0 [12], DTs create a virtual representation of a physical system, continuously synchronized with the real world to support informed decision-making. A key capability lies in simulating *what-if scenarios* [13], which predicts the impact of configuration changes before they are applied. This functionality opens new opportunities, allowing administrators to evaluate and optimize their choices without the cost and risk of deploying and testing configurations in the physical systemthe DT itself is sufficient for this purpose.

Digital Twins have also emerged in wireless networking to construct an accurate representation of the network deployment [14]. Colosseum [15] offers a comprehensive Digital Twin platform for cellular networks. It relies on a sophisticated testbed composed of dedicated Field-Programmable Gate Arrays (FPGAs), capable of replaying real-world measurements to reproduce realistic scenarios under controlled conditions. Specifically, Colosseum follows a data-driven methodology, where FPGAs emulate radio propagation based on large collections of empirical measurements. Although such a PHY-oriented Digital Twin demonstrates strong potential, its deployment comes at a considerable cost.

The DT concept has also been explored in the context of the Internet of Things (IoT) [16]. In this approach, each physical object is associated with a corresponding software agent that maintains continuous information exchange with its counterpart. Building on this representation, multi-agent algorithms can be leveraged to perform complex computations and support decision-making processes. However, many digital twins focus on the application layer and do not consider the wireless aspects.

By capturing the characteristics of the wireless infrastructure, a DT can help operate networks closer to optimal performance. For instance, Si-Mohammed *et al.* [17] introduce a Network Digital Twin (NDT) that optimizes IEEE 802.15.4 network configurations using a heuristic combined with online learning. Similarly, Masaracchia *et al.* [18] emphasize that NDTs enable the integration of AI modules, which extract insights from real-time data and allow the system to evolve dynamically by continuously tuning operational parameters. They further suggest that AI-enabled NDTs can support network design by evaluating alternative scenarios across different contexts, thereby identifying configurations that maximize Quality of Service (QoS). More broadly, machine learning and statistical methods have shown strong potential to provide actionable insights from empirical datacapabilities that are particularly vital for generating reliable predictions within NDT frameworks.

While a NDT reflects in real-time the properties of the network infrastructure, it complements localized network policies. For instance, opportunistic transmissions require local knowledge and decisions. A DT should explore the tradeoff between cost (volume of measurements, computational complexity) and accuracy. A Network Digital Twin is a tool for "scenario planning, impact analysis and change management" according to the IETF [19]. Since it executes centralized and aggregated models, calibrated with measurements, it is expected to react in a few seconds / minutes.

2.2. *Link quality estimation*

Wireless networks are know to exhibit complex links behaviors [2]: asymmetry, unreliability, and time variability are the rule. Pavkovic *et al.* [20] provide a thorough statistical evaluation of measurements in a large-scale indoor and outdoor testbed. In particular, the wireless topology is dynamic since the link quality may evolve, and the link quality is consequently expensive to estimate with a few samples. Thus, predicting link quality may help to select the best routing paths or access points [21].

The Digital Twin may also exploit PHY-layer models and metrics. The objective is to construct an accurate path loss model to predict the Packet Error Rate. Ojo *et al.* [22] combine Radial Basis Function (RBF) and a Multilayer Perceptron (MLP) neural network model to predict the path loss based on real-field measurements in a 4G LTE network. Ensemble methods can improve the accuracy by combining different predictors such as Artificial Neural Network (ANN), Long Short-Term Memory (LSTM) and CNN [23]. 3D Ray Tracing can generate data through a simulator to train a Deep Learning model [24], and thus, to reduce the needs for a pre-deployment. Unfortunately, the 3D environment has to be finely modeled. Moreover, all these PHY-based approaches require complex dedicated hardware and in field calibrations. Besides, the models are constructed for aggregated values, *i.e.* not to mimic a specific link / physical situation.

Link quality prediction consists in analyzing time-series of measurements (*i.e.* packet success / failure) to make predictions. The predictions may be short-term to predict the success of upcoming transmissions [25]. However, a Digital Twin should consider mid-term predictions to let enough time to react.

A wide range of link quality indicators has been proposed in the literature [26]. For example, 4C [27] applies machine learning to multiple PHY-layer featuressuch as Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR), and Link Quality Indicator (LQI)to estimate the probability of successfully delivering the next packet. While effective, such methods are limited to short-term predictions (*i.e.* $\approx$ hundreds of millisecond), which restricts their applicability.

Classification-based approaches can categorize links into coarse classes (good, medium, bad) [28], but these categories often group together heterogeneous links. Environmental parameters such as temperature and humidity can help to refine long-term prediction models for long-range networks [29]. However, these models are unable to react sufficiently fast (*e.g.* in a few seconds / minutes).

Many statistical techniques have been proposed for predictions with time-series. Millan *et al.* [25] are able to predict the link quality of a large-scale community network with a very good accuracy with simple Machine Learning predictors Unfortunately, the sampling rate seems insufficient for a real-time Digital Twin (one PDR measurement every 5 minutes). Formis *et al*. [30] have proved that a data-driven model using a weighted moving average can make accurate short-term predictions for Wi-Fi networks. However, IoT networks are known to send less packets, making the predictions more challenging.

Deep Learning (DL) methods such as LSTM and Gated Recurrent Unit (GRU) can significantly outperform predictions compared with classical Autoregressive Integrated Moving Average (ARIMA) models [31]. More specifically, a Recurrent Neural Network can identify repetitive patterns in the time-series to detect fluctuating links [32], while such technique has only be evaluated with synthetic data. Even Large Language Models (LLM) have been used in combination with Gate Attention (GAT) to make short-term predictions ($< 10ms$). However, it tends to combine many features (MAC, PHY, Physical Layer Convergence Protocol (PLCP)), increasing the complexity for both the training and operational phases. A Digital Twin should prioritize lightweight models to maximize the benefits while minimizing the associated costs.

Classification helps to identify patterns in the time-series. Skaperas *et al*. [33] apply a clustering algorithm based on elastic similarity measures to a link quality set of measurements. While such technique helps to identifies anomalies (*i.e*. unknown pattern), it can hardly be adapted for accurate predictions, and are rather orthogonal to our method. Regrouping links in clusters may help to reduce the interference level [34]: interfering devices are dispatched to different clusters, and receive orthogonal radio ressources. Surprisingly, the classification of links according to their long and short-term characteristics has been seldom studied.

Recently, a combination of clustering and DL techniques such as LSTM help to make service prediction in cellular networks [35]. In this article, we employ a similar approach, but with the objective of predicting the radio quality of various links rather than the traffic requests at the application level.

2.3. MAC layer parameterization

The optimization of MAC parameters in wireless standards has attracted much attention in the past, since as Anastasi *et al.* [36] have shown, variations in CSMA/CA parameters affect capacity, fairness, and collision dynamics in wireless networks, demonstrating the direct impact of MAC-layer behavior on end-to-end network performance. Aboubakar *et al*. [7] use MLP, random forest, k-Nearest Neighbor (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*. [4] 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. Unfortunately, simulators' accuracy strongly depends on the physical layer models' accuracy [37]. PHY models often deviate significantly from real-world conditions [38].

Fraile *et al*. [39] present a comprehensive experimental study quantifying the impact of MAC parameters on network performance, with a comparison between IEEE 802.15.4 and LoRa. Their results show that LoRa's MAC behavior is highly sensitive to the choice of Spreading Factor and Bandwidth. Reinforcement learning-based approaches have been proposed to optimize LoRa parameters [40], aiming to improve throughput while ensuring compliance with ISM band regulations.

