SJER Scientific Journal of Engineering Research
futuristech PT. Teknologi Futuristik Indonesia
Date of publication January 11, 2026, date of current version January 11, 2026.
e-ISSN: 3109-1725
Digital Object Identifier 10.64539/sjer.v2i1.2026.364

Article

Hybrid Machine Learning Framework for Joint Prediction of Window Mean and Bit Error Rate in SC-LDPC Decoding

Tanzeela Bibi[1,*], Hua Zhou[2], Sana Akbar[1], Lalit Awasthi[3]

[1] School of Electronics and Information Engineering, Nanjing University of Information Science &Technology, Nanjing, 210044, China; tanzeelabibi26@gmail.com
[2] School of Electronics and Information Engineering, Institute of Communication Engineering, Nanjing, 210044, China; hzhou@nuist.edu.cn
[3] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, 210044, China; 202452200021@nuist.edu.cn; lawasthi12@gmail.com
* Correspondence

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Abstract: Modern low-latency communication systems increasingly rely on spatially coupled low-density parity-check (SC-LDPC) codes combined with windowed decoding (WD) to achieve high reliability with reduced latency and memory requirements. However, evaluating the intrinsic trade-off between decoding complexity and error performance typically measured by the average window iteration count (W_{MEAN}) and bit error rate (BER) still depends on computationally intensive Monte Carlo simulations, which limits rapid system optimization and real-time design exploration. To address this limitation, this paper proposes a hybrid machine learning framework for the joint, non-iterative prediction of W_{MEAN} and BER using a single set of code and channel parameters. A high-fidelity dataset is generated through extensive SC-LDPC windowed decoding simulations across varying window sizes, coupling lengths, and signal-to-noise ratio (SNR) conditions. Based on this dataset, a multi-output Random Forest Regressor is trained to exploit the shared underlying decoding dynamics that govern both computational complexity and decoding reliability. The proposed model achieves accurate simultaneous prediction of W_{MEAN} and BER, demonstrating strong generalization performance while significantly reducing system evaluation time compared to conventional simulation-based approaches. Feature-importance analysis further reveals the dominant influence of channel quality and coupling structure on both decoding effort and error performance. These results indicate that the proposed framework provides an effective surrogate modeling tool for fast design-space exploration and informed performance–complexity trade-off analysis. The methodology enables practical optimization of high-throughput SC-LDPC decoders and supports the development of adaptive and resource-efficient communication systems.

Keywords: Hybrid Machine Learning; SC-LDPC Codes; Windowed Decoding (WD); Random Forest Regressor (RFR); Joint Prediction; Bit Error Rate (BER); Decoding Complexity; Window Mean (W_{MEAN}); Surrogate Modeling.

1. Introduction

Low-density parity-check (LDPC) codes have played a fundamental role in modern channel coding theory since their introduction by Gallager [1], with their near-capacity performance later justified within Shannon's information-theoretic framework [2]. Building upon these foundations, spatially coupled LDPC (SC-LDPC) codes have emerged as a powerful class of error-correcting codes, exhibiting the threshold saturation phenomenon that enables iterative belief-propagation (BP) decoding to approach maximum-a-posteriori (MAP) performance as the coupling length increases [3]-[5]. Owing to these properties, SC-LDPC codes are widely regarded as strong candidates for high-reliability communication systems, including optical networks, satellite links, and high-throughput wireless applications [6], [7].

Despite their excellent asymptotic performance, practical implementation of SC-LDPC codes introduces significant challenges. Conventional full-chain BP decoding incurs high computational complexity, large memory requirements, and increased latency, particularly for long coupled chains [8], [9]. To mitigate these limitations, windowed decoding (WD) has been proposed as an effective low-latency alternative that restricts iterative decoding to a sliding window over the coupled graph [10]. By limiting the decoding scope, WD significantly reduces memory usage and latency while preserving near-threshold error-correction performance [3], [6]. However, this advantage comes at the cost of an inherent trade-off between decoding complexity and error-rate performance, which is influenced by window size, coupling parameters, and channel conditions [11]-[13].

Accurately characterizing this performance–complexity trade-off remains a major challenge in SC-LDPC system design. Performance evaluation of windowed decoding typically relies on extensive Monte Carlo simulations to estimate metrics such as bit-error rate (BER) and decoding effort, particularly in low-error-rate regimes [14]. These simulations are computationally expensive and time-consuming, limiting rapid design-space exploration, hardware optimization, and real-time system adaptation. As a result, there is a growing need for efficient predictive tools that can estimate decoder behavior without executing full iterative decoding.

In parallel, machine learning (ML) techniques have gained increasing attention in the context of LDPC and SC-LDPC decoding. Prior works have explored ML-assisted decoding strategies, including near-ML decoding algorithms, reinforcement-learning-based scheduling, and reliability-driven decoding enhancements [15]-[18]. Reduced-complexity decoding methods, such as weighted bit-flipping and reliability-based approaches, have also been investigated to alleviate computational burden while maintaining acceptable error performance [19]-[21]. While these approaches focus on improving the decoding process itself, they still require iterative decoding execution and do not address the problem of predicting decoder behavior at a system level.

Motivated by this gap, this work proposes a hybrid machine learning framework for the joint, non-iterative prediction of two key performance metrics in SC-LDPC windowed decoding: the average window iteration count (window mean), which reflects decoding complexity, and the resulting bit-error rate (BER), which reflects decoding reliability. By leveraging a multi-output Random Forest Regressor trained on a high-fidelity dataset generated via controlled Monte Carlo simulations, the proposed framework exploits the intrinsic correlation between decoding complexity and error performance. This unified learning approach enables accurate surrogate modeling of SC-LDPC decoder behavior across diverse system configurations, significantly reducing reliance on computationally expensive simulations.

The main contributions of this work are summarized as follows:

- A joint surrogate modeling framework for SC-LDPC windowed decoding that simultaneously predicts decoding complexity and error performance.
- A multi-output machine learning approach that captures the inherent dependence between window mean and BER, outperforming independent single-output prediction models.
- A comprehensive evaluation and feature-importance analysis that provides insight into how system parameters influence decoding behavior, offering practical guidance for SC-LDPC design and optimization.

The rest of this paper is organized as follows: Section 2 reviews recent advances in SC-LDPC codes and decoding strategies, including windowed decoding, high-throughput hardware implementations, reduced-complexity decoding algorithms, and machine-learning-based optimization approaches. Section 3 presents the figure illustrating the methodological workflow. Section 4 presents the proposed hybrid methodology, describing the SC-LDPC system model, dataset generation process, multi-output Random Forest Regressor architecture, and evaluation metrics. Section 5 reports and analyzes the simulation results, including feature importance analysis and comparative performance evaluation against baseline models. Finally, Section 6 concludes the paper by summarizing the main findings and outlining potential directions for future research.

2. Background and Related Work

This section reviews prior research relevant to spatially coupled LDPC (SC-LDPC) codes, windowed decoding strategies, reduced-complexity decoding methods, and machine-learning-assisted approaches. The focus is on technical developments and limitations that motivate the proposed surrogate modeling framework.

2.1. Spatially Coupled LDPC Codes and Windowed Decoding

SC-LDPC codes extend conventional LDPC constructions by introducing structured coupling between adjacent code blocks, which significantly improves iterative decoding thresholds through the phenomenon of threshold saturation [3]-[5]. Analytical and design-oriented studies have demonstrated that spatial coupling enables belief-propagation (BP) decoding to achieve near-optimal performance under a wide range of channel conditions [3], [5]. These properties have been explored for both binary and non-binary SC-LDPC codes, highlighting the robustness of spatial coupling across different code structures [3], [22].

To address the high latency and memory requirements of full-chain BP decoding, windowed decoding (WD) has been proposed as a practical alternative [10]. WD restricts message passing to a sliding window of fixed size, allowing partial decoding of the coupled graph while maintaining strong error-correction performance. Several works have analyzed the impact of window size, coupling parameters, and decoding schedules on the convergence behavior and error performance of WD [10], [12], [13]. Design-oriented studies have further examined how window size and coupling structure influence finite-length performance and decoding stability [5], [7].

2.2. Decoding Complexity and Performance Trade-Offs

A fundamental challenge in LDPC and SC-LDPC decoding is the trade-off between decoding complexity and error performance. Information-theoretic analyses have established lower bounds on decoding effort and highlighted how graph structure and iterative decoding dynamics influence computational complexity [8], [9]. Practical studies have shown that decoding complexity per iteration and total decoding effort are strongly dependent on channel conditions and code parameters [8], [11].

For windowed decoding, this trade-off becomes more pronounced due to localized decoding and inter-window dependencies. Error propagation across window boundaries and premature stopping can significantly degrade performance if decoding parameters are not carefully chosen [12], [13]. Adaptive window scheduling and variable window strategies have been proposed to mitigate these effects and improve robustness under challenging channel conditions [23], [24]. However, these approaches still rely on iterative decoding execution and require extensive simulation to evaluate their performance.

