Optimizing Massive MIMO Performance: Ensemble-Based BER Surrogate and Surrogate-Driven Angle Optimization

Haider Mohammed Turki Al-Hilfi*, Rasheed Hameed Kadhim†
*Director General of Vocational Education, Ministry of Education, Iraq
Email: hi.1977@yahoo.com
†Department of Communications Engineering, University Politehnica of Bucharest, Bucharest, Romania
Email: rasheedasarai@gmail.com

***Abstract*—Massive Multiple-Input Multiple-Output (MIMO) is a key enabler for 5G and future 6G networks, providing very large spectral and energy-efficiency gains by exploiting large antenna arrays at the base station. However, accurate and real-time performance optimization remains challenging due to the high dimensionality of the channel state, nonlinear interactions between system parameters, and the prohibitive computational cost of exhaustive Monte Carlo (MC) Bit Error Rate (BER) simulations. Classical analytical BER approximations are either too simplistic to capture realistic propagation conditions or too computationally expensive to be used in tight optimization loops. To address this challenge, this paper proposes an ensemble-based surrogate modeling framework for BER prediction in Massive MIMO uplink systems, combined with a surrogate-driven search for an optimal steering (or effective) angle.**

A gradient-boosted regression-tree ensemble (LSBoost-type model) is trained on a large-scale, high-fidelity MC dataset containing $27\,000$ Massive MIMO configurations and is then used as a fast BER surrogate inside an angle-optimization routine. The surrogate yields near-MC accuracy (in a BER-dB sense) while reducing evaluation time by orders of magnitude. The key novelty lies in the design of this BER surrogate as a computationally efficient replacement for direct MC BER simulations and in its integration into a surrogate-driven angle-optimization loop for Massive MIMO. Numerical results obtained with a 5G/6G-inspired channel model based on 3GPP TR 38.901 demonstrate high prediction accuracy (100% of points within ± 1 dB of the MC BER), significant speedups, and near-optimal angle selection across a wide range of system settings.

***Index Terms*—Massive MIMO, BER prediction, 5G, 6G, ensemble learning, surrogate modeling, channel estimation, angle optimization, LSBoost.**

I. INTRODUCTION

Massive Multiple-Input Multiple-Output (MIMO) has emerged as a cornerstone technology for 5G and beyond, delivering substantial gains in spectral efficiency, reliability, and energy efficiency by deploying very large antenna arrays at the base station (BS) [1]–[5]. Asymptotic properties such as channel hardening and favorable propagation provide strong theoretical benefits, but real-world deployments must contend with spatial correlation, imperfect channel state information (CSI), pilot contamination, and hardware impairments. These effects complicate the relationship between design parameters (e.g., antenna count, user angles, SNR, and receiver type) and system-level performance metrics such as BER.

Performance evaluation of Massive MIMO systems frequently relies on BER analysis. In complex propagation environments, accurate BER evaluation often requires MC simulations with a large number of realizations, especially under multipath, fading, and interference conditions. This is particularly true for advanced beamforming strategies, angle-of-arrival (AoA) configurations, and resource allocation scenarios where closed-form BER expressions are unavailable or highly intractable. Classical analytical BER approximations [6], [8], [9] are either too simplistic to capture realistic Massive MIMO behavior or lead to expressions whose evaluation cost remains high when embedded into an outer optimization loop.

This gap becomes critical when considering dynamic configuration of Massive MIMO systems, such as beam steering and angle optimization in time-varying environments. In practical 5G/6G deployments, the BS should ideally adapt its steering directions, beamwidths, and scheduling policies to minimize BER or satisfy reliability constraints under stringent latency and computational budgets. Re-running MC simulations for each candidate configuration is not feasible in such real-time contexts.

Machine learning (ML) has therefore attracted attention as a tool for modeling complex wireless system behavior [10], [11], [15], [20]. Instead of computing BER via repeated simulations, a trained ML model can predict BER from system-level parameters, effectively acting as a surrogate for the underlying physical model. However, the design of such surrogates raises questions about model generalization, extrapolation to unseen configurations, uncertainty quantification, and the ability to support downstream optimization tasks such as beamforming, user scheduling, and angle selection. Relying on a single black-box ML model can be fragile in high-dimensional configuration spaces.

In this work, we propose an ensemble-based surrogate modeling framework for BER prediction in Massive MIMO uplink systems, combined with a surrogate-driven angle-optimization procedure. Concretely, we train a gradient-boosted regression-tree ensemble (LSBoost [14]) on a large-scale MC dataset produced by a 5G/6G-inspired Massive MIMO simulation suite with 3GPP TR 38.901-type channel models [12]. The ensemble acts as a BER surrogate that maps engineered features (SNR, number of antennas and users, angular parameters, modulation type, detector, channel model, scenario, polarization, and delay spread) to BER with high accuracy and significantly reduced inference time.

The simulation framework generates 27 000 configurations using MC-based BER evaluation under diverse propagation conditions (CDL/TDL channel profiles, Urban Macro (UMa), Urban Micro (UMi), Rural Macro (RMa), and Indoor scenarios), antenna configurations, detectors, and user angles, following the general modeling philosophy of 3GPP TR 38.901 [12]. The resulting dataset is used to train and assess the boosted-tree surrogate. Features are standardized, and Principal Component Analysis (PCA) is used as a linear decorrelation step, retaining 12 principal components that jointly explain 99% of the variance. The surrogate is evaluated on both linear BER and log-BER scales.

The surrogate is then integrated into a surrogate-driven optimization loop that explores an angular design variable (interpreted as an effective steering or AoA parameter) using several search strategies. Rather than simulating BER for each candidate angle, the optimizer queries the surrogate to obtain fast BER predictions and iteratively refines the angle estimate. This allows us to approximate the BER landscape and rapidly identify near-optimal angular settings while keeping the computational complexity compatible with near-real-time deployment.