Acknowledgments have a strong impact on the capacity [41]: more transmissions means more collisions and less bandwidth for data packets. But inversely, a bad Packet Error Rate due to a low channel gain may lead to packet losses. Thus, the wireless capacity depends on the ACK strategy.

Karmakar *et al*. [8] propose an online learning-based solution for Wi-Fi networks. They tune channel bandwidth, Modulation Coding Scheme (MCS) values, and the number of Multiple Input Multiple Output (MIMO) antennas to achieve high throughput in Wi-Fi-based topologies. In contrast, Chen *et al*. [42] adopt a deep learning approach to adapt the Contention Window. Their data is generated via simulations covering various conditions. Furthermore, [43] introduce an algorithm to dynamically adjust the level of IEEE 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.

Ye *et al*. [44] propose to select the most accurate MCS for cellular networks with an outdated channel quality indicator (CQI) feedback. Indeed, feedback is often delayed and the system may make erroneous decisions. They propose consequently a new action selection strategy to reduce the number of reconfigurations.

We summarize key related works on MAC configuration modeling in Table 1, highlighting the methodology and the focus of each one of them.

Table 1: Comparison of related works on MAC configuration modeling.

Work	Technology	Methodology	Data Source	Focus & Contribution
Si-Mohammed *et al*. [17]	Agnostic	Heuristic + Online Learning	Real testbed	Optimizes MAC configs; global modeling; relies on simulation.
Aboubakar *et al*. [7]	802.15.4	MLP, Random Forests, k-NN.	Simulation	Minimizes delay; global model; relies on simulation.
Alkaseem *et al*. [4]	802.15.4	ANN	Simulation	Predicts optimal MAC params; global modeling; relies on simulation.
Fraile *et al*. [39]	802.15.4, LoRa	Experimental Study	Real testbed	Quantifies MAC impact; per-link modeling; no predictive model.
Karmakar *et al*. [8]	Wi-Fi	Online Learning	Real testbed	Optimized bandwidth, MCS, antennas; global modeling.
Chen *et al*. [42]	Wi-Fi	Deep Learning	Simulation	Adapts contention window; global modeling; relies on simulation.
Si-Mohammed *et al*. [45]	802.15.4	Lightweight regression models	Real testbed	Predicts retransmissions under MAC configs; per-link modeling.
This work	802.15.4	Lightweight regression and clustering models	Real testbed	Predicts retransmissions under MAC configs; per-link & cluster-based modeling.

3. Problem Statement

Wireless networks are inherently lossy, and retransmissions are commonly used to increase reliability. However, raising the number of retransmissions can exacerbate congestion, leading to higher latency and reduced energy efficiency. Likewise, the choice of backoff parameters directly influences the probability of collisions.

In most deployments, network configurations are fixed at setup time. We propose instead to employ a digital twin that continuously collects performance metrics to predict the behavior of individual links and determine their optimal configuration. Since the configuration space grows exponentially with the number of tunable parameters, the challenge lies in predicting performance for configurations that have not been explicitly *tested*.

We propose in this paper to predict the Packet Reception Rate (PRR) value for a given configuration. The PRR is defined as the ratio of the number of received packets and the number of transmitted packets for each link. In the rest of the paper, we use only this KPI only (PRR) since it directly impacts for instance energy efficiency and delay. The jitter metric is more complicated to predict since it is related to the distribution of consecutive packet failures. We let this complex KPI prediction to a future work.

To address this, we adopt an experiment-driven methodology with a testbed running an IEEE 802.15.4/6LoWPAN stack, relying on the unslotted mode. This mode implements a Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) mechanism prone to collisions. Our study focuses on three key MAC parameters:

- **macMinBE and macMaxBE** (Backoff Exponent), defining the initial and maximum backoff exponent values from which the backoff duration is derived;
- **macMaxCSMABackoffs**, limiting the number of backoff attempts before a transmission is abandoned, after which the packet is dropped;
- **aMaxFrameRetries** setting the maximum number of retransmissions allowed for a frame in case of repeated transmission failures.

Figure 1 summarizes the unslotted CSMA-CA procedure.

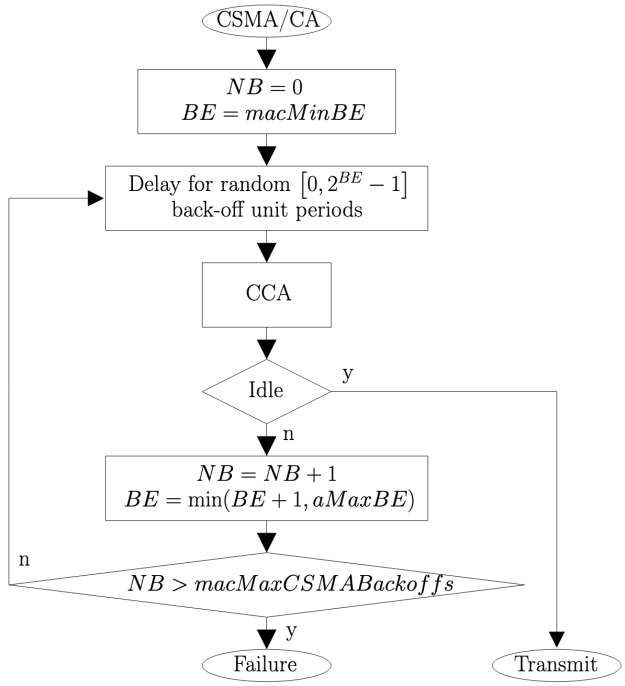

Figure 1: Unslotted CSMA-CA procedure for medium access.

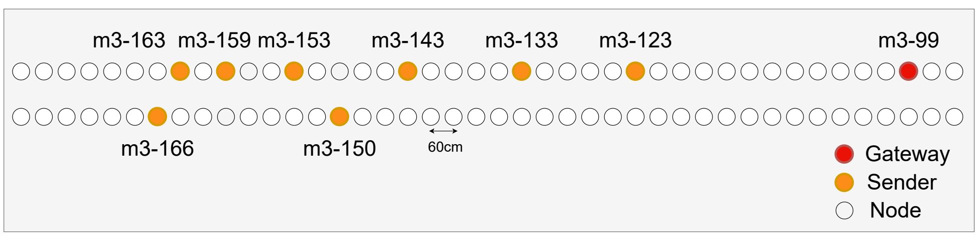

Figure 2: Use case 1: nodes placement in the FIT IoT-Lab platform, Grenoble (France). Nodes are placed in the ceils / floors, along a corridor.

The combination of parameter values results in a vast configuration space (exceeding 2,000 possible settings), each with potentially significant implications for network performance [17]. Identifying optimized configurations tailored to specific applications and environments is therefore essential.

Consider a deployed network where a node has already experienced a sequence of configurations $C = [C_1, \ldots, C_n]$, each yielding a corresponding performance outcome $KPI = [K_1, \ldots, K_n]$. Our goal is to design a model that can accurately predict the performance K_m associated with a new configuration C_m that the node has not previously tested (*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 SLICES-RI wireless testbed (`https://www.iot-lab.info/`) to conduct our experiments.

4.1. Experimental Setup

We conducted a 20-hour experiment on the Grenoble site, considering a single-hop (cellular) topology in which a gateway collects packets from eight M3 motes. The setup emulates a realistic smart building scenario, with the motes distributed along a corridor at varying distances from the gateway (see Figure 2). The building also hosts other wireless technologies (e.g., Wi-Fi), which may generate interference. The experiment was carried out using the Contiki-NG operating system running the IEEE 802.15.4 protocol stack.

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

Table 2: Medium Access Control parameters.

