THE INTERPLAY BETWEEN COMPUTATION AND COMMUNICATION

By

Ayoob Salari

A THESIS SUBMITTED IN FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN SCIENCE
AT
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING
THE UNIVERSITY OF SYDNEY

JUNE 2023

To my brother Reza

who always thought my Ph.D. was just a fancy way to

avoid getting a real job.

R.I.P

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my two exceptional supervisors for their unwavering support throughout my PhD journey. Firstly, I would like to acknowledge and express my heartfelt appreciation to Dr. Mahyar Shirvanimoghaddam for his invaluable assistance in determining and tackling research challenges. His constructive feedback throughout our meetings has significantly enhanced the calibre of my research and served as a valuable resource in overcoming research obstacles. Throughout the Covid-19 lockdown, he was a constant source of motivation and inspiration for me due to his upbeat mentality. Next, I would like to acknowledge and express my utmost appreciation to Professor Sarah Johnson from The University of Newcastle. The feedback she provided me with during our meetings proved invaluable in streamlining my thoughts and enhancing my research abilities.

I would like to extend my sincere thanks to Dr. Reza Arablouei and Dr. Basit Shahab for their valuable input and constructive feedback during our meetings. Their guidance has greatly influenced the quality and direction of this thesis.

I would like to express my great gratitude to Prof. Yonghui Li for his invaluable feedback throughout the course of this thesis. His insightful comments and suggestions have greatly enhanced the quality of my research.

I would like to thank my friends Dr. Ahmadreza Ahmadi, Mrs. Hengameh Zabihi, Dr. Bahman Lahoorpoor, Dr. Sepideh Taheri, Dr. Keyvan Agha Balaei, Dr. Mehdi Abdolvand, Mr. Hooman Hashemi zadeh, Mr. Taher Tahmasbi, and Mrs. Azadeh Rastgar for their unwavering support throughout the process of completing this thesis. Their encouragement, understanding, and belief in my abilities have been a constant source of motivation. Their willingness to listen and offer assistance whenever needed have been truly invaluable. Their friendship and presence have made this journey more enjoyable and memorable. I am fortunate to have such amazing friends who have stood by me, and I sincerely thank them for their unwavering support.

I would like to thank everyone in the Centre for IoT and Telecommunications for their support during the course of my research.

I would like to express my sincere appreciation for the profound love and encouragement of my parents Rahmatallah Salari and Eshrat Sharifi, and my sisters Shiva, Mina, Lia, Fatemeh, Sima, and Zahra, their unwavering love, support, and encouragement throughout my academic journey. Their constant encouragement, understanding, and sacrifices have been the pillars of my success. I am truly blessed to have such a loving and supportive family, and I extend my heartfelt thanks to them for their endless love and support.

Sydney, Australia Ayoob Salari
June, 2023

Statement of Originality

I hereby certify that the work embodied in the thesis is my own work, conducted under normal supervision. The thesis contains no material which has been accepted, or is being examined, for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made. I give consent to the final version of my thesis being made available worldwide when deposited in the University's Digital Repository, subject to the provisions of the Copyright Act 1968 and any approved embargo.

Ayoob Salari
School of Electrical and Information Engineering
The University of Sydney
30th June, 2023

Acknowledgment OF Authorship

I hereby certify that the work embodied in this thesis contains published papers/ scholarly work of which I am a joint author. I have included as part of the thesis a written declaration endorsed in writing by my supervisor, attesting to my contribution to the joint publications/scholarly work.

Chapter III of this thesis is based on materials published in the conference paper (C1) and the journal paper (J2) as listed in Section 1.3. My role involved conducting all simulation experiments, creating the analytical framework, interpreting the results, and drafting the papers.

Chapter IV of this thesis contains materials from the published journal paper (J3). My work in this chapter included conducting all simulation experiments, creating the analytical framework, interpreting the results, and writing the paper.

Chapter V draws upon content from conference paper (C2) and published journals (J1) and (J4) as detailed in Section 1.3. I conducted all simulations, developed the analytical framework, interpreted results, and authored (C2) and (J1). In (J4), I contributed to the introduction, conducted a simulation experiment, and contributed to the theoretical analysis.

Ayoob Salari
30th June, 2023

By signing below I confirm that Ayoob Salari contributed to all publications embodied in this thesis.

Name of Supervisor: Dr. Mahyar Shirvanimoghaddam
Signature of Supervisor:
30th June, 2023

Abstract

In this thesis, a comprehensive exploration into the integration of communication and learning within the massive Internet of Things (mIoT) is undertaken. Addressing one of the fundamental challenges of mIoT, where traditional channel estimation methods prove inefficient due to high device density and short packets; initially, a novel approach leveraging unsupervised machine learning for joint channel estimation and signal detection is proposed. This technique utilizes the Gaussian mixture model (GMM) clustering of received signals, thereby reducing the necessity for exhaustive channel estimation, decreasing the number of required pilot symbols, and enhancing symbol error rate (SER) performance. Building on this foundation, an innovative method is proposed that eliminates the need for pilot symbols entirely. By coupling GMM clustering with rotational invariant (RI) coding, the model maintains robust performance against the effects of channel rotation, thereby improving the efficiency of mIoT systems.

This research delves further into integrating communication and learning in mIoT, specifically focusing on federated learning (FL) convergence under error-prone conditions. It carefully analyzes the impact of factors like block length, coding rate, and signal-to-noise ratio on FL's accuracy and convergence. A novel approach is proposed to address communication error challenges, where the base station (BS) uses memory to cache key parameters.

Closing the thesis, an extensive simulation of a real-world mIoT system, integrating previously developed techniques, such as the innovative channel estimation method, RI coding, and the introduced FL model. It notably demonstrates that optimal learning outcomes can be achieved even without stringent communication reliability. Thus, this work not only achieves comparable or superior performance to traditional methods with fewer pilot symbols but also provides valuable insights for optimizing mIoT systems within the FL framework.

