

CNN-BASED SIGNAL DETECTION
FOR 6G ULTRA-MASSIVE MIMO SYSTEMS



A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Telecommunication
and Computer Engineering
Suranaree University of Technology
Academic Year 2023

การตรวจจับสัญญาณโดยใช้การเรียนรู้เชิงลึกแบบ CNN
ในระบบ 6G สำหรับไมโมขนานใหญ่พิเศษ



นายชิตภณ แก้วอินทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2566

**CNN-BASED SIGNAL DETECTION
FOR 6G ULTRA-MASSIVE MIMO SYSTEMS**

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for The Degree of Doctor of Philosophy.

Thesis Examining Committee



.....
(Assoc. Prof. Dr. Pichaya Chaipanya)

Chairperson



.....
(Assoc. Prof. Dr. Peerapong Uthansakul)

Member (Thesis Advisor)



.....
(Asst. Prof. Dr. Apinya Deekaikam)

Member (Thesis Co-Advisor)



.....
(Assoc. Prof. Dr. Monthippa Uthansakul)

Member



.....
(Assoc. Prof. Dr. Piyaporn Mesawad)

Member



.....
(Dr. Dheerasak Anantakul)

Member



.....
(Assoc. Prof. Dr. Pornsiri Jongkol)

Dean of Institute of Engineering



.....
(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)

Vice Rector for Academic Affairs
and Quality Assurance

ชิตภณ แก้วอินทร์ : การตรวจจับสัญญาณโดยใช้การเรียนรู้เชิงลึกแบบ CNN ในระบบ 6G สำหรับไมโมขนานใหญ่พิเศษ (CNN-BASED SIGNAL DETECTION FOR 6G ULTRA-MASSIVE MIMO SYSTEMS)

อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.พีระพงษ์ อุฑารสกุล 144 หน้า

คำสำคัญ: การตรวจจับสัญญาณ/ระบบไมโมขนานใหญ่พิเศษ/การเรียนรู้เชิงลึก/การเรียนรู้ของเครื่อง

การเปลี่ยนแปลงของเทคโนโลยีสื่อสาร มีบทบาทสำคัญต่อเครือข่าย 5G และ 6G ในอนาคต เพื่อการตอบสนองความต้องการที่เพิ่มขึ้นสำหรับการส่งข้อมูลความเร็วสูง โดยเฉพาะการใช้คลื่นความถี่มิลลิเมตร (mmWave) ใน ยุคที่ 5 (5G) ที่ช่วยปรับปรุงการส่งและรับข้อมูลจำนวนมากด้วยระบบไมโมขนานใหญ่ (Massive MIMO) สำหรับอนาคต วิทยานิพนธ์นี้ศึกษาเทคโนโลยีการส่งข้อมูลที่เร็วขึ้นและแม่นยำในเครือข่าย ยุคที่ 6 (6G) ใช้ระบบไมโมขนานใหญ่พิเศษ (Ultra-Massive MIMO :UM-MIMO) ในคลื่นความถี่ 0.3 เทราเฮิรต (THz) เพื่อออกแบบระบบสื่อสารจำลอง

วิทยานิพนธ์นี้เน้นการประยุกต์ใช้เครือข่ายประสาทเทียมแบบคอนโวลูชันหลายขนาดขนาน (PMS-CNN) ในการตรวจจับสัญญาณในระบบ UM-MIMO เพื่อเปรียบเทียบประสิทธิภาพกับเครือข่ายประสาทเทียมแบบคอนโวลูชันหน่วยความจำยาวระยะสั้น (CNN-LSTM) และกลุ่มการเรียนรู้ของเครื่องได้แก่ การเรียนรู้ของเครื่องแบบสุดขีด (Extreme Learning Machine: ELM), การเรียนรู้ของเครื่องแบบสุดขีดที่ผ่านการปรับแต่ง (Regularized Extreme Learning Machine: RELM) และการเรียนรู้ของเครื่องแบบสุดขีดที่ทนทานต่อค่าผิดปกติ (Outlier-Robust Extreme Learning Machine: ORLEM) รวมถึงใช้เทคนิคการตรวจจับสัญญาณขั้นสูง การบังคับให้เป็นศูนย์ (Zero Forcing :ZF) และวิธีการทำให้ค่าความคลาดเคลื่อนกำลังสองเฉลี่ยขั้นต่ำ (Mean Square Error: MMSE) เพื่อลดการรบกวนและเพิ่มความแม่นยำในการส่งข้อมูล ผลที่เกิดขึ้นแสดงให้เห็นว่าเครือข่ายประสาทเทียมแบบคอนโวลูชันหลายขนาดขนานสำหรับการตรวจจับสัญญาณ PMS-CNN มีประสิทธิภาพเหนือกว่าวิธีการ CNN-LSTM, ELM, RELM, ORLEM, ZF และ MMSE โดยเฉพาะในแง่ของความจุช่องสัญญาณและอัตราบิดผิดพลาด รวมถึงความซับซ้อนเชิงคำนวณที่ลดลง

สาขาวิชา วิศวกรรมโทรคมนาคม

ปีการศึกษา 2566

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

CHITTAPON KEAWIN: CNN-BASED SIGNAL DETECTION FOR 6G ULTRA-MASSIVE MIMO SYSTEMS. THESIS ADVISOR: ASSOC. PROF. PEERAPONG UTHANSAKUL, 144 PP.

Keyword: SIGNAL DETECTION/ULTRA-MASSIVE MIMO/DEEP LEARNING/MACHINE-LEARNING

The transformation of communication technology plays a crucial role in future fifth generation (5G) and sixth generation (6G) networks to meet the increasing demand for high-speed data transmission. Specifically, the use of millimeter wave (mmWave) frequencies in the 5G helps improve the transmission and reception of large volumes of data with Massive MIMO systems. For the future, this thesis studies faster and more accurate data transmission technology in 6G networks using Ultra-Massive MIMO (UM-MIMO) systems at 0.3 Terahertz (THz) frequencies to design a simulated communication system.

This thesis emphasizes the application of Parallel Multi-Scale Convolutional Neural Networks (PMS-CNN) for signal detection in UM-MIMO systems to compare its performance with Convolutional Neural Networks with Long Short-Term Memory (CNN-LSTM) and machine learning groups, including Extreme Learning Machine (ELM), Regularized Extreme Learning Machine (RELM), and Outlier-Robust Extreme Learning Machine (ORLEM). Additionally, advanced signal detection techniques such as Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) are used to reduce interference and increase data transmission accuracy. The results indicate that the PMS-CNN for signal detection outperforms CNN-LSTM, ELM, RELM, ORLEM, ZF, and MMSE methods, particularly in terms of channel capacity and bit error rate, while also offering reduced computational complexity.

School of Telecommunication Engineering
Academic Year 2023

Student's Signature

Advisor's Signature

Co-Advisor's Signature

ACKNOWLEDGEMENT

The author wishes to acknowledge the financial support from Suranaree University of Technology (SUT), which made this research possible.

I would like to express my sincere thanks to my thesis advisors, Assoc. Prof. Dr. Peerapong Uthansakul and Asst. Prof. Dr. Apinya Deekaikam for their consistent supervision, guidance, and insightful comments on various aspects of the research and thesis writing process. Their advice has been crucial in the successful completion of this study.

My thanks also go to Assoc. Prof. Dr. Piyaporn Mesawad, Assoc. Prof. Dr. Pichaya Chaipanya, Assoc. Prof. Dr. Monthippa Uthansakul, and Dr. Dheerasak Anantakul for their valuable suggestions and guidance provided as members of the examination committee.

The author is also grateful to all faculty and staff members of the School of Telecommunication Engineering and colleagues for their help and support throughout the period of this study at this university.

Finally, I would like to express my deep sense of gratitude to the scholarship program, my parents, and my advisor for their support and encouragement throughout the course of this study at Suranaree University of Technology.

Chittapon Keawin

TABLE OF CONTENTS

	Page
ABSTRACT (THAI)	I
ABSTRACT (ENGLISH)	III
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS	IV
LIST OF TABLES	VII
LIST OF FIGURES	VIII
LIST OF ABBREVIATIONS	0
CHAPTER	
I INTRODUCTION	1
1.1 Background and problem statement	1
1.2 Thesis objectives	2
1.3 Scope and limitation of the thesis	2
1.4 Contributions.....	3
1.5 Organization of the thesis.....	3
1.6 Summary	4
II THEORETICAL BACKGROUND	6
2.1 Introduction	6
2.2 Communication Technologies from 5G to 6G.....	7
2.3 Massive MIMO Technology and Ultra-Massive MIMO	9
2.4 Channel Model in UM-MIMO for signal detection.....	16
2.5 Signal Detection Method	20
2.6 Machine Learning (ML)	26
2.7 Deep Learning (DL).....	33

TABLE OF CONTENTS (Continued)

	Page
2.8 Aspects of Comparative Analysis	35
2.9 Literature Review.....	46
2.10 Application of Services 5G and 6G	58
2.11 Summary	61
III METHODOLOGY	63
3.1 Introduction.....	63
3.2 System Model.....	63
3.3 Channel Model	65
3.4 Data preparation.....	66
3.5 Convolutional Neural Network (CNN).....	69
3.6 Parallel Multi-Scale Convolutional Neural Network (PMS-CNN).....	71
3.7 Extreme Learning Machine (ELM) Techniques.....	74
3.8 Regularized Extreme Learning Machine (RELM) Techniques	78
3.9 Outliner-Robust Extreme Learning Machine (ORELM) Techniques	79
3.10 Application of Techniques to Signal Detection.....	81
3.11 Mean Squared Error (MSE) Performance Comparison Method.....	84
3.12 Bit Error Rate (BER) Performance Comparison Method	85
3.13 Channel Capacity and Outage Probability Comparison Method.....	87
3.14 Summary	91
IV RESULTS AND DISCUSSIONS.....	92
4.1 Introduction	92
4.2 Evaluation of Algorithm Performance Mean Squared Error (MSE)	92

TABLE OF CONTENTS (Continued)

	Page
4.3 Bit Error Rate Performance (BER).....	96
4.4 Channel Capacity	101
4.5 Outage Probability.....	103
4.6 Computational Complexity with Big O notation.....	105
4.7 Summary	115
V CONCLUSIONS	116
5.1 Conclusions	116
5.1 Future works.....	118
REFERENCES	119
APPENDIX.....	126
PUBLICATIONS	126
BIOGRAPHY	144

LIST OF TABLES

Table	Page
2.1 Comparison of the Performance of Classification and Data Processing.....	30
2.2 Performance Metrics: Classification and Regression Loss.....	35
2.3 Calculation of BER for the different modulation schemes.....	43
2.4 Computational Complexity Applications.....	45
2.6 The standard Bit Error Rate (BER) is determined by the requirements of the application and the Quality of Service (QoS).....	58
2.7 Services 5G and 6G with BER.....	60
2.8 Survey of research related to the proposed method.....	62
4.1 Big O Complexity Analysis for ZF.....	105
4.2 Big O Complexity Analysis for MMSE.....	106
4.3 Big O Complexity Analysis for ELM.....	106
4.4 Big O Complexity Analysis for RELM.....	107
4.5 Big O Complexity Analysis for ORELM.....	107
4.6 Big O Complexity Analysis for CNN-LSTM.....	109
4.7 Big O Complexity Analysis for PMS-CNN.....	110
4.8 The Computational Complexity by Big O notation. By $M, N, \tilde{N}, n = 10, c = 32, k = 5, k_2, k_3, k_4, k_5, k_6 = 3, 5, 7, 9, 12, p = 32, d = 509, f_1 = 2024,$ and $n_classes = 10$	111
4.9 Big O Complexity with Latency 6G.....	114
5.1 Summary of Services and Applications for 5G and 6G.....	117

LIST OF FIGURES

Figure	Page
1.1 The scenario for Ultra-Massive MIMO.....	1
2.1 Potential Capabilities of 6G Compared to 5G.....	8
2.2 The structure of element antennas in Massive MIMO.....	10
2.3 Implementing antennas in Massive MIMO and UM-MIMO.....	11
2.4 Spectral Efficiency vs. Energy Efficiency in SISO, MISO, and Massive MIMO Systems with Different Processing Methods.....	13
2.5 Promising Techniques for 6G Wireless Networks.....	13
2.6 The simple hybrid beamforming.....	14
2.7 The Structure of Hybrid Beamforming.....	15
2.8 The simple of structure communication.....	16
2.9 Schematic model of a Rayleigh fading channel (Orten, 2002).....	19
2.10 Types of Supervised Learning.....	27
2.11 Example Graph of Accidents and Population.....	28
2.12 Data Classification (a).....	29
2.13 Learning from Data to Create a Classifier (b).....	30
2.14 Clustering Data.....	32
2.15 Visualization of Dimensionality Reduction for Class Separation.....	32
2.16 Basic Structure of Deep Learning.....	33
2.17 Plot of MSE Loss (Y-axis) vs. Predictions (X-axis).....	37
2.18 Plot of MAE Loss (Y-axis) vs. Predictions (X-axis).....	38
2.19 Plot of Huber Loss (Y-axis) vs. Predictions (X-axis).....	39
2.20 Plot of Log-Cosh Loss (Y-axis) vs. Predictions (X-axis).....	41
2.21 the Bit Error Rate (BER) versus E_b / N_0 for various modulation schemes.....	44

LIST OF FIGURES (Continued)

Figure	Page
2.22 The amount of research on UM-MIMO by publication year. Updated on 24 May 2022.....	47
2.23 Array-of-subarray antenna structure.	48
2.24 BER performance of SM (solid lines) and SMX (dotted lines): $f = 1$ THz, $D = 1$ m, $M = 8$, $Q = 32$ and $ X = 64$	49
2.25 An illustration of the deep kernel learning architecture in the Gaussian process regression (Nie, 2019).	50
2.26 Simulation results for channel estimation using LS, MMSE, and DKL.....	51
Estimators.....	51
2.27 Architecture of the proposed 1D CNN-LSTM-based (Murshed, 2022).	52
2.28 SE of different algorithms for varying transmit power with $N_t = N_r = 256$ and $N_s = N_{RF} = 5$. (Murshed, 2022).	53
2.29 The structure of the proposed DCNN network.	55
2.30 NMSE performance of different CE methods.....	55
2.31 BER performance of IGA compared with AMP, EP and LMMSE under 4-QAM.	57
2.32 Training and inference for application to using.....	60
3.1 UM-MIMO system model.....	64
3.2 Ray/cluster-based UM-MIMO channel model schematic.	66
3.3 QAM Constellation Diagram.....	67
3.4 Workflow of the process.	68
3.5 The Process about of Separation of real and imaginary parts.....	68
3.6 CNN Model Architecture.....	69
3.7 The proposed PMS-CNN architecture.	71

LIST OF FIGURES (Continued)

Figure	Page
3.8 The Neural System Structure of Extreme Learning Machine.....	75
3.9 Simple Data Encoding and Decoding Process.....	86
4.1 The MSE Performance vs SNR [dB] PMS-CNN, ZF, MMSE	93
4.2 The MSE evaluation of Algorithm Performance between CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.....	94
4.3 The MSE evaluation of Algorithm Performance between ZF, MMSE CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.....	95
4.4 The BER of different Modulation in Ultra Massive MIMO 256x256.....	97
4.5 The BER of Algorithm Performance between ZF, MMSE and PMS-CNN method..	98
4.6 The BER of Algorithm Performance CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.....	99
4.7 The BER of Algorithm Performance ZF, MMSE, CNN-LSTM, ELM, RELM, ORELM and PMS-CNN method.....	100
4.8 The Channel Capacity (Mbps) vs SNR [dB]	102
4.9 Outage Probability vs Channel Capacity (bps/Hz).....	104
4.10 Big O Complexity Chart CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.	112
4.11 Big O Complexity Growth by Model CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.....	113
5.1 Application for Next communication networks (Thantharate, 2019).	118

LIST OF ABBREVIATIONS

MSE	=	Mean Square Error
UM-MIMO	=	Ultra-Massive MIMO
MIMO	=	Multiple Input Multiple Output
RELM	=	Regularized Extreme Learning Machine
ELM	=	Extreme Learning Machine
ORLEM	=	Outlier-Robust Extreme Learning Machine
CNN	=	Convolutional Neural Network
DNN	=	Deep Neural Networks
PMS	=	Parallel Multi Scale
ML	=	Machine Learning
DL	=	Deep Learning
mmWave	=	Millimeter wave
GHz	=	Gigahertz
NLOS	=	Non-Line of Sight
THz	=	Terahertz
SNR	=	Signal to Noise Ratio
ZF	=	Zero Forcing
MMSE	=	Minimum Mean Square Error
QAM	=	Quadrature Amplitude Modulation
BER	=	Bit Error Rate
SINR	=	Signal Interference to Noise Ratio
5G	=	Fifth Generation
6G	=	Six Generation
LSTM	=	Long Short-Term Memory
SM	=	Spatial Multiplexing
HB	=	Hybrid Beamforming

CHAPTER I

INTRODUCTION

1.1 Background and problem statement

This highlights the relentless evolution of communication technology, emphasizing the pivotal roles of 5G and emerging 6G networks in meeting the growing demand for high-speed and accurate data transmission. Despite the advancements of 5G, particularly with millimeter wave (mmWave) frequencies and Massive MIMO systems, there are limitations in scenarios requiring even faster and more precise data transmission, such as remote-control mechanisms, to address these constraints, the exploration has shifted towards 6G, which introduces the UM-MIMO architecture. In Figure 1.1 show structure of the scenario for Ultra-Massive MIMO, this advancement significantly increases antenna volume for both transmission and reception, utilizing Terahertz (THz) band frequencies ranging from 0.3 to 10 THz, bridging mmWave and THz bands for enhanced performance.

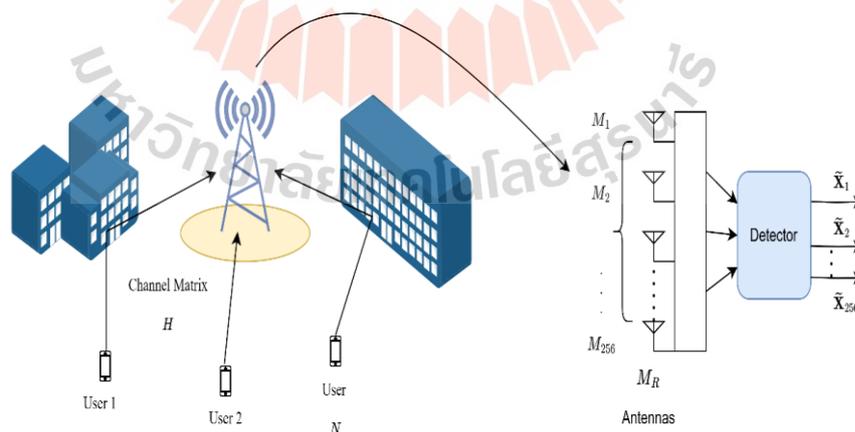


Figure 1.1 The scenario for Ultra-Massive MIMO

In infrastructure, techniques like Hybrid Precoding, Beamforming, and Spatial Multiplexing have been proposed to optimize signal transmission and improve the Signal-to-Noise Ratio (SNR). However, the functional efficacy of these systems hinges on advanced signal detection strategies. Techniques such as ZF and MMSE are crucial for enhancing data transmission speed and precision while reducing discrepancies. This research delves into advanced signal detection methods and their integration with deep learning within the UM-MIMO framework, posited as foundational for 6G technology. It explores the application of Parallel Multi-Scale Convolutional Neural Networks (PMS-CNN) for signal detection and compares its performance with CNN-LSTM and machine learning methods like ELM, RELM, and ORLEM used in Massive MIMO. The comparative analysis demonstrates that PMS-CNN outperforms these methods, particularly in terms of channel capacity and bit error rate, while offering better computational efficiency and reduced complexity, making it suitable for future expansive MIMO systems.

1.2 Thesis objectives

1.2.1 To design a new convolutional neural network model for signal detection in UM-MIMO system.

1.2.2 To compare the algorithmic performance in signal detection with other methods in deep learning and machine learning.

1.3 Scope and limitation of the thesis

1.3.1 In this thesis focuses only on spatial multiplexing for signal detection in UM-MIMO systems.

1.3.2 Training and evaluating the model are based on 256 transmit and 256 receive antennas and 256QAM modulation. Because industry can provide the number of antennas like now.

1.3.3 The proposed method is compared with ZF, MMSE traditional methods and ELM, RELM, ORELM machine learning algorithms, as well as the CNN-LSTM deep learning algorithm for signal detection, to ensure the reliability of the proposed method and demonstrate better performance.

1.4 Contributions

1.4.1 Propose PMS-CNN, a parallel multi-scale deep neural network model applied to signal detection in UM-MIMO systems.

1.4.2 For signal detection, PMS-CNN provides results that can be effectively applied to 5G and 6G networks.

1.4.3 PMS-CNN in the signal detection system can directly improve the efficiency of data transmission. Instead of using channel estimation methods that is more complicated.

1.5 Organization of the thesis

In various sections of the thesis, the outlined can be explained as follows: Chapter II reviews existing research related to this topic by outlining fundamental principles relevant to this study within the current body of knowledge, particularly in the context of signal detection for performance enhancement through the application of deep learning techniques. The focus is on theories, models, and significant findings from previous studies concerning UM-MIMO signal detection, machine learning, and deep learning.

Chapter II focuses on explaining the methods used to design the communication model in the system. UM-MIMO and deep learning for signal detection This chapter has a special focus on deep learning where models are designed to match signal detection. This chapter explores the various traditional

methods and machine learning techniques used to train and build these models. and provide insights into the results.

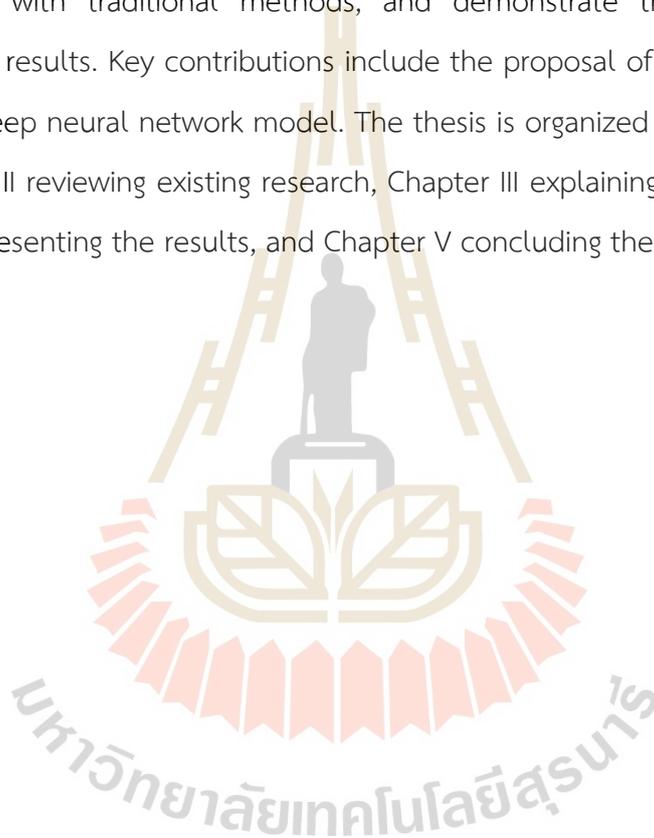
Chapter IV presents the results of the developed model. along with a detailed discussion the model results are divided into five sub-sections for analysis in terms of communication systems. The first subsection shows Evaluation of Algorithm Performance through Mean Squared Error (MSE) Computation. The second subsection shows Bit Error Rate Performance, giving us an idea of how reliably the system can transmit data. A low BER rate means less errors. Make communication more reliable the third subsection performs a comparative analysis of Channel Capacity performance, indicating the maximum rate at which data can be transmitted over a communication channel with the least amount of errors. The fourth subsection, Outage Probability, tells us the likelihood that the system will not be able to provide the desired quality of service. This measure helps evaluate the reliability of communication in different conditions. Finally, the fifth subsection Computational Complexity with Big O notation allows the performance of different algorithms to be measured and compared, independent of the hardware or software being used. This helps to understand how the algorithm works in the case of large data sizes.

Chapter V serves as the conclusion of this thesis. It summarizes the key findings and presents the conclusions drawn from the thesis. Additionally, this chapter provides valuable suggestions and suggestions for future studies.

1.6 Summary

In the current era, communication technology is advancing rapidly, making it a crucial aspect of daily life. Despite significant advancements like 5G and its Massive MIMO technology, which uses numerous antennas for data transmission, limitations persist in meeting the demands for high-speed and accurate data transmission, particularly in applications such as remote-control mechanisms. This has prompted research into 6G and Ultra-Massive MIMO, which utilizes Terahertz frequencies for

even greater antenna volume and enhanced performance. Within the 5G framework, techniques like Hybrid Precoding and Beamforming have been developed to optimize signal transmission, yet challenges remain. Effective signal detection strategies, such as ZF and MMSE, have motivated further exploration into deep learning applications within UM-MIMO systems. This thesis aims to investigate and propose new deep neural network models for signal detection, compare their performance with traditional methods, and demonstrate their efficacy through experimental results. Key contributions include the proposal of a Parallel Multi-Scale (PMS-CNN) deep neural network model. The thesis is organized into several chapters, with Chapter II reviewing existing research, Chapter III explaining the design methods, Chapter IV presenting the results, and Chapter V concluding the study.



CHAPTER II

THEORETICAL BACKGROUND

2.1 Introduction

This chapter discusses the literature review and theories related to communication technology in 2020, highlighting significant advancements. One notable development is the use of Multiple Input Multiple Output (MIMO) technology, which involves transmitting and receiving data through multiple antennas. This approach has been widely implemented and researched due to its potential to significantly increase data transmission rates and system capacity. The evolution of MIMO technology has led to the emergence of Massive MIMO, where a large number of antennas are used to improve performance. This technique has been the subject of extensive research as it addresses the growing demand for higher data rates and increased capacity in wireless communication systems. Various studies have explored its applications and benefits, emphasizing its role in enhancing communication efficiency and reliability.

Looking forward to future communication systems, UM-MIMO is anticipated to play a critical role in the development of 6G technology. This advanced form of MIMO technology aims to further boost data transmission capabilities by incorporating an even larger number of antennas. UM-MIMO is expected to be a key component in 6G systems, offering significant improvements in both physical and conceptual aspects of data transmission. This chapter reviews the latest research on Ultra-Massive MIMO, focusing on its potential to overcome existing limitations and introduce innovative solutions in wireless communication. Additionally, the chapter delves into various signal detection techniques essential for the effective implementation of MIMO systems, such as ZF, MMSE, and Deep Learning methods.

These techniques are crucial for enhancing signal accuracy and overall system performance, paving the way for more efficient and robust wireless communication networks.

2.2 Communication Technologies from 5G to 6G

The transition from 5G to 6G communication technologies represents a significant leap in the evolution of wireless communication systems. While 5G has brought about remarkable improvements in speed, latency, and connectivity, 6G is poised to push these boundaries even further. This section explores the key advancements, features, and potential impacts of 5G and 6G technologies, incorporating detailed comparisons.

2.2.1 5G Technology

5G technology, the fifth generation of mobile networks, has revolutionized the way we communicate by offering enhanced data rates, reduced latency, and improved connectivity. With speeds up to 100 times faster than 4G, 5G enables seamless streaming of high-definition video, real-time gaming, and the proliferation of the Internet of Things (IoT). Key features of 5G include.

Enhanced Mobile Broadband (eMBB): Provides significantly higher data rates, supporting applications such as virtual reality (VR) and augmented reality (AR).

Ultra-Reliable Low Latency Communications (URLLC): Ensures minimal delay, critical for applications like autonomous driving and remote surgery.

Massive Machine-Type Communications (mMTC): Connects a vast number of IoT devices, facilitating smart cities and industrial automation.

5G networks utilize a range of frequencies, including sub-6GHz and millimeter wave (mmWave) bands, to achieve these advancements. Despite these improvements, 5G also faces challenges such as limited coverage in rural areas and high deployment costs. Research and development efforts continue to address these issues, aiming to maximize the potential of 5G technology.

2.2.2 6G Technology

6G technology is currently in the research and development phase, with expectations to deliver even higher data rates and lower latency than 5G, along with supporting greater capacity and more connected devices. It aims to make communications more reliable and secure, enabling new use cases such as holographic communication, artificial intelligence (AI), and autonomous vehicles. 6G is set to incorporate new technologies and frequencies, such as Terahertz (THz) bands, to achieve these goals.

Major factors	6G	5G
Peak data rate	> 100Gb/s	10[20] Gb/s
User experience data rate	> 10Gb/s	1Gb/s
Traffic density	> 100Tb/s/km ²	10Tb/s/km ²
Connection density	> 10million/km ²	1million/km ²
Delay	< 1ms	ms level
Mobility	> 1000km/h	350km/h
Spectrum efficiency	> 3x relative to 5G	3~5x relative to 4G
Energy efficiency	> 10x relative to 5G	1000x relative to 4G
Coverage percent	> 99%	About 70%
Reliability	> 99.999%	About 99.9%
Positioning precision	Centimeter level	Meter level
Receiver sensitivity	< -130dBm	About -120dBm

Figure 2.1 Potential Capabilities of 6G Compared to 5G

In Figure 2.1, which compares the potential capabilities of 6G with 5G (Chen, 2020), several key performance indicators (KPIs) highlight the significant improvements expected with 6G. Peak Data Rate: 6G can achieve peak data rates exceeding 100 Gb/s, compared to 5G's peak rates of 10-20 Gb/s. User Experience

Data Rate: 6G aims for user experience data rates over 10 Gb/s, far surpassing 5G's 1 Gb/s. Traffic Density: 6G is expected to handle traffic densities greater than 100 Tb/s/km², compared to 5G's 10 Tb/s/km². Connection Density: 6G will support connection densities over 10 million devices per square kilometer, significantly higher than 5G's 1 million devices per square kilometer. Latency: 6G aims to reduce latency to less than 1 millisecond, compared to the millisecond-level latency in 5G. Mobility: 6G will support mobility speeds up to 1000 km/h, while 5G supports up to 350 km/h. Spectrum Efficiency: 6G will offer spectrum efficiency improvements of more than 3 times that of 5G. Energy Efficiency: 6G is expected to be over 10 times more energy efficient than 5G. Coverage: 6G aims for coverage percentages greater than 99%, compared to 5G's approximately 70%. Reliability: 6G will improve reliability to more than 99.999%, compared to 5G's approximately 99.9%. Positioning Precision: 6G will achieve centimeter-level positioning precision, improving upon 5G's meter-level precision. Receiver Sensitivity: 6G will enhance receiver sensitivity to better than -130 dBm, compared to 5G's approximately -120 dBm. In summary, the progression from 5G to 6G represents a transformative journey in the field of communication technology, while 5G has set new standards for speed and connectivity, 6G is expected to introduce groundbreaking innovations that will reshape how we interact with the digital world. This section delves into the current state and prospects of these technologies, highlighting the ongoing research and development efforts that aim to overcome existing challenges and unlock new possibilities.

2.3 Massive MIMO Technology and Ultra-Massive MIMO

Massive MIMO technology has been a cornerstone in advancing wireless communication systems, particularly in the context of 5G networks. This technology leverages many antennas at the base station to significantly improve data rates, capacity, and reliability. As we transition towards 6G, the concept of UM-MIMO is emerging, promising even greater enhancements. Massive MIMO involves the use of dozens to hundreds of antennas at the base station to serve multiple users

simultaneously. This technology offers several key benefits. **Increased Capacity:** By spatially multiplexing multiple data streams, Massive MIMO can serve many users at the same time, greatly enhancing network capacity. **Improved Spectral Efficiency:** With precise beamforming, Massive MIMO can focus energy where it is needed, improving spectral efficiency and reducing interference. The many antennas provides redundancy, making the system more robust against signal fading and interference. **Energy Efficiency,** Massive MIMO can achieve high data rates with lower transmit power, contributing to overall energy efficiency. Massive MIMO has been instrumental in achieving the high data rates and low latencies characteristic of 5G networks. However, as the demand for wireless communication continues to grow, further advancements are necessary. The current research on Massive MIMO often assumes the use of many antennas at the base station, which significantly exceeds the number of active users. Linear processing methods, such as Maximum Ratio Combining (MRC) for uplink and Maximum Ratio Transmission (MRT) for downlink, are often considered optimal for single-antenna users. These methods can effectively mitigate uncorrelated noise and intra-cell interference, as large numbers of antennas cause the channel matrices to tend to orthogonality, simplifying spatial multiplexing and providing degrees of freedom for energy-efficient and high-gain RF transmission.

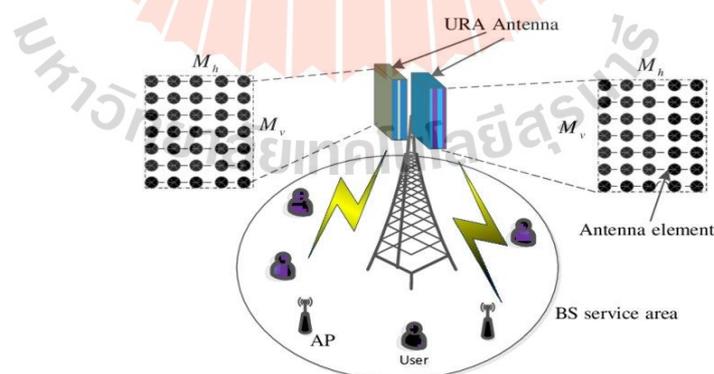


Figure 2.2 The structure of element antennas in Massive MIMO.

Massive MIMO also demonstrates high energy efficiency compared to traditional wireless systems. As shown in Figure 2.2, increasing the number of antennas at the base station improves beamforming accuracy, reducing radiated

power while maintaining overall performance. Doubling the number of antennas allows for a 3 dB reduction in transmission power. In the uplink, coherent beamforming increases array gain, reducing individual user transmission power.

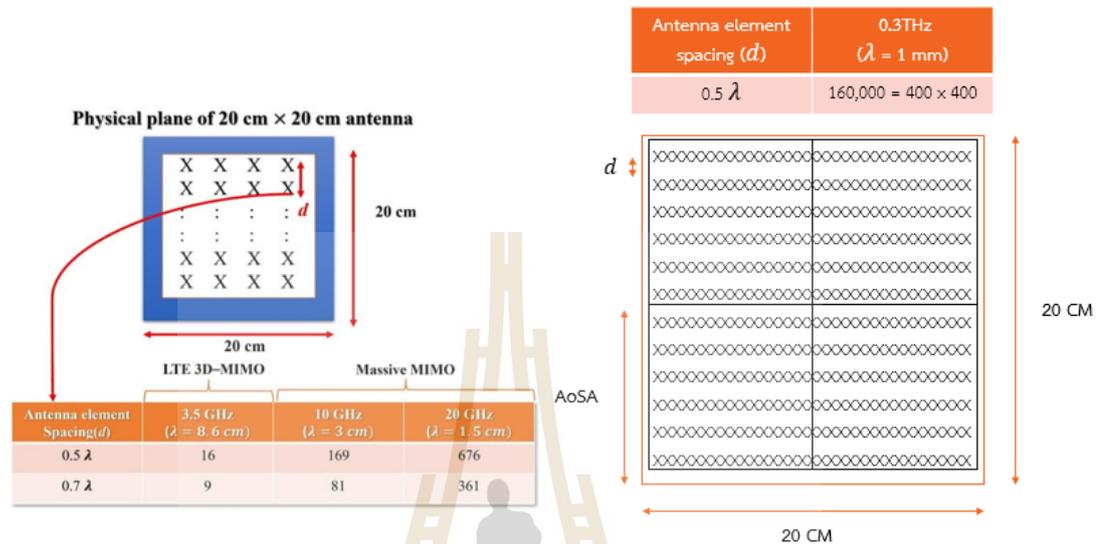


Figure 2.3 Implementing antennas in Massive MIMO and UM-MIMO.

In Figure 2.3, calculating the spacing between elements of an antenna array for a Uniform Rectangular Array (URA), we first need to determine the wavelength (λ). This λ is then used to calculate the spacing (d) between the elements. The λ can be calculated using the following equation.

$$\lambda = \frac{c}{f} \quad (2.1)$$

f is frequency using, c is the speed of light.

The basic equation for calculating the spacing between elements of an antenna array can be derived from the principles of a Uniform Linear Array (ULA) as follows.

$$d = \frac{\lambda}{2} \quad (2.2)$$

Calculating the spacing between elements in a URA, the equation is similar to that of a ULA. However, since the arrangement is in a two-dimensional grid (along both the x and y axes), the spacing between elements must be considered in both axes. The element area A_e is calculated as the product of the horizontal and vertical spacings.

$$A_e = d_x \times d_y \quad (2.3)$$

If greater spacing is required to avoid mutual coupling, the spacing d can be set to λ or less, depending on the design and system requirements.

As we look towards 6G, UM-MIMO is expected to be a key technology. UM-MIMO takes the principles of Massive MIMO to the next level by utilizing an even larger number of antennas, potentially in the range of thousands. This evolution aims to address the increasing demands for data and connectivity in the following ways. Higher Data Rates: With a greater number of antennas, UM-MIMO can support higher peak data rates, essential for applications such as real-time holographic communication and ultra-high-definition video streaming. Lower Latency: The enhanced beamforming and signal processing capabilities of UM-MIMO can further reduce latency, supporting time-sensitive applications like autonomous vehicles and industrial automation. Increased Connection Density: UM-MIMO can handle a higher density of connected devices, facilitating the expansion of the Internet of Things (IoT) and smart cities. Improved Spectral and Energy Efficiency: By leveraging advanced beamforming techniques and more precise spatial multiplexing, UM-MIMO can achieve superior spectral efficiency and reduce overall energy consumption.

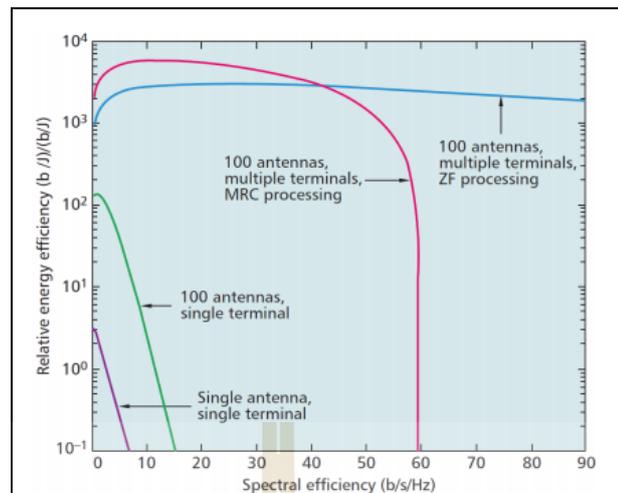


Figure 2.4 Spectral Efficiency vs. Energy Efficiency in SISO, MISO, and Massive MIMO Systems with Different Processing Methods.

In the Figure 2.4, shows the spectral efficiency versus energy efficiency in different MIMO systems with various processing methods, illustrating the superior performance of Massive MIMO over SISO and MISO systems.