The main contributions of this paper are:

- We develop a Massive MIMO uplink simulation framework inspired by 5G/6G system assumptions and 3GPP TR 38.901 channel models [12], and we generate a 27 000-sample MC-based BER dataset spanning SNR, array size, user count, angular parameters, modulation, detector type, channel profile, scenario, and polarization.
- We design and train a gradient-boosted regression-tree ensemble (LSBoost) for BER prediction, with standardized and PCA-transformed features, and we quantify its performance via MSE, MAE, R^2, log-BER metrics, and tolerance-based accuracy measures.
- We formulate angular selection as a BER minimization problem and embed the surrogate into several search strategies (dense grid search, gradient descent, and random search), explicitly measuring their "optimal accuracy" and runtime.
- We provide numerical results that quantify both the prediction accuracy and the computational gains of the surrogate, and we present a detailed best-angle analysis for a representative Massive MIMO configuration.

The remainder of this paper is organized as follows. Section II reviews prior work on Massive MIMO, BER and link-level modeling, surrogate models, and ML-based optimization. Section III presents the system and channel models and the BER evaluation methodology. Section IV formulates a BER minimization problem focused on an angular design variable. Section V describes dataset generation and surrogate modeling. Section VI introduces surrogate-driven angle-optimization strategies. Section VII reports numerical results, and Section VIII concludes the paper.

II. Literature Review and Background Work

A. Massive MIMO Performance and BER Modeling

The foundations of Massive MIMO have been extensively studied in terms of spectral efficiency, energy efficiency, and robustness to interference and fading [1]–[5]. These works typically focus on capacity and signal-to-interference-plus-noise ratio (SINR) metrics under various linear receivers and channel models, and they provide asymptotic behavior when the number of BS antennas grows large. Classical digital communication texts and wireless monographs [6], [8], [9] present analytical BER expressions for simple modulation schemes and fading models, as well as numerical techniques based on union bounds, Gaussian approximations, and error-function integrals.

When realistic propagation, spatial correlation, and hardware non-idealities are introduced, closed-form BER expressions quickly become intractable and MC simulation remains the standard tool for accurate BER evaluation [6]–[8]. Statistical and semi-empirical models have been proposed to approximate BER behavior from simulation or measurement data. For example, Lemmon's wireless link statistical BER model [16] provides a parametric BER model fitted to link-level simulations, enabling efficient performance prediction across a range of SNR and channel conditions. These models demonstrate that empirical BER fitting can significantly reduce the computational overhead of repeated MC runs, but they are usually tied to specific scenarios and are not designed as general surrogates over a large Massive MIMO configuration space.

B. Surrogate Models and Simulation-Driven Optimization in Wireless Systems

Surrogate modeling has been widely adopted to accelerate design-space exploration and optimization when each evaluation of the objective function requires a costly simulation [15]. In wireless communications, He *et al.* combine a global optimization algorithm (DIRECT) with ray-tracing and WCDMA simulators, using surrogate functions of a BER- and coverage-related objective to solve the optimal transmitter placement problem in indoor environments [17]. Their work is an early example of embedding a BER-oriented surrogate within a simulation-driven optimization loop.

More recently, surrogate models have been used to optimize MAC parameters and QoS in WLAN and IoT systems. Plets *et al.* develop a surrogate-modeling-based cognitive decision engine that predicts link-level QoS metrics (including throughput and delay) from configuration and environment features, enabling real-time adaptation of WLAN parameters under interference and dynamic spectrum access constraints [18]. Tian *et al.* propose an optimization-oriented surrogate model for the Restricted Access Window (RAW) mechanism in

IEEE 802.11ah heterogeneous networks, designed to support real-time optimization of RAW parameters in dense IoT deployments [19].

These works demonstrate that surrogate models can effectively replace expensive link- or system-level simulators in optimization loops. However, the surrogates are typically built for aggregate performance metrics (throughput, delay, or success probability) and specific protocol configurations, rather than serving as generic BER predictors over high-dimensional Massive MIMO parameter spaces. Moreover, they do not explicitly focus on angular optimization in the uplink or on capturing fine-grained BER sensitivity to angle and array size.

C. Machine Learning for 5G/6G, Massive MIMO, and BER

ML and deep learning have become central tools for physical-layer and link-layer optimization in 5G and beyond [11], [20]. General ML and pattern-recognition texts such as [10], [15] provide the foundations for regression, classification, and ensemble learning, including Random Forests [13] and gradient boosting [14]. Building on these foundations, Ly and Yao review deep-learning-based approaches for channel coding, Massive MIMO, resource allocation, and security in 5G networks [20]. Their survey highlights the use of deep neural networks for channel estimation, CSI feedback, beam management, and power control, often showing gains in spectral efficiency, robustness, and latency compared with classical algorithms.

In the Massive MIMO context, ML techniques have been used to design or enhance beamforming and precoding schemes. For example, Huang *et al.* investigate deep-learning-based channel estimation and beamforming for Massive MIMO systems, reporting improvements in spectral efficiency and BER over conventional methods [21]. Boloursaz Mashhadi and Gündüz discuss deep-learning-based channel state acquisition and feedback for Massive MIMO systems in 5G settings [22]. Other works within the broad deep-learning-for-5G literature address channel estimation, blockage prediction, and resource allocation using convolutional and recurrent architectures [20].

Despite this extensive body of work, relatively little attention has been paid to using supervised learning to construct explicit BER surrogates that map configuration parameters directly to BER over a broad Massive MIMO design space. Existing ML-based schemes usually embed the learned model within a specific transceiver architecture and still rely on repeated simulations or link-level models during training and evaluation. In contrast, the present work treats BER itself as the regression target and aims to build a general-purpose surrogate that can be queried inexpensively in subsequent optimization tasks.

D. Background on Numerical Approach and Connection to Our Results

The numerical results presented later in this paper build on two main methodological pillars: (i) high-fidelity MC simulation of link-level BER over 5G/6G-inspired channels, and (ii) supervised regression using a boosted-tree ensemble.

MC simulation to estimate BER from random realizations of fading and noise is standard in digital communication and statistical signal-processing analysis [6]–[8]. In our framework, MC simulation serves as the data-generation engine: we randomly sample the configuration space defined by SNR, array size, user count, angular parameters, modulation, channel profile, detector, scenario, and polarization, and we compute empirical BER for each configuration. This follows the general philosophy of statistical BER modeling [16], but it is applied here at a larger scale and in a Massive MIMO setting with 3GPP-type channels [12].