2.3. Reduced-Complexity Decoding Algorithms

To alleviate the computational burden of iterative BP decoding, a variety of reduced-complexity decoding algorithms have been proposed. Weighted bit-flipping (WBF) and its variants offer simplified decoding by selectively updating unreliable bits, achieving lower complexity at the cost of some performance degradation [21]. Hardware-oriented implementations have demonstrated that such approaches can significantly reduce computational load and power consumption in practical decoders [17], [19].

Reliability-driven methods, including BP-LED and related algorithms, further reduce complexity by selectively erasing or reprocessing unreliable symbols during decoding [18]. These techniques have been analyzed under various channel models, including AWGN, and have shown improved performance–complexity trade-offs compared to conventional BP decoding [18], [20]. While effective, these methods still operate within the decoding loop and do not eliminate the need for iterative message passing.

2.4. Machine Learning in LDPC and SC-LDPC Decoding

Recent years have seen growing interest in applying machine learning to LDPC decoding. ML-based approaches have been explored to approximate near-maximum-likelihood decoding, optimize message-passing schedules, and enhance decoder robustness [15], [16]. Reinforcement learning has been employed to learn adaptive decoding policies for sparse graph-based codes, demonstrating performance improvements over fixed scheduling strategies [16].

In addition to decoder-centric approaches, ML techniques have been investigated for performance modeling and system-level analysis. However, existing studies primarily focus on improving decoding algorithms rather than predicting decoding outcomes. As a result, there is limited work on surrogate models that estimate decoding complexity and error performance directly from system parameters, without executing full iterative decoding. This gap becomes particularly significant for SC-LDPC windowed decoding, where decoding complexity and reliability are intrinsically coupled and expensive to evaluate through simulation.

2.5. Research Gaps

There are still a number of significant gaps in the present literature, despite the substantial advancements mentioned above. Few studies attempt to jointly estimate decoding complexity and error-rate performance from system parameters, despite the fact that several discuss the hardware complexity, finite-length behavior, and decoding performance of SC-LDPC systems. The majority of current efforts concentrate on enhancing the decoding process itself, whether through learning-based decoders, algorithmic improvements, reliability-based corrections, or scheduling optimizations. However, they do not offer prediction tools that estimate decoder behavior prior to carrying out complete iterative decoding.

Despite growing in popularity, machine-learning-assisted decoding has mostly focused on improving message passing or optimising decoding schedules rather than creating surrogate models for system-level assessment. This creates a gap in approaches that do not require computationally demanding Monte Carlo simulations in order to quickly estimate performance indicators such as BER or average iteration count. Related studies have demonstrated the effectiveness of data-driven sur-

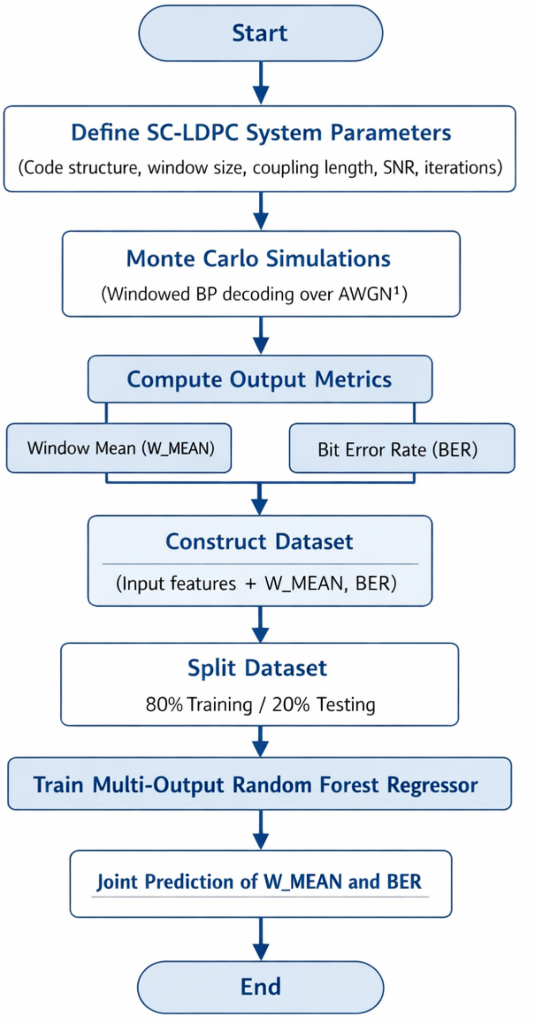

Figure 1. Methodological workflow illustrating the research stages used to develop and evaluate the proposed hybrid machine learning framework.

rogate modeling frameworks for system-level performance estimation, motivating the methodology adopted in this work [25].

Via building a multi-output Random Forest model trained on a large dataset produced via controlled SC-LDPC simulations, the current study directly overcomes these shortcomings. The suggested surrogate model reflects the associated behavior of decoding difficulty and reliability by forecasting both the average number of iterations per window W_{MEAN} and the consequent BER. Further information about how window size, SNR, coupling parameters, and other design elements interact to affect decoder performance may be found in the accompanying feature-importance study. Thus, our predictive approach offers a useful substitute for conventional full-scale simulation-based evaluation and is a major step towards data-driven design and optimization of SC-LDPC systems.

3. Research Stages

This study follows a structured, multi-stage research workflow designed to ensure transparency, reproducibility, and systematic development of the proposed hybrid machine learning framework. The overall methodology progresses from system modeling and data generation to predictive modeling and performance evaluation. Each stage builds logically upon the previous one, enabling a clear traceability between the problem formulation and the final outcomes.

Figure 1 illustrates the complete methodological workflow adopted in this work, highlighting the sequential research stages used to obtain the reported results.

3.1. Stage 1: System Modeling and Parameter Definition

The research begins with the definition of the SC-LDPC communication system under an AWGN channel. Key code, decoding, and channel parameters such as window size, coupling length, lifting factor, maximum iterations, and signal-to-noise ratio (SNR) are specified to represent realistic decoding scenarios under windowed belief propagation (BP).

3.2. Stage 2: Monte Carlo Simulation and Data Generation

Extensive Monte Carlo simulations are performed using windowed decoding to emulate practical SC-LDPC decoder behavior. For each configuration of system parameters, decoding is executed until predefined statistical stopping criteria are satisfied, ensuring reliable estimation of performance metrics.

3.3. Stage 3: Performance Metric Extraction

From each simulation run, two complementary performance metrics are extracted:

- Window Mean (W_{MEAN}), representing the average number of decoding iterations per window and serving as a measure of decoding complexity.
- Bit Error Rate (BER), representing decoding reliability and communication performance.

These metrics jointly characterize the internal computational effort and external error performance of the decoder.

3.4. Stage 4: Dataset Construction

The simulation outputs are aggregated to form a labeled dataset consisting of input feature vectors (system and channel parameters) and corresponding output targets (W_{MEAN} and BER). This dataset serves as the ground truth for training and validating the machine learning models.

3.5. Stage 5: Machine Learning Model Training

The dataset is partitioned into training and testing subsets. A multi-output Random Forest Regressor is then trained to jointly learn the nonlinear relationship between the input parameters and both output metrics. Joint learning enables the model to exploit the intrinsic correlation between decoding complexity and error performance.

Table 1. Input Features for the Multi-Output Regression Model.

Parameter	Symbol	Description	Value / Range
Lifting Factor	L_c	Determines the degree of code coupling and resulting block length.	{1,2,4,8}
Window Size	W_s	Size of the decoding window (in coupled blocks).	[8,64] (Integer)
Code Rate	R_c	Ratio of information bits to total bits.	0.5 (Fixed)
Channel SNR	SNR_{dB}	Signal-to-Noise Ratio of the AWGN channel.	[1.5,4.0] dB
Max Iterations	I	Maximum iterations allowed for the BP decoder.	50 (Fixed)
Coupling Factor	C_f	Degree of coupling between protographs.	3 (Fixed)

3.6. Stage 6: Model Evaluation and Validation

The trained model is evaluated on unseen test data using standard regression metrics, including the coefficient of determination (R^2), root mean square error (RMSE), and mean absolute error (MAE). This stage verifies the model's generalization capability and prediction accuracy across diverse SC-LDPC configurations.

3.7. Stage 7: Outcome Analysis

Finally, the trained surrogate model is analyzed to assess prediction performance and extract insights into the influence of system parameters on decoding complexity and reliability. These outcomes demonstrate the effectiveness of the proposed framework as a fast and accurate alternative to computationally expensive Monte Carlo simulations.

4. Proposed Hybrid Methodology

The creation of a data-driven model that can concurrently forecast two crucial performance metrics—decoder complexity (W_{MEAN}) and error rate (BER)— from a single set of code and channel parameters forms the basis of this study. The system model, the procedure for creating the dataset, the multi-output regression model's design, and the metrics utilized for thorough assessment are all described in this part.

4.1. SC-LDPC System Model and Data Synthesis

The method is based on a simulated communication system that employs a spatially-coupled low-density parity-check (SC-LDPC) code with a fixed-rate ($R_c = 0.5$) additive white gaussian noise (AWGN) channel. The decoding process uses the traditional Windowed Decoding (WD) technique, which is based on the Belief Propagation (BP) algorithm, in accordance with the specified system model.

