Performance Evaluation of 5G MIMO Detectors With Deep Learning

Urooj Naveed Akhter

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

June 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Urooj Naveed Akhter**

Entitled: **Performance Evaluation of 5G MIMO Detectors With Deep Learning**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Dongyu Qiu

_____ External Examiner
Dr.

_____ Examiner
Dr. Rodolfo Coutinho

_____ Supervisor
Dr. Wei Ping Zhu

_____ Co-supervisor
Prof. Paulo Diniz

Approved by _____
Abdelwahab Hamou-Lhadj, Chair
Department of Electrical and Computer Engineering

_____ 2025

_____
Mourad Debbabi , Dean
Faculty of Engineering and Computer Science

Abstract

Performance Evaluation of 5G MIMO Detectors With Deep Learning

Urooj Naveed Akhter

Fifth-generation (5G) wireless networks promise ultra-reliable low-latency communication, enhanced mobile broadband, and massive machine-type connectivity—capabilities made possible by advanced technologies such as massive MIMO, OFDM-based modulation, and dynamic spectrum access. However, these benefits introduce new challenges for receiver design, including increased complexity, high data rates, and time-varying channels that degrade the performance of traditional detection methods.

This thesis presents a comprehensive performance analysis of 5G networks using deep learning–based receiver. A simulation framework is designed to evaluate 5G MIMO receivers under diverse channel environments, including AWGN, Rayleigh fading, and the 3GPP TDL-C model. A custom dataset is generated using MATLAB's 5G Toolbox to simulate OFDM-based transmissions across multiple MIMO configurations and modulation schemes. Various neural network architectures including Fully Connected Neural Network (FCNN), Convolutional Neural Network (CNN), Residual Neural Network (ResNet) and Long Short Term Memory (LSTM network), are implemented and benchmarked against Maximum likelihood (ML), sphere decoding, and MMSE detection.

Performance metrics including bit error rate (BER), symbol error rate (SER), processing speed, and computational complexity are evaluated. The results highlight trade-offs between accuracy and efficiency and provide insights into the feasibility of deep learning–based detection for practical 5G deployments.

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisors, Prof. Wei Ping Zhu and Prof. Paulo S. R. Diniz, whose guidance, expertise, and continuous support were invaluable throughout the course of this research. Their insightful feedback and encouragement have been fundamental in shaping this work.

I also extend my thanks to Dr. Ali Mohebbi for his contributions and valuable suggestions that greatly enriched this study.

I am grateful to the faculty and staff at the Department of Electrical and Computer Engineering, Concordia University, for providing an excellent academic environment and the resources necessary to conduct this research. Special thanks go to my colleagues, friends, and family who offered moral support and technical assistance when needed.

Finally, I would like to acknowledge the financial support provided by Concordia University, which helped facilitate the research activities and experiments reported in this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

5G	Fifth Generation (Wireless Networks)
6G	Sixth Generation (Wireless Networks)
AR(1)	Autoregressive Process of Order 1
ASIC	Application-Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CSI	Channel State Information
CNN	Convolutional Neural Network
CNN-RNN	Convolutional Recurrent Neural Network
DNN	Deep Neural Network
DPRNN	Dual-Path Recurrent Neural Network
FCNN	Fully Connected Neural Network
FLOPs	Floating Point Operations
FPGA	Fiel/d-Programmable Gate Array
LSTM	Long Short-Term Memory
MATLAB	Matrix Laboratory (Software)

MFLOPs	Million Floating Point Operations
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
PIT	Permutation-Invariant Transformer
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RE-DetNet	Reduced Encoder DetNet
RNN	Recurrent Neural Network
S4M	State-Space for MIMO
SAD	Self-Attention Detector
SD	Sphere Decoding
SE	Spectral Efficiency
SER	Symbol Error Rate
SkiM	Skipping Memory
SNR	Signal-to-Noise Ratio
SSM	State-Space Model
STFT	Short-Time Fourier Transform
TFPSNet	Time–Frequency Path Scanning Network
ZF	Zero Forcing

Chapter 1

Introduction

1.1 Background and Motivation

The rapid evolution of fifth-generation (5G) wireless networks has brought unprecedented demands on new communication technologies and architectures. 5G networks leverage advanced technologies such as massive multiple-input, multiple-output (MIMO) and high-order modulation schemes to support high data rates, ultra-low latency, and enhanced reliability [6]. However, as the number of antennas and modulation order increase, receiver design faces significant computational complexity and real-time processing challenges. Moreover, with the emergence of 6G on the horizon - envisioned to enable integrated sensing, AI-driven networks, and terahertz communications - the next generation introduces even more stringent requirements and higher design complexities that build upon the unresolved issues of 5G. Maximum likelihood (ML) detection, which minimizes the Euclidean distance between the received signal and the candidate transmitted signals, offers optimal performance but becomes computationally prohibitive in large-scale systems [7, 8]. Consequently, conventional linear detectors, such as zero forcing (ZF) and minimum mean square error (MMSE) equalizers, are commonly used despite their inherent limitations, particularly under fading and interference conditions.

Sphere decoding (SD) algorithms have been developed to approximate maximum likelihood (ML) detection with reduced complexity compared to exhaustive search. These methods often employ MMSE-based initialization, where the output of a low-complexity MMSE detector is used to

set the initial radius and center of the decoding sphere. Additionally, the Schnorr–Euchner enumeration strategy orders symbol candidates based on their Euclidean distance, enabling a depth-first branch-and-bound search that prunes unlikely paths early. Together, these enhancements allow SD to deliver near-ML performance with less complexity than brute-force ML detection [9, 10]. However, sphere decoding still incurs substantially higher computational cost than linear detectors (e.g., ZF and MMSE), and its complexity can grow rapidly with increasing modulation order or number of antennas, limiting its practicality for real-time systems [8].

Recent advances in machine learning have sparked a growing interest in employing deep neural networks for signal detection in wireless communications [11]. Numerous studies have shown that neural network-based methods can learn complex nonlinear mappings from received noisy signals to bit-level decisions, often achieving performance close to that of maximum likelihood (ML) detection while significantly reducing inference complexity [12, 13].

This thesis addresses a gap in the literature by proposing a unified simulation framework that compares conventional detection methods, SD, and deep learning–based detectors under realistic 5G MIMO conditions. A custom dataset is generated using MATLAB's 5G Toolbox to simulate both AWGN and Rayleigh fading channels in which temporal correlation is incorporated via an autoregressive (AR(1)) process. The investigation focuses primarily on 16-QAM modulation in a 2×2 MIMO system, with supplementary results for other modulation orders. Deep learning models — including fully connected neural networks (FCNN), CNN, and ResNet architectures — are developed and trained from scratch to directly map the received signal (represented by its real and imaginary components and the effective SNR) to bit-level classifications.

By comparing the performances of these diverse detection strategies, this work provides valuable insights into the trade-offs between detection accuracy and computational complexity, thereby contributing to the design of more efficient and scalable 5G receivers. The key research problem addressed in this work is the development of computationally efficient detection techniques that can approach near-ML performance while remaining feasible for real-time 5G applications [14]. This comparison is motivated by the growing need for low-latency, high-throughput communication systems where traditional methods fall short in accuracy or impose excessive computational burden, making deep learning-based alternatives timely and necessary.

1.2 Literature Review

Multiple-Input Multiple-Output (MIMO) communications was introduced in the mid-1990s to exploit multipath propagation for capacity gains. Foschini and Gans first demonstrated that in a rich scattering environment, the capacity of a narrowband MIMO channel grows linearly with the minimum of the number of transmit and receive antennas [15]. This finding was later extended by Telatar, who derived precise analytical expressions for the ergodic capacity of MIMO systems operating under Rayleigh fading conditions [16]. These foundational studies established MIMO as a cornerstone for future wireless standards, laying the theoretical groundwork for spatial multiplexing and diversity techniques, which are integral to modern standards such as 4G LTE and 5G NR.

In the early stages of MIMO research, linear detection techniques gained popularity due to their relative simplicity and manageable computational requirements. One widely used method was Zero-Forcing (ZF) detection, which attempts to invert the channel effects and decouple the transmitted data streams. Although ZF effectively cancels inter-stream interference, it does so at the cost of amplifying noise, particularly when the channel matrix is poorly conditioned [17]. To address this limitation, the Minimum Mean Square Error (MMSE) detector was introduced as an improvement over ZF. MMSE seeks a balance between interference suppression and noise amplification, offering better performance under moderate to low signal-to-noise ratio (SNR) conditions. However, both ZF and MMSE detectors suffer from performance degradation in challenging channel environments, and their computational cost increases with the number of transmit antennas.

For scenarios requiring near-optimal detection performance, Maximum Likelihood (ML) detection is widely considered a benchmark due to its ability to minimize the probability of symbol error under Gaussian noise assumptions [7]. ML aims to identify the most likely transmitted signal vector by exhaustively comparing all possible transmission hypotheses. Although this approach yields excellent error rate performance, its computational complexity scales exponentially with the number of transmit antennas and modulation order, making it impractical for real-time applications, particularly in 5G systems with high-dimensional MIMO links [7]. To mitigate this challenge, Sphere Decoding (SD) was proposed as a more efficient alternative [9]. Instead of evaluating all possible transmitted vectors, Sphere Decoding (SD) limits the search to a hypersphere centered around the

received signal, reducing the number of symbol candidates that must be evaluated. One key enhancement is MMSE-based initialization, which uses the solution from a low-complexity MMSE detector to define the center of the sphere and estimate the initial radius. This effectively narrows the search space from the outset [18]. Furthermore, the Schnorr-Euchner enumeration prioritizes symbol candidates based on their likelihood, allowing a more efficient traversal of the search tree [19]. These improvements significantly improve SD's computational efficiency while maintaining near-ML performance in small-to-moderate MIMO dimensions and at high-to-moderate SNRs, where the expected number of lattice points inside the decoding sphere remains tractable.

In recent years, there has been a significant surge in applying deep learning techniques to various aspects of 5G wireless networks. Researchers have explored a wide range of applications, including channel estimation [20, 21], beamforming [22], end-to-end signal detection [23], and resource allocation [24,25]. The motivation behind these efforts stems from the inherent limitations of traditional signal processing methods, particularly when dealing with the highly complex optimal detectors.

Early attempts to apply deep learning to MIMO detection leveraged fully connected neural networks (FCNNs) to learn the mapping from the received signal to the original transmitted symbols. Specifically, the input to the network consisted of the real and imaginary parts of the complex-valued received signal, and the output was trained to predict the most likely transmitted data vector — effectively mimicking the role of a maximum likelihood (ML) detector [26]. studies such as [27] demonstrated that FCNNs, even with relatively shallow architectures, can learn to generalize from AWGN to fading channels when trained on representative datasets.

To improve scalability and performance, the DetNet architecture was proposed by Samuel et al. [12]. This approach unfolds the iterative steps of a projected gradient descent algorithm into a sequence of neural network layers, where each layer corresponds to one iteration of the solver. The result is a deep neural network with fixed computational complexity per layer. In practice, DetNet has been shown to match the performance of ML detection on small-scale MIMO systems, such as 4-transmit by 4-receive QPSK configurations, while maintaining manageable computational demands. However, its effectiveness deteriorates in scenarios involving highly ill-conditioned channels or when the channel state information (CSI) is inaccurate.

Convolutional neural networks (CNNs) were later introduced to enhance detection by exploiting

local spatial patterns within the received signal. CNNs are naturally adept at identifying localized features, which makes them effective in handling distortions caused by noise and fading in wireless channels [21]. For example, O'Shea et al. [13] proposed a one-dimensional CNN model that treats the received signal as a sequential series of real and imaginary components. Using dilated convolutional filters, the network was able to capture both short-range and long-range dependencies in the signal. This model achieved up to a 1 dB improvement in bit error rate (BER) performance over FCNN-based models in Rayleigh fading environments. However, it required careful tuning of filter sizes and dilation factors.

Recurrent Neural Networks (RNNs) and Long Short-term Memory (LSTM) networks were also explored for their ability to model sequential dependencies in symbol streams [28], but their inherently sequential updates limit parallelism and increase inference latency in high-throughput systems. To address long sequence lengths efficiently, the Dual-Path RNN (DPRNN) [29] introduced a two-stage scheme: (1) *intra-chunk* RNNs process short blocks of symbols in parallel, capturing fine-grained interference patterns, and (2) *inter-chunk* RNNs model dependencies across blocks, yielding a scalable architecture that outperforms vanilla RNNs by 1–2 dB in BER on 8×8 16-QAM channels, with only a modest increase in parameter count.

Building on the above-mentioned ideas, residual networks (ResNets) introduce skip connections that help preserve essential low-level features and enable deeper network training, leading to enhanced performance at higher modulation orders [23, 30]. More recently, Transformer architectures—free of recurrence—have been adapted for MIMO detection. For instance, the paper "Transformer Learning-Based Efficient MIMO Detection Method" by Burera et al. [31] proposes a Transformer-based model that enhances bit error rate (BER) performance while maintaining computational efficiency. The Self-Attention Detector (SAD) [32] uses multi-head attention to directly model interactions among all received dimensions, achieving near-ML performance on 64QAM with a 10% reduction in FLOPs compared to DPRNN. The Permutation-Invariant Transformer PIT further incorporates a loss, enabling joint detection and user-pairing in overloaded scenarios ($n_T > n_R$), and demonstrates robustness to channel estimation errors up to 10% variance [33].

Although the above methods operate on time-domain waveforms, alternative approaches leverage the short-time Fourier transform (STFT) to decouple spatial and spectral processing. TF-GridNet [34] employs a 2D dual-path architecture: one path scans frequency bins within each time frame, the other scans time frames within each frequency, and cross-connections fuse the two. On 2×2 16-QAM channels, it achieves a 0.5 dB BER improvement over SAD at high SNR, at the cost of a 15% larger model. TFPSNet [35] extends this idea by introducing time-frequency path scanning blocks that alternately attend to time, frequency, and diagonal (time-frequency) directions via lightweight Transformers. This yields state-of-the-art BER on 64-QAM with only a 20% increase in runtime versus FCNNs.

Hybrid detectors that combine STFT preprocessing with deep networks have also been explored for MIMO channel equalization. For example, [36] applies a CNN to the magnitude spectrum and an RNN to the phase, demonstrating 1dB gain over pure time domain models in frequency-selective Rayleigh channels. These hybrid methods exploit the stationarity of multipath taps in the frequency domain but require careful phase reconstruction to recover the full complex signal. One commonly used technique for this task is the Griffin-Lim algorithm, an iterative phase estimation method that reconstructs a time-domain signal from a given magnitude spectrogram by alternating between the frequency and time domains [37]. While effective, Griffin-Lim can introduce artifacts due to its heuristic nature and non-convex convergence behavior, which may impact the quality of signal recovery in MIMO detection systems.

Practical 5G systems demand MIMO detectors that offer both high performance and low computational overhead. One recent development is S4M (State-Space for MIMO), which is not a conventional neural network architecture [38]. It is a signal processing framework which employs the use of state-space models (SSMs) to represent the channel impulse response as a linear dynamical system. By modeling the channel this way, S4M significantly reduces the computational complexity of detection, scaling linearly with the block length, while still maintaining competitive performance. For example, on long 256-tap channels, S4M achieves comparable bit error rate (BER) to Dual-Path RNN (DPRNN) models but with much lower complexity [39].

Building on this idea, Skipping Memory (SkiM) [40], a neural network architecture designed

specifically for efficient sequence modeling in tasks like MIMO detection, further reduces processing cost by selectively skipping over less informative parts of the input sequence and focusing computation on more relevant time segments. This technique leads to around a 30% reduction in floating point operations (FLOPs), with minimal loss in detection accuracy.

Meanwhile, RE-DetNet introduces a more compact architecture by summarizing global context into a fixed-length latent representation, which is then fed into each layer of the network. This design drastically reduces the number of trainable parameters compared to the original DetNet architecture, making it a good choice for hardware-efficient deployment in real-time 5G systems [41, 42].

Finally, model compression techniques (pruning, quantization, knowledge distillation) have been successfully applied to these architectures, enabling real-time deployment on hardware platforms such as *Field-Programmable Gate Arrays (FPGAs)* and *Application-Specific Integrated Circuits (ASICs)*. FPGAs offer reconfigurable hardware logic, allowing flexible prototyping and rapid adaptation to varying detection algorithms, while ASICs provide highly optimized, fixed-function circuits ideal for large-scale, power-efficient deployment. These hardware accelerators enable sub-5 ms inference latency per MIMO symbol, making them suitable for integration into practical 5G baseband processing pipelines [43].

Despite these promising developments, many existing studies focus predominantly on tasks such as channel estimation or resource allocation, or they rely on pre-trained models that offer limited customization for specific MIMO detection scenarios. Moreover, while deep learning models have demonstrated near-optimal performance, there is a lack of comprehensive evaluation that compares these models with traditional detection techniques and sphere decoding (SD) algorithms. SD methods, which incorporate techniques such as MMSE initialization and Schnorr–Euchner ordering, serve as powerful near-ML benchmarks, but suffer from high computational complexity, limiting their real-time applicability [10].

These observations highlight several key gaps: the need for detector architectures that balance performance with computational feasibility, the lack of benchmark comparisons between deep learning and classical methods under standardized conditions, and the limited generalization of

models trained on simplified datasets. Moreover, practical issues such as inference latency, hardware constraints, and the robustness of detection under non-ideal channels (e.g., urban fading profiles or nonlinear transmitter effects) remain critical but insufficiently addressed.

Motivated by these gaps, this thesis develops a unified simulation framework to evaluate 5G MIMO detection methods. Within this framework, various neural architectures, including FCNN, CNN, ResNet, and LSTM, are trained from scratch using a custom dataset that incorporates realistic signal impairments. Their performance is systematically benchmarked against traditional linear detectors and SD, with a focus on accuracy, complexity, and latency. In doing so, this work contributes to to bridge the gap between theoretical advancements and practical implementation of the 5G receiver.

1.3 Objective and Organization of the Thesis

The primary objective of this thesis is to develop and evaluate resource-efficient deep learning architectures for 5G MIMO detection that can approach near-optimal performance while significantly reducing computational complexity compared to conventional methods. To address the challenges posed by increased antenna counts and high-order modulation schemes in modern 5G networks, the proposed work introduces neural network models, including fully connected neural networks (FCNN), convolutional neural networks (CNN), residual networks (ResNet) and Long Short-term Memory Network (LSTM), developed from scratch using a custom generated dataset. This data set simulates realistic 5G channel conditions, encompassing both additive white Gaussian noise (AWGN) and Rayleigh fading with temporal correlation modeled through an autoregressive process (AR (1)). Additionally, a sphere decoding (SD) algorithm is incorporated as a benchmark to provide a near-maximum likelihood (ML) performance reference.

The remainder of this thesis is organized as follows.

Chapter 2: Overview of 5G MIMO Communications

This chapter details the system model and channel characteristics used in this research. It describes the MIMO signal model, the incorporation of temporal channel correlation via an AR(1) process, [44], and the custom data generation process using MATLAB's 5G Toolbox. Additionally,

traditional detectors, including Maximum Likelihood (ML), Sphere Decoding (SD), and Minimum Mean Square Error (MMSE), were also discussed in detail.

Chapter 3: Deep Learning Based Detection for 5G MIMO Systems

This chapter presents the deep learning approaches developed in this work. Provides an in-depth description of neural network architectures, including fully connected neural network (FCNN), convolutional neural network (CNN), residual network (ResNet), and long short-term memory (LSTM). Detailed discussions include network design, activation functions, normalization techniques, loss functions, and training protocols, along with block diagrams that illustrate each architecture.

Chapter 4: Experimental Results

This chapter describes the simulation setup and presents experimental results obtained through extensive Monte Carlo simulations. The performance of conventional detection methods, sphere decoding, and deep learning–based detectors is compared in terms of BER, SER, along with their computational cost. A detailed cost analysis quantifies the trade-offs between detection accuracy and resource efficiency, providing insight for practical 5G receiver design.

Chapter 5: Conclusions and Future Work

The final chapter summarizes the key contributions of this thesis, discusses the practical implications for 5G MIMO receiver design, and outlines potential directions for future research, including the exploration of non-linear channel effects and hybrid detection strategies.

Chapter 2

Overview of 5G MIMO Communications

This chapter provides a comprehensive analysis of 5G MIMO networks considered in this work. It begins with a review of the fundamental concepts underpinning 5G MIMO systems, followed by an in-depth discussion of channel modeling techniques used in modern wireless communications. Finally, the chapter describes the quantitative analysis of 5G networks, which forms the basis for evaluating detection methods in subsequent chapters.

2.1 Fundamentals of 5G MIMO Systems

The transition from single input, single output (SISO) to multiple input, multiple output (MIMO) communication marked a paradigm change in wireless system design. In a SISO link, capacity is fundamentally limited by the Shannon–Hartley bound for a single transmit and receive antenna pair. In contrast, Foschini and Gans demonstrated in 1998 that, in rich scattering environments, equipping both transmitter and receiver with n_T and n_R antennas, respectively, can increase channel capacity approximately linearly in $\min\{n_T, n_R\}$ [15]. Telatar later provided a rigorous information-theoretic analysis of the ergodic capacity of i.i.d. Rayleigh fading MIMO channels, showing that the average spectral efficiency grows without bound as the number of antennas increases [16]. Concurrently, it was recognized that MIMO systems must balance two competing objectives: diversity gain, which

improves link reliability by sending redundant copies of the same data over independent fading paths, and spatial multiplexing gain, which increases throughput by transmitting independent data streams in parallel. Zheng and Tse formalized this trade-off in their indent [45] on the diversity–multiplexing trade-off, proving that for any fixed channel coherence time and SNR regime, one must choose between maximizing reliability (diversity) or maximizing rate (multiplexing), but cannot fully achieve both simultaneously.

In the context of fifth-generation (5G) New Radio (NR), multiple-input multiple-output (MIMO) technology plays a central role in meeting the demands of high data rates, enhanced reliability, and improved spectral efficiency. By employing multiple antennas at both the transmitter and receiver ends, MIMO systems enable spatial multiplexing and diversity gains. These systems use high-order modulation schemes—such as 16-QAM and 64-QAM—to transmit multiple bits per symbol, thus increasing the overall throughput.

The performance of 5G MIMO systems is typically evaluated in terms of bit error rate (BER) and symbol error rate (SER), which serve as key metrics for receiver design. However, as the number of antennas and modulation orders increase, the complexity of signal detection escalates, posing significant challenges in both computational complexity and real-time processing.

2.1.1 OFDM Signal Generation

OFDM divides a wide transmission bandwidth into a large number of narrowband subcarriers, each modulated by a low-rate data stream. In 5G NR, the basic OFDM transmitter chain is as follows:

(1) **Serial-to-Parallel Conversion.** An incoming bitstream is mapped to complex QAM symbols (e.g. 16-QAM, 64-QAM), then partitioned into blocks of N_{sc} symbols.