Table of Contents

List of Figures

List of Acronyms

1G	First generation
2G	Second generation
3G	Third generation
4G	Fourth generation
5G	Fifth generation
6G	Sixth generation
AI	Artificial Intelligence
ANN	Artificial neural networks
ARQ	Automatic repeat request
AWGN	Additive white Gaussian noise
BDM	Bit division multiplexing
BER	Bit error rate
BS	Base station
CDF	Cumulative distribution function
CGF	Cumulant generating function
CLT	Central limit theorem
CNN	Convolutional neural network
CPU	Central processing units
CSI	Channel state information
dB	Decibel
DNN	Deep neural network

EM	Expectation maximization
eMBB	Enhanced mobile broadband
EVD	Eigenvalue decomposition
EWMA	Exponentially weighted moving average
FBL	Finite blocklength
FL	Federated learning
GD	Gradient descent
GMM	Gaussian mixture model
GPU	Graphics processing units
HD	High-definition
HARQ	Hybrid automatic repeat request
IDMA	Interleave-division multiple access
IPE	Integrated pilot eigenvalue decomposition
i.i.d.	Independent and identically distributed
IoT	Internet of Things
IP	Internet protocol
IPCE	Invariant parity check equations
LDPC	Low density parity check code
LDS	Low-density spreading
LLR	Log-likelihood ratio
MA	Multiple access
MGF	Moment generating function
mIoT	Massive internet of things
ML	Machine learning
MLE	Maximum-likelihood estimator
MLD	Maximum-likelihood detection
MMSE	Minimum Mean Square Error
mMTC	Massive machine-type communication
MUSA	Multi-user shared access

MUSIC	Multiple signal classification
NOMA	Non-orthogonal multiple access
OFDM	Orthogonal frequency-division multiplexing
OMA	Orthogonal multiple access
PDF	Probability density function
PDMA	Pattern division multiple access
PEC	Packet erasure channel
PMF	Probability mass function
QAM	Quadrature amplitude modulation
QoS	Quality of service
QPSK	Quadrature phase shift keying
RB	Resource block
RI	Rotational invariant
SAGE	Space alternating generalized expectation
SCMA	Sparse code multiple access
SER	Symbol error rate
SIC	Successive interference cancellation
SINR	signal-to-interference-plus-noise ratio
SNR	Signal-to-noise ratio
SVD	Singular value decomposition
UL	Uplink
URLLC	Ultra-reliable and low-latency communications
VNA	vector network analyser

List of Symbols and Notations

X	a random variable
$[X]_u^v$	a sequence of random variables $[X]_u^v = [X_u, X_{u+1}, \ldots, X_v]$
x	values of scalar variables or the sample of random variables
$[x]_u^v$	a sequence of scalar variables or samples $[x]_u^v = [x_u, \ldots, x_v]$
$f_X(x)$	the probability density function (PDF) of a continuous X
$f_X(x\|Z = z)$	the PDF of a continuous X conditioning on $\{Z = z\}$
$\mathbb{E}[X]$	the mean of X
$\mathbb{E}[X\|Z{=}z]$	the mean of X conditioning on $\{Z = z\}$
σ_X^2	the variance of X
$\sigma_{X\|Z=z}^2$	the variance of X conditioning on $\{Z = z\}$
$\mathbf{A}$	a matrix
$\mathbf{A}^{\mathrm{T}}$	the transposition of matrix $\mathbf{A}$
$\mathbf{a}$	a row vector
$\mathcal{A}$	a set
$\mathcal{O}$	Order of complexity
$\mathcal{N}(\cdot,\cdot)$	the Gaussian distribution
N_u	number of users
N_c	number of clusters
N_t	number of time slots
N_s	number of symbols
N_p	number of power levels

N_r	number of repeated packets
M_o	modulation order
w	AWGN noise
w_c	weight of GMM cluster
ω	update parameter in FL
k	message length
n	codeword length
$\mathcal{S}$	signal constellation
Bern	Bernoulli distribution
ℓ_0	reference path-loss
r_0	reference distance
χ	large-scale shadowing
Q	tail distribution function of the standard normal distribution

Chapter 1

Introduction

The evolution of mobile communication networks began in the 1980s with the introduction of first-generation (1G) voice-only analogue technology. Nevertheless, the limitations of analogue systems led to the development of the second-generation (2G) digital network. With the advent of 2G, data applications such as short message service were supported, paving the way for future developments. In the 1990s, third-generation (3G) mobile broadband services emerged, enabling novel applications, including multimedia messaging services, video communications, and mobile television. The fourth-generation (4G) mobile broadband services experienced substantial improvements, notably in the areas of high-definition (HD) video streaming and seamless handover capabilities. As a result of the introduction of internet protocol (IP) communication, various quality of service (QoS) levels were developed to meet the varying needs of users [1, 2].

As digital transformation continues to reshape the world, the fifth-generation (5G) mobile network technology is a linchpin of this development. 5G represents a significant improvement over all previous generations of mobile networks, introducing three distinct end-user services. Enhanced mobile broadband (eMBB) is one of these services; it offers high-speed internet connectivity, increased bandwidth, moderate latency, ultra HD streaming videos, virtual reality, and augmented reality media, among other features. In addition, 5G supports massive machine-type communication (mMTC), which provides low-cost and energy-efficient long-distance and broadband communication. Furthermore, ultra-reliable low latency communication (URLLC) provides ultra-low latency, ultra-high reliability, and diverse QoS, which was not possible with conventional mobile network architecture. URLLC is intended for real-time interaction-on-demand, including remote control, vehicle-to-vehicle communication, industry 4.0, smart networks, intelligent transport systems, and other advanced applications. The capabilities of 5G technology will revolutionise many facets of our lives and have a significant impact on society as a whole [3].

mMTC addresses the unique requirements of the massive internet of things (mIoT) by providing increased device density [4], energy efficiency [5], low latency [6], enhanced coverage [7], network slicing [8], and scalability [9]. In a mIoT network, it is possible to have over a million devices per square kilometre, resulting in a massive amount of data to analyse and a significant quantity of information dispersed across numerous devices [10]. The scale of this data is staggering, with mobile traffic projected to skyrocket from 62 exabytes per month in 2020 to over 5000 exabytes per month by 2030 [11]. It is anticipated that 5G will be fully deployed by the end of 2023, but it may not be able to satisfy the network requirements associated with a projected rise in the number of connected devices in mIoT networks [12]. Particularly, 5G will struggle to meet the requirements of data- and computation-intensive mIoT applications [13].

In a mIoT ecosystem, a distinctive characteristic is the sheer number of users.