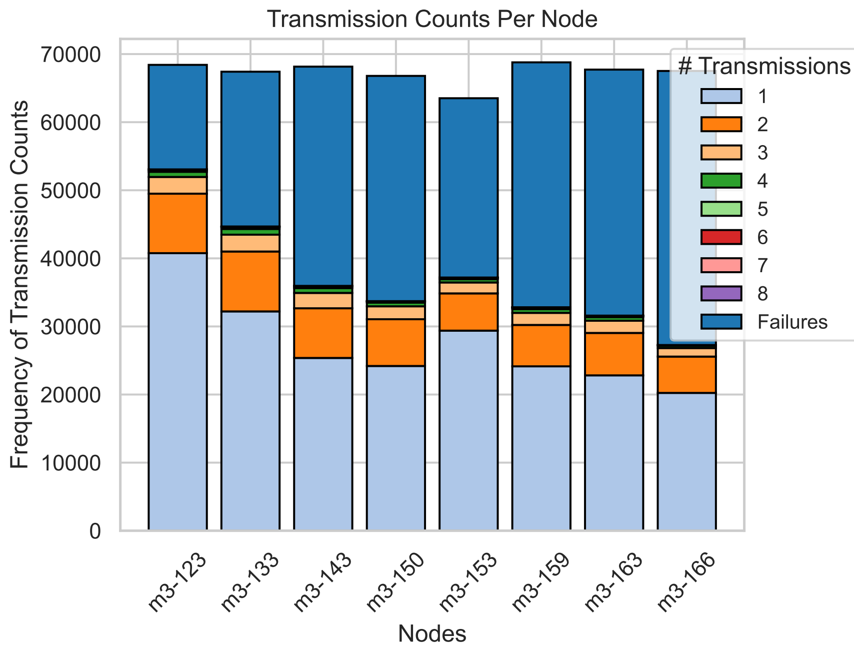

Figure 3: Histogram of the transmission counts before a packet is correctly decoded by the receiver for each of the nodes

We conducted 20 hours of experiments with different MAC configurations. Each configuration lasts 100 s, during which each node generates Constant Bit Rate (CBR) traffic toward the gateway at a rate of one packet per second.

At the beginning of each period, every node randomly selects a MAC configuration. Consequently, all nodes produce the same traffic volume at the application level, but may operate with different sets of MAC parameters. The parameter ranges are summarized in Table 2. At the end of each period, we collect per-node statistics, including the number of transmitted and successfully received packets. The total number of transmissions depends, for example, on the maximum number of allowed retransmissions at the MAC layer. We have finally 720 time periods in our experiments.

As a Key Performance Indicator (KPI), we consider the number of (re)transmissions required for a packet to be successfully delivered from a mote to the gateway. For each link (*mote* → *gateway*), we collect two time series:

1. **Configurations:** $C_l = [C_{l,1}, \ldots, C_{l,k}]$, where $C_{l,i}$ represents the configuration applied to link l during the i^{th} time interval. We denote by C the set of all possible configurations.

2. **PRR:** $PRR_l = [PRR_{l,1}, \ldots, PRR_{l,k}]$, where $PRR_{l,i}$ is the average PRR observed on link l during the i^{th} interval. The PRR is computed as the ratio between the number of packets successfully received by the gateway and the number of packets transmitted through link l.

4.2. Experimental Characterization of the Diversity

Figure 3 highlights the distribution of the number of transmissions required across all considered links. A transmission is deemed successful as soon as the receiver receives the first copy of the packet. If the packet is eventually dropped after reaching the maximum number of retransmission, the transmission is considered a failure. We observe that, for most links, the majority of packets are delivered in a single transmission, reflecting relatively good link quality

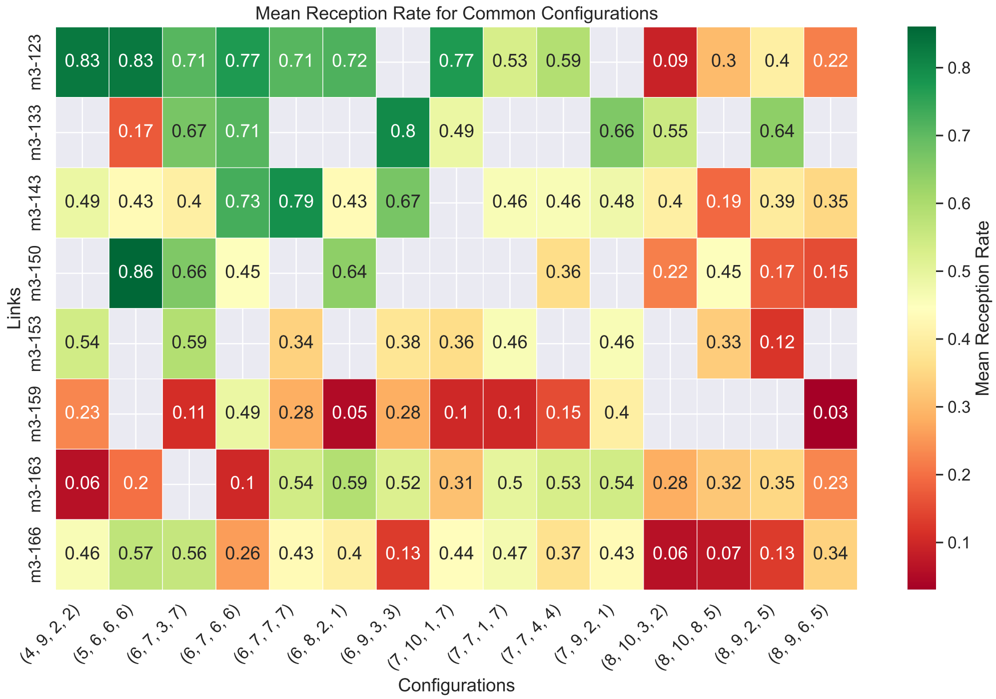

Figure 4: Mean reception rate for different MAC configurations with different links.

(especially for the nodes close to the gateway). Several links do, however, experience persistent failures, where packets remain undelivered even after the maximum number of retries.

Figure 4 shows the reception rate achieved by each mote for a given configuration. A PRR close to 1 indicates that a single transmission is typically sufficient for successful delivery to the gateway, whereas low PRR values (in red, close to 0) correspond to highly unreliable links. For clarity, each link is referred to by its transmitter (*i.e.,* mote) in the remainder of this section.

A key observation is that the same configuration (*e.g.,* (6, 9, 3, 3)) can perform very well for certain links (*e.g.,* m3-133) while being highly inefficient for others (*e.g.,* m3-166). This may seem intuitive, as links with poor physical-layer quality tend to experience high error rates, thereby reducing reliability. Besides, an interfering link may generate more or less traffic and impact the optimal MAC configuration. However, what is particularly noteworthy is that the **optimal configuration differs across links**: for instance, m3-123 achieves its best performance with *(4, 9, 2, 2)*, whereas m3-166 performs better with *(6, 8, 2, 1)*. This highlights the need to adapt configurations on a **per-link basis**an aspect that is seldom addressed in the literature.

It is important to note that, since the testbed is shared with other users, the performance of the evaluated configurations may be affected by concurrent experiments. In addition, the configuration of neighboring transmitters can also influence the performance of a given link; for instance, interfering nodes may generate more MAC traffic when configured with a higher maximum number of retransmissions. A natural extension of this work would be to analyze such dependencies in a dedicated, isolated testbed environment where only a single link is active at a time.

Even more striking is that PRR variations are not uniform across links. For instance, node m3-159 achieves significantly higher PRR with configuration *(6, 7, 6, 6)*, yet performs poorly with neighboring configurations. This highlights that the optimal configuration depends not only on the received signal strength, but also on external interference and competing traffic. We therefore advocate for a **per-link** and **per-cluster** approach, where predictive models map configurations to expected PRR both individually and collectively. The development of such data-driven models is the focus of the next section.