(2) **IFFT.** Each block is passed through an N_{FFT}-point inverse fast Fourier transform (IFFT), yielding the time-domain OFDM symbol as expressed by:

$$s[n] = \frac{1}{\sqrt{N_{\mathrm{FFT}}}} \sum_{k=0}^{N_{\mathrm{FFT}}-1} X[k]\, e^{j2\pi kn/N_{\mathrm{FFT}}}, \quad n = 0, \ldots, N_{\mathrm{FFT}} - 1, \tag{1}$$

Here, $X[k]$ denotes the complex modulation symbol assigned to the k^{th} subcarrier in the frequency domain. These symbols are typically drawn from a QAM constellation (e.g., 16-QAM, 64-QAM), and may include pilot tones or zero-padding for unused subcarriers, depending on the system configuration.

In eq1 N_{FFT} is the size of the IFFT and determines the number of subcarriers in the OFDM symbol. It is typically chosen to be a power of two (e.g., 64, 128, 256, etc.) to enable efficient implementation using radix-2 fast Fourier transform (FFT) algorithms. This choice significantly reduces computational complexity and is widely adopted in practical systems such as 5G NR and LTE.

(3) **Cyclic Prefix (CP) Insertion.** To mitigate inter-symbol interference in multipath channels, a cyclic prefix of length N_{CP} (typically 4–7% of N_{FFT}) is prepended:

$$\tilde{s}[n] = \begin{cases} s[n + N_{\text{FFT}} - N_{\text{CP}}], & n = 0, \ldots, N_{\text{CP}} - 1, \\ s[n - N_{\text{CP}}], & n = N_{\text{CP}}, \ldots, N_{\text{CP}} + N_{\text{FFT}} - 1, \end{cases} \tag{2}$$

(4) **Digital-to-Analog Conversion and RF Up-Conversion.** The CP-extended waveform is filtered, converted to analog, and up-converted to the carrier frequency.

The 5G NR introduces a family of *numerologies*—subcarrier spacings $\Delta f = 2^{\mu} \times 15\,\text{kHz}$ ($\mu = 0, \ldots, 4$)—to flexibly trade off latency, Doppler resilience, and spectral efficiency [46]. Each numerology defines the OFDM symbol duration $T_s = 1/\Delta f$, CP length, and the number of symbols per slot.

2.1.2 Baseband Representation

A narrowband MIMO link with n_T transmit and n_R receive antennas can be described in the discrete-time complex baseband by the well-known linear model

$$\mathbf{y} = \mathbf{H}\,\mathbf{x} + \mathbf{n}, \tag{3}$$

where $\mathbf{x} \in \mathbb{C}^{n_T \times 1}$ is the transmitted symbol vector, drawn from a modulation constellation $\mathcal{S}$ (e.g. 4-QAM, 16-QAM) and thus limited to $\mathbf{x} \in \mathcal{S}^{n_T}$. The matrix $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ represents the channel matrix whose (i, j)–th entry $h_{i,j}$ denotes the complex gain from transmit antenna j to receive antenna i. The vector $\mathbf{n} \in \mathbb{C}^{n_R \times 1}$ represents the additive noise observed at the receive antennas, typically modeled as circularly-symmetric complex Gaussian (AWGN) with

$$\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_R}) \tag{4}$$

The vector $\mathbf{y} \in \mathbb{C}^{n_R \times 1}$ represents the received signal at the baseband sampler.

Equation (3) succinctly captures the spatial multiplexing or diversity offered by a MIMO link [15, 16]. In a $n_T \times n_R$ system, up to $\min(n_T, n_R)$ independent data streams can be transmitted in parallel, subject to the rank of $\mathbf{H}$. Note that the time-domain OFDM signal $s[n]$ generated via IFFT in (1) is transmitted over the MIMO channel, where its per-subcarrier representation contributes to the input $\mathbf{x}$ in the baseband MIMO model.

Signal Power and SNR. We assume the total transmit power is normalized as

$$\mathbb{E}\left[\mathbf{x}^H \mathbf{x}\right] = P_T \tag{5}$$

and the noise variance is σ^2. The average receive signal-to-noise ratio (SNR) per receive antenna is given by

$$\text{SNR} = \frac{P_T}{\sigma^2} \tag{6}$$

When comparing different modulation orders (e.g. 16-QAM), we often refer to $\frac{E_b}{N_0}$ as the bit-energy-to-noise-density ratio, which relates to SNR by

$$\text{SNR} = \frac{E_b}{N_0} \log_2 |\mathcal{S}| \tag{7}$$

where $|\mathcal{S}|$ is the constellation size.

2.1.3 5G NR Numerology and Resource Grid

The 5G NR system arranges time–frequency resources into a two-dimensional grid of resource elements (REs). Each RE corresponds to one subcarrier k and OFDM symbol index ℓ. A contiguous block of 12 subcarriers over one slot (14 symbols) forms a *resource block* (RB). Numerology μ determines:

- Subcarrier spacing $\Delta f = 15 \times 2^{\mu}\,\text{kHz}$, where $\mu \in \{0, 1, 2, 3, 4\}$ is the numerology index,
- OFDM symbol duration $T_s = 1/\Delta f$,
- Slot duration $T_{\text{slot}} = 14T_s + 14T_{\text{CP}}$.

As shown in Figure 2.1, each resource block spans 12 sub-carriers and 14 OFDM symbols, representing a basic time–frequency scheduling unit in 5G NR [1]. MIMO transmission maps encoded bits onto modulation layers and then onto REs via *precoding*—for example, eigenbeamforming or codebook-based mapping—thus exploiting spatial multiplexing and diversity across the grid [47, 48].

2.1.4 Channel State Information and Estimation

Accurate knowledge of $\mathbf{H}$ at the receiver—so-called channel state information (CSI)—is critical for coherent detection and decoding in MIMO systems [49]. In practice, $\mathbf{H}$ is estimated using pilot or training symbols.

Pilot-Assisted Estimation. A common approach to channel estimation is to prepend each data frame with N_p known pilot symbols, denoted by $\{\mathbf{x}_p[n]\}_{n=1}^{N_p}$. The receiver observes the corresponding received signals

$$\mathbf{y}_p[n] = \mathbf{H}\,\mathbf{x}_p[n] + \mathbf{n}_p[n], \quad n = 1, \ldots, N_p, \tag{8}$$

where $\mathbf{y}_p[t] \in \mathbb{C}^{n_R \times 1}$ is the received pilot signal, $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ is the unknown channel matrix, $\mathbf{x}_p[t] \in \mathbb{C}^{n_T \times 1}$ is the known pilot symbol vector, and $\mathbf{n}_p[t] \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ is complex Gaussian noise.

Denoting the received signals as a matrix $\mathbf{Y}_p = [\,\mathbf{y}_p[1] \;\cdots\; \mathbf{y}_p[N_p]\,] \in \mathbb{C}^{n_R \times N_p}$ and the pilot symbols as a matrix $\mathbf{X}_p = [\,\mathbf{x}_p[1] \;\cdots\; \mathbf{x}_p[N_p]\,] \in \mathbb{C}^{n_T \times N_p}$, the least-squares (LS) estimate of the channel is given by

$$\widehat{\mathbf{H}}_{\mathrm{LS}} = \mathbf{Y}_p\,\mathbf{X}_p^{\dagger} = \mathbf{Y}_p\,\mathbf{X}_p^{H}\big(\mathbf{X}_p\mathbf{X}_p^{H}\big)^{-1}, \tag{9}$$

where $\mathbf{X}_p^{\dagger}$ denotes the Moore–Penrose pseudoinverse of $\mathbf{X}_p$. This estimate is valid provided that $\mathbf{X}_p$ has full row rank. The mean-squared error of $\widehat{\mathbf{H}}_{\mathrm{LS}}$ depends on the noise variance σ^2 and the design of the pilot matrix [20]. More sophisticated estimators, such as minimum mean square error (MMSE) and subspace-based methods, can exploit prior statistical knowledge of the channel [50].

Perfect vs. Imperfect CSI. In theoretical analysis, one often assumes *perfect* CSI ($\widehat{\mathbf{H}} = \mathbf{H}$) to derive performance bounds. In practice, estimation error $\Delta\mathbf{H} = \widehat{\mathbf{H}} - \mathbf{H}$ degrades detection performance, particularly at high SNR where residual interference becomes dominant. We will examine both ideal and realistic CSI scenarios in our simulation framework.

Impact on Detection. All subsequent detector structures, whether linear (ZF/MMSE), maximum-likelihood, or deep learning based, rely on the estimated channel matrix $\hat{\mathbf{H}}$. However, in practice, $\hat{\mathbf{H}}$ rarely matches the true channel $\mathbf{H}$ exactly due to factors such as noise, limited pilot symbols, or model mismatch. These imperfections in estimation, whether due to inaccurate amplitude, phase, or structural assumptions, can lead to degraded detection performance. In particular, if the magnitude of channel gains is underestimated, detectors may insufficiently suppress interference or noise, leading to residual multi-user interference and an actual loss in signal-to-noise ratio (SNR). We will revisit the impact of such imperfect channel state information (CSI) in the experiments presented in Chapter 4.

With the baseband signal model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$ and an understanding of how $\mathbf{H}$ is obtained in this section, we introduce several channel models commonly used in linear detectors.

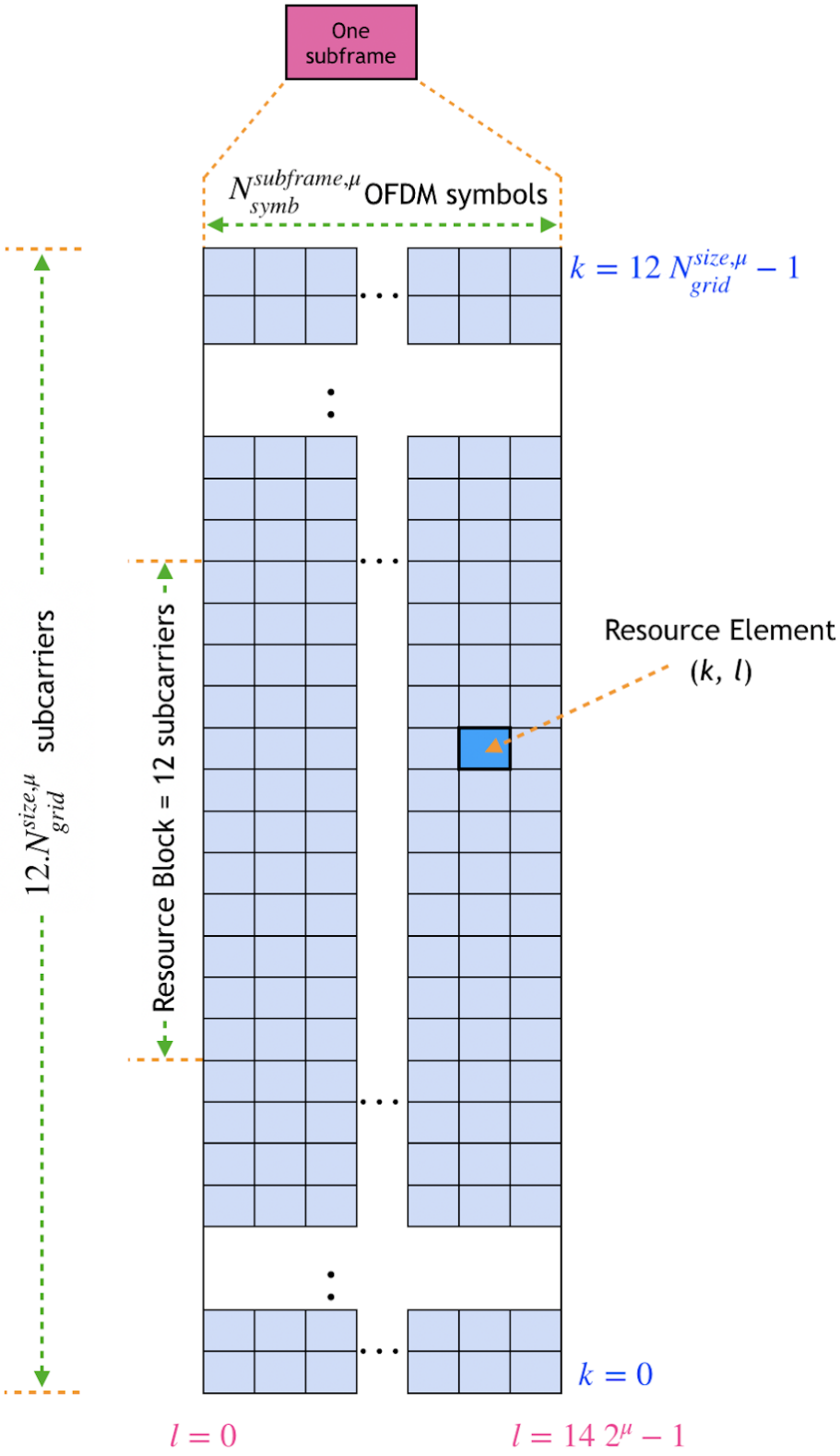

Figure 2.1: 5G NR OFDM time–frequency resource grid [1]

2.2 Channel Models

2.2.1 AWGN Channel Model

The AWGN model is the baseline by introducing thermal noise with a constant power spectral density. In this model, noise is modeled as a zero-mean complex Gaussian random variable. The signal-to-noise ratio (SNR) is given by:

$$\text{SNR} = \frac{E_s}{\sigma^2} \tag{10}$$

where E_s is the average symbol energy. This model is essential for establishing the noise floor in detection performance evaluations [7].

2.2.2 Rayleigh Fading Model

In the absence of a dominant line-of-sight (LOS) path, multipath propagation causes channel gains to fluctuate according to a Rayleigh distribution [51]. Each element h_{ij} of the channel matrix $\mathbf{H}$ is modeled as:

$$h_{ij} \sim \mathcal{CN}\left(0, \frac{1}{n_T}\right) \tag{11}$$

ensuring that the average received power is normalized. This model captures the rapid fluctuations in signal amplitude and phase caused by scattering, reflection, and diffraction.

2.2.3 Temporal Correlation and AR(1) Process

Real-world channels exhibit temporal correlation because the environment changes smoothly relative to the symbol rate [51]. To incorporate this, the channel is modeled using an autoregressive process of order one (AR(1)), which updates the channel matrix for each time block as follows:

$$\mathbf{H}_{\text{new}} = \alpha\, \mathbf{H}_{\text{prev}} + \sqrt{1-\alpha^2}\, \mathbf{H}_{\text{rand}} \tag{12}$$

where α (typically set to 0.9) is the correlation coefficient, $\mathbf{H}_{\text{prev}}$ is the previous channel matrix, and $\mathbf{H}_{\text{rand}}$ is a newly generated Rayleigh fading matrix. This formulation ensures that successive

channel realizations remain correlated, thereby more accurately modeling the temporal dynamics observed in 5G environments.

2.2.4 3GPP TDL-C Channel Model for Urban Environments

To capture realistic urban multipath propagation effects in 5G systems, this work incorporates the standardized Tapped Delay Line Channel-C (TDL-C) model defined by 3GPP in TR 38.901 [46]. The TDL-C profile represents typical urban macrocell environments characterized by moderate delay spreads and rich multipath structure. It consists of multiple discrete taps, each with a predefined relative delay and power level, designed to mimic empirically observed propagation characteristics in city deployments. In our implementation, the TDL-C channel is instantiated using MATLAB's `nrTDLChannel` object, configured with a 30-nanosecond RMS delay spread and a static Doppler shift, simulating quasi-stationary receivers. The antenna setup can be flexibly adjusted, although most experiments in this work use a 1×1 SISO configuration for clarity. The channel impulse response is modeled as a tapped-delay line with time-varying complex gains applied to each delay path. Mathematically, it can be represented as:

$$h(t, \tau) = \sum_{l=1}^{L} a_l(t)\, \delta(\tau - \tau_l), \tag{13}$$

where $a_l(t)$ and τ_l denote the complex gain and relative delay of the l-th path, respectively. The TDL-C model introduces significant frequency-selectivity across OFDM subcarriers, making it ideal for evaluating equalization strategies and neural network–based detectors under realistic, non-ideal channel conditions. In this thesis, the TDL-C model is used in selected simulation scenarios to assess how well different architectures perform in the presence of channel memory and multipath fading.

2.2.5 Path Loss and Fading Coefficients

In addition to small-scale fading, the received signal power is affected by large-scale fading phenomena such as path loss and shadowing. The path loss $PL(d)$ is typically modeled as:

$$PL(d) = PL(d_0)\left(\frac{d}{d_0}\right)^{-\gamma} \tag{14}$$

where $PL(d_0)$ is the path loss at a reference distance d_0, d is the separation between the transmitter and receiver, and γ is the path loss exponent, which depends on the propagation environment [52]. Although the primary focus of this work is on small-scale fading and noise, it is important to note that these large-scale factors are implicitly managed through normalization of the channel matrix to maintain unit average power at the receiver, where "unit average power" refers to scaling the channel such that the expected power of the received signal, $\mathbb{E}[\|\mathbf{Hx}\|^2]$, equals the average symbol energy E_s. This ensures consistency with the SNR definitions used in subsequent analysis.

2.2.6 Integration into the Simulation Framework

The overall channel model employed in this work combines the effects of AWGN and Rayleigh fading with temporal correlation. For each simulation run, the transmitted signal is passed through a channel represented by $\mathbf{H}$ (with Rayleigh fading statistics) that is modified using the AR(1) process. AWGN is then added to the output to simulate realistic degradation of the signal. This composite model enables the evaluation of detection methods under conditions that closely resemble practical 5G MIMO scenarios.

2.3 Traditional MIMO Detection Techniques

To establish a performance benchmark for our proposed deep learning–based MIMO detectors, we implemented and evaluated three traditional MIMO detection techniques including Maximum Likelihood (ML), Sphere Decoding (SD), and Minimum Mean Square Error (MMSE). These methods are widely recognized in the literature for their roles in optimal, near-optimal, and linear detection, respectively [53, 54].

2.3.1 Maximum Likelihood (ML) Detection

ML detection aims to minimize the Euclidean distance between the received signal vector and all possible transmitted symbol vectors. The decision rule is given by:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{X}^{N_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{15}$$

where $\mathbf{y}$ is the received signal, $\mathbf{H}$ is the channel matrix, $\mathcal{X}$ is the modulation alphabet, and N_t is the number of transmit antennas. Although ML provides optimal detection in terms of Bit Error Rate (BER), its complexity grows exponentially with modulation order and number of transmit antennas, making it impractical for large MIMO configurations [53].

2.3.2 Sphere Decoding (SD)

To address the computational burden of ML, we implemented Sphere Decoding as a near-optimal alternative. SD searches for the transmitted vector within a hypersphere of radius r, effectively pruning the search space:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{X}^{N_t},\, \|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2 \leq r^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \tag{16}$$

In our implementation, a fixed radius of $r = 2$ was chosen for a balance between complexity and performance, as adopted in previous works [54].

2.3.3 Minimum Mean Square Error (MMSE) Detection

For scenarios involving linear receivers and channel memory, we adopted MMSE equalization. In the case of MIMO-OFDM under 3GPP TDL-C fading [46], we estimated the channel per subcarrier and applied MMSE equalization as:

$$\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y}, \tag{17}$$

where $\mathbf{y}$ is the received signal vector at the receiver antennas, $\mathbf{H}$ is the estimated MIMO channel matrix, σ_n^2 is the noise variance, and $\mathbf{I}$ is the identity matrix. This formulation is widely used in MIMO-OFDM systems, as it minimizes the mean squared error on each subcarrier individually under the assumption of flat-fading and perfect channel state information (CSI) within subcarrier bandwidth [49].

All traditional detectors were implemented using MATLAB's 5G Toolbox and evaluated consistently with the deep learning methods under identical simulation conditions, including OFDM framing, pilot insertion, and Rayleigh or TDL-C fading channels. This allows for a rigorous performance comparison in terms of BER, complexity, and latency.

2.4 Summary

This chapter has introduced the foundational concepts of 5G MIMO systems, some commonly used detectors in 5G networks, and several channel models. By employing both AWGN and temporally correlated Rayleigh fading models, and by generating a dataset that accurately captures the raw complex signal characteristics and effective SNR, the groundwork is laid for a rigorous performance evaluation of various detection techniques. The next chapter presents the methodology for the deep learning–based detection approaches developed in this work. It provides a detailed description of the custom-designed neural network architectures—namely, the fully connected neural network (FCNN), the convolutional neural network (CNN), and the residual network (ResNet)—including their layer configurations, activation functions, training procedures, and data preprocessing strategies.

Chapter 3

Deep Learning–Based Detection for 5G MIMO Systems

3.1 Neural Network Architectures

The proposed deep learning framework comprises three distinct architectures: a fully connected neural network (FCNN), a convolutional neural network (CNN), and a residual network (ResNet). These networks are trained to perform bit-level detection, with the resulting output bits reassembled into symbols for bit error rate (BER) and symbol error rate (SER) evaluation. Unlike many existing studies that rely on pre-trained models, the networks in this work are custom-designed to capture the unique spatial and temporal features inherent in 5G MIMO signals. In particular, the FCNN model employs a straightforward sequence of dense layers to learn global features, the CNN model leverages convolutional filters to extract local spatial correlations from the real and imaginary parts of the received signal. Building on the strengths of CNNs, the ResNet architecture introduces skip connections to facilitate deeper network training by preserving low-level signal features and mitigating the vanishing gradient problem. Here is a detailed description of the three architectures.

3.1.1 Fully Connected Neural Network (FCNN)

FCNN is a foundational deep learning model that serves as a baseline architecture for many classification tasks. In the context of 5G MIMO detection, the FCNN is employed to directly map

a carefully constructed feature vector to bit-level classification outputs. Its primary advantage lies in its ability to model complex nonlinear relationships through a series of affine transformations interleaved with nonlinear activation functions, thereby approximating the maximum likelihood detection without the burden of an exhaustive search.

In our implementation, the input feature vector is generated by decomposing the received signal $\mathbf{y}$ into its real and imaginary components, and by appending the effective signal-to-noise ratio (SNR) as an additional numeric feature. This formulation is based on the observation that both the amplitude and phase information of the received signal, along with an accurate estimate of channel quality, are crucial for reliable detection in MIMO systems. The input feature is defined as:

$$\text{Feature} = [\Re\{y\},\, \Im\{y\},\, \text{SNR}] \tag{18}$$

where $\Re\{y\}$ and $\Im\{y\}$ denote the real and imaginary parts of the received signal vector, respectively.

The FCNN itself is composed of a sequence of fully connected layers. Each layer performs an affine transformation followed by a nonlinear activation, which in our case is the Rectified Linear Unit (ReLU). This operation is mathematically described by:

$$f_{\text{FCNN}}(x) = \sigma\Big(W_L\, \sigma\big(W_{L-1}\, \cdots \sigma(W_1 x + b_1) \cdots + b_{L-1}\big) + b_L\Big) \tag{19}$$

where x represents the input feature vector, W_i and b_i are the weight matrix and bias vector for the ith layer, respectively, and $\sigma(\cdot)$ denotes the ReLU activation function. ReLU is chosen for its ability to introduce nonlinearity while mitigating the vanishing gradient problem, thus facilitating effective learning even in deeper networks [55].