Typically, these users operate under low-power conditions which inherently restrict their memory, computational capabilities, and communication potential. This reality poses a significant challenge to the stability and reliability of the communication link as establishing such a link often demands high radio resources, considerable power, and extensive processing. Consequently, the limitations of the mIoT network may lead to an inability for the base station (BS) to consistently maintain and estimate the channel with numerous users, impacting the quality of communication. This forms a fundamental paradox between computation and communication in mIoT networks. This paradox comes from the fact that the core objective of an IoT system involves the distribution of numerous sensors, each of which collects and transmits data. However, the feasibility of transmitting this data directly is constrained due to IoT's inherent limitations. An alternative approach entails the application of local learning algorithms before transmitting the data. Hence, the implementation of a mIoT network necessitates the exploration of effective communication strategies between the numerous users and the BS. This becomes a critical aspect, as even with distributed learning, there will still be multiple communication rounds between users and the BS. Moreover, it is important to devise communication strategies that are not only effective but also resource-efficient. In particular, the goal is to avoid the conventional process of sending pilots to a massive number of users and then estimating the channel. A more efficient approach would involve extracting channel information directly from the received signal.

In the context of the escalating data volume and complexity within mIoT networks, the traditional approaches to storing, analyzing, and visualizing such data are increasingly challenged [14]. Artificial intelligence (AI), with its promising capabilities, provides compelling solutions to these challenges [15]. The incorporation of AI into mIoT is not merely a general statement; instead, it is a concrete strategy designed to address specific problems [16]. One such issue, prevalent in the mIoT environment, is the problem of massive communication. The challenge of managing communication between the BS and a large number of users can potentially be mitigated using AI-assisted communication techniques [17]. This involves eliminating the need to transmit pilot signals by instead using received signals as a dataset. Applying clustering techniques, one can learn about the channel directly from this data. Such an approach can greatly enhance the efficiency of communication in a mIoT ecosystem. The AI-assisted communication approach, as described above, holds the potential to significantly enhance mIoT networks. By improving areas such as channel estimation and signal detection, AI-assisted communication not only promises a higher throughput but also accommodates a larger number of connections.

Having proposed an AI-assisted communication strategy to manage the massive communication challenges in mIoT networks, it is equally important to address the issue of managing the substantial amount of data transmitted by the multitude of users. For this purpose, the concept of communication-assisted AI is brought into play. Communication-assisted AI exploits the capabilities of communication networks to support distributed learning and holds great potential to significantly impact the future of mIoT networks by addressing their inherent challenges in a more intelligent and efficient manner [18].

This thesis aims to contribute to the development of ubiquitous AI in mIoT networks, particularly through enhancing the communication and learning processes inherent in these complex networks. The mIoT environment is characterized by an enormous number of devices, each transmitting vast quantities of data. Consequently, managing this massive communication in an efficient manner poses a significant challenge, one that I propose to address using AI-assisted communication techniques. Moreover, the primary aim of IoT networks is not merely to capture the individual data from each user. The true power of IoT lies in harnessing data-driven insights for decision-making or understanding certain phenomena. Hence, the application of distributed learning approaches can serve to alleviate the communication burden by effectively distilling critical insights from the massive data generated in these networks. My research in this thesis focuses on exploring the interplay between communication and computation. The aim is to understand to what extent I can relax the communication requirements while still achieving a similar level of performance. Therefore, this thesis delves into these two critical facets of mIoT networks - massive communication and distributed learning - with the goal to identify and leverage synergies between them. This dual approach is instrumental in contributing to the development of efficient and intelligent solutions tailored for the next generation of wireless IoT networks.

In the first part of this thesis, I will tackle the challenge of channel estimation and signal detection in mIoT, viewing it as a data analysis problem that I aim to address. Implementing mIoT is problematic due to a lack of available channels. Existing multiple access (MA) methods, such as orthogonal multiple access (OMA), assign radio resources non-overlappingly, allowing only one device to use them [19, 20].

With numerous devices and limited radio resources, OMA-based resource allocation poses a performance barrier. Non-orthogonal multiple access (NOMA) was introduced in the context of 5G networks with the aim of facilitating communication between a large number of devices and the BS [21]. NOMA allows for users to transmit simultaneously. However, to successfully detect the users, a joint channel estimation and signal detection is necessary. The complexity of NOMA demands scheduling and user pairing, but the overarching objective is to establish a network where IoT users can transmit without the need for rigorous scheduling.

I will focus on AI-assisted communication, which involves the utilization of machine learning (ML) algorithms to improve signal processing performance, including channel estimation and signal detection, as well as signal encoding and decoding [22]. Conventional communication techniques rely on the assumption that the transmitter has access to the channel state information (CSI) and employs lengthy block-length codes to approach Shannon's capacity limit. To obtain reliable CSI, various channel estimation techniques have been proposed, including blind, semi-blind, and training-based approaches. Among these, training-based channel estimation utilizing extended pilot symbols has proven to be a reliable strategy for accurately estimating the channel [23]. In contrast, blind channel estimation utilizes the inherent properties of the transmitted signal to estimate the channel without requiring a training symbol, albeit at the cost of increased complexity and lower performance [24]. Semi-blind channel estimation schemes strike a balance between throughput, accuracy, and complexity by combining the advantages of both training-based and blind schemes [25]. In mIoT systems, even a small number of training symbols can lead to substantial performance degradation due to the limited size of the transmitted packets. To overcome the challenges posed by the short packet size in mIoT systems, there is a need to develop ML algorithms that are well-suited for such applications.

The utilization of ML exhibits the capability to provide efficient and effective services throughout different layers of networks. Unsupervised learning techniques, specifically clustering algorithms, can be leveraged in the physical layer to accomplish tasks like channel estimation and signal detection [26]. The implementation of these algorithms facilitates the system in attaining higher levels of accuracy and reliability while concurrently reducing the necessity for lengthy pilot symbols. K-means and Gaussian mixture models (GMM) are the two main clustering techniques [27]. K-means groups data points based on their distance from the cluster center. It ultimately implies that the closest two data elements are, the more similar they are. GMM employs a probabilistic clustering of data points. Each cluster is uniquely characterized by a Gaussian distribution [28]. I assume that the noise follows a Gaussian distribution, resulting in the received signal exhibiting characteristics of a Gaussian mixture. Under these conditions, the application of clustering techniques like GMM can be particularly effective. The utilization of clustering techniques in conjunction with NOMA eliminates the requirement for extended pilot symbols for precise channel estimation and enhances the throughput. Additionally, it facilitates the transmission of data by multiple users concurrently over a shared radio resource. Thus, it is an ideal choice for mIoT applications [29].