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 for the link PRR 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 ($c \in C$) 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. Lighweight Machine Learning models used for training

We focus on lightweight prediction models, as minimizing computational complexity is essential. Since one model must run continuously for each link, the Digital Twin can only be sustainable if the computational overhead remains reasonable. For this purpose, we rely on classical regression models to predict the mean reception ratio for a given configuration:

1. **Support Vector Regression (SVR)**: finds a hyperplane in a high-dimensional feature space that maximizes the margin from the training data points [46];

2. **Gradient Boosting**: builds prediction models sequentially, where each new model reduces the errors of the previous ones, and combines them into a stronger ensemble [47];

3. **Decision Trees**: recursively splits the data based on input feature values, producing a tree structure where nodes represent features, branches correspond to decision rules, and leaves denote outcomes [48].

These models are particularly attractive because of their low computational complexity, even if more sophisticated alternatives exist. Rather than fixing a single predictor, our Digital Twin operates as a meta-model, selecting the most appropriate predictor for each link. We argue that there is a fundamental trade-off between accuracy and computational efficiency. Consequently, our solution can dynamically switch between models when needed (see Figure 5), as proposed in [49].

5.3. Data used to train the models

We explore four distinct approaches to train our prediction models: i) **global regression** assumes all the links behave similarly, ii) **single-Link regression** considers that all the links are singular, and we use only data from the link to train *its* model, iii) **single-link k-Nearest Neighbor** considers a link specific, but non regressive model (exploring the k-nearest neighbor technique), iv) **cluster regression** constructs clusters of links behaving similarly.

5.3.1. Global Regression

To maximize the training dataset, we consider all links together. This is a common assumption in the literature, where a generic model is derived to estimate the Key Performance Indicator of the MAC layer from its parameters. A single, generalized model predicts the mean reception rate for any link, assuming they all exhibit similar behavior. While each link has unique characteristics, training on aggregated data can help capture broad patterns that apply across the network. Thus, we exploit a global dataset containing all links combined as training data for our regression model. This approach assumes that network-wide trends can be captured in a unified model, though its effectiveness may be reduced in highly heterogeneous environments, as we will highlight in the performance evaluation.

5.3.2. Single-Link Regression

Each link is modeled independently, with its own specialized predictor. This provides a more personalized and targeted estimation of performance, as the model captures link-specific behaviors. The drawback, however, is the higher need for training data and computational resources, since a separate model must be trained and maintained for each link. While this strategy may increase prediction accuracy, it also introduces the risk of overfitting. In practice, one dataset per link is used to predict the PRR.

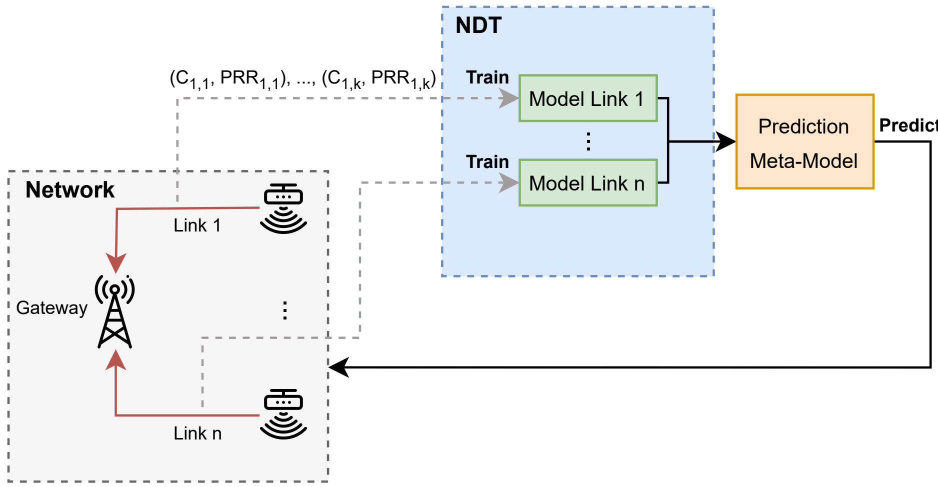

Figure 5: Prediction models are trained with data coming from the network and can be used to predict the outcome (*i.e.,* PRR) of unexplored configurations.

5.3.3. Single-link k-Nearest Neighbors (kNN)

Predictions for a specific (untested) configuration are based on *similar configurations*. The underlying assumption is that configurations with similar parameter values yield similar performance metrics, making this a simple yet potentially less precise approach compared to more sophisticated models. In practice, we apply the k-means algorithm to cluster the training data based on configuration similarity by minimizing the within-cluster variance, and then use the mean reception rate of the closest cluster as the predicted value.

5.3.4. Cluster Regression

we assume that similar links can be grouped into clusters. Links that share similar characteristics (*e.g.* PRR behavior, physical distance to the gateway, or observed interference conditions) behave similarly and we can thus regroup them in the same class. Note that the notion of similarity is closely tied to the chosen clustering technique: it can correspond to physical proximity between nodes (in the case of topological clustering) or to similarities in PRR distributions (for the other clustering methods). Regrouping links in clusters helps to improve both generalization (we can predict the behavior of a link from others) and accuracy (more data is used for training).

A regression model is trained on the combined data of each cluster, capturing common patterns among similar links while still accounting for heterogeneity across clusters.

We consider three different approaches for clustering the links:

1. Geographical positions *(`cluster-geo`).* We regroup links where emitters are located in the same geographical area. Thus, we group links according to the physical coordinates (x, y, z) of their emitter in the testbed. The k-means algorithm is applied to the node positions, ensuring that nodes that are spatially close belong to the same cluster. This approach reflects potential similarities in propagation conditions and interference patterns due to their geographic proximity. It is worth noting that we consider only geographical information, without any knowledge on the link qualities. We refer to this approach as the *cluster topology* approach in the remainder of this paper

2. Mean Packet Reception Rate (PRR) *(`cluster-mean-PRR`).* Links are clustered based on their average PRR of a link, whatever its configurations. Thus, we regroup links with the same average PDR, whatever the time and the configuration. Each link is represented by a single feature: its mean PDR.

Consequently, we apply the k-means algorithm applied to this average PDR value. This clustering emphasizes performance-based grouping, independent of spatial layout. This approach is referred to as the *cluster mean* approach in what is next.

Table 3: Main notation used in the paper

Variable	Meaning
C_l	Set of configurations explored by the link l
$PRR_{l,i}$	Packet Receiving Rate of link l during the time interval i
$config_{l,i}$	Configuration used for the link l during the time interval i
$config_{l,i} = \{p_k \mid k \in [1, n_{params}]\}$	p_k is the value of the k^{th} parameter of the configuration $config_{l,i}$
$dist(config_{l_1,i}, config_{l_2,j})$	Euclidean distance defined for any pair of configuration
$\tilde{config}_k^\star$	Representative configuration of each class of links

3. Configurations ranking similarity *(`cluster-rank`)*. The objective is here to regroup links that perform *similarly*, exhibiting similar PRR values for similar configurations. We compute the distance between two configurations using a euclidean distance in the parameters values. More precisely, we group links where the ordered set of PRR values according to the configuration is similar.