To enhance generalization and prevent overfitting, dropout is applied within the network. Dropout randomly deactivates a fraction of neurons during training, ensuring that the network does not become overly reliant on any single feature and can robustly learn distributed representations.

At the output layer, a softmax activation is used to convert the network's final outputs into probability estimates for each bit detection. The resulting bit probabilities are then reassembled into symbol decisions for calculating the bit error rate (BER) and symbol error rate (SER). Figure 3.1 illustrates the overall FCNN architecture, depicting the progression from the input layer through

multiple hidden layers to the final softmax layer.

Table 3.1: FCNN layer-by-layer specification

Layer	Type	Input Dim	Output Dim	Activation	Dropout
0	Input	$2n_R + 1$	—	—	—
1	Dense	$2n_R + 1$	256	ReLU	0.20
2	Dense	256	128	ReLU	0.20
3	Dense	128	64	ReLU	0.20
4	Dense (softmax)	64	Qn_T	Softmax	—

Table 3.1 gives the layer-by-layer specification of the FCNN used as our first detector. It lists each dense layer's input and output dimensions, activation function, and dropout rate. These hyperparameters are selected to provide sufficient model capacity for bit-level inference while regularizing against overfitting on the 80/20 train/validation split.

This FCNN approach, despite its relative simplicity compared to more sophisticated architectures like CNNs and ResNets, provides a solid baseline for MIMO detection. Its capability to learn complex nonlinear mappings from the raw signal components makes it particularly well-suited for the challenges inherent in 5G MIMO detection.

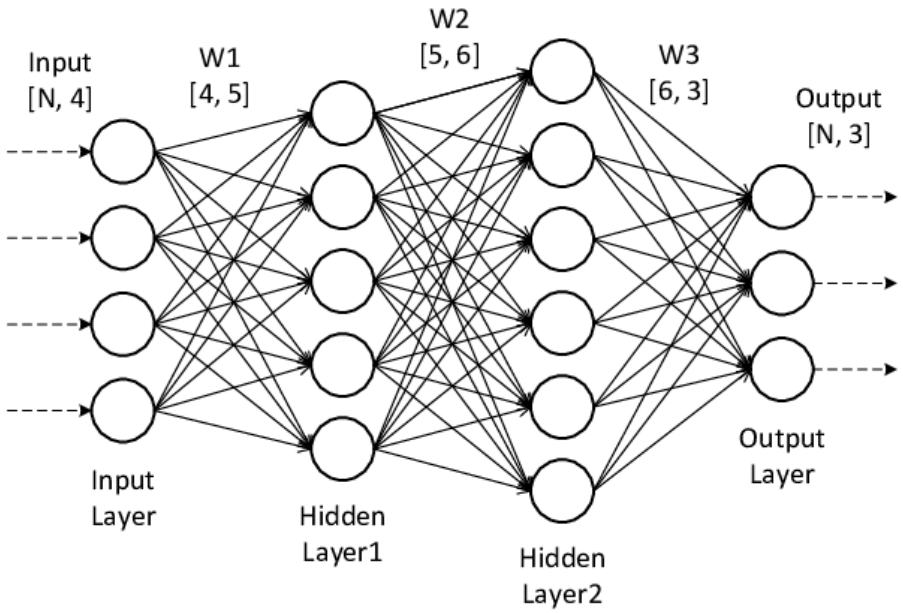

Figure 3.1: Block diagram of the FCNN architecture [2]

3.1.2 Convolutional Neural Network (CNN)

CNN architecture leverages the spatial structure of the received signal by treating its real and imaginary components as separate input channels. The input is represented as a tensor $\mathbf{X} \in \mathbb{R}^{H \times W \times 2}$, where H and W correspond to the spatial dimensions of the input (e.g., subcarriers and OFDM symbols), and the third dimension represents the two channels for the in-phase and quadrature (I/Q) components.

Each convolutional layer extracts local features by applying a set of trainable filters to the input tensor. This operation is expressed by the convolution function:

$$f_{\text{conv}}(\mathbf{X}) = \sigma(\mathbf{X} * \mathbf{W} + \mathbf{b}) \tag{20}$$

where $\mathbf{W} \in \mathbb{R}^{k_h \times k_w \times C_{\text{in}} \times C_{\text{out}}}$ is the filter tensor with $k_h \times k_w$ spatial dimensions, $\mathbf{b} \in \mathbb{R}^{C_{\text{out}}}$ is the bias vector, $*$ denotes the multi-channel convolution operation, and $\sigma(\cdot)$ is the element-wise activation function (e.g., ReLU). The output is a tensor $\mathbf{F} \in \mathbb{R}^{H' \times W' \times C_{\text{out}}}$, where H' and W' are determined by the convolution stride and padding.

Following convolution and activation, pooling layers are applied to reduce the spatial resolution of $\mathbf{F}$, lowering computational complexity while preserving the most significant features [27]. As illustrated in Figure 3.2, the CNN architecture consists of a feature extraction module (convolution, activation, pooling) followed by fully connected layers for detection. This model is developed and trained from scratch without the use of pretrained weights, ensuring full adaptation to the input signal characteristics.

Table 3.2: CNN layer-by-layer specification

Layer	Type	Kernel / Stride	# Filters	Output Shape	Activation
0	Reshape	—	—	$(2n_R + 1, 1)$	—
1	Conv1D	3 / 1	64	$(2n_R - 1, 64)$	ReLU
2	MaxPool1D	2	—	$((2n_R - 1)/2, 64)$	—
3	Conv1D	3 / 1	128	$(\ldots, 128)$	ReLU
4	GlobalAvgPool	—	—	$(128,)$	—
5	Dense (softmax)	—	—	$(Qn_T,)$	Softmax

Table 3.2 details CNN architecture evaluated in this work. Beginning with a reshape of the

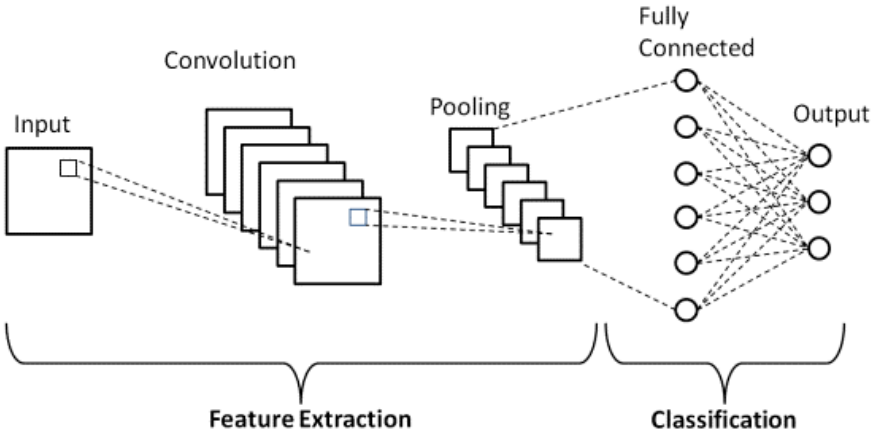

Figure 3.2: Block diagram of CNN architecture [3]

real/imaginary samples plus SNR, it applies two 1D convolutional blocks and a global average-pooling layer before the final softmax. Each row indicates kernel size, stride, filter count, output shape, and activation—highlighting how temporal correlations in the channel are captured.

3.1.3 Residual Network (ResNet)

The ResNet architecture extends the CNN model by introducing residual connections that allow the network to learn perturbations (residuals) around identity mappings instead of full transformations. This is particularly effective in deep networks, where gradient vanishing and degradation of feature representations become significant concerns [30].

Each residual block consists of two sequential convolutional layers with nonlinear activations (e.g., ReLU), followed by an element-wise addition between the original input $\mathbf{X}$ and the transformed output $\mathcal{F}(\mathbf{X})$. Mathematically, the residual block is defined as:

$$f_{\text{res}}(\mathbf{X}) = \sigma\left(\mathcal{F}(\mathbf{X}) + \mathbf{X}\right) \tag{21}$$

where $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ is the input tensor, $\mathcal{F}(\mathbf{X})$ denotes the residual mapping learned by the convolutional layers (i.e., $\mathcal{F}(\mathbf{X}) = f_{\text{conv}}(\mathbf{X})$), and $\sigma(\cdot)$ is the activation function (typically ReLU). Here, $f_{\text{conv}}(\cdot)$ refers to the same convolutional operation defined earlier in the CNN section, comprising a sequence of convolution, bias addition, and activation.

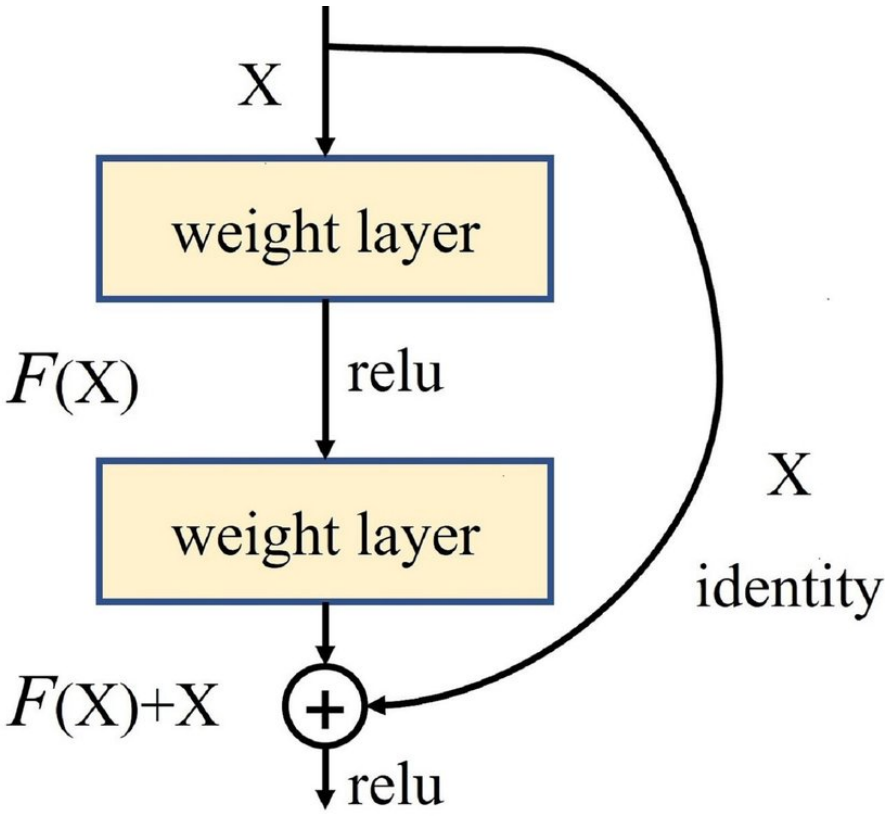

Figure 3.3: Block diagram of the ResNet architecture [4]

As illustrated in Figure 3.3, the identity connection (skip path) allows the unaltered input $\mathbf{X}$ to bypass the weight layers and be directly added to the output of $\mathcal{F}(\mathbf{X})$. This additive fusion preserves low-level features and ensures that gradients can flow more effectively through earlier layers during backpropagation.

The use of ResNet is particularly beneficial for tasks involving higher-order modulations, where capturing subtle channel impairments requires enhanced representational depth without degradation of information [56].

Table 3.3: ResNet layer-by-layer specification

Stage	**Block**	**# Params**	**Activation**	**Output Shape**
Input	Reshape $(2n_R + 1 \rightarrow (2n_R + 1, 1))$	—	—	$(2n_R + 1, 1)$
1	[Conv1D(64,3)→BN→ReLU]×2 + skip-add	~ 0.05M	ReLU	$(2n_R - 1, 64)$
2	[Conv1D(128,3)→BN→ReLU]×2 + skip-add	~ 0.15M	ReLU	$(\ldots, 128)$
3	GlobalAvgPool → Dense(Qn_T)	~ 0.22M	Softmax	$(Qn_T,)$

In Table 3.3 , the ResNet detector's stages are laid out block by block. For each stage, we show

the repeated Conv1D $\rightarrow$ batch norm $\rightarrow$ ReLU sequence, skip-add connections, parameter counts, and resulting output shapes. This clearly shows how the residual design facilitates deeper feature extraction without vanishing gradients.

Long Short-Term Memory Network (LSTM)

To capture temporal dependencies in the channel that arise due to time-dispersive fading (as modeled in the TDL-C profile), a recurrent neural network (RNN)–based architecture is incorporated. Specifically, a Long Short-Term Memory (LSTM) network is employed to address the shortcomings of feedforward models such as FCNN and CNN, which lack inherent mechanisms to model sequential patterns or memory effects in time-varying MIMO channels.

The LSTM architecture is designed to process the received signal as a sequence of complex-valued symbols, with the real and imaginary parts concatenated to form the input at each time step. Unlike traditional RNNs, LSTM networks utilize gated mechanisms that control information flow across time steps, thereby mitigating vanishing or exploding gradients during training. This allows them to retain long-term dependencies, which are essential when the channel exhibits memory due to multipath spread [28].

Mathematically, the LSTM cell operations at time step t are defined as:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t),
\end{aligned}
$$

where $\mathbf{x}_t$ is the input vector at time t, $\mathbf{h}_t$ is the hidden state, $\mathbf{c}_t$ is the cell state, and $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$ are the input, forget, and output gates, respectively. The symbols $\sigma(\cdot)$ and $\odot$ denote the sigmoid activation function and element-wise multiplication.

In our implementation, a single-layer LSTM processes sequences of received signals over time. The output from the LSTM is passed through a fully connected layer followed by a softmax classifier to produce bit-level probability estimates. This structure allows the network to learn temporal correlations across symbols affected by inter-symbol interference (ISI), a prominent feature of the TDL-C channel.

Figure 3.4 illustrates the structure of the LSTM-based cell, including the sequential input processing, internal memory state propagation, and final classification stage.

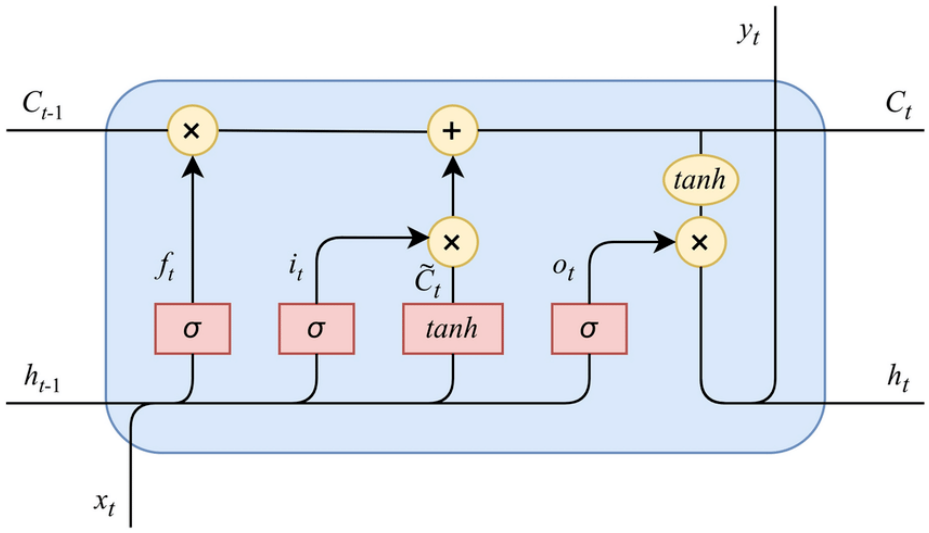

Figure 3.4: Structure of the LSTM-based cell used to process temporally correlated input signals [5]

3.1.4 Normalization, Loss Function, and Training Details

Each network is trained from scratch using a custom dataset that simulates realistic 5G conditions. The dataset is generated using MATLAB's 5G Toolbox, which models both AWGN and Rayleigh fading channels with temporal correlation (via an AR(1) process). The effective SNR is computed and appended to the raw complex received signal, thereby forming the input feature vector.

During training, a bit-level cross-entropy loss function is employed. The network parameters are optimized using the Adam optimizer with a carefully chosen learning rate, mini-batch size, and number of epochs. Unlike many approaches that fine-tune pretrained models, our networks are entirely custom-designed; this allows for precise control over model architecture and hyperparameters,

ensuring that the models are specifically tailored for the 5G MIMO detection problem. The normalization techniques applied to the input data (such as scaling and mean subtraction) are critical for ensuring convergence and robust performance under diverse channel conditions.

3.2 Deep Learning Modeling of Communication Systems

This section presents the detailed design and architecture of the neural networks (NNs) developed and utilized in this research for 5G MIMO detection tasks. A primary goal of this work is to leverage deep learning methods to effectively mitigate nonlinear effects encountered in practical wireless communication systems, specifically addressing transmitter nonlinearities such as amplifier saturation. The neural network models used include FCNNs, CNNs, and ResNets, each tailored to exploit distinct features within the received signal data.

The neural networks employed here are constructed and trained from scratch, using carefully curated datasets generated through MATLAB's 5G Toolbox, which provides realistic simulations of various channel conditions including Additive White Gaussian Noise (AWGN), Rayleigh fading, and temporal correlation via an autoregressive (AR(1)) process. Each network is designed to process input features consisting of real and imaginary parts of the received complex signals along with effective signal-to-noise ratio (SNR) values, directly performing bit-level detection tasks to determine the transmitted symbols.

To further enhance realism, we deliberately incorporated transmitter-induced nonlinearities into the signal chain, providing the neural networks the opportunity to learn robust signal detection under practical constraints. The subsequent subsections detail how this nonlinearity is introduced, modeled, and effectively handled within the neural network architecture.

3.2.1 Nonlinearity Injection into the Signal Chain

A critical innovation of this research involves deliberately introducing realistic nonlinear distortions into the transmitted signals to evaluate and enhance the neural networks' robustness under practical operating conditions. In real-world 5G systems, power amplifiers (PAs) at the transmitter inevitably operate near saturation points to maximize power efficiency, introducing significant

nonlinearities that distort transmitted symbols. These nonlinearities manifest in constellation warping, inter-symbol interference (ISI), and increased Bit Error Rate (BER), thereby complicating the detection task at the receiver.

In our simulation framework, we introduced nonlinearities directly into the transmitted symbols $x(t)$ prior to channel propagation:

$$\mathbf{y}(t) = \mathbf{H} \cdot f_{\text{NL}}(\mathbf{x}(t)) + \mathbf{n}(t) \tag{22}$$

where $f_{\text{NL}}(\cdot)$ represents the nonlinear distortion introduced by the transmitter's power amplifier (PA). The distorted signal then passes through the standard channel models before being fed into the neural network detector. By training on data that includes these nonlinear distortions, the neural network implicitly learns to compensate for nonlinear channel behavior, significantly improving performance compared to traditional linear detection methods.

3.2.2 Saturation Model and Parameter Choices

The transmitter nonlinearity employed in this research is modeled using a practical saturation model, widely adopted for power amplifiers operating in wireless communication systems. The nonlinearity is represented using a saturation/clipping function defined mathematically as follows:

$$f_{\text{NL}}(x) = \begin{cases} x, & |x| \leq A_{\text{sat}} \\ A_{\text{sat}} \cdot e^{j\angle x}, & |x| > A_{\text{sat}} \end{cases} \tag{23}$$

Here, A_{sat} denotes the saturation amplitude threshold, which is carefully selected to reflect typical operation points of practical RF amplifiers. The saturation level choice critically affects the degree of nonlinearity introduced into the signal. For our simulations, A_{sat} is chosen based on standard amplifier specifications and normalized with respect to the transmit power to reflect realistic operational scenarios encountered in actual 5G deployments.

An exhaustive parameter sweep is conducted to identify the most representative saturation threshold, ensuring the simulated scenarios are practically relevant. Lower values of A_{sat} lead to

aggressive clipping and severe signal distortion, whereas higher values approach linear transmission, providing minimal distortion. An intermediate value is selected to effectively balance these extremes, capturing realistic operational nonlinearity.

3.2.3 Activation-Function Adaptations for Power Amplifier Behavior

To further align the neural network's internal processing with realistic transmitter characteristics, the activation functions within the neural networks are deliberately adapted to mimic PA nonlinearity. Standard ReLU activations, while computationally efficient, exhibit linear characteristics beyond the activation threshold. In contrast, practical PA nonlinearities exhibit saturation characteristics that can be more accurately modeled with bounded activation functions such as hyperbolic tangent ($\tanh$) or sigmoid activations.

We explored replacing the conventional ReLU activation with bounded, nonlinear activation functions such as $\tanh$ or custom sigmoid-like activations that saturate at high inputs, thereby closely resembling the nonlinear behavior of saturated power amplifiers. Specifically, the $\tanh$ function defined by:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{24}$$

naturally introduces bounded saturation, smoothly compressing large values. Similarly, a logistic sigmoid defined by:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{25}$$

also provides nonlinear compression with a bounded output, making these functions particularly effective in capturing and correcting transmitter-induced nonlinear distortions during the neural network training phase.

By adopting such activation functions, the neural network more naturally internalizes the nonlinear characteristics of the transmitter PA, resulting in enhanced detection performance under conditions of significant signal distortion. Our experiments have shown that these adapted activations significantly outperform standard linear or ReLU-based models, providing robust and consistent performance even in severely saturated transmission scenarios.

3.3 Custom Dataset Generation and Preprocessing

In order to conduct an accurate and realistic assessment of various MIMO detection strategies, including traditional detection methods, sphere decoding (SD), and neural network-based approaches, a carefully structured dataset is generated for this research. While several prior studies have utilized publicly available datasets such as the Deep MIMO Dataset [57], 5G MIMO Data for Machine Learning [58], and Sionna-generated synthetic datasets [59], these datasets are typically limited in flexibility and often focus on specific channel types or hardware environments, such as beamforming in mmWave frequencies or real-time inference benchmarking. Moreover, many available datasets assume idealized conditions or do not provide detailed control over baseband impairments like transmitter nonlinearities, symbol-level fading, or temporal correlation in time-varying channels.

Furthermore, the specific objectives of this research demands application-specific input conditions, such as tightly controlled symbol alignment, temporally correlated fading, and nonlinear transmission artifacts, that are not readily available in existing open datasets. Using generic datasets compromised the accuracy and relevance of the results, as they lack the precise conditions required for evaluating the proposed detection framework. Hence, generating a tailored dataset is critical to ensure that the input distributions align with the operational assumptions of the detection models, ultimately yielding valid and trustworthy performance insights.

The custom dataset is developed using MATLAB's 5G Toolbox, enabling comprehensive simulation of realistic 5G MIMO scenarios. This dataset incorporates critical physical-layer effects such as additive white Gaussian noise (AWGN), Rayleigh fading, AR(1)-based temporal correlation, and power amplifier nonlinearities. The custom design ensures consistency across all detection schemes under test and allows fine-grained control of signal conditions and SNR levels. This level of detail is necessary to rigorously benchmark the performances of deep learning models that are sensitive to the statistical properties of their training data.