While clustering algorithms can be effectively used to estimate the channel in mIoT networks, a challenge remains. Even with these advanced techniques, the necessity for transmitting a limited number of pilot symbols persists. This requirement poses a constraint on the system, introducing additional complexity and potential inefficiency. This is particularly the case in massive mIoT networks, where every improvement in resource utilization can have a significant impact on overall system performance. Moreover, despite their utility, clustering techniques such as K-means and GMM do not inherently support the elimination of all pilot symbols. This is an important issue since the transmission of pilot symbols can consume valuable radio resources and limit the capacity for concurrent data transmission by multiple users. Therefore, to tackle this issue and further optimize communication efficiency in mIoT networks, a different approach is required. An approach that can bypass the need for transmitting pilot symbols while maintaining or potentially enhancing the accuracy and reliability in data transmission.

In order to further enhance the throughput of mIoT, it is possible to utilize the rotational invariant (RI) code alongside the GMM. In the RI codes, it is proven that any modulated codeword that has undergone rotation is deemed a valid codeword sequence. Furthermore, it is observed that all rotated versions of a given sequence are generated by the same information input and can be decoded to yield the same information output. The RI code is a convolutional code with non-linear characteristics, which can be efficiently decoded through the utilization of the Viterbi decoder over the code trellis. By doing so, it is possible to eliminate the necessity of utilizing any pilot symbol [30, 31].

In the second part of my thesis, I will focus on critical steps to facilitate and enhance learning within these mIoT networks. The distinctive obstacles presented by mIoT networks necessitate not only effective communication but also advanced learning mechanisms to optimize the utilization of the vast quantities of information generated by these numerous devices. The conventional approach of learning, wherein users transmit their localized data to a BS equipped with processing capabilities for the purpose of implementing the learning model, is not well-suited for mIoT. This is due to limitations in privacy and power, as well as inadequate communication bandwidth. These issues pose a significant burden on both the communication links and the BS, highlighting the need for new approaches to address this challenge of mIoT [32]. The emerging technology of communication-assisted AI, specifically edge learning, offers computation services at the network edge [33]. This results in reduced energy consumption and latency while also alleviating the burden on communication links.

Edge learning refers to two forms of learning based on where data processing takes place: centralized edge learning and distributed edge learning. In centralized edge learning, the AI model is trained by transferring private raw data from edge devices to a central server. However, this approach raises concerns about preserving user data privacy. While the power of computing chips like central processing units (CPUs) and graphics processing units (GPUs) have improved over time, leading to the development of specialized AI chipsets, distributed edge learning has become more prevalent [34]. In the decentralized mode, individual computing nodes engage in the training of their own model using locally sourced data, thereby ensuring the preservation of private information at the local level.

Federated learning (FL) is an example of distributed edge learning that uses the distributed computing resources available at the network edge. In the FL framework, data privacy is prioritized by sharing less sensitive updates, such as gradients or model updates, rather than raw data. This approach avoids the need to upload private raw data to a central server, which can raise privacy concerns. FL involves a process where individual users train their own localized models using their own data and the global parameter. Once training is complete, the updated model parameters are transmitted to the BS. The BS then adjusts the global model parameter through a weighted average of updates from users and broadcasts the updated global parameters throughout the network. This process is repeated iteratively until convergence is achieved [35, 36]. FL intends to train the model on a large number of learners, taking into account communication limitations imposed by bandwidth, power, privacy, and security [37].

Having established the importance of both efficient communication and advanced learning in the mIoT, it is noteworthy to consider the inherent trade-off between these two objectives. Specifically, while the absolute reliability of communication is a priority in certain scenarios when the goal is learning, one can afford to relax this stringency to a certain extent. Learning through FL usually consists of iterative processes, where the accuracy of outcomes improves across multiple iterations. Given this, the need for perfect reliability in every single communication exchange can be mitigated. Instead, even if errors occur during the initial communication, the iterative nature of the learning process ensures that the accuracy of the output model gradually enhances over time, ultimately leading to optimal learning outcomes [38].

In massive IoT, the primary focus is not on individual user data but rather on the application of acquired knowledge. Nevertheless, due to the large number of users, there remains a significant level of communication between users and the BS, which prompts the exploration of using FL techniques. In resource-constrained environments, I employ NOMA integrated with GMM clustering to manage massive communication. Although NOMA is prone to errors in short-packet communication, I will demonstrate that FL can effectively mitigate these issues. My approach allows for FL in mIoT, enables non-orthogonal resource block allocation, and guarantees convergence.

This thesis examines the interplay between communication and learning within mIoT networks. By relaxing the strict requirements for communication, I propose to use the GMM clustering method for channel estimation and signal detection and utilize FL for learning. By bridging communication and learning in this way, I aim to create a more balanced and efficient framework for mIoT networks, one that capitalizes on the strengths of both aspects while minimizing their respective challenges. This sets the stage for the core investigation that will unfold in the body of this thesis.

1.1 Research Problems and Contributions

Most existing research on mIoT focuses on long-packet communication and assumes that the BS has sufficient resources to meet the needs of mIoT users, while users can transmit all their data regardless of their limited resources. In this thesis, I demonstrate that because mIoT packets are short, traditional communication methods are unsuitable for mIoT. Instead, I propose an effective solution for mIoT channel estimation and signal detection based on clustering techniques. I also address the second outstanding challenge in the learning process over mIoT networks, which assumes perfect communication between users and the BS, and that users have sufficient resources to transmit all their data to the BS. My proposed solution is based on FL and can handle erroneous communication between users and the BS, which can be a significant challenge in mIoT scenarios. Throughout this dissertation, I aim to address the following unresolved research challenges. In addition, contributions will be summarized for each research problem.

1.1.1 What novel techniques can be employed to enhance channel estimation and signal detection in mIoT?

The sheer number of users in a mIoT environment, coupled with the constraint of short packet transmissions, significantly challenges the traditional approach to channel estimation. This conventional method relies heavily on extended pilot symbols for precise channel estimation, which becomes infeasible in mIoT applications where every bit of resource matters. The use of even a few pilot symbols in this context can incur substantial efficiency loss due to the large-scale and short-packet nature of the network.

Addressing this problem necessitates a resource-conscious strategy. One such method is the application of NOMA, an effective technique for handling large-scale networking. The superposition of signals enables concurrent transmission of data by multiple users over a shared radio resource block (RB). Successive interference cancellation (SIC) is a technique employed in the receiver for separating the signal of individual users [39]. The implementation of NOMA can be achieved through either power-domain or code-domain techniques among users [40]. The analysis of NOMA is primarily conducted in the asymptotic blocklength regime and, more recently, in the finite blocklength (FBL) regime. The research has demonstrated that NOMA is an effective means of achieving superior resource utilization and energy efficiency[41–43]. The NOMA technology facilitates the provision of services to a greater number of users despite the constraints of limited channel resources, thereby enabling the establishment of massive connectivity over limited radio resources [44].