Input Features (X) have six fundamental, non-redundant factors make up the machine learning model's input vector X which collectively define the channel state, the decoding process, and the code structure. Table 1 lists these attributes together with the sampling ranges used to generate the dataset.

4.1.1 Decoding Process and Message Passing

In order to minimize time at the expense of error correcting capability, the WD process applies the iterative belief update to a sliding window of size W_s throughout the connected chain. Passing messages between variable nodes (v) and check nodes (c) on the Tanner Graph representation of the code is the main function of this procedure, which is a specific application of Factor Graph decoding. Each node separately changes its belief about whether a bit is accurate or whether a parity check is satisfied. This repeated, local information exchange is the core of the decoding process. The recursive formula provides the variable node update equation, which establishes the message $m_{n\to c}^{(t)}$ delivered from variable node n to check node c at iteration t.

$$m_{n\to c}^{(t)} = N_n + \sum_{a' \in H(v)\backslash c} m_{a'\to c}^{t-1} \tag{1}$$

The channel log-likelihood ratio (LLR), which is the first extrinsic information regarding the reliability of the received bit straight from the noisy channel, is represented here by N_n. Before any decoding is done, this LLR is essentially a measure of the belief state. The set of all check nodes linked to the variable node (v), omitting the check node (c), is denoted as H(v)\c. The summing term compiles all of the extrinsic data (parity constraints) that were obtained from every other check node in the preceding iteration (t − 1). By continuously integrating the noisy channel evidence with the structural constraints imposed by the code, this iterative process guarantees that the decision on a bit's value is refined. This leads to an improved estimation with each subsequent iteration until a valid code word is found or a maximum iteration limit is reached. A practical implementation element that is essential for high-throughput systems is the use of the sliding window (W_s), which reduces latency and memory use by limiting the computationally demanding belief propagation to a smaller, localized portion of the received data.

Output Targets (Y) shows for a given input X, the multi-output model is intended to generate a two-dimensional output vector Y = $[W_{MEAN}, BER]$. These metrics measure both the outward communication dependability and the internal computing cost at the same time.

Table 2. summarizes the configuration and hyperparameter settings of the machine learning models used to evaluate the proposed W_{MEAN} and BER prediction framework.

Category	Description / Setting
Learning Paradigm	Supervised regression
Prediction Strategy	Joint (multi-output) prediction
Target Metrics	Window Mean (W_{MEAN}), Bit Error Rate (BER)
Primary Model	Multi-Output Random Forest Regressor
Baseline Models	Linear Regression (LR), Decision Tree Regressor (DT), Single-Output Random Forest Regressors
Input Features	Window size, coupling length, lifting factor, channel SNR *(code rate and max iterations fixed)*
Number of Trees	100
Tree Depth	Grown until purity or minimum sample threshold
Feature Selection per Split	Random subset of input features
Bootstrap Sampling	Enabled
Training/Test Split	80% / 20%
Random Seed	Fixed for reproducibility
Optimization Criterion	Mean Squared Error (MSE)
Evaluation Metrics	R^2, RMSE, MAE
Prediction Mode	Non-iterative surrogate estimation

a. Window Mean(W_{MEAN}): This represents the average number of iterations carried out throughout all decoding windows (M) during a simulation run. It is a direct indicator of the decoder's computing effort and acts as the main internal complexity metric. Increased processing time and power consumption are indicated by a greater W_{MEAN} value, which is usually necessary when the channel quality is low or the code is running close to its performance limit. This is how the metric is computed.

$$W_{MEAN} = \frac{1}{M}\sum_{j=1}^{M} I_j \quad (2)$$

Where I_j denotes how many iterations were carried out on the j^{th} decoding window. The metric offers a reliable, empirical evaluation of the dynamic operational load of the decoder, which varies according to the degree of channel impairments, by averaging this number of iterations.

b. Bit Error Rate (BER): This is the proportion of wrongly decoded bits N_{err} to all sent information bits N_{total}. It controls the total quality of service and is the crucial external indicator of communication dependability. After the decoding process is finished, it indicates the system's ultimate failure rate, and its prediction is crucial for system design and specification. The BER is computed as follows:

$$\text{BER} = \frac{N_{err}}{N_{total}} \quad (3)$$

4.1.2. Dataset Generation

Extensive Monte Carlo simulations were used to create a high-fidelity training dataset of 2,000 distinct labeled samples. The most resource-intensive stage is creating this dataset, but it is essential to ensuring the surrogate model's predicted accuracy. Every simulation point was run under strict termination requirements to guarantee statistical stability for the low error rate labels—a need for accurately characterizing system performance at realistic operating points. In particular, the simulation was run for a specific set of **X** parameters until it reached a minimal criteria of either 100 frame errors or 10^7 decoded bits. This extensive simulation depth reduces the possibility of noise and bias in the training labels by ensuring that even extremely small error probabilities are predicted with a high degree of certainty. The robust multi-output regression model is trained and validated using this synthetic dataset, which serves as the essential ground truth.

4.2. Multi-Output Regression Model Architecture

The Random Forest Regressor (RFR) algorithm, which was deliberately selected for its resilience in managing the intricate, non-linear, and frequently discontinuous relationships typical of contemporary coding system performance measures, is the basis for the Multi-Output Regression Model used in the suggested framework. RFR is a strong and dependable option for non-parametric regression because it can implicitly handle complex feature interactions without requiring explicit modeling and because it is relatively insensitive to feature scaling and possible outliers in the Monte Carlo dataset.

The machine learning models were configured to jointly predict decoding complexity and decoding reliability, represented by the window mean (W_{MEAN}) and bit error rate (BER), respectively. A multi-output Random Forest Regressor was selected as the primary model due to its ability to capture nonlinear relationships and shared dependencies between the two target metrics.

Baseline models, including linear regression, decision tree regression, and independent single-output Random Forest models, were used for comparative evaluation. In Table 2, all models were trained using the same input feature set and data partitioning to ensure a fair and consistent performance comparison.

4.2.1. Model Justification

Theoretically, it is crucial to use a single multi-output model, $f(\mathbf{X}) \rightarrow \mathbf{Y}$, instead of training two different single-output models, (one for W_{MEAN} and one for BER). The correlation and shared structure between the two goal metrics (W_{MEAN} and BER) may be captured and used by the model thanks to this unified methodology. The interdependence between the internal complexity and the exterior reliability are closely related because they are both produced concurrently by the same physical process—the iterative belief-propagation decoder acting on the code and channel parameters. For instance, circumstances that limit the ability to repair errors (higher BER) are frequently the same ones that increase the decoding effort (higher W_{MEAN}). The model gains from a richer shared representation when they are trained together, using the prediction of one metric to increase the accuracy of the other. When compared to standalone training, this integrated learning approach eventually improves generalization and prediction accuracy while condensing the final predictive model for more streamlined and effective deployment in real-world communication system controllers.

4.2.2. Random Forest Regressor Details

By creating an aggregate forecast from T distinct decision trees, the RFR functions as an ensemble learning technique. Two basic sources of randomness are used to build each tree independently: random feature selection, which only takes into account a random subset of the input features at each split point, and bootstrap aggregation (bagging), which trains each tree on a random subset of the training data with replacement. This dual-randomization method maximizes the model's capacity to generalize to unforeseen code and channel circumstances by lowering the model's variance and reducing the chance of overfitting to noisy training data. By mathematically combining the predictions from each individual tree, $\hat{\mathbf{Y}}_t$, the final multi-output prediction $\hat{\mathbf{Y}}$ is obtained for each given input vector. In particular, the final result is the ensemble's average of each tree prediction:

$$\hat{\mathbf{Y}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{Y}}_t\,(\mathbf{X}) \tag{4}$$

The stability and robustness of the final model output are guaranteed by this averaging procedure. Each component tree $\hat{\mathbf{Y}}_t(\mathbf{X})$ predicts the entire target vector $[\hat{y}_{W_{MEAN}}, \hat{y}_{BER}]$ in this multi-output scenario, and the ensemble average is calculated independently for each target dimension. Setting the Number of Estimators (T) to 100, which offers a steady and enough ensemble size to take advantage of the advantages of aggregation without incurring excessive computing cost, was one of the key hyper-parameters used in the training process. Additionally, until all leaves were pure (containing samples of just one class) or contained fewer than a specified number of samples, the maximum tree depth was permitted to grow naturally. By avoiding early constraints, this approach enables the model to accurately represent the intricacy of the underlying data structure. The Monte Carlo dataset was carefully divided into a testing set (20%) that was utilized only for the final, objective performance evaluation and a sizable training set (80%) that was used for model fitting. To guarantee that the entire assessment procedure is statistically repeatable, a fixed random seed was carefully applied during this partitioning.