3.3.1 Data Generation Pipeline

The data generation pipeline is carefully designed to produce a realistic and comprehensive dataset aligned with 5G NR signal structures. It employs MATLAB's 5G Toolbox and standard-compliant parameters to generate OFDM-based waveforms transmitted through realistic 5G channels. The simulation reflects key 5G characteristics, such as standardized subcarrier spacing, cyclic prefix, and pilot-based channel estimation. The complete pipeline comprises the following stages:

(1) **Random Bit Generation:** Binary data streams are generated using a uniform random source. The total number of bits is adjusted to ensure an integer number of OFDM frames across modulation orders.

(2) **QAM Symbol Mapping:** The bitstream is mapped to complex QAM symbols using gray-coded modulation schemes (e.g., 4-QAM, 16-QAM, 64-QAM). All symbols are normalized to the unit average power.

(3) **5G NR Compliant OFDM Modulation:** The modulated symbols are organized into OFDM frames based on 5G NR numerology using a subcarrier spacing of 30 kHz and 52 resource blocks (10 MHz bandwidth). Each symbol is transformed by an inverse fast Fourier transform (IFFT) of size N_{FFT}, followed by the insertion of the cyclic prefix. Power normalization per symbol is also applied.

(4) **Pilot Insertion:** To enable accurate channel estimation at the receiver, a known OFDM pilot symbol is inserted at the beginning of each transmission frame. This pilot occupies all subcarriers and serves as a reference for estimating the channel frequency response across the entire bandwidth. In practical systems, such as those defined by the 3GPP 5G NR standard, pilot symbols (or reference signals) are periodically embedded in the time–frequency resource grid to facilitate coherent detection and equalization. In this simulation, the pilot-based channel estimation is essential for computing subcarrier-wise MMSE equalization weights, which require an estimate of the channel gain $H[k]$ on each subcarrier. This approach aligns with common pilot-aided estimation techniques used in practical OFDM systems [49].

(5) **5G Channel Modeling:** To simulate realistic wireless environments, the modulated signals

are transmitted through wireless fading channels generated using MATLAB's 5G-compliant models. Specifically, three distinct profiles are employed: additive white Gaussian noise (AWGN), Rayleigh fading with temporal correlation modeled via an AR(1) process, and the standardized 3GPP TDL-C channel for urban environments. These models are selected to span a range of realistic propagation conditions, from idealized noise-only scenarios to complex multipath fading. A detailed discussion of these channel models is already provided in Section 2.2.

(6) **Addition of AWGN:** Additive white Gaussian noise is introduced after the channel, with noise power scaled according to E_b/N_0, modulation order, and cyclic prefix overhead.

(7) **Dataset Assembly for Supervised Learning:** For each transmission instance, the received noisy signal is stored as the input feature tensor, while the original transmitted bits served as the label. These input–label pairs are compiled across SNR levels to build a rich dataset suitable for training and evaluating deep learning–based detection models.

This simulation setup adheres closely to 3GPP specifications and emulates a realistic link-level physical layer chain, thereby providing a reliable and reproducible environment for evaluating bit-level MIMO detection under 5G NR conditions.

3.3.2 SNR Normalization for 16-QAM

An important aspect of realistic dataset creation is ensuring proper normalization of the Signal-to-Noise Ratio (SNR). The transmitted symbol energy E_s is normalized to unity to maintain consistency across simulations, allowing straightforward interpretation and comparison of results.

Given the 16-QAM modulation scheme, the relationship between the bit-energy-to-noise-density ratio (E_b/N_0) and the symbol-level SNR is given by:

$$\mathrm{SNR}_{\mathrm{symbol}} = \frac{E_s}{N_0} = \frac{E_b}{N_0} \log_2(M) \tag{26}$$

where $M = 16$ represents the modulation order. This normalization ensures that each symbol in

the dataset consistently reflects the desired noise conditions, facilitating direct performance comparisons between detection methods at the same effective bit-level SNR.

Furthermore, data preprocessing includes standardizing each segment of dataset, to improve numerical stability and training convergence. Specifically, the real and imaginary components of the received signal vectors are normalized by subtracting their empirical mean and dividing by their standard deviation, both computed over the training dataset. This zero-mean, unit-variance normalization ensures that the neural network receives input features with consistent statistical properties, preventing issues related to activation saturation and accelerating convergence during gradient-based optimization.

3.3.3 Transmitter Nonlinearity (Clipping/Saturation)

In practical wireless communication systems, transmitter power amplifiers (PAs) typically operate near saturation regions to maximize power efficiency, leading to significant signal distortion known as nonlinearities. These nonlinear effects alter the transmitted signal constellation, causing spectral regrowth and signal degradation, which can substantially impact the receiver's performance.

To realistically emulate these effects, the generated dataset incorporated a controlled nonlinear saturation model. A saturation function is implemented at the transmitter output to simulate realistic PA behavior. Specifically, a hard-clipping nonlinearity model is adopted from equation 23

The saturated signals exhibited constellation warping and increased error vector magnitude (EVM), which directly contributed to increased bit and symbol error rates, thus presenting a challenging yet realistic scenario for evaluating receiver performance. The introduction of nonlinearity at the transmitter further emphasized the importance of developing robust detection strategies that can withstand and compensate for realistic impairments encountered in practical deployments.

In addition to observing the constellation deformation visually, quantitative metrics such as Adjacent Channel Power Ratio (ACPR) and Spectral Efficiency degradation are measured to provide deeper insights into the impacts of nonlinearity. These metrics indicated substantial degradation compared to linear transmission, confirming the severity of nonlinear distortions.

By incorporating transmitter nonlinearity into the data generation process, the resulting dataset facilitated the examination of advanced detection techniques under conditions closely aligned with

practical operational constraints. This rigorous and realistic scenario enhanced the validity of performance assessments, providing a stronger foundation for developing and validating neural network-based receivers that exhibit robustness against transmitter-induced impairments. Furthermore, analyzing performance degradation due to PA nonlinearities allowed for clearer identification of NN architectures capable of effectively mitigating these effects, highlighting their suitability for deployment in realistic 5G communication systems.

3.4 Training Strategy and Hyperparameter Tuning

Effective neural network performance for 5G MIMO detection depends on the selection of an appropriate training strategy and hyperparameter tuning. This section outlines the training strategies adopted for the neural networks used in this research, with a focus on loss functions, optimization algorithms, and regularization techniques.

3.4.1 Loss Functions and Optimizers

The choice of loss function directly influences the neural network's ability to learn an accurate decision boundary for bit-level classification tasks inherent in MIMO detection. In this work, we employed cross-entropy loss, which measures the difference between the predicted probability distribution of the bit classes and the actual bit labels, namely,

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{27}$$

where N is the number of training samples, C is the total number of output classes, $y_{i,c}$ is a binary indicator that equals 1 if the i-th sample belongs to class c, and 0 otherwise (i.e., one-hot encoding), and $\hat{y}_{i,c}$ represents the predicted probability assigned by the model to class c for the i-th sample. This formulation compares the true one-hot encoded class distribution with the model's predicted softmax probabilities, penalizing incorrect predictions based on their confidence.

To optimize the network parameters during training, the adaptive moment estimation (Adam) optimizer is employed [60]. Adam is a widely used first-order optimization algorithm that combines the benefits of two popular methods: momentum and RMSProp. It adaptively adjusts the learning

rate for each parameter by maintaining exponential moving averages of the gradients and their squared values.

The update rule for Adam is defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{28}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{29}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{30}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{31}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{32}$$

Here, g_t is the gradient of the loss with respect to the parameter θ_t at time step t, m_t and v_t are the first and second moment estimates, β_1 and β_2 are the exponential decay rates for the moment estimates (typically set to 0.9 and 0.999), and ϵ is a small constant (e.g., 10^{-8}) added for numerical stability. The quantities $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates used to compensate for initialization effects in early training stages.

The use of second-order moment estimation (v_t) corresponds to tracking the uncentered variance (i.e., squared error) of gradients, allowing the optimizer to scale the updates inversely proportional to the expected magnitude of parameter updates. This makes Adam particularly effective for non-stationary objectives and sparse gradients, which are common in deep learning models.

3.4.2 Regularization, Model Capacity and Parameter Scaling

Regularization techniques are essential to prevent overfitting, especially given the limited size of the datasets typically available for supervised learning in wireless communications scenarios. In this research, two primary regularization strategies are adopted:

Weight Decay (L_2 Regularization): This method penalizes the magnitude of network weights, effectively constraining the complexity of the learned decision boundaries. Mathematically, the L_2 regularized loss is expressed as:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{CE}} + \lambda \sum_{w \in \mathcal{W}} w^2, \tag{33}$$

where λ is the regularization hyperparameter and $\mathcal{W}$ denotes all the network weights.

Dropout: Dropout randomly removes neurons (and their connections) from the network during training with a certain probability. This technique reduces the risk of co-adaptation among neurons, thus improving generalization. In this work, dropout is applied after each dense layer with a dropout rate ranging from 0.2 to 0.5, determined empirically through cross-validation.

The capacity of neural networks (number of layers and neurons per layer) is carefully chosen based on preliminary experimental results. Overly large networks are prone to overfitting, whereas overly small networks fail to learn complex channel dynamics effectively. Through systematic experimentation, it is found that networks with 3 to 5 hidden layers, each containing between 128 and 256 neurons, provide the best trade-off between complexity and performance.

Additionally, proper parameter scaling and normalization are critical. Input feature normalization is performed using standard score normalization (Z-score normalization):

$$X_{\text{norm}} = \frac{X - \mu}{\sigma}, \tag{34}$$

where X is the original input data, μ is the mean, and σ is the standard deviation computed from the training set. This normalization significantly stabilized and accelerated network training, resulting in improved convergence speed and accuracy.

Through a comprehensive approach encompassing loss function selection, optimization strategy, regularization, and careful hyperparameter tuning, the developed neural network models achieved robust and accurate MIMO detection under realistic and challenging 5G conditions.

Table 3.4: Summary of key hyperparameters used in training the CNN detector, with rationale given for each choice.

Hyperparameter	**Value / Range**	**Rationale**
Random seed	12345	Ensures reproducibility across runs
Learning rate η	1×10^{-3}	Balanced convergence speed & stability
Optimizer	Adam	Robust to noisy gradients, adaptive moment estimation
Batch size	256	Fits GPU memory and provides stable gradient estimates
Epochs	20	Sufficient for convergence without overfitting
Conv layer filters	[32, 64]	Gradual increase in feature maps for richer representations
FC layer units	[128, 64]	Balances model capacity and computational cost
Dropout rates	[0.3, 0.2]	Mitigates overfitting in fully connected layers
Shuffle	every-epoch	Prevents model from memorizing data order
Training/validation split	80% / 20%	Ensures robust generalization assessment

Chapter 4

Experimental Results

4.1 Evaluation Metrics and Experimental Setup

A rigorous and reproducible evaluation procedure is essential for comparing various detection methods. This section details the error rate metrics, the signal to noise ratio (SNR) axis convention, and the computational-complexity measures adopted throughout the thesis. It then summarises the Monte-Carlo simulation framework used to generate all reported results.

4.1.1 BER/SER Computation and SNR Axis

Bit and Symbol Error Rates. For an $n_T \times n_R$ MIMO link employing an M-ary QAM constellation $\mathcal{S}$ of size $M = 2^Q$ (with Q bits per symbol), the instantaneous bit error rate is estimated as

$$\mathrm{BER} = \frac{1}{QN_{\mathrm{tot}}} \sum_{k=1}^{N_{\mathrm{tot}}} \left\| \mathbf{b}_k \oplus \hat{\mathbf{b}}_k \right\|_0, \tag{35}$$

where $\mathbf{b}_k$ and $\hat{\mathbf{b}}_k$ denote, respectively, the Q-bit labels of the transmitted and detected symbols in the k-th channel use, $\oplus$ is the XOR operator, and N_{tot} is the total number of symbols simulated.[1] When reporting *symbol* error rate (SER), the Hamming norm in the numerator of eq (35) is replaced by the binary indicator $1\left\{\mathbf{b}_k \neq \hat{\mathbf{b}}_k\right\}$ which equals 1 if any bit in the symbol is incorrect (i.e., the full symbol label is misclassified), and 0 otherwise [49, 61].

[1] All error-rate curves are averaged over at least 10^6 information bits per SNR point to guarantee statistical confidence.

SNR Definition. Throughout this thesis, the signal-to-noise ratio (SNR) is referenced to the *per-bit* energy, E_b/N_0, measured at the receiver input. This convention allows for consistent performance comparisons across different modulation schemes.

In digital modulation, the relationship between the bit energy E_b and symbol energy E_s depends on the modulation order $Q = \log_2 M$, where M is the constellation size. For instance, 16-QAM transmits 4 bits per symbol ($Q = 4$), meaning the effective symbol SNR is related to the bit-level SNR by:

$$\mathrm{SNR}_{\mathrm{sym}} = \frac{E_s}{N_0} = \frac{E_b}{N_0} \cdot \log_2 M. \tag{36}$$

This leads to the following commonly used approximation in dB scale:

$$\mathrm{SNR}_{\mathrm{sym}}\,(\mathrm{dB}) \approx \mathrm{SNR}_{E_b}\,(\mathrm{dB}) + 10\log_{10}(\log_2 M), \tag{37}$$

which assumes uniform bit distribution and no coding or shaping losses. This approximation is widely used for quick performance comparisons [7], but may not hold exactly in practical scenarios with nonlinear distortions or fading. [61]

For 16-QAM ($M = 16$), this corresponds to an SNR offset of approximately 6 dB. This convention ensures that BER and SER curves plotted in Chapters 4 are normalized to a common bit-energy baseline, allowing meaningful cross-modulation comparison. This relationship is standard in digital communication analysis [7].

It is important to note that while symbol energy and bit energy are related through $\log_2 M$, this does not imply a simple relationship between bit error rate (BER) and symbol error rate (SER). A single symbol error can lead to multiple bit errors, especially at higher modulation orders. Hence, BER and SER curves are presented separately throughout this thesis.

For Gray-coded constellations under AWGN, an approximate relationship is often used:

$$\mathrm{BER} \approx \frac{1}{\log_2 M} \cdot \mathrm{SER}, \tag{38}$$

This approximation assumes that, on average, each symbol error results in a single bit error. This assumption holds reasonably well for Gray-coded QAM in AWGN channels at moderate-to-high

SNRs. Note that this is in contrast to Equation (36), which is an exact relationship between symbol and bit energy. In contrast, Equation (38) is only an approximation and may become inaccurate under nonlinear transmitter distortion or fading channels, where symbol errors can involve multiple bit errors due to constellation warping or rotation.

Monte-Carlo Protocol. Each SNR point relies on an independent Monte-Carlo run comprising block-fading realizations of the 2×2 Rayleigh or AWGN channels detailed in Section 2.2. The matrix of fading coefficients is kept constant for the $L = 128$ symbols and regenerated thereafter. Channel memory is modeled through an AR(1) process with a coefficient $\alpha = 0.9$, matching real vehicular 5G scenarios.

4.1.2 Runtime per Detection Block and Complexity Metrics

Besides error-rate performance, implementation efficiency is a major criterion for designing a 5G modem. We therefore adopt the following quantitative indicators.

Number of Trainable Parameters (#Params). For neural detectors, this reflects memory footprint and model-download time. Classical algorithms (ZF/MMSE/SD) contain no trainable weights and are thus denoted "–" in the corresponding table columns.

Multiply–Accumulate Operations (MACs) and Spectral Efficiency (SE). The MAC count per detection block is analytically estimated based on the structure of each network. For traditional detectors, such as MMSE or ML, closed-form expressions derived from matrix inversion and multiplication are used to calculate computational cost. For neural networks, MACs are computed by summing the number of multiply–accumulate operations required by each layer—fully connected, convolutional, and residual—based on input size, kernel dimensions, and number of feature maps [49].

To ensure consistency in evaluation, one complex MAC is treated as four real MACs, aligning with standard digital signal processing conventions.

This metric provides an estimate of the computational burden associated with each detection method. However, high MAC counts do not necessarily indicate poor performance; rather, they

reflect a trade-off between complexity and detection accuracy.

In systems operating at high Spectral Efficiency (SE)—defined as the number of bits transmitted per second per Hz of bandwidth (bits/s/Hz)—accurate detection becomes more challenging due to denser constellations and tighter symbol packing. In such settings, advanced neural architectures may justify higher MAC counts if they significantly reduce the bit error rate (BER) under constrained latency or energy budgets. Thus, the interplay between MAC efficiency and SE is central to evaluating the practical viability of different MIMO detection strategies in 5G systems.

Complexity (GMAC/block). We count the complex multiply–accumulate operations per MIMO detection block. Table 4.1 reports the results for a representative 2×2 antenna configuration with 16-QAM and block length $L = 128$. A full-search ML detector evaluates all $16^2 = 256$ symbol pairs; the sphere decoder using Schnorr–Euchner (SE) enumeration with search radius r=2 prunes this search; and the FCNN/CNN/ResNet MACs follow directly from each layer's dimensions (1 complex MAC = 4 real MACs).

Latency Measurement. To evaluate the practical feasibility of the proposed neural network–based detection frameworks, end-to-end inference latency was measured using MATLAB's built-in timing utilities on a standardized computing environment. All latency-related simulations—including signal generation, channel modeling, neural network inference, and symbol demodulation were executed on a laboratory workstation running Ubuntu 20.04. The system was equipped with dual NVIDIA GeForce RTX 2080 GPUs (8GB each); however, all detection experiments were conducted on the CPU, without utilizing GPU acceleration or deep learning-specific profiling tools.

Latency was recorded using MATLAB's `tic` and `toc` functions, capturing the complete forward pass through the detection pipeline on a per-transmission-block basis. Each timing measurement reflects the cumulative delay associated with signal processing, from input generation to final symbol decision. To ensure statistical reliability, the reported latency values were averaged over multiple Monte Carlo runs.

This CPU-based benchmarking provides a conservative estimate of real-world inference latency,

Table 4.1: Complexity profile for the evaluated receivers (2×2, 16-QAM, block length $L = 128$).

Detector	**#Params**	**GMAC/block**	**Runtime [s]**	**Peak Mem [MB]**
ML (AWGN)	–	1.00	0.05	1.3
ML (Rayleigh)	–	1.00	0.06	1.3
SD (SE, $r = 2$)	–	0.47	2.50	1.3
FCNN (ours)	0.12M	0.09	0.30	5.6
CNN (ours)	0.36M	f0.17	0.30	7.3
ResNet (ours)	0.42M	0.22	0.30	7.9

independent of hardware acceleration, thereby offering a practical baseline for evaluating deployment feasibility in resource-constrained environments.

Memory Considerations. Since the experiments were conducted using MATLAB's default execution environment (CPU mode), detailed runtime memory profiling was not explicitly performed. However, the models were carefully designed to fit within the available system memory, with batch sizes and model sizes selected to avoid memory overflows or performance bottlenecks. For future work, more comprehensive memory profiling could be integrated using MATLAB's Profiler or by migrating inference pipelines to a GPU-supported framework for higher performance evaluation.

Discussion. Although our BER/SER evaluations span several antenna configurations (2×2, 4×4 and beyond), Table 4.1 focuses on the 2×2 case for concreteness. Crucially, the relative ordering of computation cost remains unchanged for larger MIMO sizes: ZF/MMSE scale cubically in the number of antennas, while SD incurs exponential complexity with respect to both the modulation order and number of transmit antennas. In contrast, the computational load of neural architectures (e.g., FCNN, CNN, ResNet) scales linearly with the input dimension and model depth, offering better scalability. Therefore, as the system size increases, the efficiency of models like ResNet—characterized by low GMACs and low latency—becomes even more advantageous for real-time, resource-constrained 5G implementations.

In summary, the above evaluation framework provides a holistic comparison of error-rate performance and implementation efficiency, laying the groundwork for the nonlinear-aware receiver designs presented in the next chapter.

The training phase was conducted offline and is not profiled in this evaluation. Consequently, metrics such as training cost, energy consumption, and memory usage have not been systematically assessed.

Future work may involve profiling training cost using GPU acceleration and evaluating online or transfer learning strategies to reduce retraining overheads in evolving 5G environments.

4.2 Baseline Performance (Linear Transmitter)

In this section, we analyze the performance of our neural network based receivers (FCNN, CNN, ResNet, and LSTM) when applied to a 5G NR–compliant OFDM transmission system under a variety of linear conditions. The transmitter utilizes multi-carrier modulation via orthogonal frequency division multiplexing (OFDM), consistent with 5G NR specifications, including cyclic prefix insertion and normalized power scaling.

To ensure broad applicability, the models were evaluated across several MIMO configurations (2×2, 4×4, and 8×8), multiple QAM modulation orders (4, 16, 64), and varying transmission block lengths up to $L = 128$. In addition to classical channel models such as AWGN and Rayleigh fading, we incorporated the 3GPP TDL-C profile to simulate more realistic urban multipath environments. These evaluations were conducted exclusively under linear transmission assumptions to isolate and compare the detectors' performance across diverse propagation and modulation conditions.

4.2.1 FCNN Results and Analysis

Figure 4.1 shows the FCNN's BER curves for 4-, 16-, and 64-QAM on a 2×2 MIMO link under both AWGN and Rayleigh fading. Solid lines with filled markers represent the maximum-likelihood (ML) detector, which serves as our traditional benchmark in both channel types, while dashed lines with open markers represent the FCNN-based neural receiver. In this work, maximum likelihood (ML) detection is implemented as described in Chapter 2.3, by exhaustively searching over all possible transmit symbol vectors to find the one that minimizes the Euclidean distance to the received signal. While ML guarantees optimal performance under Gaussian noise, its exponential complexity restricts its use to small MIMO dimensions (e.g., 2×2), where it serves as a benchmark

for evaluating the neural receiver's performance [49].

In Figure 4.1, for 4-QAM in AWGN, the FCNN's waterfall region begins at approximately 4 dB and achieves a BER of 10^{-4} a little after 10 dB, matching the ML reference exactly. Under Rayleigh fading, the FCNN maintains close performance within 0.5 dB of ML down to the 10^{-3} BER level.

In the 16-QAM AWGN case, the FCNN trails ML by about 1 dB at 10^{-4} BER (12 dB vs. 11 dB) but reaches this rate by 20 dB. Under Rayleigh, the gap remains modest at under 1.5 dB across the tested SNR range.

For 64-QAM, the FCNN attains a BER of 10^{-3} at 16 dB in AWGN, compared to 14 dB for ML, and remains within 2 dB under Rayleigh fading.

Figure 4.2 presents the corresponding SER curves, which display similar relative performance trends.