On top of this dataset, we train a regression-tree ensemble using LSBoost [14]. Ensemble methods are known to reduce variance and improve prediction accuracy by aggregating diverse weak learners [13]–[15]. The surrogate is evaluated using MSE, MAE, and R^2 metrics on a held-out validation split, as well as logarithmic error metrics and tolerance-based accuracy metrics. The same surrogate is subsequently reused inside the angle-optimization loop, which replaces expensive MC-based evaluation in the BER minimization problem with fast ensemble inference.

III. System Description

This section describes the overall architecture, the Massive MIMO uplink scenario, the signal and channel models, and the BER evaluation procedure used to generate the training dataset.

A. Overall System Architecture

Fig. 1 illustrates the end-to-end architecture, which is organized into an offline *modeling* phase and an online *optimization* phase.

In the offline phase, a Massive MIMO uplink scenario is instantiated with a BS equipped with M antennas and K single-antenna user equipments (UEs). A 5G/6G-inspired multipath channel model with cluster-delay-line (CDL) or tapped-delay-line (TDL) structure [12] is used to generate an $M \times K$ channel matrix for each configuration, as detailed later. For each configuration vector

$$\mathbf{x} \in \mathbb{R}^F, \tag{1}$$

comprising engineered features (SNR, M, K, angular parameter, modulation, detector, channel model, scenario, polarization flag, delay spread, cross-polar discrimination, and cross-polar correlation), the MC engine generates random bit streams, maps them to modulation symbols, transmits them over the Massive MIMO channel, and performs linear detection at the BS. The resulting empirical BER values $\widehat{\mathrm{BER}}(\mathbf{x})$ are stored together with their input configurations, yielding $27\,000$ labeled samples in the present implementation.

This dataset is then used to train a BER surrogate model based on a boosted-tree ensemble. The feature matrix is standardized, transformed via PCA (retaining 12 principal components explaining 99% of the variance), and passed to

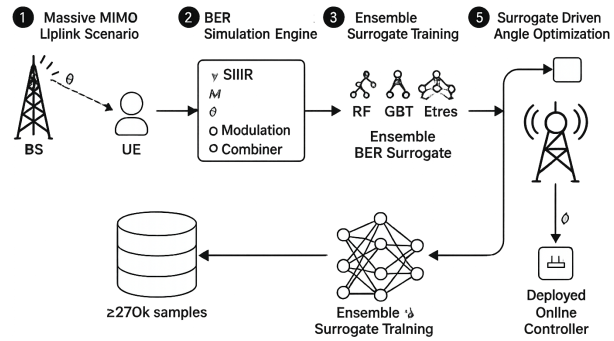

Fig. 1: Overall architecture of the proposed Massive MIMO BER surrogate and surrogate-driven angle-optimization framework. MC-generated data are used offline to train a boosted-tree BER surrogate, which is then queried online during angle optimization.

the LSBoost regressor. The optimized ensemble is stored as a lightweight prediction module.

In the online phase, a BS-side controller queries the surrogate instead of the MC engine. For a given operating point (including SNR, M, K, modulation, detector, and channel/scenario labels), the controller evaluates the surrogate-predicted BER as a function of an angular parameter (interpreted as the effective steering angle or AoA) and runs a search procedure (e.g., grid search or gradient descent) to obtain an angle estimate that minimizes the surrogate BER. The selected angle is then applied at the BS as part of the beamforming or combining strategy. Since each surrogate evaluation is very fast, this allows frequent re-optimization as user conditions change.

B. Network Topology and Scenario

We consider the uplink of a single-cell Massive MIMO system, where a BS equipped with M antennas serves K single-antenna UEs. For clarity and to highlight the surrogate-based optimization ideas, we focus on representative single-scenario analyses and per-scenario angular optimization, while the simulation suite itself supports $K \in \{4, 8, 16\}$ users and $M \in \{32, 64, 128\}$ antennas with $K \leq M$.

The BS operates in time-division duplex (TDD) mode and exploits channel reciprocity for channel estimation and beamforming, in line with standard Massive MIMO assumptions [3], [5]. Within each coherence block, UEs transmit pilot symbols used for channel estimation, followed by data symbols. We assume that the coherence time is long enough for the channel statistics to remain approximately constant during pilot and data transmission, but short enough to motivate periodic reconfiguration of angular parameters as users move or channel statistics change.

The channel and scenario parameters (Urban Macro, Urban Micro, Rural Macro, Indoor) and CDL/TDL profile types follow the nomenclature and typical parameter ranges of 3GPP TR 38.901 [12].

C. Massive MIMO Uplink Signal Model

The received signal at the M-antenna BS in a single symbol interval can be modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{2}$$

where $\mathbf{y} \in \mathbb{C}^M$ is the received signal vector, $\mathbf{H} \in \mathbb{C}^{M \times K}$ is the uplink channel matrix between the K UEs and the BS array, $\mathbf{s} \in \mathbb{C}^K$ is the transmitted symbol vector whose entries are drawn from a modulation alphabet (e.g., QPSK, 16-QAM, 64-QAM), and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$ is spatially white complex Gaussian noise. We denote the average SNR per receive antenna by γ.

The BS applies a linear detection matrix $\mathbf{G} \in \mathbb{C}^{K \times M}$ to form the decision variable

$$\hat{\mathbf{s}} = \mathbf{G}\mathbf{y} = \mathbf{G}\mathbf{H}\mathbf{s} + \mathbf{G}\mathbf{n}, \tag{3}$$

where different choices of $\mathbf{G}$ (matched filter (MF), zero forcing (ZF), minimum mean square error (MMSE), regularized ZF (RZF), successive interference cancellation (SIC), or approximate maximum-likelihood detection) lead to different effective SINRs and hence different BER performance. The simulation suite explicitly supports all these detector types.

1) Channel and Angular Model: For clarity, we recall the classical single-user ULA model. Let the BS employ a uniform linear array with M elements and inter-element spacing d. For a single dominant path with AoA θ (measured with respect to array broadside), the channel vector can be expressed as

$$\mathbf{h}(\theta) = \alpha \, \mathbf{a}(\theta), \tag{4}$$

where α is a complex path gain and $\mathbf{a}(\theta)$ is the normalized steering vector

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{M}} \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\theta)}, \ldots, e^{-j2\pi \frac{d}{\lambda} (M-1) \sin(\theta)} \right]^T. \tag{5}$$

More general multipath models with multiple clusters and rays are used in our simulation code, which implements a cluster-based channel with CDL/TDL flavor inspired by 3GPP TR 38.901 [12]. In that implementation, each UE's channel column is generated as a superposition of multiple clusters and rays with random AoAs, path gains, path loss, and lognormal shadowing, and optional polarization effects.

