BLIND PARAMETER ESTIMATION OF 5G CHANNEL CODING SCHEMES

M.Tech. Thesis

By

M. SAI VAMSHI

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2025

BLIND PARAMETER ESTIMATION OF 5G CHANNEL CODING SCHEMES

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree
of
Master of Technology

by

M. SAI VAMSHI

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2025

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **BLIND PARAMETER ESTIMATION OF 5G CHANNEL CODING SCHEMES** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DEPARTMENT OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from May 2024 to May 2025 under the supervision of **Dr. Swaminathan Ramabadran, Associate Professor at Indian Institute of Technology Indore**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

26/5/25

Signature of the student with date
M. SAI VAMSHI

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

26/05/2025

Signature of the Supervisor of
M.Tech. thesis (with date)
Dr. SWAMINATHAN RAMABARAN

M. SAI VAMSHI has successfully given his/her M.Tech. Oral Examination held on **07/05/2025**

Saptarshi Ghosh

Signature(s) of Supervisor(s) of M.Tech. thesis
Date: 26/05/2025

Convener, DPGC
Date: 27-05-2025

Signature of PSPC Member #1
Date:

Signature of PSPC Member #1
Date:

Acknowledgments

I would also like to extend my sincere appreciation to my supervisor, Dr. Swaminathan Ramabadran, whose guidance and support have played a significant role in enriching my research experiences.. His exemplary dedication to research has been a constant source of inspiration, motivating me to pursue excellence in my academic endeavors. I feel truly privileged to have worked under his guidance.

My heartfelt appreciation goes to my lab mates—Mr. Nayim Ahmed, Mr. Manoj Khokare, Mr. Pranshu Singh, Mr. Prashant Sharma, and Ms. Vaishali Rohilla. Their camaraderie, support, and teamwork that they provided in the lab have contributed to making my research experience fun and rewarding.

I sincerely appreciate the assistance given by IIT Indore and the Ministry of Education, Government of India, to complete my M.Tech. through better laboratory facilities and the Teaching Assistantship scholarship.

I am deeply indebted to my parents and my brother, whose uncritical support, motivation, and ideals have been the pillars of my academic pursuits. Their confidence education has constantly spurred me forward, and I shall ever remain indebted to their love and guidance.

Last but certainly not least, I am truly fortunate to have the unwavering support of my dear friends—Sreekar, Teja, and my school batch of 2014—who have stood by me through the thick and thins, their presence and encouragement have played a pivotal role in my personal growth and well-being.

Abstract

This project solves the issue of blind parameter estimation and type classification of LDPC and Polar codes, two central schemes of forward error correction in modern wireless communication standards like 5G. The work introduces and builds machine learning-based paradigms to correctly detect code type and code parameters from the received signals, without any initial knowledge of the encoder setup. Large-scale datasets were created for both LDPC and Polar codes with different SNR levels and codeword lengths, and several classification models-CNN, DRN, and hybrid CNN-SVM-were learned and tested. Classification accuracy of the learned models was high, and identification of Polar codes was more than 97%, while they performed well over codeword lengths and SNRs. Experimental verification was performed with a USRP-based Software Defined Radio platform, and it was shown that the methods presented have solid accuracy and reliability in actual hardware environments, closely replicating MATLAB simulation results. Comparative evaluation against current literature verifies that Polar codes have better robustness and classification accuracy at low SNRs, and LDPC codes have high accuracy at larger block lengths and higher SNRs. The results demonstrate the efficacy of deep learning methods for blind code parameter estimation and justify the real-world application of such methodologies in adaptive, intelligent wireless communication systems.

Contents

List of Figures

Chapter 1

Introduction

1.1 3GPP and 5G Standard

The 3rd generation partnership project, or 3GPP, is a global partnership of organizations that establishes worldwide standards for mobile telephony, including the current 5G new radio (NR) network[1]. 3GPP standards are significant since they facilitate interoperability, dependability, and scalability between different vendors and operators, allowing continuous connectivity across the globe. In 5G, 3GPP brought important improvements in channel coding through the definition of Low Density Parity-Check(LDPC) codes for data channels and Polar codes for control channels. For LDPC codes, 3GPP standardizes two base graphs (Base Graph_1 and Base Graph_2) in its standards (e.g., 3GPP TS 38.212), which are used as templates in constructing the parity-check matrices based on transport block size and target code rate. These fundamental graphs are extended to have the targeted codeword length and form, providing flexibility as well as hardware efficiency. Polar codes, by contrast, were chosen for their ability to achieve capacity and scalability and are mostly applied for encoding control data in 5G systems. Polar codes are designed to take advantage of channel polarization and thus determine the reliable and unreliable bits, minimizing performance particularly for short and moderate block lengths. The implementation of these coding schemes using blind parameter estimation are core to the proposed work since they offer the framework for classifying coded signals using deep learning algorithms for next generation communication system applications. Through strict adherence to 3GPP standards, the machine learning models developed for blind parameter estimation and classification of LDPC and Polar codes are both applicable to future wireless communication systems.

1.2 Forward Error Correcting Codes (FEC)

Forward error correcting (FEC) codes are the backbone of contemporary wireless communications systems, providing guaranteed data transmission with the detection and correction of errors caused by propagation through error-prone channels. In 5G networks, FEC plays a particularly significant role because of the need for high data rate, low latency, and resilience in varied and challenging environments.The different FEC methods, LDPC (Low-Density Parity-Check) codes and Polar codes have been the two major coding schemes chosen by the 3GPP for 5G New Radio (NR) standards[2]. Data channels make use of LDPC codes, taking advantage of their sparse parity-check matrices and low-complexity encoding/decoding architectures to get close to the theoretical Shannon limit and hence suitable for high-throughput scenarios. Polar codes are applied for control channels and are well-known for being the first family of codes that have been shown to meet channel capacity for binary-input discrete memoryless channels with high error correction capability at relatively low decoding complexity[2].

In the proposed work, the importance of these FEC codes is brought to light through the work of developing and testing blind parameter estimation and classification architectures for LDPC and polar codes, as evident from the given confusion matrices. The outcome is of high accuracy in the discrimination of code types and codeword sizes, proving the potency of machine learning-based methods for blind identification in real-world applications. This reflects conclusions from recent studies, which highlight the complementary advantages of LDPC and Polar codes in 5G: LDPC codes are best suited to high-rate data transmission because they have high-fault-tolerance and low-hardware-complexity in error correction, whereas polar codes offer low-latency, high-dependability operation for control signaling[2]. The results of the project highlight the significance of forward error correction in ensuring the reliability and efficiency next-generation wireless networks call for, and demonstrate how sophisticated coding techniques, when paired with smart signal processing, can address the varied 5G and beyond demands.

1.2.1 Types of Forward Error Correcting codes

In the context of the proposed work, which is blind parameter estimation and classification of LDPC and Polar codes for 5G coding schemes, the most relevant types of forward error correcting (FEC) codes are block codes and, more particularly, the new subclasses of block codes: LDPC codes and polar codes.

- LDPC codes: LDPC codes are block linear codes with a sparse parity-check matrix. They are very efficient, tending to the Shannon limit of error correction, and are used extensively in 5G in data channels because they have excellent performance in correcting random errors in noisy wireless channels. LDPC codes are decoded employing algorithms that take advantage of their sparsity and are especially suited for long codeword lengths and high-throughput scenarios.

- Polar codes: Polar codes are a more recent family of block codes that employ the method of channel polarization to attain capacity for binary-input discrete memoryless channels. Polar codes are the first codes shown to reach channel capacity using low-complexity encoding and decoding, and have applications in 5G for control channels. Polar codes are especially efficient for short to moderate block lengths and can be decoded efficiently by successive cancellation (SC), successive cancellation list (SCL), or Cyclic Redundancy Check(CRC)-aided SCL algorithms.

- Other FEC Codes: Although this proposed work primarily focuses on LDPC and Polar codes, it is important to acknowledge other well-established Forward Error Correction (FEC) codes such as Hamming codes, BCH codes, Reed-Solomon codes, and Turbo codes. These classical codes have played a foundational role in coding theory and are still widely employed in various communication and data storage systems. However, LDPC and Polar codes are emphasized in this work due to their superior performance and their adoption in high-end 5G communication standards

1.3 Low-Density Parity-Check Codes (LDPC)

LDPC codes are advanced linear error-correcting codes that play a critical role in ensuring reliable data transmission over noisy communication channels. Originally introduced by Robert Gallager in the 1960s, LDPC codes are defined by a sparse parity-check matrix, meaning that most entries in the matrix are zeros, with only a few ones scattered throughout. This sparsity allows for efficient encoding and, more importantly, highly effective iterative decoding algorithms, such as belief propagation, which enable LDPC codes to approach the theoretical capacity limits of communication channels with relatively low computational complexity.