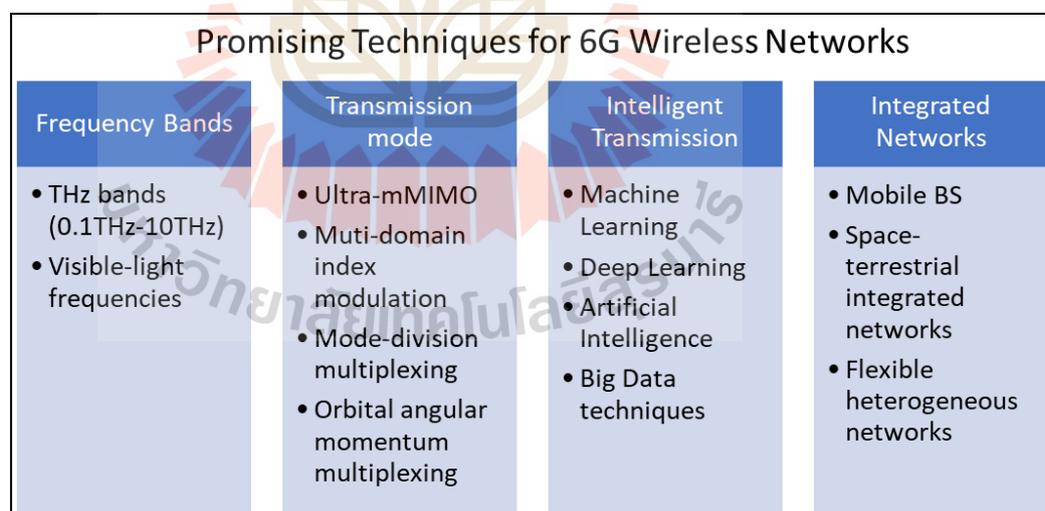


Figure 2.5 Promising Techniques for 6G Wireless Networks

The development of UM-MIMO for future 6G networks includes several promising techniques, categorized into four main areas: Frequency Bands, Transmission Mode, Intelligent Transmission, and Integrated Networks, as shown in Figure 2.5. For

example, 6G will incorporate Terahertz bands (0.1 THz to 10 THz) and utilize UM-MIMO technology, considering a higher number of antennas. Research focuses on hybrid beamforming, hybrid precoding, and spatial multiplexing to achieve optimal performance (Dilli, 2021).

2.3.1 Hybrid beamforming

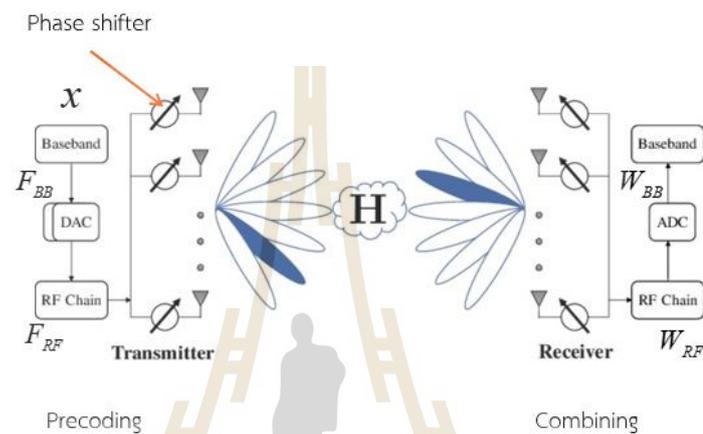


Figure 2.6 The simple hybrid beamforming.

In Figure 2.6, Hybrid beamforming is a technique used in wireless communication systems to optimize the signal-to-noise ratio (SNR) at the receiver. It combines beamforming, which adjusts the signal strength in different directions, with digital precoding, which modifies the transmitted signal using digital signal processing techniques. The goal is to improve the performance of wireless communication systems, especially those with limited RF chains. Traditional digital precoding requires many RF chains, which can be expensive and complex. Hybrid beamforming reduces the number of required RF chains by performing precoding in both analog and digital domains. In the analog domain, hybrid precoding uses beamforming to adjust the phase and amplitude of signals at each antenna, while digital signal processing techniques optimize the signals in the digital domain to improve SNR at the receiver. This approach achieves high performance with lower cost and complexity shown in Figure 2.7.

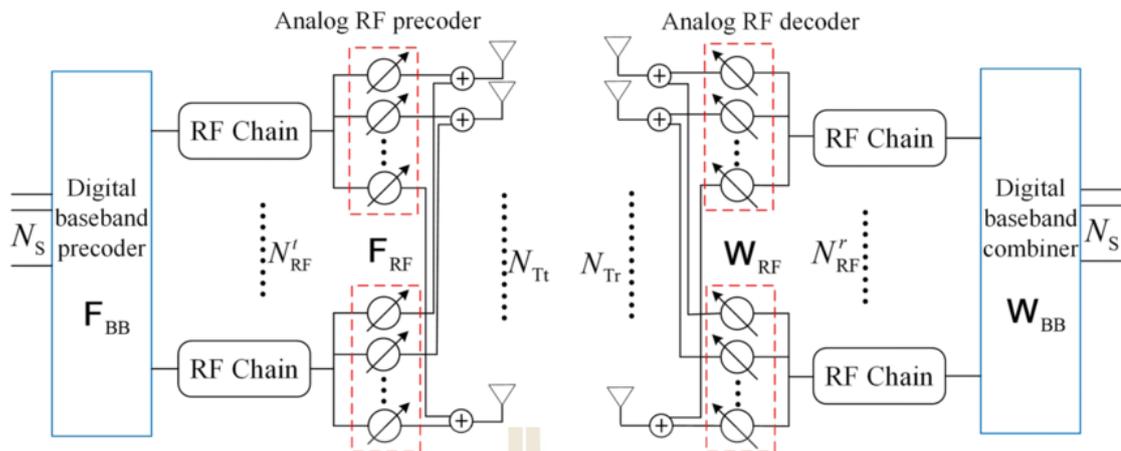


Figure 2.7 The Structure of Hybrid Beamforming.

Hybrid precoding can be applied in various wireless communication systems, including cellular networks, wireless local area networks (WLAN), and wireless personal area networks (WPAN). It is also considered a key technology for 5G and beyond wireless communication systems (Zhao, 2020).

2.3.2 Spatial Multiplexing

Spatial multiplexing is a technique that uses multiple transmission paths to send signals simultaneously, increasing the overall data rate of the system. This technique improves the performance of wireless communication systems by enabling multiple data streams to be transmitted concurrently, supporting multiple users simultaneously. In summary, hybrid beamforming and hybrid precoding focus on optimizing SNR at the receiver, while spatial multiplexing increases data rates by transmitting multiple streams simultaneously. These techniques can be combined to enhance the overall performance of wireless communication systems.

The implementation of UM-MIMO opens new possibilities for various applications. Smart Cities and IoT, supporting a massive number of connected devices, UM-MIMO can enable efficient data collection and communication in smart city environments. Advanced Manufacturing, Low-latency, high-reliability communication is crucial for Industry 4.0 applications, including automated production lines and remote-controlled machinery. Enhanced Mobile Broadband, UM-MIMO will enhance user

experiences in densely populated areas, ensuring consistent high-speed connectivity. In summary, UM-MIMO represents the next frontier in wireless communication technology, building on the foundation of Massive MIMO to meet the demands of future 6G networks. This section explores the advancements, applications, and challenges associated with this transformative technology, emphasizing its potential to revolutionize the way we connect and communicate.

2.4 Channel Model in UM-MIMO for signal detection

In all forms of communication, signals travel through a medium known as a channel. During transmission, these signals can become distorted or subjected to various forms of noise, which are introduced as the signal passes through the channel. Accurately decoding the received signal with minimal errors involves removing the distortions and noise imposed by the channel from the received signal. This process entails identifying the characteristics of the channel through which the signal has passed. The techniques or processes used to identify these channel characteristics are known as signal detection.

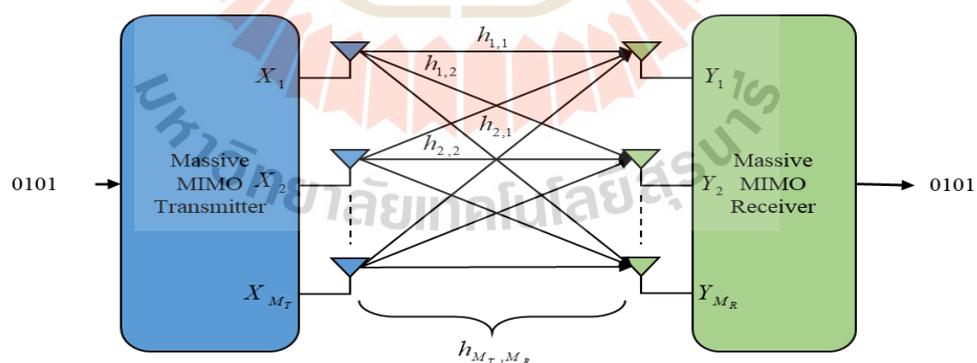


Figure 2.8 The simple of structure communication

In Figure 2.8, which the simple of structure communication, X represents the transmitted signal after modulation, and Y represents the received signal. The channel between the transmitter and receiver is denoted by H , with M_T and M_R

representing the number of transmit and receive antennas, respectively. The relationship between X and Y can be described by the equation.

$$Y = H \times X + n \quad (2.4)$$

Where n_{M_R} is the noise signal at the transmitting antenna. When written in matrix form, it is as follows.

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{M_R} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,M_T} \\ h_{21} & h_{22} & \cdots & h_{2,M_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_R,1} & h_{M_R,2} & \cdots & h_{M_R,M_T} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M_T} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{M_R} \end{bmatrix} \quad (2.5)$$

When considering signal detection, we can transfer the variables to estimate the channel as follows.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M_T} \end{bmatrix} \quad (2.6)$$

The channel matrix estimate X can be computed as

$$X = H^H Y \quad (2.7)$$

where $(\circ)^H$ denotes the conjugate transpose.

There are various current approaches to channel estimation. The scenarios can be described as follows.

2.4.1 Ray/Cluster Channel

The Ray/Cluster Channel model is based on the Saleh-Valenzuela channel model as cited in (Heath, 2016). This model is widely recognized for characterizing mmWave channels, which are marked by significant reflections and limited diffraction. Such channels are particularly relevant in contemporary wireless communication systems, including 5G and beyond, due to their higher frequency operations and the resultant propagation characteristics. Mathematically, the ray/cluster channel matrix can be expressed as.

$$H = \sum_{v=1}^{N_{clust}} \sum_{u=1}^{N_{rays}} \beta_{u,v} a_{rx}(AoA_{u,v}) a_{tx}(AoD_{u,v})^* \quad (2.8)$$

Here, the channel H is composed of N_{clust} clusters, with each cluster containing N_{rays} rays. The u -th ray resides within the v -th cluster. The parameter $\beta_{u,v}$ represents the complex gain associated with each ray. Additionally, $AoD_{u,v}$ refers to the Angle of Departure from the transmitting array, and $AoA_{u,v}$ denotes the Angle of Arrival at the receiving array. The array response vectors for transmission and reception are represented by $a_{tx}(AoD_{u,v})$ and $a_{rx}(AoA_{u,v})$ respectively. Here is the revised section with the new references related to Massive MIMO for millimeter-wave communication. Recent studies have further refined and validated the Ray/Cluster Channel model for mmWave communications. For instance, research conducted by (Li, 2023), emphasizes the importance of accurately modeling the spatial and temporal characteristics of clusters to enhance the performance of mmWave systems. Similarly, (Wang, 2024) explore the impact of different environmental factors on the clustering behavior of rays, offering insights into more precise channel estimation techniques.

These advancements highlight the ongoing evolution of channel modeling techniques, ensuring that the Ray/Cluster Channel model remains robust and applicable to the latest wireless communication technologies.

2.4.2 Rayleigh Fading Channel

The Rayleigh Fading Channel (Bernard, 1997) is a wireless communication channel model that describes signal fading due to the random scattering of signals by objects in the environment. This model is named after Lord Rayleigh, who was the first to describe the phenomenon of fading in the context of wireless communication. A Rayleigh-faded channel is characterized by random variations in the amplitude and phase of the received signal. The amplitude of the received signal follows a Rayleigh distribution, and the phase is uniformly distributed between 0 and 2π . The fading is assumed to be slow, meaning that the channel remains constant over the duration of a symbol. Mathematically As stated in Equation (2.1), H represents the channel response. Received signal Y , where X is the transmitted signal and n is the additive white Gaussian noise (AWGN). In a Rayleigh fading environment, the channel impulse response H can be modeled as a complex Gaussian process with zero mean. This implies that both the real and imaginary parts of H are independent and identically distributed (i.i.d.) Gaussian processes with zero mean and equal variance.

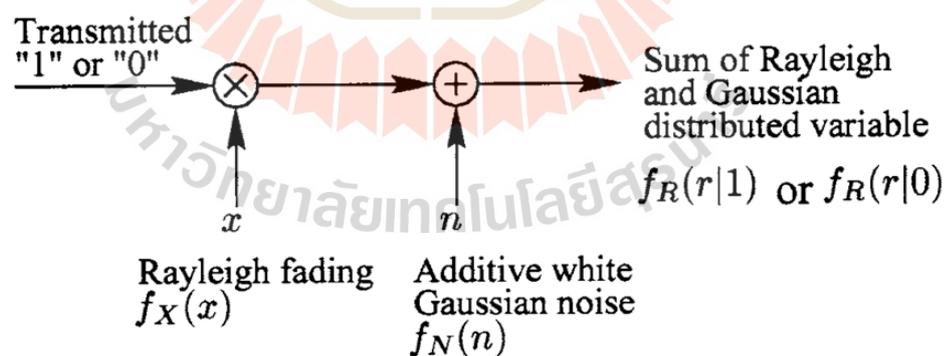


Figure 2.9 Schematic model of a Rayleigh fading channel (Orten, 2002).

In Figure 2.9. Rayleigh fading is commonly observed in wireless communication systems such as mobile communications and wireless local area networks (WLANs), where signals scatter off buildings, trees, and other objects in the environment. The Rayleigh fading channel model is useful for analyzing the performance of wireless

communication systems under fading conditions and for designing equalization and diversity techniques to combat fading, a Rayleigh-faded channel is described by the fact that its entries are drawn from a complex normal distribution with mean 0 and variance 1:

$$[H]_{m,n} \sim N_C(0,1) \quad (2.9)$$

Therefore, from equation 2.7, we can derive an initial signal detection method. However, since this technique is fundamental, it is explained here to provide an understanding of the basic principles of signal detection. The widely used signal detection techniques are ZF and MMSE. Recent advancements in understanding Rayleigh fading have been made by leveraging machine learning techniques to predict and mitigate the impact of fading on signal quality. For instance, (Li, 2023) proposed a deep learning framework for predicting channel state information (CSI) in Rayleigh fading environments, demonstrating significant improvements in communication reliability and efficiency. Additionally, (Marye, 2022) explored adaptive modulation schemes to dynamically adjust transmission parameters based on real-time channel conditions, further enhancing system performance.

2.5 Signal Detection Method

Signal detection is a critical component in wireless communication systems, especially in the presence of interference and noise. Effective signal detection ensures reliable data transmission and enhances overall system performance. Several methods have been developed to address the challenges posed by complex communication environments, each with its strengths and limitations. This section outlines key signal detection techniques employed in modern communication systems, including both traditional linear methods and advanced approaches leveraging machine learning.

2.5.1 Zero Forcing (ZF)

ZF is a linear signal detection technique used to combat interference in multiple-antenna systems. The ZF method works by inverting the channel matrix to nullify the interference from other signals. Although it is effective in reducing inter-symbol interference, it can amplify noise, especially when the channel matrix is nearly singular, leading to suboptimal performance in low signal-to-noise ratio (SNR) scenarios. ZF is an early approach to channel estimation, detecting signals by applying various parameters according to the theory from Equation 3.1. This estimation method is classified as a 1D estimator, meaning channel estimation is performed using test cycles in a single dimension, whether frequency or time. The ZF signal detection method reduces the mean square error between the received signal and the estimated value. It can be described as follows (Trotobas, 2020).

$$\hat{H}_{ZF} = \arg \min_{\hat{H}_{ZF}} \|Y - \hat{H}_{ZF} X\|^2 \quad (2.10)$$

Conversely, if we consider signal detection, it can be expressed as.

$$\tilde{x}(y) = \operatorname{argmin}_x \|y - Hx\|^2 \quad (2.11)$$

Comparing with Equation 2.7, we observe that the constraint of the constellation on x has been removed, significantly reducing complexity. For an invertible square matrix, the solution is given by.

$$\tilde{x}(y) = H^\dagger y \quad (2.12)$$

where H^\dagger is the pseudo-inverse of H , equal to H^{-1} if the matrix is square and invertible. If the matrix is not invertible or not square, we use the pseudo-inverse instead. When $M_T \leq M_R$ and there are at least M_T linearly independent columns in

\mathbf{H} , the pseudo-inverse, sometimes called the Moore-Penrose pseudo-inverse, is defined by.

$$\mathbf{H}^\dagger = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \quad (2.13)$$

The complexity of obtaining \mathbf{H}^\dagger from \mathbf{H} is approximately cubic in M_T for square matrices. However, obtaining $\tilde{\mathbf{x}}(y)$ from y is done in linear time in M_T .

Given that X is the matrix with the order of the transmitting antennas, $X_{M_T} = 1, 2, 3, \dots, M_T$, of the total number of transmitting antennas as follows.

$$X_N(p) = [X_1(p), \dots, X_{M_T}(p)]^T \quad (2.14)$$

where $(\circ)^T$ denotes the conjugate transpose.

2.5.2 Minimum Mean Square Error (MMSE)

The MMSE detection technique aims to minimize the mean square error between the transmitted and received signals. Unlike ZF, MMSE takes into account both the noise and the interference. By finding a balance between noise amplification and interference reduction, MMSE provides a more robust performance across a range of SNR conditions. It does so by combining the received signal with a weighting factor that accounts for the noise variance.

The ZF technique can result in noise amplification if the smallest singular value of \mathbf{H} is too low. This can be quantified using the concept of the condition number of the matrix \mathbf{H} . The condition number of \mathbf{H} is a measure of the relative magnitude of the singular values of \mathbf{H} , defined as the ratio between the largest and smallest singular values of \mathbf{H} . When the condition number is close to unity, the matrix is said to be well-conditioned. However, if the condition number is large, the matrix is ill-conditioned (Trotobas, 2020).

Reducing the sensitivity of linear receivers to the conditioning of the matrix \mathbf{H} , a regularization term can be added to the objective function in Equation 2.8 as follows:

$$\tilde{\mathbf{x}}(\mathbf{y}) = \operatorname{argmin}_x \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda \|\mathbf{x}\|^2 \quad (2.15)$$

For $\lambda > 0$, the solution can be rewritten as.

$$\tilde{\mathbf{x}}(\mathbf{y}) = (\mathbf{H}^H \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^H \mathbf{y} = (\mathbf{H}\mathbf{H}^H + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (2.16)$$

This method, known as the Linear Minimum Mean-Square Error (L-MMSE) technique, minimizes the mean square error in estimating \mathbf{x} among all linear detectors. It solves the following problem.

$$\tilde{\mathbf{x}}(\mathbf{y}) = \operatorname{argmin}_{\mathbf{s} \in \mathcal{E}_{x,n}} \left[\|\mathbf{s} - \mathbf{x}\|^2 \right], \quad (2.17)$$

such that $\mathbf{s} = \mathbf{A}\mathbf{y} + \mathbf{b}$,

For any $M_T \times M_R$ matrix \mathbf{A} and vector $\mathbf{b} \in \mathcal{C}^{M_T}$. Note that the minimization is over all linear functions of \mathbf{y} , where \mathbf{A} and \mathbf{b} are chosen to minimize the expected value over the randomness in \mathbf{x} and \mathbf{n} (assuming the channel matrix \mathbf{H} is known and not random). If \mathbf{x} is Gaussian (instead of from a discrete constellation), this results in an MMSE detector as well. Compared to the Maximum Likelihood (ML) detection technique, both ZF and MMSE linear detectors are easier to implement, but their Bit Error Rate (BER) performance is inferior. Recent research has focused on enhancing MMSE detection techniques, leveraging computational advancements and machine learning to address its limitations. (Minango, 2018) proposed a robust MMSE approach that dynamically adapts to changing channel conditions. Their method utilizes real-time channel state information to adjust the detection parameters, thereby improving the BER

performance without significantly increasing computational complexity. This adaptive approach ensures that the MMSE detector can maintain high performance even in fluctuating environments, making it more reliable for practical applications.

2.5.3 Maximum Likelihood Detection (MLD)

Maximum Likelihood Detection is a nonlinear detection method that searches for the transmitted signal vector that is most likely to have produced the observed received signal vector. MLD offers optimal performance in terms of error rate, as it considers all possible transmitted signal vectors. However, its computational complexity increases exponentially with the number of antennas and modulation order, making it impractical for real-time applications with a large number of antennas (Jeon, 2018).

Mathematically, the MLD problem can be formulated as follows. Given the received signal \mathbf{Y} , the transmitted signal vector \mathbf{X} is estimated by maximizing the likelihood.

$$\hat{\mathbf{X}} = \arg \max_x P(\mathbf{Y} | \mathbf{X}) \quad (2.18)$$

In the case of additive white Gaussian noise (AWGN), the likelihood function $P(\mathbf{Y} | \mathbf{X})$ can be written as.

$$P(y | x) = \frac{1}{(\pi N_0)^{N_r}} \exp\left(-\frac{1}{N_0} \|y - Hx\|^2\right) \quad (2.19)$$

where N_0 is the noise variance, M_R is the number of received signals, and \mathbf{H} is the channel matrix. To simplify the maximization, we can equivalently minimize the negative logarithm of the likelihood function.

$$\hat{x} = \arg \min_x \|y - Hx\|^2 \quad (2.20)$$

This formulation implies that MLD searches for the \mathbf{x} that minimizes the Euclidean distance between the received signal \mathbf{y} and the product $\mathbf{H}\mathbf{x}$.

Given that \mathbf{x} belongs to a finite set of possible transmitted symbols, the minimization can be expressed as.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (2.21)$$

where \mathcal{X} represents the set of all possible transmitted symbol vectors.

Due to the exhaustive search over all possible combinations of \mathbf{x} , the computational complexity of MLD is $\mathcal{O}(|\mathcal{X}|^{M_T})$, where M_T is the number of transmit antennas. This exponential complexity makes MLD impractical for large-scale systems, despite its optimal performance in terms of minimizing the error probability.

2.5.4 Deep Learning-Based Detection

Recent advancements in machine learning have led to the development of deep learning-based signal detection techniques. These methods employ neural networks to learn and adapt to the characteristics of the channel (Wang, 2024). By training on large datasets, deep learning models can capture complex relationships and dependencies that traditional methods might miss. This approach provides high accuracy in varying channel conditions and is particularly useful in scenarios where the channel characteristics change dynamically. Deep learning-based detection leverages various neural network architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), to model the intricate patterns of wireless channels. The primary advantage of these models is their ability to generalize from the training data, enabling robust performance even in challenging and rapidly changing environments. Mathematically, deep learning-based detection can be formulated as follows. Given the received signal \mathbf{y} , a neural network model f_θ parameterized by θ is used to estimate the transmitted signal \mathbf{x} .

$$\hat{x} = f_{\theta}(y) \quad (2.22)$$

The parameters θ are learned from the training data by minimizing a suitable loss function, such as the mean square error (MSE) between the true transmitted signals and the estimated signals.

$$\theta = \arg \min_{\theta} \sum_{i=1}^N \|x_i - f_{\theta}(y_i)\|^2 \quad (2.23)$$

where N is the number of training samples, and x_i and y_i are the transmitted and received signals for the i -th training sample, respectively. A notable advantage of deep learning-based detection is its flexibility to incorporate various types of channel state information (CSI) and to adapt to different modulation schemes. This adaptability makes it a powerful tool for modern wireless communication systems, including 5G and beyond.

2.6 Machine Learning (ML)

The field of machine learning has evolved in tandem with artificial intelligence. In fact, machine learning has been present since the early days of artificial intelligence. Many scientists were interested in creating machines that could learn from data, leading to various experimental methods. The most notable among these methods is the artificial neural network. Over time, general linear models based on statistical principles were developed, and methods of probabilistic reasoning were advanced, especially in applications such as automatic disease diagnosis. Broadly speaking, machine learning can be categorized in several ways based on the type of input data or training datasets and the types of machine learning approaches, as outlined below.

2.6.1 Supervised Learning

Supervised learning, or learning with a supervisor, allows a computer to solve problems independently after learning from a set of example data over a period of time. The principles of supervised learning can be applied in two main forms classification and regression (Draper, 1998). Supervised learning involves training a model on a labeled dataset, meaning that each training example is paired with an output label. The goal is for the model to learn a mapping from inputs to outputs that can be applied to new, unseen data. Supervised learning is widely used in various applications, such as image and speech recognition, medical diagnosis, and spam detection (LeCun, 2015; Goodfellow, 2016), which is shown in Figure 2.10.

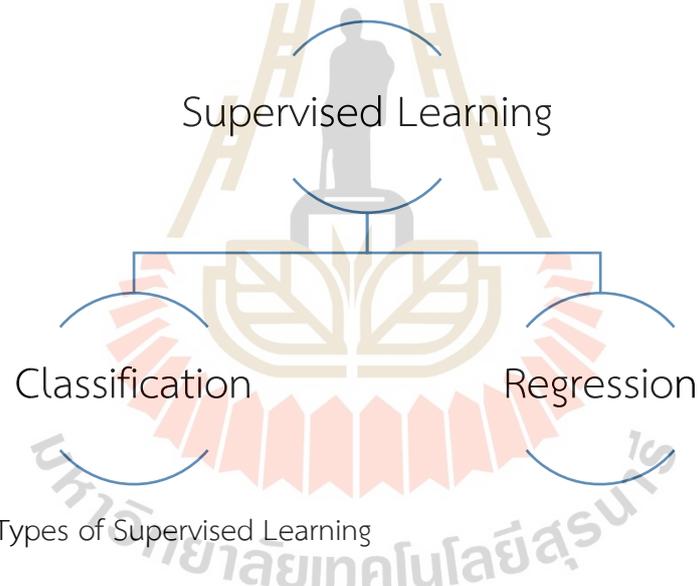


Figure 2.10 Types of Supervised Learning

2.6.1.1 Regression Analysis

Regression analysis, specifically simple regression analysis, studies the relationship between two variables, denoted as X and Y , that have a linear relationship as follows.

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (2.24)$$

where β_0 is the intercept, β_1 is the slope, and ϵ is the error term. Simple regression analysis assumes that the variable X is predetermined and the value of Y changes according to X . Each value of X pairs with a corresponding value of Y , and when these pairs are plotted, they form a regression line that represents the average relationship between the variables.

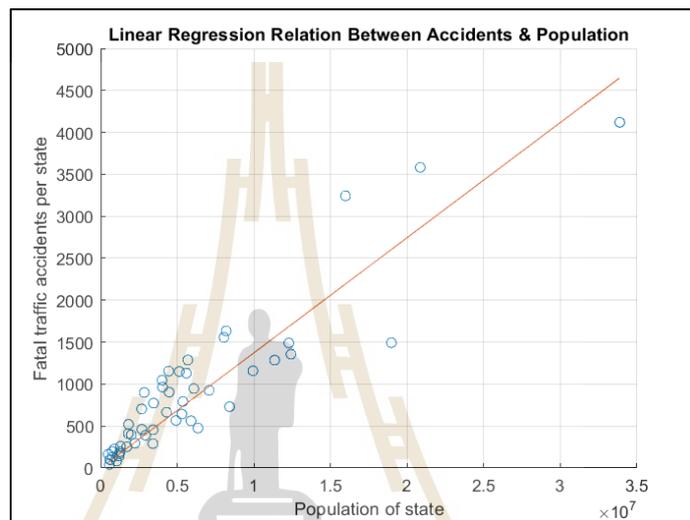


Figure 2.11 Example Graph of Accidents and Population

In Figure 2.11, this example, X represents the number of accidents and Y represents the population. The data is hypothetical and demonstrates a regression line, which shows the linear relationship between the two variables. The least squares method (LS) is commonly used to estimate the parameters β_0 and β_1 , where β_0 is a constant and β_1 is the slope of the line. This estimation directly affects the regression line's change between X and Y , which researchers refer to as the regression coefficient or prediction coefficient (Goodfellow, 2016; Montgomery, 2021).

2.6.1.2 Classification Analysis

Classification involves categorizing data into distinct classes. The general process of classification entails an algorithm attempting to approximate a function f from the training set and producing $f(x)$ in the form of distinct

categories. A crucial aspect of classification is defining the target data clearly. For example, determining whether students in Class A pass or fail an exam. The analysis of Class A students' data using classification will yield two outcomes: pass or fail. Similarly, in credit approval, the goal might be to determine whether a loan application is approved or denied. However, targets can be more than two categories, such as classifying flowers into four groups based on features like color, petal size, and origin.

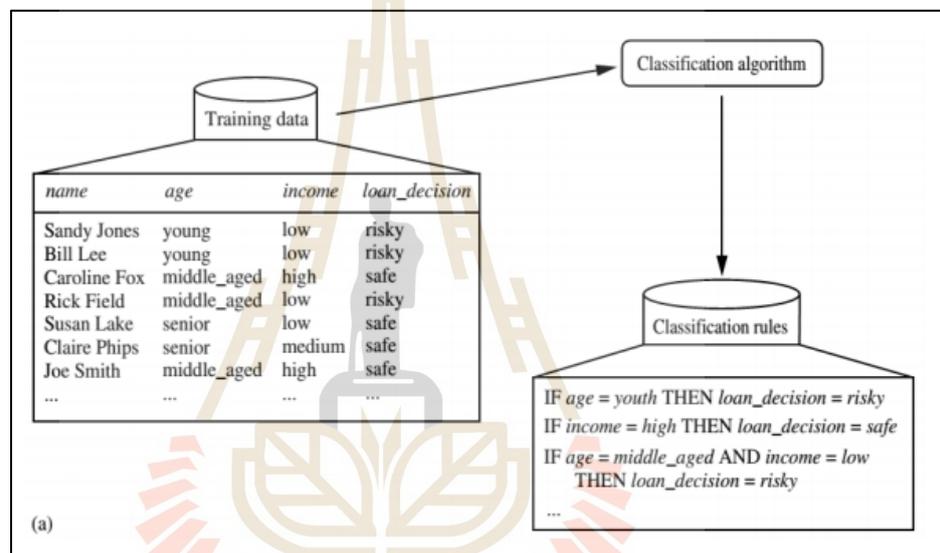


Figure 2.12 Data Classification (a)

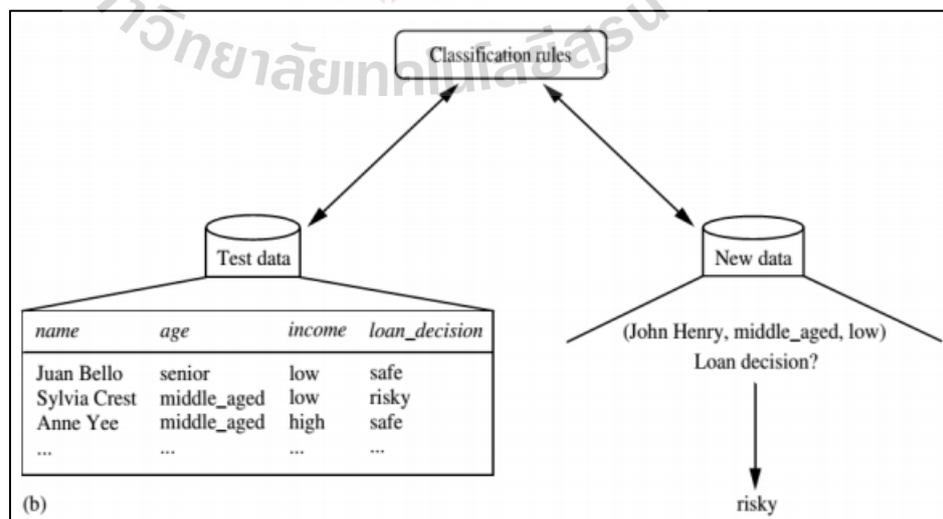


Figure 2.13 Learning from Data to Create a Classifier (b)

In Figures 2.12 and 2.13, the process of borrowing and lending. Figure 2.12 shows the creation of a classifier from input data, where each recorded data point includes a set of attributes that describe the characteristics of individuals borrowing money and categorizes them as either low-risk or high-risk borrowers. The process of creating a classifier is often called "learning" or "training," where the classification algorithm is applied to the data. Each recorded data point includes a set of attributes describing its characteristics, along with a class label attribute indicating its category. The input dataset used for creating the classifier is called the "training data."

In the second step of classification, shown in Figure 2.13, the created classifier is used to predict new data's category. Initially, the classifier is tested and evaluated for accuracy. If we use the training data for testing, the classifier's accuracy will be high due to overfitting. However, testing with a different dataset (test set) containing the class label attribute allows for an accurate evaluation of the classifier's performance. The classifier's accuracy is the percentage of correctly classified data points, matching the class label of each recorded data point. Once the classifier's accuracy is satisfactory, it can be used to classify new, unseen data ("unknown" or "previously unseen" data). For instance, the classifier created in Figure 2.12 can be used to decide on loan applications. Figures 2.12 and 2.13 depict testing the classifier for accuracy. Common criteria for evaluating classification methods include accuracy, precision, recall, and F1 score (Bishop, 2006; Tan, 2013).

Table 2.1 Comparison of the Performance of Classification and Data Processing.

Accuracy	This concerns the ability of the constructed classifier to correctly classify previously unseen data. The accuracy can be evaluated using one or more datasets that are separate from the training dataset.
----------	---

Table 2.1 Comparison of the Performance of Classification and Data Processing.
(Continued)

Speed	This pertains to the computation time required for both the creation of the classifier and the classification or prediction of data.
Robustness	This involves the classifier's or predictor's ability to make accurate predictions from noisy or incomplete initial data.
Scalability	This relates to the efficiency of constructing classifiers or predictors when dealing with large volumes of data.
Interpretability	This concerns the extent to which the classifier or predictor can be understood by users.

2.6.1 Unsupervised Learning

In supervised learning, the correct answer is known and provided in the form of labels. However, in unsupervised learning, the exact answer is unknown, and the goal is for the machine to discover the underlying structure in the data. For instance, in the study of DNA structure, we might aim to identify the genes that influence eye color, such as blue or black. While we do not know which specific part of the DNA affects eye color, we have DNA data from individuals that can be categorized into two groups: one with blue eyes and the other with black eyes. The objective is for the machine to identify which parts of the DNA distinctly differentiate between the two groups and are consistent within each group. Unsupervised learning can be divided into two main types, clustering and dimensionality reduction. These can be explained as follows.

2.6.2.1 Clustering

Clustering algorithms examine only the input data without providing an outcome. For instance, in a population survey to identify patterns in the data, we need to know how many classes the data comprises. We must specify what each group discovered by the tool represents. This allows us to reduce the number

of labels to be as many as the classes. In Figure 2.14, for example, clustering data into two dimensions (data input consisting of two values, x and y) without knowing the groups beforehand. We instruct the tool to find a way to separate the data into four groups, and the result is the boundaries of each group, as illustrated in the following figure. This approach is comprehensively discussed in (Hastie, 2009).

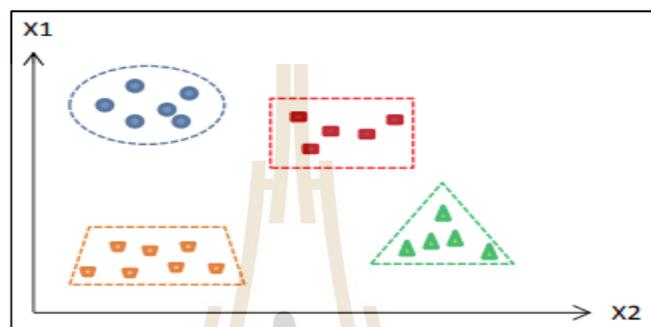


Figure 2.14 Clustering Data

2.6.2.2 Dimensionality Reduction

In Figure 2.15, dimensionality reduction compresses the data by reducing the number of dimensions, making it unnecessary to store incomplete data while still being able to classify it. The goal is to reduce the data to two dimensions and still achieve good class separation. This technique is well-articulated in (Bishop, 2006) and (Raschka, 2015).

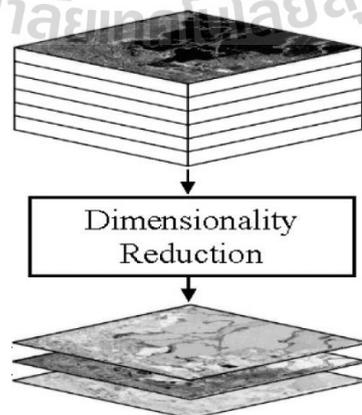


Figure 2.15 Visualization of Dimensionality Reduction for Class Separation.

In unsupervised learning, the goal is to find hidden patterns or intrinsic structures in the input data. Clustering is one such technique, which involves grouping data points with similar characteristics together. Another technique is dimensionality reduction, which simplifies the data while retaining its essential features, enabling efficient data analysis and visualization. These methodologies are foundational in the field of machine learning, as detailed in (Raschka, 2015; Murphy, 2012).

2.7 Deep Learning (DL)

Deep learning is a subset of machine learning that utilizes artificial neural networks (ANNs) with multiple layers to learn from data. These layers are called hidden layers because the inputs and outputs are not directly observable. The architecture of these neural networks is deep, meaning they have multiple layers, typically more than two or three, as shown in the following Figure 2.16.

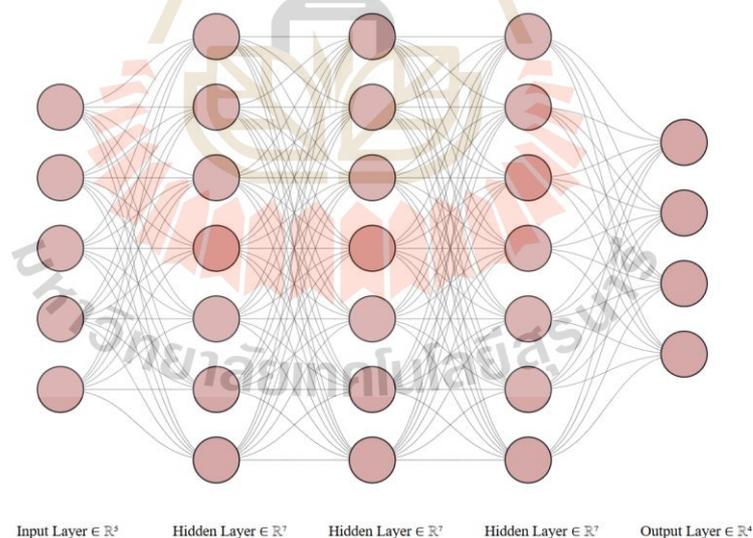


Figure 2.16 Basic Structure of Deep Learning

Deep learning models require training and testing on large amounts of data. They learn to recognize patterns and make predictions by adjusting the weights of the connections between neurons in each layer. These weights are adjusted during the training process using an algorithm called backpropagation, which uses the error

between the predicted and actual outcomes to modify the weights. Deep learning models can be applied to a variety of tasks, such as image recognition, speech recognition, natural language processing, and even playing games. They achieve state-of-the-art performance in many of these tasks and are used in applications like autonomous driving, medical imaging, and virtual personal assistants. For an overview of the field, see (Schmidhuber, 2015). Deep learning models can be classified into different types, such as feedforward neural networks (FFNN), recurrent neural networks (RNN), convolutional neural networks (CNN), and autoencoders. Each type has its own unique characteristics that are tailored to specific problems. Can explain as follows. Feedforward Neural Networks (FFNN) these are the simplest type of artificial neural network architecture where information moves in one direction forward from the input nodes, through the hidden nodes (if any), to the output nodes. There are no cycles or loops in the network. Recurrent Neural Networks (RNN) unlike feedforward neural networks, RNNs have connections that form directed cycles. This means that the output from some neurons can be fed back into the network, allowing them to maintain a form of memory. They are particularly useful for tasks that involve sequential data, such as time series analysis or natural language processing. For more details, refer to (Hinton, 2012). Convolutional Neural Networks (CNN) these are particularly effective for spatial data, such as images. They use convolutional layers that apply a convolution operation to the input, passing the result to the next layer. This helps in capturing spatial hierarchies in images, making CNNs very powerful for image recognition tasks. The effectiveness of CNNs in image classification is demonstrated by (Krizhevsky, 2012). Autoencoders these are a type of neural network used to learn efficient codings of input data. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction. They are also used for unsupervised learning of features. In summary, machine learning is a broad term that encompasses any method of teaching computers to learn from data, whereas deep learning is a subset of machine learning that uses deep neural networks to learn from data. Deep

learning models are particularly powerful because they can learn hierarchical representations of the data, which allows them to excel at tasks like image and speech recognition.