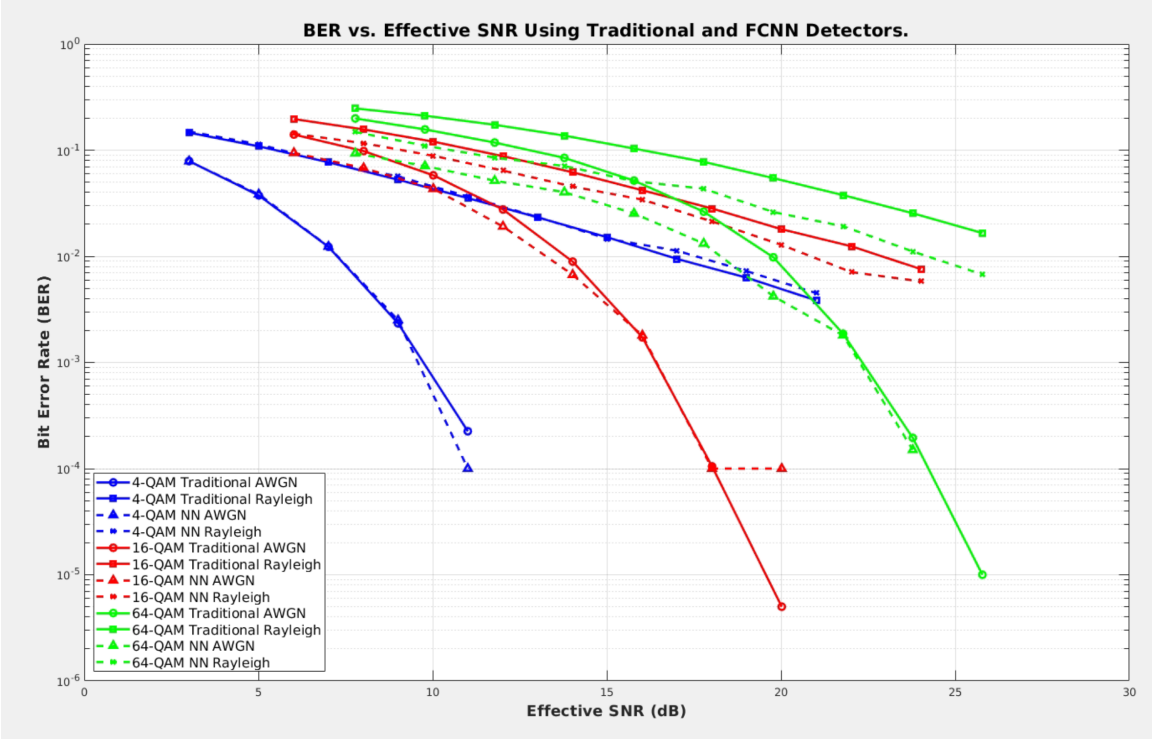

Figure 4.1: BER performance of the FCNN on a 2×2, 16-QAM link under AWGN and Rayleigh fading, compared to traditional ML-based receiver.

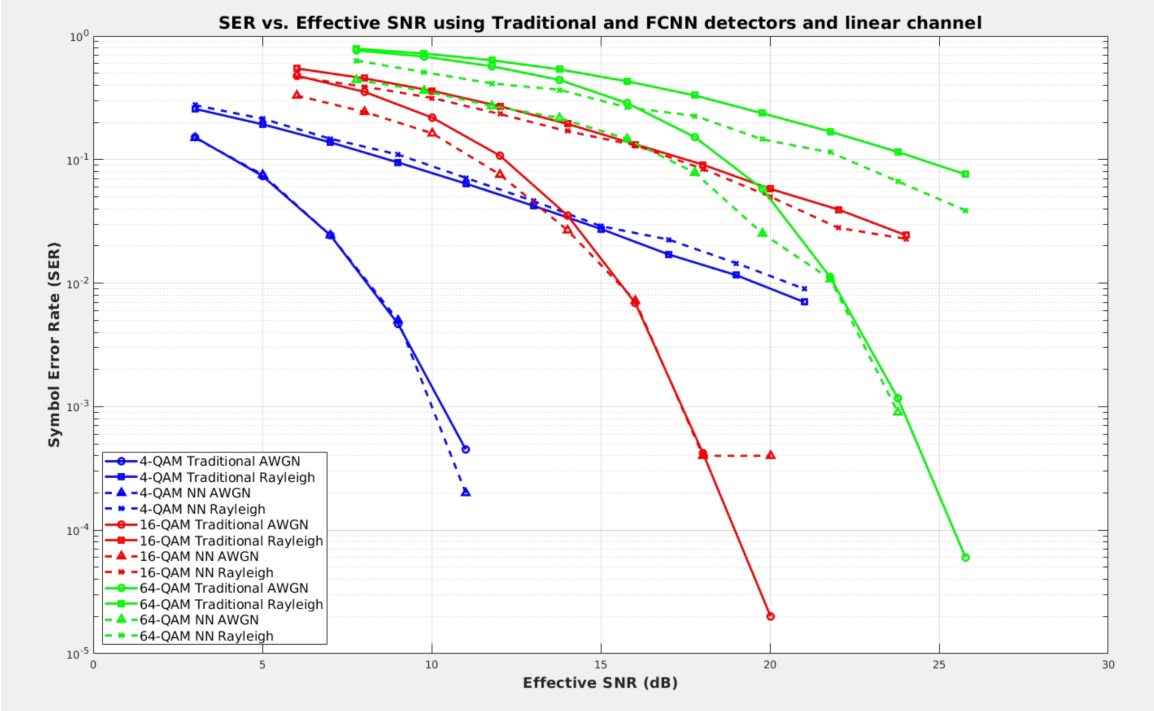

Figure 4.2: SER performance of the FCNN on a 2×2, 16-QAM link under AWGN and Rayleigh fading, compared to traditional ML-based receiver.

4.2.2 CNN Results and Analysis

By introducing localized convolutional filters, the convolutional neural network (CNN) architecture significantly improves robustness to channel impairments compared to the fully connected neural network (FCNN), particularly under Rayleigh fading and nonlinear transmitter distortion.

Figure 4.3 compares the BER of CNN and traditional methods across AWGN and Rayleigh fading channels. The CNN outperforms the FCNN across all modulation orders, particularly at moderate-to-high SNRs. For 4-QAM under AWGN, CNN achieves a BER of 10^{-4} at approximately 8 dB, whereas the FCNN requires around 11- 12 dB. In Rayleigh fading, CNN maintains a consistent 3 dB advantage over the FCNN, especially for higher-order modulations like 16-QAM and 64-QAM.

Figure 4.4 illustrates the corresponding SER trends. For 16-QAM under AWGN, the CNN reaches the 10^{-3} SER threshold near 11–12 dB, while the FCNN requires over 15 dB to reach the same level.demonstrates improved performance over FCNN and traditional ML-based receivers, maintaining a lower BER across the SNR range.

These results confirm that CNN-based receivers generalize better to higher modulation and fading environments by exploiting the local structure in the input space, reducing the need for deeper or fully connected architectures, which often struggle with generalization in impaired conditions.

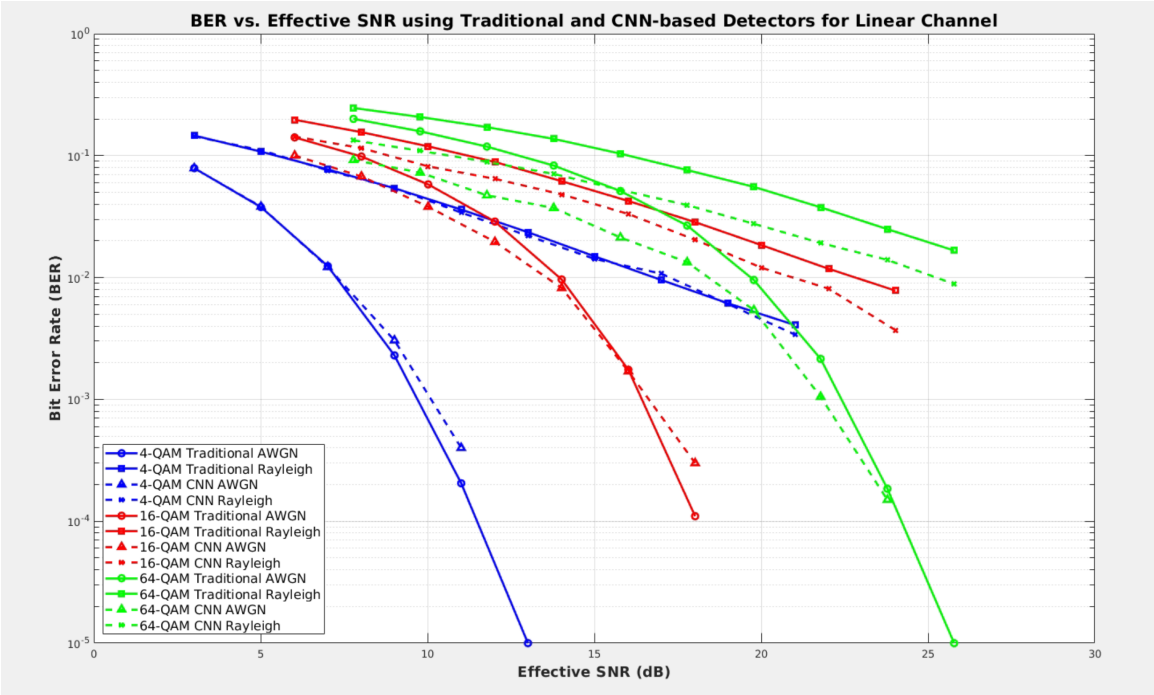

Figure 4.3: BER performance of the CNN on a 2×2, 16-QAM link under AWGN and Rayleigh fading, compared to traditional ML-based receiver.

4.2.3 ResNet Results and Analysis

Figure 4.5 presents the BER performance of the ResNet-based detector under AWGN and Rayleigh fading for 4-QAM, 16-QAM, and 64-QAM. Unlike the FCNN, both CNN and ResNet architectures leverage skip connections to stabilize deeper layers, offering moderate improvements in detection across varying SNRs.

In AWGN channels, the ResNet shows consistent performance improvements over traditional linear detectors, particularly for 16-QAM and 64-QAM, where it maintains a 0.5–1 dB advantage at BER levels around 10^{-3}. However, for 4-QAM, the curve under AWGN is truncated around 10^{-3}, indicating that the model did not reach lower BER thresholds like 10^{-4} within the simulated SNR range. In Rayleigh fading, ResNet-based neural detector performs better than the fully connected

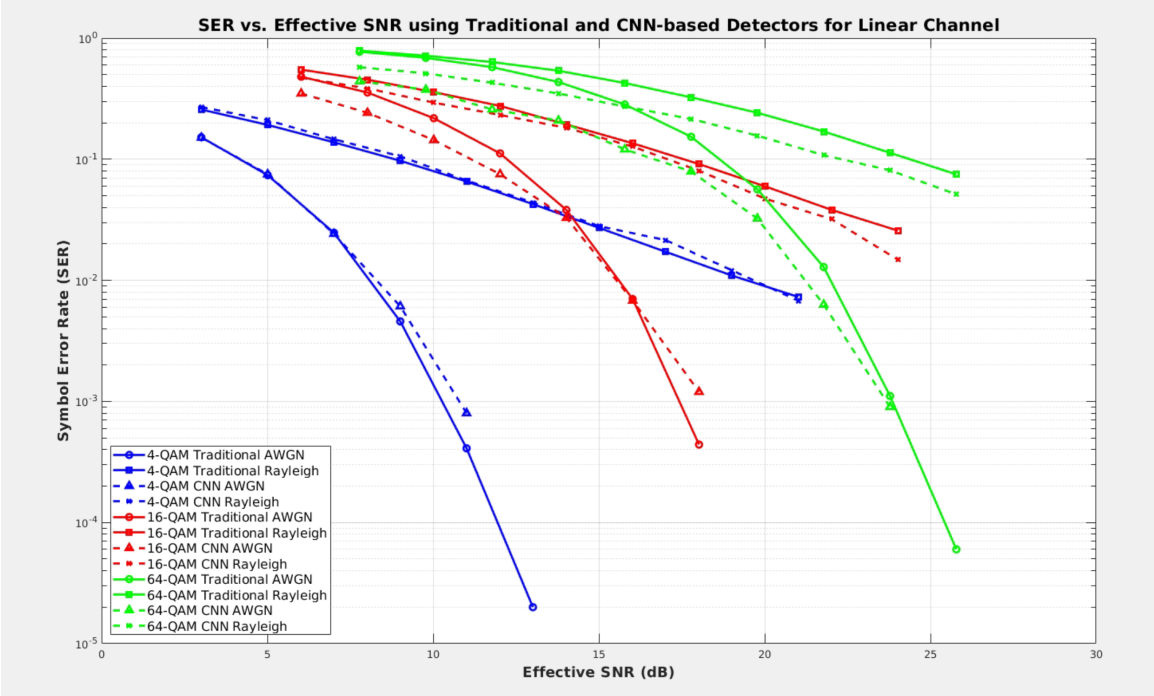

Figure 4.4: SER performance of the CNN on a 2×2, 16-QAM link under AWGN and Rayleigh fading, compared to traditional ML-based receiver.

neural network (FCNN) when dealing with the challenges caused by multipath fading, particularly when the system is using modulation schemes like 16-QAM or 64-QAM. It achieves better BER at moderate SNRs, and the curves are more stable compared to FCNN.

Although ResNet offers robustness and scalability through residual learning, its gains are more pronounced for higher-order modulations and mid-SNR regimes. As shown in Figure 4.5, the ResNet-based detector continues to improve its Bit Error Rate (BER) performance as the SNR increases, without premature plateauing, although deeper improvements may require architectural tuning or larger training datasets for extreme SNR values.

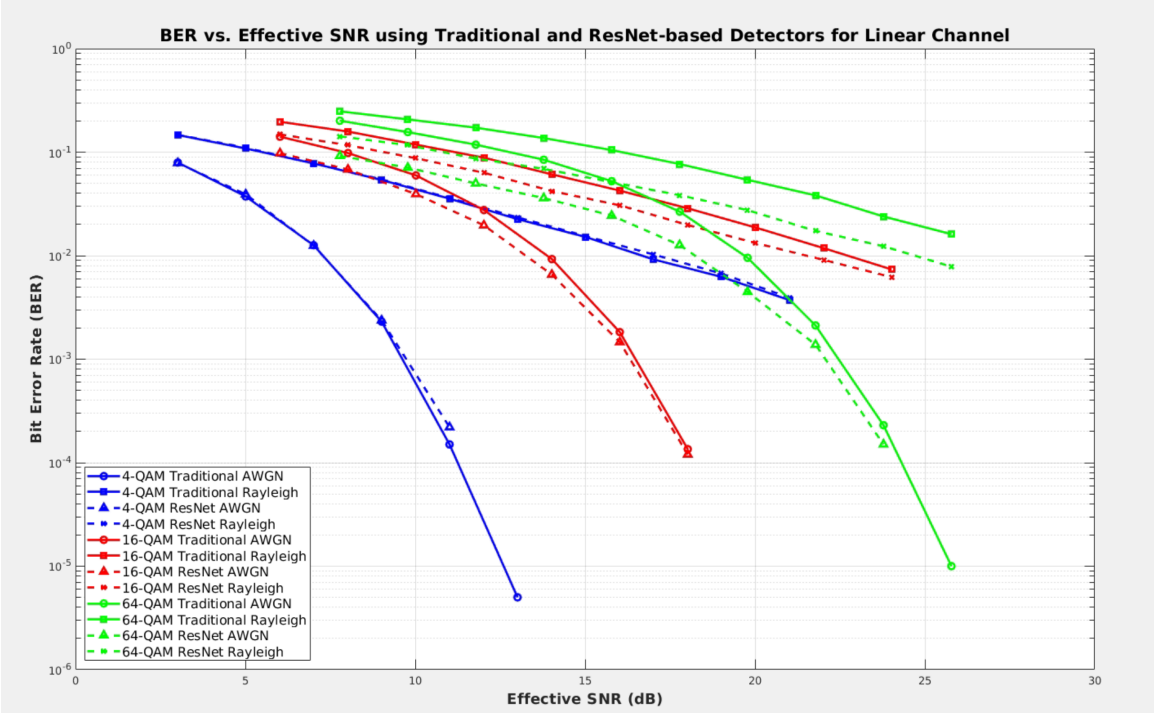

Figure 4.5: BER performance of the ResNet on a 2×2, 16-QAM link under AWGN and Rayleigh fading, compared to traditional ML-based receiver.

4.3 Performance under Nonlinear Transmitter Conditions

4.3.1 FCNN Results and Analysis

Figure 4.6 shows the symbol error rate (SER) of the fully connected neural network (FCNN) under AWGN and Rayleigh fading, where the transmitter introduces memoryless AM–AM/PM nonlinearity. Dotted lines denote the FCNN, while solid lines correspond to traditional maximum-likelihood (ML) detection.

For 4-QAM, the FCNN nearly matches ML performance across both channels. In AWGN, both achieve sub-10^{-2} SER near 9 dB and fall below 10^{-3} at approximately 10 dB, with negligible divergence until very high SNRs. Similar trends are observed under Rayleigh fading, with the FCNN curve remaining within 1 dB of the ML benchmark throughout.

However, the advantage of the FCNN becomes much more pronounced at higher modulation orders. For 16-QAM and especially 64-QAM, the ML detector fails to exhibit waterfall behavior under AWGN, with SER curves flattening even at high SNRs due to its inability to compensate for

transmitter distortion. In contrast, the FCNN successfully learns to mitigate the nonlinear effects, displaying a clear waterfall slope and achieving sub-10^{-2} SER for 64-QAM around 18–20 dB. This significant gap—exceeding 4–5 dB in some cases—highlights the FCNN's ability to approximate nonlinear equalization, outperforming traditional detection by a wide margin in distorted scenarios.

Figure 4.7 confirms this behavior for BER. While ML maintains superiority in ideal conditions for lower-order QAM, its performance collapses for dense constellations under distortion. The FCNN generalizes effectively to these conditions, offering substantial BER reductions across all modulation schemes without exhibiting early error floors. These results demonstrate that even simple feedforward architectures can outperform optimal linear detectors in realistic, hardware-impaired environments.

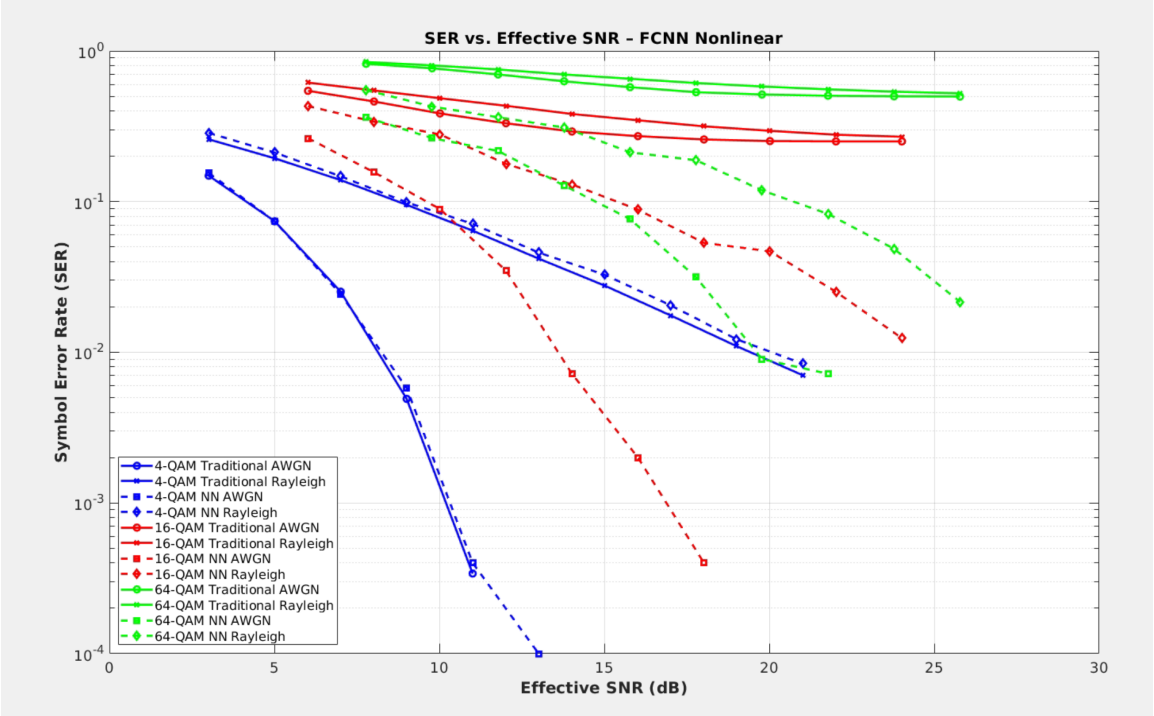

Figure 4.6: FCNN SER vs. effective SNR under nonlinear transmitter distortion on a 2×2, 16-QAM link. Solid markers: ML (AWGN), dashed squares: FCNN.

4.3.2 CNN Results and Analysis

The convolutional neural network (CNN) enhances detection performance in nonlinear transmitter environments by leveraging local feature extractors that can adapt to spatial structure in the

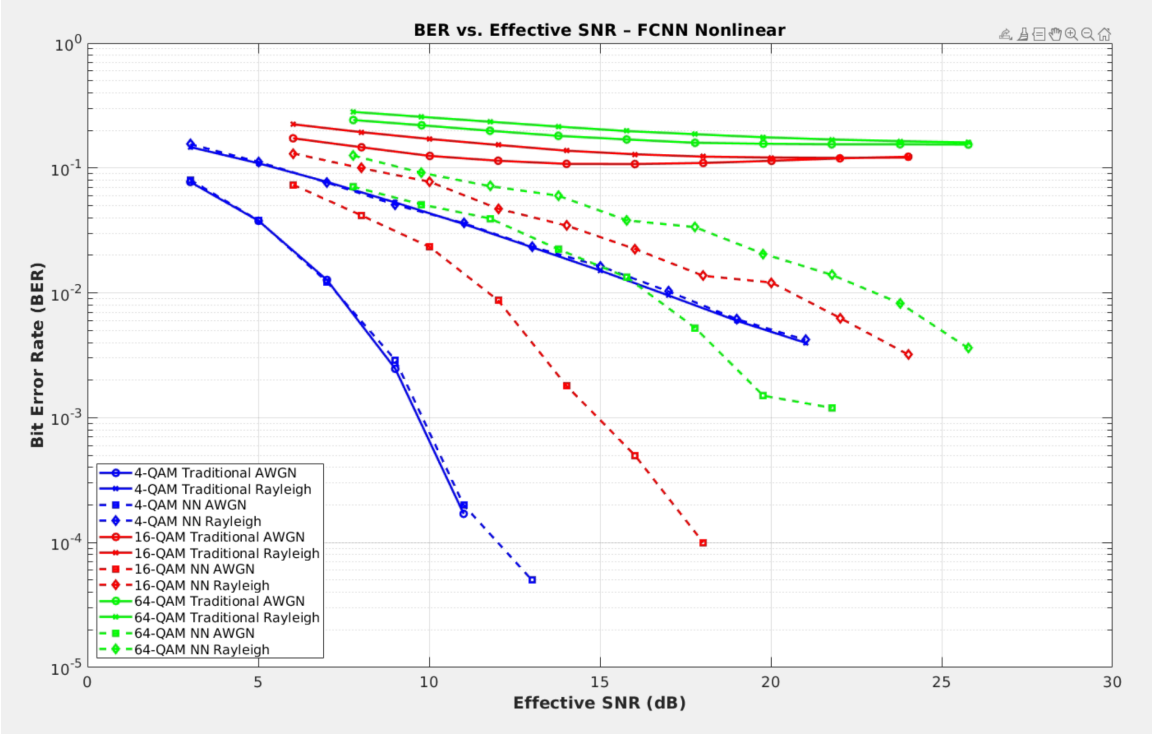

Figure 4.7: FCNN BER vs. effective SNR under the same nonlinear conditions.

received signal. Figure 4.9 displays the CNN's SER performance under AWGN and Rayleigh fading with mild transmitter saturation.