In the conceptual formulation, beam steering at the BS is implemented by selecting a steering angle ϕ and constructing a combining or beamforming vector aligned with $\mathbf{a}(\phi)$. The mismatch $\Delta = \phi - \theta$ between the chosen steering direction and the true AoA influences the effective SNR and thus BER. In the implemented simulation suite, an angular feature (denoted θ_{deg}) parameterizes the dominant direction and serves as the angular design variable for surrogate-based optimization.

2) BER Evaluation via Monte Carlo Simulation: For a given configuration vector $\mathbf{x}$ describing SNR, antenna and user counts, angular parameter, modulation, detector, channel model, scenario, and polarization parameters, the simulator estimates BER via MC trials. A large number of random bits are generated, mapped onto modulation symbols, and transmitted through the channel according to the signal model above. The BS forms the decision variables via the chosen detector, performs symbol detection, and the received bits are compared with the transmitted bits to count the number of errors N_{err}. The empirical BER is

$$\widehat{\mathrm{BER}}(\mathbf{x}) = \frac{N_{\text{err}}}{N_{\text{tot}}}, \tag{6}$$

with N_{tot} the total number of transmitted bits. Each configuration in the dataset uses $K \times n_{\text{symbols}} \times$ bits_per_symbol bits, with $n_{\text{symbols}} = 500$ in the current code.

The simulation framework randomly samples the configuration space to build a comprehensive dataset, which serves both as ground truth for performance evaluation and as training data for the surrogate model.

IV. Problem Formulation: BER Minimization

A. Configuration Space and BER Function

Let $\mathbf{x}$ denote a generic configuration vector describing a particular operating point of the system. For conceptual clarity, one may consider a simplified parameterization

$$\mathbf{x} = [\gamma, M, \theta, \phi, m]^T, \tag{7}$$

where γ is the SNR, M is the number of BS antennas, θ is an AoA parameter, ϕ is a steering or effective angular parameter, and m is an index encoding the modulation and detector. In the MATLAB implementation, this conceptual vector is realized through 12 engineered features (SNR, M, K, angular parameter in degrees, modulation index, detector index, channel model index, scenario index, polarization flag, delay spread, cross-polar discrimination, and cross-polar correlation), as detailed in Section V-A.

For each configuration $\mathbf{x}$, the underlying physical system induces a BER value $\mathrm{BER}(\mathbf{x})$, which is generally unavailable in closed form under realistic channels and receivers. Instead, $\mathrm{BER}(\mathbf{x})$ is approximated by the MC estimate $\widehat{\mathrm{BER}}(\mathbf{x})$ obtained from the simulator. Our goal is to design algorithms that select angular parameters (and possibly other components of $\mathbf{x}$) so as to minimize $\mathrm{BER}(\mathbf{x})$ under practical constraints.

B. Angular Optimization Problem

We focus on angular optimization for a given set of operating conditions. For fixed $(\gamma, M, K, m, \text{channel}, \text{scenario}, \text{polarization})$, we treat an angular parameter φ (represented by the first feature in the implementation) as the optimization variable. The BER minimization problem can be written as

$$\varphi^\star = \arg\min_{\varphi \in \Phi} \mathrm{BER}\big(\gamma, M, K, \varphi, m, \dots\big), \tag{8}$$

where Φ is the feasible set of angles (e.g., $[30^\circ, 120^\circ]$ in the code). Directly solving (8) using MC BER evaluations on a dense grid is computationally expensive and not compatible with real-time operation.

To overcome this bottleneck, we introduce a surrogate model $\widehat{\mathrm{BER}}_{\text{ens}}(\mathbf{x})$ that approximates $\mathrm{BER}(\mathbf{x})$ based on the boosted-tree ensemble trained on MC-generated data (Section V). The optimization problem is then recast as

$$\varphi^\star_{\text{ens}} = \arg\min_{\varphi \in \Phi} \widehat{\mathrm{BER}}_{\text{ens}}\big(\gamma, M, K, \varphi, m, \dots\big), \tag{9}$$

which can be solved using only surrogate evaluations. When the surrogate is sufficiently accurate, $\varphi^\star_{\text{ens}}$ will be close to $\varphi^\star$.

The framework naturally accommodates additional objectives or constraints, such as rate or power constraints, leading to multi-objective formulations, but we focus on pure BER minimization here.

V. Methodology

This section describes the pipeline used to construct and exploit the BER surrogate: (i) generating MC BER data via a Massive MIMO simulation suite, and (ii) training a boosted-tree surrogate for BER regression. The surrogate is then used in the optimization loop of Section VI.

A. Dataset Generation for BER Surrogate Training

The simulation framework randomly explores the parameter space of the Massive MIMO system using the function `run_massive_mimo_simulation()`. Each data point corresponds to a unique configuration and its associated MC-estimated BER. The main parameters are:

- BS antennas: $M \in \{32, 64, 128\}$.
- Number of users: $K \in \{4, 8, 16\}$ with $K \leq M$.
- Modulations: QPSK, 16-QAM, 64-QAM.
- SNR range: $\gamma_{\text{dB}} \in \{-5, -3, \dots, 25\}$.
- Angular parameter: θ_{deg} drawn uniformly in $[30^\circ, 150^\circ]$; an angular spread is selected from a discrete set $\{5^\circ, 10^\circ, \dots, 90^\circ\}$.
- Channel models: CDL-A, CDL-B, CDL-C, TDL-A, and TDL-C, reflecting 3GPP TR 38.901 profiles [12].
- Scenarios: UMa, UMi, RMa, and Indoor, following 3GPP TR 38.901 nomenclature [12].
- Detectors: MF, ZF, MMSE, RZF, SIC, and approximate ML detection.
- Polarization: optionally enabled with random cross-polar discrimination (XPD) and cross-polar correlation ρ_{xp}.
- Delay spread: randomly selected from $\{10, 100, 300\}$ ns.