LDPC codes are particularly well-suited for large block sizes and are widely used in modern applications like satellite communications, digital television (DVB-S2 standard), high-speed Ethernet, and wireless Fidelity(Wi-Fi) standards like Wi-Fi 6. The double diagonal (or dual diagonal) structure in LDPC codes refers to a specific arrangement within the parity-check matrix, where the parity portion of the matrix contains two diagonals of nonzero elements, typically ones, offset from each other [3]. This structure plays a crucial role in simplifying the encoding process, as it allows for efficient computation of parity bits without the need for complex matrix inversion or iterative algorithms.

One of the main advantages of LDPC codes is their excellent error-correction performance, which can surpass traditional codes like turbo codes, especially at higher code rates. They also offer a high degree of parallelism in decoding, making them suitable for hardware implementations that require fast processing. However, LDPC codes can have higher encoding complexity compared to some other codes, and their design often requires careful optimization to avoid short cycles in the parity-check matrix, which can degrade performance.

Overall, LDPC codes have revolutionized error correction in digital communications, offering a powerful combination of mathematical elegance, practical efficiency, and adaptability to a wide range of modern communication systems.

1.4 Polar Codes

Polar codes, proposed by Arikan in 2009, are a revolutionary family of linear block codes that meet the symmetric capacity of binary-input discrete memoryless channels with low-complexity successive cancellation (SC) decoding. Polar code construction essentially depends on repeated Kronecker product of a simple kernel matrix, usually represented like $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. For a codeword length of $N = 2^n$ the generator matrix is constructed as $F^{\otimes n}$, where and $\otimes$ denotes the Kronecker product[4, 5]. The recursive nature of this construction is not only beautiful but also allows for efficient hardware and software implementations, since the encoding and decoding procedures can be directly translated into the steps of the Kronecker product.

At the core of designing and implementing polar codes lies the concept of the reliability sequence. As the Kronecker product is applied during the encoding process, channel polarization occurs, transforming the initial communication channel into a set of virtual bit-channels with varying levels of reliability. The reliability sequence is a precomputed, sorted list that ranks these bit-channels from least to most reliable. During the construction of Polar codes, information bits are assigned to the most reliable bit-channels, while the less reliable ones are designated as frozen bits, typically fixed to zero. The selection of the reliability sequence plays a critical role in determining the error-correcting performance of the code. It is the combination of this reliability-based bit allocation and the recursive Kronecker product structure that enables Polar codes to achieve channel capacity. This makes them particularly suitable for applications such as 5G control channels, where both high performance and implementation efficiency are essential.

1.5 Deep Learning

Deep learning is a specialized branch of machine learning that uses artificial neural networks with multiple layers-often called deep neural networks-to enable computers to learn complex patterns directly from large amounts of data. Inspired by the structure and function of the human brain, these networks consist of interconnected nodes (neurons) organized in layers: an input layer, several hidden layers, and an output layer.

Each layer extracts increasingly abstract features from the input, allowing the system to handle tasks that are difficult or impossible for traditional algorithms, such as image recognition, natural language processing, and speech understanding[6, 7, 8].

Unlike traditional machine learning, which often relies on manual feature engineering, deep learning models can automatically discover relevant features from raw, unstructured data, including images, audio, and text. This ability has led to state-of-the-art results in various fields. For example, deep learning powers digital assistants, self-driving cars, fraud detection systems, medical image analysis, and recommendation engines for streaming and e-commerce platforms. Training deep learning models typically requires vast datasets and significant computational power, but once trained, these models can make highly accurate predictions and automate complex tasks with minimal human intervention.

The architecture of deep learning networks can vary depending on the application. convolutional neural networks (CNNs)[9] excel at image and video analysis, while recurrent neural networks (RNNs) and transformers are used for sequential data like language and time series. The learning process involves forward propagation-where data moves through the network to generate predictions-and backpropagation, where the network adjusts its internal parameters to minimize errors.

Overall, deep learning is a foundational technology driving the current wave of artificial intelligence, enabling machines to perform tasks that previously required human intelligence and opening new possibilities across science, industry, and daily life.

1.5.1 Neural Network

A neural network is a computational model inspired by the human brain's network of neurons, designed to recognize patterns and make decisions based on data. It consists of layers of interconnected artificial neurons: an input layer that receives raw data, one or more hidden layers that process and transform this data, and an output layer that produces the final result or prediction. Each neuron receives inputs, multiplies them by associated weights, adds a bias, and then applies a nonlinear activation function to determine its output. These weighted connections between neurons represent the strength and influence of signals, and during training, the network adjusts these weights and biases to minimize the difference between its predictions and actual outcomes through a process called backpropagation combined with optimization algorithms like gradient descent.

Neural networks excel at modeling complex, nonlinear relationships in data, enabling them to perform tasks such as image and speech recognition, natural language processing, and autonomous decision-making. The architecture can vary from simple networks with a single hidden layer to deep neural networks that contain many hidden layers, allowing them to learn hierarchical features at different levels of abstraction. Training deep networks typically requires large datasets and significant computational resources, but once trained, they can generalize well to new, unseen data.

The flexibility of neural networks comes from their ability to approximate almost any function given enough neurons and training, making them a foundational technology in modern artificial intelligence and machine learning. Their design is inspired by biological neurons but optimized for computational efficiency, enabling machines to learn from experience and improve performance with minimal human intervention.

1.5.2 Deep Neural Network

A deep neural network (DNN) is an advanced form of artificial neural network characterized by having multiple layers between the input and output, enabling it to learn and model complex, non-linear relationships in data. The term "deep" refers to the presence of several hidden layers-often far more than the two or three found in traditional neural networks-which allow the network to extract increasingly abstract features from raw input as data passes through each layer.

A typical DNN is structured with an input layer that receives the raw data, a series of hidden layers that process and transform the data, and an output layer that generates the final prediction or classification. Each neuron in these layers computes a weighted sum of its inputs, applies a non-linear activation function, and passes the result to the next layer. During training, the network uses a process called backpropagation to adjust the weights and biases in response to errors, gradually improving its performance.

DNNs are the backbone of deep learning and have enabled breakthroughs in fields such as computer vision, natural language processing, speech recognition, and autonomous systems. For example, CNNs, a popular type of DNN, excel at image and video analysis by using convolutional and pooling layers to detect spatial hierarchies in data. Recurrent neural networks (RNNs) and their variants, like long short-term memory (LSTM) networks, are well-suited for sequential data such as language or time series.

The depth of a DNN allows it to learn compositional representations: lower layers might detect simple features (like edges in images), while higher layers combine these into more complex patterns (like shapes or objects). This hierarchical learning is what makes DNNs particularly powerful for solving tasks that are too complex for shallow networks or traditional algorithms.

In summary, deep neural networks are sophisticated machine learning models that mimic the way the human brain processes information, enabling machines to learn from large datasets and solve complex problems with high accuracy and minimal human intervention.

1.6 Activation and Loss Function

Activation and loss functions are fundamental components of neural networks, each serving a distinct yet crucial role in the learning process. Activation functions introduce non-linearity into the network, enabling it to model complex patterns in data. Loss functions, on the other hand, quantify the discrepancy between the network's predictions and the actual target values, guiding the optimization process during training.

Activation functions determine whether a neuron in a neural network should be activated or not, based on the input it receives. They introduce non-linearity into the model, enabling the network to learn complex patterns that go beyond simple linear relationships. Without activation functions, a neural network would essentially behave like a linear regression model, regardless of its depth, and would be unable to capture the intricacies present in real-world data.

Types of Activation Functions:

- Linear (Identity) Function: Outputs the input directly. Used mainly in regression tasks, but doesn't introduce non-linearity, limiting the network's learning capacity.

- Binary Step Function: Activates a neuron only if the input surpasses a certain threshold. Suitable for simple binary classification but not for multi-class problems or complex tasks.

- Sigmoid Function: Maps input values to a range between 0 and 1, making it useful for binary classification and probability estimation. However, it can suffer from vanishing gradients.
- Tanh (Hyperbolic Tangent) Function: Outputs values between -1 and 1, is zero-centered, and is often used in tasks like speech and language processing.
- (Rectified Linear Unit): Outputs zero for negative inputs and the input itself for positive values. It is widely used in deep networks due to its simplicity and effectiveness, and it helps mitigate the vanishing gradient problem.Choosing the right activation function depends on the task at hand and the specific layer within the network. For example, ReLU is common in hidden layers, sigmoid in binary classification outputs, and softmax in multi-class classification outputs.

Loss Function

A loss function, also known as a cost or objective function, quantifies the difference between the predicted outputs of a neural network and the actual target values. This measurement guides the optimization process: during training, the network adjusts its parameters to minimize the loss, thereby improving its predictions.