Let $PRR(l, \tilde{\mathbf{config}}_i)$ denotes the observed performance (e.g., PRR) of link l under configuration $\tilde{\mathbf{config}}_\mathbf{i}$, and C be the set of all the configurations. Table 3 regroups the notation used in the rest of the paper. The similarity is higher if the following condition holds:

$$\exists config_1, config_2 \in C | PRR(l_1, \tilde{config}_1) > PRR(l_1, \tilde{config}_2) \wedge PRR(l_2, \tilde{config}_1) > PRR(l_2, \tilde{config}_2) \tag{2}$$

Inversely the similarity is decreased if:

$$\exists config_1, config_2 \in C | PRR(l_1, \tilde{config}_1) > PRR(l_1, \tilde{config}_2) \wedge PRR(l_2, \tilde{config}_1) < PRR(l_2, \tilde{config}_2) \tag{3}$$

More specifically, let a configuration i be represented by a tuple

$$\mathbf{config_i} = (p_{i,1}, p_{i,2}, \ldots, p_{i,d})$$

where each parameter takes values in a finite range ($p_{ij} \in [p_j^{\min}, p_j^{\max}]$). Each parameter is classically normalized to avoid bias:

$$\tilde{\mathbf{p}}_j = \frac{p_{i,j} - p_j^{\min}}{p_j^{\max} - p_j^{\min}} \in [0, 1].$$

Thus, a normalized configuration is denoted as

$$\tilde{\mathbf{config}}_i = (\tilde{p}_{i,1}, \tilde{p}_{i,2}, \ldots, \tilde{p}_{i,d}).$$

We consider the euclidean distance to compare two normalized configurations $\tilde{\mathbf{config}}_i$ and $\tilde{\mathbf{config}}_j$:

$$dist(\tilde{\mathbf{config}}_\mathbf{i}, \tilde{\mathbf{config}}_\mathbf{j}) = \sqrt{\sum_{k=1}^{d} \left(\tilde{p}_{i,k} - \tilde{p}_{j,k}\right)^2}. \tag{4}$$

Configurations are considered equivalent (and merged into a single representative configuration) if their distance, $d(\tilde{\mathbf{c}}, \tilde{\mathbf{c}}')$, falls below a tolerance threshold τ, interpreted as a maximum allowed per-parameter deviation. Thus, only one representative configuration $\tilde{\mathbf{config}}_k^\star$ is retained for each equivalence class, resulting in the set $\{\tilde{\mathbf{config}}_1^\star, \ldots, \tilde{\mathbf{config}}_K^\star\}$.

Once the representative configurations are determined, the performance associated with each of them corresponds to the performance observed for the closest configuration that was effectively explored by the link.

$$PRR(l, \tilde{\mathbf{config}}_k^\star) = PRR\Big(l, \arg\min_{\tilde{\mathbf{config}} \in C_l} dist(\tilde{\mathbf{config}}, \tilde{\mathbf{config}}_k^\star)\Big),$$

For each link l, we construct a ranked vector

$$\mathbf{r}_l = \mathrm{rank}\Big(\{PRR(l, \tilde{\mathbf{c}}_1^\star), PRR(l, \tilde{\mathbf{c}}_2^\star), \ldots, PRR(l, \tilde{\mathbf{c}}_m^\star)\}\Big),$$

where rank(·) assigns an integer ordering to the m representative configurations.

Similarity between links l and l' is then quantified using Spearman rank correlation, defined as

$$\rho_{spearman}(l, l') = 1 - \frac{6 \sum_{i=1}^{m} (r_{l,i} - r_{l',i})^2}{m(m^2 - 1)}.$$

The corresponding Spearman distance is simply

$$d_S(l, l') = 1 - \rho_{spearman}(l, l').$$

Finally, hierarchical clustering is applied to the distance matrix $\{d_S(l, l')\}$, which reveals groups of links that exhibit comparable performance trends and highlights structural similarities in how they react to parameter variations.

5.4. Data Splitting

We must take temporality into account, as recent measurements are more relevant than older ones. Indeed, network conditions are time-varying (*e.g.,* external interference or obstacles). To reflect this, we split the dataset chronologically: the first $S\,\%$ of configurations (in temporal order) are used as training data, while the remaining configurations serve as testing data.

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

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. To ensure robustness, we repeated our 20-hour experiments six times and averaged the corresponding results. We considered 8 different nodes located along a corridor to model a certain diversity. While the sequence of configurations remains unchanged, other experiments on the testbed or variations in *e.g*. Wi-Fi traffic may introduce fluctuations. We evaluate the prediction accuracy of each method using the **Mean Squared Error (MSE)**, defined as follows:

$$MSE(l) = \frac{1}{n} \sum_{i=1}^{n} \left(PRR_{l,i} - \widehat{PRR}_{l,i} \right)^2 \tag{5}$$

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 6 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 highest for all links using the *single-link regression* model, which relies solely on measurements from the individual link to construct the model. This confirms our earlier observation that links are highly heterogeneous, and aggregating them into a single model can result in the loss of link-specific characteristics.

The *global regression* model, which uses data from all links, may produce poor predictions for *unusual* links (*e.g.,* m3-123). The *single-link k-NN* approach performs comparatively worse than the other methods. This method computes the expected PRR based on the k-nearest neighbor configurations. Unfortunately, this predictor struggles to capture the complex relationships among configurations.

Interestingly, the performance gap is particularly pronounced for links with characteristics that deviate significantly from the network average, *e.g.,* $m3$-123, which exhibits the best link quality. Finally, the mean 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.

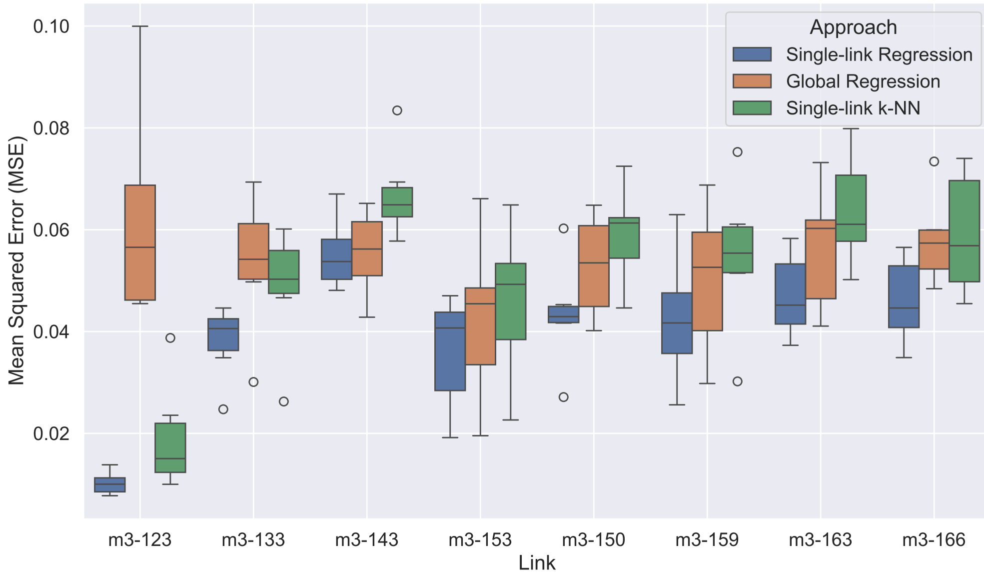

Figure 6: Mean Square Error rate of the estimated Packet Reception Rate for different links and prediction models.

6.2. Impact of the training length

Next, we examine the impact of the training length of the dataset. A shorter training period may improve the model's responsiveness to time-varying conditions, but it can also reduce prediction accuracy due to the smaller amount of data available for training.

Figure 7 presents the distribution of prediction errors for the three approaches across different training split ratios, ranging from 0.1 to 0.7 (i.e., 10% to 70% of the dataset used for training). To reduce bias, the testing set is kept fixed as the subsequent 30% of the dataset, regardless of the training split ratio (see Figure 8). Consequently, we minimize the temporal bias.