4.2.3. Comparative Reference Models Used for Evaluation

Three types of baseline models were used in order to thoroughly evaluate the performance of the suggested Multi-Output Random Forest Regressor (RFR) model. In order to provide a fair and understandable performance comparison, these comparative references were chosen to reflect gradually rising levels of model complexity.

a) Linear Regression (LR): The most straightforward benchmark is linear regression, which offers a lower-bound reference for prediction accuracy. Since LR is a completely linear model, it makes the assumption that the input features and the output variables have a straight proportional connection. Incorporating LR is crucial to illustrate the underlying complexity of the learning problem, even though this assumption is rarely true for real-world signal or system-level prediction problems. It draws attention to how much feature interactions and non-linear correlations affect prediction quality. Therefore, the extra value of using non-linear modeling techniques is directly reflected in performance advantages above LR.
b) Decision Tree Regressor (DT): By recursively dividing the feature space to represent intricate, hierarchical relationships in the data, the Decision Tree model adds non-linearity. In contrast to LR, DT can naturally capture sudden changes, thresholds, or interactions among predictors because it does not reliant on global functional assumptions. However, a single tree usually shows considerable variance and is prone to overfitting. Because of this, DT is a perfect mid-tier baseline to highlight the significance of ensemble averaging. Enhancements over DT demonstrate the

value of employing aggregated estimators, like Random Forests, to produce predictions that are more reliable and broadly applicable.

c) Two Single-Output RFR Models: Two distinct Random Forest Regressor models were independently trained, one for predicting W_{MEAN} and another for BER, in order to assess the role of joint learning. These single-output RFR models already provide a solid foundation by including the benefits of non-linear representation and ensemble learning. The advantages of simultaneous prediction can be directly evaluated by contrasting them with the suggested Multi-Output RFR. Shared feature representations, less training duplication, and enhanced generalization through cross-variable dependency learning are possible benefits. It shows that joint modeling successfully reflects the intrinsic correlation between W_{MEAN} and BER if the multi-output strategy performs better than these separately optimized models.

4.3. Evaluation Metrics

Three commonly used regression measures were used to objectively quantify the multi-output Random Forest Regressor (RFR) model's performance on the unseen 20% test subset. This rigorous evaluation procedure is essential for guaranteeing the model's capacity to generalize to new, untested code and channel parameters as well as for offering an objective evaluation of its prediction potential. The following metrics are applied to a collection of n test samples, where y_i is the actual target value and $\hat{y}_i$ is the model's predicted value:

a) Coefficient of Determination(R^2): This measure calculates the percentage of the dependent variable's variance (either W_{MEAN} or BER) that can be predicted from the independent input variables $(\mathbf{X})$. Essentially, R^2 measures how closely the model's predictions resemble the actual data points. Since the baseline for R^2 is 0, the model is no more effective than just forecasting the target variable's mean for each input. The mean of the actual target values across the test set is represented by $\bar{y}$. Excellent predictive power and a high degree of fit between the model and the actual data are shown by a score near 1, which suggests that the model effectively captures the underlying theoretical links between the code/channel parameters and the performance indicator. The R^2 score is computed using the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5}$$

b) Root Mean Square Error (RMSE): The square root of the average of the squared errors is represented by this metric. It gives the average magnitude of the mistake in the target variable's particular units (e.g., FLOPs for W_{MEAN} or a unitless value for BER). The RMSE is especially susceptible to outliers or notable prediction failures since squaring the errors before averaging gives larger errors more weight. The RMSE is an important measure of the precision of the model since engineering applications need minimizing prediction mistakes. The RMSE is computed as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

c) Mean Absolute Error (MAE): The average of the absolute discrepancies between the actual observation and the prediction is represented by this metric. Because MAE penalizes all mistakes linearly, it is less susceptible to outliers than RMSE and offers an easy-to-understand picture of the usual prediction error magnitude. When it comes to practically evaluating the average deviation one might anticipate from the model, MAE is especially helpful. MAE offers a readily comprehensible measure of the average error in estimating the error rate and the complexity of the decoder, respectively, for the two targets. The MAE is computed as follows:

$$MAE = \frac{1}{n}\sum_{i=}^{n}|y_i - \hat{y}_i| \tag{7}$$

A thorough evaluation profile is produced by combining these three metrics: RMSE emphasizes the significance of significant mistakes, R^2 evaluates overall fit, and MAE provides a reliable indicator of average prediction accuracy.

5. Results and Analysis

The dataset, the interdependence of important variables, and the predictive behavior of the suggested Multi-Output Random Forest Regressor (RFR) are all thoroughly analyzed. The objective is to jointly estimate two important performance metrics of the SC-LDPC decoder: the bit-error rate (BER) for decoding reliability and the mean window parameter W_{MEAN} for decoding complexity. In order to comprehend the distribution, variation, and correlations among input parameters like coupling width, lifting size, node degrees, window size, SNR values, and decoding iterations, the dataset is analyzed. These features have a direct impact on the decoding pro-

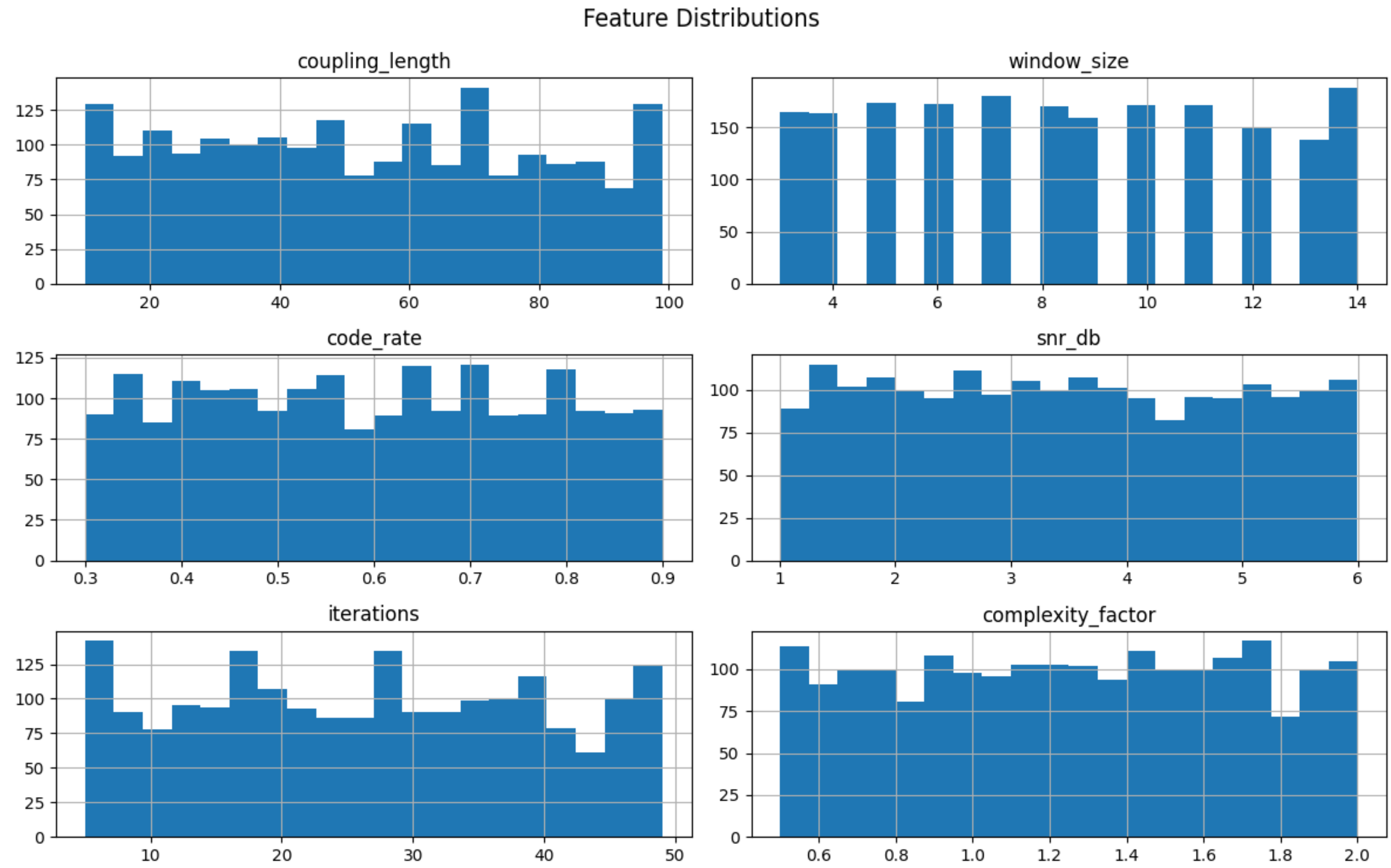

Figure 2. Feature Distributions.

cess's difficulty and the error-rate performance that results. Knowing these dependencies guarantees that the learning model represents the interactions that arise in real-world SC-LDPC decoding settings in addition to the individual effects of each parameter. The ability of the Multi-Output RFR to learn the joint mapping between the two outputs and the system parameters is then assessed. The model takes use of correlations between complexity and reliability, which frequently change jointly as decoding conditions change, by simultaneously predicting W_{MEAN} and BER. Metrics including mean absolute error, coefficient of determination (R2), and cross-validation performance are used to evaluate the model's prediction accuracy, generalizability, and resilience. Overall, the extended analysis shows how the multi-output learning technique enables effective design and optimization of SC-LDPC systems by offering a unified framework for predicting both computing cost and decoding performance.