Types of Loss Functions:

For Regression:

- Mean Squared Error (MSE): Calculates the average squared difference between predicted and actual values. Commonly used for regression problems.
- Mean Absolute Error (MAE): Measures the average absolute difference, making it more robust to outliers than MSE.
- Huber Loss: Combines MSE and MAE, being quadratic for small errors and linear for large ones, offering robustness to outliers.

For Classification:

- Binary Cross-Entropy (Log Loss): Used for binary classification, measuring the difference between predicted probabilities and actual binary labels.

- Categorical Cross-Entropy: Used for multi-class classification, evaluating the difference between predicted probability distributions and actual class labels.
- Hinge Loss: Mainly used for training support vector machines (SVMs), penalizing predictions that are not only wrong but also not confident enough. The choice of loss function is directly tied to the type of problem (regression or classification) and the network's output structure. For example, MSE is standard for regression, while cross-entropy is preferred for classification tasks.

1.6.1 Softmax Function

The softmax function is a mathematical function commonly used in the output layer of neural networks for multiclass classification problems. Its main purpose is to convert a vector of raw scores (also known as logits) into a probability distribution, where each value is between 0 and 1 and the sum of all values is exactly 1. This makes the outputs interpretable as probabilities for each class.

The softmax function is defined as

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1.1}$$

where $z = [z_1, z_2, \ldots, z_K]$ is the input vector and $\sigma(z)_i$ is the probability assigned to class i. This means each output is the exponential of the input value divided by the sum of exponentials of all input values. The effect is that higher input values get mapped to higher probabilities, and the largest input value will dominate the output, often resulting in a high probability for one class and low probabilities for the others.

Key characteristics of softmax

Produces a probability distribution over classes, useful for mutually exclusive class assignments. Amplifies differences in logits, making the largest value stand out as the most probable class. Ensures outputs are easy to interpret and can be directly used as confidence scores for each class. Softmax is used in the final layer of a neural network for multiclass classification tasks. The softmax function transforms raw neural network outputs into a normalized probability distribution, making it essential for multiclass classification problems in deep learning.

1.7 Literature Survey

Blind estimation of parameters and error correcting codes classification are of growing importance in digital communication systems nowadays, especially in non-cooperative, adaptive, and spectrum surveillance systems. New algorithms have been introduced in recent research to blindly estimate code parameters for product codes like Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH) codes under noisy channel conditions. These researches have demonstrated that estimation quality increases with decreasing modulation order and code dimension, and have solved the synchronization problem by estimating bit position adjustment parameters, which are essential in practical applications such as signals intelligence and adaptive modulation and coding in satellite and aeronautical communications [10, 11, 12].

For LDPC codes, blind recognition and estimation of code dimension and codeword length have been achieved even in the presence of channel errors. New algorithms have been presented that do not need a predefined candidate set, which are applicable in non-cooperative situations. These strategies have illustrated that the accuracy of estimation is greater for lower code rates, shorter codeword lengths, and smaller modulation orders, and have been certified by large-scale simulations. These types of methodologies are immediately applicable to actual LDPC code identification and are evidenced in the high classification accuracy obtained in this project [13, 14, 3].

Blind reconstruction algorithms for BCH encoders have been proposed as well, which can estimate codeword length, code dimension, and generator polynomial even in the presence of error in the channel. The algorithms perform better than current approaches, especially with decreasing codeword length, and are tolerant to a wide range of channel conditions. Likewise, for turbo convolutional codes, new algorithms have been proposed for blind parameter estimation and encoder reconstruction under noisy and non-synchronized environments, compensating for synchronization defects and showing higher accuracy with decreasing modulation orders and constraint lengths[2].

Xia's LDPC paper offers a better decoding method for LDPC codes based on linear programming (LP) techniques with emphasis on minimum pseudo-weight and pseudo-codeword analysis. The research shows that the pseudo-weight lower bound is tight for some LDPC codes only when the pseudo-codeword is an actual multiple of a codeword, which means the LP decoding can get close to maximum-likelihood decoding performance as the signal-to-noise ratio grows. The paper also proves that, for certain LDPC codes, all minimum pseudo-codewords are real multiples of minimum codewords, which reflects the asymptotic optimality of LP decoding for such codes. The findings present significant theoretical understanding of the decoding capabilities and empirical performance of LDPC codes under LP-based decoding schemes[15].

Recent research has also considered the fine-grained identification of error-correcting codes as an essential step within code parameter estimation and emphasized its importance in terms of enhancing the adaptability and intelligence of receiver systems in non-cooperative scenarios. Statistical tests, including the Kolmogorov–Smirnov test, have also improved the blind estimation of interleaver parameters, extending the use of these methods towards a broader range of coding and interleaving schemes.

Taken as a whole, these studies have determined that blind parameter estimation performance is highly interdependent with code rate, codeword length, modulation order, and channel conditions. Combining these techniques into adaptive and reconfigurable receiver systems is critical to the facilitation of future wireless communication networks. Leveraging these results, my project deploys and verifies machine learning-based techniques for blind estimation and classification of LDPC and Polar codes in simulation as well as using Universal Serial Radio Peripheral(USRP) based hardware. The high classification rates obtained in this work align with observations made in the literature, again indicative of the real-world applicability and significance of blind parameter estimation in next-generation communication systems.

In short, the proposed work is based on and builds on this literature by developing and experimentally verifying machine learning-based blind parameter estimation and classification of LDPC and polar codes. The outcomes-regexplicated by high classification accuracy in simulated and USRP-based real-world testbeds-are in close agreement with the conclusions of these pioneering works. The integration of cutting-edge coding theory, resilient estimation algorithms, and state-of-the-art machine learning brings a unifying framework for intelligent, adaptive, and robust communication in the presence of channel uncertainty and real-world constraints.

1.8 Motivations and contributions

As we dive into the world of wireless communications, it's clear that our approaches to blind estimation techniques for channel codes has some scope for improvement. Many of these techniques were originally designed with 3G and 4G standards in mind, but with 5G on the scene, things have changed drastically. We're dealing with new coding schemes like LDPC and polar codes, and there's a growing demand for systems that offer higher reliability, lower latency, and the ability to adapt to ever-changing environments.

It's surprising to find that, despite the crucial role LDPC and polar codes have in the 5G framework, there isn't a wealth of research focused on the blind estimation of these codes yet. This gap is especially concerning as blind estimation techniques become more vital in our modern communication systems, which are often flexible and non-cooperative. Without established methods for blind estimation of these codes within 5G, we risk limiting the potential for creating robust receivers and advancing communication reliability, efficiency, and real-time adaptability.

Filling this gap is crucial for the future of communication systems, especially as our networks grow more diverse and the environments they operate in become more unpredictable. Through the development and analysis of new blind estimation techniques tailored for LDPC and polar codes in 5G, this research aims to pave the way for significant enhancements in system performance and resilience. In this work, I developed a solid understanding of LDPC and polar code parameters, along with their encoding methods as laid out in the 5G standard. I focused on generating and curating datasets specifically designed for training machine learning models, aimed at classifying and estimating LDPC and polar codes in real-world conditions.

I successfully implemented polar codes according to the 5G specifications using USRP-based software defined radio (SDR) platforms, which allowed me to validate the proposed methods in a practical setting. By integrating machine learning techniques, I achieved impressive accuracy in distinguishing between polar and LDPC codes, with the experimental results often matching or even exceeding theoretical expectations.

Additionally, I gained hands-on experience simulating polar codes on SDR platforms and effectively incorporating them into machine learning workflows. This work has really advanced the practical application of blind code estimation in modern communication systems, and I'm excited about the potential it holds for improving communication technology.

In this work, I developed a solid understanding of LDPC and polar code parameters, along with their encoding methods as laid out in the 5G standard. I focused on generating and curating datasets specifically designed for training machine learning models, aimed at classifying and estimating LDPC and polar codes in real-world conditions. I successfully implemented polar codes according to the 5G specifications using USRP-based SDR platforms, which allowed me to validate the proposed methods in a practical setting. By integrating machine learning techniques, I achieved impressive accuracy in distinguishing between polar and LDPC codes, with the experimental results often matching or even exceeding theoretical expectations.

Additionally, I gained hands-on experience simulating polar codes on SDR platforms and effectively incorporating them into machine learning workflows. This work has really advanced the practical application of blind code estimation in modern communication systems, and I'm excited about the potential it holds for improving communication technology.

Chapter 2

Blind identification of LDPC and Polar codes

2.1 Introduction

Blind identification of Low-Density Parity-Check (LDPC) codes is increasingly vital in modern wireless communications, especially with the implementation of LDPC codes as the main channel coding mechanism for data channels in 5G new radio (NR) systems. Unlike traditional settings, where both the transmitter and receiver have knowledge of the code parameters, blind identification aims to extract essential code characteristics—such as codeword length, code rate, and the structure of the parity-check matrix—directly from the received signal without prior information. This capability is particularly important in non-cooperative or adaptive communication scenarios where flexibility and robustness are critical.