2.8 Aspects of Comparative Analysis

In measuring the performance of data from the application of machine learning and deep learning, there are several methods, such as evaluating the communication system and assessing the stability of the data processed through machine learning. These methods are as follows.

2.8.1 Data Loss Function

In measuring the performance of data from the application of machine learning, there are various methods, as shown in Table.

Table 2.2 Performance Metrics: Classification and Regression Loss.

Classification	Log Loss
	Focal Loss
	KL Divergence/Relative Entropy
	Exponential Loss
	Hinge Loss
Regression	Mean Square Error / Quadratic Loss
	Mean Absolute Error
	Huber Loss / Smooth Mean Absolute Error
	Log cosh Loss
	Quantile Loss

The types of regression loss measurements are as follows.

2.8.1.1 Mean Square Error, Quadratic Loss (L2 Loss)

Mean Square Error (MSE) is the most used regression loss function. It measures the average of the squares of the errors that is the average squared difference between the estimated values (predictions) and the actual value.

This metric is particularly useful because it punishes larger errors more than smaller ones, making it very effective for capturing the quality of model predictions (LeCun, 2016; Montgomery 2021; Tan, 2013). Mathematically, MSE is defined as.

$$MSE = \frac{\sum_{i=1}^n (y_i - y_i^p)^2}{n} \quad (2.25)$$

Where y_i represents the actual target value. y_i^p represents the predicted value.

n is the number of observations.

The derivation of this equation starts from the basic principle of least squares, which seeks to minimize the sum of the squared differences between the observed and predicted values. This minimization process leads to the MSE formula, providing a clear and effective way to measure prediction accuracy. Below is a plot of the MSE function where the true target value is 100, and the predicted values range from -10,000 to 10,000. The MSE loss (Y-axis) reaches its minimum when the prediction (X-axis) is equal to 100. The range of MSE is from 0 to ∞ , as it only takes non-negative values due to the squaring operation in the formula. This visualization helps to understand how deviations from the true value affect the MSE. As predictions move away from the actual target value, the MSE increases quadratically, emphasizing the penalty for larger errors.

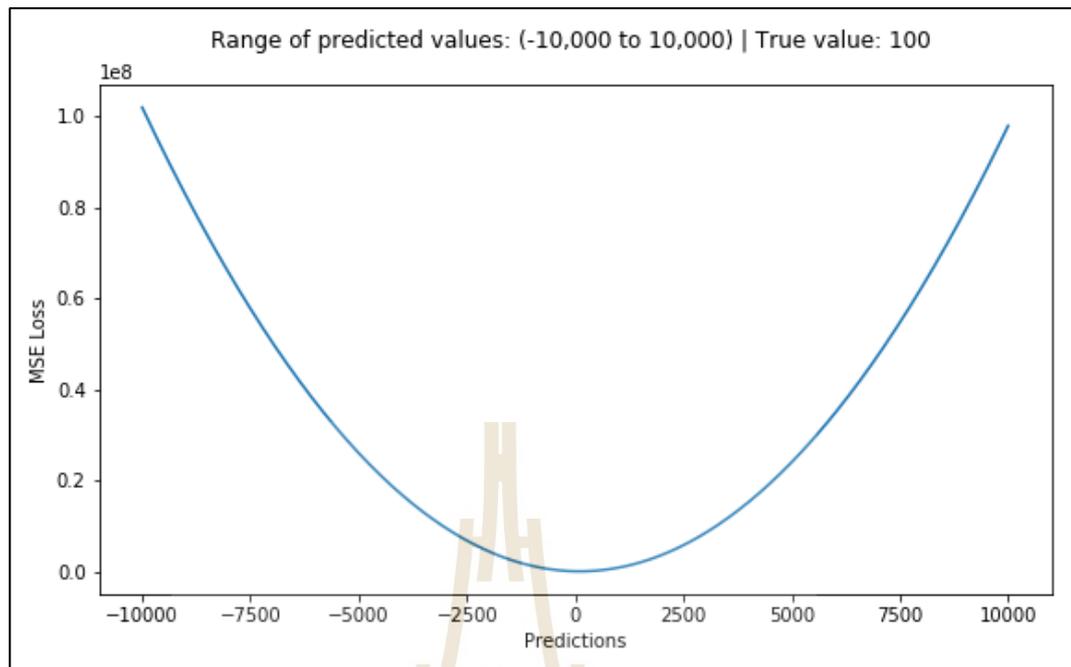


Figure 2.17 Plot of MSE Loss (Y-axis) vs. Predictions (X-axis)

In Figure 2.17. MSE is widely used in regression analysis and is fundamental in training algorithms for machine learning models. It provides a straightforward approach to evaluate the accuracy of model predictions and is an essential tool in various applications, such as finance, health care, and engineering, where precise prediction is crucial.

2.8.1.2 Mean Absolute Error (L1 Loss)

Mean Absolute Error (MAE) is another loss function used for regression models (Hunter, 2007; Pedregosa, 2011). MAE is the sum of the absolute differences between the target values and the predicted values. It measures the average magnitude of the errors in a set of predictions, without considering their direction. If the direction is considered, it is called the Mean Bias Error (MBE), which is the sum of the signed errors. The range of MAE is from 0 to ∞ . The MAE is defined mathematically as.

$$MAE = \frac{\sum_{i=1}^n |y_i - y_i^p|}{n} \quad (2.26)$$

Where y_i represents the actual target value. y_i^p represents the predicted value.

n is the number of observations. MAE is particularly useful because it is less sensitive to outliers compared to the Mean Square Error (MSE), as it does not square the error term. This makes MAE a more robust metric in certain applications. Below is a plot of the MAE function where the true target value is 100, and the predicted values range from -10,000 to 10,000. The MAE loss (Y-axis) is minimized when the prediction (X-axis) equals the true value.

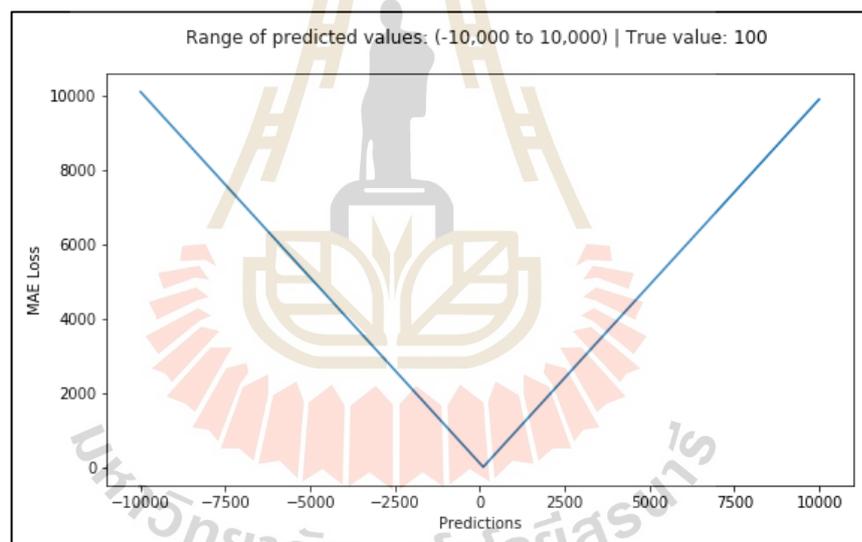


Figure 2.18 Plot of MAE Loss (Y-axis) vs. Predictions (X-axis)

In Figure 2.18. The MAE graph demonstrates how the average absolute error varies with predictions. The MAE provides a clear indication of the average prediction error magnitude, making it a straightforward and interpretable loss function for regression tasks.

2.8.1.3 Huber Loss (Smooth Mean Absolute Error)

Huber Loss is less sensitive to outliers in data compared to the Mean Squared Error (MSE). Additionally, it is differentiable at 0. Essentially, it

combines the best properties of both MSE and Mean Absolute Error (MAE). Huber Loss behaves like the absolute error when the error is large and like the squared error when the error is small. The threshold at which it transitions from one to the other is determined by the hyperparameter δ (delta), which can be tuned. Huber Loss approaches MSE when δ is close to 0 and approaches MAE when δ is very large. The Huber Loss is defined as:

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \quad (2.27)$$

Where y is the actual target value. $f(x)$ is the predicted value. δ is the threshold parameter that determines the point where the loss function changes from quadratic to linear. The following plot illustrates the behavior of Huber Loss. The Y-axis represents the Huber Loss, and the X-axis represents the predicted values.

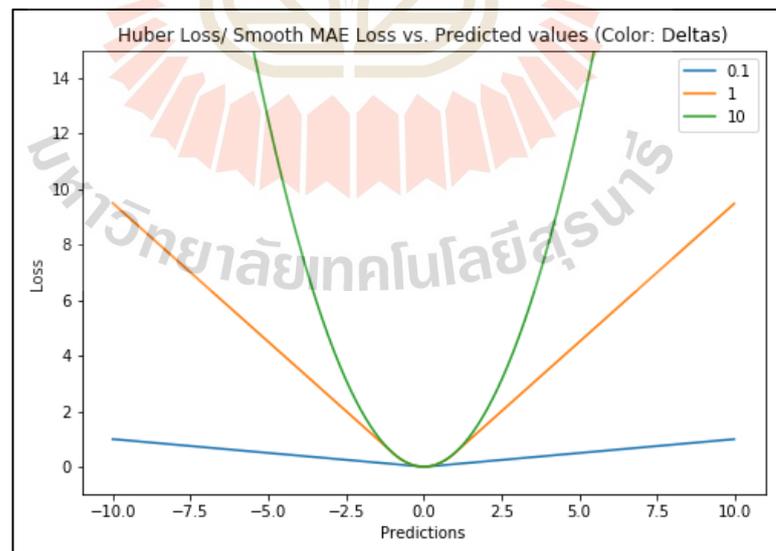


Figure 2.19 Plot of Huber Loss (Y-axis) vs. Predictions (X-axis)

In Figure 2.19. Huber Loss combines the advantages of both MSE and MAE by providing a quadratic loss for small errors and a linear loss for large errors. This

characteristic makes it robust to outliers while maintaining sensitivity for smaller errors, making it a versatile loss function in regression tasks (Huber, 1964).

2.8.1.4 Log-Cosh Loss

Log-Cosh is another loss function used in regression tasks that is smoother than the L2 loss. Log-Cosh is the logarithm of the hyperbolic cosine of the prediction error. It combines the benefits of Mean Squared Error (MSE) and Mean Absolute Error (MAE) while being less sensitive to outliers. The Log-Cosh Loss is defined as.

$$L_{\gamma}(y, y^p) = \sum_{i=1}^n \log(\cosh(y_i^p - y_i)) \quad (2.28)$$

Where y_i represents the actual target value. y_i^p represents the predicted value.

n is the number of observations. One of the main advantages of Log-Cosh Loss is that $\log(\cosh(x))$ is approximately equal to $\frac{x^2}{2}$ for small x and $|x| - \log(2)$ for large x . This means that Log-Cosh Loss behaves like MSE for small errors and like MAE for large errors, making it robust to occasional incorrect predictions. It inherits all the benefits of Huber Loss and is differentiable twice everywhere, unlike Huber Loss.

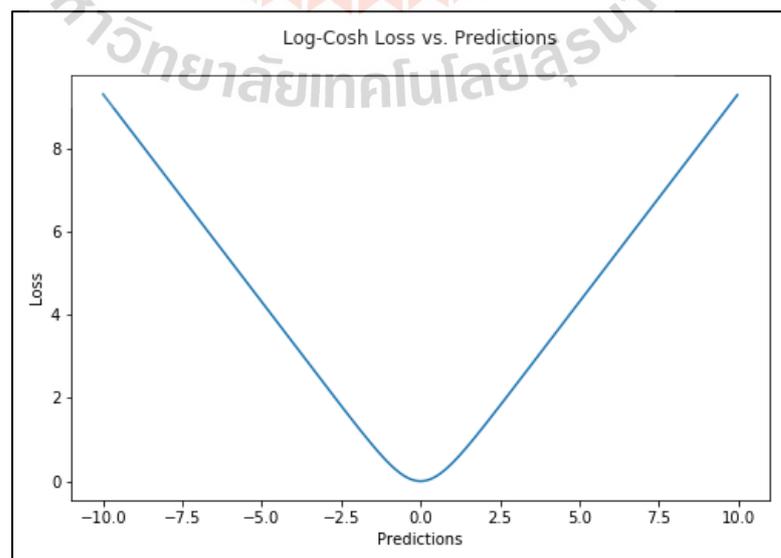


Figure 2.20 Plot of Log-Cosh Loss (Y-axis) vs. Predictions (X-axis)

In Figure 2.20. Log-Cosh Loss provides a balance between the sensitivity of MSE and the robustness of MAE, making it a versatile and efficient loss function for regression tasks. It smooths the transition between different error scales and provides stability in the presence of outliers, making it a reliable choice for various machine learning applications (Charbonnier, 1994).

2.8.1.5 Quantile Loss

In real-world prediction problems, uncertainty in our predictions is often a crucial factor to consider. Knowing the range of predictions, rather than just point estimates, can significantly improve decision-making processes for many business problems. Quantile Loss becomes beneficial when we are interested in predicting intervals instead of just point estimates. Prediction intervals from least squares regression rely on the assumption that the residuals (actual values minus predicted values) have constant variance across the values of the independent variables. Linear regression models that violate this assumption cannot be trusted. We cannot abandon the concept of linear regression modeling altogether by asserting that non-linear functions or tree-based models will always be better in such scenarios. This is where quantile loss and quantile regression come into play, as quantile regression provides reasonable prediction intervals even for residuals with non-constant variance or non-normal distributions, as discussed by (Koenker, 1978). Quantile regression aims to estimate the conditional quantiles of the response variable given certain values of the predictor variables. Quantile Loss is an extension of Mean Absolute Error (MAE); when the quantile is the 50th percentile, it becomes MAE. The idea is to choose the quantile value based on whether we want to give more weight to positive errors or negative errors. The loss function attempts to penalize overestimation and underestimation differently based on the chosen quantile value (γ). For instance, a quantile loss function with $\gamma = 0.25$ penalizes

overestimation more and tries to keep predictions slightly below the median. The Quantile Loss function is defined as:

$$L_{\gamma}(y, y^p) = \sum_{i: y_i < y_i^p} (\gamma - 1) |y_i - y_i^p| + \sum_{i: y_i > y_i^p} \gamma |y_i - y_i^p| \quad (2.29)$$

Where y_i represents the actual target value. y_i^p represents the predicted value. γ is the quantile value.

This loss function provides different penalties for overestimation and underestimation based on the quantile value, making it suitable for constructing prediction intervals that accommodate varying levels of uncertainty and non-constant variance in the data. Quantile regression and quantile loss are valuable tools for estimating conditional quantiles, offering robust prediction intervals even in the presence of heteroscedasticity or non-normal residual distributions. By selecting appropriate quantiles, we can tailor our prediction intervals to better reflect the underlying uncertainty in the data. For an in-depth exploration of these methods, refer to Koenker's comprehensive work on quantile regression (Koenker, 2005).

2.8.2 Bit Error Rate (BER) Analysis for Different Modulation Schemes

In digital communication systems, the Bit Error Rate (BER) is a key parameter used to measure the performance of a communication channel. BER is defined as the ratio of the number of bits received incorrectly to the total number of bits transmitted. It provides insight into the reliability of the communication system under various conditions of noise, interference, and distortion.

2.8.2.1 BER for BPSK (Binary Phase Shift Keying)

BPSK is one of the simplest and most robust modulation schemes. The BER for BPSK is given by.

$$BER = Q \left(\sqrt{\frac{2E_b}{N_0}} \right) \quad (2.30)$$

where:

- BER is the bit error rate,
- $Q(\cdot)$ is the Q-function,
- E_b is the energy per bit,
- N_0 is the noise power spectral density.

For high Signal-to-Noise Ratio (SNR), the BER can be approximated by:

$$BER \approx \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (2.31)$$

For how to find BER calculations for various modulation schemes. The researcher has compiled the methods and equations in the table as follows.

Table 2.3 Calculation of BER for the different modulation schemes

Modulation Schemes	Calculation BER	Remarks
BPSK	$BER \approx \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$	Equation (2.32)
QPSK	$BER \approx \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{2N_0}} \right)$	Equation (2.33)
16-QAM	$BER \approx \frac{3}{2} \operatorname{erfc} \left(\sqrt{\frac{4E_b}{10N_0}} \right)$	Equation (2.34)
64-QAM	$BER \approx \frac{7}{6} \operatorname{erfc} \left(\sqrt{\frac{3E_b}{7N_0}} \right)$	Equation (2.35)
256-QAM	$BER \approx \frac{15}{8} \operatorname{erfc} \left(\sqrt{\frac{10E_b}{17N_0}} \right)$	Equation (2.36)

In summary, the Bit Error Rate (BER) for different modulation schemes varies depending on the modulation order and the Signal-to-Noise Ratio (SNR). Higher-order modulation schemes, such as 64-QAM and 256-QAM, offer higher data rates but require higher SNR to achieve the same BER as lower-order schemes like BPSK and QPSK. Understanding the BER performance of these modulation schemes is crucial

for designing reliable and efficient communication systems, as discussed in (Proakis, 2008).

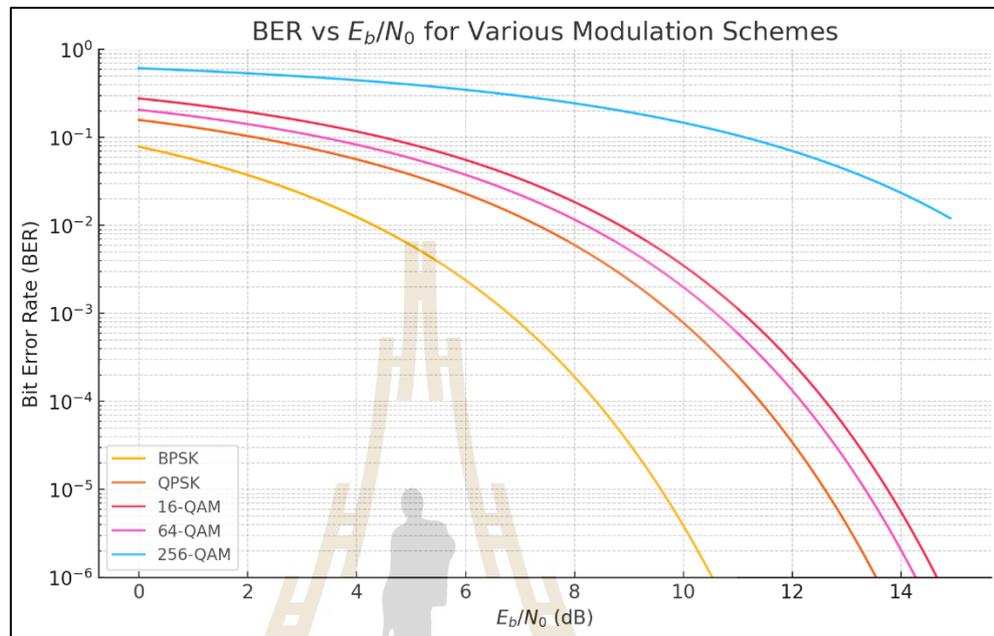


Figure 2.21 the Bit Error Rate (BER) versus E_b/N_0 for various modulation schemes

In Figure 2.21, showing the Bit Error Rate (BER) versus E_b/N_0 for various modulation schemes including BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM. The graph demonstrates how the BER decreases with increasing E_b/N_0 for each modulation scheme, highlighting the trade-off between higher data rates and the required signal-to-noise ratio (SNR). This trade-off is critical in communication system design, as detailed by (Sklar, 2001; Haykin, 2008).

2.8.3 Big O notation for Computational Complexity

Big O notation is a mathematical representation used to describe the asymptotic behavior of functions. It provides a high-level understanding of the time or space complexity of an algorithm, especially in terms of its performance relative to the size of its input.

Given functions $f(n)$ and $g(n)$, we say that $f(n) = O(g(n))$ if there exist positive constants c and n_0 such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$. This notation

captures the upper bound of an algorithm's growth rate, ignoring constant factors and lower-order terms to focus on the most significant component as the input size n grows. In essence, Big O notation provides an abstract measurement of an algorithm's efficiency, enabling comparison of different algorithms regardless of machine-specific constants and implementation details.

2.8.3.1 Application in Computational Complexity

About of algorithm analysis, Big O notation helps compare the efficiency of different algorithms by providing a common framework to describe their time and space requirements. Here are key properties and applications relevant to this discussion. In the field of computational complexity, understanding the efficiency of various algorithms is crucial. The Big O notation is a mathematical notation used to describe the upper bound of an algorithm's time or space complexity, providing a common framework for comparing different algorithms and many studies have mentioned the process of analytical thinking (Cormen, 2009; Golub, 2013; Strang, 2006; Nocedal, 2006; Trefethen, 1997; Higham, 2002). This notation helps in analyzing how the runtime or space requirements of an algorithm grow as the input size increases. The following table summarizes the time complexities of several common matrix operations and iterative methods used in computational complexity. Each entry includes a brief description of the operation and its associated Big O notation. This information is essential for selecting the most appropriate algorithm based on the specific requirements of a given problem, ensuring optimal performance and resource utilization.

Table 2.4 Computational Complexity Applications

Operation	Vector Complexity	Matrix Complexity	Description
Addition Subtraction	$O(N)$	$O(MN)$	Element-wise addition/subtraction of two vectors a and b of size N or two matrices A and B of size $M \times N$.

Table 2.4 Computational Complexity Applications (Continued)

Scalar Multiplication	$O(N)$	$O(MN)$	Multiplication of vector \mathbf{a} of size N or matrix \mathbf{A} of size $M \times N$ by a scalar c .
Dot Product	$O(N)$	N/A	Inner product of two vectors \mathbf{a} and \mathbf{b} of size N .
Matrix-Vector Multiplication	N/A	$O(MN)$	Multiplication of matrix \mathbf{A} of size $M \times N$ by vector \mathbf{b} of size N .
Matrix-Matrix Multiplication	N/A	$O(MKN)$	Multiplication of matrix \mathbf{A} of size $M \times K$ by matrix \mathbf{B} of size $K \times N$.
Element-wise Division	$O(N)$	$O(MN)$	Element-wise division of two vectors \mathbf{a} and \mathbf{b} of size N or two matrices \mathbf{A} and \mathbf{B} of size $M \times N$.
Matrix Inversion	N/A	$O(N^3)$	Inverting a square matrix \mathbf{A} of size $N \times N$.
Element-wise Operations	$O(N)$	$O(MN)$	Element-wise operations (e.g., ReLU, sigmoid) on vector \mathbf{a} of size N or matrix \mathbf{A} of size $M \times N$.
Activation Function	$O(N)$	$O(MN)$	Applying activation function on vector \mathbf{a} of size N or matrix \mathbf{A} of size $M \times N$.

2.9 Literature Review

This section describes research studies that have been reviewed and validated by researchers to provide a consistent basis and to improve the results of this research. By collecting analytical work related to UM-MIMO in the past 4 years in the IEEE database that includes academic journals and academic conferences, a total of 23 works. Updated on 24 May 2022.

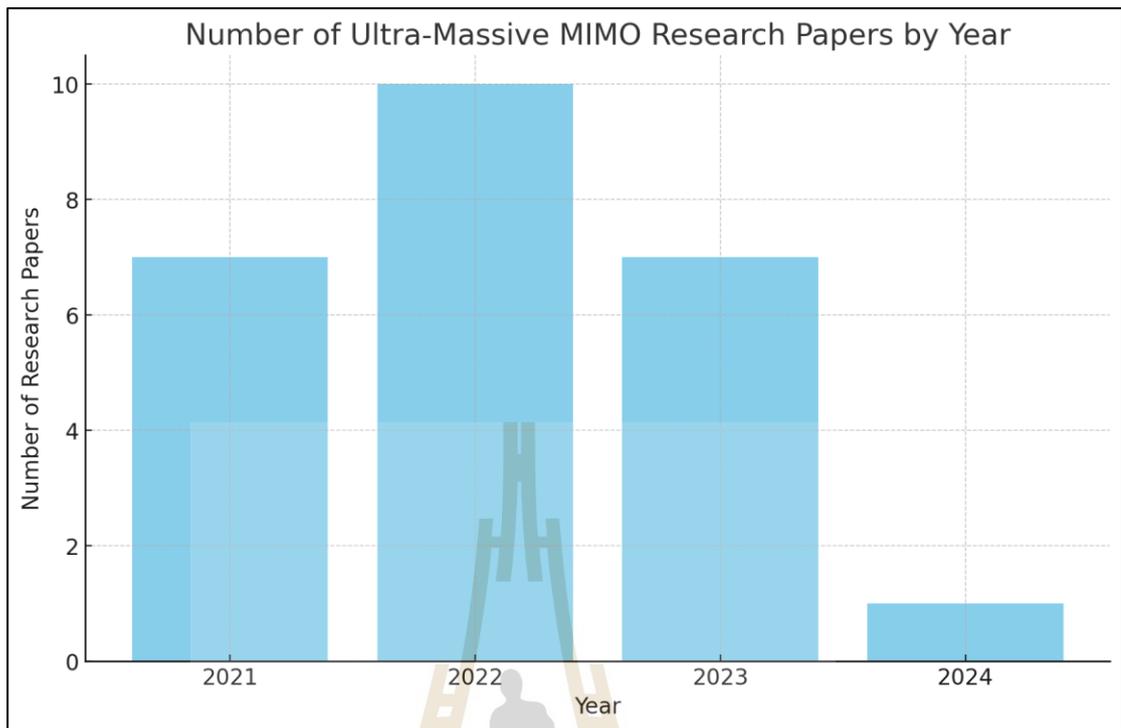


Figure 2.22 The amount of research on UM-MIMO by publication year. Updated on 24 May 2022.

In Figure 2.22, which only a few studies have been done on channel estimation and signal detection, with 1 research related to signal detection and 4 research related to Channel estimation. The researcher will first discuss the origins of this work, which the details are as follows:

(Sarieeddeen, 2019), research focuses on the use of Spatial Modulation (SM) techniques for UM-MIMO systems operating in the terahertz frequency range. (THz), which is one of the first works to address Ultra-Massive MIMO. The focus of the study is to address the long-range transmission problem at THz frequencies, which are known to have propagation losses and inherent limitations. very high energy This research presents the use of graphene sheets to create high-density nano-antennas. This can increase beamforming efficiency. However, this design still has limitations in the number of spatial degrees of freedom available for spatial adjustment.

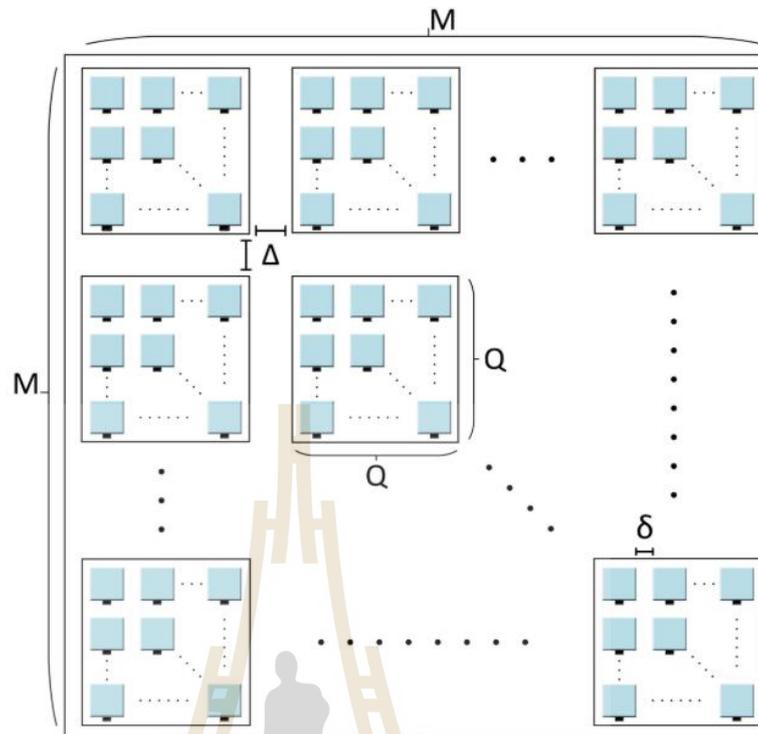


Figure 2.23 Array-of-subarrays antenna structure.

In Figure 2.23, the proposed antenna arrangement structure in this research is crucial for enhancing performance and communication range in the Terahertz frequency band using UM-MIMO and Spatial Modulation techniques. This figure illustrates an Array-of-Subarrays (AoSA) configuration, consisting of multiple subarrays. Each subarray contains a significant number of nano-antennas. This subarray arrangement, comprising numerous nano-antennas, facilitates signal tuning and control, allowing adjustments based on frequency and communication distance.

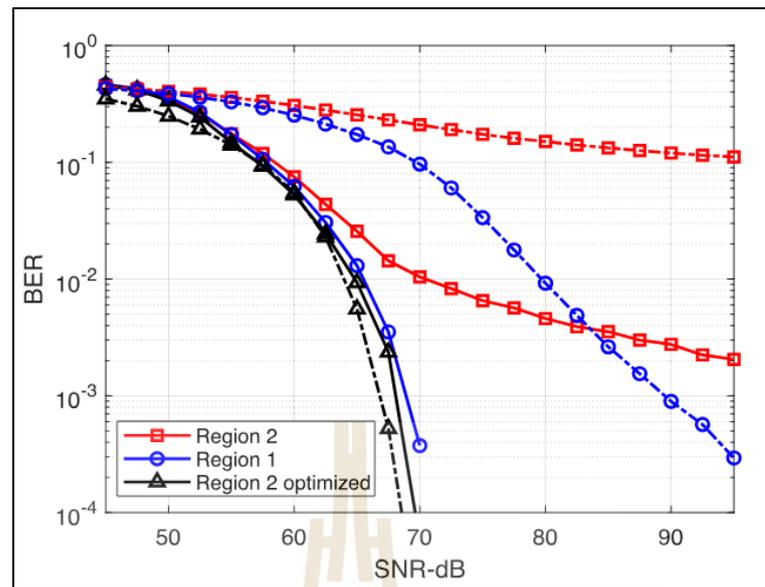


Figure 2.24 BER performance of SM (solid lines) and SMX (dotted lines): $f = 1$ THz, $D = 1$ m, $M = 8$, $Q = 32$ and $|X| = 64$.

In Figure 2.24, the simulation compares the performance of Bit Error Rate (BER) between two communication methods, Spatial Modulation (SM) and Spatial Multiplexing (SMX) at a frequency of 1 THz and a distance of 1 meter in a 64×64 MIMO system ($M=8$). The system uses nano-antennas with 1024-element SAs ($Q=32$) and 64-QAM for the modulation scheme. The results demonstrate that the use of optimized antenna tuning in a 64×64 MIMO system at 1 THz significantly reduces the BER for both SM and SMX compared to the unoptimized case. Particularly for SMX, there is a notable reduction in BER with antenna optimization. However, SM exhibits greater stability in scenarios with poor channel quality, underscoring the importance of antenna optimization for enhancing communication performance in UM-MIMO systems at Terahertz frequencies. On the other hand, the results presented in this paper have limitations due to the excessively high SNR range of 45-95 dB, which may not be cost-effective for some organizations or users with budget constraints. Despite the high SNR, there could still be interference from other stronger signals in certain conditions, necessitating further management.

Based on the review of previous articles, the researchers have explored the application of Deep Learning and Machine Learning in UM-MIMO systems. In the work by (Nie, 2019), a method for channel estimation in UM-MIMO communication systems operating in the frequency range of 0.06-10 THz is presented. This method utilizes Deep Kernel Learning (DKL) in conjunction with Gaussian Process Regression (GPR) to enhance the accuracy and efficiency of channel estimation in systems with antenna arrays comprising more than 1,000 elements. DKL reduces the computational complexity compared to traditional linear estimation methods such as Least Squares (LS) and MMSE.

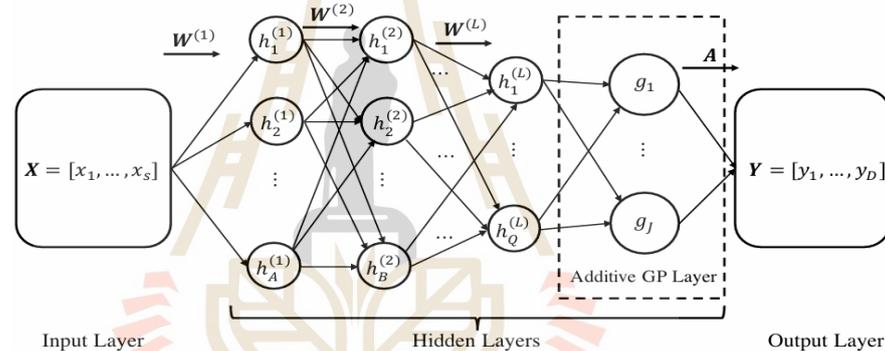


Figure 2.25 An illustration of the deep kernel learning architecture in the Gaussian process regression (Nie, 2019).

In Figure 2.25, The GPR as presented in this work, is a statistical estimation method that employs Gaussian Processes (GP) to estimate the relationship between input and output variables without requiring a predefined functional form for the relationship.

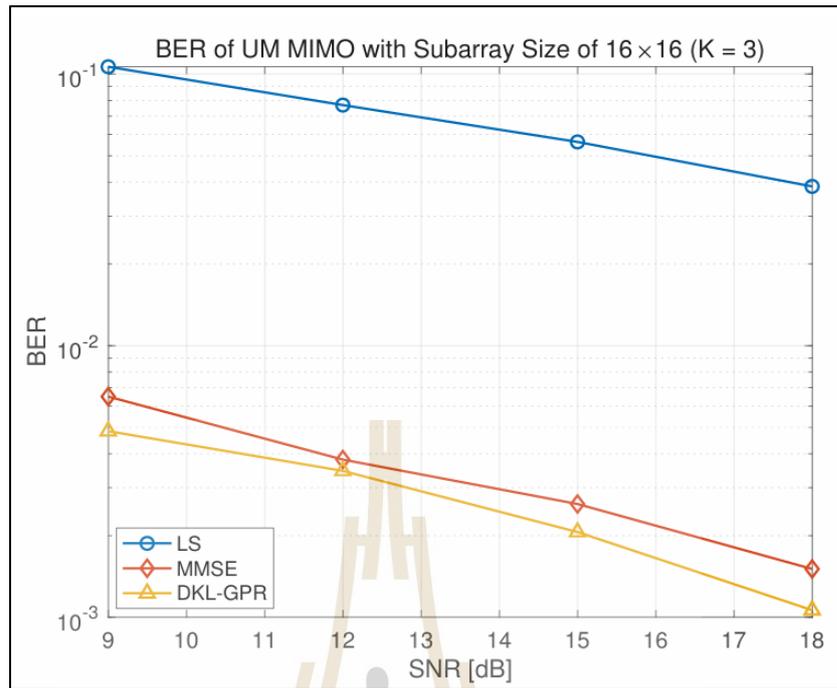


Figure 2.26 Simulation results for channel estimation using LS, MMSE, and DKL Estimators.

In Figure 2.26, presents the simulation results comparing the Bit Error Rate (BER) performance for channel estimation using LS, MMSE, and Deep Kernel Learning with Gaussian Process Regression (DKL-GPR) in a UM MIMO system with a subarray size of 16×16 . The results indicate that DKL-GPR consistently achieves a lower BER compared to MMSE and LS across all SNR values, with DKL-GPR showing the best performance, followed by MMSE and LS, respectively. The reduction in subarray size further enhances the performance of DKL-GPR significantly, due to the improved resolution of the beamspace in channel estimation.

Advantages, no predefined functional form required: GPR does not necessitate assumptions about the relationship between input and output variables, making it adaptable to complex data relationships. High accuracy: GPR often provides more accurate estimates compared to linear methods.

Disadvantages, computational complexity, GPR involves high computational complexity, especially as the size of the data increases, leading to longer

computation times for the covariance matrix. Kernel function selection: The performance of GPR depends on choosing an appropriate kernel function, which may require multiple trials.

In the work by (Murshed, 2022), a beamforming method for UM-MIMO systems operating in the Terahertz (THz) frequency band is presented. This method employs a Deep Neural Network (DNN) that integrates a 1 D Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM), referred to as the Fusion Separation Network. The aim of this approach is to reduce computational complexity while maintaining high spectral efficiency.

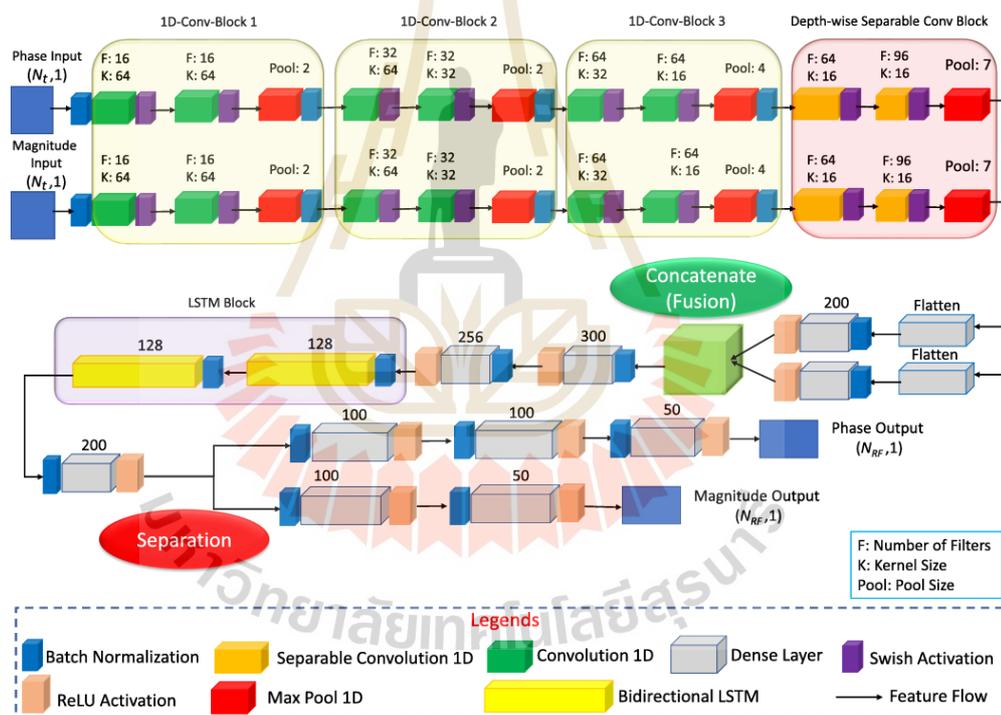


Figure 2.27 Architecture of the proposed 1D CNN-LSTM-based (Murshed, 2022).