In the first contribution of the thesis, I present an innovative methodology for joint channel estimation and signal detection by utilizing clustering algorithms. The proposed methodology involves grouping incoming signals into distinct clusters based on their modulation characteristics, which consolidates signals that are influenced by both minor and significant noise and interference into a unified cluster. Consequently, the need for precise estimation of channel rotation is mitigated, and a coarse approximation of channel rotation suffices, leading to a notable reduction in the number of pilot symbols required for channel estimation. Moreover, I provide a mathematical analysis to characterize the performance of the clustering algorithm, which includes the error rate. I demonstrate the convergence of the algorithm and establish an upper limit on its margin of error. Additionally, I compute a boundary for the estimation error of the channel, taking into account both gain and phase errors. This boundary is utilized to obtain a mathematical expression for the symbol error rate (SER). I evaluate the proposed clustering-based methodology against existing techniques that use either training-based or semi-blind channel estimation. I also examine the practical implementation of the suggested algorithm for joint channel estimation and signal detection in a grant-free NOMA scenario, which is advantageous for mIoT applications due to the unpredictability of mIoT users' behavior.

1.1.2 Could the incorporation of channel coding strategies into clustering techniques serve to further enhance the performance of channel estimation and signal detection in mIoT networks?

While clustering algorithms have shown significant promise in facilitating channel estimation within mIoT networks, they currently require the transmission of a minimal number of pilot symbols. The ideal strategy, however, would entirely eliminate the need for transmitting these pilot symbols, whilst simultaneously maintaining or potentially enhancing the level of precision and reliability in data transmission.

By introducing RI coding at the transmitter, I can improve the joint channel estimation and signal detection technique based on the clustering algorithm even more. This method compensates for channel phase rotation without needing any pilot symbols for channel estimation, increasing the algorithm's efficiency.

RI codes exhibit an important property wherein all modulated codewords are valid regardless of their phase rotation. This property ensures that all rotated versions of the same sequence decode back to the same information input. RI codes are non-linear convolutional codes that can be decoded effectively using the Viterbi decoder over the code trellis. I also investigate the use of low-density parity check (LDPC) codes as a channel coding technique. To enable demapping without pilot symbols, I explore all possible channel rotations and compute the syndrome for each phase. The phase with the highest number of checked syndromes is then selected as the channel phase. Through extensive simulations, I demonstrate that the performance of RI codes is superior to that of LDPC codes. In order to assess the effectiveness of the proposed scheme, I also conducted a comparison with the maximum-likelihood detector (MLD), which employs a small number of pilot symbols to estimate the channel. My simulation results demonstrate that the MLD requires approximately 8 pilot symbols to achieve comparable bit error rate (BER) performance as the proposed scheme, which operates without any pilot symbols. This implies that the MLD would experience a reduction in throughput of approximately 16% under the same target BER, assuming a user packet size of 50 bits and the use of a rate-1/2 code with a 4-ary modulation.

1.1.3 How can the accuracy of federated learning be maintained when faced with error-prone communication channels between users and BS in a mIoT system?

I investigate the feasibility of using distributed learning to train deep network models in a mIoT setup where users transmit their data to a BS with computational capabilities. However, given concerns about privacy, limited power, and inadequate communication bandwidth in mIoT scenarios, this approach is often impractical and can place a significant strain on the communication links. To overcome these challenges, I propose the use of FL methods that allow for the training of ML models without the need to collect raw data from users, thereby preserving their privacy.

I analyze the impact of communication errors on FL in a practical scenario. Specifically, I consider a scenario where the BS may not always successfully receive local updates from the devices due to wireless channel unreliability. To address this challenge, I propose two cases of FL: one where the BS has memory and one where it does not. In each case, I use the saddle-point approximation to determine the probability distribution of the global update in each communication round. My analysis reveals that the expected total loss of the FL will converge in the case where the BS does not have memory. Furthermore, I prove that FL converges to the global minimum of the loss function when the BS possesses memory to store past updates. These results provide insights into the robustness of FL in the presence of communication errors, which is critical for its practical implementation in real-world scenarios.

I examine the impact of having memory at the BS on the performance of FL in a wireless network. Specifically, I consider two different approaches: one where the BS stores each user's prior local update, and another where the BS stores global updates. I analyze how the blocklength, coding rate, and signal-to-noise ratio (SNR) impact the convergence of FL in each approach. My results demonstrate that even a single memory unit at the BS can have a significant effect on the system's performance. I also show that simply increasing the number of communication rounds or decreasing the coding rate may not necessarily lead to the optimal loss function for a given training duration. Instead, careful selection of the coding rate is necessary to minimize the loss for each SNR regime.

I analyze the performance of FL under communication errors in two practical scenarios. Firstly, I investigate a mIoT configuration based on real-world measurements [45], where 196 IoT devices transmit short packets to a BS. I calculate the packet error rate as a function of block-length, coding rate, and SNR using the normal approximation method [46]. Furthermore, I examine the time required for FL convergence under different rates and SNRs. Secondly, I evaluate a scenario where IoT devices transmit long packets to train neural networks using the MNIST digits dataset [47]. I assess the impact of instantaneous received SNR on model accuracy and explore the effect of rate and SNR on the system's performance.

1.2 Thesis Outline

The remainder of this thesis is organized as follows.

In Chapter 2, the literature review focuses firstly on the advancement of channel estimation and signal detection through the utilization of AI-assisted communication and provides an overview of various unsupervised learning clustering techniques. Specifically, the GMM clustering methodology will be presented. Chapter 2 then delves into the topics of distributed learning and their application for mIoT. Finally, preliminary information is presented regarding the FL model for mIoT.

In Chapter 3, I propose a new approach for mIoT uplink with NOMA that jointly estimates the channel and detects the signal using unsupervised ML. I apply a GMM to cluster received signals and optimize decision regions to enhance SER performance. My proposed clustering-based approach achieves an SER performance comparable to that of the conventional MLD with full CSI, when the received powers of users differ significantly. I analyze the accuracy tradeoff of my approach and the blocklength, as clustering algorithm accuracy depends on the number of available symbols at the receiver. I provide a comprehensive performance analysis of my approach and derive a theoretical bound on its SER performance. Simulation results confirm the effectiveness of my approach, and the calculated theoretical bound accurately predicts its SER performance. I extend my study to a practical grant-free NOMA scenario and show that my approach's performance is nearly equivalent to the optimal MLD with full CSI, which typically requires long pilot sequences.