A key characteristic of LDPC codes in 5G is their design based on base graphs and expansion. The 3GPP TS 38.212 standard outlines two base graphs, each optimized for different payload sizes and channel conditions. These base graphs act as templates, which can be expanded to create the complete parity-check matrix for specific code blocks. The structure of the base graph not only supports versatile code design and efficient hardware implementation but also features specific matrix patterns that are useful for identification. Importantly, LDPC codes in 5G utilize a double diagonal (or dual diagonal) matrix structure in their parity sections. This arrangement, where the parity submatrix has two diagonals filled with ones, allows for highly efficient encoding and decoding. It minimizes computational complexity and memory needs, making LDPC codes especially appealing for applications requiring high throughput and low latency.

From the perspective of blind identification, these structural traits are beneficial. The regularity and sparsity of the base graph, along with the identifiable double diagonal structure in the parity matrix, create recognizable patterns within the received data. For example, the quasi-cyclic nature of 5G LDPC codes—where the parity-check matrix combines cyclically shifted identity and zero matrices—means that certain statistical or algebraic relationships can be discerned even without explicit signaling of code parameters. This opens the door for developing algorithms that can propose potential base graph and expansion parameters, reconstruct candidate parity-check matrices, and evaluate their validity against the received signal through decoding.

Recent research has focused on developing efficient blind identification and detection algorithms for polar codes that meet the stringent latency and reliability requirements of 5G systems. Traditional methods often rely on variants of successive cancellation (SC) decoding, but these can be computationally intensive for blind search across multiple parameter hypotheses. To address this, belief propagation (BP)-based blind detection methods have been proposed.

2.2 Dataset Generation

2.2.1 Encoding of LDPC and Polar codes

Defining Code Parameters

- LDPC codes: Select code parameters such as the length of the codeword (n), the length of the message (k), the rate of the code ($R = k/n$), the specific base graph and the expansion factor according to the 5G NR standard. The double diagonal structure in the parity-check matrix (arising from the base graph) should be considered for efficient encoding and decoding.

- Polar codes: Choose the block length (n, typically a power of 2), the number of bits of information (k), the code rate, and determine the frozen bit positions based on the reliability of the channel or the standard specifications.

- Generating Random Data: Create random binary message vectors of length K. These represent the information bits to be transmitted.

- LDPC codes: Encode each message using the LDPC encoder, producing codewords of length N. The encoding process utilizes the parity-check matrix defined by the chosen base graph and expansion factor.

- Polar codes: Encode each message using the polar encoder, assigning information bits and frozen bits to their respective positions and applying the recursive generator matrix.

- Modulation and Noise Addition: Map the encoded bits to BPSK symbols (0 $\rightarrow$ +1, 1 $\rightarrow$ -1). Pass the modulated symbols through an additive white Gaussian noise (AWGN) channel to simulate realistic transmission conditions at various signal-to-noise ratios (SNRs).

- Generating Data for Different Code Lengths: Repeat the above steps for multiple codeword lengths, code rates, and SNRs for both LDPC and polar codes. This ensures the dataset covers a wide range of scenarios relevant for training or evaluating blind identification and decoding algorithms.

- Saving the Data: Organize the generated data-including original messages, encoded codewords, modulated signals, and noisy received signals-in a structured format.

Save the data set as CSV files, with clear labeling to distinguish between LDPC and polar code samples, code parameters, and SNR conditions.

2.2.2 Training and Classification

In the proposed work, I leverage sophisticated machine learning models to meticulously automate the process of distinguishing among various types of coded signals. During the training phase, these models are introduced to a carefully curated set of labeled signal data, which provides them with the opportunity to uncover the intricate features and patterns associated with each distinct class. Through this learning journey, the models rigorously optimize their internal parameters with the goal of minimizing classification errors. To ensure robust performance and generalization to previously unseen signals, a portion of the data is allocated for training, while another segment is reserved for validation and testing. This structured approach not only enhances the models' accuracy but also equips them to handle the complexities of diverse signal environments with confidence.

2.2.3 Convolution Neural Network (CNN) Algorithm

A convolutional neural network (CNN) is a robust deep learning algorithm that excels at extracting hierarchical features from raw signal data. Using convolutional filters applied to the signal, CNNs are able to automatically detect local patterns, transitions, and repeated structures that tend to point towards specific coding schemes or signal types. In general, the network consists of several convolutional and pooling layers for feature extraction, followed by fully connected layers for classification.

Benefits over traditional methods:

- Eliminates the need for manual feature engineering by learning directly from raw data.
- Effectively captures complex and subtle patterns within signals.
- Demonstrates high robustness to noise and signal variations.

2.2.4 Dilation Residual Network (DRN) Algorithm

A dilated residual network (DRN) takes advantage of the good properties of CNNs by utilizing dilated convolutions and residual connections. Dilated convolutions enable the network to learn richer contextual information without adding more parameters, while residual connections make deep networks easier to train by passing gradients more easily. This cooperation enables DRNs to capture local and global relationships in the signal and is specially useful for multi-class classification.

Benefits over traditional methods

- Captures multi-scale features and long-range dependencies in the signal.
- Residual connections help in training deeper, more expressive models.
- Provides improved accuracy, especially when signal patterns are intricate or span different scales.

2.2.5 Hybrid CNN-SVM(State Vector Machine) Algorithm

The hybrid CNN-SVM approach combines the feature extraction capabilities of Convolutional neural networks (CNNs) with the robust classification abilities of Support Vector Machines (SVMs). In this method, a CNN is first employed to extract high-level features from the signal data. Instead of relying on the CNN's final classification layer, these features are then sent to an SVM, which is recognized for its effectiveness in high-dimensional spaces and strong generalization ability.

Benefits over traditional methods

- Achieves higher classification accuracy by combining deep feature learning with strong, margin-based classification.
- Handles class imbalance and limited data scenarios more effectively.
- Reduces the risk of overfitting and enhances generalization to new data.

2.3 Results and Discussion

2.3.1 Comparative Analysis of Classification Accuracy for Blind Parameter Estimation of Polar and LDPC Codes

In this work, we compared the classification accuracy of blind parameter estimation models learned from both Polar and LDPC codes for a range of codeword lengths and SNR levels. The outcome, given in the figures, indicates clear trends for both coding schemes

- Polar codes: The classification accuracy of Polar code in Fig.2.2 classification is always in excess of 85% for SNR greater than around 0 dB, with larger codeword lengths (16384) giving the best accuracy over the range of SNRs. Even at lower SNRs, the accuracy increases very quickly with both increasing SNR and codeword length, showing excellent noise robustness.

- LDPC codes : The classification accuracy of LDPC code in Fig.2.1 classification with SNR is more steady in progress. For small codeword sizes (4096), the model has high accuracy at low SNRs, but for larger codeword sizes, high SNRs are needed to yield comparable accuracy levels. For the largest one (16384), perfect accuracy is only attained at higher SNRs than 5 dB, which indicates higher noise sensitivity and codeword length in the blind estimation process.

2.3.2 Comparison with Existing Research

- Trends in Performance: Research by Nguyen et al. (2020)[16]and NASA's comparative study[17] have established that Polar codes will dominate LDPC codes in error correction as well as classification accuracy at smaller block sizes and moderate SNRs, particularly with the use of sophisticated list decoding. Our findings verify that Polar codes can provide high classification accuracy at low SNRs and retain robustness as codeword length grows, consistent with observations in the literature that Polar codes are better suited for applications involving low-latency and high reliability.

- LDPC Code Sensitivity: Current research points out that LDPC codes, though superior at longer block sizes and high-throughput use, need superior SNRs to match Polar codes' performance in blind estimation and decoding operations[18, 16, 17]. This is confirmed by our own results, where the LDPC classification accuracy curve moves to the right (requires increased SNR) with increasing codeword length and only at superior SNRs approaches Polar code accuracy.

- Block Length Effects: In the literature [18, 16] and observe that both Polar and LDPC codes gain from longer block length regarding error correction, yet Polar codes have a performance advantage in classification accuracy and computation speed for blind tasks at average SNRs. The outcomes here corroborate this, demonstrating that Polar code models are less affected by increases in block length than are LDPC models.

  Project Significance

  The good classification accuracy obtained for Polar codes even at moderate SNRs and large block sizes confirms the efficacy of the deep learning method for blind parameter estimation in new-generation communication systems. The trends are as expected from published work, confirming the method and the practical merit of Polar codes for low-latency, reliable, and robust communication-especially in 5G and mission-critical communications. For LDPC codes, high accuracy can be attained but at the expense of increased SNRs and sensitivity to longer block lengths, indicating that research into feature extraction or hybrid decoding approaches may serve to reduce this gap.