In Figure 2.27, the CNN-LSTM model used for fusion and signal separation in UM-MIMO systems. The input comprises phase and magnitude signals that pass through three 1 D-Convolution Blocks. Each block includes convolutions, batch normalization, activation (ReLU), and pooling to reduce data size and extract

essential features. The processed data then goes through a Depth-wise Separable Convolution Block, performing separable convolution and pooling again. The outputs from the CNN are concatenated and fed into LSTM Blocks, which consist of bidirectional LSTM layers to capture temporal relationships. The data processed by the LSTM is separated into phase and magnitude signals through dense layers and flattening, resulting in distinct Phase Output and Magnitude Output for hybrid beamforming in UM-MIMO systems. Batch normalization and activation (Swish) techniques are employed to enhance processing accuracy and reduce computational complexity.

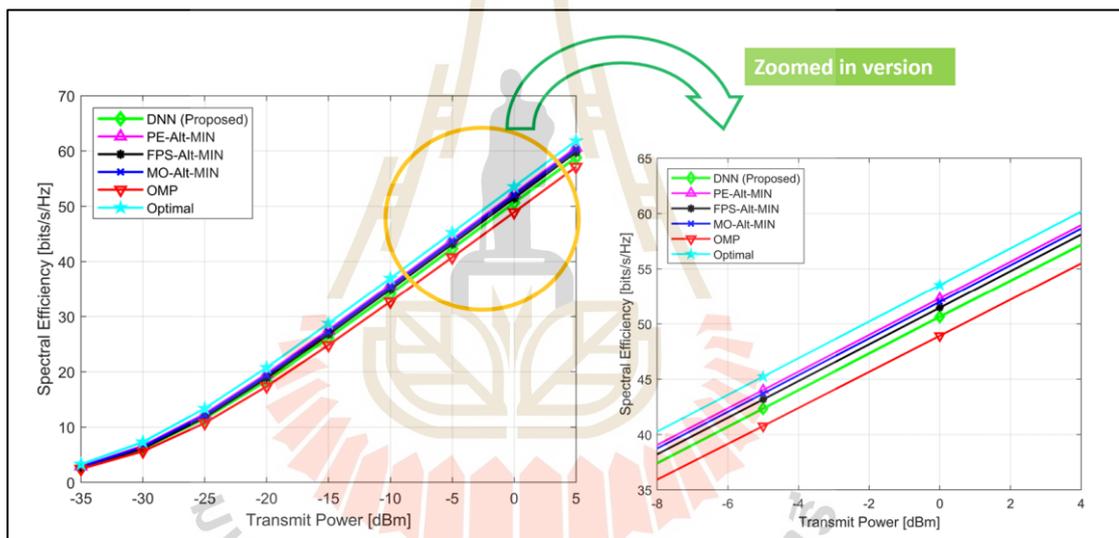


Figure 2.28 SE of different algorithms for varying transmit power with $N_t = N_r = 256$ and $N_s = \text{NRF} = 5$. (Murshed, 2022).

In Figure 2.28, the left graph shows that the Deep Neural Network (DNN) demonstrates spectral efficiency comparable to other techniques as the transmission power increases, particularly at higher power levels. The DNN (Proposed) performs closely to the optimal value while consistently outperforming the Orthogonal Matching Pursuit (OMP) method across all power levels. The right graph zooms in on the power range from -8 dBm to 4 dBm, providing more detailed comparisons. This confirms that the DNN performs well at low transmission power levels, similar to

other Alternating Minimization techniques, but with greater stability and superior performance in several aspects. Although the DNN-based method reduces computational complexity compared to the Alternating Minimization (Alt-Min) method, it still faces limitations in real-time processing due to the model's complexity.

In the research by (Chen, 2021), a Hybrid Spherical- and Planar-Wave Model (HSPM) and channel estimation (CE) method using a Deep Convolutional Neural Network (DCNN) for UM-MIMO systems in THz band are presented. The HSPM integrates the characteristics of both planar-wave and spherical-wave to enhance the accuracy of THz UM-MIMO system analysis and design. Additionally, a two-stage channel estimation mechanism is developed. In the first stage, the DCNN estimates the channel parameters of the reference subarrays. In the second stage, the geometric relationships between the parameters of the remaining subarrays are used to construct the full channel matrix. The simulation results demonstrate that the HSPM provides high accuracy over various communication distances, array sizes, and carrier frequencies. The DCNN quickly and accurately estimates parameters, significantly reducing processing complexity. However, the computations involved with HSPM and using DCNN for channel estimation require substantial computational resources, which can lead to prolonged processing times and increased complexity, as illustrated. Shown in Figure 2.29.

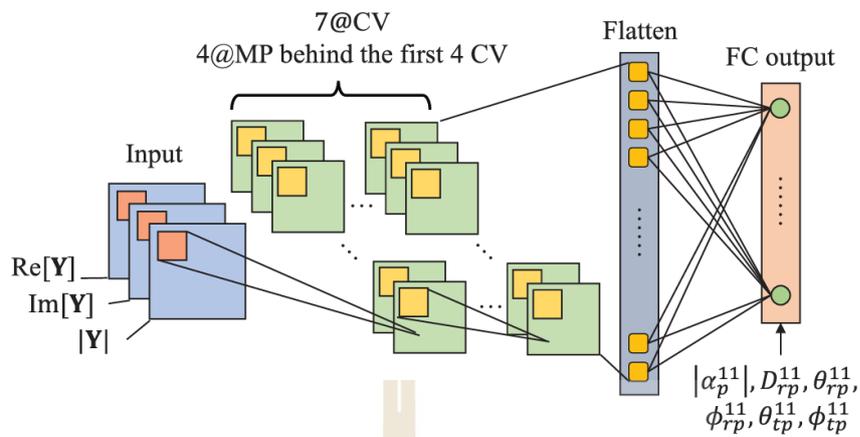


Figure 2.29 The structure of the proposed DCNN network.

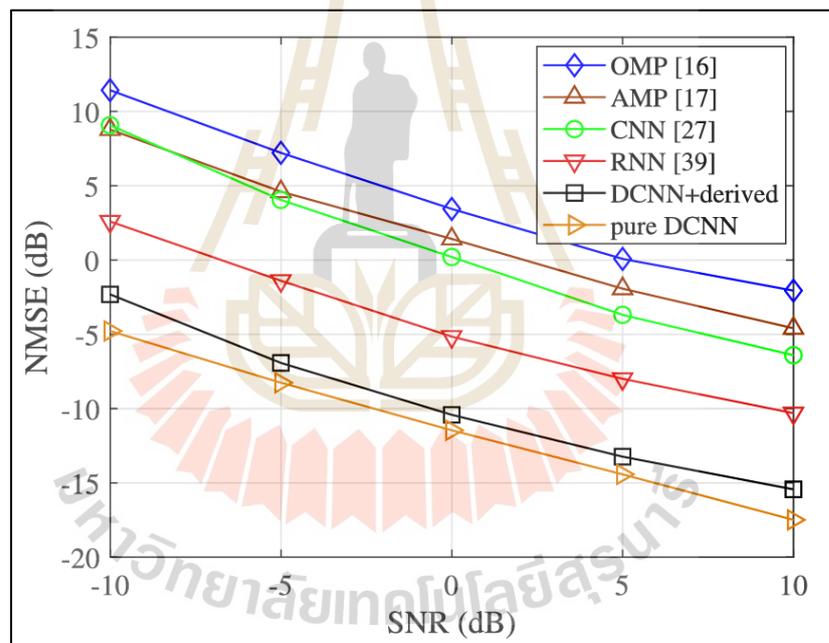


Figure 2.30 NMSE performance of different CE methods.

In Figure 2.30, the comparison of Normalized Mean Squared Error (NMSE) for various channel estimation techniques is shown as the SNR varies from -10 dB to 10 dB. The techniques compared include Orthogonal Matching Pursuit (OMP), Approximate Message Passing (AMP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), DCNN+derived, and pure DCNN. The results indicate that the pure DCNN (orange line) achieves the lowest NMSE across all SNR values, signifying

the highest accuracy in channel estimation. This is followed by DCNN+derived (black line), which has a performance close to pure DCNN. On the other hand, OMP (blue line) exhibits the highest NMSE, indicating the lowest accuracy in channel estimation. These results confirm that DCNN-based techniques are more accurate and efficient in channel estimation compared to traditional methods and other deep learning techniques.

In addition to channel estimation, (Ju, 2024) notes that channel estimation (CE) is often more complex than signal detection due to the need for calculating intricate matrices and the high computational resources required, especially in MIMO systems with a large number of antennas. Channel estimation demands precise parameter tuning and extensive training time. In contrast, signal detection can be performed more quickly, and its efficiency can be significantly enhanced using deep learning techniques such as DetNet and OAMP-Net, or other algorithms.

In the research by Jiyuan Yang [46], a method for signal detection (SD) in Ultra-Massive Multiple-Input Multiple-Output (UM MIMO) systems is presented using the Information Geometry Approach (IGA). This method detects signals by calculating the a posteriori probability estimates of the transmitted symbol vectors and then finding the maximum of these probability estimates using an iterative m-projection process between submanifolds with different constraints. Simulation results show that IGA-SD is a promising and efficient method for signal detection in UM MIMO systems. However, there are some limitations, such as the constraints of the Central Limit Theorem (CLT): the accuracy of CLT-based estimation is high only when the number of antennas and users is large. In cases where the number of antennas or users is small, the accuracy of the estimation may decrease.

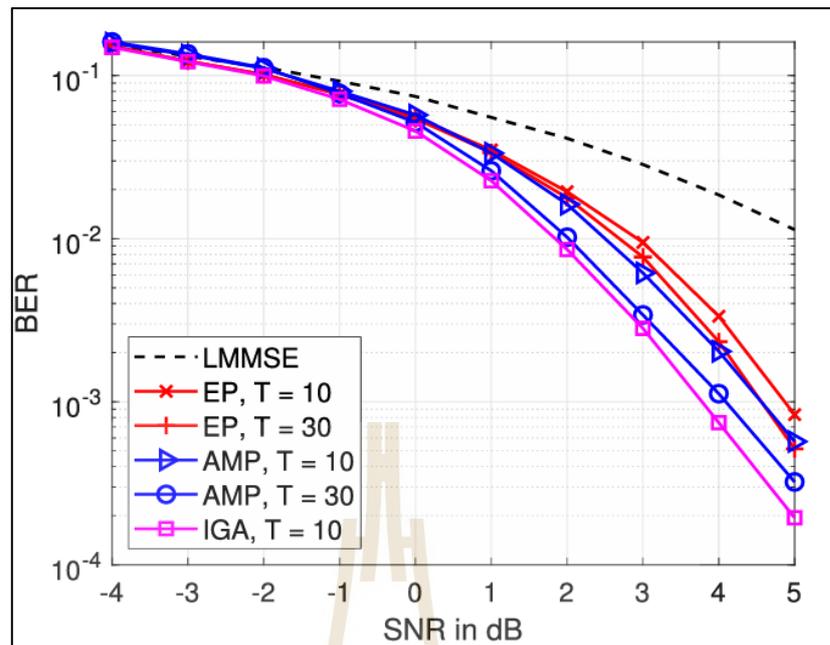


Figure 2.31 BER performance of IGA compared with AMP, EP and LMMSE under 4-QAM.

In Figure 2.31, compares the Bit Error Rate (BER) of various signal detection techniques in UM-MIMO systems as the Signal-to-Noise Ratio (SNR) varies from -4 dB to 5 dB. The techniques compared include Linear Minimum Mean Squared Error (LMMSE), Expectation Propagation (EP) with 10 and 30 iterations, Approximate Message Passing (AMP) with 10 and 30 iterations, and Information Geometry Approach (IGA) with 10 iterations. The results show that IGA (pink line) achieves the best performance across all SNR values, with the lowest BER, indicating its superior capability in minimizing signal detection errors. This is followed by AMP and EP, while LMMSE has the highest BER, indicating the lowest performance in signal detection. Multiple iterations improve the accuracy of signal detection and reduce the BER, but they also increase complexity, processing time, and resource usage.

2.10 Application of Services 5G and 6G

3GPP Commits to Develop 6G Specifications (3GPP), there was a discussion about information on the range of Bit Error Rate (BER) values, a crucial metric for assessing the quality of data communication systems. Different ranges of BER values impact various applications and have distinct considerations (Uusitalo, 2021).

Table 2.6 The standard Bit Error Rate (BER) is determined by the requirements of the application and the Quality of Service (QoS).

BER	Applications	Key Considerations
10^0	Testing or transmitting insignificant data	High data loss, suitable for testing or transmitting easily lost data
10^{-1}	Communications with acceptable errors, high noise environments	High data loss, suitable for non-critical communications such as servers with frequent updates
10^{-2}	Communications requiring moderate reliability, general communications with acceptable errors	Low data loss, suitable for general communications such as medium quality video streaming or voice communications
10^{-3}	General communications	Suitable for communications requiring moderate reliability such as IoT sensor data
10^{-4}	Communications requiring high accuracy	Suitable for communications needing high accuracy such as high-resolution video streaming
10^{-5}	Communications requiring high reliability	Suitable for high reliability communications such as unit control or medium distance communications
10^{-6}	Communications requiring very high reliability	Suitable for very high reliability communications such as remote medical applications or high precision unit control

In this Table 2.6, it can be seen that higher BER values (e.g., 10^0) are suitable for testing or transmitting less critical information due to higher data loss. Conversely, lower BER values (e.g., 10^{-6}) are appropriate for communication requiring high reliability, such as telemedicine or precision control in critical sectors.

In 5G and 6G networks (Jain, 2022; Nguyen, 2021), selecting the appropriate services that align with the specified BER values in the table is crucial to achieving optimal communication performance tailored to the specific requirements of each service from Table 2.7. Enhanced Mobile Broadband (eMBB) has a BER range of $10^{-4} - 10^{-2}$, suitable for high-bandwidth communications such as high-definition video streaming, high-speed internet usage, and video conferencing. This service is ideal for applications that require high data volumes and speeds but can tolerate some errors. In contrast, Ultra-Reliable and Low-Latency Communications (uRLLC) with a BER range of $10^{-5} - 10^{-3}$ focuses on communications that demand high reliability and low latency, such as remote control, telemedicine, and industrial automation. This service requires extreme accuracy and reliability in data transmission. Massive Machine Type Communications (mMTC), with a BER range of $10^{-6} - 10^{-2}$, is designed for communications involving a large number of IoT devices, such as sensors in smart cities and supply chain tracking systems. It emphasizes connecting a vast number of devices simultaneously while maintaining acceptable data transmission stability. Long-Distance and High-Mobility Communications (LDHMC), with a BER range of $10^{-5} - 10^{-3}$, is suited for communications that involve high mobility and long distances, such as high-speed vehicle communications and communication between space stations and the earth. This service demands high reliability in data transmission to ensure no data loss, even under high mobility conditions. Finally, Extremely Low-Power Communications (ELPC) with a BER range of $10^{-6} - 10^{-4}$ is used for ultra-low-power communications, such as wearable health devices and low-power sensors in IoT systems. This service is ideal for applications that require energy efficiency while maintaining high reliability in data transmission.

Table 2.7 Services 5G and 6G with BER.

Services	Network	BER
Enhanced Mobile Broadband (eMBB)	5G/6G	$10^{-4} - 10^{-2}$
Ultra-Reliable and Low-Latency Communications (uRLLC)	5G/6G	$10^{-5} - 10^{-3}$
Massive Machine Type Communications (mMTC)	5G/6G	$10^{-6} - 10^{-2}$
Long-Distance and High-Mobility Communications (LDHMC)	6G	$10^{-5} - 10^{-3}$
Extremely Low-Power Communications (ELPC)	6G	$10^{-6} - 10^{-4}$

In Figure 2.32, in general, models can process various applications rapidly due to their workflow being divided into two main stages.

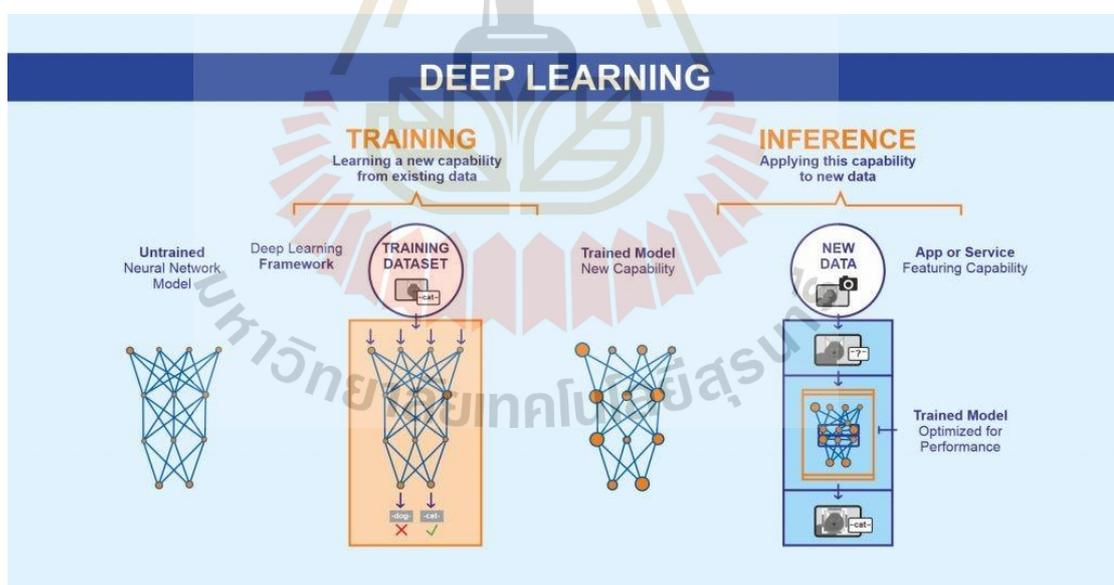


Figure 2.32 Training and inference for application to using

Training Stage: In this stage, the model is trained using pre-prepared data (Training Data), where the model learns the characteristics and patterns of the data through the adjustment of parameters (e.g., weights and biases). The training process is

typically time-consuming and resource-intensive (e.g., using GPUs or TPUs) but is done only once or when the model needs to be updated.

Inference Stage: Once the model is trained, it can be used to process new data immediately without needing to retrain the model. In this stage, the model takes new input data and processes it through its layers to produce an output, such as classifying signals or analyzing images. This process is fast and efficient because the model already possesses the knowledge gained from the training stage. The inference process can be executed swiftly because CNNs are well-suited for parallel processing, allowing them to fully leverage high-performance hardware like GPUs and CPUs. Therefore, when deploying a trained CNN model, you simply feed the input data into the model, and it will quickly and accurately generate the desired output.

2.11 Summary

All of this pertains to the methodologies and background of the thesis, including the selection of algorithms for improving channel estimation in UM-MIMO systems. The researchers aim to apply deep learning to 6G communication systems, considering the Terahertz frequency band and signal detection to identify the most suitable methods for this research. The researcher has surveyed various academic articles such as academic journals and academic conferences. Which has been summarized in the following Table 2.8. The research survey results found that there were only three articles using signal detection methods without applying ML or DL. Additionally, most articles focused on hybrid precoding or hybrid beamforming. Only two articles employed the spatial multiplexing system, incorporating different channel models and methods of interest. Therefore, this thesis focuses on applying DL in the spatial multiplexing system using the Ray/cluster channel model with the signal detection method in the UM-MIMO system.

Table 2.8 Survey of research related to the proposed method.

Ref.	System model		Channel model			Method			Model	
	HB	SM	R/C	RC	LoS	CE	SD	Etc.	ML	DL
(Sarieeddeen, 2019)	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
(Nie, 2019)	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗
(Murshed, 2022)	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓
(Chen, 2021)	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓
(Ju, 2024)	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓
(Yang, 2024)	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗
(Yan, 2020)	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
(Yan, 2022)	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
(Morsali, 2020)	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗
(Hu, 2022)	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
(Jamali, 2020)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗
(Yan, 2021)	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
(Elbir, 2021)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓
(Sarieeddeen, 2021)	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗
(Tarboush, 2021)	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
(Shahjalal, 2021)	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗
Proposed in Thesis	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓

Where is HB: Hybrid Beamforming, SM: Spatial Multiplexing, CE: Channel Estimation, SD: Signal Detection, R/C: Ray/Cluster Channel, RC: Rayleigh Channel, LoS: Line of Sight.

CHAPTER III

METHODOLOGY

3.1 Introduction

In this chapter, the methodological approach adopted for detecting signals in UN-MIMO systems is thoroughly explored. This exploration is grounded in the advanced application of deep learning techniques, with a spotlight on the PMS-CNN. The investigation particularly emphasizes the integration of CNN within the ELM 1 Dimension (1D) framework. The aim is to critically assess the effectiveness of these innovative methodologies for signal detection tasks.

Furthermore, this study undertakes a comparative analysis by juxtaposing traditional signal detection methods, such as ZF and MMSE, against a spectrum of machine learning strategies. These include the ELM, RELM, and ORELM. This comparative evaluation focuses on gauging the performance of these methodologies through established metrics, namely the MSE and BER. The objective is to provide a nuanced understanding of the operational effectiveness of these approaches in the context of signal detection, thereby contributing to the broader discourse on technological advancements in signal processing.

3.2 System Model

The use of MATLAB for simulating UM-MIMO systems is widely accepted among researchers. In this study, we configured the simulations to include 256 antennas at both the transmitter and receiver ends to enhance the efficiency of signal detection. Such a configuration enables the formulation of relevant equations. For example, consider a simplistic system model that includes transmitting antennas M_T and receiving antennas M_R , embodying the concept of spatial multiplexing.

The relationship between the transmitted and received signals in this framework can be summarized using the memoryless MIMO flat fading channel model, represented by the narrowband model. As show in Figure 3.1.

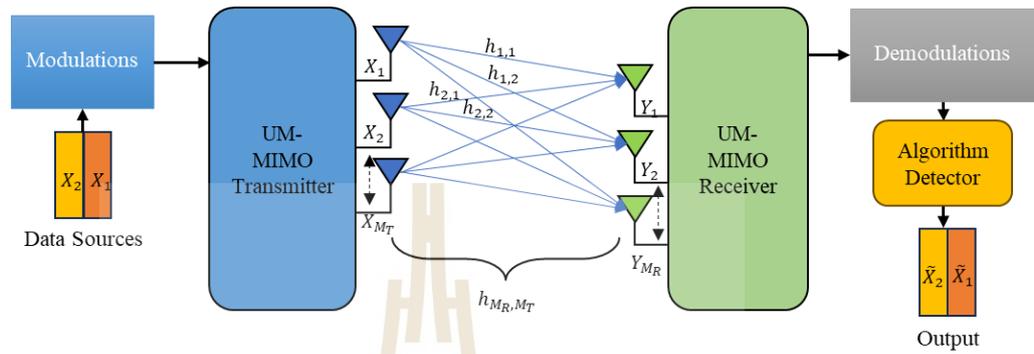


Figure 3.1 UM-MIMO system model.

$$Y = Hx + n \quad (3.1)$$

Within the framework of equation,

Y is the received signal vector at the M_R receive antennas.

H is the channel matrix between M_T transmit antennas and M_R receive antennas, showing how the signal from each transmit antenna affects each receive antenna.

x is the transmit signal vector from the M_T transmit antennas.

n is the additive white Gaussian noise vector at the M_R receive antennas.

The equation can be represented in matrix form as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{M_R} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,M_T} \\ h_{21} & h_{22} & \cdots & h_{2,M_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_R,1} & h_{M_R,2} & \cdots & h_{M_R,M_T} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M_T} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{M_R} \end{bmatrix} \quad (3.2)$$

3.3 Channel Model

This section delves into the examination of various channels, predicated on the potentiality of realistic scenarios. To approximate real-world conditions as closely as possible, the Saleh-Valenzuela signal model has been employed. This model posits that the signal comprises a conglomerate of discrete ray bundles, a characterization that is quintessential for mmWave signals, notable for their high reflectivity and minimal dispersion. From a mathematical perspective, the ray/cluster signal matrix can be delineated as follows.

$$H = \sum_{v=1}^{N_{clust}} \sum_{u=1}^{N_{ray}} \beta_{u,v} a_{M_R} (AoA_{u,v}) a_{M_T} (AoD_{u,v})^* \quad (3.3)$$

Within the ambit of Massive MIMO systems, the channel matrix H embodies the cumulative effect of myriad signal paths interfacing the transmitter and receiver arrays. The variables u and v within the equation are indices representing specific subarrays within the transmitting array M_T and the receiving array M_R , correspondingly. The propagation of signals is envisaged through the prism of several 'clusters,' each imbued with unique attributes. The notation N_{clust} designates the number of such clusters, serving as a pivotal parameter in the channel modeling process that encapsulates the environment's multipath complexity. Each cluster comprises numerous 'rays', denoted by N_{rays} , symbolizing individual signal trajectories that cumulatively augment the aggregate signal received. The symbol $\beta_{u,v}$ delineates the complex gain associated with a signal path connecting the u -th subarray at the transmitter to the v -th subarray at the receiver, incorporating both signal attenuation and phase shift phenomena encountered during transmission. Additionally, $(AoD_{u,v})$ and $(AoA_{u,v})$ are paramount angular parameters, with $(AoD_{u,v})$, the Angle of Departure, delineating the vector along which the signal projects from the transmitting subarray u , and $(AoA_{u,v})$, the Angle of Arrival,

indicating the vector along which the signal is apprehended by the receiving subarray v . Lastly, the functions $\mathbf{a}_{M_T}(\text{AoD}_{u,v})$ and $\mathbf{a}_{M_R}(\text{AoA}_{u,v})$ articulate the antenna array responses at the transmitter and receiver, respectively. These functions play a crucial role in modulating the signal processing by the antenna arrays, predicated on the angles of departure and arrival. The notation '*' signifies the conjugate transpose. Which has a communication structure as show in the following Figure 3.2.

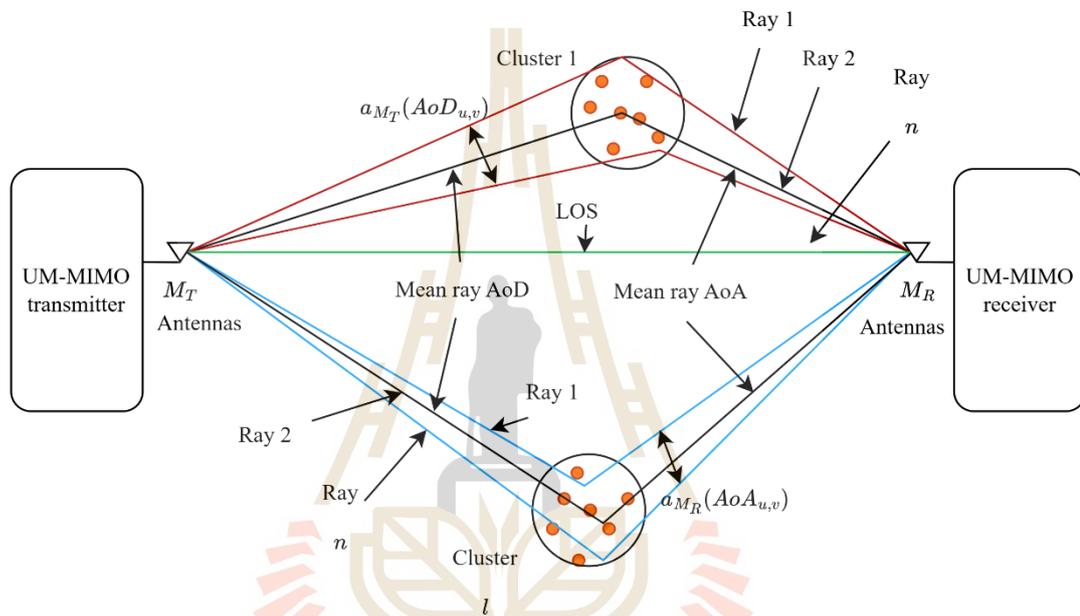


Figure 3.2 Ray/cluster-based UM-MIMO channel model schematic.

3.4 Data preparation

Information from Figure 3.1, in studying the constellation diagrams of various QAM schemes, we found that using 256QAM has significant drawbacks in terms of Bit Error Rate (BER). The increased number of signal levels in 256QAM results in smaller distances between each signal level, making the signals more susceptible to noise and distortion in the channel. Consequently, the BER of 256QAM is higher compared to modulation schemes with fewer signal levels, such as 64QAM or 16QAM. Therefore, in environments with high noise levels or poor channel quality, using 256QAM may be unsuitable as it leads to more errors in data transmission. As show

in Figure 3.3. Given this, the use of CNN for signal detection in UM-MIMO systems is an appropriate choice. CNN can learn and distinguish complex signal characteristics from the provided data. Using modulation signals with 256QAM helps the model capture the intricate relationships between the signals and the distortions occurring in the channel effectively, particularly in high-noise conditions. Additionally, employing CNN improves the accuracy of signal prediction and detection, enhancing the overall reliability and precision of UM-MIMO systems.

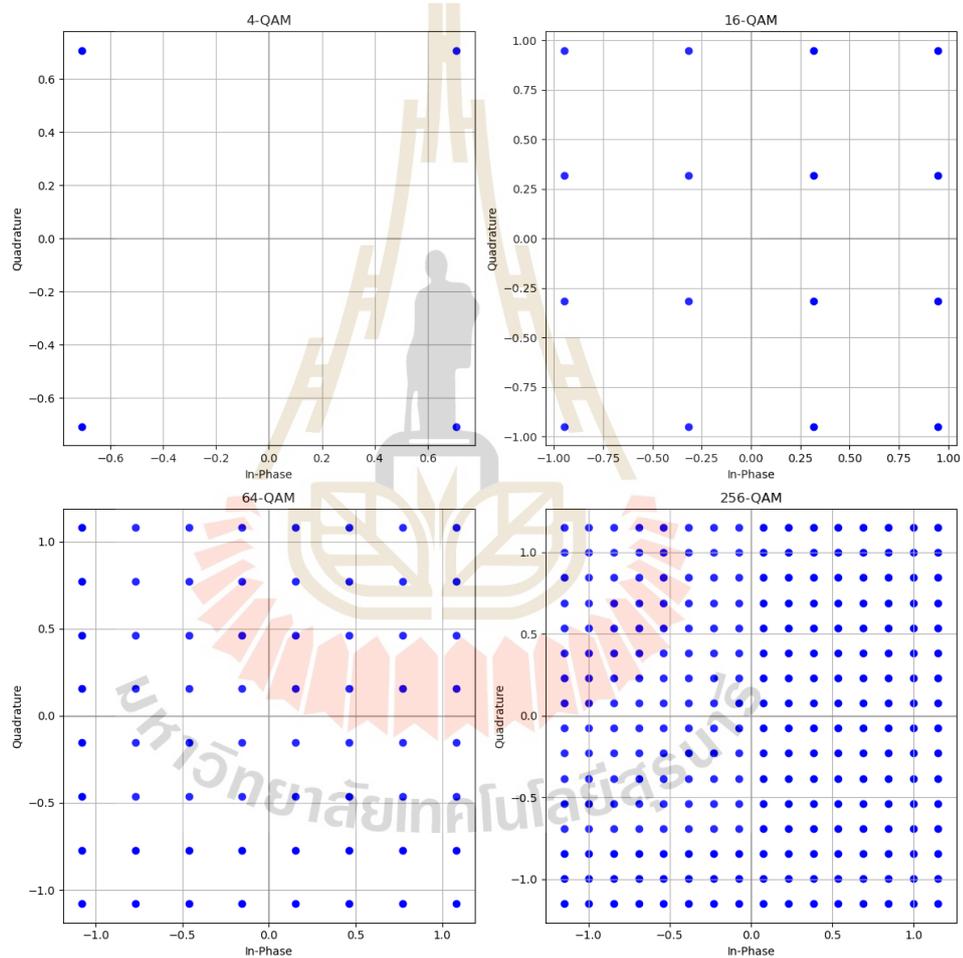


Figure 3.3 QAM Constellation Diagram.

This section describes the preparation of data to be imported into the proposed model. which the data obtained from the system simulation UM-MIMO data selection uses appropriate data sets related to signal detection principles. And the dataset will be divided into training set and test set. But the process of preparing

a data set requires separating the real and imaginary parts of complex numbers. This method is called Because of the programming limitations of deep learning processing, values cannot be directly inputted. This is an advantage as researchers can learn the dataset and validate the values that will be used in deep learning. The researcher will explain the methods for preparing the data as follows.

The first step, from Figure 3.4, in red line box involves collecting test results from simulating the communication system, incorporating real and imaginary values. The size of this dataset is contingent upon the number of antennas specified for the task. This initial preparation phase is crucial for accurately reflecting the complexity and dynamics of the communication system under study.

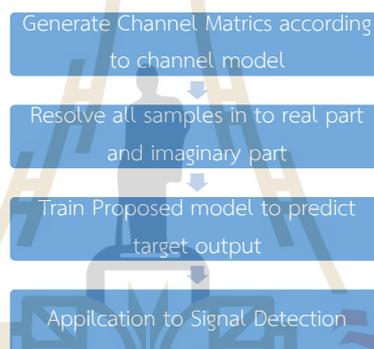


Figure 3.4 Workflow of the process.

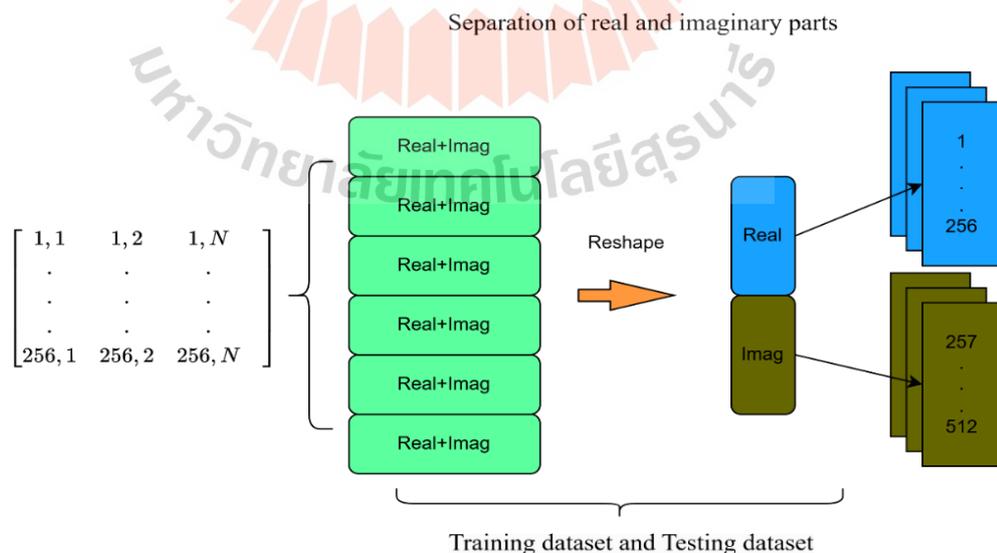


Figure 3.5 The Process about of Separation of real and imaginary parts

In Figure 3.5, the second step involves manipulating the data matrix through reformatting. Because understanding the behavior of neural networks is important for preparing training data, in Python, using a library like PyTorch, there are specific requirements for structuring the data to facilitate processing. This step must ensure that the dataset is in the correct format for the neural network to process it efficiently. It emphasizes the importance of fitting the data to the input requirements of the neural network model.

3.5 Convolutional Neural Network (CNN)

In this section, the application of CNN for signal detection in UM-MIMO systems involves several interconnected mathematical steps. The following combines the equations from each step to provide a comprehensive understanding of CNN operations in this context, as shown in Figure 3.4.

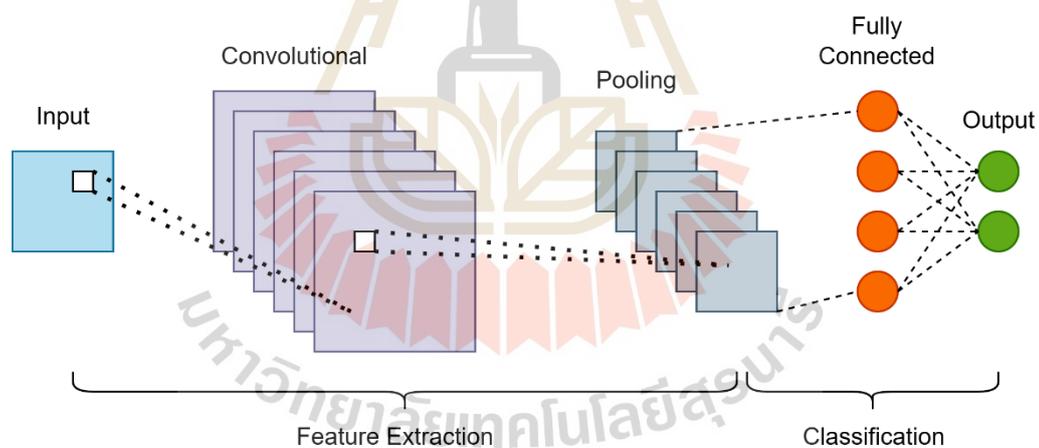


Figure 3.6 CNN Model Architecture.

As explained in Figure 3.6, there are four main steps: Convolution Operation, Activation Function, Pooling Layer, and Fully Connected Layer, leading to the final predicted value.

Convolution Operation, the convolution operation involves sliding a kernel w over the input x (which represents received signal data) to create a feature map S :

$$S(t) = (x * w)(t) = \sum_{i=-m}^m x(t+i)w(i) \quad (3.4)$$

Activation Function, the feature map $S(t)$ is then passed through an activation function to introduce non-linearity. The commonly used ReLU function is defined as:

$$f(S(t)) = \max(0, S(t)) \quad (3.5)$$

Pooling Layer, the output from the activation function $f(S(t))$ is then sent to a pooling layer, such as Max Pooling, to reduce the dimensionality of the data:

$$P(t) = \max(f(S(t)), f(S(t+1)), \dots, f(S(t+n-1))) \quad (3.6)$$

Fully Connected Layer, the output from the pooling layer $P(t)$ is then flattened into a vector and passed through a fully connected layer, which aggregates the information and produces the final output:

$$z = Wp + b \quad (3.7)$$

Where W is the weight matrix. p is the vector of the pooled results. b is the bias vector. z is the output vector.