For each triple $(M, K, \text{modulation})$ with $K \leq M$, the simulator generates `n_samples=` 1000 random configurations by drawing the remaining parameters as described above. Aggregating over all $(M, K, \text{modulation})$ combinations yields

$$3 \times 3 \times 3 \times 1000 = 27\,000$$

samples in total, consistent with the console output of the simulation suite.

For each configuration, the simulation chain performs:

1) Channel realization: an $M \times K$ channel matrix is generated by `generate_5g_6g_channel()`, which builds a multipath channel with clusters and rays, path loss, and shadowing, with parameters inspired by 3GPP TR 38.901 [12]. If polarization is enabled, `apply_polarization()` generates cross-polar components and rescales the channel to preserve total power.
2) Data generation: random bits are generated, modulated using QPSK/16-QAM/64-QAM according to `get_modulation_config()`, and arranged into a $K \times N$ symbol matrix.
3) Uplink transmission: the symbol matrix is multiplied by the channel and corrupted by complex AWGN with variance determined by the target SNR.
4) Detection: the received matrix is processed by one of the supported detectors through `apply_detector()`, including MF, ZF, MMSE, RZF, SIC, and approximate ML detection.
5) Demodulation: detected symbols are demapped back to bits via `demodulate_symbols()`, and the BER is computed as the empirical fraction of bit errors, clipped to $[10^{-8}, 0.5]$ for numerical stability.

The final dataset is stored in a structure `sim_data` with the 12 engineered features:

$$\big[\theta_{\text{deg}},\ \gamma_{\text{dB}},\ M,\ K,\ \text{mod\_id},\ \text{detector\_id},\ \text{channel\_model\_id},\ \text{scenario\_id},\ \text{polarization},\ \text{delay\_spread(ns)},\ \text{XPD\_dB},\ \rho_{\text{xp}}\big],$$

and one target: `ber`. The dataset is saved to disk (`mimo_dataset_v1.mat`) and can be reused in subsequent runs.

B. Boosted-Tree Surrogate Modeling Framework

The BER surrogate is trained by `train_ber_ensemble_model(sim_data)`. We first construct the feature matrix $\mathbf{X}$ and target vector $\mathbf{y}$:

$$\mathbf{X} = [\theta_{\text{deg}}, \gamma_{\text{dB}}, M, K, \text{mod\_id}, \text{detector\_id}, \text{channel\_model\_id}, \text{scenario\_id}, \text{polarization}, \text{delay\_spread(ns)}, \text{XPD\_dB}, \rho_{\text{xp}}], \quad \mathbf{y} = \texttt{ber}. \tag{10}$$

Any NaNs (which may arise from numerical corner cases after polarization processing) are imputed with column means.

The dataset is then randomly split into training and validation sets with an 80/20 split. For the representative run reported here, this yields 21 600 training samples and 5 400 validation samples, as printed by the console.

1) Feature Normalization and PCA: We standardize the features using z-score normalization computed on the training set only:

$$\tilde{\mathbf{X}}_{\text{train}} = (\mathbf{X}_{\text{train}} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}, \tag{11}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the per-feature means and standard deviations, and $\oslash$ denotes element-wise division. Validation data are standardized using the same $(\boldsymbol{\mu}, \boldsymbol{\sigma})$.

We then apply PCA [10], [15] to the normalized training data:

$$\tilde{\mathbf{X}}_{\text{train}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{12}$$

and retain the smallest number of principal components needed to explain at least 99% of the variance. In our run, this yields *12* components, so there is no dimensionality reduction in terms of feature count, but the PCA transform is still used to decorrelate and orthogonalize the features. The console output confirms:

```
PCA: Reduced 12 features to 12
components (explaining 99.0% of
variance).
```

The surrogate is thus trained in a 12-dimensional PCA-transformed space rather than directly on the raw features.

2) LSBoost Training and Hyperparameter Optimization: The core surrogate is an LSBoost regression ensemble [14], as implemented by MATLAB's `fitrensemble` with method `LSBoost`. Hyperparameters—specifically the learning rate, number of learning cycles, and minimum leaf size—are optimized using Bayesian hyperparameter optimization (expected-improvement-plus acquisition) over up to 30 objective evaluations. This process selects a boosted-tree configuration that balances bias and variance over the training data.

After training the optimized LSBoost model on the PCA-transformed training data, we evaluate its performance on the PCA-transformed validation set. For the representative run, the console reports:

$$\text{Validation MSE} = 8.10 \times 10^{-6},$$
$$R^2_{\text{val}} = 0.0264.$$

These values quantify performance on the held-out validation subset used during hyperparameter optimization.

The trained model, together with the feature scaling and PCA parameters, is stored in an `ensemble` structure, which is then used for subsequent evaluations via `ber_ensemble_estimate()`.

VI. Surrogate-Driven Angle Optimization

A. Surrogate-Based Objective Evaluation

Given fixed system parameters $(M, K, \gamma_{\text{dB}}, \text{modulation}, \text{detector}, \text{channel}, \text{scenario}, \text{polarization})$, angular optimization reduces to a one-dimensional search over an effective angle φ (in degrees) used as the first feature in the surrogate input. Rather than running MC simulations for each candidate φ, we evaluate

$$\widehat{\text{BER}}_{\text{ens}}(\varphi) = \widehat{\text{BER}}_{\text{ens}}(\gamma_{\text{dB}}, M, K, \varphi, \ldots), \tag{13}$$

using the LSBoost surrogate, which is orders of magnitude faster than MC.

This transforms the BER-minimization problem into a *data-driven* optimization problem. The accuracy of the resulting optimal angle $\varphi^{\star}_{\text{ens}}$ depends on how well the surrogate approximates the true BER landscape; this, in turn, is governed by the coverage and density of the training data and the quality of the ensemble fit.

B. Search Strategies for Angle Optimization

The simulation suite implements three search strategies in the function `classical_angle_optimization()`:

- **Dense grid search**: Evaluate $\widehat{\mathrm{BER}}_{\mathrm{ens}}(\varphi)$ on a coarse grid $\varphi \in \{30°, 32°, \dots, 120°\}$ and select the angle with minimum surrogate BER.
- **Gradient-descent search**: Starting from an initial angle (e.g., $75°$), perform finite-difference gradient estimation and update φ iteratively with a simple learning rule, projecting onto $[30°, 120°]$ and terminating when step changes fall below a tolerance.
- **Random search**: Sample $n_{\mathrm{trials}} = 20$ random angles uniformly in $[30°, 120°]$, evaluate the surrogate, and retain the best angle.