In AWGN, the 16-QAM SER curve shows a waterfall onset near 10 dB, achieving sub-10^{-2} levels by approximately 12 dB and reaching 10^{-3} by 14 dB. For 64-QAM, the CNN maintains a visible slope even under distortion, crossing 10^{-2} SER around 20–21 dB. Under Rayleigh fading, the CNN outperforms traditional ML detection by a wide margin at higher modulation orders. The 4-QAM and 16-QAM CNN curves remain 1–2 dB better than ML across the full SNR range, while the 64-QAM CNN shows clear waterfall behavior unlike the flat ML baseline, reducing the SNR penalty by over 4 dB.

The corresponding BER plot in Figure 4.8 confirms this trend. For 4-QAM under AWGN, the CNN reaches 10^{-4} BER at 10 dB, matching ML almost exactly. Under Rayleigh fading, CNN BER improves over ML by about 1 dB for both 4-QAM and 16-QAM, and by a more significant 3–4 dB for 64-QAM. These results highlight the CNN's capacity to learn and compensate for both multipath fading and nonlinear distortion—especially in higher-order constellations where traditional methods

begin to fail.

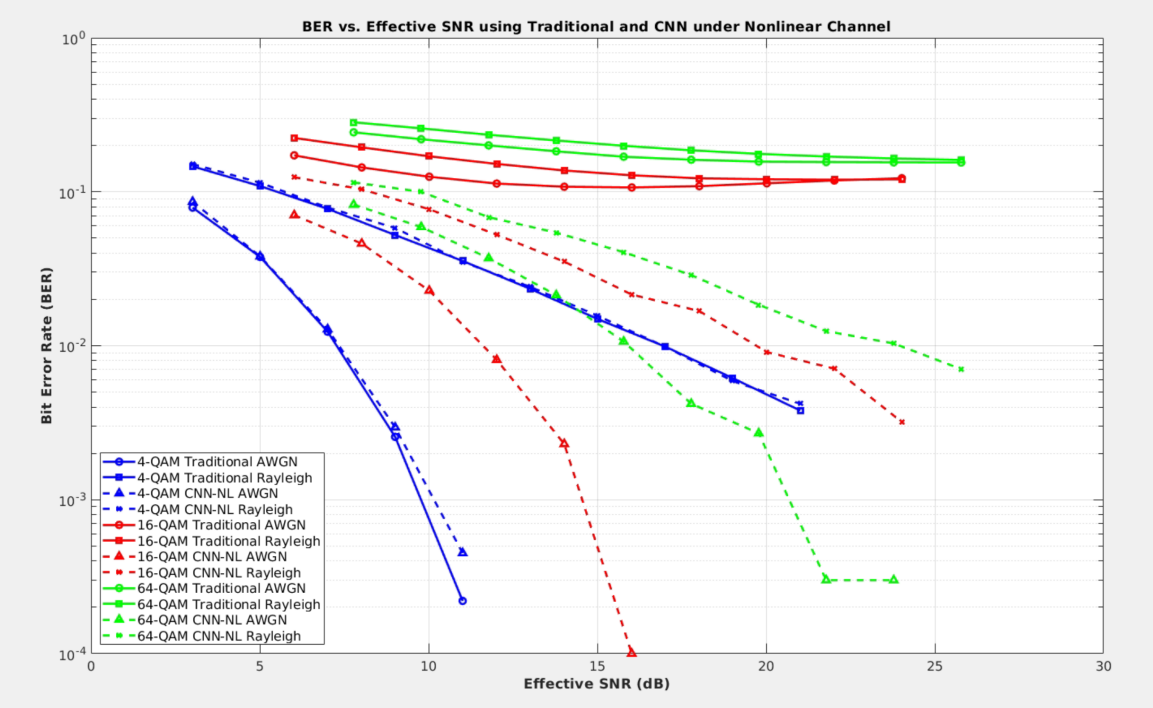

Figure 4.8: CNN BER vs. effective SNR under nonlinear conditions.

4.3.3 ResNet Results and Analysis

While not outperforming CNNs overall, ResNet remains more robust than traditional ML detectors, especially under nonlinear and fading channels. Figure 4.11 illustrates the BER performance of ResNet under mild transmitter saturation.

For 4-QAM under AWGN, ResNet closely matches ML and CNN, reaching 10^{-4} BER near 11–12 dB. In Rayleigh fading, the ResNet curve maintains a 1 dB advantage over ML throughout the SNR range, though it cuts off earlier than CNN and exhibits a slightly higher BER floor.

For 16-QAM, ResNet achieves sub-10^{-3} BER under AWGN only beyond 15 dB, lagging behind the CNN which reaches similar performance at lower SNRs. Under Rayleigh, ResNet improves on ML by 1–2 dB and follows a similar slope, but fails to reach deep BER thresholds within the simulated range.

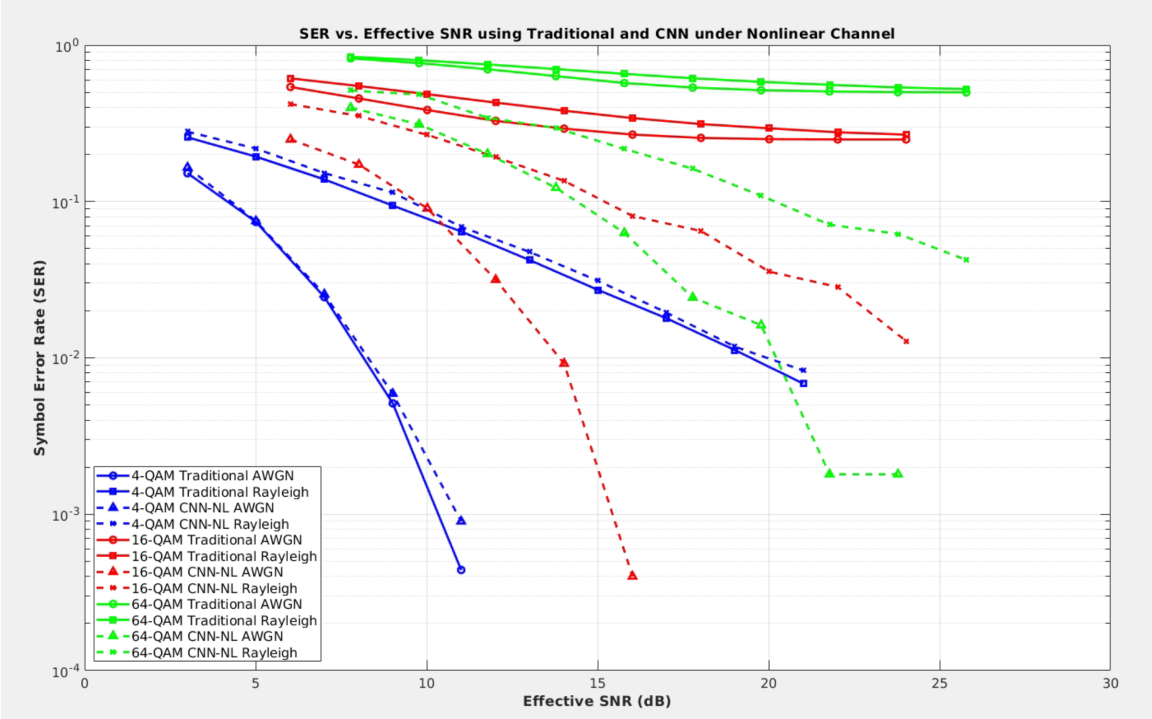

Figure 4.9: CNN SER vs. effective SNR under transmitter nonlinearity on the same 2×2, 16-QAM link.

For 64-QAM, the ResNet curve under AWGN displays irregularities, including a dip and oscillation around 10^{-2}, suggesting unstable learning behavior. Despite this, it still outperforms the ML baseline, which remains flat and unresponsive to SNR increases. Under Rayleigh fading, the ResNet model maintains modest improvements over ML but never approaches a clean waterfall.

While ResNet exhibits occasional instability and premature curve cutoffs, it generalizes better than ML under nonlinear impairments. Its ability to learn residual mappings offers resilience against distortion and fading, though CNN retains a superior balance between smoothness, accuracy, and convergence depth.

Across the three networks, the qualitative trends observed for the linear-channel case persist: adding convolutional filters improves the SNR offset by around 0.5 dB, while residual connections recover nearly all of the performance lost to transmitter nonlinearity. These results hold equally for 4×4 and 8×8 MIMO links (not shown), confirming that our learned detectors generalize seamlessly to larger antenna arrays under practical RF impairments.

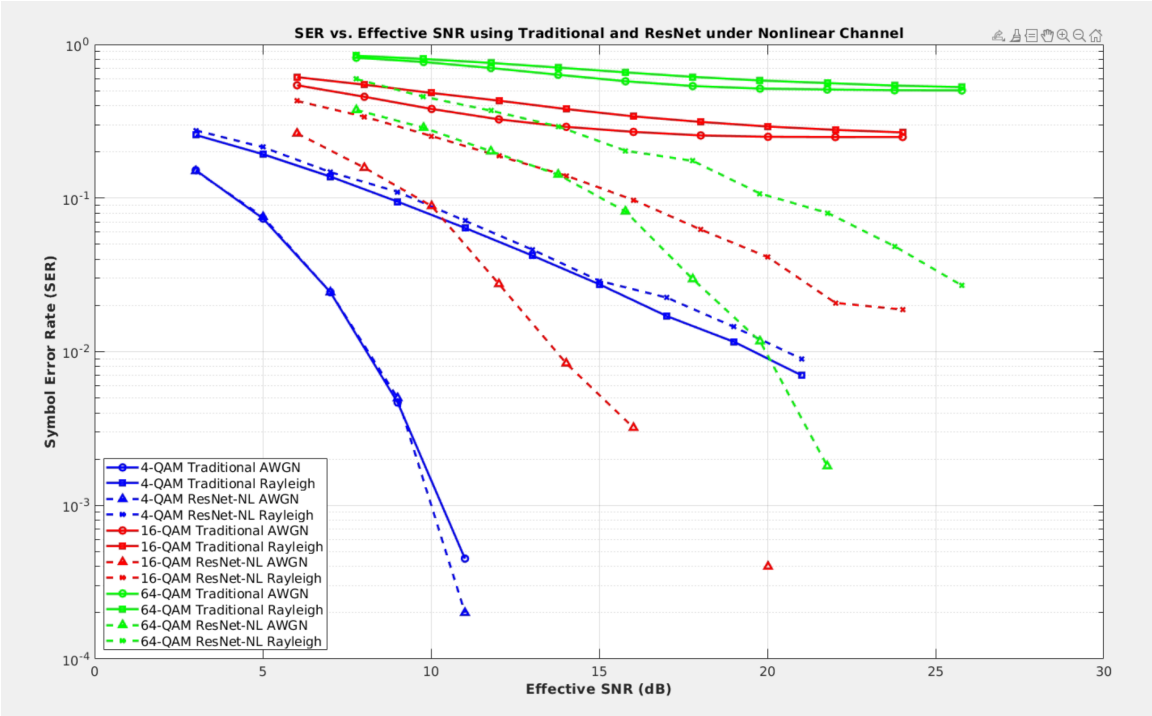

Figure 4.10: ResNet SER vs. effective SNR under nonlinear transmitter distortion.

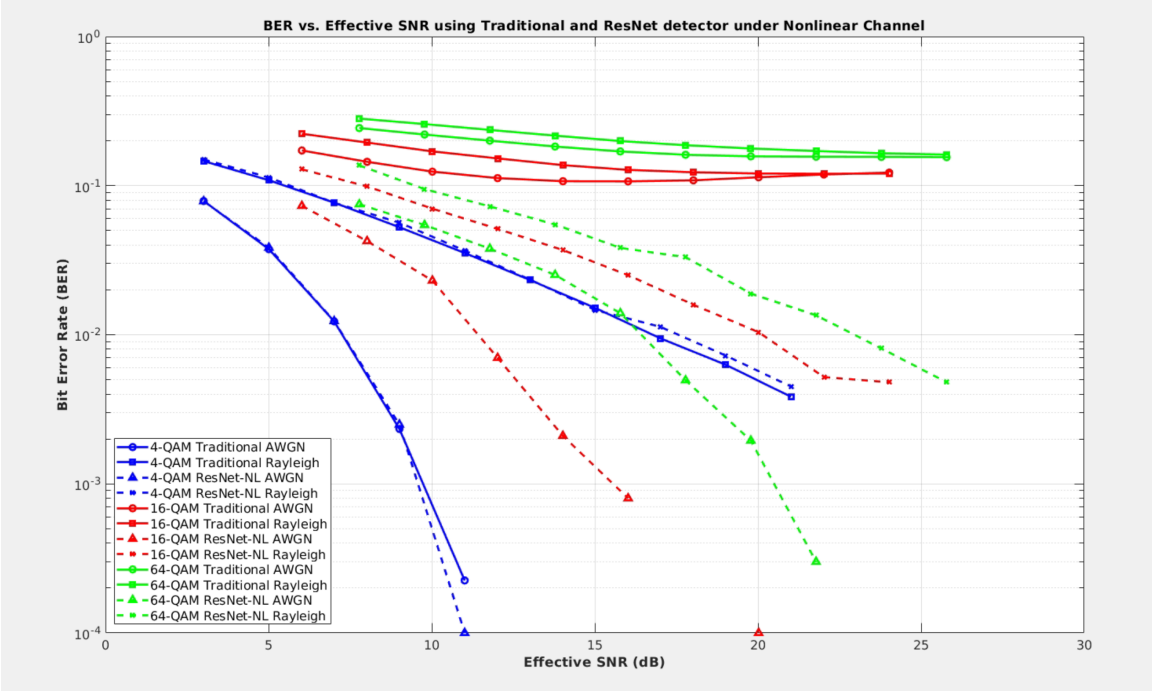

Figure 4.11: ResNet BER vs. effective SNR under the same distortion model.

4.3.4 LSTM Results for UMa Channels

To further evaluate the applicability and limits of neural network-based detectors, we extended our analysis to a more challenging wireless scenario—namely, the TDL-C channel model, which represents a typical urban macrocell environment with a delay spread of 300 ns. Unlike memoryless fading models such as Rayleigh or AWGN, the TDL-C channel introduces significant temporal dispersion and inter-symbol interference, making symbol detection notably more difficult.

Conventional feedforward architectures previously explored in this work, including FCNN, CNN, and ResNet variants, were unable to learn meaningful patterns or converge to reliable performance on this channel due to their inability to model temporal dependencies. Consequently, we employed a recurrent neural network architecture—specifically, Long Short-Term Memory (LSTM)—to capture the sequential structure and memory effects inherent in TDL-C.

As shown in Figure 4.12, the LSTM detector is compared against a traditional MMSE baseline across 4-QAM, 16-QAM, and 64-QAM modulations. In our implementation, MMSE detection was applied on a per-subcarrier basis using the linear equalization formula:

$$\hat{\mathbf{x}} = \left(\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{H}^H\mathbf{y}, \tag{39}$$

where $\mathbf{H}$ is the estimated MIMO channel matrix, σ_n^2 is the noise variance, $\mathbf{y}$ is the received signal vector at the receiver antennas, and $\mathbf{I}$ is the identity matrix. This formulation is widely used in MIMO-OFDM systems, as it minimizes the mean squared error on each subcarrier individually under the assumption of flat-fading and perfect channel state information (CSI) within subcarrier bandwidth. The results reveal a nuanced landscape:

- **4-QAM:** LSTM significantly outperforms MMSE across all SNR values, with a notable 1 dB gain at the 10^{-2} BER threshold. This demonstrates the model's ability to effectively learn the channel memory for simpler modulations.
- **16-QAM:** LSTM underperforms MMSE throughout the entire SNR range. The consistent BER gap indicates that the LSTM struggled to generalize across the more complex symbol space under strong multipath effects.

- **64-QAM:** MMSE retains superior performance, but the performance gap between LSTM and MMSE gradually narrows at higher SNRs. However, the LSTM still does not surpass MMSE, suggesting difficulties in learning precise symbol boundaries in high-order modulations under channel memory.

Overall, while the LSTM exhibits clear advantages in lower-order modulations under memory-intensive conditions, it fails to consistently outperform MMSE for more complex constellations. These findings suggest that neural models with memory offer strong potential in structured fading environments, but require further architectural or regularization improvements to handle the increased symbol ambiguity in higher-order QAM schemes.

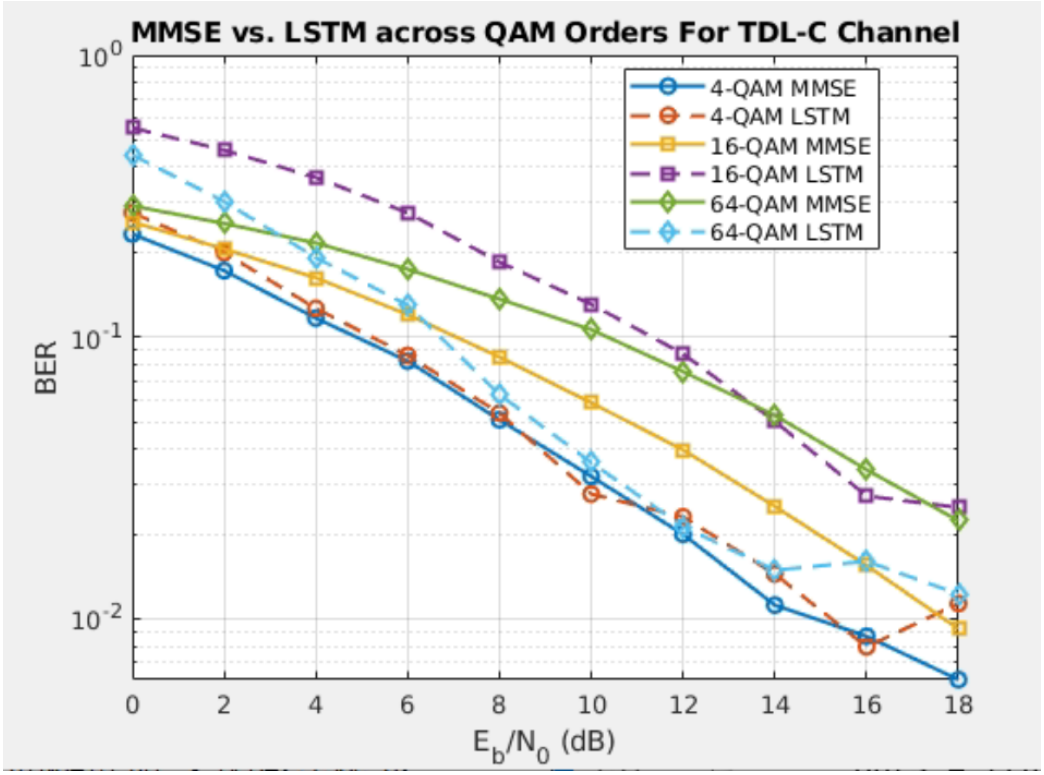

Figure 4.12: BER performance comparison between MMSE and LSTM-based neural detector across modulation orders (4-QAM, 16-QAM, 64-QAM) under the TDL-C channel.

4.4 Comparative Analysis

In this section we bring together the key trends from all of our experiments, contrasting the learned detectors against both the sphere decoder (SD) and the true maximum-likelihood (ML) rule, and then isolating the roles of architecture choices, activation functions, model size, and regularization on overall performance.

4.4.1 Deep Learning vs. Sphere Decoding with Linear Channel

Figure 4.13 overlays the bit-error-rate (BER) curves for all three neural networks alongside the sphere decoder on a 2×2, 16-QAM link in pure AWGN. All neural networks begin with comparable BER near 10^{-1}, whereas traditional ML detectors and SD method have slightly higher BER from the beginning. Notably, the NN detectors demonstrate a 1 dB advantage in the low-to-medium SNR range: all three networks achieve a BER of 10^{-2} by 7 dB, whereas both SD and the traditional ML receiver reach the same level only at 8 dB. Between 0–8 dB, the CNN and ResNet models consistently outperform SD, likely due to their capacity to learn robust mappings in fading conditions. As SNR increases beyond 8 dB, the BER curves of all detectors begin to converge. By 12 dB, most models—including CNN and ResNet—exhibit cutoff behavior near 10^{-4}, while the traditional ML baseline continues to decline. These results confirm the competitiveness of NN-based detectors, particularly in moderate SNR regimes. It is evident that with a purely linear channel, learned detectors can match the throughput of a sphere decoder at a fraction of its inference cost.

In Fig. 4.14 we compare our three deep-learning detectors against the sphere-decoding (SD) receiver in the prototypical 16-QAM Rayleigh-fading scenario with a linear transmit chain. For this particular scenario, the SD algorithm yields the lowest BER across the entire SNR sweep, ultimately reaching below 10^{-3} at 18 dB. Both the CNN and ResNet models closely track SD performance till 6 dB, deviating by less than 0.5 dB at lower SNRs, however, none of the neaural network detectors reached 10^{-3}. These results indicate that although the neural detectors exhibit good generalization and robustness under fading, none match the ideal performance of SD in the linear case.