5.1. Data Characteristics and Relationships

To comprehend the statistical behavior of the input parameters and their relationship to the goal variables, a preliminary analysis of the dataset was carried out. Before training the prediction model, this first study is crucial to confirming the data's quality, balance, and informativeness. It is feasible to determine if the dataset fully captures the range of SC-LDPC decoding scenarios, including both favorable and difficult operating conditions, by examining the distributions, ranges, and variability of the features. Such an analysis also aids in identifying any problems that can impair model performance, such as skewed feature ranges, missing values, outliers, or redundancy. Furthermore, examining the connections between input parameters and desired outputs reveals which characteristics might have the biggest impact on complexity and dependability and provides early insight into the underlying decoding behavior. This fundamental knowledge guarantees that the learning process that follows is based on a dataset that is both statistically sound and representative of actual system dynamics, ultimately allowing the Multi-Output RFR model to produce predictions that are more precise and dependable.

The two goal outputs and the histograms for each of the six input features are shown in Figure 2. These distributions show how frequently various parameter values occur and if the dataset sufficiently covers the whole SC-LDPC configuration design space. The dataset is well-balanced and free of substantial clustering or biases that could distort the learning process, as evidenced by the comparatively consistent distribution throughout the ranges of SNR, coupling width, lifting size, node degrees, and iteration counts. This wide coverage guarantees that the dataset contains both normal and edge-case decoding situations, such as high-SNR scenarios where decoding stabilizes and complexity decreases, as well as low-SNR

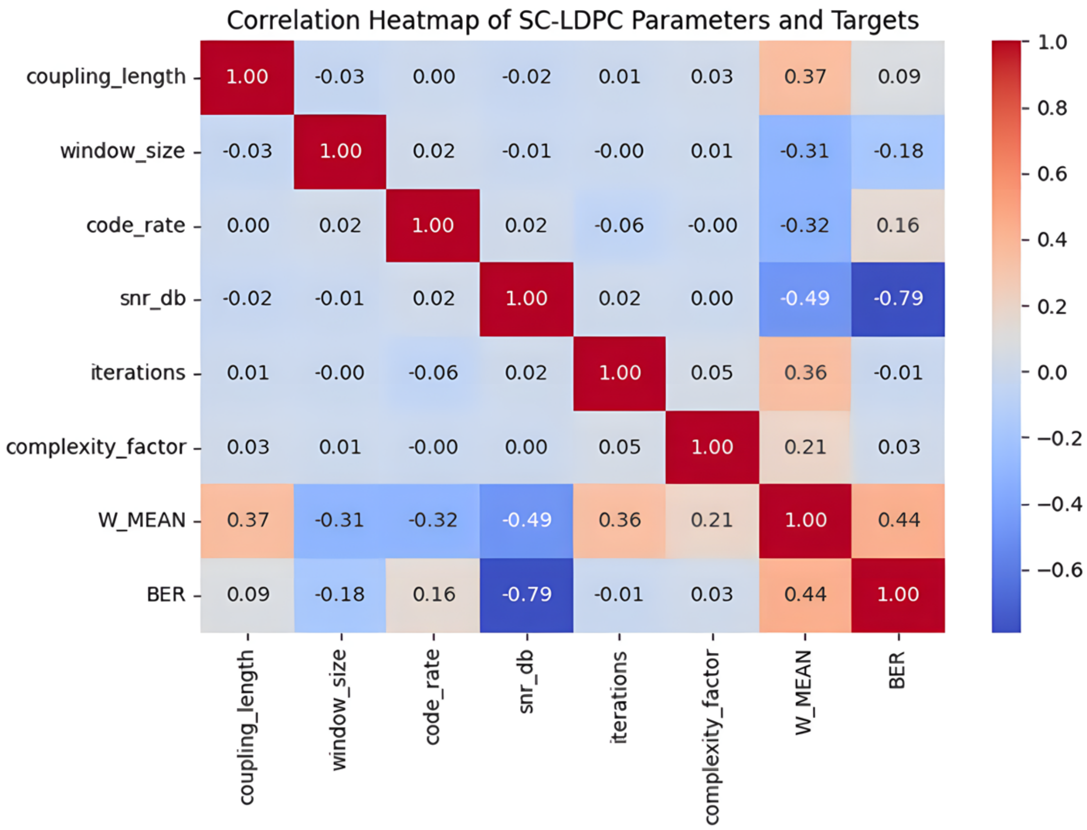

Figure 3. Correlation Heatmap.

regimes linked to high BER and higher decoding complexity. The learning model is exposed to the inherent unpredictability and nonlinear behavior of SC-LDPC decoding by using such a broad range of operational points. As a result, the Multi-Output RFR can consistently generalize to unknown configurations, learn resilient patterns instead of memorizing isolated cases, and more successfully capture delicate relationships between parameters. In the end, this varied feature space representation improves the model's predictive power and helps produce more precise estimates of decoding complexity and dependability.

By displaying the pairwise Pearson correlation coefficients between all variables, Figure 3 delves deeper into these correlations and provides a better understanding of how various parameters affect decoder behavior. According to known communication theory, there is a strong negative connection between SNR_{dB} and BER $\rho = -0.79$, which confirms that the probability of bit errors drastically lowers as channel conditions improve (i.e., higher SNR). This robust correlation demonstrates that SNR is the primary determinant of reliability in SC-LDPC decoding. Furthermore, a moderately positive correlation ($\rho = 0.44$) between W_{MEAN} and BER shows that worse channel conditions not only raise error rates but also make the decoder work harder, either by requiring more iterations, larger window sizes, or more time to reach the convergence threshold. This implies that reliability and decoding difficulty are intrinsically related, with difficult circumstances putting greater demands on the decoding process.

Moreover, decoding complexity is significantly influenced by other characteristics. The impact of code structure is seen in the moderate connection between coupling length and W_{MEAN} ($\rho = 0.37$). Longer coupling lengths generally introduce higher memory and connectivity between sections, which can affect how many window locations must be processed during decoding. Iteration count also correlates with W_{MEAN} ($\rho = 0.36$), supporting the intuitive hypothesis that longer iterative updates typically result in higher decoding effort. When taken as a whole, these correlations demonstrate how channel quality, code structure, and decoder configuration all work together to shape the computational and performance properties of SC-LDPC systems.

In addition to the correlation analysis, Figure 4 shows density contours and scatter plots that graphically depict the relationships and interactions between the most important factors. Higher SNR values consistently correspond to lower error rates, demonstrating the inverse relationship between SNR and BER across all pertinent plots. This reinforces the channel quality's major influence on decoding reliability. The scatter plots show more nuanced correlations and nonlinear patterns that go beyond this main trend and are not well represented by straightforward correlation coefficients For example, as SNR drops, W_{MEAN} gradually rises, indicating the decoder's increased processing work in noisy environments.