Combining all the equations, we get the following sequence, summarizing the entire process.

$$z = W \cdot \max(\max(0, \sum_{i=-m}^m x(t+i)w(i)), \max(0, \sum_{i=-m}^m x(t+1+i)w(i)), \dots, \max(0, \sum_{i=-m}^m x(t+n-1+i)w(i))) + b \quad (3.8)$$

This interconnected sequence of steps shows how numerical data representing signals in UM-MIMO systems is processed through the different layers of a CNN. The process involves extracting important features, introducing non-linearity, reducing dimensionality, and finally connecting and aggregating the information to predict the output, which is crucial for effective signal detection.

3.6 Parallel Multi-Scale Convolutional Neural Network (PMS-CNN)

In the field of signal detection and regression analysis, traditional CNN are commonly used for feature extraction and prediction tasks. However, CNNs with a single scale of convolutional filters might not be sufficient to capture the multi-scale nature of signal data, where patterns of various sizes and temporal lengths are present. To address this limitation, we propose PMS-CNN model to enhance feature extraction by incorporating multiple convolutional layers with different kernel sizes, capturing both fine and coarse features simultaneously.

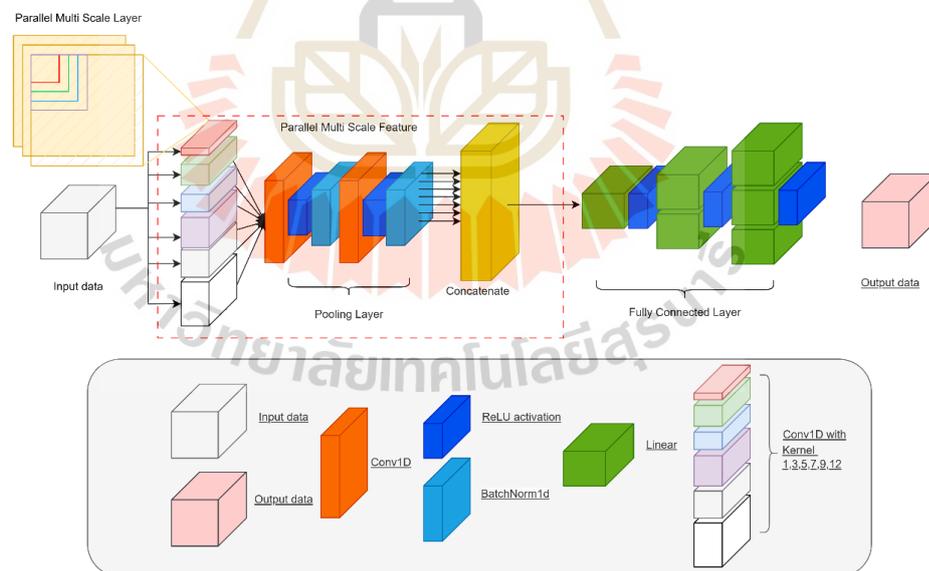


Figure 3.7 The proposed PMS-CNN architecture.

In Figure 3.7, PMS-CNN operations involve using multiple kernels w_j (where j represents different kernel sizes: 1, 3, 5, 7, 9, 12) sliding over the input x (which represents received signal data) to create multiple sets of feature maps S_j .

$$S_j(t) = (x * w_j)(t) = \sum_{i=-m_j}^{m_j} x(t+i)w_j(i) \quad (3.9)$$

Where S_j is the feature map for kernel size j at position t . $x(t+i)$ is the input data at position $(t+i)$. $w_j(i)$ is the weight of the kernel of size j at position i . m_j is the half-width of the kernel size j .

Activation Functions, each set of feature maps $S_j(t)$ is then passed through an activation function to introduce non-linearity. For the ReLU function.

$$f(S_j(t)) = \max(0, S_j(t)) \quad (3.10)$$

Where $f(S_j(t))$ is the output of the ReLU function applied to $S_j(t)$.

Batch Normalization, after ReLU activation, batch normalization is applied to each feature map $f(S_j(t))$.

$$\hat{S}_j(t) = \frac{S_j(t) - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}} \quad (3.11)$$

Where $\hat{S}_j(t)$ is the normalized feature map. μ_j is the mean of the feature map $S_j(t)$. σ_j^2 is the variance of the feature map $S_j(t)$. ϵ is a small constant added for numerical stability.

Multi-Scale Pooling Layers, the normalized feature maps $\hat{S}_j(t)$ are then sent to multi-scale pooling layers, such as Max Pooling, to reduce the dimensionality:

$$P_j(t) = \max(f(S_j(t)), f(S_j(t+1)), \dots, f(S_j(t+n-1))) \quad (3.12)$$

Where $P_j(t)$ is the pooled feature map. n is the pooling window size.

Feature Map Concatenation, the pooled feature maps $P_j(t)$ from different kernel sizes are concatenated to form a combined feature map P_{concat} .

$$P_{\text{concat}} = [P_1(t), P_2(t), \dots, P_k(t)] \quad (3.13)$$

Where P_{concat} is the concatenated feature map containing features from all scales.

Fully Connected Layers (Linear), the concatenated feature map P_{concat} is flattened into a vector and passed through fully connected (linear) layers:

$$z = WP_{\text{concat}} + b \quad (3.14)$$

Where W is the weight matrix of the fully connected layer. P_{concat} is the flattened vector of the concatenated multi-scale pooled results. b is the bias vector. z is the output vector representing the detected signal.

Combining all the equations, we get the following sequence, summarizing the entire process.

$$z = W \cdot [\max(\max(0, \sum_{i=-m_1}^{m_1} x(t+i)w_1(i)), \dots), \max(\max(0, \sum_{i=-m_k}^{m_k} x(t+i)w_k(i)), \dots)] + b \quad (3.15)$$

This interconnected sequence of steps shows how numerical data representing signals in UM-MIMO systems is processed through the various layers of PMS-CNN. The process involves extracting important features at multiple scales, introducing non-linearity, reducing dimensionality, and finally connecting and aggregating the information to predict the output, which is crucial for effective signal detection. The PMS-CNN architecture leverages the concept of multi-scale feature extraction, based on the observation that signals often contain patterns of varying sizes. Traditional CNNs with fixed kernel sizes may not effectively capture these diverse patterns. By employing multiple parallel convolutional layers with different kernel sizes, PMS-CNN

can simultaneously extract fine, medium, and coarse features, providing a more comprehensive representation of the signal data. The use of parallel convolutional layers allows the network to process different aspects of the input signal independently. This is particularly useful in signal detection tasks where the relevant features may occur at different scales. The subsequent channel pooling and concatenation steps ensure that the extracted features are effectively combined, enhancing the network's ability to learn and generalize from the data.

3.7 Extreme Learning Machine (ELM) Techniques

The ELM is a notable breakthrough in the domain of machine learning, primarily designed to tackle the computing requirements and efficiency limitations of conventional learning methods. ELM was developed in 2004 to address the increasing demand for processing frameworks that can effectively handle intricate, high-dimensional data sets and produce efficient learning results. This situation emphasized the need for creative methods that might provide both adaptability in data management and enhanced algorithmic efficiency. The fundamental essence of ELM is rooted in the architectural principles of Single-layer Feedforward Neural Networks (SLFNs), a deliberate design decision that significantly impacts its operational effectiveness. The SLFN design is distinguished by its inherent simplicity, as it comprises a solitary layer of concealed nodes that facilitate the conversion of input data into a linearly separable or more readily modelled space. Subsequently, an output layer is employed to establish a mapping between these transformed inputs and the intended outputs. The primary differentiation of ELM resides in its learning mechanism, specifically in the random assignment and fixed weighting of the weights between the input layer and the hidden layer. ELM circumvents the computationally demanding process of iteratively adjusting weights by backpropagation, unlike traditional neural networks. In ELM, the weights between the hidden layer and the output layer are the only parameters that are optimized. These weights are typically computed in a single step by solving a linear solution. This

methodology effectively decreases the duration of training while maintaining optimal performance. One notable characteristic of ELM is its ability to provide universal approximations and exhibit strong generalization performance. Although the starting weights and biases are set randomly, ELM may effectively simulate complex nonlinear functions and reach generalization performance that is equivalent to or even better than typical feedforward neural networks. Additionally, ELM often achieves a significantly quicker learning speed. The efficiency and simplicity of ELM render it highly attractive for a wide range of applications, encompassing function approximation, classification, regression, and deep learning tasks, among others. ELM presents a viable alternative to gradient-based learning methods by addressing their limitations, including local minima, overfitting, and the requirement for extensive hyperparameter tuning. This approach capitalizes on the advantages of feedforward neural networks while substantially reducing the computational load associated with their training process.

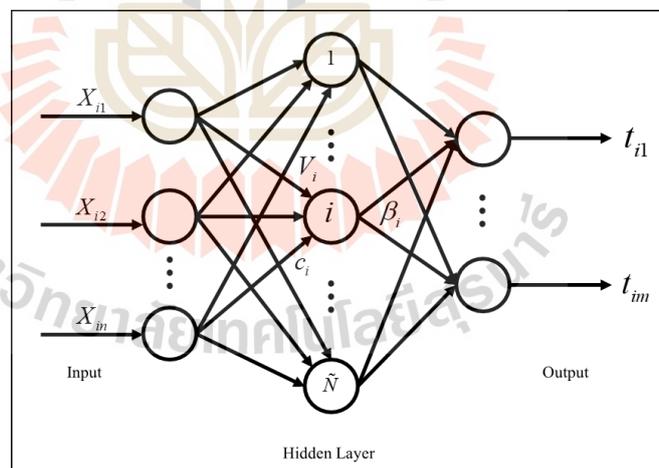


Figure 3.8 The Neural System Structure of Extreme Learning Machine.

In Figure 3.8, we illustrate the operational structure of the ELM, where the variables involved can be represented by the following equation:

$$\hat{t}_j = \sum_{i=1}^N \beta_i o(c_i \cdot X_j + V_i), \quad j=1,2,\dots,N \quad (3.16)$$

Where is X_i denotes the input data values.

t_i represents the output data values.

c_i is the weight vector of the input data for the hidden layer, expressed as $c_i = [c_{i1}, c_{i2}, \dots, c_{im}]^T$.

β_i denotes the output weight vector of the hidden layer, with $\beta = [\beta_{i1}, \beta_{i12}, \dots, \beta_{im}]^T$.

V_i is the bias term for the hidden layer.

o represents the activation function of the hidden layer.

This equation encapsulates the computational process within ELM, highlighting the transformation of input data through the hidden layer using a combination of weights, biases, and an activation function, to produce the predicted output \hat{t}_j for each input X_j . Given the complete dataset of N as previously described, Equation 3.16 can be analogously represented as a linear system:

$$\mathbf{H}\beta = \mathbf{T} \quad (3.17)$$

where the activation function o applied to inputs and parameters over all data instances is expressed as.

$$o(c_i, \dots, c_N, V_i, \dots, V_N, X_i, \dots, X_N) = \begin{bmatrix} o(c_1 \cdot X_1 + V_1) & \cdots & o(c_N \cdot X_1 + V_N) \\ \vdots & \ddots & \vdots \\ o(c_1 \cdot X_N + V_1) & \cdots & o(c_N \cdot X_N + V_N) \end{bmatrix}_{N \times N} \quad (3.18)$$

Here, \mathbf{H} denotes the output matrix of the hidden layer nodes in Single-hidden Layer Feedforward Neural Networks (SLFNs), represented by the i -th row and j -th

column of outcomes from the activation function matrix σ in the hidden layer. The vectors β and \mathbf{T} are defined as follows.

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}_{N \times m}^T, \quad \mathbf{T} = [\mathbf{T}_1, \dots, \mathbf{T}_N]_{N \times m}^T \quad (3.19)$$

To minimize the network cost function $\mathbf{T} - \mathbf{H}\beta$, where \mathbf{T} is the target output matrix, ELM theory posits that the parameters of the hidden layer nodes c_i and X_i can be randomly assigned without the need for iterative adjustment based on incoming data. Thus, Equation 3.16 becomes a linear model equation, and the output weights can be analyzed by deriving the least squares solution of the linear system as follows.

$$\beta = \mathbf{H}^\dagger \mathbf{T} \quad (3.20)$$

Where \mathbf{H}^\dagger represents the Moore-Penrose pseudoinverse of the hidden layer output matrix \mathbf{H} , given that the number of training samples typically exceeds the number of hidden layer nodes. This can be reformulated as:

$$\beta = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{T} \quad (3.21)$$

This approach enables the determination of the output weights for ELM, showcasing its efficiency in solving regression and classification problems within a linear framework, without necessitating iterative training methods commonly used in other neural network models.

3.8 Regularized Extreme Learning Machine (RELM) Techniques

The ELM framework lies in its adeptness at mitigating training errors. Nonetheless, this proficiency may precipitate an undue escalation in model complexity, or overfitting, thereby attenuating the model's predictive precision. Bartlett's theory posits that, within the context of SLFNs, an inverse relationship exists between the magnitude of weights, training errors, and the network's generalization capability; a reduced norm of weights and training errors enhances the model's generalizability. Consequently, an ELM model poised for optimal generalization should ideally facilitate an equilibrium between minimizing training error and regulating output weights' norm. This equilibrium can be achieved by standardizing with a parameter C , thus engendering the RELM, distinguished by its enhanced generalizability due to this regularization criterion. A salient feature distinguishing RELM from its ELM counterpart is the concurrent diminution of training errors and the imposition of a regularization scheme on the output weights via a regularization parameter. This dynamic can be mathematically articulated as:

$$\min_{\beta} C \| T - H\beta \|^2 + \|\beta\|^2 \quad (3.22)$$

This equation is further distilled into an equivalent constrained optimization problem.

$$\min_{\beta} C \| a \|^2 + \|\beta\|^2 \quad (3.23)$$

where $a = T - H\beta$, and reframing this in the Lagrangian formalism yields:

$$L(\beta, a, \lambda) = C \| a \|^2 + \|\beta\|^2 + \lambda^T (H^T - H\beta - a) \quad (3.24)$$

Here, $a = [a_1, a_2, a_3, \dots, a_N]^T$ denotes the vector of error variables for N instances, and λ represents the vector of Lagrange multipliers. The Karush–Kuhn–Tucker (KKT) optimization conditions are thus defined.

$$\begin{cases} \frac{\partial L}{\partial \beta} = 0 \rightarrow 2\beta - H^T \lambda = 0 \\ \frac{\partial L}{\partial a} = 0 \rightarrow 2Ca - \lambda = 0 \\ \frac{\partial L}{\partial \lambda} = 0 \rightarrow T - H\beta - a = 0 \end{cases} \quad (3.25)$$

This formulation facilitates the determination of β as.

$$\hat{\beta} = (H^T H + IC)^{-1} H^T T \quad (3.26)$$

It is pivotal to acknowledge that when the training sample size is lesser than the number of hidden nodes, a methodological adjustment can be applied to decrease computational complexity:

$$\hat{\beta} = H^T (HH^T + IC)^{-1} T \quad (3.27)$$

Through this methodology, the ELM and its regulated variant, RELM, strategically optimize the interplay between training error reduction and weight norm regularization, thus achieving superior generalization capabilities.

3.9 Outliner-Robust Extreme Learning Machine (ORELM) Techniques

Typically, ORELM anomalies constitute only a fraction of the total training samples for the training error a . This phenomenon, while not comprehensively elucidated, is well understood to be more accurately reflected by the ℓ_0 -norm rather than the ℓ_2 -norm. Consequently, the pursuit for minimizing the output

weights β incorporates a reduced ℓ_2 -norm to diminish the training error a , as delineated by:

$$\min_{\beta} C \|a\|_0 + \|\beta\|_2^2 \quad (3.28)$$

where it depends on $a = T - H\beta$.

However, the problem articulated in Equation 3.28 addresses a non-convex challenge, which can be more straightforwardly resolved by reformulating into a convex relaxation. This alternative approach retains the distributive characteristics essential for detection and component analysis without forfeiting the model's sparsity, achievable through the ℓ_1 -norm. Clearly, replacing the ℓ_0 -norm with the ℓ_1 -norm not only validates the sparsity assumption but also facilitates an overall dimensional reduction. This results in the formulation:

$$\min_{\beta} \|a\|_1 + \frac{1}{C} \|\beta\|_2^2 \quad (3.29)$$

based on $a = T - H\beta$.

Equation 3.29 proposes a convex optimization challenge with constraints, particularly amenable to the Augmented Lagrange Multiplier (ALM) methodology. The augmented Lagrangian function is thus defined.

$$L_{\mu}(a, \beta, \lambda) = \|a\|_1 + \frac{1}{C} \|\beta\|_2^2 + \lambda^T (H^T - H\beta - a) + \frac{\mu}{2} \|T - H\beta - a\|_2^2 \quad (3.30)$$

Let λ represent the vector of Lagrange multipliers, with μ as the penalty function, in accordance with recommendations. The selection of $\mu = 2N/T$ optimizes the variables (a, β) and λ within the ALM framework by progressively minimizing the Lagrangian function to obtain optimal values (λ_k, μ) for iterative refinement in the ALM process as.

$$(\mathbf{a}_{k+1}, \boldsymbol{\beta}_{k+1}) = \arg \min_{\mathbf{a}, \boldsymbol{\beta}} L_{\mu}(\mathbf{a}, \boldsymbol{\beta}, \lambda_k), \quad \lambda_{k+1} = \lambda_k + \mu(\mathbf{T} - \mathbf{H}\boldsymbol{\beta}_{k+1} - \mathbf{a}_{k+1}) \quad (3.31)$$

The iterative process, predominantly computational in each iteration, focuses on the inversion of $\mathbf{H}^T \mathbf{H} + \frac{2}{C\mu} \mathbf{I}$. This matrix, constant across iterations, can be pre-calculated to enhance computational efficiency. Concurrently, it's noteworthy that different continuity techniques may expedite convergence with fewer iterations, although such equations necessitate more complex computations, even with fast Singular Value Decomposition (SVD) techniques. It is also important to note that while this iterative method may not be the fastest route to achieving high accuracy when training data are devoid of anomalies, it remains the most effective in attaining optimal precision levels. However, when data anomalies are present—which is often the case in real-world applications—it becomes necessary to halt the iterations once a predefined maximum iteration count (MaxIter) is reached for practicality.

3.10 Application of Techniques to Signal Detection

This section will apply deep learning and machine learning techniques discussed from sections 3.4 to 3.8, illustrated in the subsequent figure. The researchers will designate the input data, or Training data, as the received channel, represented by parameter \mathbf{Y} , and the output data, or Teaching data, as the channel between parameters \mathbf{Y} and \mathbf{X} within the communication system discussed in section 3.2. to 3.3 This structure is depicted in Figure 3.1. Figure 3.2 demonstrates the dataset grouping for testing in the Convolutional Neural Network (CNN). The dataset includes both real and imaginary values, necessitating the extraction of imaginary values and their integration into the dataset alongside real values. This arrangement is crucial for ensuring that the dataset reflects the full spectrum of information present in the communication system, with parameters referenced according to the number of antennas defined in the communication system setup. The organization and preprocessing of data in these stages are pivotal for the effective application of

machine learning techniques, particularly CNN, in signal detection. By meticulously preparing the data to include both real and imaginary components and ensuring compatibility with neural network requirements, researchers can optimize the model's learning process and enhance its predictive accuracy. These preparatory steps underscore the detailed and methodical approach required to harness the full potential of machine learning in complex applications such as communication systems analysis, as depicted in Figure 3.4. In the context of the communication system, the channel H is modeled in accordance with the Saleh-Valenzuela theoretical framework, striving to approximate real-world conditions as closely as possible. Through random value generation in line with equation (2.1) and system simulation via MATLAB, the researchers ensured that the number of transmit and receive antennas are equivalently matched.

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,M_T} \\ h_{21} & h_{22} & \cdots & h_{2,M_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_R,1} & h_{M_R,2} & \cdots & h_{M_R,M_T} \end{bmatrix} \quad (3.32)$$

The signal from the transmit side, X , fundamentally influences the signal as it directly pertains to:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M_T} \end{bmatrix} \quad (3.33)$$

For the receive side Y , the process encompasses signal modulation and demodulation, with a focus specifically tailored towards examining signal detection techniques, employing fundamental encoding strategies to concentrate on the detection aspects. Appropriate adaptations, such as optimizing the number of nodes in the hidden layer and varying parameters in each machine learning algorithm, are instrumental in refining the learning process. In particular, Regularized Extreme

Learning Machines (RELM) incorporate certain parameters that regulate variability and diminish the risk of overfitting, resulting in more stable output data and potentially reducing operation times during certain phases. This pivotal parameter, known as the regularization parameter C , is not determined by a definitive rule but is rather contingent upon the empirical outcomes of the processed data, including operation duration and post-learning performance efficiency. Hence, the resulting output is a set of signal detection values processed through the machine learning operation as follows.

Signal detection value using PMS-CNN.

$$X_{PMS} \quad (3.34)$$

Signal detection value using ELM.

$$X_{ELM} \quad (3.35)$$

Signal detection value using RELM.

$$X_{RELM} \quad (3.36)$$

Signal detection value using Optimized RELM (ORELM).

$$X_{ORELM} \quad (3.37)$$

Subsequently, researchers will compute the efficiency of each algorithm to further ascertain the feasibility and effectiveness of the applied methodologies. Implementation of ZF and MMSE Signal Detection Techniques. Following the application of various signal detection techniques, researchers are obliged to demonstrate different types of signal detection methods to serve as benchmarks for

comparing performance across different metrics. Two commonly utilized methods are ZF and MMSE, each defined by their respective equations.

For ZF:

$$X_{ZF} = YH^H (HH^H)^{-1} \quad (3.38)$$

For MMSE:

$$X_{MMSE} = YH^H \left(HH^H + \frac{\sigma_n^2}{\sigma_h^2} I \right)^{-1} \quad (3.39)$$

These techniques, as discussed in Chapter 2, pertain to the theories underlying ZF and MMSE signal detection. In summary, both methods are widely implemented due to their broad applicability and effectiveness in measuring signal detection performance. These methods have been adopted in various research innovations due to their conceptual simplicity and ease of implementation.

3.11 Mean Squared Error (MSE) Performance Comparison Method

Quantify the efficacy of signal detection, multiple metrics exist that articulate the overall performance. Among these, Mean Squared Error (MSE) is a standard and widely utilized metric, particularly in regression analyses. The MSE methodology encapsulates the average of the squares of errors, providing a robust measure of the estimator's precision. The formula for computing MSE is given by:

$$MSE_N = \frac{1}{N} \sum_{i=1}^N (x - \hat{x})^2 \quad (3.40)$$

where x symbolizes the authentic channel signal derived from the simulated communication system, and \hat{x} represents the estimated channel signal deduced

from various methodologies such as ZF, MMSE, ELM, RELM, ORELM, and PMS-CNN. The MSE computation aggregates the square of the error values, systematically disregarding the direction of the discrepancy, thus allowing for an unbiased assessment of deviation magnitude. This measurement is pivotal for determining the accuracy of different signal detection methodologies across a spectrum of Signal-to-Noise Ratios (SNR), thus providing a clear and comparative understanding of their performance relative to the genuine signal propagation characteristics.

3.12 Bit Error Rate (BER) Performance Comparison Method

Upon the completion of precision assessment through Mean Squared Error (MSE), researchers progress to a more nuanced analysis focusing on the integrity and reliability of data transmission within communication systems. This subsequent phase involves a meticulous examination of the Bit Error Rate (BER), a pivotal metric that quantifies the proportion of bits that have been incorrectly identified or altered during the signal detection process, across a variety of deployed methodologies. The BER serves as a paramount indicator of transmission quality, reflecting the effectiveness of signal detection algorithms in preserving the original data content amidst the inherent noise and interference present in communication channels. It measures the frequency at which errors occur in the received signal relative to the total number of bits transmitted during a specified time interval. This metric is instrumental in evaluating the performance of different signal detection techniques, offering a direct comparison of their capability to accurately interpret and reconstruct the transmitted data. A comprehensive analysis of BER involves the simulation or practical implementation of signal transmission scenarios, where data is encoded, transmitted through a simulated or actual communication channel, and then decoded using the signal detection algorithms under review. The decoded data is then compared to the original transmitted data, with discrepancies indicating errors. These errors are tallied and expressed as a ratio to the total transmitted bits to yield the BER. By examining the BER, researchers can identify which signal detection

methods are more resilient to errors induced by channel conditions, such as noise, fading, and interference, thereby gauging the robustness and reliability of these methods in real-world communication scenarios. This detailed exploration, encapsulated in Figure 3.9, underscores the critical role of BER in the comparative assessment of signal detection algorithms, highlighting its utility in determining the most efficient and reliable methods for data transmission within complex communication systems.

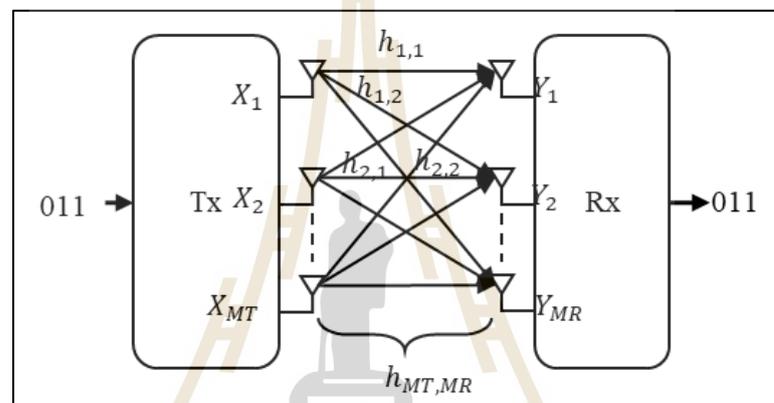


Figure 3.9 Simple Data Encoding and Decoding Process

A method involving the inversion of Equation (3.1) is employed, frequently denominated as Maximum Ratio Combining (MRC) or Maximum Likelihood Estimation (MLE) in scholarly literature. This procedure is encapsulated by the following formulation.

$$X = H^H Y \quad (3.41)$$

This equation represents the mathematical inversion pertaining to the data transmitted. Upon obtaining X , the subsequent step involves calculating the data's error rate. This calculation is facilitated through the utilization of specific functions within the MATLAB programming environment, namely 'qammod' for Quadrature Amplitude Modulation (QAM) encoding and 'qamdemod' for QAM decoding. These

functions play a pivotal role in the encoding and decoding processes, respectively, enabling the precise evaluation of the fidelity of data transmission. The 'qammod' function modulates a sequence of input data into a complex symbol set based on the QAM scheme, effectively preparing the data for transmission through the communication channel. Conversely, the 'qamdmod' function demodulates the received QAM signals, extracting the transmitted data from the complex symbols. By applying these functions, researchers can quantitatively assess the accuracy of the data recovery process, thereby determining the efficacy of the signal detection and data transmission methods under investigation.

3.13 Channel Capacity and Outage Probability Comparison Method

3.13.1 Channel Capacity

Calculating Channel Capacity in systems utilizing Signal Detection is crucial for evaluating the performance of communication systems, particularly in MIMO (Multiple Input Multiple Output) and Massive MIMO systems. The following section will explain the theory related to Channel Capacity used in Signal Detection. The theory of Channel Capacity refers to the maximum capability of a channel to transmit information without errors, considering the limitations of noise and interference. For MIMO systems, calculating Channel Capacity depends on the channel matrix \mathbf{H} , which is an $M_R \times M_T$ matrix, where M_R represents the number of receiving paths and M_T represents the number of transmitting paths. The Channel Capacity C can be calculated using the following equation:

$$C = B \log_2 \det \left(\mathbf{I}_{M_R} + \frac{P}{N_0 M_T} \mathbf{H} \mathbf{H}^H \right) \quad (3.42)$$

Where P is the total transmit power, N_0 is the noise power, \mathbf{H} is the channel matrix, \mathbf{H}^H is the Hermitian transpose of \mathbf{H} , \mathbf{I}_{M_R} is the identity matrix of size $M_R \times M_R$, B is bandwidth (Hz). In Signal Detection, directly obtaining the Channel

Matrix \hat{H} from the estimated H using various techniques is not feasible. This is because H is the result of signal detection that has passed through the communication channel processes and signal filtering, which do not retain all necessary information about H . However, there is a way to estimate H from the received signal Y and the transmitted signal X . This process is called Channel Estimation, and the method of interest to the researchers is Least Squares Estimation to find H for each X detected. Using Least Squares Estimation (LSE) to find H has a significant advantage in its ability to approximate H by minimizing the overall discrepancy between the received signal and the predicted signal. This method seeks to find H that minimizes the sum of squared errors, leading to accurate estimates even in the presence of noise or uncertainties in the signal data. Additionally, Least Squares Estimation involves straightforward computations and can be effectively applied in real systems. The equation is as follows.

$$H = YX^p \quad (3.43)$$

Where X^p is Moore-Penrose pseudo-inverse of X which can be calculated by.

$$X^p = (X^T X)^{-1} X^T \quad (3.44)$$

In cases where X is a non-square matrix or cannot be inverted, pseudo-inverse is used. It will help to estimate the value correctly. From the LSE method, the researcher was able to return to calculate the channel capacity in Equation (3.32) and obtain the following equations.

Channel Capacity for ZF:

$$C_{ZF} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{ZF} H_{ZF}^H \right) \quad (3.45)$$

Channel Capacity for MMSE:

$$C_{MMSE} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{MMSE} H_{MMSE}^H \right) \quad (3.46)$$

Channel Capacity for ELM:

$$C_{ELM} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{ELM} H_{ELM}^H \right) \quad (3.47)$$

Channel Capacity for RELM:

$$C_{RELM} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{RELM} H_{RELM}^H \right) \quad (3.48)$$

Channel Capacity for ORELM:

$$C_{ORELM} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{ORELM} H_{ORELM}^H \right) \quad (3.49)$$

Channel Capacity for CNN-LSTM:

$$C_{CNNLSTM} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{CNNLSTM} H_{CNNLSTM}^H \right) \quad (3.50)$$

Channel Capacity for PMS-CNN:

$$C_{PMS} = B \log_2 \det \left(I_{M_R} + \frac{P}{N_0 M_T} H_{PMS} H_{PMS}^H \right) \quad (3.51)$$

3.13.2 Outage Probability

Outage Probability is the likelihood that the channel capacity will fall below a specified threshold R . It is commonly used as a metric to evaluate the reliability of communication systems in environments with noise and uncertain signal interference. After deriving equations (3.35-3.41), the Outage Probability can be calculated as follows. Outage Probability for ZF:

$$P_{\text{out,ZF}} = \mathbb{P}(C_{\text{ZF}} < R) \quad (3.52)$$

Outage Probability for MMSE:

$$P_{\text{out,MMSE}} = \mathbb{P}(C_{\text{MMSE}} < R) \quad (3.53)$$

Outage Probability for ELM:

$$P_{\text{out,ELM}} = \mathbb{P}(C_{\text{ELM}} < R) \quad (3.54)$$

Outage Probability for RELM:

$$P_{\text{out,RELM}} = \mathbb{P}(C_{\text{RELM}} < R) \quad (3.55)$$

Outage Probability for ORELM:

$$P_{\text{out,ORELM}} = \mathbb{P}(C_{\text{ORELM}} < R) \quad (3.56)$$

Outage Probability for CNN-LSTM:

$$P_{\text{out,CNNLSTM}} = \mathbb{P}(C_{\text{CNNLSTM}} < R) \quad (3.57)$$

Outage Probability for PMS-CNN:

$$P_{\text{out,PMS}} = \mathbb{P}(C_{\text{PMS}} < R) \quad (3.58)$$

3.14 Summary

This section delves into the conceptualization and computational modeling of UM-MIMO communication systems, with a particular focus on signal detection efficacy. It adheres to a theoretical framework of communication system principles, proposing hypotheses regarding the impact of an increased antenna count on the dynamics of signal detection. This exploration seeks to elucidate the nuanced effects both advantageous and detrimental that an augmentation in antenna numbers exerts on signal detection processes. Furthermore, the discussion extends to the ramifications of refining channel resolution, examining the potential shifts in detection outcomes and the intricacies involved. The application of deep learning techniques is introduced as a novel approach, reflecting its burgeoning relevance across a spectrum of research areas. The suitability and adaptability of deep learning methodologies for enhancing signal detection in UM-MIMO systems are critically evaluated. In pursuit of empirical validation, this analysis incorporates the assessment of key performance indicators, including error rates and mean square deviations. Such metrics are instrumental in quantifying the operational efficacy of the proposed models, facilitating a comprehensive understanding of their practical viability and the optimization avenues for signal detection in UM-MIMO communication systems.

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter proposed result of Deep learning performance in UM-MIMO with compare another method such as ZF, MMSE, ELM, RELM, ORLEM, CNN-LSTM and PMS-CNN. This section presents the outcomes of the process undertaken to prepare the PMS-CNN dataset specifically tailored for signal detection tasks. The preparation of this dataset is critical for the effective training of PMS-CNN model, which are designed to recognize and interpret complex patterns within the data. This process involves the collection, cleaning, and augmentation of data to create a comprehensive dataset that reflects a wide array of signal scenarios.

4.2 Evaluation of Algorithm Performance Mean Squared Error (MSE)

This section elucidates the assessment of algorithmic performance by computing the Mean Squared Error (MSE), a critical metric in evaluating the precision of algorithms in various computational tasks. The MSE computation serves as a quantitative measure to gauge the discrepancy between the predicted values generated by the algorithm and the actual values within the dataset. This evaluation process involves the systematic application of the algorithm across a designated dataset and the subsequent calculation of MSE to determine the algorithm's accuracy and efficiency in data analysis and prediction. The computation of MSE is instrumental in identifying the algorithm's strengths and weaknesses, enabling researchers to refine and optimize its performance. By analyzing the MSE values, it becomes possible to discern the algorithm's capability to handle complex datasets and its sensitivity to variations within the data. This assessment not only contributes

to the enhancement of the algorithm's accuracy but also its applicability to a broader range of computational problems. The meticulous examination of MSE values under different conditions and scenarios facilitates a deeper understanding of the algorithm's performance, guiding further improvements and adaptations. In the examination of error metrics, researchers have conducted simulations involving 256 transmit antennas and 256 receive antennas in a Spatial Multiplexing configuration to approximate the communication channels using the following methodologies.

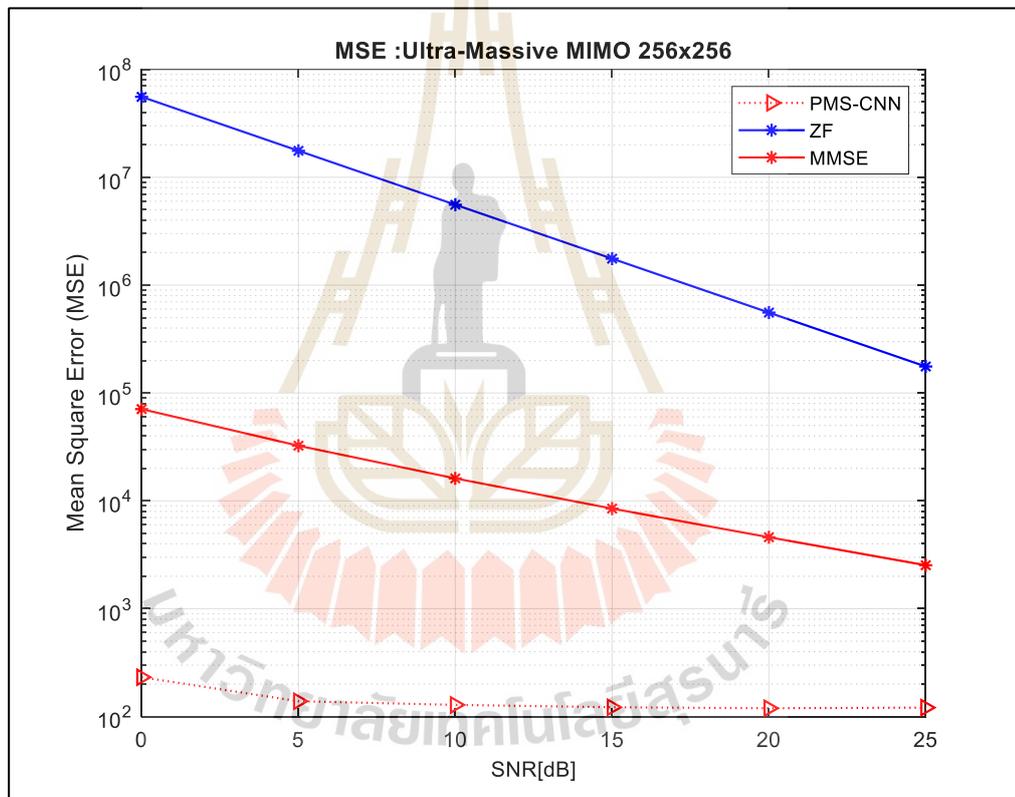


Figure 4.1 The MSE Performance vs SNR [dB] PMS-CNN, ZF, MMSE

In Figure 4.1, the MSE performance of three algorithms—PMS-CNN, ZF, and MMSE in an UM-MIMO system using 256QAM modulation across a range of SNR from 0 dB to 25 dB. The PMS-CNN algorithm shows remarkably low MSE throughout the SNR range, starting just above 10^2 and demonstrating a minimal increase, indicating excellent robustness and accuracy in signal detection. In contrast, the ZF algorithm starts with

a higher MSE around 10^8 and experiences a more gradual reduction, suggesting it is less effective at lower SNRs but improves significantly as SNR increases. The MMSE algorithm demonstrates a consistent and sharp decline in MSE from around 10^7 to below 10^4 , highlighting its effectiveness in reducing error as the signal quality improves. These distinct MSE profiles underline the varied capabilities of these algorithms to handle complex modulation schemes like 256QAM in high-density MIMO environments, crucial for optimizing system performance under varying noise conditions.

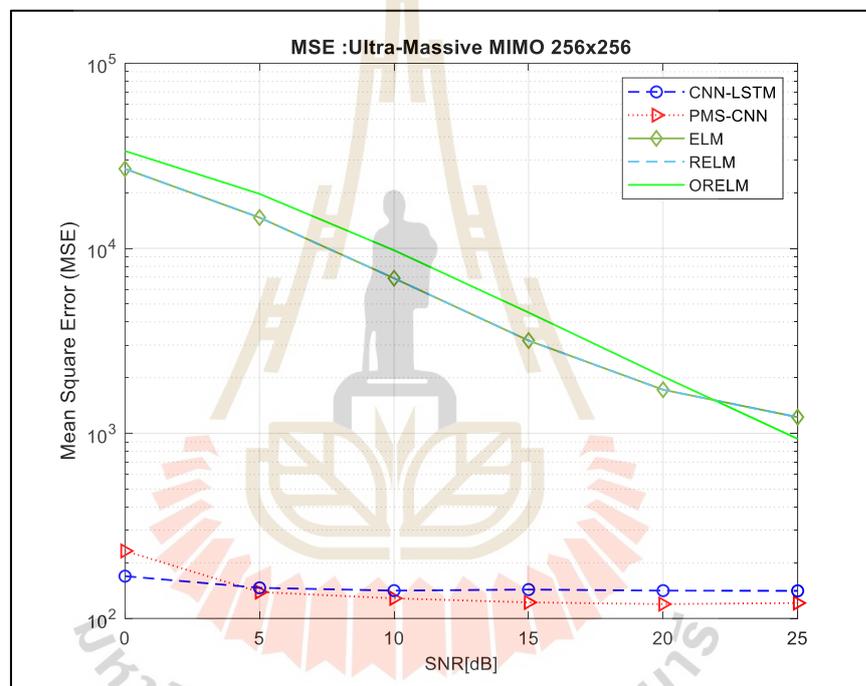


Figure 4.2 The MSE evaluation of Algorithm Performance between CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.