In Chapter 4, the focus is on the uplink of mIoT with power-domain NOMA. Utilizing a clustering method, I propose an algorithm that performs both detection and estimation without the use of pilot symbols. Specifically, RI coding is employed to aid signal detection at the receiver without sending pilot symbols. The GMM is utilized to cluster the received signals automatically without supervision and to optimize decision boundaries, thereby improving the BER performance. Simulation results indicate that the proposed scheme achieves nearly the same BER performance as that of the conventional maximum likelihood receiver with full channel state information, even without the use of pilot symbols.

Chapter 5 presents an FL problem over a wireless channel, which takes into account coding rate and packet transmission errors. Communication channels are modeled as packet erasure channels (PEC), where the probability of erasure depends on the block length, code rate, and SNR. Despite the fluctuations in the instantaneous loss of FL, I prove that the expectation of loss converges, even in the presence of packet erasure. To enhance the performance of FL in the presence of packet erasure, I propose a paradigm where the BS utilizes memory. Specifically, I suggest two schemes in which the BS retains either the most recent local updates or the most recent global parameters in case of packet erasure. I investigate the effect of coding rate, SNR, and BS memory on the convergence of FL. I examine a realistic scenario of a massive IoT with error-prone transmissions, both in short- and long-packet communications. The simulation results demonstrate that even a single memory unit has a significant impact on the efficiency of FL in erroneous communication.

Chapter 6 is the culmination of the thesis, synthesizing research from prior chapters into a broad exploration of a mIoT system. Here, I leverage the GMM clustering technique for joint channel estimation and signal detection while employing diverse channel coding. The chapter emphasizes the delicate balance between learning and communication reliability within the FL framework. It demonstrates that one can achieve optimal learning results even when communication reliability is somewhat relaxed. Additionally, Chapter 6 introduces a mathematical framework to evaluate the error probability of a mIoT system. This enables us to estimate the learning performance of the mIoT system under the FL model, providing a fresh perspective on optimizing mIoT operations within the FL context.

Chapter 7 serves as the conclusion of this thesis, summarizing the major contributions and identifying potential areas for future research.

1.3 Related Publications

The following is a list of publications in refereed journals and conference proceedings produced during my Ph.D. candidature.

Journal Papers

(J1) A. Salari, S. Johnson, B. Vucetic, and M. Shirvanimoghaddam, "Rate- Convergence Tradeoff of Federated Learning over Wireless Channel," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2023.3304327, 2023.

(J2) A. Salari, M. Shirvanimoghaddam, M. B. Shahab, R. Arablouei, and S. Johnson, "Design and Analysis of Clustering-Based Joint Channel Estimation and Signal Detection for NOMA," *IEEE Transaction on Vehicular Technology*, DOI: 10.1109/TVT.2023.3313650, 2023.

(J3) A. Salari, M. Shirvanimoghaddam, M. B. Shahab, Y. Li and S. Johnson, "NOMA Joint Channel Estimation and Signal Detection Using Rotational Invariant Codes and GMM-Based Clustering," *IEEE Communications Letters*, vol. 26, no. 10, pp. 2485-2489, Oct. 2022

(J4) M. Shirvanimoghaddam, A. Salari, Y. Gao and A. Guha, ”Federated Learning With Erroneous Communication Links,” *IEEE Communications Letters*, vol. 26, no. 6, pp. 1293-1297, June 2022.

Conference Papers

(C1) A. Salari, M. Shirvanimoghaddam, M. B. Shahab, R. Arablouei and S. Johnson, ”Clustering-based Joint Channel Estimation and Signal Detection for Grant-free NOMA,” *IEEE Globecom Workshops*, pp. 1-6, 2020.

(C2) A. Salari, M. Shirvanimoghaddam, B. Vucetic, S. Johnson, “Design and analysis of random-access scheme for clustering-based detectors in grant-free NOMA,” *Asilomar Conference on Signals, Systems, and Computers*, Oct. 2023.

Chapter 2

Background

This chapter provides an overview and background on the interplay between learning and communication. I begin by elucidating the concept of NOMA and its relevance in the context of mIoT applications. I explain how NOMA can address the unique challenges posed by mIoT deployments, providing efficient connectivity for a multitude of devices. Moving forward, I dive into a detailed examination of various methods employed for channel estimation in conventional communication systems. Evaluating these methods, I establish the foundation for introducing the GMM clustering algorithm that holds promise for accurate channel estimation. Lastly, I delve into FL as a framework for integrating learning into communication systems, outlining the fundamental components and highlighting its potential for enhancing learning performance.

2.1 Non-Orthogonal Multiple Access

One of the main challenges for mIoT applications is that the number of devices is usually large while the available spectrum is limited. To tackle this, NOMA has emerged as a promising technology that allows multiple users to simultaneously transmit their data over the same radio resource [48–50]. Two main categories of NOMA are power-domain and code-domain [51]. While in power-domain NOMA, users transmit with different power levels depending on their channel conditions, code-domain NOMA relies on assigning unique codes to the users. Some prominent code-domain NOMA techniques are sparse code multiple access (SCMA) [52], multi-user shared access (MUSA) [53], interleave-division multiple access (IDMA) [54], and low-density spreading (LDS) [55]. There are also other NOMA techniques, such as bit division multiplexing (BDM) [56] and pattern division multiple access (PDMA) [57]. Compared to code-domain NOMA, power-domain NOMA achieves higher spectral efficiency without the need for significant alterations to existing communication networks, making it a more feasible solution within the same bandwidth [58].

In power-domain NOMA, power allocation is performed based on the channel conditions of the users. Assuming two users are transmitting to BS, user 1 and user 2, their power allocations can be denoted as P_1 and P_2, respectively. The received signal at the BS for the NOMA system can be modeled as follows:

$$y = \sqrt{P_1} h_1 x_1 + \sqrt{P_2} h_2 x_2 + n, \tag{2.1.1}$$

Where y is the received signal at the BS, h_1 and h_2 are the channel gains and x_1 and x_2 represent the information symbols for user 1 and user 2, respectively, and n represents the additive white Gaussian noise (AWGN). As is shown in Fig. 2.1, at the receiver, SIC is employed to decode the signals of different users. Initially, the receiver estimates the channel of the user with the stronger channel condition, decodes its signal, and then subtracts it from the received signal. This allows the receiver to estimate the channel of the user with the weaker channel condition and subsequently decode its signal [59].