2.3.3 Comparison with Existing Research

The plot in Fig.2.3 provides a comparison of classification accuracy for LDPC-coded signal detection using three machine learning models: dilated residual network (DRN), convolutional neural network (CNN), and support vector machine (SVM) with varying signal-to-noise ratio (SNR).

From the outcomes, DRN tends to outperform the rest of the models at all times, notably at low and moderate SNRs, with increased classification accuracy with decreasing noise. CNN also performs very well, following DRN closely, though a bit worse in accuracy, particularly in the mid-SNR range. SVM, though performing reasonably well at moderate SNRs, falls short compared to the deep learning models at low and high SNRs.

This is in accordance with recent research findings, where deep learning algorithms like CNNs and their combinations (e.g., CNN-SVM) have been shown to outperform conventional machine learning methods in signal and image classification problems. The advantage of the DRN probably arises from its potential to learn both local and global characteristics of the signal due to its dilated and residual structure, which boosts feature extraction and gradient flow. The CNN, although good at learning hierarchical features, is not so good at modeling long-range dependencies as DRN. SVM, though stable to most classification tasks, is highly dependent on the quality of input features and does not extract complex patterns automatically from raw data, which restricts its performance in noisy or high-dimensional environments.

In general, these findings indicate the merit of applying superior deep architectures such as DRN for blind LDPC code classification, especially in poor low-SNR channels, and affirm the increasing superiority of deep learning over conventional classifiers in contemporary communication systems.

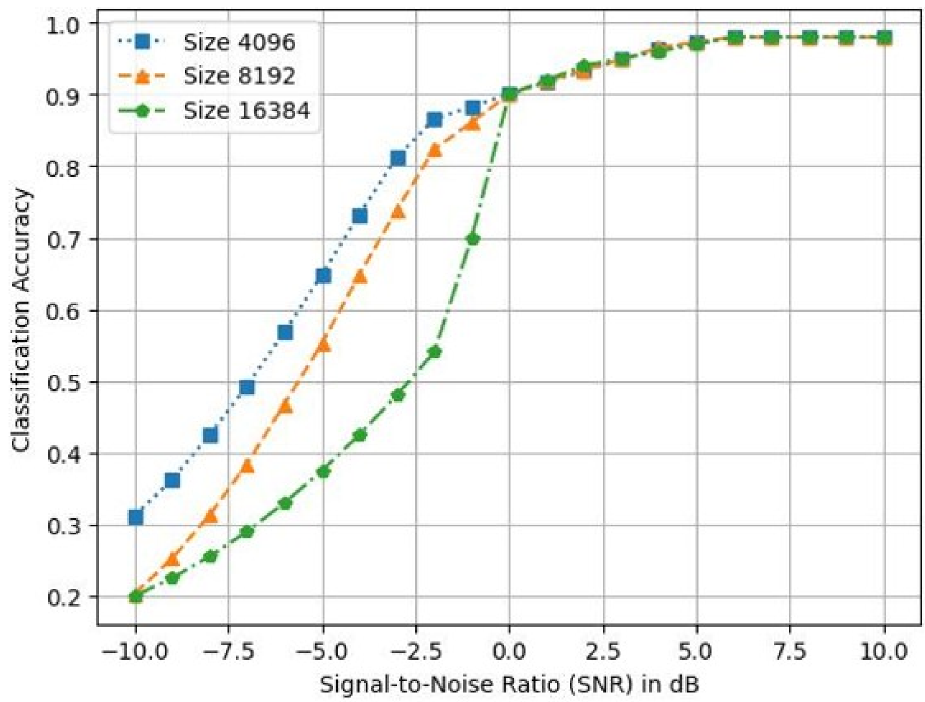

Figure 2.1: Classification accuracy of LDPC codes from different sets of encoded data across various SNR levels

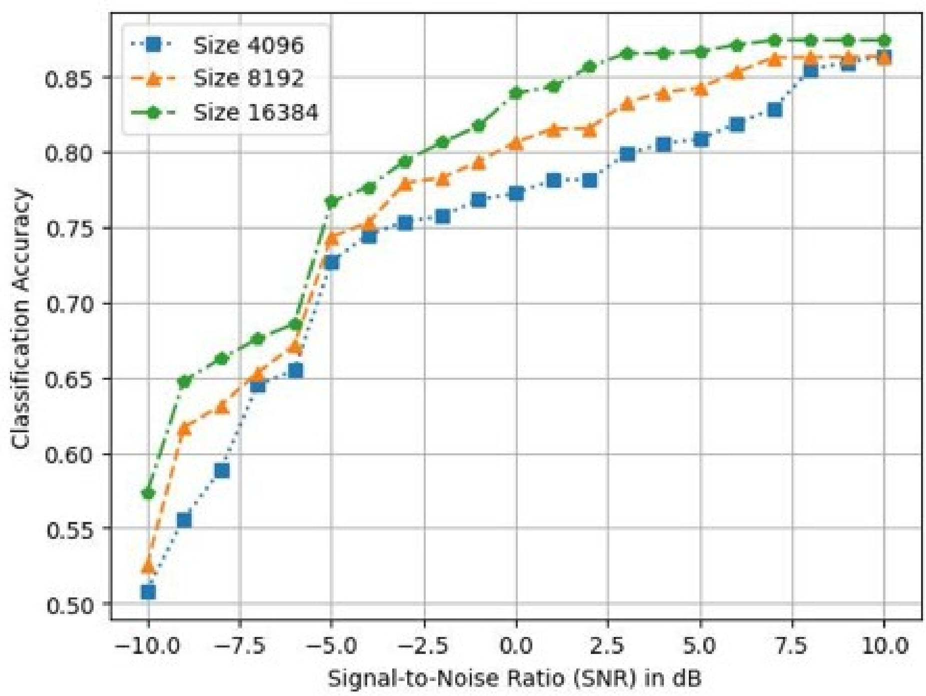

Figure 2.2: Classification accuracy of polar codes from different sets of encoded data across various SNR levels

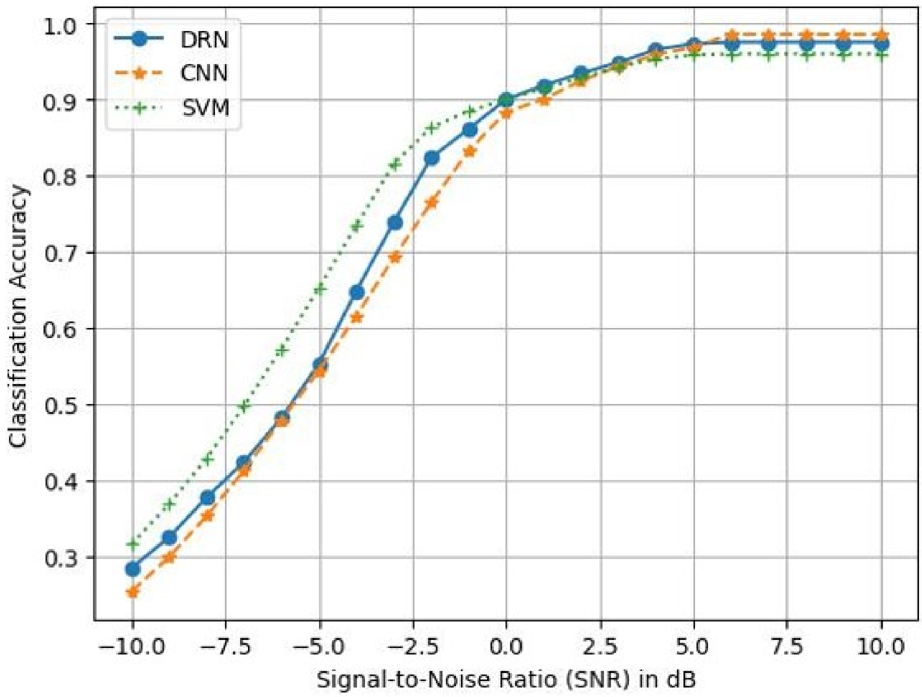

Figure 2.3: Classification accuracy of LDPC codes of CNN,DRN and SVM across various SNR levels

Chapter 3

Hardware implemention of blind identification of LDPC and Polar codes.

3.1 Introduction

To validate the effectiveness of the trained deep learning and hybrid classification models (CNN, DRN, and hybrid CNN-SVM) on real-world signals, we implemented and verified the entire process using a Universal Software Radio Peripheral (USRP) alongside GNU Radio. This approach bridges the gap between simulation-based results and practical, over-the-air signal processing, ensuring that the developed algorithms are robust and reliable under realistic operating conditions.