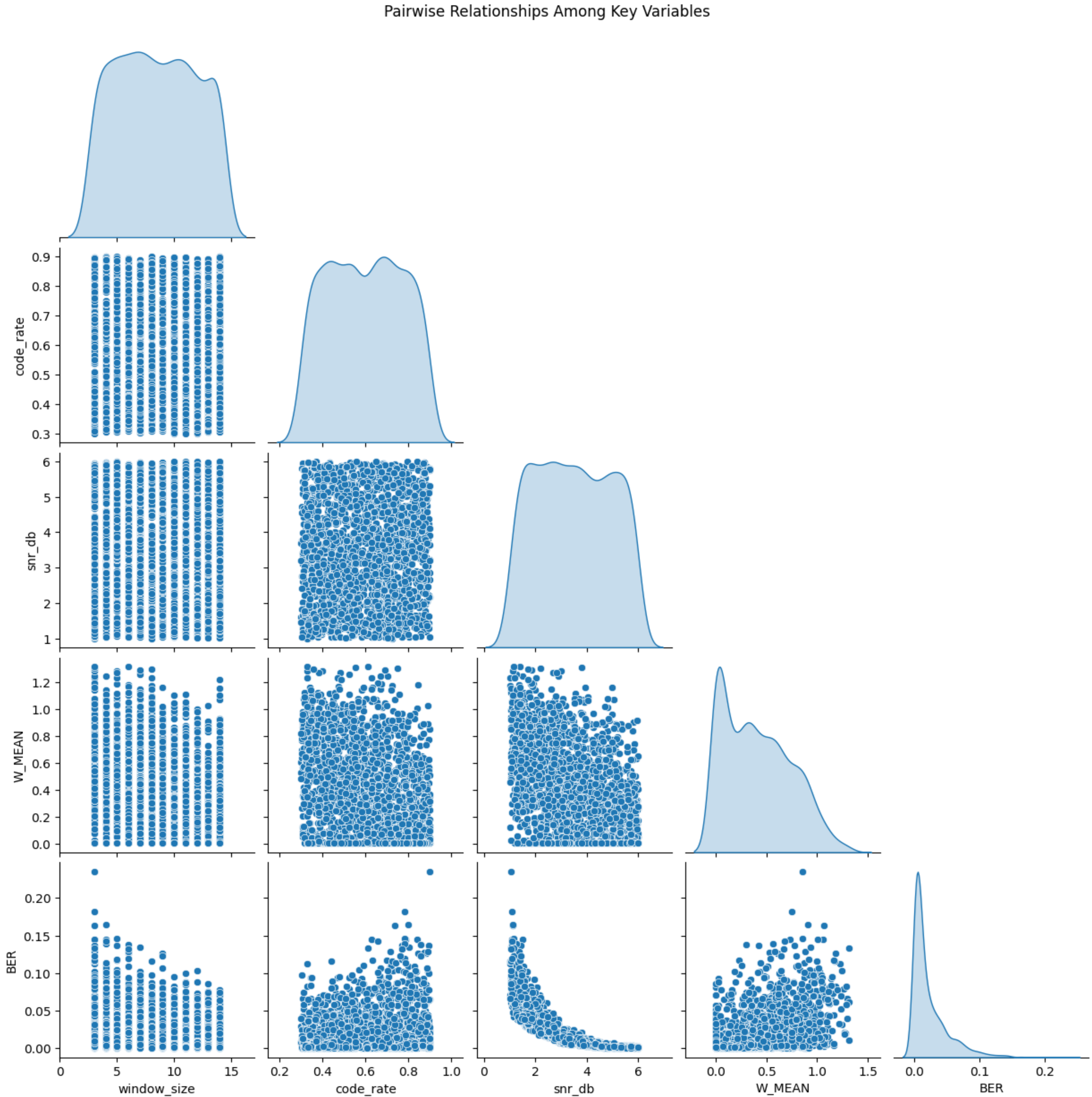

Figure 4. Pairwise Relationship Among Key Variable.

The influence of code design on decoding complexity is further demonstrated by the correlation between increased W_{MEAN} and increases in coupling length and other code-structural factors. In addition to highlighting areas of high concentration, the density contours illustrate less common, extreme scenarios that the model must also manage and demonstrate where parameter combinations frequently occur. All things considered, these pairwise visualizations demonstrate that the dataset encompasses a broad range of SC-LDPC operating circumstances, offer a deeper understanding of the interactions within the dataset, and highlight the nonlinear correlations between input parameters and target measures. This thorough understanding provides a solid basis for training the Multi-Output Random Forest Regressor, guaranteeing that it can acquire precise and broadly applicable mappings from system characteristics to both decoding difficulty and reliability.

5.2. Feature Importance Analysis

Feature significance scores were taken from the trained Multi-Output Random Forest Regressor (RFR) model in order to determine which input features have the greatest impact on decoding complexity and reliability. A better knowledge of the variables influencing SC-LDPC decoder performance is made possible by this analysis, which offers insightful information about how each system parameter impacts the target metrics BER and W_{MEAN}. It is feasible to identify which characteristics have a more marginal impact on decoding behaviour and which have a dominant function by calculating the relative contribution of each feature.

Each feature's individual influence as well as its relationships with other parameters are reflected in the feature significance scores, which are calculated depending on how much each feature lowers the prediction error across all trees in the forest. Features with high significance scores have a significant impact on decoding results since they have a big influence on the model's decisions. Lower scores, on the other hand, indicate traits that are redundant in relation to other factors or that provide less information. In addition to helping to interpret the model's predictions, this analysis offers system designers useful advice: by concentrating on the most important parameters, one can optimize SC-LDPC configurations more successfully, possibly lowering decoding complexity or enhancing error-rate performance without needless modifications to less significant settings.

All things considered, feature importance analysis acts as a link between domain expertise and data-driven modeling, emphasizing important factors influencing SC-LDPC decoder behavior and guiding both model improvement and practical system design choices.

SNR_{dB} is the most important parameter, accounting for about 30% of the model's predictive performance, followed by coupling length (~19%) and iteration count (~17%), according to Figure 5 (Feature Importance Plot). Since greater SNR values considerably lower error rates and, thus, the necessary decoding effort, these results unequivocally show the major importance of channel quality in influencing both decoding reliability and computing complexity. Longer coupling creates more interdependence among variable nodes, which in turn influences the mean window size and the number of iterations required for convergence. This significant contribution of coupling length emphasizes the significance of code structure. Another important characteristic is the number of iterations, which indicates the decoder's direct control over computational burden. More iterations inevitably result in more decoding effort, particularly in difficult channel conditions.

While they affect decoder performance, the remaining features such as lifting size, node degrees, and window size show smaller but nonetheless significant contributions, indicating that their effects are more nuanced or context-dependent. When taken as a whole, these insights offer a clear hierarchy of variables influencing SC-LDPC decoding, connecting decoder configurations, code design decisions, and physical channel circumstances to reliability and complexity results. This knowledge not only confirms that the model is capable of capturing realistic dependencies, but it also provides system designers with useful advice: while changes to less significant features may result in marginal gains, optimizing the most influential parameters can yield significant improvements in decoding efficiency and performance.

5.3. Multi-Output Model Prediction Performance

To ensure an objective evaluation of the Multi-Output Random Forest Regressor (RFR) model's capacity for generalization, its predictive performance was thoroughly assessed using a different 20% test set that was not used during training. This assessment sheds light on the model's accuracy in estimating reliability (BER) and decoding complexity (W_{MEAN}) across various SC-LDPC setups.

Key performance indicators, such as the Coefficient of Determination R^2, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) for each target variable, are compiled in Table 3. Higher values indicate stronger predictive power. The R^2 values show the percentage of variance in the target variables that the model can explain. While MAE offers a complementary measure of average absolute error and an intuitive sense of usual prediction accuracy, RMSE quantifies the average magnitude of prediction errors, highlighting bigger variations. When combined, these indicators provide a thorough understanding of the model's performance by striking a balance between overall predictive consistency and sensitivity to outliers.

The findings show that the Multi-Output RFR can effectively capture the intricate, nonlinear interactions between the decoder's performance measures and system parameters. High R^2 values along with low RMSE and MAE attest to the model's strong generalization to new data, accurately forecasting bit-error rates and computational complexity under a range of channel and code conditions. These results demonstrate the multi-output learning approach's promise as a dependable tool for system optimization and performance evaluation by validating its applicability for simultaneously modeling interdependent SC-LDPC decoding outputs.

The model effectively captures more than 71% of the variance in decoding complexity across the test set, as indicated by the $\boldsymbol{R^2}$ value of 0.7106 for $\boldsymbol{W_{MEAN}}$. Achieving this level of explanatory power indicates strong generalization capability given the highly nonlinear and dynamic nature of the SC-LDPC decoding process, where complexity is influenced simultaneously by channel conditions, coupling structure, local node degrees, and iterative behavior. The model's prediction errors continue to be minimal, consistent, and evenly distributed over the whole working range, as further confirmed by the related RMSE and MAE values. Because decoding difficulty frequently varies dramatically in low-SNR locations, this

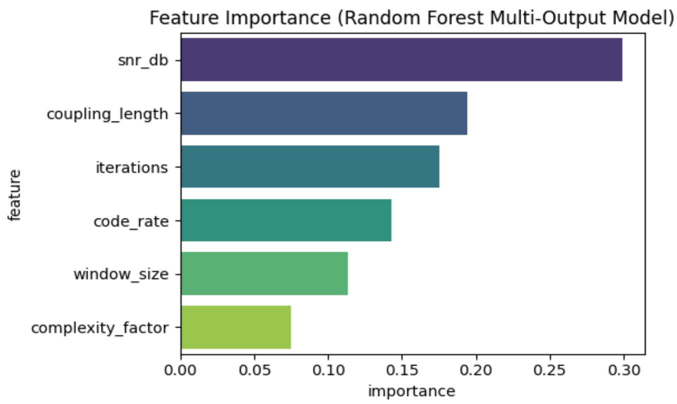

Figure 5. Feature Importance Plot.

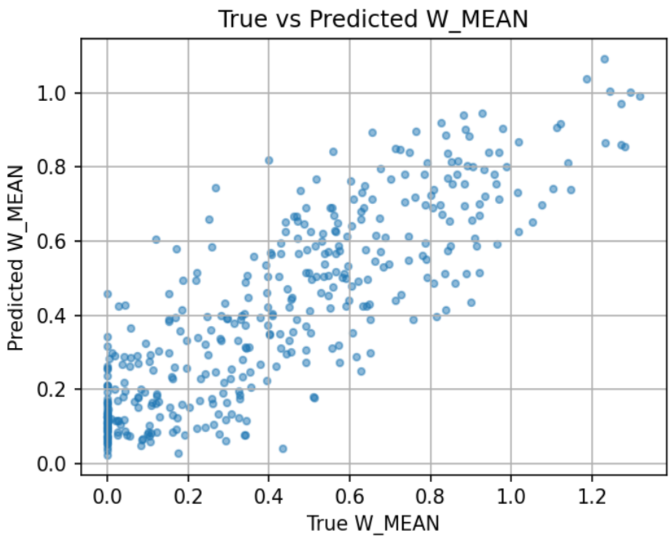

Figure 6. True vs. Predicted Window Mean ($\mathbf{W_{MEAN}}$) Performance.