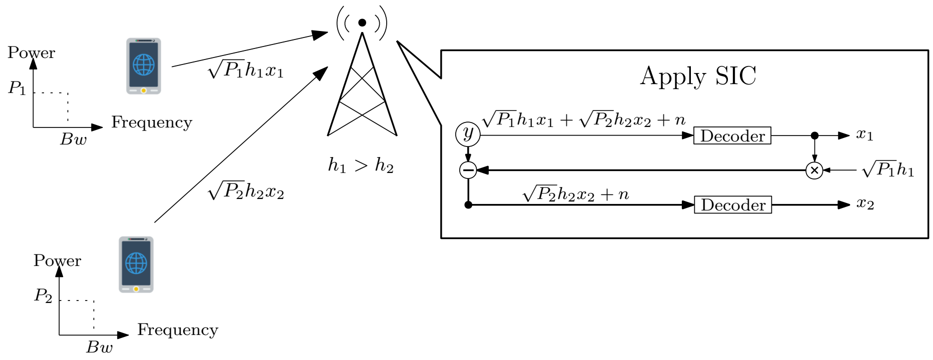


Figure 2.1: Uplink NOMA framework

The analysis of power domain-NOMA's system-level end-to-end uplink (UL) outage probabilities is presented in [60], where the authors derive exact closed-form expressions. In [61], the authors explore the implementation of power domain NOMA as the MA technique in a cell-free system, resulting in improved spectral efficiency for the UL and downlink of satellite and terrestrial networks. The application of the power domain NOMA to extend the lifetime of wireless-powered MIMO systems is discussed in [62]. In [63], power domain NOMA is utilized for efficient user activity detection and multi-user detection. The throughput analysis of UL NOMA is conducted in [64], where the authors propose an algorithm that adaptively controls user access based on observable channel feedback to maintain optimal channel loading and achieve maximum throughput. [65] focuses on signal detection and optimal power allocation for an UL cognitive radio network-NOMA system using an optimal deep learning model. Finally, [66] examines the impact of the power imbalance factor in a 2-user power domain NOMA system's UL, with a focus on minimizing average error probability in terms of power in MIMO with power control.

Not only does NOMA allow multiple devices to share the same radio resources boosting the network capacity significantly, but it can also reduce the signaling overhead and latency by allowing the grant-free UL connection [67]. Current wireless networks allocate data transmission slots to users through a process called random access, which is a multi-step handshake between the BS and the user. When there are many users, which is the case in mIoT, this grant-based access suffers from excessive signaling overhead that takes up significant resources to establish a connection. This problem is even more challenging in grant-free access since access collisions must be considered, and sufficient knowledge of participant users is unavailable. Subsequently, it can result in a substantial efficiency loss since the typical data size is comparable to the overhead signal. The incorporation of NOMA with grant-free access, which is a lightweight random access protocol, is considered to be a key enabler of massive connectivity in IoT [68–70].

To efficiently handle random access in mIoT, [71] presents a grant-free random access scheme based on classifier diversity combining using independent component analysis. In [72], the authors employ stochastic geometry tools to develop a practical framework for analyzing UL performance in large-scale multi-cell networks under grant-free NOMA and short packet transmission. To tackle collision challenges, [73] proposes a distributed Q-learning-assisted grant-free random access scheme. Enhancing network connectivity and spectral efficiency is the objective of [74], which introduces a semi-grant-free NOMA scheme that allows grant-based and grant-free users to share spectrum resources. Additionally, [75] adopts deep reinforcement learning to facilitate decision-making in grant-free NOMA systems, with the aim of reducing collisions and improving system throughput in an unknown network environment.

2.2 Channel Estimation in Communication

In the following subsections, the reader will be guided through the array of techniques leveraged to gather CSI. The discussion will initially focus on the concept of blind estimation, following which, emphasis will be placed on training-based estimation, and ultimately, semi-blind estimation will be explained.

2.2.1 Blind Channel Estimation

The technique of blind estimation leverages the statistical attributes of received symbols, typically making use of second-order statistics or higher, to assess the communication channel. A multitude of approaches, such as the Viterbi algorithm, multiple signal classification (MUSIC), and space alternating generalized expectation (SAGE), have been brought forth for blind estimation. Nevertheless, the standard methods most often implemented include singular value decomposition (SVD) and eigenvalue decomposition (EVD). As it generally outperforms EVD, SVD is predominantly chosen for blind estimation applications. Broadly, blind estimation techniques can be partitioned into two categories: subspace-based methods and recursive-based methods. The former estimates the channel at the termination of each frame, while the latter carries out the estimation upon the receipt of every symbol [76].

The blind technique for channel estimation makes use of independent component analysis, working together with SVD. The first step in this process entails determining the SVD of the covariance matrix associated with the signal delivered to the destination.

$$\mathbf{\Sigma}_Y = \mathbb{E}[\mathbf{Y}\mathbf{Y}^T] = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{2.2.1}$$

Here, $\mathbf{\Sigma}_Y$ signifies the covariance matrix of the matrix $\mathbf{Y}$. Meanwhile, $\mathbf{U}$ stands as the orthogonal matrix, and $\mathbf{\Lambda}$ embodies the matrix containing singular values. Utilizing these two matrices, one can differentiate the signal space from the noise space, i.e. $\mathbf{U} = \begin{bmatrix} \mathbf{U}_{\text{signal}} & \mathbf{U}_{\text{noise}} \end{bmatrix}$ and $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{\text{signal}} & 0 \\ 0 & \mathbf{\Lambda}_{\text{noise}} \end{bmatrix}$. By means of the projection technique, the received signal will be subjected to whitening.

$$\mathbf{Y}_w = \mathbf{\Lambda}_{\text{signal}}^{-0.5} \mathbf{U}_{\text{signal}}^T \mathbf{Y}, \tag{2.2.2}$$

Subsequently, a matrix denoted as $\mathbf{W}$ will be introduced with the aim of minimizing the mutual information of the received signal [77]. This will enable the estimation of the communication channel, $\hat{\mathbf{H}}_{\text{blind}}$ as

$$\hat{\mathbf{H}}_{\text{blind}} = \mathbf{U}_{\text{signal}} \mathbf{\Lambda}_{\text{signal}}^{0.5} \mathbf{W}, \tag{2.2.3}$$

2.2.2 Training Based Channel Estimation

Under training-based estimation, users send long pilot symbols to the BS for the purpose of channel estimation. Despite the low computational complexity of this estimation method, it suffers from a number of disadvantages such as significant throughput reduction in FBL. Following this, I will investigate two principal approaches to training-based estimation.