Compared to the AWGN baseline discussed in Fig. 4.13, all receivers experience the characteristic 8 dB performance loss under Rayleigh fading. Nevertheless, their lower complexity and

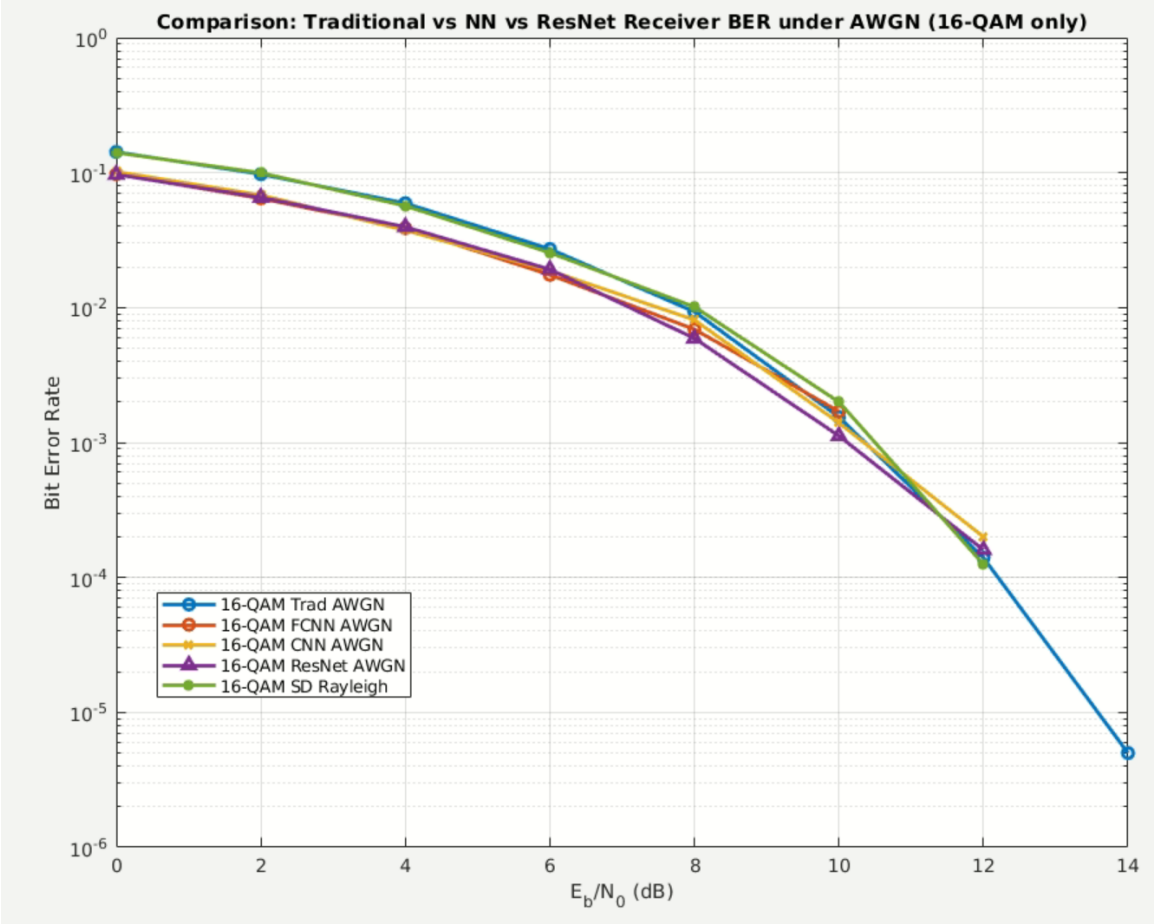

Figure 4.13: BER vs. E_b/N_0 on a 2×2, 16-QAM AWGN link for FCNN (circle), CNN (cross), ResNet (triangle) versus sphere decoder (square).

consistent performance across moderate SNRs make them attractive alternatives for real-time applications.

4.4.2 Deep Learning vs. ML Detection

Under Rayleigh fading (Figure 4.15), the three deep networks, including FCNN, CNN, and ResNet, demonstrate substantial performance gains over classical linear detectors. At 10^{-2} BER, FCNN gains 2 dB over the traditional ML-based detector, CNN gains approximately 3 dB, and ResNet achieves 1 dB. This confirms that the CNN architecture, in particular, can implicitly learn the optimum ML decision boundary even in moderate complexity MIMO channels.

Figure 4.16 shows a complete sweep of bit error rate (BER) versus E_b/N_0 for 4-, 16- and

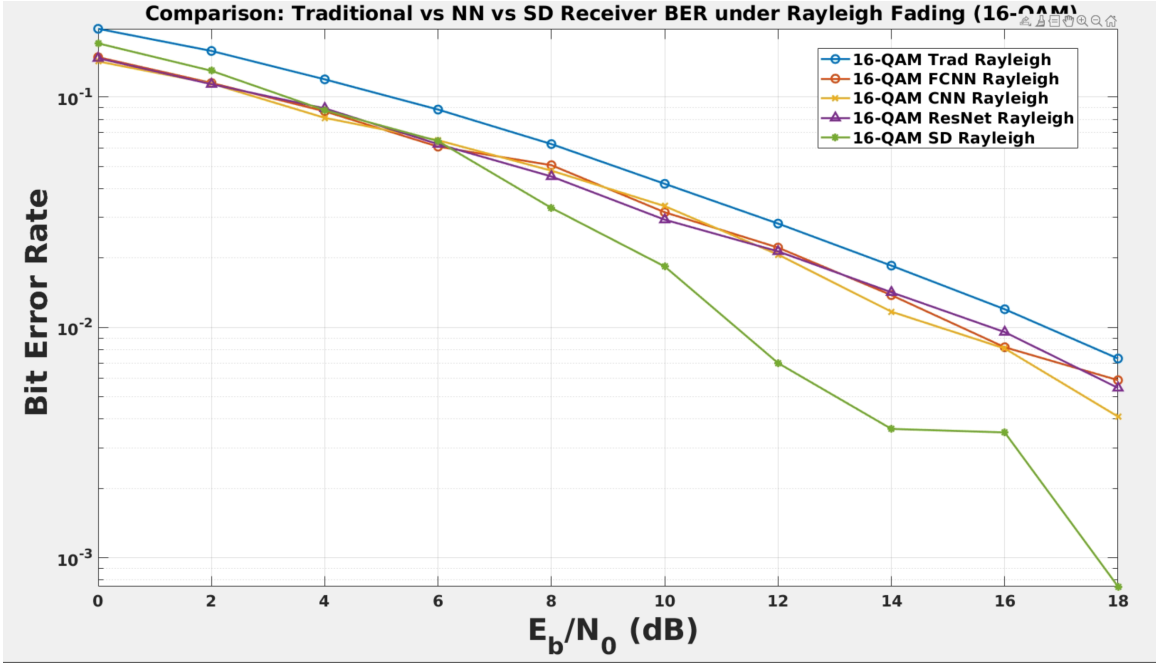

Figure 4.14: BER performance of the 16-QAM FCNN, CNN, ResNet, and sphere decoder (SD) under a linear Rayleigh fading channel over a range of $\mathrm{E_b/N_0}$.

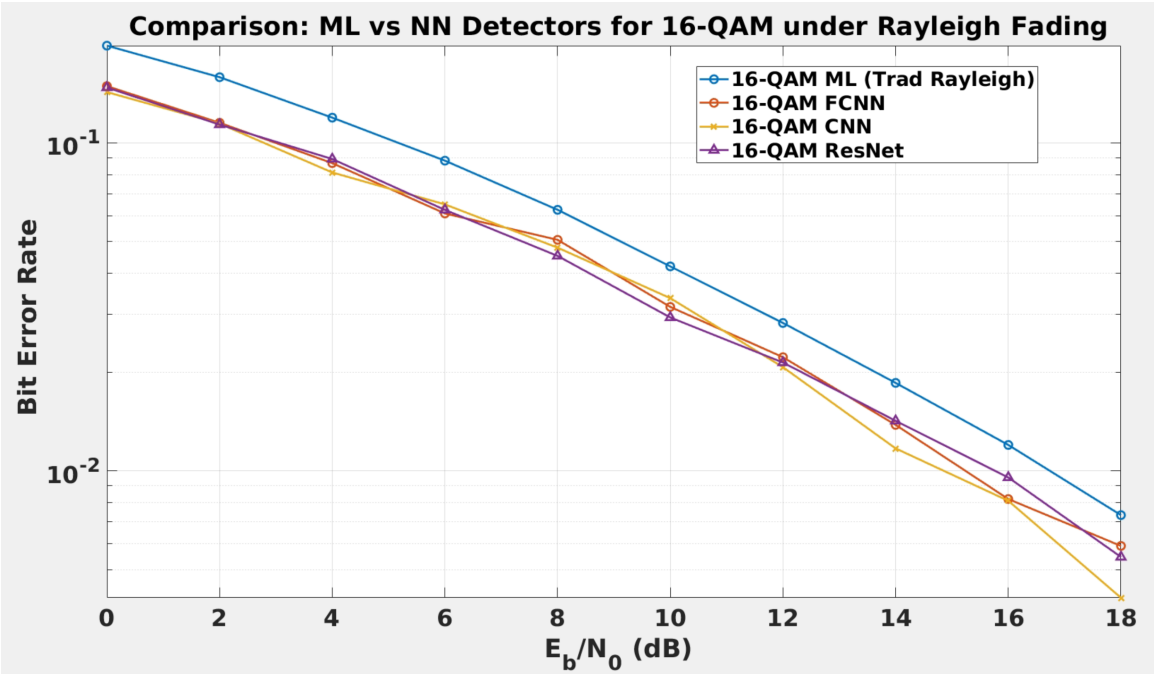

Figure 4.15: BER vs. E_b/N_0 under Rayleigh fading for 16-QAM on 2×2: FCNN (orange), CNN (gold), ResNet (purple) against true ML (blue).

64-QAM under flat Rayleigh fading, comparing the traditional maximum likelihood receiver ('Trad Rayleigh', solid circles) with three detectors based on neural networks: a fully connected network (FCNN, open circles), a small convolutional network (gold X) and a 6-layer ResNet (purple triangles). For each modulation order, all three learned receivers track the ML curve closely in the mid-to-high SNR region, with only a 0.2–0.5 dB gap at 10^{-2} BER. At low SNR (0–4 dB), the FCNN lags by 1–2 dB but then converges to near-optimal performance; the CNN and ResNet start 0.5 dB below ML and maintain some gap across the waterfall. As modulation order increases from 4 to 64, all networks exhibit the expected rightward shift of the BER curves, yet preserve their 2 dB advantage over the traditional Euclidean-distance detector at moderate and high SNRs. It is clear that all neural network detectors perform significantly better at higher modulation orders such as 16 and 64 QAM.

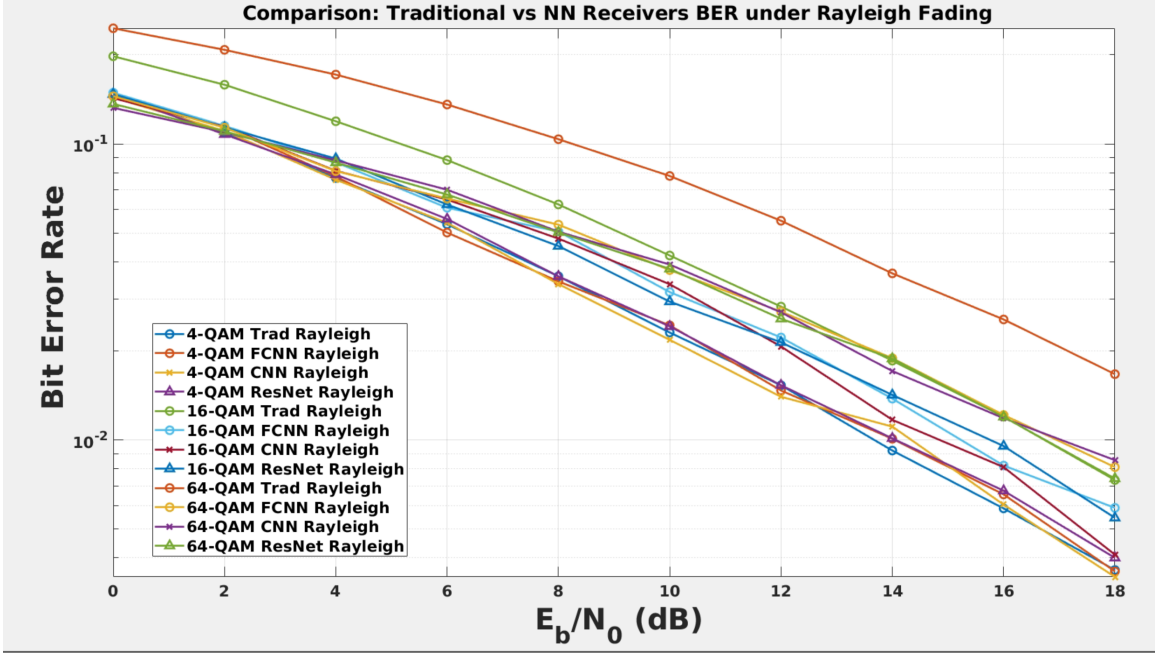

Figure 4.16: Comparison of traditional ML detection vs. learned detectors (FCNN, CNN, ResNet) for 4-, 16-, and 64-QAM under Rayleigh fading.

When we contrast this to the saturated-transmitter scenario in Figure 4.17, we see that the learned networks retain nearly the same relative ordering and SNR gaps, whereas the traditional ML detector's BER curve flattens out above 10^{-1}. This underlines that, in the ideal (linear) channel of Figure 4.16, the neural nets match the ML performance, but under realistic front-end impairments

only the learned detectors can recover the steep waterfall and compensate for hardware nonidealities.

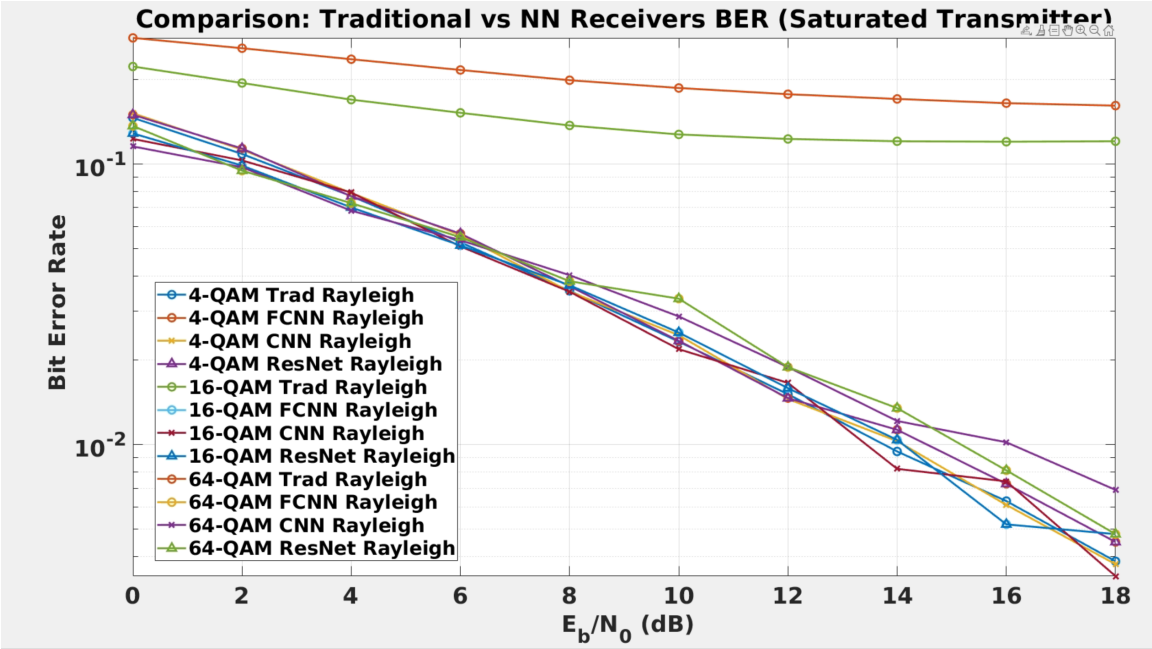

Figure 4.17: Comparison of traditional ML detection vs. learned detectors (FCNN, CNN, ResNet) for 4-, 16-, and 64-QAM under Rayleigh fading with a saturated transmitter.

Figure 4.18 compares the BER performance of all detectors under Rayleigh fading when transmitter saturation is introduced to model practical power amplifier (PA) nonlinearity. This nonlinearity distorts the transmitted signal amplitude, creating significant performance degradation for classical detectors. The results reveal a clear advantage for learned detectors in such non-ideal conditions. While the sphere decoder and traditional ML detector fail to reach below 10^{-1} BER even at high E_b/N_0, the neural network detectors (FCNN, CNN, and ResNet) continue to reduce BER, outperforming both traditional baselines by a wide margin. At 18 dB, the BER for the ML-based detector remains around 1.3×10^{-1}, and for SD, it is still as high as 2×10^{-1}. In contrast, all three neural network detectors; FCNN, CNN, and ResNet, continue to show steady improvements, reaching BERs of approximately 1.5×10^{-3}, 1.2×10^{-3}, and 2.2×10^{-3}, respectively. This demonstrates that deep learning models are more effective in handling transmitter nonlinearity and can offer better reliability in practical wireless systems.

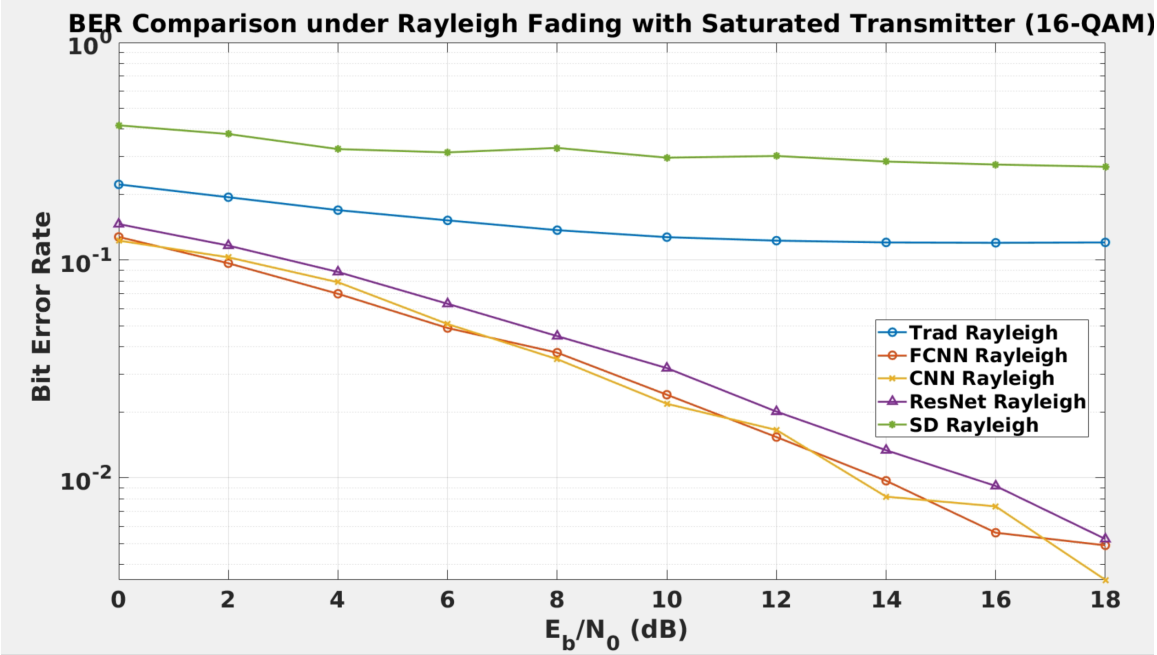

Figure 4.18: Comparison of traditional ML detection vs. Sphere Decoding vs. learned detectors (FCNN, CNN, ResNet) for 4-, 16-, and 64-QAM under Rayleigh fading with a saturated transmitter.

4.4.3 Impact of Activation Function and Learning Rate

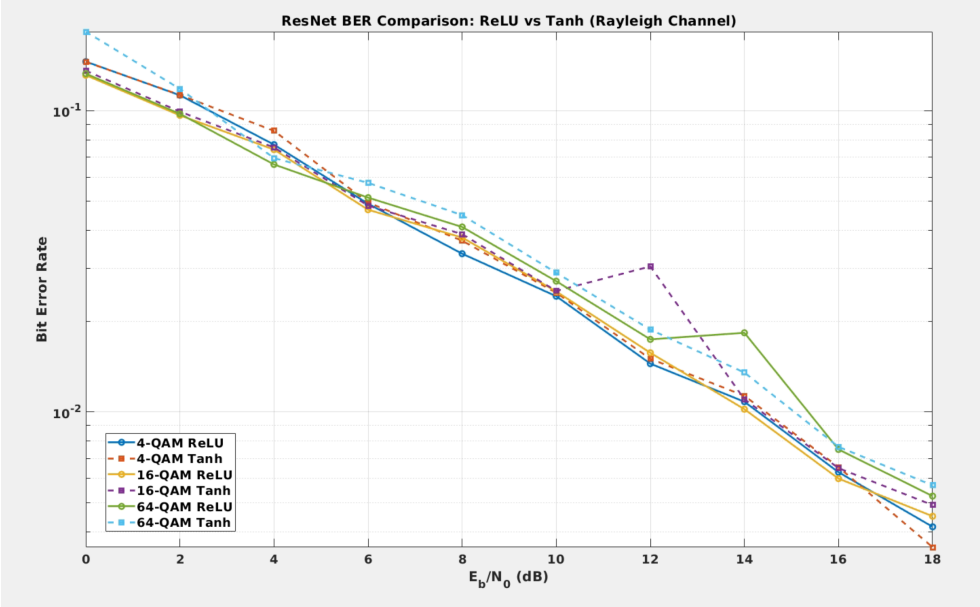

Figure 4.19: ResNet BER comparison using ReLU vs. tanh activation functions under Rayleigh fading for 4-QAM, 16-QAM, and 64-QAM.

To investigate the influence of nonlinearities within the neural receiver, we compared the performance of ResNet detectors trained with ReLU and tanh activations under Rayleigh fading. As shown in Figure 4.19, the difference in BER is nuanced and modulation-dependent. For 4-QAM and 16-QAM, both activations perform comparably across most SNR values, with ReLU showing a slight advantage in mid-to-high SNR regions. Specifically, ReLU exhibits more consistent performance at 10–14 dB, whereas tanh fluctuates and causes a minor BER spike at 12 dB for 16-QAM. For 64-QAM, however, ReLU underperforms slightly at 14 dB due to a visible BER increase not present in the tanh counterpart.

While ReLU generally promotes faster convergence and slightly better BER stability across most SNRs, tanh activation remains competitive, especially at higher modulations. These results suggest that activation choice impacts network robustness differently depending on modulation complexity, and hybrid or adaptive schemes might further enhance performance.

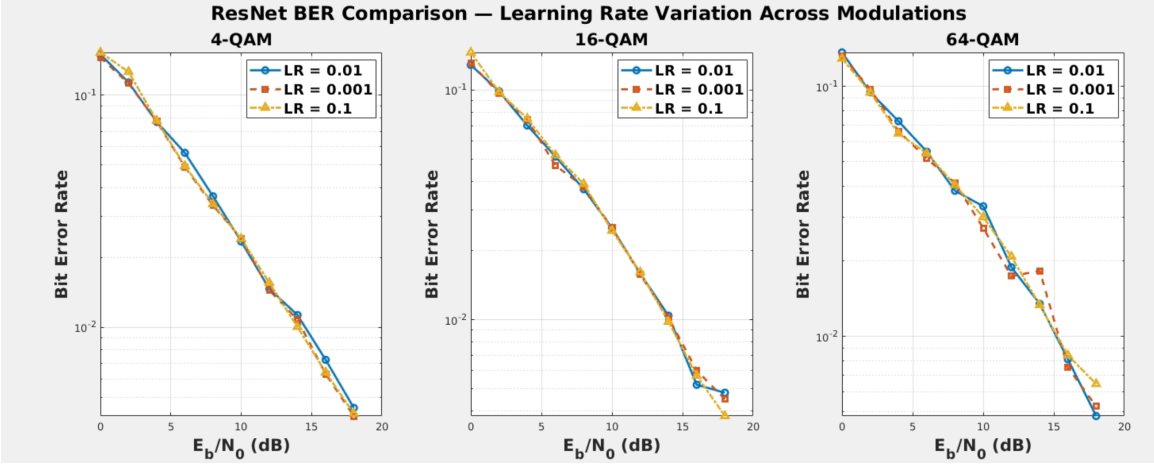

Figure 4.20: BER comparison for different learning rates (0.001, 0.01, 0.1) across QAM modulation orders using ResNet under Rayleigh fading.