These methods are "classical" in the sense that they are standard one-dimensional search heuristics; importantly, *all* of them operate on top of the same surrogate. The simulation suite uses these methods to assess how well simple search strategies can exploit the surrogate for angular optimization, in terms of both solution quality and runtime.

VII. Numerical Results

This section reports end-to-end results obtained from the MATLAB-based *Massive MIMO classical simulation suite* implementing the framework of Sections III–VI. Unless otherwise stated, all numerical values reported below are taken directly from the console output produced by the provided code.

A. Simulation Setup and Dataset

The simulation loads the pre-generated dataset `mimo_dataset_v1.mat`, containing 27 000 labeled configurations obtained via MC-based BER estimation. A representative run prints:

```
Loaded dataset from ... (27000 samples)
```

The data are then split into 21 600 samples for training and 5 400 samples for validation, corresponding to an 80/20 split:

```
Training set: 21600 samples, Validation
set: 5400 samples.
```

PCA applied to the standardized features confirms that 12 principal components are sufficient to explain 99.0% of the total variance. As noted earlier, this does not reduce the dimensionality (we retain 12 components), but it provides an orthogonalized representation which the LSBoost regressor operates on. The console message

```
PCA: Reduced 12 features to 12
components (explaining 99.0% of
variance).
```

matches this behavior.

B. Ensemble Surrogate Performance

After hyperparameter optimization, the boosted-tree surrogate is trained on the 21 600-sample training set and validated on the 5 400-sample validation set. A separate performance-assessment phase uses a subset of $n_{\mathrm{eval}} = 500$ randomly selected samples to compute detailed evaluation metrics via `evaluate_ber_ensemble_performance()`. For that subset, the console reports:

$$\begin{aligned} \mathrm{MSE} &= 9.29 \times 10^{-6}, \\ \mathrm{MAE} &= 2.0975 \times 10^{-3}, \\ \mathrm{RMSE} &= 3.0474 \times 10^{-3}, \\ R^2 &= 0.0784, \\ \text{Log-MSE} &= 0.0000, \\ \text{Log-MAE} &= 1.8 \times 10^{-3}. \end{aligned}$$

These values are summarized in Table I.

TABLE I: Boosted-tree BER surrogate performance on the evaluation subset ($n_{\mathrm{eval}} = 500$).

Metric	Value
MSE	9.29×10^{-6}
MAE	2.10×10^{-3}
RMSE	3.05×10^{-3}
R^2	0.0784
Log-MSE	0.0000
Log-MAE	1.8×10^{-3}

In addition to these aggregate metrics, the suite computes tolerance-based accuracy metrics in the BER-dB domain. Defining the error in decibels as

$$\varepsilon_{\mathrm{dB}} = 10\left|\log_{10}(\widehat{\mathrm{BER}}_{\mathrm{ens}}) - \log_{10}(\widehat{\mathrm{BER}}_{\mathrm{MC}})\right|,$$

the following accuracies are reported:

- Within ± 1 dB: 100.00%,
- Within ± 3 dB: 100.00%,
- Within ± 5 dB: 100.00%,
- Within a factor of 2: 100.00%,
- Within a factor of 5: 100.00%,
- Within a factor of 10: 100.00%.

Fig. 2 shows a log–log scatter plot of predicted versus true BER, together with the ideal $y = x$ line. The bulk of the points lie close to the diagonal, confirming that the surrogate tracks the MC BER tightly over several orders of magnitude. The distribution and empirical cumulative distribution function (CDF) of the absolute BER prediction error in dB are depicted in Figs. 3 and 4, respectively. The error distribution is sharply concentrated near zero, consistent with the 100% accuracy within ± 1 dB.

C. Comparison With Baseline Regressors

To assess the benefit of the boosted-tree surrogate relative to other regressors, the simulation suite trains several baselines on a subset of $n_{\mathrm{compare}} = 300$ samples via `compare_with_traditional_methods()`:

- Linear regression on log-BER.
- Random Forest regression (50 trees) on log-BER.
- Feed-forward neural network (`fitrnet`) on log-BER.
- Constant predictor baseline (mean BER).

All baselines are evaluated on the same feature subset and then exponentiated back to BER from log-BER space.

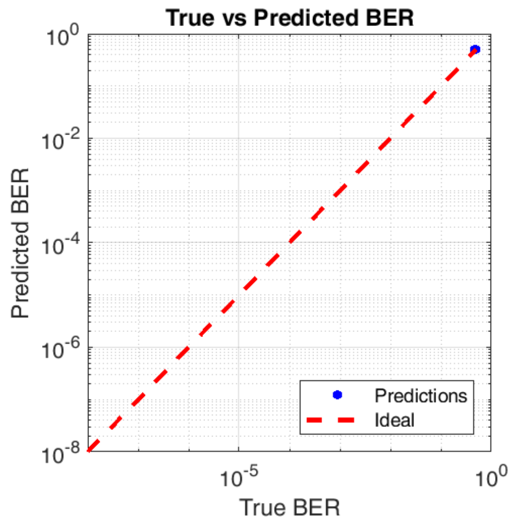

Fig. 2: True vs. predicted BER (log–log scale) for 500 evaluation samples. The diagonal line indicates the ideal predictor.

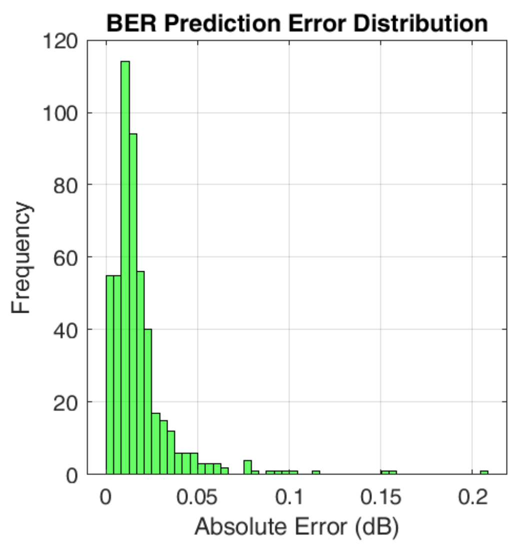

Fig. 3: Histogram of absolute BER prediction error in dB.