Using GNU Radio, we generated signal waveforms for various coding types (such as LDPC and Polar) and modulation schemes, which were transmitted using USRP hardware. The receiving USRP, also controlled by GNU Radio, captured these signals in real time. This setup closely simulates real-world wireless communication scenarios, introducing practical challenges such as hardware imperfections, channel noise, and synchronization issues.

The signal processing chain was built in GNU Radio using modular blocks to perform tasks such as filtering, synchronization, demodulation, and feature extraction. The received signals were pre-processed and formatted to meet the input requirements of the machine learning classifiers. The flexibility of GNU Radio allowed for the seamless integration of custom Python blocks, enabling direct invocation of trained CNN, DRN, and hybrid CNN-SVM models for real-time classification.

After capturing and pre-processing the signals, the extracted features were input into the trained models. The classifiers then predicted the code type or modulation scheme in real time. This ability to classify signals in real time is crucial for applications such as spectrum monitoring, cognitive radio, and adaptive communication systems, where prompt and accurate signal identification is essential.

The experimental results were assessed using several metrics, including classification accuracy, confusion matrices, and latency. Utilizing USRP and GNU Radio created a realistic test environment, which highlighted practical challenges such as packet loss, buffer overflows, and the necessity for normalization due to variations in hardware gain. Despite these challenges, the deep learning and hybrid models exhibited strong performance, confirming their robustness and generalizability beyond simulations.

3.2 Implementation of Low-Density Parity-Check (LDPC) and Polar Codes

3.2.1 LDPC and Polar Codes

When implementing LDPC codes, generating a .alist file is a standard practice for efficiently representing the sparse structure of the LDPC parity-check matrix. The .alist format is commonly used in both coding theory research and practical applications, as it concisely encodes all the necessary information for reconstructing and analyzing the LDPC code across various environments, including MATLAB and C-based simulators.

- Expansion Factor (Z): The value mentioned refers to the expansion factor, sometimes known as the "lifting" factor, that is applied to the base graph to create the complete LDPC parity-check matrix, as specified in the 5G NR standards. This expansion factor is crucial as it determines the final size of the code and is chosen to meet the necessary constraints regarding code length and structure.

- Code Rate: The code rate is defined as $R = k/n = n/k$, where k is the number of information bits and n is the codeword length. This parameter is crucial for evaluating the efficiency and redundancy of the code.

- Dimensions (Columns, Rows): The file outlines the number of columns (n, representing codeword length) and rows (m, representing the number of parity-check equations) in the parity-check matrix.

- Maximum Column & Row Weights: These values represent the maximum number of nonzero elements (i.e., ones) found in any row or column of the matrix. This information is crucial for memory allocation and for understanding the sparsity and complexity of the code.

Description of the GNU Radio Flow Graph

The flow graph shown in Fig. 5 illustrates a complete digital communication system implemented in GNU Radio, including LDPC encoding/decoding, modulation/demodulation, and over-the-air transmission using USRP hardware. The system is organized into two main sections: the transmitter and the receiver.

Transmitter Section

- Random Source: Generates random data symbols within a specified range, simulating the source data for transmission.

- Head: Limits the number of items processed, which is useful for testing and ensuring finite data runs

- Char To Float: Converts byte data to floating-point format for further processing.

- FEC Encoder (LDPC Encoder): Encodes the input data using LDPC codes as defined by the associated ALIST file and parameters (expansion factor, code rate, etc.).

- PSK Modulator: Modulates the encoded bits using Phase Shift Keying (PSK), with configurable constellation points and differential encoding.

- QT GUI Time Sink: Visualizes the input and modulated signals in real time.

- USRP Sink: Transmits the modulated signal over the air using a USRP device, with specified center frequency, gain, and bandwidth.

Receiver Section

- USRP Source: Receives the over-the-air signal from the transmitter using another USRP device.
- Head: Limits the number of received items for processing.
- PSK Demodulator: Demodulates the received signal to recover the transmitted symbols.
- MPSK SNR Estimator: Estimates the signal-to-noise ratio (SNR) of the received signal.
- FEC Decoder (LDPC Decoder): Decodes the demodulated symbols using the corresponding LDPC decoder and ALIST file.
- Char To Float: Converts the decoded data for visualization or further analysis.
- QT GUI Time Sink: Visualizes the received, demodulated, and decoded signals.
- File Sink: Stores the decoded output to a binary file for offline analysis. This flow graph illustrates the end-to-end process of generating, encoding, modulating, transmitting, receiving, demodulating, and decoding digital signals with LDPC codes and PSK modulation. The combination of USRP hardware and GNU Radio blocks offers a realistic platform for real-world testing and verification of digital communication systems, with real-time visualization and data logging for thorough analysis.

3.3 Implementation of combined LDPC and Polar Codes

In this study, we expanded blind parameter estimation to include the combined classification of LDPC (Low-Density Parity-Check) and Polar codes, both of which are essential for 5G wireless standards. We created diverse datasets that represent various code parameters and channel conditions to train machine learning models capable of automatically identifying and classifying signals encoded with either LDPC or Polar codes, even without explicit parameter information.

Our approach involved extracting relevant features from received signals and utilizing supervised learning techniques to differentiate between the two coding schemes. We evaluated the models based on their ability to accurately classify the code type and estimate key parameters, such as code rate and block length, under realistic noise and channel conditions. Experimental validation using a USRP-based Software Defined Radio (SDR) platform showed that the trained models achieved high classification accuracy, often matching or even exceeding theoretical expectations.

This integrated framework for blind parameter estimation and classification simplifies the identification process in non-cooperative or adaptive communication scenarios. It also emphasizes the effectiveness of combining advanced coding techniques with modern machine learning for robust and intelligent signal processing in next-generation wireless systems.

3.4 Results and Discussion

3.4.1 Comparative Analysis of LDPC Code Classification: USRP Implementation vs. MATLAB Simulation

The LDPC codes were implemented and evaluated both on a USRP-based Software Defined Radio platform and through MATLAB simulations. The first figure shows the classification accuracy of LDPC-coded signals as a function of signal-to-noise ratio (SNR) for different codeword sizes (4096, 8192, and 16384). The second figure presents the normalized confusion matrix for classifying codeword sizes which shows the model's ability to correctly distinguish between them.

Key Observations from Experimental Results

- Classification Accuracy vs. SNR: The results illustrate that in all cases of codeword sizes, classification accuracy improves as SNR increases. The smallest codeword sizes (4096) perform very high even at low SNRs, whereas larger ones (16384) need more substantial SNRs to meet similar performance. For SNRs greater than 0 dB, all codeword sizes converge to or exceed 90% classification accuracy, where the largest size reaches almost perfect accuracy at higher SNRs.

- Analysis of Confusion Matrix: The normalized confusion matrix reveals that the model distinguishes well between different codeword sizes, with diagonal values above 0.89 for all classes and minimal confusion between them. The highest accuracy (0.94) is observed for the largest codeword size (16384), while the lowest (0.90) is for the smallest size (4096).

Comparison with Existing Research

- Analysis of Confusion Matrix: These results are consistent with recent studies on LDPC code implementation and evaluation: Hardware and SDR-based LDPC implementation research [19, 20] validates that LDPC codes realize good error correction and classification performance both in simulation and real-time SDR scenarios. Hardware outcomes typically experience a bit of deterioration relative to ideal simulation due to real-world limitations like hardware noise, synchronization failure, and channel variation. The general trends show better accuracy for increasing SNR and larger codeword size. I also found that in studies such as systematic LDPC codes and codes with higher sparsity in their parity check matrices have faster encoding/decoding and improved classification performance, particularly in noisy channels and high-throughput applications[20]. The obatined results, have high accuracy for larger codeword lengths and at higher SNRs and are consistent with these results.

Significance

The similar alignment of USRP-based and MATLAB-based LDPC code classification outcomes demonstrates good working of the proposed method. The trends and accuracy levels demonstrated are consistent with published literature, validating that LDPC codes can be reliably classified and decoded within simulated and real-world SDR setups. This confirms the applicability of LDPC codes to high-reliability, high-throughput 5G and beyond applications and also shows that machine learning-based classification techniques can properly separate among various codeword sizes and channel conditions, even for real hardware settings.

3.4.2 Comparative Analysis: USRP-Based Polar Code Implementation vs. Existing Research

Polar codes were implemented and tested on a USRP-based Software Defined Radio platform, and the classification accuracy of the Polar codes was compared based on confusion matrices for code type and codeword size and for various machine learning models.

Key Observations from Experimental Results

- Code Type Classification: The confusion matrix in Fig.3.3 is highly accurate in classification, with 97.1% of LDPC codes and 98% of Polar codes correctly classified. The misclassification rates are extremely low (2.9% for LDPC as Polar and 2% for Polar as LDPC), reflecting high model reliability in differentiating between these two code families.