As shown in the figure, *single-link regression* consistently outperforms the other methods across all training split ratios. For certain links (*e.g., m*3-123), increasing the size of the training dataset noticeably reduces prediction errors. However, for most links, the error remains relatively stable. This stability likely arises because link behavior, on average and over extended periods, tends to be relatively stationary, so adding more data does not always yield substantial performance gains. Nonetheless, links experiencing temporary disturbances may benefit more from larger training datasets. Finally, the results indicate that *global regression* can achieve satisfactory performance when the target link exhibits characteristics similar to those of the rest of the network.

6.3. Evaluating the clustering approach

In order to strength the evaluation of our clustering approach, we conduct an additional experiment involving a larger network of 24 nodes (see Figure 9) with the same traffic intensity (1 packet per second), during 12 hours. We ran the experiment 4 different times, averaged the results, and we use the first 70% of the data for training, and the remaining 30% for testing. Clustering is then performed using the three different methods described in Section 5.3.4. For the `cluster-rank` approach, we used a normalized Euclidian distance to compute similarities between configurations with a tolerance of $\tau = 20\%$, and the Spearman correlation to clusterize the links according to their rankings. We consider a total number of four (04) clusters for the different approaches.

Since the single-link approach represents the highest accuracy in the previous section, we compare it with the three clustering-based methods (Figure 10). Note that, for readability purposes, we only show the performance for eight links (two per topological cluster). We observe that all clustering approaches achieve reasonably low prediction errors, showing that grouping links is a viable alternative to per-link modeling. As expected, the single-link approach remains the most accurate overall, since it is fully tailored to the behavior of each individual link. However, this method is challenging when a digital twin has to predict the performance of an untested link. Exploiting a cluster regression helps to tackle such challenge.

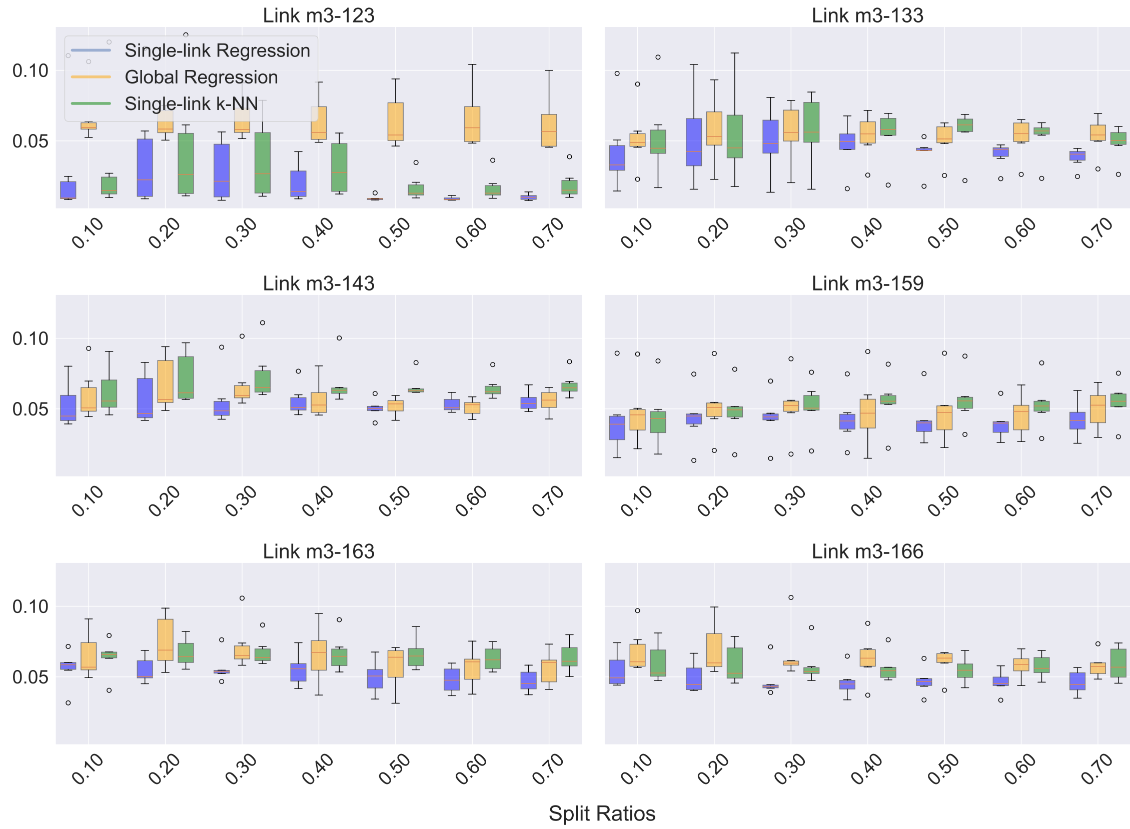

Figure 7: Impact of the prediction model on the absolute prediction error of the PRR of the different links in the testbed.

The errors vary across links: for reliable nodes such as *m3-110* or *m3-131*, all methods perform similarly with very low errors. For more unstable nodes like *m3-168*, prediction accuracy differs substantially depending on the clustering approach. `cluster-rank` offers better results for multiple links (*e.g.* *m3-156* & *m3-168*) than both `cluster-mean-PRR` and `cluster-geo`. Grouping links with similar PRR for similar configurations helps to improve accuracy.

`cluster-geo` still provides a solid compromise, achieving performance closer to the single-link baseline. Thus, nodes close from each other tend to exhibit similar tendencies, because of *e.g.* localized external interference. `cluster-mean-PRR` shows the highest errors overall, highlighting the limitations of oversimplifying link behavior by exploiting only the mean PRR of the links. Overall, these results emphasize the heterogeneity of link behaviors and the critical importance of selecting an appropriate clustering strategy.

Finally, it is worth noting that single-link models generally require less training time per model, as they are trained using data from a single link, whereas clustering-based approaches aggregate data from multiple links. Inference time is comparable across approaches due to the use of lightweight models. The primary difference lies in the computational footprint: maintaining one model per link scales linearly with the number of links, while clustering reduces the number of models by sharing them across links.

6.3.1. Towards generalization

We illustrate here how geographic information can help us to predict the performance of an entirely new link. This generalization property is at the core of a DT, enabling what-if scenarios. We may for instance predict the performance of a network is a device is attached to another gateway / access point. The key difficulty, however, is the absence of prior data since this link do not yet exist.

To overcome this limitation, we leverage `cluster-geo`, our only one approach that does not rely on pre-existing measurements for the calibration. We regroup links with emitters located in the same geographical area in the same *cluster*. We assume that these links may be have similarly, with for instance the same amount of external interference. Figure 11 considers 3 additional nodes*m3-101, m3-134*, and *m3-171*. Each of these nodes is mapped to its corresponding cluster according to its geographical location, using a static, manual approach. A dynamic clustering algorithm is

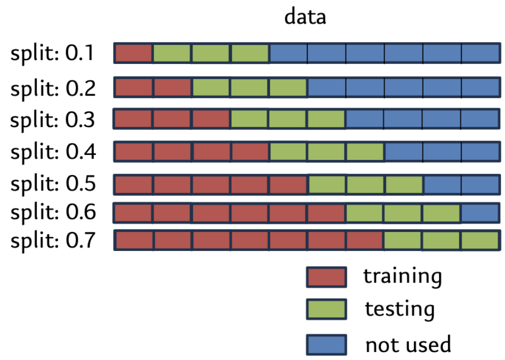

Figure 8: Splitting strategy for studying the training length impact: split 0.1 means that the first 10% of a PRR timeseries is used for training, the next 30% are used to test the performance of our prediction model, and the rest of data is unused.