In Figure 4.2, this details the Mean Square Error (MSE) performance of various signal detection algorithms in an UM-MIMO system, measured across SNR levels from 0 dB to 25 dB. The PMS-CNN algorithm, depicted with a dotted red line, consistently shows the lowest MSE, starting around 10^2 and dropping below 10^3 even at low SNR levels, highlighting its exceptional accuracy and robustness against noise. The CNN-LSTM model, indicated by the blue dashed line, also performs well with its MSE

beginning slightly above 10^3 and descending as SNR increases, showing effective noise handling capabilities though not as strong as PMS-CNN. The other algorithms, ELM (solid green line), RELM (dashed green line), and ORELM (solid green line with diamonds), start with higher MSEs around 10^5 , but they show a significant improvement trend, especially beyond 10 dB SNR, dropping to levels around 10^3 to 10^4 , which indicates that while they are less efficient at very low SNR, their performance becomes competitive as signal conditions improve. This detailed depiction of MSE variations helps in evaluating the algorithms' performance in managing errors relative to signal quality, crucial for optimizing detection strategies in high-density MIMO configurations.

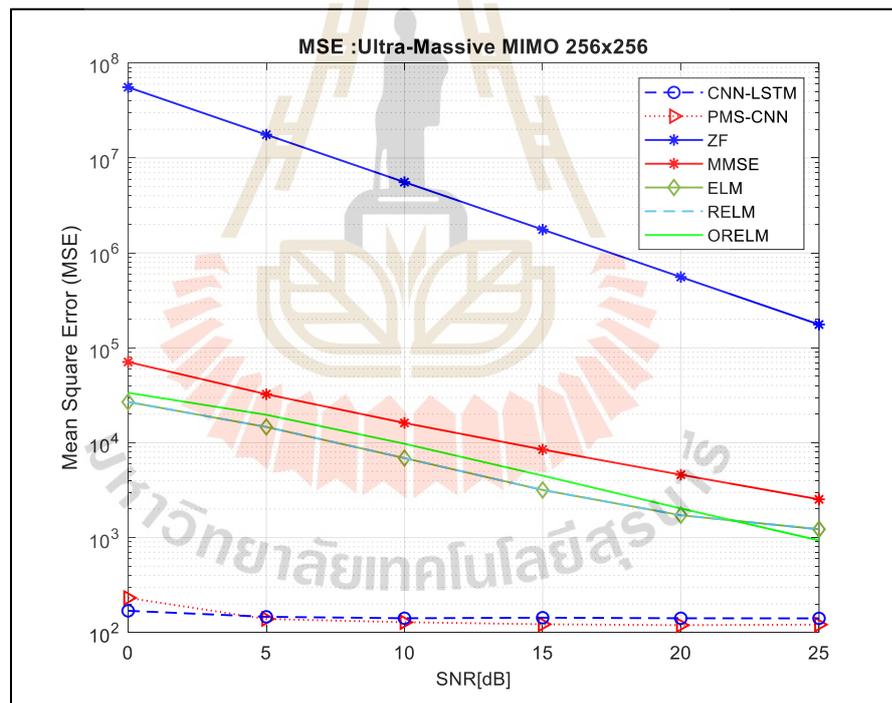


Figure 4.3 The MSE evaluation of Algorithm Performance between ZF, MMSE CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.

In Figure 4.3, This graph effectively delineates the diverse error-handling capabilities of these algorithms, providing crucial insights for selecting the appropriate signal detection method based on specific error tolerance and SNR conditions. This graph

shows a detailed comparison of the MSE performance across various signal detection algorithms in an UM-MIMO system employing 256QAM modulation, with the SNR spanning from 0 dB to 25 dB. The PMS-CNN algorithm, represented by the dotted red line, stands out with the lowest MSE throughout the SNR range, beginning just above 10^2 and demonstrating minimal error as SNR increases, which underscores its exceptional effectiveness in a high-interference environment. The CNN-LSTM model, depicted by the blue dashed line, also showcases commendable error reduction capabilities, with MSE values starting around 10^3 and consistently improving, indicating strong adaptability to increasing SNR levels. The ZF algorithm, shown with the solid red line, has a relatively high initial MSE around 10^7 but exhibits a steady decrease in error, becoming more competitive as SNR approaches 25 dB. The MMSE algorithm, marked by the green line with diamonds, starts with a slightly better position than ZF near 10^6 and reduces errors more effectively, especially at higher SNR levels, reflecting its robust design for noise mitigation. The ELM, RELM, and ORELM algorithms are illustrated with various green lines and shapes, starting with MSEs in the range of 10^5 to 10^6 and showing gradual improvements. However, they maintain higher error rates compared to the more sophisticated PMS-CNN and CNN-LSTM models, indicating a lower performance in handling the complex error landscapes of 256QAM modulation in UM-MIMO systems.

4.3 Bit Error Rate Performance (BER)

In this section, BER is an important metric used to evaluate the performance of communication systems in delivering accurate data under various conditions. This comparison was performed over a signal-to-noise ratio (SNR) range from 0 dB to 30 dB, providing clear insight into each algorithm's ability to reduce signal processing errors. The researchers present these results with the aim of highlighting how these algorithms perform under increased noise levels. This is a common challenge in high-density MIMO environments. This helps in understanding the robustness and effectiveness in maintaining communication integrity in real-world situations.

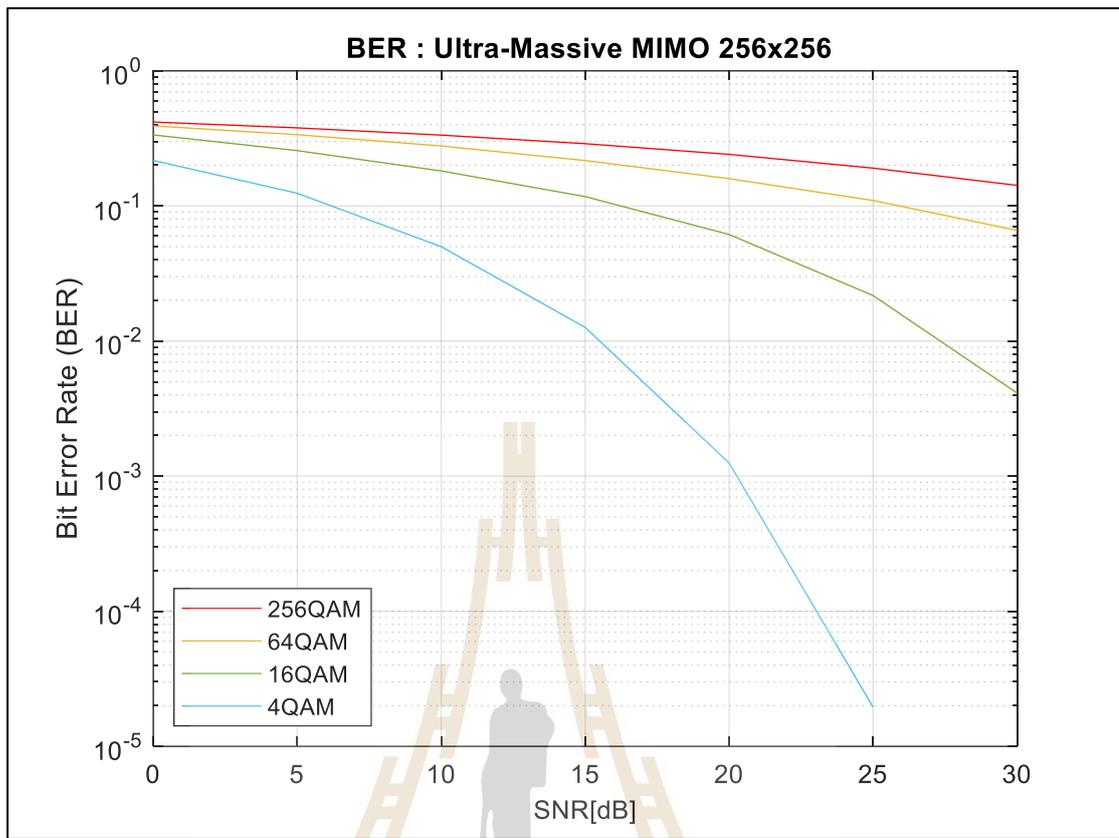


Figure 4.4 The BER of different Modulation in Ultra Massive MIMO 256x256

In Figure 4.4, this graph depicts the Bit Error Rate (BER) performance across different modulation schemes within an UM-MIMO system employing a 256x256 antenna array. Each line represents a modulation scheme varying from 4QAM to 256QAM, analyzed over a range of SNR from 0 dB to 30 dB.

In communications theory, the BER is a critical measure of the number of bit errors per unit time or per number of total bits transmitted. The BER gives us insight into the effectiveness of the communication link under varying conditions and modulation schemes. In this graph, each modulation scheme's susceptibility to noise and interference as SNR increases is clearly visible. Higher-order modulations like 256QAM and 64QAM offer higher data rates because they pack more bits into each symbol. However, they also show higher BER at lower SNR values compared to schemes like 4QAM, which is more robust but offers lower data throughput. UM-MIMO systems, which capitalize on spatial multiplexing to significantly increase

channel capacity, choosing the right modulation scheme is crucial. These systems can benefit from higher-order modulation schemes like 256QAM at high SNR levels, where the BER becomes acceptably low, maximizing data throughput without sacrificing reliability. At lower SNR levels, the graph demonstrates how higher-order modulations become less feasible as their BER rises sharply, indicating a greater likelihood of error in detected signals. This analysis is essential for optimizing the modulation scheme in UM-MIMO systems based on the operational environment's SNR conditions, ensuring that the system can achieve the highest possible throughput while maintaining data integrity and minimizing errors. The use of 256QAM in this setup, as shown, would be most advantageous in high SNR scenarios where the system's advanced signal processing capabilities can effectively mitigate the inherent noise and error risks associated with such high-density signal environments.

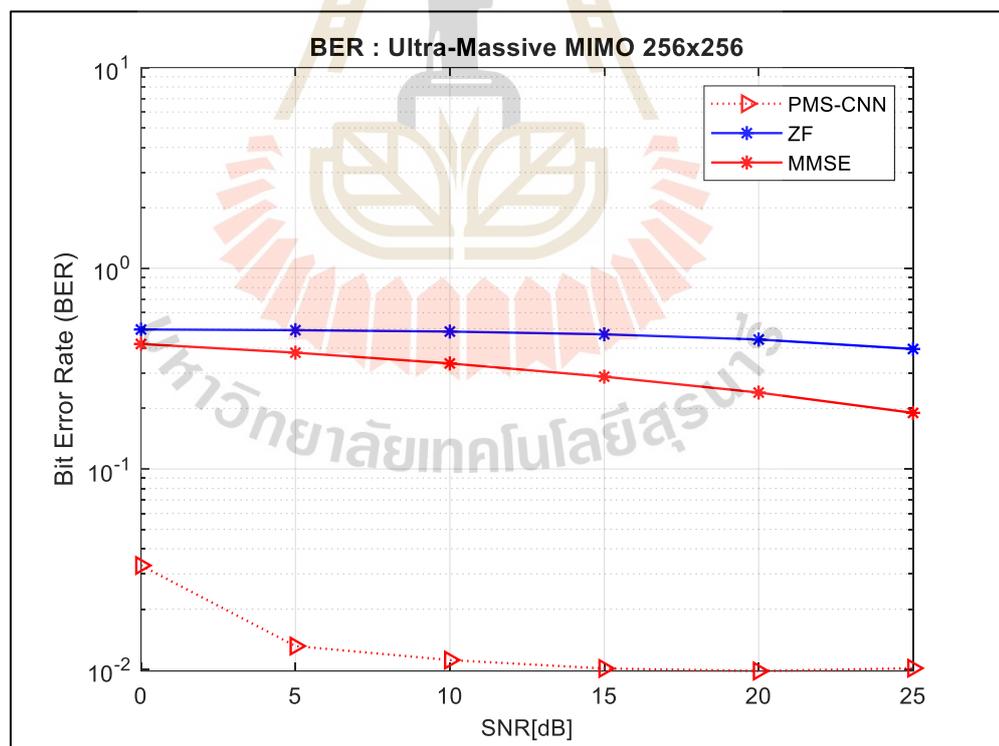


Figure 4.5 The BER of Algorithm Performance between ZF, MMSE and PMS-CNN method.

In Figure 4.5, the BER performance of three signal detection algorithms PMS-CNN, ZF, and MMSE in an UM-MIMO system using 256QAM modulation across varying SNR. The PMS-CNN algorithm shows excellent improvement in BER as the SNR increases, indicating its superior error correction abilities, particularly effective beyond 15 dB. In comparison, the ZF method maintains a relatively constant BER around 10^{-1} across all SNR levels, showing limited improvement with increased SNR, which suggests it may struggle with the high noise levels typical with 256QAM. Meanwhile, the MMSE algorithm starts with higher error rates at lower SNR but improves significantly to approach PMS-CNN performance at higher SNR, demonstrating its effectiveness in reducing errors as the signal quality improves. This visualization underscores the capabilities of these algorithms in managing errors under different noise conditions, especially in a complex modulation scheme like 256QAM, highlighting the robust performance of PMS-CNN in scenarios demanding high data integrity.

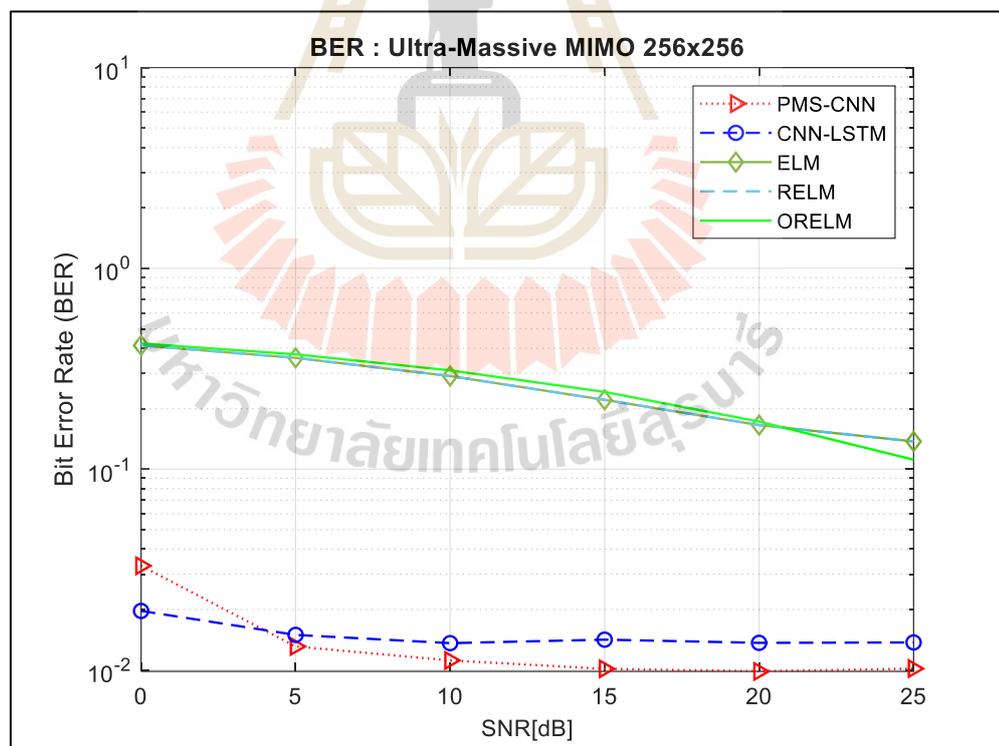


Figure 4.6 The BER of Algorithm Performance CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.

In Figure 4.6, show the BER performance of various algorithms in an UM-MIMO system using 256QAM modulation across SNR from 0 to 25 dB. The PMS-CNN algorithm, indicated by the dotted red line, demonstrates exceptional performance with a BER starting below 10^{-2} and further improving, indicating highly effective error correction. The CNN-LSTM, shown with the blue dashed line, also achieves low BER, maintaining steady improvement across the SNR range. Meanwhile, the ELM, RELM, and ORELM algorithms, represented by green lines, exhibit higher initial BERs but show significant improvement as SNR increases. This visualization highlights the robustness of these algorithms in handling complex modulation schemes and their efficiency in reducing errors in a high-density MIMO setting, making them vital for ensuring reliable signal detection.

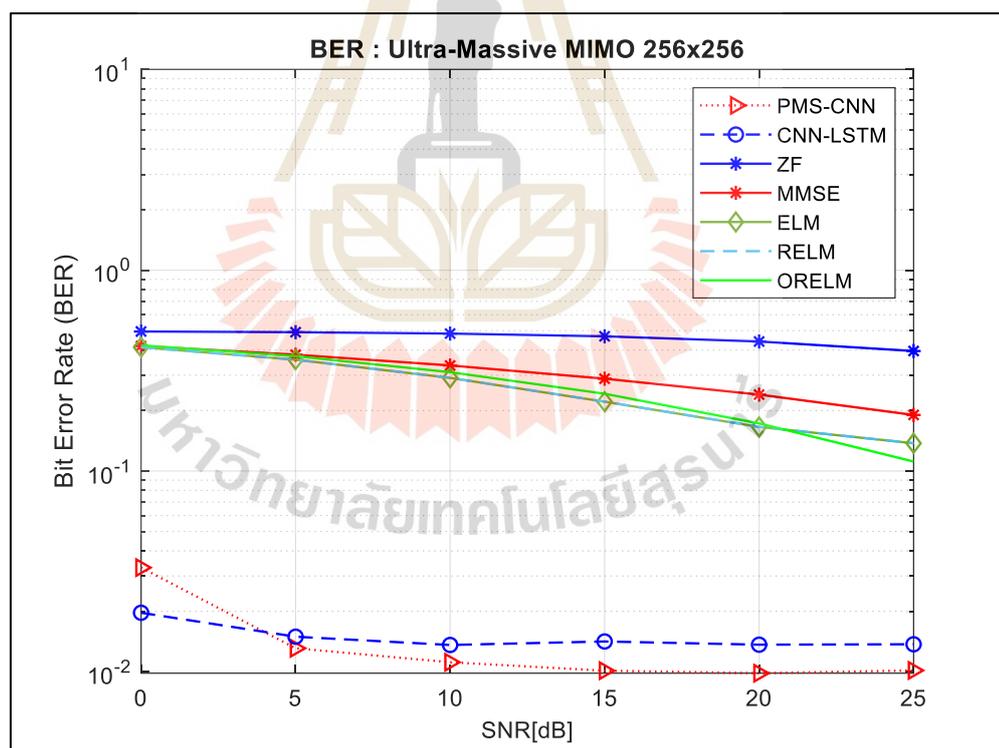


Figure 4.7 The BER of Algorithm Performance ZF, MMSE, CNN-LSTM, ELM, RELM, ORELM and PMS-CNN method.

In Figure 4.7, the BER performance of various algorithms in an UM-MIMO system using 256QAM modulation, across a range of SNR from 0 dB to 25 dB. The PMS-CNN algorithm, indicated by the dotted red line, shows the best performance with a significantly lower BER, particularly noticeable as the SNR increases, demonstrating its high efficiency in error correction. CNN-LSTM, shown by the blue dashed line, also maintains a relatively low and stable BER across all SNR levels, suggesting its robustness in complex signal environments. In contrast, ZF and MMSE, marked by the green and red solid lines respectively, exhibit higher BERs but show gradual improvement with increased SNR. The ELM, RELM, and ORELM algorithms, represented by the green lines with different markers, perform comparably, each reducing BER as SNR increases, though not as effectively as PMS-CNN or CNN-LSTM. This visualization highlights the effectiveness of these algorithms in managing signal detection errors in a high-density MIMO configuration under the demanding conditions of 256QAM modulation.

4.4 Channel Capacity

Channel capacity is one of the most important characteristics in designing and evaluating the performance of a communication system. Especially in the system Ultra-Massive MIMO, which uses many antennas to increase data transfer capacity and signal quality. Theoretically, Channel Capacity represents the maximum capacity to transmit data over a communication channel without errors. Under specified noise conditions or SNR. As shown in the following.

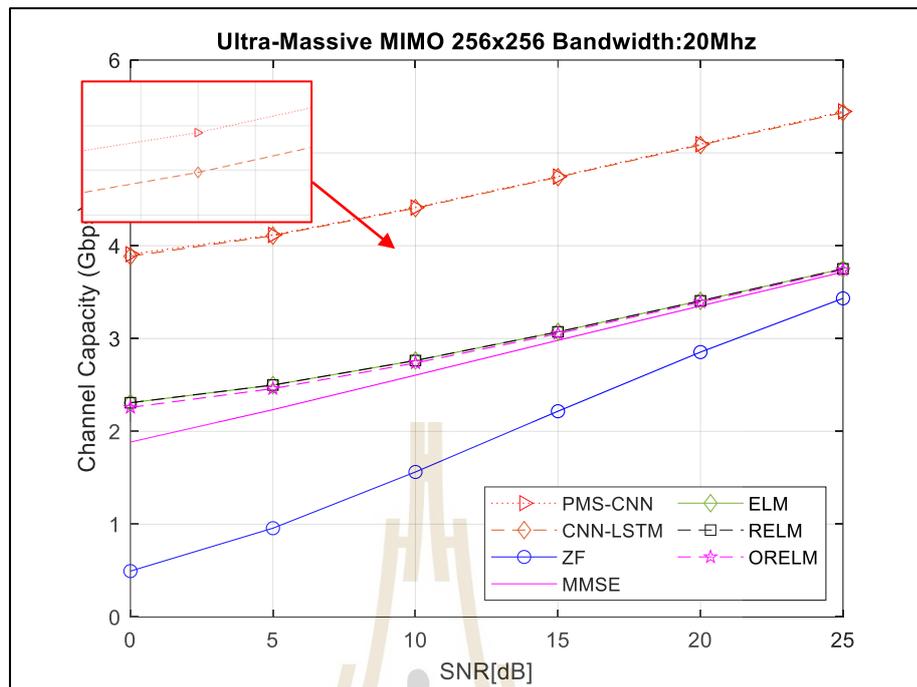


Figure 4.8 The Channel Capacity (Mbps) vs SNR [dB]

In Figure 4.8, UM-MIMO 256x256 Bandwidth: 20MHz, it shows a comparison of channel capacity between different signal detection algorithms within a UM-MIMO system using 256QAM modulation, where both the transmitter and receiver have 256 antennas. The comparison is conducted over a SNR range from 0 dB to 25 dB. The PMS-CNN algorithm, represented by the red dotted line with an open triangle symbol, continuously achieves the highest channel capacity throughout the tested SNR range. It starts at approximately 4 Gbps at 0 dB SNR and increases to about 5.5 Gbps at 25 dB SNR, demonstrating the best performance in signal processing and spatial multiplexing capabilities in the Ultra-Massive MIMO system. Meanwhile, the CNN-LSTM algorithm, depicted by the red dotted line with a diamond symbol, shows nearly identical performance to PMS-CNN across all SNR levels, highlighting its ability to leverage complex deep learning models for efficient signal interpretation. The MMSE algorithm (purple line) and ZF algorithm (blue line) show lower performance, with capacities of about 2.5 Gbps and 1.5 Gbps respectively at 25 dB SNR, indicating lower capabilities in handling noise and interpreting signals compared to PMS-CNN

and CNN-LSTM. The ELM, RELM, and ORELM algorithms (green, pink, and black lines respectively) show grouped results with channel capacities increasing up to around 3 Gbps at 25 dB SNR, which is lower compared to the PMS-CNN and CNN-LSTM algorithms.

4.5 Outage Probability

The following graph provides a detailed visualization of the outage probability against channel capacity for various signal detection algorithms utilized within an UM-MIMO system. Outage probability is a crucial metric in telecommunications that indicates the likelihood of a system's capacity falling below a required threshold, impacting the quality and reliability of the communication service. Each curve represents a different algorithm's performance in maintaining sufficient channel capacity under varying conditions, showcasing their respective abilities to handle high capacity demands typical of advanced MIMO systems. This comparison is pivotal for assessing the robustness and efficiency of each method, particularly in scenarios where maintaining a high data transmission rate is critical for system functionality.

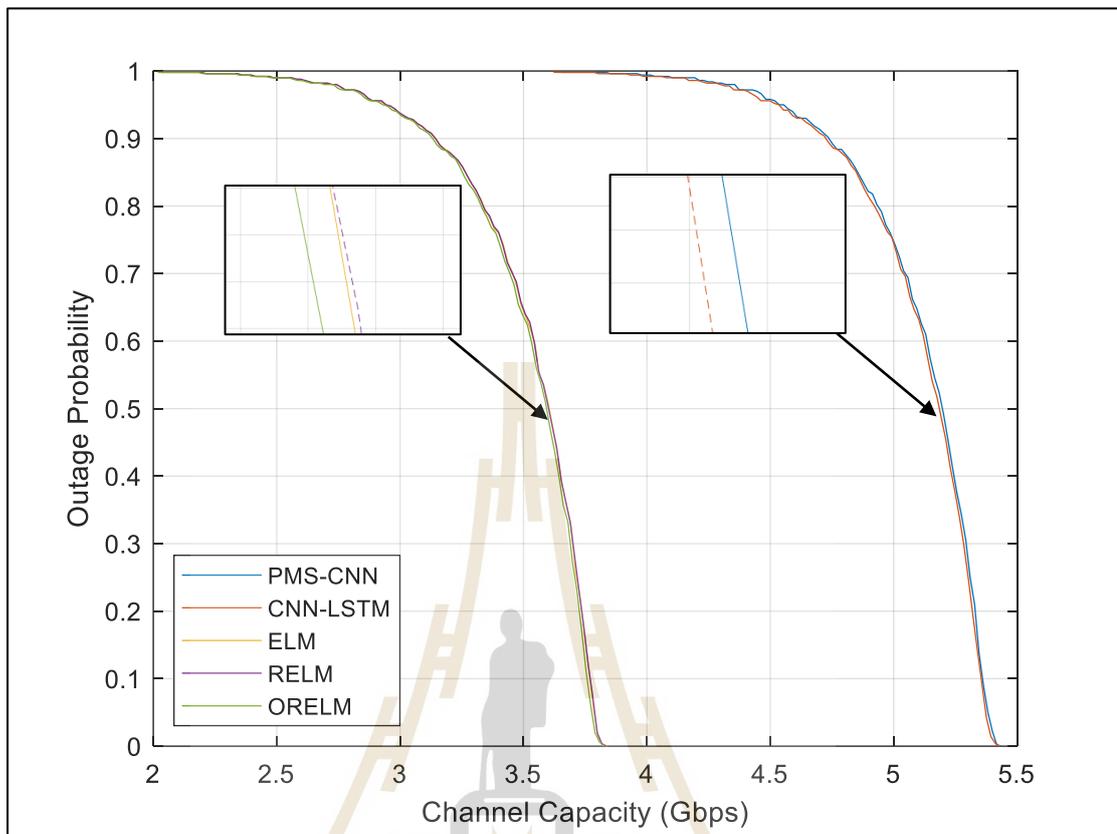


Figure 4.9 Outage Probability vs Channel Capacity (bps/Hz)

In Figure 4.9, shows the outage probability versus capacity for different signal detection algorithms in the UM-MIMO system, including advanced machine learning techniques evaluated using 256QAM adjustments. The theoretical foundations are likely based on equations similar to equations (3.52) to (3.58), which describe the analytical relationship between channel capacity and outage probability. From the results, PMS-CNN (blue line) shows nearly zero outage probability at approximately 5.5 Gbps, indicating exceptionally high reliability and optimal channel capacity maintenance. CNN-LSTM (red line) exhibits an outage probability nearly equivalent to PMS-CNN, outperforming other techniques and maintaining better channel capacity. In contrast, ELM (orange line), RELM (brown line), and ORELM (green line) show higher outage probabilities, with the probability decreasing more slowly as capacity increases. At zero outage probability, these algorithms achieve a capacity of around

3.8 Gbps. In summary, PMS-CNN and CNN-LSTM offer superior reliability and channel capacity maintenance compared to other machine learning methods.

4.6 Computational Complexity with Big O notation

In this section, the researcher shows the results of calculating the complexity of Algorithms using the Big O Notation method are widely used in algorithm verification and comparison work. The researchers performed calculations with all algorithms used in this work, including ZF, MMSE, RELM, ORELM, CNN-LSTM, and including the PMS-CNN method. For processing ZF and MMSE methods, the basic assumption used for complexity analysis is that the calculation uses matrix access and matrix inversion in different ways. From the foregoing, consider from Table 2.4, Table 4.1 to Table 4.7 as follows.

Table 4.1 Big O Complexity Analysis for ZF

Step	Equation	Complexity	Methodology
Matrix Inversion	$H^{-1} = (H^H H)^{-1}$	$O(M^3)$	Calculate the Hermitian matrix H^H (size $N \times M$), then compute $H^H H$ (size $M \times M$), and finally the inverse $(H^H H)^{-1}$.
Matrix Multiplication	$\hat{s} = H^{-1}y$	$O(M^2)$	Multiply the inverse matrix (size $M \times M$) with the signal vector y (size $M \times 1$).
Overall Complexity		$O(M^3 + M^2)$	Combine the complexities: $O(M^3)$ for inversion and $O(M^2)$ for multiplication.

Table 4.2 Big O Complexity Analysis for MMSE

Step	Equation	Complexity	Methodology
Matrix Inversion	$(\mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I})^{-1}$	$O(M^3)$	Calculate the Hermitian matrix \mathbf{H}^H , then compute $(\mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I})$ and finally the inverse.
Matrix Multiplication	$\hat{\mathbf{s}} = (\mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^H \mathbf{y}$	$O(M^2)$	Multiply the inverse matrix (size $M \times M$) with the Hermitian \mathbf{H}^H and the signal vector \mathbf{y} .
Overall Complexity		$O(M^3 + M^2)$	Combine the complexities: $O(M^3)$ for inversion and $O(M^2)$ for multiplication.

Table 4.3 Big O Complexity Analysis for ELM

Step	Equation	Complexity	Methodology
Matrix Inversion	\mathbf{H}^\dagger	$O(M^3)$	Calculate the inverse of matrix \mathbf{H} (size $M \times M$).
Matrix Multiplication	$\beta = \mathbf{H}^\dagger \mathbf{T}$	$O(M^2)$	Multiply the inverse matrix (size $M \times M$) with vector \mathbf{T} (size $M \times 1$).
Activation Function	$c_i \cdot X_j + V_i$	$O(N)$	Multiply vector X_j (size $N \times 1$). With constant c_i and add V_i .
Activation Function	$\sigma(c_i \cdot X_j + V_i)$	$O(1)$	Apply activation function with complexity $O(1)$.
Scalar Multiplication	$\beta_i \sigma(c_i \cdot X_j + V_i)$	$O(1)$	Multiply σ by β_i with complexity $O(1)$.
Summation	$\sum_{i=1}^{\tilde{N}} \beta_i \sigma(c_i \cdot X_j + V_i)$	$O(\tilde{N} \cdot N)$	Sum \tilde{N} terms, each with complexity $O(N)$.
Overall Complexity		$O(M^3 + M^2 + \tilde{N} \cdot N)$	Combine the complexities of all steps.

Table 4.4 Big O Complexity Analysis for RELM

Step	Equation	Complexity	Methodology
Matrix Inversion	$(H^T H + \lambda I)^{-1}$	$O(M^3)$	Calculate the pseudo-inverse of matrix H (size $M \times M$).
Matrix Multiplication	$(H^T H + \lambda I)^{-1} H^T$	$O(M^2 N)$	Multiply the inverse matrix (size $M \times M$) with vector H^T (size $M \times N$).
Matrix Multiplication	$(H^T H + \lambda I)^{-1} H^T T$	$O(MN)$	Multiply the result (size $M \times N$) with vector T (size $M \times 1$).
Activation Function	$c_i \cdot X_j + V_i$	$O(N)$	Multiply vector X_j (size $N \times 1$). With constant c_i and add V_i .
Activation Function	$\sigma(c_i \cdot X_j + V_i)$	$O(1)$	Apply activation function like sigmoid, ReLU with complexity $O(1)$.
Scalar Multiplication	$\beta_i \sigma(c_i \cdot X_j + V_i)$	$O(1)$	Multiply σ by β_i with complexity $O(1)$.
Summation	$\sum_{i=1}^{\tilde{N}} \beta_i \sigma(c_i \cdot X_j + V_i)$	$O(\tilde{N} \cdot N)$	Sum \tilde{N} terms, each with complexity $O(N)$.
Overall Complexity		$O(M^3 + M^2 N + MN + \tilde{N} \cdot N)$	Combine the complexities of all steps.

Table 4.5 Big O Complexity Analysis for ORELM

Step	Equation	Complexity	Methodology
Matrix Inversion	$(H^T H + \frac{2\mu}{C} I)^{-1}$	$O(M^3)$	Calculate the pseudo-inverse of matrix H (size $M \times M$).

Table 4.5 Big O Complexity Analysis for ORELM (Continued)

Matrix Multiplication	$(H^T H + \frac{2\mu}{C} I)^{-1} H^T$	$O(M^2 N)$	Multiply the inverse matrix (size $M \times M$) with vector H^T (size $M \times N$).
Matrix Multiplication	$(H^T H + \frac{2\mu}{C} I)^{-1} H^T (y - e_k + \frac{\lambda_k}{\mu})$	$O(MN)$	Multiply the result (size $M \times N$) with vector $(y - e_k + \lambda_k / \mu)$ (size $M \times 1$).
Compute x using $Proj_M$	$x = Proj_M(b - e + \frac{\lambda}{\mu})$	$O(N)$	Based on the $Proj_M$ unction, which can be approximated as $O(N)$.
Compute e using $shrink$	$e = shrink(b + \frac{\lambda}{\mu} - A \cdot x, \mu^{-1})$	$O(N)$	The $shrink$ function is linear in complexity, and matrix multiplication $A \cdot x$ has complexity $O(N)$.
Update λ	$\lambda = \lambda + \mu \cdot (b - A \cdot x - e)$	$O(N)$	Updating λ has complexity $O(N)$.
Activation Function	$c_i \cdot X_j + V_i$	$O(N)$	Multiply vector X_j (size $N \times 1$). With constant c_i and add V_i .
Activation Function	$\sigma(c_i \cdot X_j + V_i)$	$O(1)$	Apply activation function like sigmoid, ReLU with complexity $O(1)$.
Scalar Multiplication	$\beta_i \sigma(c_i \cdot X_j + V_i)$	$O(1)$	Multiply σ by β_i with complexity $O(1)$.
Summation	$\sum_{i=1}^{\tilde{N}} \beta_i \sigma(c_i \cdot X_j + V_i)$	$O(\tilde{N} \cdot N)$	Sum \tilde{N} terms, each with complexity $O(N)$.
Overall Complexity		$O(M^3 + M^2 N + MN + \tilde{N} \cdot N + N)$	Combine the complexities of all steps.

M is Number of receive antennas (size of the matrix H).

N is Number of transmit antennas (size of the vector y).

\tilde{N} is Number of conditions used in the equation.

The calculation of Big O Notation in this case involves a structural analysis of the CNN-LSTM and PMS-CNN model used for comparison in this thesis. This analysis will examine the operations of each step and the processes in the code in detail to obtain the mathematical complexity that represents the number of operations as the data size increases. The summary is shown in the following Table 4.6.

Table 4.6 Big O Complexity Analysis for CNN-LSTM

Step	Matrix/Vector Operation	Complexity (Big O)
Conv1d Layer	$x \in \mathbb{R}^{n \times c_1} \cdot W \in \mathbb{R}^{k \times m_1}$	$O(n \cdot c_1 \cdot k \cdot m_1)$
BatchNorm1d Layer	Normalization of $n \cdot m_1$	$O(n \cdot m_1)$
Conv1d Layer	$x \in \mathbb{R}^{n \times m_1} \cdot W \in \mathbb{R}^{k \times c_2}$	$O(n \cdot m_1 \cdot k \cdot c_2)$
BatchNorm1d Layer	Normalization of $n \cdot c_2$	$O(n \cdot c_2)$
MaxPool1d Layer	$\max(x)$	$O(n)$
Repeat ConvBlock	2 more times	$3 \times (\text{ConvBlock})$
Linear Layer	$x \in \mathbb{R}^{d_1} \cdot W \in \mathbb{R}^{d_1 \times d_2}$	$O(n \cdot d_1 \cdot d_2)$
LSTM Layer	$h_i = \sigma(W \cdot [h_{i-1}, x_i] + b)$	$O(n \cdot h^2)$
Linear Layer	$x \in \mathbb{R}^{f_1} \cdot W \in \mathbb{R}^{f_1 \times f_2}$	$O(f_1 \cdot f_2)$
MaxPool1d Layer	$\max(x)$	$O(n)$
Linear Layer	$x \in \mathbb{R}^{f_2} \cdot W \in \mathbb{R}^{f_2 \times n_classes}$	$O(f_2 \cdot n_classes)$
Overall Complexity		$O(n \cdot (c_1 \cdot k \cdot m_1 + m_1 \cdot k \cdot c_2 + d_1 \cdot d_2 + h^2 + f_1 \cdot f_2 + f_2 \cdot n_classes))$

In this Table 4.6, where n is size of the input sequence. c_1 Number of input channels for the first Conv1d layer. k is kernel size. m_1 is number of output channels (filters) for the first Conv1d layer. c_2 is number of input channels for the second Conv1d

layer. d_1 is number of features before the first linear layer. d_2 is number of features after the first linear layer. h is hidden size for the LSTM layer. f_1 is number of features after the first linear layer in the classifier. f_2 is number of features after the second linear layer in the classifier. $n_classes$ is number of output classes.

Table 4.7 Big O Complexity Analysis for PMS-CNN

Step	Matrix/Vector Operation	Complexity (Big O)
Conv1d Layer 1	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{1 \times c}$	$O(n \cdot c)$
Conv1d Layer 2	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{3 \times c}$	$O(n \cdot c \cdot k_2)$
Conv1d Layer 3	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{5 \times c}$	$O(n \cdot c \cdot k_3)$
Conv1d Layer 4	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{7 \times c}$	$O(n \cdot c \cdot k_4)$
Conv1d Layer 5	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{9 \times c}$	$O(n \cdot c \cdot k_5)$
Conv1d Layer 6	$x \in \mathbb{R}^{n \times c} \cdot W \in \mathbb{R}^{12 \times c}$	$O(n \cdot c \cdot k_6)$
Channel Pooling	$x \in \mathbb{R}^{n \times p} \cdot W \in \mathbb{R}^{k \times p}$	$O(n \cdot c \cdot k \cdot p)$
Flattening	$x \in \mathbb{R}^{n \times p}$	$O(n \cdot p)$
Linear Layer 1	$x \in \mathbb{R}^d \cdot W \in \mathbb{R}^{d \times f_1}$	$O(d \cdot f_1)$
Linear Layer 2	$x \in \mathbb{R}^{f_1} \cdot W \in \mathbb{R}^{f_1 \times n_classes}$	$O(f_1 \cdot n_classes)$
Overall Complexity		$O(n \cdot (c + c \cdot (k_2 + k_3 + k_4 + k_5 + k_6 + k \cdot p) + p + d \cdot f_1 + f_1 \cdot n_classes))$

In this Table 4.7, where n is size of the input sequence. c Number of input channels for the first Conv1d layer. k is kernel size for Channel Pooling layer. k_2, k_3, k_4, k_5, k_6 are kernel sizes for Conv1d layers (3, 5, 7, 9, 12). p is number of pooling channels d is number of features before the first linear layer. f_1 is number of features after the first linear layer in the classifier. $n_classes$ is number of output classes. These detailed accounts ensure a robust understanding of the computational demands and theoretical underpinnings of each algorithm, complete with supporting equations that clarify the source of each complexity component. From the results obtained, the researcher will summarize and put in a Table. 4.8.