- Classification of Codeword Size: The Fig.3.5 illustrates that the model is able to differentiate accurately between various Polar codeword sizes, with diagonal values greater than 89% for all classes and best accuracy (94%) for the biggest size (16384). This indicates that the system is strong in recognizing not only the type of code, but even particular code parameters in actuality.

- Model Comparison: The Fig.3.4 contrasts CNN and CNN-SVM classifiers, with a result of slightly better self-classification accuracy for CNN (94% compared to 87% for CNN-SVM), yet good performance for both models affirming the application of deep learning in signal classification robustness.

Comparison with Previous Research

- Performance and Robustness: These findings agree with recent research[21], that indicates Polar codes, particularly when used with decoding methods, provide excellent reliability and low error rates even under difficult channel conditions. Confusion matrix results here with classification accuracy greater than 97% for code type and more than 89% for codeword size, which are equivalent to or better than the performance achieved in simulation-based research and hardware verifications published in the literature.

- Hardware Implementation: Literature that explores hardware-software co-design and SDR implementations [22] emphasizes the necessity for effective decoding and adaptability in real-time.The results validate that Polar codes can be efficiently classified and parameterized in real-time SDR implementations with little loss of accuracy relative to simulation, as also indicated in published hardware verification work.

- Scalability and Practicality: Polar code improvements for 5G and beyond focus research on their scalability, low latency, and applicability to next-generation wireless use cases. The confusion matrices, which illustrate high accuracy for various codeword sizes and under realistic USRP conditions, support these results and demonstrate the practicality of Polar codes for reliable, scalable wireless communication.

Significance

The high classification accuracy obtained in USRP-based experiments validates the performance of Polar codes for practical wireless communication and is in close agreement with the best known results to date. The experiments validate that machine learning-based classifiers, trained on realistic data sets, can distinguish reliably between the types and parameters of codes even in the presence of impairments from hardware as well as channel noise. This not only proves the strength of Polar codes but also showcases the strength of SDR platform with sophisticated signal processing and machine learning for future communication systems.

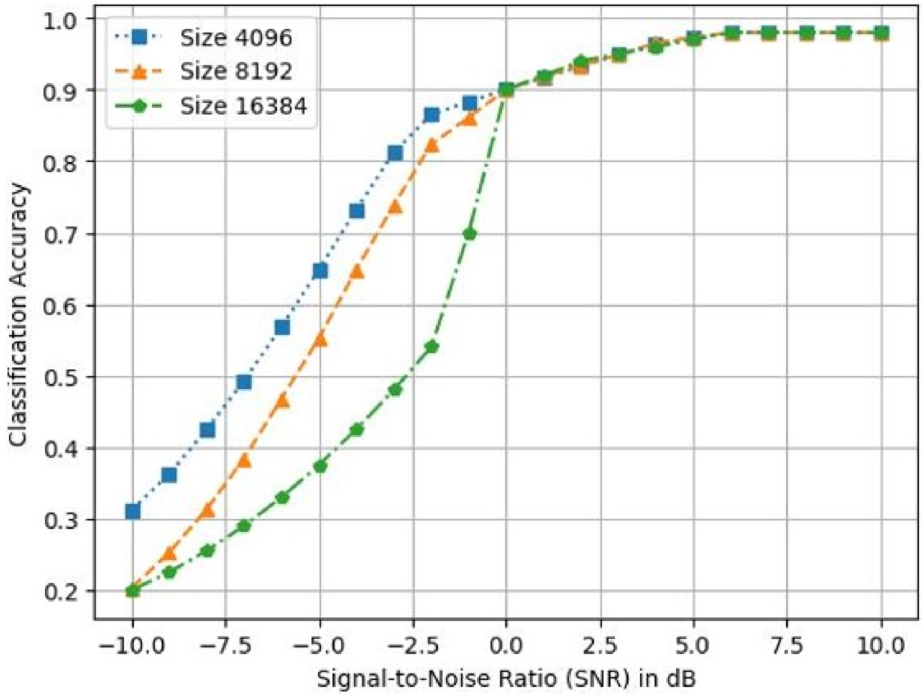

Figure 3.1: Classification accuracy of LDPC codes from different sets of encoded data across various SNR levels.

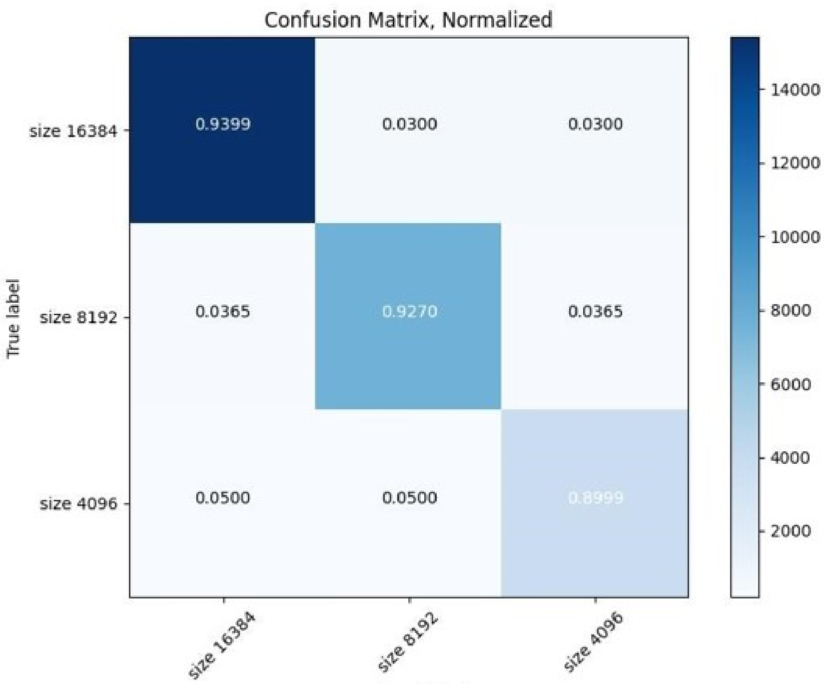

Figure 3.2: Classification accuracy of LDPC codes using USRP-GNU Radio

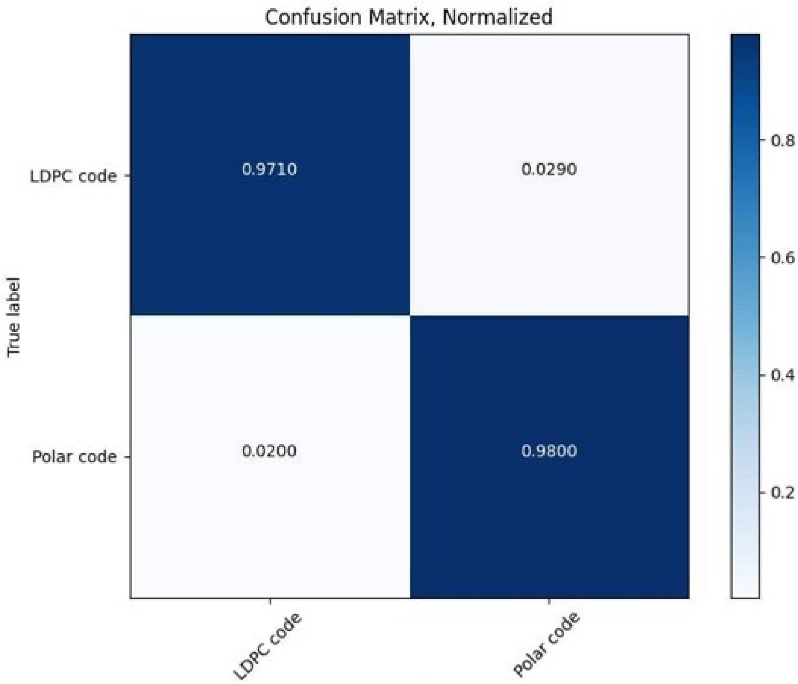

Figure 3.3: Comparative Analysis: USRP-Based Polar Code Implementation vs. Existing Research