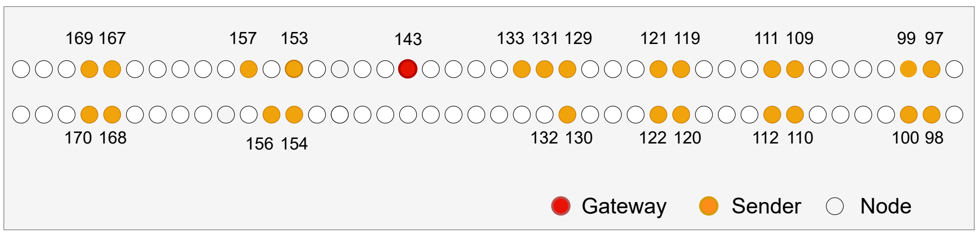

Figure 9: Use case 2: nodes placement in the FIT IoT-Lab platform, Grenoble (France). Nodes are placed in the ceils / floors, along a corridor.

let to a future work.

Brown nodes correspond to newly added nodes (*i.e.,* nodes excluded from the training process) used to assess the models ability to generalize. The red rectangle indicates a topological cluster whose nodes data are all used to train the model in this approach. To summarize, for each cluster, we train a model using the measurements of its cluster, except the data of the newly added nodes.

Figure 12 reports the MSE obtained for the three new nodes under `cluster-geo`. Results confirm that the prediction errors remain low, even without prior knowledge of the new nodes beyond their position. Interestingly, the error levels vary across nodes, as already observed in Figure 6. Yet, the magnitudes remain consistent with those observed for the other nodes of the same cluster. This highlights both the accuracy and the robustness of the clustering method for generalization tasks, and confirms its potential as a powerful enabler of what-if scenario evaluations in digital twins.

7. Conclusion

In this work, we addressed the challenge of developing predictive models to evaluate the impact of unknown configurations, focusing on MAC parameters in IEEE 802.15.4 networks and their influence on the number of (re)transmissions. We proposed and compared four modeling strategies: (i) a global regression model, (ii) a single-link regression model, (iii) a single-link k-NN model, and (iv) a cluster-based model. Using a real-world testbed, we deployed a wireless network and performed an in-depth analysis, which revealed strong heterogeneity in link behaviors and underscored the benefits of single-link models. Among them, the single-link regression model achieved high accuracy in predicting link quality under unseen configurations. The cluster-based model, on the other hand, successfully groups similar links while maintaining good predictive accuracy. Furthermore, the cluster-based approach enables generalization by estimating the performance of links for which no prior measurements are available.

Note that we deliberately focused in this work on lightweight regression models in order to balance prediction accuracy and computational complexity. While more complex models (e.g., random forests, LSTMs, or graph neural networks) were evaluated, they were either less accurate in our setting or significantly more costly to deploy. The selected models provide the best trade-off between performance and efficiency. Although additional hyperparameter tuning could further improve accuracy, this is not central to our contribution and can be easily explored in future work using the open-sourced code and dataset.

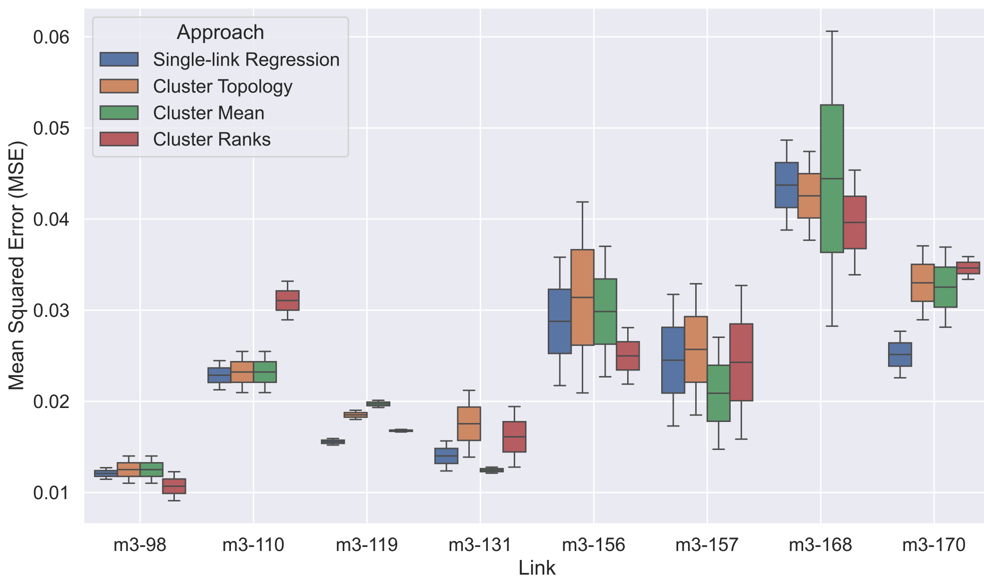

Figure 10: Comparison of single-link and clustering-based methods.

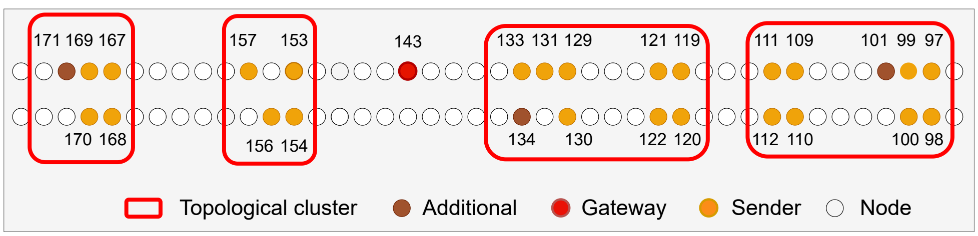

Figure 11: Use case 2: nodes placement in the FIT IoT-Lab platform, Grenoble (France). Nodes are placed in the ceils / floors, along a corridor. We add 3 nodes (in brown) to test the generalization property of the prediction algorithm. Additional nodes are placed in the middle or the boundary of a *cluster*.

As future work, we plan to extend this study by incorporating additional performance metrics, such as latency and jitter, to provide a more comprehensive evaluation. This is particularly relevant as multiple applications with different Service Level Agreements are expected to coexist, and MAC layer configurations should adapt not only to environmental conditions (*e.g.* signal strength) but also to application requirements (*e.g.* minimum reliability or maximum latency). Another important step is to embed these models into Digital Twin environments for wireless networks. We aim to investigate how different models can be integrated to support more complex *what-if* scenarios, leveraging their complementary strengths to predict the impact of configuration changes.

Code sharing

For the sake of reproducibility, we provide the source code/dataset in the following link: `https://github.com/SamirSim/NDT-IoTJ-2025`

8. Acknowledgement

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

References

[1] A. Ullah, Fawad, A. Nadeem, M. Arif, M. M. Bashir, and W. Choi, "6g internet-of-things assisted smart homes and buildings: Enabling technologies, opportunities and challenges," *Internet of Things*, vol. 32, p. 101658, 2025.

Figure 12: Mean Square Error of the PRR predicted by our model for the three additional emitters.

[2] M. Z. n. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, pp. 7–es, Jun. 2007.

[3] A. Seretis and C. D. Sarris, "An overview of machine learning techniques for radiowave propagation modeling," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 3970–3985, 2022.

[4] B. R. Al-Kaseem, H. S. Al-Raweshidy, Y. Al-Dunainawi, and K. Banitsas, "A new intelligent approach for optimizing 6lowpan mac layer parameters," *IEEE Access*, vol. 5, pp. 16 229–16 240, 2017.

[5] N. Santi, R. Grünblatt, B. Foubert, A. Hameed, J. Violos, A. Leivadeas, and N. Mitton, "Automated and reproducible application traces generation for iot applications," in *Symposium on QoS and Security for Wireless and Mobile Networks*. ACM, Nov. 2021, pp. 17–24.