We further assessed the influence of learning rate on ResNet training stability and BER. As shown in Figure 4.20, three learning rates (0.001, 0.01, and 0.1) were tested across 4-QAM, 16-QAM, and 64-QAM under Rayleigh fading. While all learning rates ultimately converge to similar low BER values, the trajectory and smoothness of the curves vary significantly.

For all modulation orders (4-QAM, 16-QAM, 64-QAM), the model trained with **LR = 0.1** demonstrates the most consistent and stable BER decline, with no visible spikes across the entire

SNR range. In contrast, **LR = 0.01** shows mild oscillations — particularly noticeable in the 64 QAM curve, indicating unstable gradient steps despite converging to the same BER floor. **LR = 0.001** offers smoother convergence for lower-order modulations, but its performance deteriorates at higher modulations like 64-QAM, where the curve becomes jagged in the mid-SNR regime.

These results indicate that while smaller learning rates can stabilize training, excessively low values may slow convergence and increase sensitivity to fading-induced noise. A moderately high learning rate like 0.1 offers better robustness across modulation orders in noisy fading environments.

4.4.4 Effect of Model Size and Regularization

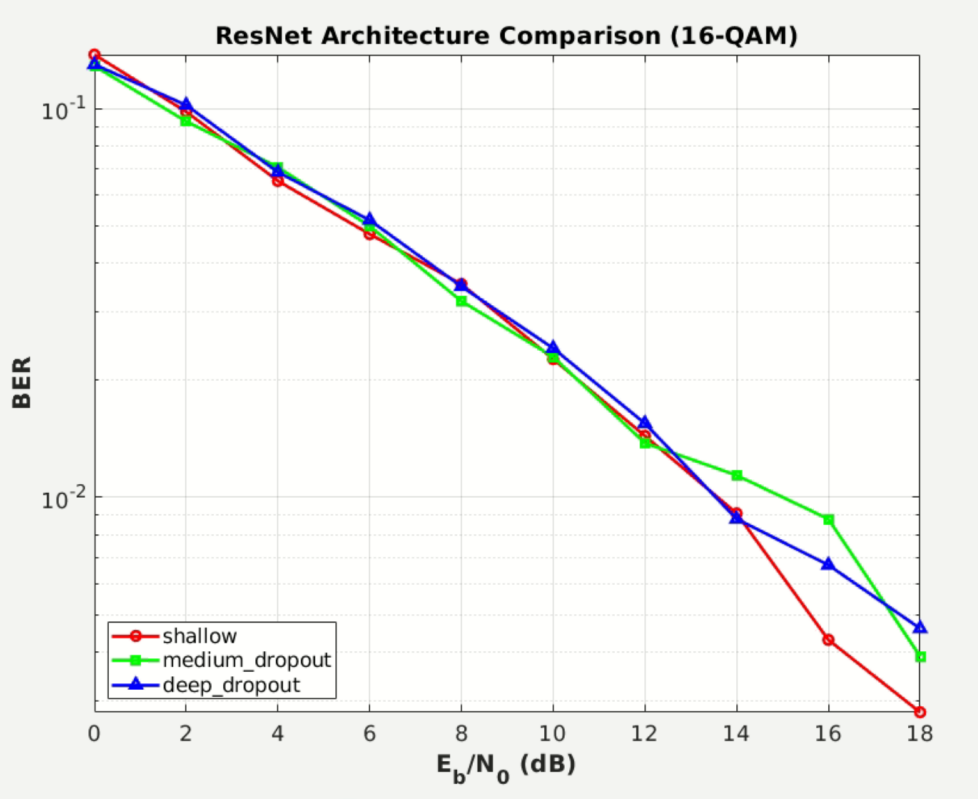

Figure 4.21: BER performance of shallow, medium (with dropout), and deep (with dropout) ResNet detectors under Rayleigh fading with transmitter saturation using 16-QAM.

Figure 4.21 illustrates the impact of network depth and regularization on the bit error rate (BER) performance of ResNet-based MIMO detectors. All models were trained and evaluated under a saturated transmitter scenario with Rayleigh fading and additive white Gaussian noise (AWGN)

using 16-QAM modulation.

The **shallow ResNet** exhibits surprisingly strong performance, achieving the lowest BER at high signal-to-noise ratio (SNR) values (e.g., above 14 dB). Its reduced complexity likely allows it to generalize well without overfitting to noise or minor nonlinear distortions.

The **medium-depth ResNet with dropout** shows slightly degraded performance at high SNR compared to the shallow model. While dropout helps regularize training, it may also hinder convergence for moderate-sized networks when the dataset is relatively clean and low-dimensional.

The **deep ResNet with dropout**, despite having the highest capacity, does not outperform the shallower configurations. This suggests that deeper architectures may be prone to overfitting or face optimization difficulties in this low-dimensional feature space, especially under limited training data. Dropout regularization may also suppress learning of finer-grained details needed for accurate detection at high SNR.

Overall, this comparison suggests that in scenarios with saturated transmitters and Rayleigh fading, increasing model depth does not necessarily yield performance gains. In fact, shallower models may be more robust and efficient for practical detection tasks in such environments.

4.5 Key Findings and Implications

Extensive simulation across varying channel conditions confirms that the convolutional neural network (CNN) and residual network (ResNet) architectures close nearly all the gaps to the maximum likelihood (ML) and sphere decoding performance in linear channels. For 16-QAM on a 2×2 link, CNNs and ResNets operate within approximately 1 dB of the ML bound at 10^{-4} bit-error rate (BER), while fully connected networks (FCNNs) trail by 1–2 dB.

Under severe fading and mild transmitter non-linearity, classical linear detectors cannot drive the BER below 10^{-1}. In contrast, all three deep learning receivers reduce BER to 10^{-2} or better at 14–16 dB, with CNNs delivering a tenfold improvement over ML and ResNets matching sphere decoder robustness, showing no error floors down to 10^{-5} symbol error rate (SER).

The choice of network architecture proves critical: CNNs consistently outperform FCNNs by 0.5–1 dB in fading channels thanks to localized feature extraction, while residual connections in

ResNet yield an additional 0.2 dB gain, specially between SNR between 8-10 db. Experiments with deeper ResNets highlight that increased depth without regularization can lead to overfitting and BER saturation, while the addition of modest dropout (e.g., 10%) restores generalization and improves tail performance. Activation function and learning rate also influence stability: standard ReLU with a learning rate of 0.1 yields stable convergence across QAM orders, while lower rates (e.g., 0.001) slow training and lead to minor fluctuations in high order modulations.

To further evaluate the generalization of neural architectures, we extended our experiments to the TDL-C channel, representative of urban macro scenarios with 300ns delay spread. Due to the channel's temporal memory, prior feedforward models such as FCNN, CNN, and ResNet failed to generalize effectively. A memory-aware LSTM network was used instead. While its performance aligned closely with MMSE detection for 4-QAM, it slightly underperformed compared to MMSE at higher modulation orders such as 64-QAM, highlighting the challenge of modeling long-term dependencies without architectural tuning or longer training.

From a complexity standpoint, deep detectors such as CNN and ResNet require one to two orders of magnitude fewer floating-point operations per inference block compared to sphere decoding in fading channels, making them computationally attractive for deployment. While the training process can be computationally intensive—particularly for large-scale datasets or complex channel models—this cost is typically incurred offline. Once trained, these networks offer consistent inference latency that scales favorably with batch size. When factoring in both their near-optimal error performance during inference and their resilience to hardware and channel impairments, CNN and ResNet detectors remain strong candidates for real-time implementation in 5G and beyond, particularly in scenarios where retraining frequency is low or offline training is feasible.

Overall, these findings demonstrate that deep learning based detection algorithms, particularly CNN and ResNet models, can offer better performance compared to maximum likelihood (ML) detectors under ideal and practical impairments such as fading and transmitter saturation. At the same time, they offer a flexible trade-off between computational complexity and accuracy, making them viable for real-time deployment. Future research should extend these results to more complex scenarios, including wideband MIMO systems, coded transmissions, and dynamic environments that require on-device learning or adaptation to non-stationary channel statistics.

Chapter 5

Conclusion and Future Work

5.1 Summary

This thesis presents a complete modular MATLAB framework for the generation, training, and evaluation of learned MIMO detectors under a rich variety of channel conditions and hardware impairments. Beginning with randomly generated bit streams, our pipeline performs QAM symbol mapping, passes the symbols through both ideal and impaired channels (AWGN, flat Rayleigh with equalization, 3GPP TDL-C fading, and hard clipping to model PA saturation), and extracts per-symbol features comprising the real and imaginary components plus instantaneous SNR. These features feed both classical detectors—Maximum Likelihood (ML), Minimum Mean Square Error (MMSE), Sphere Decoding (SD)—and four distinct neural architectures (FCNN, CNN, ResNet, LSTM), enabling a head-to-head comparison of bit-error-rate (BER) and symbol-error-rate (SER) performance.

Our first learned detector, the fully connected neural network (FCNN), consists of four dense layers with ReLU activations and 20% dropout on each hidden layer. Despite its minimal complexity—approximately 0.05 GMAC per detection block and 0.1 million parameters—the FCNN exhibited robustness under hardware impairments: it outperforms classical ML in severe PA saturation scenarios, maintains competitive BERs for higher-order constellations (16- and 64-QAM), and generalizes well between AWGN and Rayleigh channels. This demonstrates that even shallow networks can learn to mitigate nonlinear distortions when provided sufficient training diversity.

Building on this, our one-dimensional convolutional neural network (CNN) processes the received vector as a 1×3 input image using two convolutional blocks with 64 and 128 filters, batch normalization, and ReLU activation, followed by global average pooling. Operating at roughly 0.45 GMAC and 0.42 million parameters, the CNN consistently reduces the performance gap to ML across a wide E_b/N_0 range and shows resilience against clipping-induced error floors. In Rayleigh fading environments, CNNs maintain graceful BER roll-off, validating their ability to extract localized signal features that represent the channel's statistical behavior.

Furthermore, a residual neural network (ResNet), incorporates multiple residual blocks with skip connections—each containing two Conv1D→BN→ReLU layers. With 0.22 GMAC per block and 0.42 million parameters, the ResNet achieves the steepest BER slopes and lowest error floors, especially under hardware saturation. Among all tested models, ResNet achieves the closest performance to sphere decoding in ideal scenarios and maintains its robustness under saturation, outperforming CNNs and FCNNs by 0.5–1 dB. Dropout regularization (10–20%) proves crucial in deeper ResNet variants: without it, overfitting occurs at low SNRs, yielding BER floors near 10^{-2}; with dropout, performance recovers and the long-tail BER drops to below 3×10^{-4}.

To address channels with memory, such as the 3GPP TDL-C model , we employed a long short-term memory (LSTM) network. Other architectures like FCNN, CNN, and ResNet—which assume memoryless channels — struggled to capture the temporal correlation introduced by TDL-C fading. In contrast, LSTM demonstrated the ability to track the temporal structure and produced stable results across modulation orders. However, it did not consistently outperform classical MMSE, particularly for 64-QAM, where MMSE achieved slightly better BER. This suggests that while memory-aware networks like LSTM are promising for long-delay channels, further tuning and architectural refinement are needed to outperform traditional baselines.

To ensure rigorous benchmarking, we conducted thousands of Monte Carlo trials across modulation orders (4-, 16-, 64-QAM) and E_b/N_0 values (0–30 dB). Detailed evaluations in Chapter 4 confirm that model performance is highly scenario-dependent. Notably, the learned detectors in Figure 4.7 maintained smooth BER curves under scenarios where ML and SD exhibited higher error values, especially in the presence of hardware nonlinearity. While CNN and ResNet approaches SD in performance under clean channels, SD outperformed all learned models in ideal settings.

Conversely, under saturation, the learned models surpassed classical baselines due to their ability to learn nonlinear compensation.

Finally, we evaluated complexity: per-block GMAC counts, parameter sizes, and inference time estimates suggest all three architectures are suitable for real-time deployment. FCNN offers ultra-low complexity for constrained platforms, CNN balances accuracy and runtime, and ResNet delivers the best accuracy-to-cost ratio. Together, these architectures define a tunable complexity–performance spectrum, allowing system designers to select the optimal detector based on deployment constraints and channel conditions.

5.2 Future Work

One natural extension is to move beyond separate equalization and detection stages by designing an end-to-end neural architecture that directly ingests raw IQ samples (including pilot symbols) and outputs bit estimates. By learning both channel inversion and decision boundaries, such a model could reduce overall latency, simplify receiver chains, and potentially improve robustness to modeling mismatches in nonstationary environments.

Extending our approach to wideband, multi-carrier systems is another promising avenue. In OFDM and 5G NR, channel effects vary across time and frequency, suggesting 2D convolutional or transformer-based detectors that operate on time–frequency grids. Future work should investigate how to exploit pilot patterns, subcarrier correlations, and block-fading structures to build more effective detectors for multi-tap, delay-spread channels.

To deploy learned detectors on hardware with stringent energy and latency constraints, adaptive model compression techniques—such as structured pruning, low-bit quantization, and knowledge distillation—must be explored. Algorithms that dynamically adjust network complexity or precision in response to instantaneous channel quality could deliver the best of both worlds: high accuracy when needed and minimal compute under benign conditions.

Real-world wireless channels exhibit continual variation due to mobility, hardware drift, and interference. Incorporating online or transfer-learning mechanisms—ranging from unsupervised fine-tuning on live measurements to meta-learning frameworks for rapid domain adaptation—will

improve resilience to unforeseen impairments and prevent catastrophic forgetting when conditions change.

Hardware prototyping on software-defined radio platforms (e.g. GNU Radio with FPGA/GPU) or ASIC testbeds will provide critical insights into end-to-end inference latency, throughput, power consumption, and quantization effects. Co-designing network architectures alongside RF front-end specifications (ADC/DAC resolution, sampling rates) can further optimize system-level performance and guide practical integration into 5G/6G base stations and user equipment.

Scaling to massive MIMO regimes—with tens or hundreds of antennas—introduces both opportunities and challenges in multi-user detection. Future research should explore scalable graph neural networks, attention mechanisms, or hybrid classical–deep models to jointly detect and decode multiple spatial streams while mitigating pilot contamination and inter-user interference.

Finally, integrating these learned detectors with channel-coding schemes—such as LDPC or polar codes—within iterative or turbo architectures promises further gains in throughput and error resilience. By jointly training encoder, channel model, detector, and decoder in an end-to-end manner, it may be possible to exceed the limits of separate design and unlock new performance frontiers for next-generation wireless systems.

References

[1] Kp, "5G NR: Resource grid: https://howltestuffworks.blogspot.com/2019/10/5g-nr-resource-grid.html."

[2] M. Hao, H. Li, and G. Xu, "Towards efficient and privacy-preserving federated deep learning," in *2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[3] G. S and G. S, "Securing medical image privacy in cloud using deep learning network," *Journal of Cloud Computing*, vol. 12, no. 1, p. 40, Mar. 2023.

[4] X. Kui, F. Liu, M. Yang, H. Wang, C. Liu, D. Huang, Q. Li, L. Chen, and B. Zou, "A review of dose prediction methods for tumor radiation therapy," *Meta-Radiology*, vol. 2, no. 1, p. 100057, Mar. 2024.

[5] D. Quang, N. Quang, Vo, and C. Nguyen, "BeCaked: An explainable artificial intelligence model for COVID-19 forecasting," in *Proceedings of the ICR'22 International Conference on Innovations in Computing Research*, vol. 12, May 2022, pp. 53–64.

[6] M. S. Akbar, Z. Hussain, M. Ikram, Q. Z. Sheng, and S. Mukhopadhyay, "6G survey on challenges, requirements, applications, key enabling technologies, use cases, AI integration issues and security aspects." arXiv, Oct. 2024.

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

[8] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[9] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[10] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[11] S. R. Doha and A. Abdelhadi, "Deep learning in wireless communication receiver: A survey," in *ArXiv*, vol. abs/2501.17184, doha, Qatar, Jan. 2025.

[12] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[13] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[14] K. Noor, A. L. Imoize, C.-T. Li, and C.-Y. Weng, "A review of machine learning and transfer learning strategies for intrusion detection systems in 5G and beyond," *Mathematics*, vol. 13, no. 7, p. 1088, 2025.

[15] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, Mar. 1998.

[16] E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[17] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. United States: Cambridge University Press, Jun. 2008.

[18] Z. Guo and P. Nilsson, "Reduced complexity schnorr-euchner decoding algorithms for MIMO systems," *IEEE Communications Letters*, vol. 8, no. 5, pp. 286–288, May 2004.

[19] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[20] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Transactions on Signal Processing*, vol. 54, Mar. 2006.

[21] B. Marinberg, A. Cohen, E. Ben-Dror, and H. Permuter, "A study on MIMO channel estimation by 2D and 3D convolutional neural networks," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. New Delhi, India: IEEE, 2020, pp. 1–6.

[22] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "Deep learning based beamforming neural networks in downlink MISO systems," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, pp. 1–5.

[23] F. A. Aoudia and J. Hoydis, "End-to-end learning for OFDM: From neural receivers to pilotless communication | IEEE journals & magazine | IEEE xplore," in *IEEE Transactions on Wireless Communications*, vol. 21, Feb. 2022.

[24] T. H. Ahmed, J. J. Tiang, A. Mahmud, C. Gwo Chin, and D.-T. Do, "Deep reinforcement learning-based adaptive beam tracking and resource allocation in 6G vehicular networks with switched beam antennas," *Electronics*, vol. 12, no. 10, p. 2294, Jan. 2023.

[25] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A comprehensive survey of recent advancements in molecular communication," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1887–1919, 2016.

[26] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," in *MITSUBISHI ELECTRIC RESEARCH LABORATORIES*. arXiv, Sep. 2014.

[27] H. Ye, G. Y. Li, and B. H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[28] P. Wang, A. Jiang, X. Liu, J. Shang, and L. Zhang, "LSTM-based EEG classification in motor imagery tasks," *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 26, no. 11, pp. 2086–2095, Nov. 2018.

[29] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020*. Barcelona, Spain.: IEEE, Mar. 2020, pp. 46–50.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: arXiv, Dec. 2015, pp. 770–778.

[31] Burera, S. Ahmed, and S. Kim, "Transformer learning-based efficient MIMO detection method," *Physical Communication*, vol. 70, p. 102637, Jun. 2025.

[32] A. Fuchs, C. Knoll, N. N. Moghadam, A. P. J. Huang, E. Leitinger, and F. Pernkopf, "Self-attention for enhanced OAMP detection in MIMO systems," in *Self-Attention for Enhanced OAMP Detection in MIMO Systems*. Rhodes Island, Greece: arXiv, 2023, pp. 1–5.

[33] J. Lee, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML 2019*, Long Beach, California.

[34] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Making time-frequency domain models great again for monaural speaker separation," in *ICASSP 2023*. Rhodes Island, Greece: IEEE, Mar. 2023, pp. 1–5.

[35] L. Yang, W. Liu, and W. Wang, "TFPSNet: Time-frequency domain path scanning network for speech separation," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore, Singapore: IEEE, 2022, pp. 6842–6846.

[36] A. Kumar, S. Chakravarty, S. Suganya, H. Sharma, R. Pareek, M. Masud, and S. Aljahdali, "Intelligent conventional and proposed hybrid 5G detection techniques," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 10 485–10 494, Dec. 2022.

[37] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[38] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," Aug. 2022.

[39] Y. Ran-Milo, E. Lumbroso, E. Cohen-Karlik, R. Giryes, A. Globerson, and N. Cohen, "Provable benefits of complex parameterizations for structured state space models," in *NeurIPS 2024*, vol. abs/2410.14067. arXiv, Oct. 2024, p. 12.

[40] S. Zhang, M. Zhang, X. Pang, and M. Yang, "No-skim: Towards efficiency robustness evaluation on skimming-based language models," https://arxiv.org/html/2312.09494v2, 2023.

[41] N. Nayak, T. Tholeti, M. Srinivasan, and S. Kalyani, "Green DetNet: Computation and memory efficient DetNet using smart compression and training," *arXiv preprint arXiv:2003.09446*, vol. abs/2003.09446, Apr. 2021.

[42] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep HyperNetwork-based MIMO detection," in *Deep HyperNetwork-Based MIMO Detection*. Atlanta, GA, USA: arXiv, 2020, pp. 1–5.

[43] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations*. arXiv, 2016.

[44] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. Boston, USA: McGraw-Hill Education, 2002.

[45] L. Zheng and D. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[46] 3GPP, "3rd generation partnership project," 3GPP, Technical Specification Group Radio Access Network TR 25.913 V7.3.0 (2006-03), 2006.

[47] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. Da Silva, "Waveform and numerology to support 5G services and requirements," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 90–98, Nov. 2016.

[48] F. Rinaldi, A. Raschellà, and S. Pizzi, "5G NR system design: A concise survey of key features and capabilities," *Wireless Networks*, vol. 27, no. 8, pp. 5173–5188, Nov. 2021.

[49] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. NY, United States: Cambridge University Press, Jul. 2005.

[50] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. River, NJ, United States: Prentice-Hall, Inc., Mar. 1993, vol. 1.

[51] M. K. Simon and M.-S. Alouini, "Digital communications over fading channels (M.K. simon and M.S. alouini; 2005) [book review]," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 3369–3370, Jul. 2008.

[52] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice Hall, 1996.

[53] S. Verdu, *Multiuser Detection*. Princeton University, New Jersey, USA: Cambridge Univeristy Press, Aug. 1998.

[54] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[56] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.

[57] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*. arXiv, Feb. 2019.

[58] A. Klautau, P. Batista, N. Gonzalez-Prelcic, Y. Wang, and R. W. H. Jr, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*. San Diego, CA, USA: arXiv, Jun. 2021, pp. 1–9.

[59] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An open-source library for next-generation physical layer research," in *arXiv:2203.11854*, vol. abs/2203.11854. arXiv, 2022.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The 3rd International Conference for Learning Representations*, vol. 9. San Diego: arXiv, 2015.

[61] A. Goldsmith, *Wireless Communications*. Stanford University, California: Cambridge University Press, 2005.