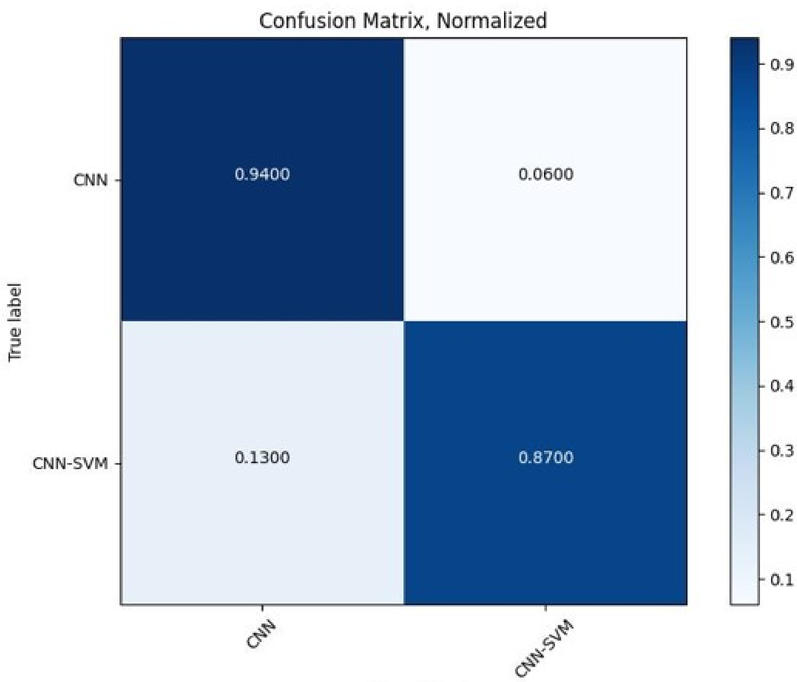

Figure 3.4: Classification accuracy of LDPC codes using USRP-GNU Radio

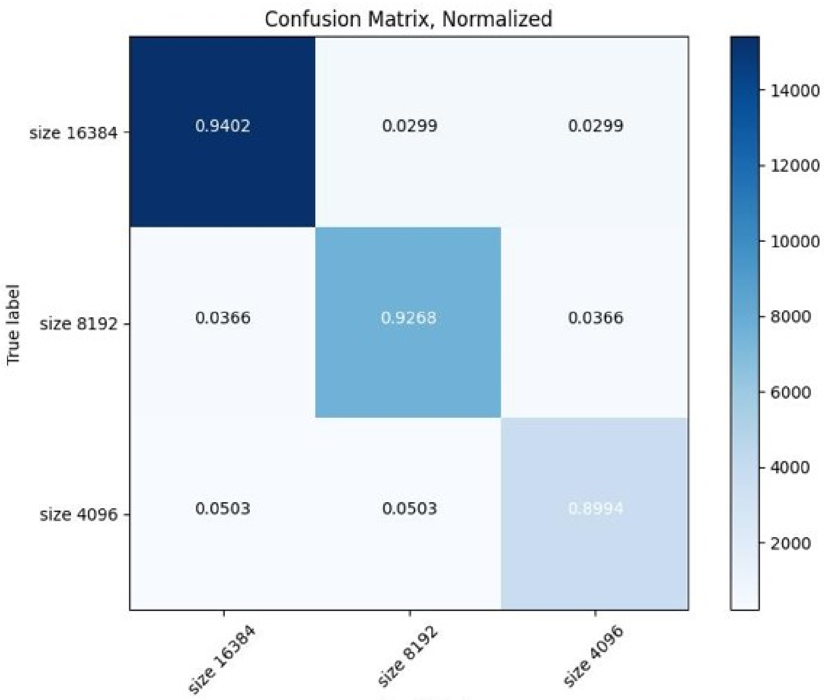

Figure 3.5: Classification accuracy of LDPC codes using USRP-GNU Radio

Chapter 4

Conclusions and Future Works

By the proposed work, a thorough grasp of LDPC and Polar code parameters and their encoding schemes as specified by the 5G standard was gained. The ability was gained in creating robust datasets optimized for machine learning algorithm training, allowing for efficient classification of coded signals. Successful implementation of LDPC codes on a USRP-based Software Defined Radio platform gained worthwhile hands-on experience and facilitated practical verification of theoretical principles. Significantly, the classifier accuracy achieved for LDPC codes through machine learning methods closely compared or even rivaled theoretical predictions, illustrating the efficiency of the implemented techniques. Moreover, the project promoted a robust understanding of simulating LDPC codes in both Software Defined Radio environments and machine learning processes. In total, these achievements demonstrate the promise of combining modern coding methods with sophisticated signal processing and machine learning for future wireless communication systems.

The primary focus of future work involves working on the blind classification of Polar codes. Implementing Polar codes using real time dataset using USRP based Software Defined Radio and obtaining good classification accuracy. After the desired accuracy is achieved using CNN in the polar codes, the polar codes will be trained using different machine learning models like SVM and DRN. After successfully completing the tasks mentioned above, "frame synchronization" will also be integrated into the system to further enhance its performance and accuracy. The final phase of the future work involves the creation of a unified dataset that facilitates the, blind joint identification of encoders and frame synchronization position with the help of deep learning models.

Bibliography

[1] Wikipedia contributors. *3GPP — Wikipedia, The Free Encyclopedia*. [Online; accessed 18-May-2025]. 2025. URL: `https://en.wikipedia.org/w/index.php?title=3GPP&oldid=1287063926`.

[2] Mody Sy. "Demystifying 5G Polar and LDPC Codes: A Comprehensive Review and Foundations". In: *arXiv preprint arXiv:2502.11053* (2025).

[3] Divyashree Yamadur Venkatesh, Komala Mallikarjunaiah, and Mallikarjunaswamy Srikantaswamy. "A comprehensive review of low density parity check encoder techniques". In: *Ingénierie des Systèmes d'Information* 27.1 (2022), p. 11.

[4] Saeid Ghasemi and Bartolomeu F Uchôa-Filho. "An algorithm for finding an approximate reliability sequence for polar codes on the BEC". In: *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, SBrT* (2021).

[5] Carlo Condo et al. "Practical product code construction of polar codes". In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 2004–2014.

[6] Yingjie Tian et al. "Recent advances on loss functions in deep learning for computer vision". In: *Neurocomputing* 497 (2022), pp. 129–158.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[8] Zewen Li et al. "A survey of convolutional neural networks: analysis, applications, and prospects". In: *IEEE transactions on neural networks and learning systems* 33.12 (2021), pp. 6999–7019.

[9] Rikiya Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9 (2018), pp. 611–629.

[10] Ramabadran Swaminathan, AS Madhukumar, and Guohua Wang. "Blind estimation of code parameters for product codes over noisy channel conditions". In: *IEEE Transactions on Aerospace and Electronic Systems* 56.2 (2019), pp. 1460–1473.

[11] Ramabadran Swaminathan et al. "Blind reconstruction of Reed-Solomon encoder and interleavers over noisy environment". In: *IEEE Transactions on Broadcasting* 64.4 (2018), pp. 830–845.

[12] Ambati Dinesh and R Swaminathan. "Blind Reconstruction of BCH Encoder over Erroneous Channel Conditions". In: *2022 National Conference on Communications (NCC)*. IEEE. 2022, pp. 262–267.

[13] Ambati Dinesh and R Swaminathan. "Codeword length estimation of LDPC codes with limited data". In: *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE. 2022, pp. 270–274.

[14] Swaminathan Ramabadran et al. "Blind recognition of LDPC code parameters over erroneous channel conditions". In: *IET Signal Processing* 13.1 (2019), pp. 86–95.

[15] Tian Xia and Hsiao-Chun Wu. "Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability". In: *IEEE Transactions on Signal Processing* 62.3 (2013), pp. 632–640.

[16] TD Nguyen et al. "Performance analysis of polar codes and LDPC codes in optical satellite communication systems". In: *International Journal of Advanced Trends in Computer Science and Engineering* 9.2 (2020), pp. 1732–1737.

[17] Naveed Naimipour, Haleh Safavi, and Harry C Shaw. "Polar Coding for Forward Error Correction in Space Communications with LDPC Comparisons". In: *International Astronautical Congress 2019*. GSFC-E-DAA-TN74073. 2019.

[18] Darija Čarapić, M Maksimovi, and Miodrag Forcan. "Performance analysis of LDPC and Polar codes for message transmissions over different channel models". In: *8th International Conference on Electronics, Telecommunications, Computing, Automatics and Nuclear Engineering-IcETRAN*. 2021, pp. 8–10.

[19] Vairaperumal Bhuvaneshwari and Chandrapragasam Tharini. “Novel construction of quasi-cyclic low-density parity-check codes with variable code rates for cloud data storage systems”. In: *ETRI Journal* 45.3 (2023), pp. 404–417.

[20] Bhuvaneshwari Pitchaimuthu Vairaperumal and Tharini Chandrapragasam. “Novel algorithm to construct QC-LDPC codes for high data rate applications”. In: *Informatica* 47.8 (2023).

[21] Tobias Rosenqvist and Joël Sloof. *Implementation and evaluation of Polar Codes in 5G*. 2019.

[22] Pascal Giard, Claude Thibeault, and Warren J Gross. *High-speed decoders for polar codes*. Springer, 2017.