Table 4.8 The Computational Complexity by Big O notation. By M , N , \tilde{N} , $n = 10$, $c = 32$, $k = 5$, $k_2, k_3, k_4, k_5, k_6 = 3, 5, 7, 9, 12$, $p = 32$, $d = 509$, $f_1 = 2024$, and $n_classes = 10$.

Algorithm	Computational Complexity Formula	The Overall Complexity
ZF	$O(M^3 + M^2)$	$O(1.01 \times 10^3)$
MMSE	$O(M^3 + M^2)$	$O(1.01 \times 10^3)$
ELM	$O(M^3 + M^2 + \tilde{N} \cdot N)$	$O(1.2 \times 10^3)$
RELM	$O(M^3 + M^2 N + MN + \tilde{N} \cdot N)$	$O(2.2 \times 10^3)$
ORELM	$O(M^3 + M^2 N + MN + \tilde{N} \cdot N + N)$	$O(3.2 \times 10^3)$
CNN-LSTM	$O(n \cdot (c_1 \cdot k \cdot m_1 + m_1 \cdot k \cdot c_2 + d_1 \cdot d_2 + h^2 + f_1 \cdot f_2 + f_2 \cdot n_classes))$	$O(4.614336 \times 10^7)$
PMS-CNN	$O(n \cdot (c + c \cdot (k_2 + k_3 + k_4 + k_5 + k_6 + k \cdot p) + p + d \cdot f_1 + f_1 \cdot n_classes))$	$O(4.6885 \times 10^5)$

According to Table 4.8, we have details for calculating according to the specified values for calculations in this table. Each value is represented by a defined parameter. It helps to understand the temporal complexity in the computational model that meets the conditions and data volume being simulated. Using this table can help clearly differentiate the different models, and better compare complexity between different models.

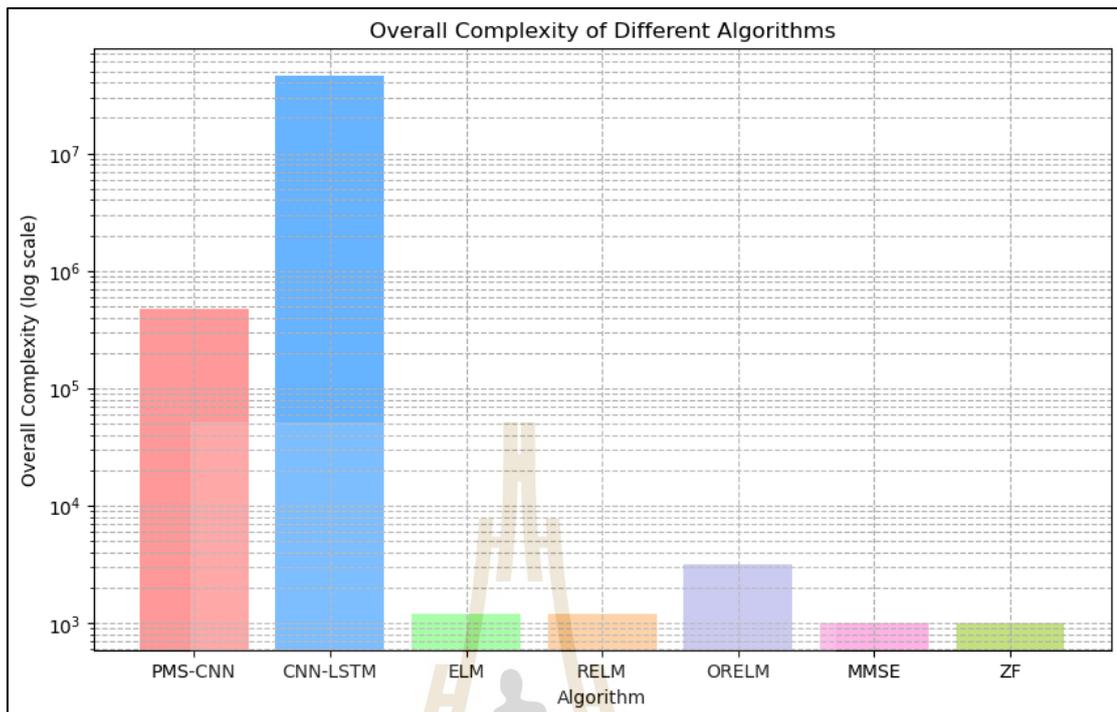


Figure 4.10 Big O Complexity Chart CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.

In Figure 4.10, the chart shows the computational complexity of various algorithms used in signal detection for UM-MIMO systems, using a logarithmic scale on the y-axis to represent complexity levels from 10^2 to 10^8 . The proposed PMS-CNN method has a moderate computational complexity, shown by the blue bar at around 10^5 , indicating it is efficient and suitable for applications where a balance between performance and resource consumption is crucial. The CNN-LSTM model, represented by the tall red bar, has the highest complexity, close to 10^7 , indicating it requires significant computational resources. This might be justifiable by potentially better detection performance but could be impractical for resource-constrained environments. Other methods like ZF, MMSE, ELM, RELM, and ORELM exhibit lower complexities at 10^3 , making them viable alternatives that balance between complexity and performance. These are shown with purple, orange, cyan, green, and pink bars, respectively. This visualization helps in comparing these methods in terms

of resource efficiency, which is crucial in selecting algorithms for practical deployments in systems handling UM-MIMO operations.

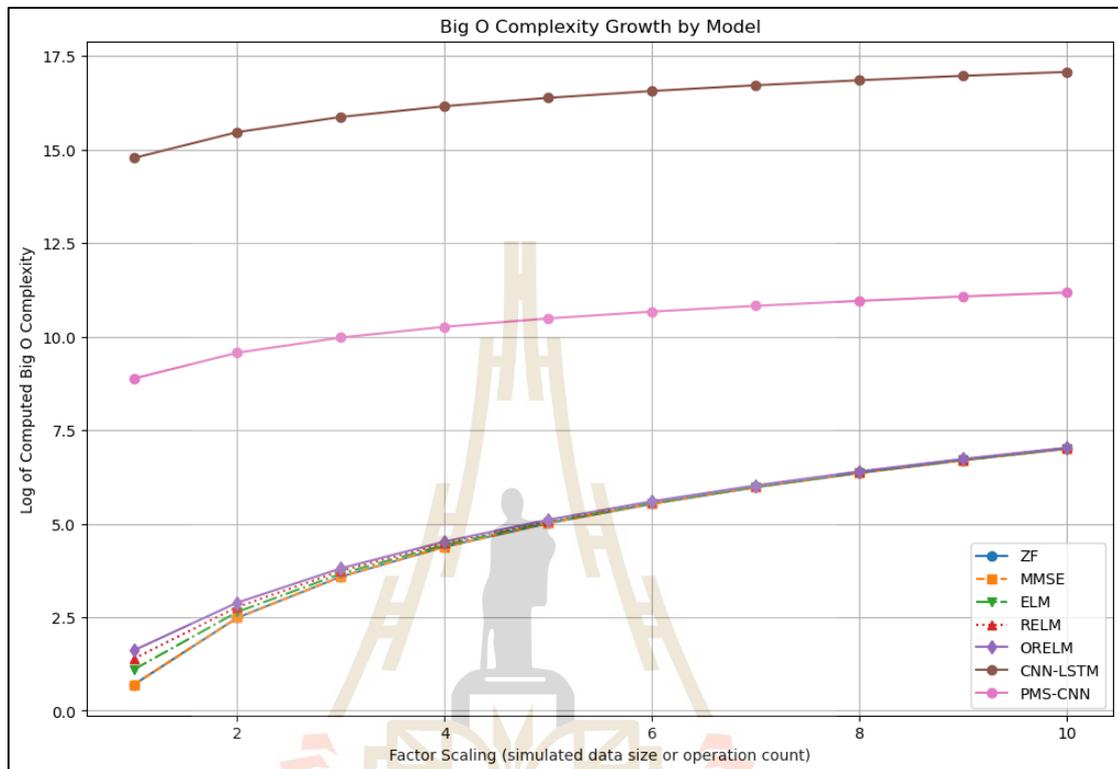


Figure 4.11 Big O Complexity Growth by Model CNN-LSTM, ELM, RELM ORELM and PMS-CNN method.

In Figure 4.11, the growth of Big O complexity of various computational models used to detect signals for the UM-MIMO system is presented. The logarithmically scaled plot shows the CNN-LSTM model with a significant increase in complexity, starting at 10^{15} and increasing to approximately $10^{17.5}$ at the highest scaling factor. This indicates the need for substantial resources. In contrast, the proposed PMS-CNN model shows effective scalability, starting at $10^{10.5}$ and increasing only moderately to $10^{12.5}$, emphasizing its suitability for resource-constrained situations. Other models, including ZF, MMSE, ELM, RELM, and ORELM, start with a complexity of approximately $10^{2.5}$ to $10^{2.7}$ and show a slight increase to approximately $10^{4.5}$ to $10^{5.0}$, making them usable in applications that require lower computational costs.

These results highlight the trade-off between computational efficiency and resource efficiency, which is crucial for selecting the appropriate model in UM-MIMO systems depending on specific application requirements and resource limitations.

In section 2.10, which discusses the application of deep learning in real-world scenarios, to evaluate the model's performance with latency in 6G systems that support usage in the range of 0.1-1 ms, the calculation involves transforming the overall Big O notation results from Table 4.8 and using the results to compute based on the following equation

$$\text{Time (ms)} = \frac{\text{Overall Complexity}}{\text{FLOPS}} \times 1000 \quad (4.1)$$

This FLOPS stands for Floating Point Operations Per Second and is a measure of computer performance, particularly in fields of scientific calculations that require floating-point calculations. By using this equation, one can estimate the latency or time required for a model to process data, which is crucial for evaluating the performance of models in 6G systems with stringent latency requirements.

Table 4.9 Big O Complexity with Latency 6G

Algorithm	PMS-CNN	CNN-LSTM	ELM	RELM	ORELM	MMSE	ZF
Time on CPU (ms)	0.000887	0.087261	0.000002	0.000002	0.000006	0.000002	0.000002

From Table 4.9, it can be concluded that the PMS-CNN can achieve low latency usage, enabling its practical application in UM-MIMO systems to support 6G applications.

4.7 Summary

This chapter thoroughly assessed the performance of various signal detection algorithms in an UM-MIMO system, particularly focusing on the integration of advanced machine learning techniques like CNN-LSTM and ELM variants with 256QAM modulation. Detailed analyses were presented on MSE and BER, where the PMS-CNN method consistently showed superior performance by achieving the lowest MSE and BER across various SNR levels, highlighting its efficiency and robustness in handling high-density signal environments. Additionally, channel capacity evaluations revealed that higher-order modulation schemes maximize throughput at higher SNRs, balancing data rate enhancements with acceptable error rates. Outage probability and computational complexity assessments further illuminated the practical implications of deploying these algorithms in real-world settings, emphasizing the importance of selecting appropriate techniques that align with specific system demands and operational conditions to optimize both performance and resource utilization in next-generation wireless communication systems.

CHAPTER V

CONCLUSIONS

5.1 Conclusions

The PMS-CNN method within UM-MIMO systems demonstrates exemplary performance across various benchmarks critical for modern telecommunications. Detailed evaluations in MSE consistently show the PMS method maintaining extremely low MSE values, far outpacing competitors like ZF, MMSE, and various ELM adaptations, especially evident across a broad spectrum of SNR levels. This robust performance in error minimization attests to its superior signal detection capabilities, making it highly effective in environments demanding high precision. Furthermore, the BER analysis confirms the PMS-CNN superior ability to maintain low error rates, thereby ensuring high-quality signal integrity under diverse operational conditions. This is particularly advantageous for applications that require reliable high data throughput with minimal permissible error margins.

PMS-CNN offers a powerful and efficient approach for signal detection and regression. Its ability to extract and combine multi-scale features makes it well-suited for handling complex and varied signal data, providing a robust solution for real-world applications. In terms of system capacity and reliability, the PMS-CNN method significantly excels by leveraging the spatial multiplexing strengths of UM-MIMO systems to achieve the highest data transmission rates, as evidenced in the channel capacity evaluations. It consistently shows the highest throughput levels at varying SNRs, effectively utilizing higher-order modulation schemes like 256QAM without sacrificing performance. Additionally, the algorithm demonstrates a near-zero outage probability across extensive capacity ranges, underscoring its reliability and making it a preferred choice for critical communication sectors. Despite its computational demands, the PMS-CNN method manages to balance complexity and performance

efficiently, making it scalable and suitable for widespread implementation in large-scale MIMO systems.

In Table 5.1 can be explained as follows. Enhanced Mobile Broadband (eMBB), Video streaming may experience occasional buffering, but downloading large files ensures rapid data transfer. Using the internet simultaneously by multiple people in an office requiring high data transmission speeds enhances work efficiency. Although there may be some errors, they do not significantly impact performance. With throughput reaching up to 5.442 Gbps, these services become more efficient, providing a better user experience. Massive Machine Type Communications (mMTC), IoT devices that are not highly critical and can tolerate communication errors, such as low-importance sensors. High throughput enables the support of many devices transmitting data simultaneously.

Table 5.1 Summary of Services and Applications for 5G and 6G

Modulation	BER	Throughput (Gbps)
256QAM	10^{-2}	5.442 Gbps
Services		
Enhanced Mobile Broadband (eMBB)		
Massive Machine Type Communications (mMTC)		

This blend of high efficiency, reliability, and superior performance establishes the PMS-CNN method as a pivotal solution for advancing the capabilities of next-generation wireless communication networks.

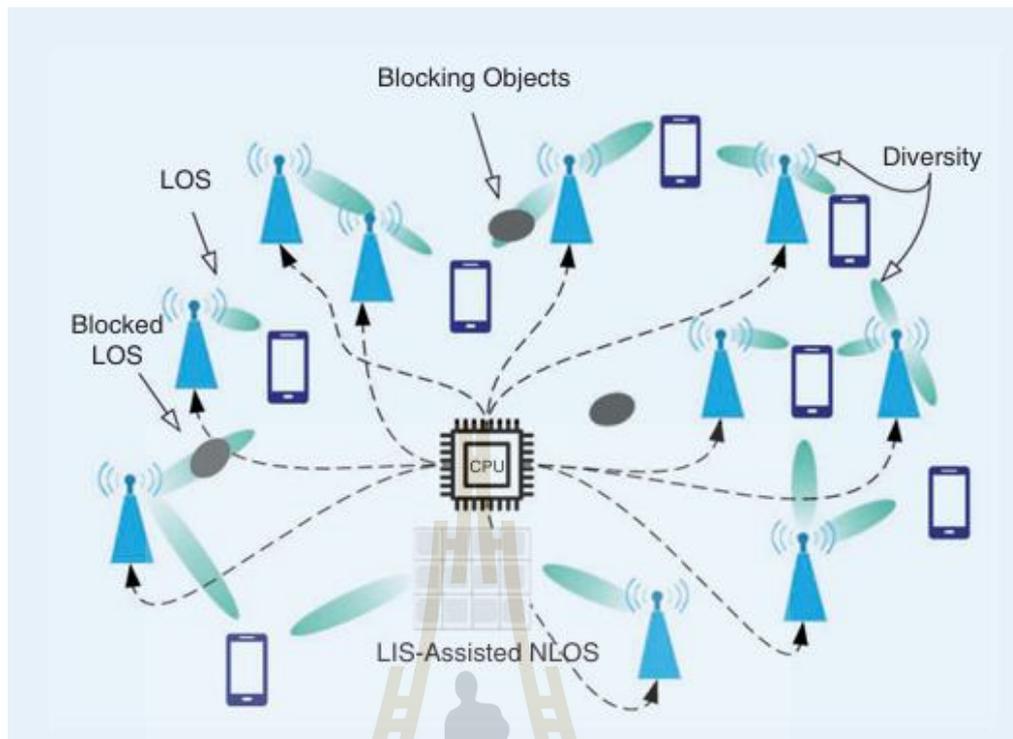


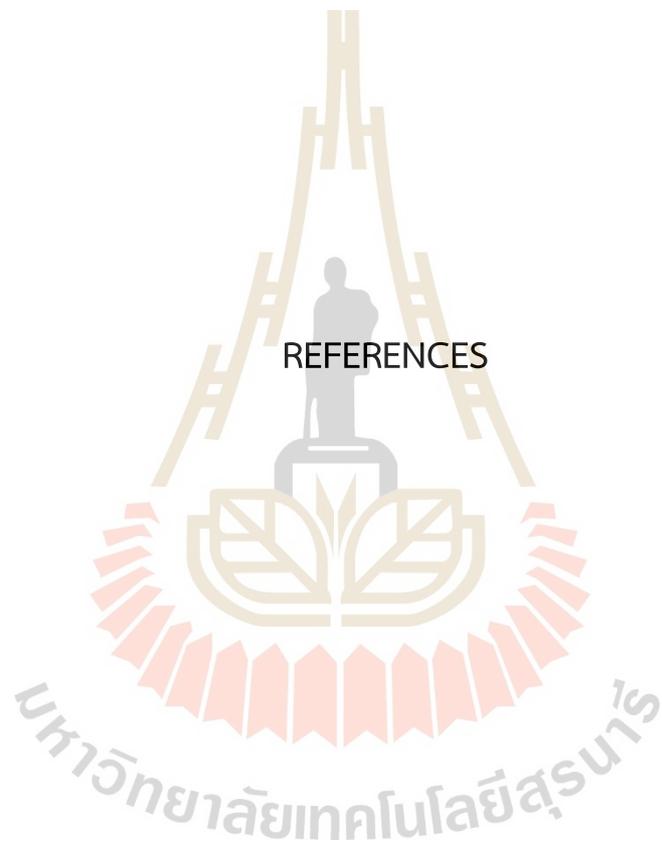
Figure 5.1 Application for Next communication networks (Thantharate, 2019).

In Figure 5.1, The PMS-CNN can achieve low latency usage, enabling its practical application in UM-MIMO systems to support 6G applications. If the model is implemented, it will be in the form of specialized chips that are developed and integrated into the Radio Access Network (RAN), Core Network, User and Control Plane. PMS-CNN can be used in the RAN for signal detection and automatic resource management, enhancing communication efficiency and reducing latency.

5.1 Future works

In this thesis, the researchers simulated the system using signal detection in UM-MIMO using a deep learning computer program. In the future, it will be very good to apply it to operations to help optimize communication networks. The model can also be used to develop channel estimates. Or channel capacity to make the system more efficient than before by looking at the Channel Capacity results that the researchers have studied, including data transmission rates.

REFERENCES



REFERENCES

- Chen, S., Liang, Y. C., Sun, S., Kang, S., Cheng, W., & Peng, M. (2020). Vision, requirements, and technology trend of 6G: How to tackle the challenges of system coverage, capacity, user data-rate and movement speed. *IEEE Wireless Communications*, 27(2), 218-228.
- Zhao, J., Liu, J., Ni, S., & Gong, Y. (2020). Enhancing transmission on hybrid precoding based train-to-train communication. *Mobile Networks and Applications*, 25, 2082-2091.
- Dilli, R., Ch, R., & Jordhana, D. (2021). Ultra-massive multiple input multiple output technologies for 6G wireless networks. *Engineered Science*, 16, 308-318.
- Heath, R. W., Gonzalez-Prelcic, N., Rangan, S., Roh, W., & Sayeed, A. M. (2016). An overview of signal processing techniques for millimeter wave MIMO systems. *IEEE journal of selected topics in signal processing*, 10(3), 436-453.
- Li, Y., & Sun, J. (2023). A Model-Driven Channel Estimation Method for Millimeter-Wave Massive MIMO Systems. *Sensors*, 23(5), 2638.
- Wang, S., & Ji, Z. (2024). A Wideband Non-Stationary 3D GBSM for HAP-MIMO Communication Systems at Millimeter-Wave Bands. *Electronics*, 13(4), 678.
- Bernard Sklar (July 1997). "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization". *IEEE Communications Magazine*. 35 (7): 90–100.
- Marye, Y. W. (2022). A Stochastic Confocal Elliptic-Cylinder Channel Model for 3D MIMO in Millimeter-Wave High-Speed Train Communication System. *Electronics*, 11(13), 1948. DOI: 10.3390/electronics11131948.
- Trotobas, B., Nafkha, A., & Louët, Y. (2020). A review to massive MIMO detection algorithms: Theory and implementation. *Advanced Radio Frequency Antennas for Modern Communication and Medical Systems*.

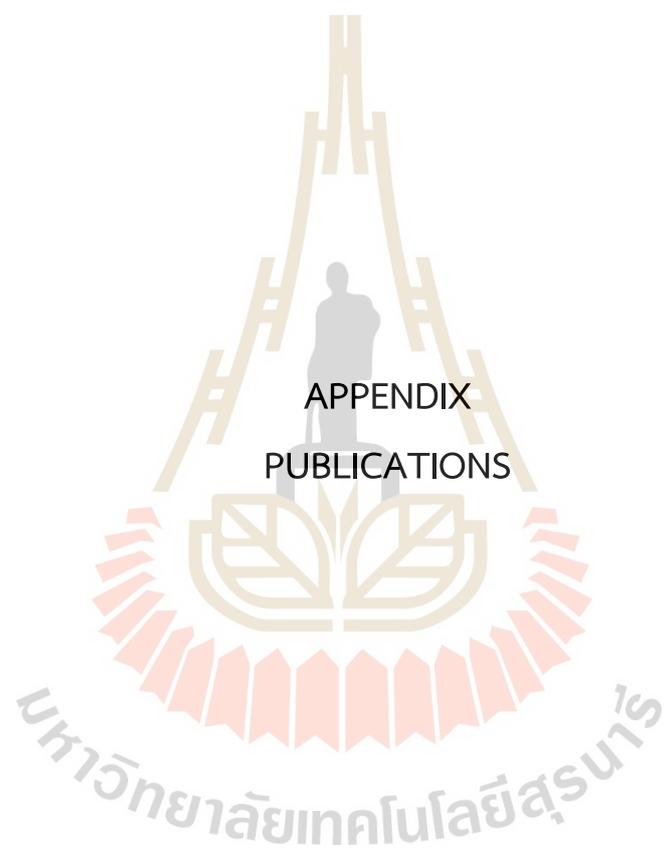
- Minango, J., Altamirano, C.D. and de Almeida, C. (2018). Performance difference between zero-forcing and maximum likelihood detectors in massive MIMO systems. **Electronics Letters**, 54(24), 1464-1466.
- Jeon, Y. S., Lee, N., Hong, S. N., & Heath, R. W. (2018, June). A low complexity ML detection for uplink massive MIMO systems with one-bit ADCs. In 2018 IEEE 87th Vehicular Technology Conference (VTC Spring) (pp. 1-5). IEEE.
- Wang, Q., Zhou, T., Zhang, H., Hu, H., Pignaton de Freitas, E., & Feng, S. (2024). Deep Learning-Based Detection Algorithm for the Multi-User MIMO-NOMA System. *Electronics*, 13(2), 255.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (Vol. 326). John Wiley & Sons.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.
- Tan, P. N., Steinbach, M., & Kumar, V. (2013). **Introduction to Data Mining**. Pearson.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The

- shared views of four research groups. *IEEE Signal processing magazine*, 29(6), 82-97.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1), 73-101.
- Charbonnier, P., Blanc-Féraud, L., Aubert, G., & Barlaud, M. (1994). Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2), 298-311.
- Koenker, R., & Bassett, G. (1978). Regression Quantiles. *Econometrica*, 46(1), 33-50.
- Koenker, R. (2005). *Quantile Regression*. Cambridge University Press.
- Proakis, J. G., & Salehi, M. (2008). *Digital Communications*. McGraw-Hill.
- Sklar, B. (2001). *Digital Communications: Fundamentals and Applications*. Prentice Hall.
- Haykin, S. (2008). *Communication Systems*. John Wiley & Sons.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Johns Hopkins University Press.
- Strang, G. (2006). *Linear Algebra and Its Applications* (4th ed.). Cengage Learning.
- Nocedal, J., & Wright, S. J. (2006). *Numerical Optimization* (2nd ed.). Springer.
- Trefethen, L. N., & Bau, D. (2022). *Numerical linear algebra*. Society for Industrial and Applied Mathematics.

- Higham, N. J. (2002). Accuracy and stability of numerical algorithms. Society for industrial and applied mathematics.
- Sarieddeen, H., Alouini, M. S., & Al-Naffouri, T. Y. (2019). Terahertz-band ultra-massive spatial modulation MIMO. *IEEE Journal on Selected Areas in Communications*, 37(9), 2040-2052.
- Nie, S., & Akyildiz, I. F. (2019, December). Deep kernel learning-based channel estimation in ultra-massive MIMO communications at 0.06-10 THz. In *2019 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.
- Murshed, R. U., Ashraf, Z. B., Hridhon, A. H., Munasinghe, K., Jamalipour, A., & Hossain, M. D. (2022). A CNN-LSTM-based Fusion Separation Deep Neural Network for 6G Ultra-Massive MIMO Hybrid Beamforming. *arXiv preprint arXiv:2209.12893*.
- Chen, Y., Yan, L., & Han, C. (2021). Hybrid Spherical- and Planar-Wave Modeling and DCNN-Powered Estimation of Terahertz Ultra-Massive MIMO Channels. *IEEE Transactions on Communications*, 69(10), 7063-7078.
- Ju, H., Zhang, H., Li, L., Li, X., & Dong, B. (2024). A comparative study of deep learning and iterative algorithms for joint channel estimation and signal detection in OFDM systems. *Signal Processing*, 109554.
- Yang, J., Chen, Y., Gao, X., Slock, D. T. M., & Xia, X. G. (2024). Signal Detection for Ultra-Massive MIMO: An Information Geometry Approach. *IEEE Transactions on Signal Processing*, 72, 824-837.
- Ngo, H. Q., Larsson, E. G., & Marzetta, T. L. (2013). Energy and spectral efficiency of very large multiuser MIMO systems. *IEEE Transactions on Communications*, 61(4), 1436-1449.
- Yan, L., Han, C., & Yuan, J. (2020). A dynamic array-of-subarrays architecture and hybrid precoding algorithms for terahertz wireless communications. *IEEE Journal on Selected Areas in Communications*, 38(9), 2041-2056.
- Yan, L., Han, C., & Yuan, J. (2022). Energy-efficient dynamic-subarray with fixed true-time-delay design for terahertz wideband hybrid beamforming. *IEEE Journal on Selected Areas in Communications*, 40(10), 2840-2854.

- Morsali, A., Haghghat, A., & Champagne, B. (2020). Generalized framework for hybrid analog/digital signal processing in massive and ultra-massive-MIMO systems. *IEEE Access*, 8, 100262-100279.
- Hu, X., Deng, R., Di, B., Zhang, H., & Song, L. (2022). Holographic beamforming for ultra massive MIMO with limited radiation amplitudes: How many quantized bits do we need?. *IEEE Communications Letters*, 26(6), 1403-1407.
- Jamali, V., Tulino, A. M., Fischer, G., Müller, R. R., & Schober, R. (2020). Intelligent surface-aided transmitter architectures for millimeter-wave ultra massive MIMO systems. *IEEE open journal of the communications society*, 2, 144-167.
- Yan, L., Chen, Y., Han, C., & Yuan, J. (2021). Joint inter-path and intra-path multiplexing for terahertz widely-spaced multi-subarray hybrid beamforming systems. *IEEE Transactions on Communications*, 70(2), 1391-1406.
- Elbir, A. M., Mishra, K. V., & Chatzinotas, S. (2021). Terahertz-band joint ultra-massive MIMO radar-communications: Model-based and model-free hybrid beamforming. *IEEE Journal of Selected Topics in Signal Processing*, 15(6), 1468-1483.
- Sarieddeen, H., Abdallah, A., Mansour, M. M., Alouini, M. S., & Al-Naffouri, T. Y. (2021). Terahertz-band MIMO-NOMA: Adaptive superposition coding and subspace detection. *IEEE Open Journal of the Communications Society*, 2, 2628-2644.
- Tarboush, S., Sarieddeen, H., Chen, H., Loukil, M. H., Jemaa, H., Alouini, M. S., & Al-Naffouri, T. Y. (2021). TeraMIMO: A channel simulator for wideband ultra-massive MIMO terahertz communications. *IEEE Transactions on Vehicular Technology*, 70(12), 12325-12341.
- Shahjalal, M., Rahman, M. H., Ali, M. O., Chung, B., & Jang, Y. M. (2021, August). User clustering techniques for massive MIMO-NOMA enabled mmWave/THz communications in 6G. In *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 379-383). IEEE.
- Thantharate, A., Paropkari, R., Walunj, V., & Beard, C. (2019, October). DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0762-0767). IEEE.

- Faisal, A., Sareddeen, H., Dahrouj, H., Al-Naffouri, T. Y., & Alouini, M. S. (2020). Ultramassive MIMO systems at terahertz bands: Prospects and challenges. *IEEE Vehicular Technology Magazine*, 15(4), 33-42.
- Jain, P., Gupta, A., & Kumar, N. (2022). A vision towards integrated 6G communication networks: Promising technologies, architecture, and use-cases. *Physical Communication*, 55, 101917.
- Nguyen, V. L., Lin, P. C., Cheng, B. C., Hwang, R. H., & Lin, Y. D. (2021). Security and privacy for 6G: A survey on prospective technologies and challenges. *IEEE Communications Surveys & Tutorials*, 23(4), 2384-2428.
- Uusitalo, M. A., Rugeland, P., Boldi, M. R., Strinati, E. C., Demestichas, P., Ericson, M., ... & Zou, Y. (2021). 6G vision, value, use cases and technologies from European 6G flagship project Hexa-X. *IEEE access*, 9, 160004-160020.
- Thantharate, A., Paropkari, R., Walunj, V., & Beard, C. (2019, October). DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0762-0767). IEEE.
- Orten, P., & Svensson, A. (2002). Sequential decoding of convolutional codes for Rayleigh fading channels. *Wireless Personal Communications*, 20, 61-74.



APPENDIX
PUBLICATIONS

List of Publications

International Journal Paper

- Keawin, C., Innok, A., & Uthansakul, P. (2024, March). Optimization of Signal Detection Using Deep CNN in Ultra-Massive MIMO. In *Telecom* (Vol. 5, No. 2, pp. 280-295). MDPI.
- Innok, A., Keawin, C., & Uthansakul, P. (2022). Optimization of Channel Estimation Using ELMx-based in Massive MIMO. *CMC-COMPUTERS MATERIALS & CONTINUA*, 73(1), 103-118.





Article

Optimization of Signal Detection Using Deep CNN in Ultra-Massive MIMO

Chittapon Keawin¹, Apinya Innok² and Peerapong Uthansakul^{1,*}

¹ School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand; d6400507@g.sut.ac.th

² Department of Telecommunications Engineering, Rajamangala University of Technology Isan, Nakhon Ratchasima 30000, Thailand; apinya.io@muti.ac.th

* Correspondence: uthansakul@sut.ac.th; Tel.: +66-850865588

Abstract: This paper addresses the evolving landscape of communication technology, emphasizing the pivotal role of 5G and the emerging 6G networks in accommodating the increasing demand for high-speed and accurate data transmission. We delve into the advancements in 5G technology, particularly the implementation of millimeter wave (mmWave) frequencies ranging from 30 to 300 GHz. These advancements are instrumental in enhancing applications requiring massive data transmission and reception, facilitated by massive MIMO (multiple input multiple output) systems. Looking towards the future, this paper forecasts the necessity for faster data transmission technologies, shifting the focus toward the development of 6G networks. These future networks are projected to employ ultra-massive MIMO systems in the terahertz band, operating within 0.1–10 THz frequency ranges. A significant part of our research is dedicated to exploring advanced signal detection techniques, helping to mitigate the impact of interference and improve accuracy in data transmission and enabling more efficient communication, even in environments with high levels of noise, and including zero forcing (ZF) and minimum mean square error (MMSE) methods, which form the cornerstone of our proposed approach. Additionally, signal detection contributes to the development of new communication technologies such as 5G and 6G, which require a high data transmission efficiency and rapid response speeds. The core contribution of this study lies in the application of deep learning to signal detection in ultra-massive MIMO systems, a critical component of 6G technology. We compare this approach with existing ELMx-based machine learning methods, focusing on algorithmic efficiency and computational performance. Our comparative analysis included the regularized extreme learning machine (RELM) and the outlier robust extreme learning machine (ORELM), juxtaposed with ZF and MMSE methods. Simulation results indicated the superiority of our convolutional neural network for signal detection (CNN-SD) over the traditional ELMx-based, ZF, and MMSE methods, particularly in terms of channel capacity and bit error rate. Furthermore, we demonstrate the computational efficiency and reduced complexity of the CNN-SD method, underscoring its suitability for future expansive MIMO systems.

Keywords: signal detection; ELM; deep learning; ultra-massive MIMO



Citation: Keawin, C.; Innok, A.; Uthansakul, P. Optimization of Signal Detection Using Deep CNN in Ultra-Massive MIMO. *Telecom* 2024, 5, 280–295. <https://doi.org/10.3390/telecom5020014>

Received: 6 February 2024

Revised: 18 March 2024

Accepted: 26 March 2024

Published: 29 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wireless communication technology has continuously evolved, especially with the development of multiple input multiple output (MIMO) systems that use multiple receiving and sending antennas for data transmission. This advancement has led to the extensive study of massive MIMO systems due to the increasing demand for higher data transmission capacities, a critical component as we look ahead to 6G communication systems [1,2]. Ultra-massive MIMO has been identified as a key technology for enhancing data transmission in the 6G network [3–5]. Currently, 5G networks utilize various frequency bands, including sub-6GHz and millimeter wave (mmWave), catering to virtual environments and the

Internet of Things (IoT). However, research and development into 6G are ongoing, aiming to support more connected devices and higher capacities, while providing faster data rates and lower latency than 5G [6]. Additionally, 6G seeks to enhance communication security and reliability and may introduce new applications, such as holographic communication. Technologies for AI and autonomous cars are also being developed in anticipation of 6G. Compared to 5G, 6G is expected to increase data rates by 10 to 100 times, supporting up to Tb/s and 10 Gb/s user experience data rates. Moreover, 6G could use flexible frequency-sharing technology, to optimize frequency reuse. Ultra-massive MIMO technology, a crucial aspect of 6G's future, can be categorized into four main areas: frequency bands, transmission mode, intelligent transmission, and integrated network. As mentioned, 6G will support the use of terahertz frequencies ranging from 0.1 THz to 10 THz, which will aid in the development of systems and meet future needs such as the application of AI, medical devices, or autonomous driving systems that require lower latency and higher precision. This also includes high-speed internet access, allowing everyone to access increased communication resources [7,8].

In order to overcome limitations and highlight differences in wireless communications, signal detection techniques like zero forcing (ZF) and MMSE (minimum mean square error) are utilized, along with deep learning techniques [9,10]. Current 5G technology aims to provide higher data rates and lower latency than 4G, accommodating new applications like virtualization and more connected devices. Ultra-massive MIMO considers using more than 128 antenna elements at both the transmitter and receiver [11]. The Saleh–Valenzuela (SV) channel model has been selected by many researchers and is applicable in various communication formats, including hybrid beamforming, hybrid precoding, and spatial multiplexing [12]. Machine learning architecture, particularly ELMx-based systems, that includes ELM is widely used in communication for channel estimation [13], including articles of interest regarding the application of AI in traffic prediction [14].

The main contributions of this work are summarized as follows:

- We propose the CNN-SD, which integrates three machine learning algorithms of ELM, RELM, and ORELM for signal detection. The hidden layer bias and input weight in CNN-SD are randomly generated from distributions [15–17].
- We foresee greater complexity with the larger number of antennas. The application of deep learning to signal detection contributes to improved performance and reduced complexity, rather than using more complex channel estimation methods.
- We developed a modeling framework for detailed learning and simultaneous regression, incorporating real and imaginary components of complex matrices into the input layers of artificial neural networks to minimize potential errors. This approach allows us to reduce the system's overall complexity and enhance its efficiency.

The simulation results demonstrate that, in terms of mean square error (MSE), bit error rate (BER), channel capacity, outage probability, and computational time, CNN-SD performed better for signal detection.

The remainder of this paper is summarized as follows: Section 2 details the materials and methods. The proposed CNN-SD algorithm is specifically described in Section 2. Section 3 gives simulation figures to illustrate the signal detection performance of the proposed algorithm. Section 4 is the conclusions.

2. Materials and Methods

For the construction of a system model, we developed a comprehensive communication framework emphasizing ultra-massive MIMO technology, with the objective of facilitating the computation of a diverse array of results. This will be explained in Section 2.1, including massive MIMO and ultra-massive MIMO.

2.1. Fundamentals System Models

A simple system model has transmitting antennas (M_T) and receiving antennas (M_R). The relation between the transmitted and received signal is spatial multiplexing. The memoryless MIMO flat fading channel (narrowband model) is given by

$$Y = Hx + n \tag{1}$$

According to the framework of Equation (1), the matrix H , which represents a signal response channel matrix with dimensions ($M_R \times M_T$), is utilized to clarify the application of both massive MIMO and ultra-massive MIMO systems. Furthermore, the variable n , defined as an additive white complex Gaussian noise vector with dimensions ($M_R \times 1$), is recognized as a primary noise model in information theory. This model is specifically crafted to replicate the effects of the diverse random processes commonly observed in natural settings. The matrix representation thus effectively captures the relationship between transmitted and received signals, illustrating the intricate dynamics of signal propagation in these advanced communication systems as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{M_R} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,M_T} \\ h_{21} & h_{22} & \cdots & h_{2,M_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_R,1} & h_{M_R,2} & \cdots & h_{M_R,M_T} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M_T} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{M_R} \end{bmatrix} \tag{2}$$

2.1.1. Massive MIMO

Massive MIMO has been much researched regarding channel response. First, as illustrated in Figure 1, we investigate a typical massive MIMO system. A block diagram is assumed for delivering data sources from the X (vector of transmitted signals) to the Y (vector of received signals). In this model, the channel parameter h represents the communication link between nodes X and Y , which clearly shows the behavior of Equation (1). Rayleigh channels are commonly used in massive MIMO systems, due to their ease and accuracy in simulating the internal conditions of the communication channel. They can also estimate the impact of low-risk distribution in a large number of wireless communications. However, it should be noted that using Rayleigh channels in massive MIMO often results in simulating channel states invariantly. Since the simulation is local, this approach may yield results that are not entirely accurate to natural conditions. Including the future possibilities of 6G, studies across various works have found that the current number of antennas in massive MIMO still face operational limitations in the THz frequency bands, starting with as few as 16 antennas. Meanwhile, the concept of ultra-massive MIMO, discussed in this research, explores the use of a significantly larger number of antennas, up to 256, representing a full-capacity model. Further details will be provided in the subsequent section.