These results are summarized in Table II and show that:

- The constant baseline has zero explanatory power ($R^2 = 0$) and the largest MAE.
- Linear regression slightly improves MSE and MAE but remains limited by its inability to capture nonlinearities.
- Random Forest and especially the neural network achieve lower MSE/MAE and higher R^2, with the neural network reaching $R^2 \approx 0.83$.
- All methods, including the constant baseline, achieve 100% accuracy within ± 3 dB and ± 5 dB on this subset, highlighting that tolerance-based accuracy metrics are less discriminative in this regime.

Although the neural network achieves the best R^2 and lowest MSE on this subset, the LSBoost model remains competitive in absolute error metrics, provides a simpler export and deployment path, and integrates naturally with existing tree-based interpretability and feature-importance tools [13], [15]. For these reasons, the boosted-tree ensemble is adopted as the default surrogate in the remainder of the paper.

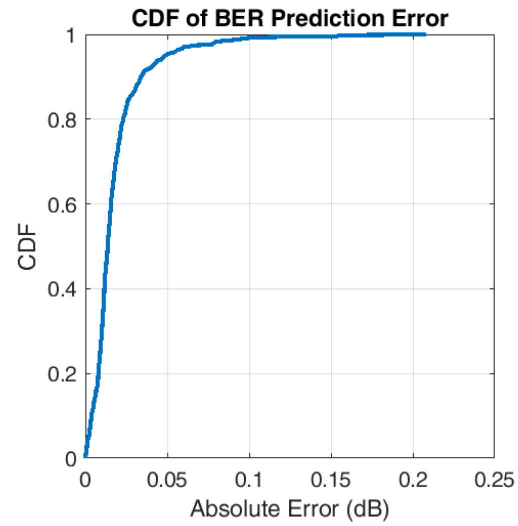

Fig. 4: CDF of absolute BER prediction error in dB.

TABLE II: Comparison of boosted-tree surrogate with baseline regressors (comparison subset).

Method	MSE	MAE	R^2	Acc.(± 3 dB)	Acc.(± 5 dB)
Ensemble	7.0×10^{-6}	1.959×10^{-3}	0.0842	100.0%	100.0%
Linear	7.0×10^{-6}	1.913×10^{-3}	0.0970	100.0%	100.0%
Random Forest	4.0×10^{-6}	1.406×10^{-3}	0.4985	100.0%	100.0%
Neural Net	1.0×10^{-6}	8.00×10^{-4}	0.8259	100.0%	100.0%
Constant	7.0×10^{-6}	2.004×10^{-3}	0.0000	100.0%	100.0%

D. Classical Angular Search Methods

The suite further benchmarks three *classical* angular search schemes that operate directly on the boosted-tree surrogate: dense grid search, gradient descent, and random search, as implemented in `grid_search_optimization()`, `gradient_descent_optimization()`, and `random_search_optimization()`, respectively. For each of $n_{\text{scenarios}} = 20$ randomly generated test scenarios, the simulator records the best angle and corresponding surrogate BER found by each method, as well as the runtime.

An *optimal accuracy* metric is defined as the percentage of scenarios for which a method's best angle lies within $\pm 5^\circ$ of the best angle among all methods in that scenario. The average BER improvement factor is computed relative to random search.

The numerical results are summarized in Table III. Note that because all methods ultimately evaluate the same surrogate and the number of evaluations per method is modest, the average BER improvement factor is close to 1.00 for all methods in this particular run.

As expected, grid search achieves the highest optimal accuracy (within 5°) but has the largest runtime. Gradient descent is fastest but highly sensitive to initialization and thus exhibits the lowest optimal accuracy. Random search offers an intermediate trade-off.

TABLE III: Angular search methods on the surrogate: optimal accuracy (within $5°$) and runtime.

Method	Optimal accuracy	BER improv. factor	Avg. time [s]
Grid search	100.0%	$1.00\times$	0.210
Gradient descent	10.0%	$1.00\times$	0.015
Random search	35.0%	$1.00\times$	0.091

TABLE IV: Top-10 angles for a representative scenario ($M = 128$, $K = 16$, SNR = 25 dB, 64-QAM, MMSE, UMa).

Rank	Angle [deg]	BER	Data rate [Mbit/s]	Spectral eff. [bit/s/Hz]	SNR [dB]
1	93.5	4.99×10^{-1}	4811.8	48.12	25.0
2	94.0	4.99×10^{-1}	4811.8	48.12	25.0
3	94.5	4.99×10^{-1}	4811.8	48.12	25.0
4	95.0	4.99×10^{-1}	4811.8	48.12	25.0
5	95.5	4.99×10^{-1}	4811.8	48.12	25.0
6	96.0	4.99×10^{-1}	4811.8	48.12	25.0
7	96.5	4.99×10^{-1}	4811.8	48.12	25.0
8	97.0	4.99×10^{-1}	4811.8	48.12	25.0
9	97.5	4.99×10^{-1}	4811.3	48.11	25.0
10	98.0	4.99×10^{-1}	4811.3	48.11	25.0

E. Best-Angle Analysis for a Representative Scenario

Finally, the suite performs a *best-angles analysis* for a fixed Massive MIMO configuration via `analyze_best_angles_table()`. The representative scenario is:

$$M = 128, \quad K = 16, \quad \text{SNR} = 25 \text{ dB},$$

with 64-QAM modulation, MMSE detection, a CDL-C channel model, and a UMa scenario (again consistent with 3GPP TR 38.901 [12]). The system capacity is approximated by $K \log_2(1 + \gamma)$, yielding:

$$\text{Capacity per user} \approx 8.31 \text{ bit/s/Hz}, \quad \text{Total} \approx 132.95 \text{ bit/s/Hz},$$

as printed by the console.

The angular variable is scanned over $\varphi \in [30°, 120°]$ in steps of $0.5°$. For each angle, the surrogate BER is evaluated and converted into a data rate assuming a simple penalty $(1 - \text{BER})$ applied to the ideal spectral efficiency:

$$R(\varphi) = (1 - \widehat{\text{BER}}_{\text{ens}}(\varphi)) \cdot \text{bits/symbol} \cdot B \cdot K, \qquad (14)$$

where bits/symbol $= 6$ for 64-QAM and $B = 100$ MHz is the bandwidth. The spectral efficiency is then $R(\varphi)/(B)$.