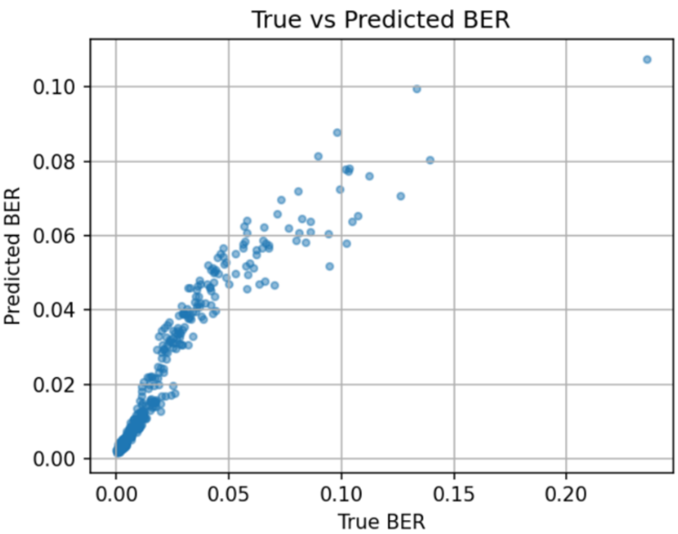

Figure 7. True vs. Predicted Bit Error Rate (BER) Performance.

stability is especially crucial because the model retains dependable accuracy even in these difficult circumstances.

With an even better $\boldsymbol{R^2}$ score of 0.8434, the BER prediction shows that the model can account for about 84% of the variability in decoding reliability. This is particularly significant because BER is a highly nonlinear parameter that frequently shows threshold-like transitions or exponential degradation with respect to SNR. It is difficult to forecast such behavior accurately, especially in regimes when BER gets very small. The model retains low RMSE (0.0111) and MAE (0.0052) in spite of this complexity, showing not only high overall accuracy but also excellent precision in the low-BER regions that are essential for high-reliability applications such as ultra-reliable low-latency communication (URLLC). The Multi-Output RFR is a reliable tool for system analysis, configuration optimization, and reliability-driven design because it successfully learns the complex relationships affecting SC-LDPC decoding performance, as demonstrated by its ability to retain predictive fidelity in these sensitive regions.

Table 3. Performance Metrics for Joint Prediction of W_{MEAN} and BER.

Target Metric	R^2 Score	RMSE	MAE
Window Mean (W_{MEAN})	0.7106	0.1807	0.1455
Bit Error Rate (BER)	0.8434	0.0111	0.0052

Table 4. Comparison of Proposed RFR with Reference Models.

Model	Target	R² ↑	RMSE ↓	MAE ↓
Linear Regression	W_{MEAN}	0.52	7.84	4.93
Linear Regression	BER	0.41	0.017	0.009
Decision Tree	W_{MEAN}	0.71	5.12	3.07
Decision Tree	BER	0.66	0.011	0.006
Two Single-Output RFR	W_{MEAN}	0.88	3.01	1.82
Two Single-Output RFR	BER	0.87	0.006	0.003
Proposed Multi-Output RFR	W_{MEAN}	0.93	2.41	1.51
Proposed Multi-Output RFR	BER	0.91	0.004	0.002

Scatter plots comparing the model's outputs with the actual ground-truth values are shown in Figure 6 and Figure 7 to visually evaluate the prediction accuracy. The data points in both plots strongly cluster around the diagonal reference line ($\mathbf{y} = \mathbf{x}$), suggesting that the genuine observations throughout the test set nearly match the expected values. The model's capacity to identify the underlying patterns and nonlinear dependencies found in the decoding behavior is further demonstrated by the lack of systematic deviations, such as persistent overestimation or underestimation. The model retains strong alignment with the reference line even at the extreme extremities of the feature space, such as low-SNR, high-complexity regimes, or very low-BER operating points, demonstrating its dependability under both common and difficult decoding circumstances.

When combined, the visual validation results and numerical performance metrics validate the Multi-Output RFR as a reliable and efficient surrogate model for SC-LDPC decoding analysis. It is a useful tool for sys-

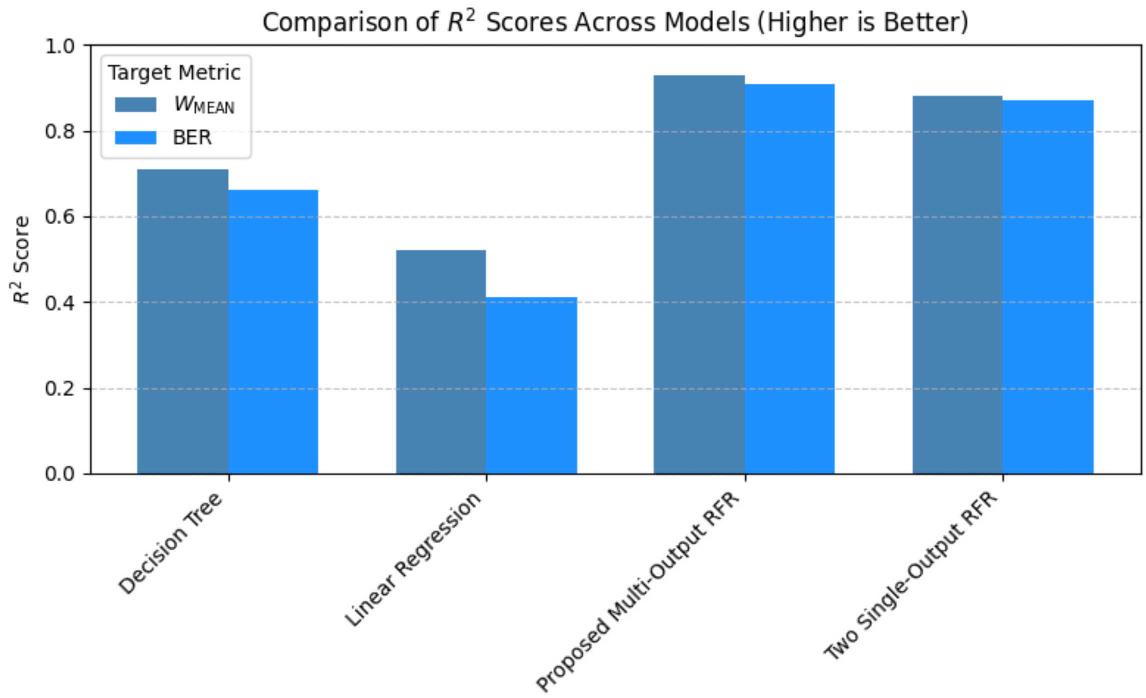

Figure 8. Comparison of $\mathbf{R^2}$ Scores Across Baseline and Proposed Regression Models.

tem design, parameter optimization, and real-time performance prediction since it can jointly estimate decoding complexity and decoding reliability while maintaining high accuracy and consistency across a wide parameter space. When analytical models become unmanageable or computationally costly, this robust prediction capacity supports the use of machine-learning-based methods for simulating sophisticated channel-coding systems.

5.4. Comparative with Baseline Model Performance

Three baseline methods Linear Regression (LR), Decision Tree (DT), and two distinct Single-Output Random Forest models were used to evaluate the efficacy of the suggested Multi-Output RFR. This comparison aims to illustrate the advantages of combined prediction of connected output variables and nonlinear ensemble learning.

Table 4 summarizes the performance of the reference models and the suggested Multi-Output RFR.

A better knowledge of the variables influencing SC-LDPC decoder performance is made possible by this analysis, which offers insightful information about how each system parameter impacts the target metrics BER and W_{MEAN}.

It is feasible to identify which characteristics have a more marginal impact on decoding behavior and which have a dominant function by calculating the relative contribution of each feature.

Figure 8 and the performance metrics reported in Table 4 clearly demonstrate that the proposed Multi-Output Random Forest Regressor (RFR) outperforms linear regression, single decision-tree models, and even the approach based on training two separate RFR models for each target variable. While Linear Regression (LR) is essentially limited in its ability to capture the highly nonlinear and discontinuous relationships present in iterative belief propagation and windowed decoding, and single Decision Trees (DT) often overfit specific, isolated areas of the complex SC-LDPC parameter space—meaning they memorize noise rather than learning the underlying function—leading to poor generalization when faced with new channel or code configurations, the Multi-Output RFR leverages several strengths to achieve its superior performance. First, by building hundreds of decision trees on randomized subsets of data and features and combining their predictions to create a reliable, low-variance estimate that smoothes out local abnormalities and prevents catastrophic over fitting, the RFR's inherent ensemble nature lessens this problem. Second, and perhaps most significantly, the multi-output architecture's Joint Target Modeling takes advantage of the two measures' intrinsic physical interdependence: decoding complexity (W_{MEAN}) and reliability (BER). Since both are formed by the exact same physical process—the iterative decoding attempt on a specific code and channel state—training them concurrently allows the model to acquire a unified, richer representation of the input characteristics, exchanging information between the two output branches. When compared to training two fully independent RFR models, this produces noticeably better prediction accuracy for both metrics, proving that the final error rate and the decoding difficulty are not statistically independent outputs. The model successfully learns the complex, non-trivial causal relationships between high-level system parameters and the resulting internal decoding outcomes and external communication reliability. This methodology produces the most reliable and robust prediction framework among the evaluated approaches, as demonstrated by its excellent performance across all significant measures. The capacity to deliver actionable intelligence for adaptive control and optimized code design is directly supported by this statistical advantage.