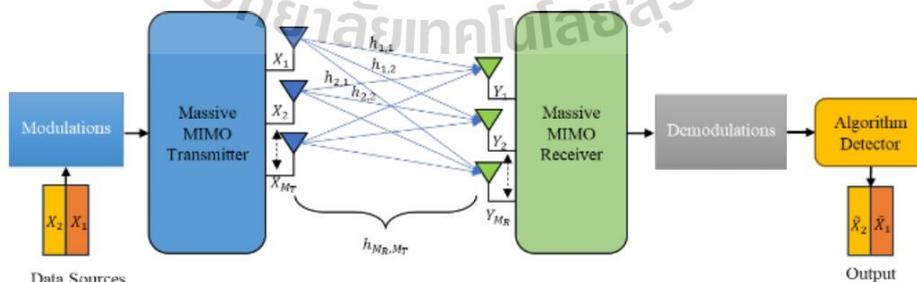


Figure 1. The structure of massive MIMO system.

2.1.2. Ultra-Massive MIMO

This paper focuses on different channels based on potential real situations. This was considered as much as possible by using the Saleh–Valenzuela signal model, in which the signal consists of a combination of discrete radiation bundles. This is a typical model for mmWave signals, which have high reflectivity and low diffusion. Mathematically, the ray/cluster signal matrix can be represented as

$$H = \sum_{v=1}^{N_{clust}} \sum_{u=1}^{N_{ray}} \beta_{u,v} a_{rx}(AoA_{u,v}) a_{tx}(AoD_{u,v})^* \tag{3}$$

where u and v denote the number of subarrays at M_T and M_R , and where the signal comprises cluster N_{clust} is the cluster angle. In each group, N_{ray} is the ray, where each ray is considered in the loop of the simulation, accounting for the multipath effects encountered by the signal. Cluster $\beta_{u,v}$ shows complex gain. In addition, $AoD_{u,v}$ is the angle of departure (from the transmitting array) and $AoA_{u,v}$ is the angle of arrival (from the receiver array). The subsequent sending and receiving arrays are $a_{tx}(AoD_{u,v})$ and $a_{rx}(AoA_{u,v})$, respectively.

This section focuses on signal detection in ultra-massive MIMO systems, which is crucial for transmitting large amounts of data across multiple transmitting antennas arranged in a matrix, while simultaneously dealing with interference signals. k -QAM modulation is commonly employed to introduce and simulate intricate scenarios in ultra-massive MIMO systems, specifically in relation to the proposed modulation schemes for 6G. The modulation process is controlled by the phase of constellation mapping. The procedure involves receiving binary bits as input, converting them into complex numbers, and subsequently using them as symbols. The analysis involves a multiple-input multiple-output (MIMO) wireless system that experiences flat fading. This system utilizes multiple transmitting antennas M_T and multiple receiving antennas M_R . The symbol $X_N(p)$ denotes the transmission conducted by the antenna M_T at a precise moment p . The symbols that have been sent are organized into a vector of length M_T such as $X_N(p) = [X_1(p), \dots, X_{M_T}(p)]^T$, which is referred to as

$$X_N(p) = |N(p)| \cos(\arg\{N(p)\}) \cos(2\pi f_c p) - |N(p)| \sin(\arg\{N(p)\}) \sin(2\pi f_c p) \tag{4}$$

k is the number of modulations that covers all QAM modulation,

where $|u(t)|$ and $argu(t)$ are the amplitude and phase of the complex baseband signal, respectively.

2.2. Traditional Method

This paper focuses on the critical role of specific technology or methods in enhancing a multiple-input multiple-output (MIMO) system in Figure 2. By leveraging this technology, we aim to augment the system’s capacity, primarily by improving performance metrics such as the bit error rate. Our study involves a comprehensive comparison between conventional signal detection techniques, including ZF and MMSE, and advanced methods based on the ELMx algorithm, namely ELM, RELM, and ORELM. Moreover, we introduce the convolutional neural network-based signal detection CNN-SD algorithm, proposing it as a novel approach for further performance enhancement in MIMO systems.

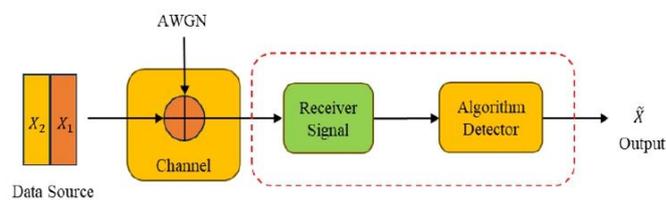


Figure 2. A block diagram of how to estimate signal detection.

2.2.1. Zero Forcing Detector

Zero forcing was an early concept in channel estimation. By detecting signals, this type of estimation is classified as a one-dimensional (1D) estimator, which means that the channel estimation is performed using test cycles. In one dimension, whether frequency or time, (ZF) signal detection reduces the amount of squared error between the received signal and the estimated value. As a result, it is possible to find \tilde{X}_{ZF} signal detection by

$$\hat{H}_{ZF} = \operatorname{argmin}_{H_{ZF}} \|Y - H_{ZF}X\|^2 \quad (5)$$

On the other hand, the signal detection can be written in the following form:

$$\tilde{X} = H^T Y \quad (6)$$

All transmitting and receiving antennas, the impulse response signals are encapsulated and presented in the form of a matrix by

$$\tilde{X}_{ZF} = YH^H(HH^H)^{-1} \quad (7)$$

In this formulation, $(*)^H$ is designated as the conjugate transpose of the matrix, and $(*)^{-1}$ pertains to the inverse of the matrix.

2.2.2. MMSE Signal Detector

The MMSE signal detector is an algorithms referenced in many works. It is calculated similarly to channel estimation [18,19]. It is the second comparator that we used for signal estimation, because it is the most commonly used and most complex calculation method. As a result, it is more accurate than the ZF signal detection given by

$$\tilde{X}_{MMSE} = \operatorname{argmin} \|Y - \tilde{X}_{MMSE}H\|^2 \quad (8)$$

The detection strategy includes accounting for noise during the computation process, which is measured by

$$\tilde{X}_{MMSE} = YH^H(HH^H + \frac{\sigma_n^2}{\sigma_h^2}I)^{-1} \quad (9)$$

In this context, I denotes the identity matrix of size $M_T \times M_R$, while σ_n^2 represents the noise variance, which is inversely proportional to the signal-to-noise ratio (SNR). Additionally, all signal response energies have been normalized, as exemplified by

$$E\{|h_{M_R, M_T}|^2\} = \sigma_h^2 \quad (10)$$

2.3. Machine Learning Method

2.3.1. Extreme Learning Machine (ELM)

This study introduces an accelerated learning method for single hidden layer feedforward networks (SLFNs) [20,21], which is designed to effectively manage networks with \tilde{N} hidden neuron nodes. Additionally, the method is tailored to handle $\tilde{N} \leq N$ training samples. The focus is on significantly speeding up the learning process in SLFNs, thereby enhancing training efficiency and effectiveness. Extreme learning machine (ELM), an algorithm within the machine learning domain that employs neural networks, is known for its impressive efficiency in regression tasks and an accelerated learning rate. This was substantiated through both theoretical analysis and empirical validation. The architectural framework of ELM is detailed in Figure 3, providing a clear visual representation of its structure. n and m respectively denote the numbers of input and output data. The analysis utilizes the ELM training process with N training samples (X_i, t_i) , where $X_i = [X_{i1}, X_{i2}, \dots, X_{in}]^T$

represents the input data, and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T$ the output data. The approach to detecting patterns in SLFNs employs a mathematical model, outlined as follows:

$$\hat{t}_j = \sum_{i=1}^{\tilde{N}} \beta_i o(c_i \cdot X_j + V_i), j = 1, 2, \dots, N \tag{11}$$

$c_i = [c_{i1}, c_{i2}, \dots, c_{im}]^T$ is the input weight vector that connects to the i -th hidden neuron, and $\beta = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ refers to the output weight vector of the i -th hidden neuron node. The term V_i is the bias of the i -th hidden neuron node, and c_i indicates the activation function in SLFNs. Unlike other machine learning algorithms, ELM can randomly assign the input weight c_i and bias V_i . A significant feature of ELM, contributing to its regression performance, is its verification of zero error, which allows an accurate approximation of all N samples as $i = 1Ntj - tj = 0$; i.e., the N equations can thus be compactly expressed as

$$H\beta = T, \text{ where } \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times m}, T = \begin{bmatrix} T_1 \\ \vdots \\ T_N \end{bmatrix}_{N \times m} \tag{12}$$

$$H(c_1, \dots, c_N, V_1, \dots, V_N, X_1, \dots, X_N) = \begin{bmatrix} o(c_1 \cdot X_1 + V_1) & \dots & o(c_N \cdot X_1 + V_N) \\ \vdots & \dots & \vdots \\ o(c_1 \cdot X_N + V_1) & \dots & o(c_N \cdot X_N + V_N) \end{bmatrix}_{N \times \tilde{N}} \tag{13}$$

where H is the neural network's hidden layer output matrix and T is the training data target matrix. The variable H represents the output matrix of the hidden layer in the neural network, whereas T represents the matrix that contains the training data targets. Moreover, β was identified as the solution with the minimum magnitude in the least-squares approach, which is essential to the linear system of the ELM solution.

$$\hat{\beta} = H^P T \tag{14}$$

where $(*)^P$ is the Moore–Penrose pseudoinverse.

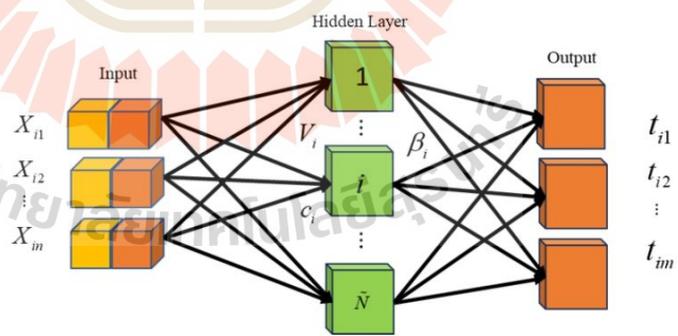


Figure 3. An illustration of an extreme learning machine.

2.3.2. Regularized Extreme Learning Machine (RELM)

While the ELM has proven effective in numerous applications, the implementation of the \tilde{N} method is recommended to circumvent overfitting and underfitting. As the study in [22] suggests, smaller norm parameters in the RELM designed for SLFNs with a sigmoid function can lead to improved generalization. This equation is versatile and applicable to

various activation functions and hidden neuron nodes like kernels, specifically to resolve challenges in the l_2 norm of β . Consequently, RELM can be described as a method by which

$$\underset{(\beta_0, \beta) \in \mathbb{R}^{N \times 1}}{\text{minimize}} \quad \frac{C}{2} \|H\beta + \beta_0 - t\|_2^2 + \frac{(1-a)}{2} \|\beta\|_2^2 + a\|\beta\|_1 \quad (15)$$

When just the l_2 norm penalty ($a = 0$) with $\beta_0 = 0$ is considered, the RELM formula is given where C and β_0 are regularization parameters.

$$\hat{\beta} = (H^H H + \frac{1}{C})^{-1} H^H t \quad (16)$$

2.3.3. Outlier-Robust Extreme Learning Machine (ORELM)

Recent modifications to the ORELM aimed to enhance its proficiency in the l_1 norm for outlier-producing techniques. Such advancements can be realized through the application of the ELM [23], as elaborated by

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \|\tau H\beta - t\|_1 \quad (17)$$

The optimal solution to the outlined optimization problem is obtained utilizing the standard l_2 norm.

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \|\tau H\beta - t\|_1 + \frac{1}{2C} \|\beta\|_2^2 \quad (18)$$

The specific processes of the ELMx-based algorithm are summarized in Algorithm 1. Noting, the ELMx based algorithm is used to obtain the estimated channel matrix in Algorithm 1.

Algorithm 1 ELM, RELM and ORELM algorithm.

- 1: **Input:** Real and Imaginary number $X \in \mathbb{R}^{n \times l}$, $Y \in \mathbb{R}^{n \times l}$
 - 2: **Output:** \tilde{X} .
 - 3: **Initialization:** Randomly assign input weight c_i and bias V_i for each hidden neuron i , where $i = 1, 2, \dots, L$.
 - 4: **Hidden Layer Output Matrix Calculation:**
 - 5: **for** $i = 1$ to N **do**
 - 6: Calculate the hidden layer output vector t_i for each input X_i using the activation function $o(x)$:
 - 7: $t_i = o(c_i \cdot x_i + V_i)$
 - 8: **end for**
 - 9: Form the hidden layer output matrix H using all H_i .
 - 10: Output using Equations (14), (16) and (18).
-

2.4. Proposed Method

Architecture of The Proposed Convolutional Neural Network for Signal Detection (CNN-SD)

A model of a serial data processing system known as a convolutional neural network (CNN) was developed to recognize and manipulate widely connected data. The basic equation is as follows:

$$O(t) = \sum_{z=0}^{L-1} I(z) \cdot w(t-z) + b \quad (19)$$

$O(t)$ represents the output of the 1D convolution at position t on the output signal O , $I(z)$ denotes the value of the input data signal I at position z , $w(t-z)$ corresponds to the value of the filter w at position $(t-z)$, and b is the bias term added to the result after the

convolution operation if bias is used. L is the length of the input signal I and the length of the filter w , which is also the size of the filter. This equation shows how to perform 1D Convolution using the input signal I and filter, and including the bias term b if given. To obtain $O(t)$ in the output signal I , which multiplies and adds values from all positions in I , we use the filter w , shifted by I positions on I . To process the fully connected layer, it receives data from the previous layer as a vector of variable values or feature values. The importance of each feature can be adjusted by multiplying it by a weight and adding a bias value to allow the system to learn how the data looks and create connections between features. The linear layer

$$y = Wx + b \quad (20)$$

y is the output vector of the linear layer with dimensions $(m, 1)$, where m is the number of output layers we need. W is a weight matrix with dimensions (m, n) and is used to multiply the input x . x is an input data vector with dimensions $(n, 1)$, where n is the number of data features. b is a bias vector with dimension $(m, 1)$ and is added to the result after multiplication by W . The rectified linear unit (ReLU) is a mathematical function commonly used in neural networks and deep learning as an activation function to process data within a layer. The results obtained by multiplying the feature values by weights and adding bias values are sent to an activation function to generate the output values of this layer. The ReLU activation function sets each layer value to zero if it is less than zero and retains the same value if it is greater than or equal to zero. The ReLU enables neural networks to learn effectively and accelerates their computations. The functional representation of the neural network ReLU function is as follows:

$$f(x) = \max(0, x) \quad (21)$$

where if the value x is positive or equal to 0, then the value $f(x)$ will be x itself. If the value of x is less than 0, then $f(x)$ will be zero. In the initial phase of data preparation, the role of information is critical. This phase is methodically divided into two primary categories, as delineated in Figure 4. The first category, designated as 'training data', consists of variably prepared data from ultra-massive MIMO systems, intended to facilitate the algorithm's capabilities in computation and memory retention. The second category, named 'teaching data', is structured to enable the algorithm to assimilate and accurately reproduce the desired outputs. The differentiation between these data types is established based on their channel response attributes within the ultra-massive MIMO communication system after the data preparation stage. The sequential procedures of the CNN-SD algorithms are comprehensively outlined in Algorithm 2, providing a clear framework for the algorithmic workflow.

Algorithm 2 CNN-SD algorithm.

- 1: **Input:** Real and Imaginary number $X \in \mathbb{R}^{n \times l}$, $Y \in \mathbb{R}^{n \times l}$
 - 2: **Output:** \tilde{X} .
 - 3: **Initialization:** Randomly assign input weight W_i and bias b_i for each hidden neuron i , where $i = 1, 2, \dots, L$.
 - 4: **procedure** PROCESSING(X)
 - 5: $X_{conv1} \leftarrow \text{ReLU}(W_{conv1} * X + b_{conv1})$
 - 6: $X_{conv2} \leftarrow \text{ReLU}(W_{conv2} * X_{conv1} + b_{conv2})$
 - 7: $X_{flat} \leftarrow \text{Flatten}(X_{conv2})$
 - 8: $X_{fc1} \leftarrow \text{ReLU}(W_{fc1} \cdot X_{flat} + b_{fc1})$
 - 9: $X_{fc2} \leftarrow \text{ReLU}(W_{fc2} \cdot X_{fc1} + b_{fc2})$
 - 10: $X_{fc3} \leftarrow \text{ReLU}(W_{fc3} \cdot X_{fc2} + b_{fc3})$
 - 11: $\tilde{X} \leftarrow W_{predict} \cdot X_{fc3} + b_{predict}$
 - 12: **return** \tilde{X}
 - 13: **end procedure**
-

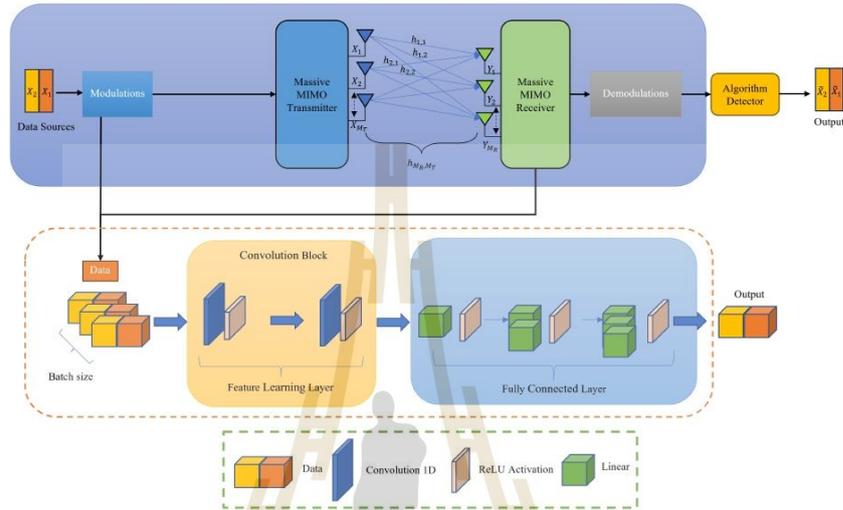


Figure 4. The process for CNN-SD.

2.5. Channel Capacity

The concept of the Shannon capacity of a channel is centered on the maximum data rate that can be reached within a certain bandwidth (BW) and at a specific signal-to-noise ratio. This theoretical capacity suggests a decrease in bit error rate (BER) that is challenging to achieve in real-world scenarios. However, as link level design techniques evolve, the actual data rate for noise channels is approaching this theoretical boundary, as discussed in [24]. Often measured in bits per second (bps)/Hz or equivalent units, a high channel capacity denotes the communication system’s capability for swift and effective data transmission. It is formulated as

$$C = \log_2 \det \left[I_{M_R} + \frac{P_t}{P_n M_T} H H^H \right] \tag{22}$$

Here, I_{M_R} is the identity matrix with dimensions $(M_R \times M_R)$, H characterizes the channel response sized size $(M_T \times M_R)$, $(*)^H$ is the conjugate transpose, and P_t/P_n quantifies the signal noise ratio (SNR). In estimating the capacity of the channel, our examination included channel responses obtained through the LS method, the prevalent MMSE method, and three additional approaches rooted in machine learning applications, namely ELM, RELM, and ORELM. The formula for calculating the estimated capacity within the CNN-SD model is prescribed as follows (6) to produce (23).

2.6. Outage Probability

Another primary performance indicator in communication techniques is the outage probability, as discussed in [20]. This is often expressed as a percentage and depends on the channel’s state and the interference in the system. Outage probability measures the risk that a communication signal may not be received or may be lost. If the outage probability is low, the communication system is efficient in transmitting data. The outage probability can be determined as follows:

$$P_{out}(C_{es} < R) \tag{23}$$

where R is the rate of capacity. As a consequence, the most favorable course of action for the transmitter is to employ data encryption. This decision is contingent upon the channel gain

being sufficient to accommodate the desired rate denoted as R . Under such circumstances, the attainment of dependable communication becomes feasible; otherwise, an outage is incurred. In the presence of a fading gain represented as h , one may conceive of the channel as permitting the flow of information. Provided that the volume of data surpasses the designated rate, the prospect of reliable decoding becomes attainable. The Saleh channel's outage probability, as a function of the transmission rate R , can be expressed as follows:

$$P_{out}(R) = 1 - \exp\left(\frac{-2^R - 1}{SNR}\right) \quad (24)$$

P_{out} signifies the outage probability of the system, characterized by how the destination performs detection when relying only on the received signals from the relay node.

2.7. Total Loss of Algorithm

2.7.1. Mean Square Error (MSE)

The efficacy of machine learning algorithms can be analyzed through multiple approaches. Consequently, mean squared error (MSE) was chosen for the performance analysis, to yield definitive conclusions, as corroborated by references [11,12]. This metric, frequently used in performance evaluation, requires calculating the error \tilde{X} , which represents the difference from the actual X , and then averaging this error. Based on this difference, the gradient of loss is computed and utilized for backpropagating the weights. The next step involves applying gradient descent in the subsequent training phase to lower losses. The loss function used in regression is presented by

$$MSE_N = \frac{1}{N} \sum_{i=1}^N (X - \tilde{X}_i)^2 \quad (25)$$

2.7.2. Training Loss

The total loss can help us check the feasibility of the dataset between trainings, because it can specify the range to test to obtain an appropriate dataset. Practically, the mean loss is calculated for each batch and then averaged across all the batches within an epoch. This provides a comprehensive assessment of the model's performance on the training data, helping to follow its progress over time. Considering the fast-processing time, calculating the total number of training rounds in epochs is performed as follows:

$$T_B = T_D / B_S \quad (26)$$

where T_B is the total number of batches, T_D is total data size, and B_S is the batch size. For determination of the number of samples that will be distributed across the network, the batch size is essential. When training a deep learning model, the average loss is typically computed over the entire training dataset, in small batches of samples. This is known as batch training. This is because training the model on the entire dataset in one go could be computationally expensive, and it could also cause the model to overfit on the training data.

2.7.3. Validation Loss

The formula for calculating validation loss is likely MSE (25) but we only considered the amount of validation data to calculate, where M is the number of an epochs and f is the loss function. However, we calculated the validation loss as

$$V = \frac{1}{M} \sum f(\hat{Y}_d, Y_d) \quad (27)$$

where \hat{Y}_d is prediction data and Y_d is teaching data. In addition, the interpretation of the results was very important and was divided into three parts: Underfitting refers to a situation in scenario 1 where the results indicate that additional training is required to

decrease the loss experienced during the training process. Alternatively, we can enhance the training data by either acquiring additional samples or augmenting the existing data. Overfitting occurs when, in scenario 2, the validation loss surpasses the training loss. In scenario 3, a good fit is characterized by the training loss and validation loss decreasing and reaching a stable point.

3. Result and Discussion

3.1. Dataset Setup

The dataset employed consisted of three components: training data, testing data, and validation data. These components encompassed two datasets: the received signal data set, denoted as Y , and the transmitted data set, denoted as X , in accordance with the principles of communication systems. We collected and simulated a total of 100,000 datasets.

3.2. MSE and BER

This section of the study focuses on evaluating the mean square error (MSE) and bit error rate (BER), as well as assessing the effectiveness of zero forcing (ZF), minimum mean square error (MMSE), and ELMx-based signal detection approaches in confirming the performance of several CNN-SD algorithms. These strategies were evaluated in ultra-massive MIMO systems that included 256 transmitting and receiving antennas, each with 256QAM modulation mapping and a certain number of pilots. The attributes and approaches of the ELMx-based and CNN-SD algorithms are explained in Algorithms 1 and 2, correspondingly. The results of the comparative mean squared error (MSE) performance are shown in Figure 5. It was noted that ZF, because of its more rudimentary approach, was less successful compared to the other four methodologies. On the other hand, MMSE outperformed the least squares (LS) method, because of its increased complexity. The ELMx-based methodology, a machine learning technique, outperformed both ZF and MMSE by effectively using large training and testing datasets to accurately replicate the necessary data. The CNN-SD approach distinguished itself by showing a superior performance, which was attributed to its unique structural arrangement, which is optimized for exceptional results.

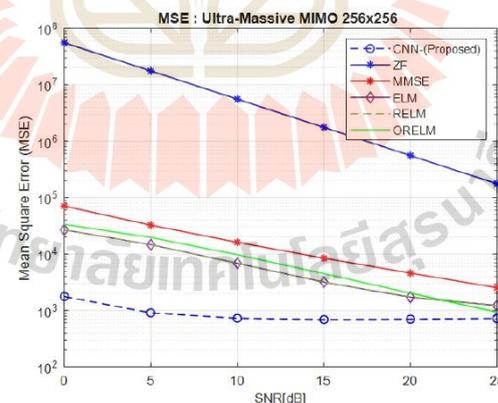


Figure 5. The performance of MSE in ultra-massive MIMO systems.

Figure 6 shows the performance of BER for all signal detection techniques in an ultra-massive MIMO-based communication system, which included 256 transmitting antennas (MT) and 256 receiving antennas (MR). The CNN-SD outperformed the fundamental approaches such as LS, MMSE, and ELMx-based machine learning in signal detection. The

results revealed that the BER performance of the CNN-SD algorithm was the best among all methods.

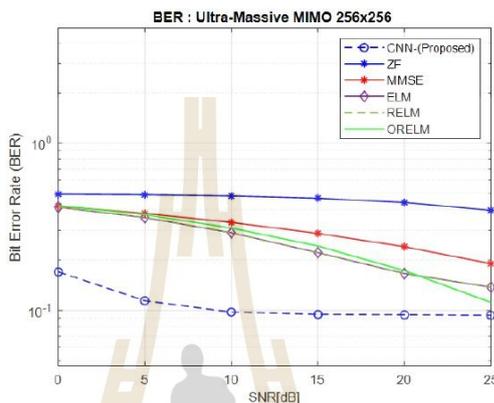


Figure 6. The performance of BER in ultra-massive MIMO systems.

3.3. Model Validation

Figure 7 shows the performance of the good fit learning curve for training loss and validation loss, and it shows that the values were closely grouped around the range of 1–3%. Therefore, the model exhibited a strong aptitude for learning. This model can be utilized to make precise predictions for previously unseen data. Alternatively, it can be referred to as a model that exhibits the ability to apply its knowledge to unfamiliar data. We divided the 100,000 datasets into 60% training data, 20% testing data, and 20% validation data, for our proposed method.

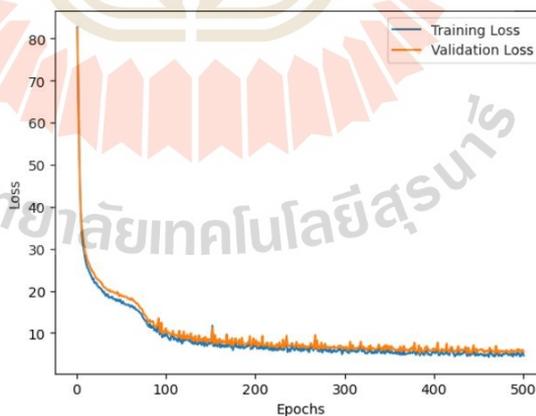


Figure 7. The performance of the CNN-SD model.

3.4. Computational Time

Figure 8 shows that the computation time was different in the computational comparison between ELMx and CNN-SD, according to the computational complexity for the

number of nodes from 2048 nodes. The result for ELMx, such as ELM RELM and ORELM, revealed that the computational time was higher than CNN-SD because the data management in CNN-SD had a batch size that could help reduce the use of computer resources. This is because training on a large dataset can reduce communication time between processors. It can significantly help reduce the use of computer resources and the management of data in RAM. This is important when training models that are data-intensive or that have complex structures.

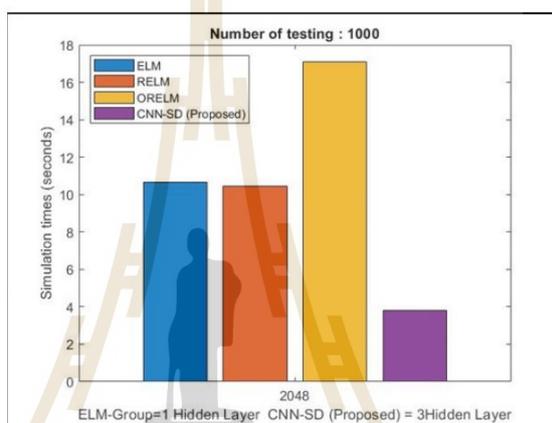


Figure 8. The computational times for ELMx and CNN-SD.

3.5. Channel Capacity and Outage Probability

Another focus of this section was on the performance metrics of channel capacity and outage probability within ultra-massive MIMO systems. Equation (19) was used for processing and comparing the channel capacity, as presented in Figures 9 and 10.

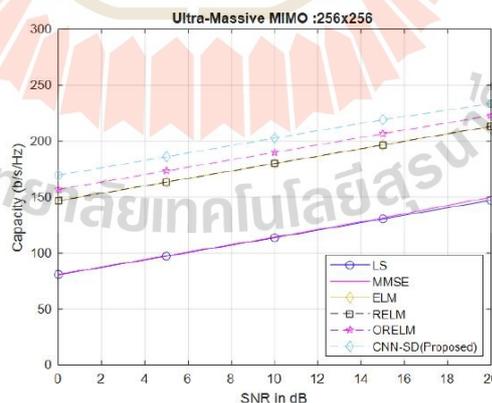


Figure 9. The channel capacity performance of ELMx and CNN-SD in ultra-massive MIMO systems.

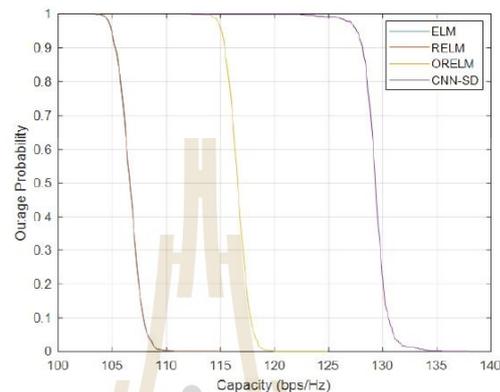


Figure 10. The outage probability performance of ELMx and CNN-SD in ultra-massive MIMO systems.

Meanwhile, for determining the outage probability, Equations (20) and (21) were utilized, as demonstrated in Figure 10. The test result for channel capacity vs. SNR shows that the machine learning algorithm ELM, RELM, and ORELM techniques provided a lower channel capacity than the CNN-SD techniques. In Figure 9, the CNN-SD at 10 SNRdB had a capacity 200 bps/Hz more than the ELMx-based techniques. It is apparent that the ultra-massive MIMO channel is highly likely to achieve high capacity. At a 90% probability level, the capacities stand at 105 bps/Hz for ELM, 105 bps/Hz for RELM, and 116 bps/Hz for ORELM. Consequently, this demonstrates a significant likelihood of high channel capacity when compared to ELM, RELM, and ORELM. The CNN-SD, with a capacity of 128 bps/Hz, exhibited excellent performance in line with the method, as depicted in Figure 10.

4. Conclusions

Ultra-massive MIMO systems, representing a significant future advancement, incorporate additional communication antennas and new techniques or procedures that can enhance system efficiency and problem-solving capabilities. This paper focused on finding methods that lead to more effective communication and reduce issues. One such approach was the introduction of a CNN-SD deep learning signal detection technique within an ultra-massive MIMO system. The authors employed various techniques for comparison with the LS, MMSE, and ELMx-based machine learning groups. Three algorithms of ELM, RELM, and ORELM, along with the proposed CNN-SD, were examined to assess their performance for signal detection. The analysis of MSE, BER, capacity, and outage probability demonstrated that CNN-SD outperformed the other algorithms. Therefore, for future systems utilizing ultra-massive MIMO, CNN-SD emerges as the optimal choice for signal detection. In our future work, we aim to enhance the efficacy of our proposed method by employing a strategy that integrates score combination techniques and multi-scale neural networks, as well as exploring additional neural network architectures for performance optimization.

Author Contributions: Conceptualization, P.U. and A.I.; methodology, C.K., P.U., and A.I.; software, C.K.; supervision, P.U. and A.I.; validation, P.U. and A.I.; formal analysis, P.U. and A.I.; funding acquisition, P.U.; investigation, C.K.; project administration, P.U. and A.I.; resources, C.K., P.U., and A.I.; data curation, C.K.; writing—original draft preparation, C.K.; writing—review and editing, C.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Huo, Y.; Lin, X.; Di, B.; Zhang, H.; Hernando, F.J.L.; Tan, A.S.; Mumtaz, S.; Demir, Ö.T.; Chen-Hu, K. Technology Trends for Massive MIMO towards 6G. *arXiv* **2023**, arXiv:2301.01703.
- Wang, X.; Kong, L.; Kong, F.; Qiu, F.; Xia, M.; Arnon, S.; Chen, G. Millimeter wave communication: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1616–1653. [\[CrossRef\]](#)
- Zheng, Y.; Wang, C.X.; Yang, R.; Yu, L.; Lai, F.; Huang, J.; Feng, R.; Wang, C.; Li, C.; Zhong, Z. Ultra-massive MIMO channel measurements at 5.3 GHz and a general 6G channel model. *IEEE Trans. Veh. Technol.* **2022**, *72*, 20–34. [\[CrossRef\]](#)
- Dilli, R.; Chandra, R.; Jordhana, D. Ultra-Massive MIMO Technologies for 6G Wireless Networks. *Eng. Sci.* **2021**, *16*, 308–318. [\[CrossRef\]](#)
- Wang, C.X.; Wang, J.; Hu, S.; Jiang, Z.H.; Tao, J.; Yan, F. Key Technologies in 6G Terahertz Wireless Communication Systems: A Survey. *IEEE Veh. Technol. Mag.* **2021**, *16*, 27–37. [\[CrossRef\]](#)
- Faisal, A.; Sariahdeed, H.; Dahrouj, H.; Al-Naffouri, T.Y.; Alouini, M.S. Ultramassive MIMO Systems at Terahertz Bands: Prospects and Challenges. *IEEE Veh. Technol. Mag.* **2020**, *15*, 33–42. [\[CrossRef\]](#)
- Murshed, R.U.; Ashraf, Z.B.; Hridhon, A.H.; Munasinghe, K.; Jamalipour, A.; Hossain, M.F. A CNN-LSTM-based Fusion Separation Deep Neural Network for 6G Ultra-Massive MIMO Hybrid Beamforming. *IEEE Access* **2023**, *11*, 38614–38630. [\[CrossRef\]](#)
- Sariahdeed, H.; Alouini, M.S.; Al-Naffouri, T.Y. Terahertz-Band Ultra-Massive Spatial Modulation MIMO. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2040–2052. [\[CrossRef\]](#)
- Lee, Y.; Sou, S.I. On improving gauss-seidel iteration for signal detection in uplink multiuser massive MIMO systems. In Proceedings of the 2018 3rd International Conference on Computer and Communication Systems (ICCS), Nagoya, Japan, 27–30 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 268–272.
- Jiang, Y.; Varanasi, M.K.; Li, J. Performance analysis of ZF and MMSE equalizers for MIMO systems: An in-depth study of the high SNR regime. *IEEE Trans. Inf. Theory* **2011**, *57*, 2008–2026. [\[CrossRef\]](#)
- Larsson, E.G.; Edfors, O.; Tufvesson, F.; Marzetta, T.L. Massive MIMO for next generation wireless systems. *IEEE Commun. Mag.* **2014**, *52*, 186–195. [\[CrossRef\]](#)
- Chen, S.; Liang, Y.C.; Sun, S.; Kang, S.; Cheng, W.; Peng, M. Vision, requirements, and technology trend of 6G: How to tackle the challenges of system coverage, capacity, user data-rate and movement speed. *IEEE Wirel. Commun.* **2020**, *27*, 218–228. [\[CrossRef\]](#)
- Mai, Z.; Chen, Y.; Du, L. A Novel Blind mmWave Channel Estimation Algorithm Based on ML-ELM. *IEEE Commun. Lett.* **2021**, *25*, 1549–1553. [\[CrossRef\]](#)
- Keramidi, I.P.; Moscholios, I.D.; Sarigiannidis, P.G. Call Blocking Probabilities under a Probabilistic Bandwidth Reservation Policy in Mobile Hotspots. *Telecom* **2021**, *2*, 554–573. [\[CrossRef\]](#)
- Heath, R.W.; Gonzalez-Prelcic, N.; Rangan, S.; Roh, W.; Sayeed, A.M. An overview of signal processing techniques for millimeter wave MIMO systems. *IEEE J. Sel. Top. Signal Process.* **2016**, *10*, 436–453. [\[CrossRef\]](#)
- Nguyen, V.L.; Lin, P.C.; Cheng, B.C.; Hwang, R.H.; Lin, Y.D. Security and privacy for 6G: A survey on prospective technologies and challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2384–2428. [\[CrossRef\]](#)
- Gao, X.; Dai, L.; Yuen, C.; Zhang, Y. Low-complexity MMSE signal detection based on Richardson method for large-scale MIMO systems. In Proceedings of the 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall), Vancouver, BC, Canada, 14–17 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–5.
- Nakai-Kasai, A.; Wadayama, T. MMSE signal detection for MIMO systems based on ordinary differential equation. In Proceedings of the GLOBECOM 2022–2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 6176–6181.
- Jin, F.; Liu, Q.; Liu, H.; Wu, P. A Low Complexity Signal Detection Scheme Based on Improved Newton Iteration for Massive MIMO Systems. *IEEE Commun. Lett.* **2019**, *23*, 748–751. [\[CrossRef\]](#)
- Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
- Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2012**, *42*, 513–529. [\[CrossRef\]](#) [\[PubMed\]](#)
- Deng, W.; Zheng, Q.; Chen, L. Regularized Extreme Learning Machine. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; pp. 389–395. [\[CrossRef\]](#)

23. Zhang, K.; Luo, M. Outlier-robust extreme learning machine for regression problems. *Neurocomputing* **2015**, *151*, 1519–1527. [[CrossRef](#)]
24. Sareddeen, H.; Abdallah, A.; Mansour, M.M.; Alouini, M.S.; Al-Naffouri, T.Y. Terahertz-band MIMO-NOMA: Adaptive superposition coding and subspace detection. *IEEE Open J. Commun. Soc.* **2021**, *2*, 2628–2644. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



BIOGRAPHY

Mr. Chittapon Keawin was born in Nakhon Ratchasima, Thailand, in 1996. He graduated with a bachelor's degree in Telecommunication Engineering in 2019 from Rajamangala University of Technology Isan (RMUTI) and a master's degree in Telecommunication and Computer Engineering in 2021 from Suranaree University of Technology (SUT), respectively. Currently, he is pursuing his Ph.D. in Telecommunication and Computer Engineering at Suranaree University of Technology, Nakhon Ratchasima, Thailand. His research interests include wireless communication, Massive MIMO, Channel Estimation techniques, Signal Detection techniques, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL).