[6] K. Brun-Laguna, P. Minet, T. Watteyne, and P. H. Gomes, "Moving beyond testbeds? lessons (we) learned about connectivity," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 15–27, 2019.

[7] M. Aboubakar, P. Roux, M. Kellil, and A. Bouabdallah, "An efficient and adaptive configuration of ieee 802.15.4 mac for communication delay optimisation," in *International Conference on Network of the Future (NoF)*. IEEE, Oct. 2020, pp. 1–7.

[8] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "An online learning approach for auto link-configuration in ieee 802.11 ac wireless networks," *Computer Networks*, vol. 181, p. 107426, 2020.

[9] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "Routenet: Leveraging graph neural networks for network modeling and optimization in sdn," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260–2270, 2020.

[10] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, and T. Watteyne, "Fit iot-lab: A large scale open experimental iot testbed," in *World Forum on Internet of Things (WF-IoT)*. IEEE, Dec. 2015, pp. 459–464.

[11] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "Smartsantander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.

[12] M. Javaid, A. Haleem, and R. Suman, "Digital twin applications toward industry 4.0: A review," *Cognitive Robotics*, vol. 3, pp. 71–92, 2023.

[13] G. Baruffaldi, R. Accorsi, and R. Manzini, "Warehouse management system customization and information availability in 3pl companies: A decision-support tool," *Industrial Management & Data Systems*, vol. 119, no. 2, pp. 251–273, 08 2018.

[14] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong, "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2230–2254, 2022.

[15] M. Polese, L. Bonati, S. D'Oro, P. Johari, D. Villa, S. Velumani, R. Gangula, M. Tsampazi, C. Paul Robinson, G. Gemmi, A. Lacava, S. Maxenti, H. Cheng, and T. Melodia, "Colosseum: The open ran digital twin," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5452–5466, 2024.

[16] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020.

[17] S. Si-Mohammed, A. Bardou, T. Begin, I. G. Lassous, and P. Vicat-Blanc, "Ns+ ndt: Smart integration of network simulation in network digital twin, application to iot networks," *Future Generation Computer Systems*, vol. 157, pp. 124–144, 2024.

[18] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6g: Taxonomy, research challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2137–2150, 2022.

[19] C. Zhou *et al.*, "Network digital twin: Concepts and reference architecture," https://datatracker.ietf.org/doc/draft-zhou-nmrg-digitaltwin-network-concepts/, IETF, draft, 2025.

[20] B. Pavkovic, F. Theoleyre, D. Barthel, and A. Duda, "Experimental analysis and characterization of a wireless sensor network environment," in *Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*. ACM, 2010, pp. 25–32.

[21] M. L. F. Sindjoung and P. Minet, "Estimating and predicting link quality in wireless iot networks," *Annals of Telecommunications*, vol. 77, no. 5, pp. 253–265, 2022.

[22] S. Ojo, M. Akkaya, and J. C. Sopuru, "An ensemble machine learning approach for enhanced path loss predictions for 4g lte wireless networks," *International Journal of Communication Systems*, vol. 35, no. 7, p. e5101, 2022.

[23] M. K. Elmezughi, O. Salih, T. J. Afullo, and K. J. Duffy, "Path loss modeling based on neural networks and ensemble method for future wireless networks," *Heliyon*, vol. 9, no. 9, Sep. 2023, publisher: Elsevier. [Online]. Available: https://doi.org/10.1016/j.heliyon.2023.e19685

[24] K. Qiu, S. Bakirtzis, H. Song, J. Zhang, and I. Wassell, "Pseudo ray-tracing: Deep leaning assisted outdoor mm-wave path loss prediction," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1699–1702, 2022.

[25] 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*, vol. 93, pp. 342–358, 2015.

[26] G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "Machine learning for wireless link quality estimation: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 696–728, 2021.

[27] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," *ACM Trans. Sen. Netw.*, vol. 10, no. 2, Jan. 2014.

[28] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 71–83, Jul. 2007.

[29] D. Szafranski, "Predictability of lorawan link quality based on weather data: Insights from a long-term study," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2024, pp. 249–258.

[30] G. Formis, G. Cena, L. Wisniewski, and S. Scanzio, "Accurate and efficient prediction of wi-fi link quality based on machine learning," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2025.

[31] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "Deepchannel: Wireless channel quality prediction using deep learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 443–456, 2020.

[32] M. Xu, W. Liu, J. Xu, Y. Xia, J. Mao, C. Xu, S. Hu, and D. Huang, "Recurrent neural network based link quality prediction for fluctuating low power wireless links," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1212

[33] S. Skaperas, L. Mamatas, and V. Tsaoussidis, “A link-quality anomaly detection framework for software-defined wireless mesh networks,” *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 495–510, 2024.

[34] M. Z. Hassan, M. J. Hossain, J. Cheng, and V. C. M. Leung, “Device-clustering and rate-splitting enabled device-to-device cooperation framework in fog radio access network,” *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1482–1501, 2021.

[35] H. Lu, B. Wang, M. Li, W. Xu, H. Shen, F. Jiang, and R. Cheng, “Monthly service prediction for 4g/5g systems: A short time series-based neural network solution,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 12, pp. 2175–2188, 2026.

[36] G. Anastasi, M. Conti, and M. Di Francesco, “A comprehensive analysis of the mac unreliability problem in ieee 802.15. 4 wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 52–65, 2010.

[37] L. M. Feeney, “Towards trustworthy simulation of wireless mac/phy layers: a comparison framework,” in *ACM MSWiM*, 2012, pp. 295–304.

[38] H. Fontes, R. Campos, and M. Ricardo, “A trace-based ns-3 simulation approach for perpetuating real-world experiments,” in *Workshop on ns-3*, Jun. 2017, pp. 118–124.

[39] L. P. Fraile, S. Tsampas, G. Mylonas, and D. Amaxilatis, “A comparative study of lora and ieee 802.15.4-based iot deployments inside school buildings,” *IEEE Access*, vol. 8, pp. 160 957–160 981, 2020.

[40] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, “Optimizing and updating lora communication parameters: A machine learning approach,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 884–895, 2019.

[41] M. O. Farooq and T. Kunz, “Contiki-based ieee 802.15.4 channel capacity estimation and suitability of its csma-ca mac layer protocol for real-time multimedia applications,” *Mobile Information Systems*, vol. 2015, no. 1, p. 398637, 2015.

[42] C. Chen, J. Li, V. Balasubramaniam, Y. Wu, Y. Zhang, and S. Wan, “Contention resolution in wi-fi 6-enabled internet of things based on deep learning,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5309–5320, 2020.

[43] D. Camps-Mur, M. D. Gomony, X. Pérez-Costa, and S. Sallent-Ribes, “Leveraging 802.11n frame aggregation to enhance qos and power consumption in wi-fi networks,” *Computer Networks*, vol. 56, no. 12, pp. 2896–2911, 2012.

[44] X. Ye, Y. Yu, and L. Fu, “Deep reinforcement learning based link adaptation technique for lte/nr systems,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 7364–7379, 2023.

[45] S. Si-Mohammed and F. Theoleyre, “Data-driven prediction models for wireless network configuration,” in *International Conference on Advanced Information Networking and Applications*. Springer, 2025, pp. 442–453.

[46] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Advances in Neural Information Processing Systems*, vol. 9, 1996.

[47] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[48] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, 1986.

[49] S. Si-Mohammed and F. Theoleyre, “Per link data-driven network replication towards self-adaptive digital twins,” in *International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. IEEE, 2025.