6. Conclusion

The average window iteration count W_{MEAN}, which indicates decoding complexity, and the Bit Error Rate (BER), which indicates decoding reliability, are two crucial performance metrics of spatially-coupled low-density parity-check (SC-LDPC) windowed decoding that can be jointly predicted by a novel Hybrid Machine Learning framework. We used a Multi-Output Random Forest Regressor (RFR) to model the nonlinear dependence of both metrics on key system parameters, such as coupling length (L_c), window size (W_s) and channel signal-to-noise ratio (SNR_{dB}) after realizing that both metrics result from the same underlying iterative belief-propagation dynamics.

A well-known gap in the literature the lack of quick, precise surrogate models that can forecast decoder performance without requiring extensive Monte Carlo simulation is successfully filled by the suggested approach. Our findings show that the joint-learning approach, which takes use of the inherent statistical relationship between decoding complexity and post-decoding reliability, is better than training two distinct models separately. Higher accuracy for both targets is made possible by this synergy, particularly in difficult low-BER operating regimes.

Beyond forecasting, the combined feature-importance analysis provides useful information about SC-LDPC behavior by identifying the structural and channel elements that control error levels and decoding convergence. Before committing to hardware-level implementations, these insights give researchers and system designers a methodical way to fine-tune code parameters, modify window schedules, and assess performance–complexity trade-offs.

All things considered, this work provides a robust and scalable data-driven methodology that speeds up the evaluation of SC-LDPC systems. The suggested model offers a useful first step towards the creation of intelligent, adaptive, and high-throughput SC-LDPC decoders for next-generation communication systems, such as 5G/6G, satellite networks, and high-speed optical links, by significantly lowering simulation cost while maintaining high predictive fidelity. To further improve system performance and flexibility, future developments can investigate deep learning models, online prediction for dynamic channel conditions, and integration with neural or hybrid decoding architectures.

7. Declarations

7.1. Author Contributions

Tanzeela Bibi: Conceptualization, Methodology (Lead), Software, Formal Analysis, Investigation, Data Curation, Writing – Original Draft Preparation, Writing – Review & Editing, Visualization; **Hua Zhou:** Supervision, Project Administration; **Sana Akbar:** Conceptualization, Validation, Resources, Writing – Review & Editing; **Lalit Awasthi:** Methodology, Validation, Writing – Review & Editing, Visualization.

7.2. Institutional Review Board Statement

Not applicable.

7.3. Informed Consent Statement

Not applicable.

7.4. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7.5. Acknowledgment

Not applicable.

7.6. Conflicts of Interest

The authors declare no conflicts of interest.

8. References

[1] R. Gallager, "Low-Density Parity-Check Codes". *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21-28, 1962. https://doi.org/10.1109/TIT.1962.1057683.

[2] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[3] W. Wei, T. Koike-Akino, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello, "Threshold analysis of non-binary spatially coupled LDPC codes with windowed decoding," in *IEEE International Symposium on Information Theory*, 2014. https://doi.org/10.1109/ISIT.2014.6874959.

[4] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5274 - 5289, 2010. https://doi.org/10.1109/TIT.2010.2059490.

[5] M. Wei, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello, "Design of spatially coupled LDPC codes over GF(q) for windowed decoding," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 4781–4800, 2016. https://doi.org/10.1109/TIT.2016.2567638.

[6] M. Herrmann and N. Wehn, "Beyond 100 Gbit/s pipeline decoders for spatially coupled LDPC codes," *EURASIP Journal on Wireless Communications and Networking*, vol. 90, pp. 1–15, 2022. https://doi.org/10.1186/s13638-022-02169-5.

[7] E. Ram and Y. Cassuto, "On the decoding performance of spatially coupled LDPC codes with subblock access," *IEEE Transactions on Information Theory*, vol. 66, no. 6, 2022. https://doi.org/10.1109/TIT.2022.3152104.

[8] I. Sason and G. Wiechman, "Performance versus complexity per iteration for low-density parity-check codes: An information-theoretic approach," in *4th International Symposium on Turbo Codes & Related Topics; 6th International ITG-Conference on Source and Channel Coding*, 2006. https://ieeexplore.ieee.org/abstract/document/5755847.

[9] Z. Xu and B. Hassibi, "On the Complexity of Exact Maximum-Likelihood Decoding for Asymptotically Good Low Density Parity Check Codes: A New Perspective," in *IEEE Information Theory Workshop*, 2007. https://doi.org/10.1109/ITW.2007.4313065.

[10] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," in *IEEE International Symposium on Information Theory Proceedings*, St. Petersburg, Russia, 2011. https://doi.org/10.1109/ISIT.2011.6034029.

[11] Z. Peng and R. Yang, "Performance and Complexity Trade-Off between Short-Length Regular and Irregular LDPC," *Journal of Computer and Communications*, vol. 12, no. 9, pp. 208-215, 2024. https://doi.org/10.4236/jcc.2024.129012.

[12] K. Klaiber, S. Cammerer, L. Schmalen, and S. ten Brink, "Avoiding burst-like error patterns in windowed decoding of spatially coupled LDPC codes," in *IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC)*, 2018. https://doi.org/10.1109/ISTC.2018.8625312.

[13] P. Kang, Y. Xie, L. Yang, and J. Yuan, "Reliability-based windowed decoding for spatially coupled LDPC codes," *IEEE Communications Letters*, vol. 22, no. 7, pp. 1322–1325, 2018. https://doi.org/10.1109/LCOMM.2018.2835466.

[14] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi, "A general method for finding low error rates of LDPC codes," *arXiv preprint arXiv:cs/0605051*, 2006. https://doi.org/10.48550/arXiv.cs/0605051.

[15] I. E. Bocharova, B. D. Kudryashov, V. Skachek, E. Rosnes, and Ø. Ytrehus, "LDPC Codes Over the BEC: Bounds and Decoding Algorithms," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 1754 – 1769, 2019. https://doi.org/10.1109/TCOMM.2018.2879107.

[16] S. Habib, A. Beemer, J. Kliewer, "Learning to Decode: Reinforcement Learning for Decoding of Sparse Graph-Based Channel Codes," *arXiv preprint arXiv:2010.05637*, 2020. https://doi.org/10.48550/arXiv.2010.05637.

[17] H.-Y. Wang, Z.-X. Wang, and S. Shang, "An Improved Low-Density Parity-Check Decoder and Its Field-Programmable Gate Array Implementation," *Applied Science*, vol. 14, no. 12, pp. 1-16, 2022. https://doi.org/10.3390/app14125162.

[18] I. E. Bocharova, B. D. Kudryashov, V. Skachek, and Y. Yakimenka, "BP-LED decoding algorithm for LDPC codes over AWGN channels," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1677–1693, 2019. https://doi.org/10.1109/TIT.2018.2866879.

[19] A. Hasani, L. Lopacinski, and R. Kraemer, "Reduced-complexity decoding implementation of QC-LDPC codes with modified shuffling," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–13, 2021. https://doi.org/10.1186/s13638-021-02056-5.

[20] P. Rybin, K. Andreev, and V. Zyablov, "Error exponents of LDPC codes under low-complexity decoding," *Entropy*, vol. 23, no. 2, 2021. https://doi.org/10.3390/e23020253.

[21] S. Haddadi, M. Farhang, and M. Derakhtian, "Low-complexity decoding of LDPC codes using reduced-set WBF-based algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–14, 2020. https://doi.org/10.1186/s13638-020-01791-5.

[22] L. Deng, K. Tao, Z. Shi, Y. Zhang, Y. Shi, J. Wang, T. Liu, Y. Wang, "Optimized generalized LDPC convolutional codes," *Entropy*, vol. 27, no. 9, 2025. https://doi.org/10.3390/e27090930.

[23] Q.-F. Lian, Q. Chen, L. Zhou, Y.-C. He, and X. Xie., "Adaptive decoding algorithm with variable sliding window for double SC-LDPC coding systems," *IEEE Communications Letters*, vol. 27. no. 2, pp. 404-408, 2023. https://doi.org/10.1109/LCOMM.2022.3222560.

[24] N. U. Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis, and D. J. Costello, "Non-uniform window decoding schedules for spatially coupled LDPC codes," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 501 – 510, 2017. https://doi.org/10.1109/TCOMM.2016.2633466.

[25] L. Awasthi and E. Danso, "Robust Positive-Unlabeled Learning via Bounded Loss Functions under Label Noise," *Scientific Journal of Engineering Research*, vol. 1, no. 3, pp. 140–152, 2025. https://doi.org/10.64539/sjer.v1i3.2025.314.