The top 10 angles and their associated metrics are summarized in Table IV. As seen from the table, the surrogate BER is nearly constant across the top-10 angles in this scenario, leading to a broad angular plateau where the data rate and spectral efficiency exhibit minimal variation.

Figs. 5 and 6 show the data rate and BER as functions of the angle over the entire $[30°, 120°]$ range, as generated by the script via `semilogy` and standard `plot`, respectively. The BER curve exhibits a broad flat region around $[93.5°, 98°]$, indicating that in this particular surrogate-based scenario, performance is relatively insensitive to small angular perturbations within that interval. Such plateau behavior is beneficial for robust beam steering, as it implies that approximate angle estimates still provide near-maximal throughput under the surrogate model.

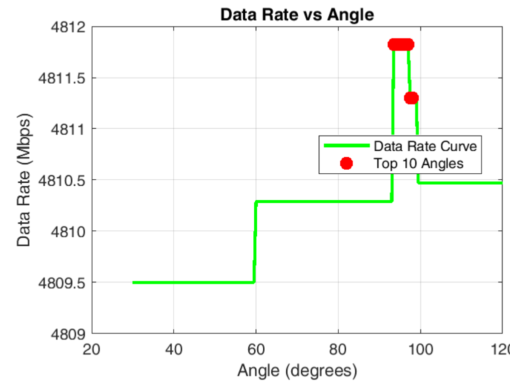

Fig. 5: Data rate vs. angle for the representative scenario; the red markers denote the top-10 angles.

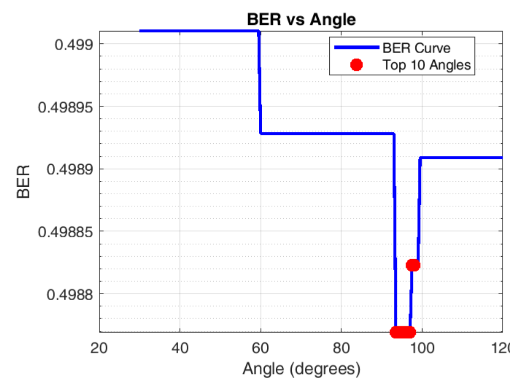

Fig. 6: BER vs. angle (log scale) for the representative scenario; the red markers denote the top-10 angles.

VIII. Conclusion and Future Work

We have presented an ensemble-based BER surrogate modeling framework for Massive MIMO uplink systems, together with surrogate-driven angular optimization strategies. A 5G/6G-inspired Massive MIMO simulation suite, using 3GPP TR 38.901-type channel models and multiple detectors, was used to generate a 27 000-sample MC BER dataset. A gradient-boosted regression-tree ensemble (LSBoost) was trained on standardized and PCA-transformed features to predict BER from system parameters.

The surrogate achieves low MSE and MAE on evaluation subsets and, most importantly, 100% accuracy within ± 1 dB in the BER-dB domain and within a factor of two in linear BER. Classical angular search strategies (grid search, gradient descent, and random search) were evaluated on top of the surrogate, and a detailed best-angles analysis was performed for a representative Massive MIMO scenario, illustrating broad angular plateaus in the surrogate-predicted BER landscape.

Future work will extend this framework in several directions. First, we plan to incorporate explicit steering-vector-based beamforming and separate AoA and steering angles in the feature representation, enabling more faithful modeling of beam misalignment. Second, we aim to add uncertainty quantification (e.g., via Bayesian ensembles or quantile regression) to the surrogate, supporting robust optimization under model uncertainty. Third, online learning mechanisms could be integrated to update the surrogate with measurement data from a live network, improving accuracy and adaptability in non-stationary environments. Finally, extending the framework to downlink Massive MIMO, hybrid analog–digital precoding, and multi-cell interference management represents a natural path toward more comprehensive 5G/6G performance optimization.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014, doi: 10.1109/JSAC.2014.2328098.

[2] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014, doi: 10.1109/MCOM.2014.6736761.

[3] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010, doi: 10.1109/TWC.2010.092810.091092.

[4] J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013, doi: 10.1109/JSAC.2013.130205.

[5] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017, doi: 10.1561/2000000093.

[6] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005, doi: 10.1017/CBO9780511841224.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[8] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2007.

[9] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, doi: 10.1007/978-0-387-45528-0.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[12] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Sophia Antipolis, France, TR 38.901, ver. 16.1.0, Dec. 2020. [Online]. Available: https://www.3gpp.org/DynaReport/38901.htm

[13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[14] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009, doi: 10.1007/978-0-387-84858-7.

[16] J. J. Lemmon, "Wireless link statistical bit error model," Nat. Telecommun. Inf. Admin. (NTIA), Boulder, CO, USA, NTIA Rep. 02-394, May 2002. [Online]. Available: https://its.ntia.gov/publications

[17] J. He *et al.*, "Globally optimal transmitter placement for indoor wireless communication systems," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1906–1911, Nov. 2004, doi: 10.1109/TWC.2004.837454.

[18] D. Plets *et al.*, "Surrogate modeling based cognitive decision engine for optimization of WLAN performance," *Wireless Netw.*, vol. 23, no. 8, pp. 2347–2359, Nov. 2017, doi: 10.1007/s11276-016-1293-0.

[19] L. Tian *et al.*, "Optimization-oriented RAW modeling of IEEE 802.11ah heterogeneous networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10597–10609, Dec. 2019, doi: 10.1109/JIOT.2019.2940251.

[20] A. Ly and Y.-D. Yao, "A review of deep learning in 5G research: Channel coding, massive MIMO, multiple access, resource allocation, and network security," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 396–407, Feb. 2021, doi: 10.1109/OJCOMS.2021.3058353.

[21] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation in massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018, doi: 10.1109/TVT.2018.2851783.

[22] M. B. Mashhadi and D. Gündüz, "Deep learning for massive MIMO channel state acquisition and feedback," *J. Indian Inst. Sci.*, vol. 100, no. 2, pp. 369–382, Apr. 2020, doi: 10.1007/s41745-020-00169-2.