Maximum Likelihood Estimator

The maximum likelihood estimator (MLE) computes the parameter value that maximizes the likelihood of the observations. Supposing that the sequence transmitted by users encompasses known pilot symbols followed by unknown data, the total transmitted and received symbols are represented by $\mathbf{S} = [\mathbf{S}_p \mathbf{S}_d]$ and $\mathbf{Y} = [\mathbf{Y}_p \mathbf{Y}_d]$, respectively. The maximum likelihood appraisal of the channel matrix $\mathbf{H}$, in accordance with the training sequence, is outlined as follows:

$$\hat{\mathbf{H}}_{\text{MLE}} = \left(\mathbf{Y}_p \mathbf{S}_p^H\right) \left(\mathbf{S}_p \mathbf{S}_p^H\right)^{-1}, \tag{2.2.4}$$

For a reliable estimation pilot sequence should be relatively large which causes throughput degradation [78].

Minimum Mean-Square Error

The minimum mean-square error (MMSE) is a Bayesian estimator that utilizes a loss function defined as the second moment of the error [79]. The MMSE estimation of the channel matrix $\mathbf{H}$, derived from the training sequence, can be expressed as:

$$\hat{\mathbf{H}}_{\text{MMSE}} = \mathbf{Y}_p \left(\mathbf{S}_p^H \mathbf{S}_p + N_0 \mathbf{I}_p\right)^{-1} \mathbf{S}_p, \tag{2.2.5}$$

where N_0 is the noise density.

2.2.3 Semi-blind Channel Estimation

Semi-blind channel estimation presents an opportunity to achieve a favorable compromise between accuracy, complexity, and spectral efficiency. In this subsection, different approaches to semi-blind channel estimation will be presented.

Expectation Maximization algorithm

The estimate of channel matrix $\mathbf{H}$ based on both received pilot and data signals can be written as

$$\hat{\mathbf{H}} = \arg\max_{\mathbf{H}} \ \log\ p(\mathbf{Y}|\mathbf{H}), \tag{2.2.6}$$

The task of deriving a closed-form solution for this incomplete data problem is widely acknowledged to be challenging [80]. Another approach to address the problem in equation 2.2.6 is to employ the expectation maximization (EM) algorithm. The EM algorithm is an iterative procedure that updates the channel estimate ($\hat{\mathbf{H}}^{i+1}$) based on the previous estimate ($\hat{\mathbf{H}}^{i}$) according to the following procedure:

$$\hat{\mathbf{H}}_{\mathrm{EM}}^{i+1} = \arg\max_{H} \ \mathbb{E}_{p(\mathbf{S}_d|\mathbf{Y},\hat{\mathbf{H}}_{\mathrm{EM}}^{i})} \left(\log p(\mathbf{Y}, \mathbf{S}_d|\mathbf{H})\right), \tag{2.2.7}$$

Here, $(\mathbf{Y}, \mathbf{S}_d)$ denotes the complete data. As shown in equation 2.2.7, the computation comprises an expectation evaluation (E-step) and a maximization (M-step). With each iteration of the EM algorithm, the likelihood function gradually rises until it reaches a local maximum [81]. Considering a packet of length N where the first L symbols are pilot symbols, the EM iterates over the following two steps until the convergence [76]:

- **E-Step:** By leveraging $p(\mathbf{S}_d|\mathbf{Y}, \hat{\mathbf{H}}_{\mathrm{EM}}^{i})$, the conditional probability of $\mathbf{S}_d$ given $\mathbf{Y}$ and the previous estimates of the unknown parameters, the mean and covariance of the received signal at the $(i)^{\mathrm{th}}$ iteration can be computed in the following manner:

$$\begin{aligned} \mu_j^i &= \left(\hat{\mathbf{H}}_{\mathrm{EM}}^{i^H}\hat{\mathbf{H}}_{\mathrm{EM}}^{i} + \sigma_n^2\mathbf{I}\right)^{-1} \hat{\mathbf{H}}_{\mathrm{EM}}^{i^H}\mathbf{y}[j], \\ \mathbf{\Sigma}^i &= \sigma_n^2 \left(\hat{\mathbf{H}}_{\mathrm{EM}}^{i^H}\hat{\mathbf{H}}_{\mathrm{EM}}^{i} + \sigma_n^2\mathbf{I}\right)^{-1}, \end{aligned} \tag{2.2.8}$$

where $j \in \{L, \cdots, N\}$ is a unknown data symbol and σ_n is the variance of noise.

- **M-Step:** The channel estimation at the $(i + 1)^{\mathrm{th}}$ iteration is computed as follows [82]:

$$\hat{\mathbf{H}}_{\mathrm{EM}}^{i+1} = \left(\mathbf{Y}_p\mathbf{S}_p^H + \sum_{j=L}^{N} \mathbf{y}[j]\mu_j^{i^H}\right) \times \left(\mathbf{S}_p\mathbf{S}_p^H + \sum_{j=L}^{N} \mu_j^i\mu_j^{i^H} + \mathbf{\Sigma}^i\right)^{-1}, \tag{2.2.9}$$

Integrated Pilot-EVD

In the process of integrated pilot-EVD (IPE) channel estimation, the primary step involves utilizing the second-order statistics of the received signal. Initiating by identifying the EVD of the auto-correlation matrix for the received signal, one can distinguish between the noise and signal subspace through the hierarchical structure of eigenvalues. This is followed by the establishment of a channel matrix that maintains orthogonality with the noise subspace. Up to this stage, the methodology mimics that of blind estimation. However, blind estimation's inherent high computational cost and slow convergence rate necessitate the use of a limited set of training symbols. This enables us to have an acceptable initial channel estimation and significantly assists in complexity reduction [83].

Assuming $\mathbf{G}$ is the signal space calculated from EVD, the estimated channel vector can be written as

$$\hat{\mathbf{H}}_{\mathrm{IPE}} = \left(\mathbf{G}^H\mathbf{G} + \mathbf{I}\right)^{-1} \hat{\mathbf{H}}_{\mathrm{pilot}}, \tag{2.2.10}$$

where $\hat{\mathbf{H}}_{\mathrm{pilot}}$ is the estimated channel using pilots.

2.3 AI-assisted Communication

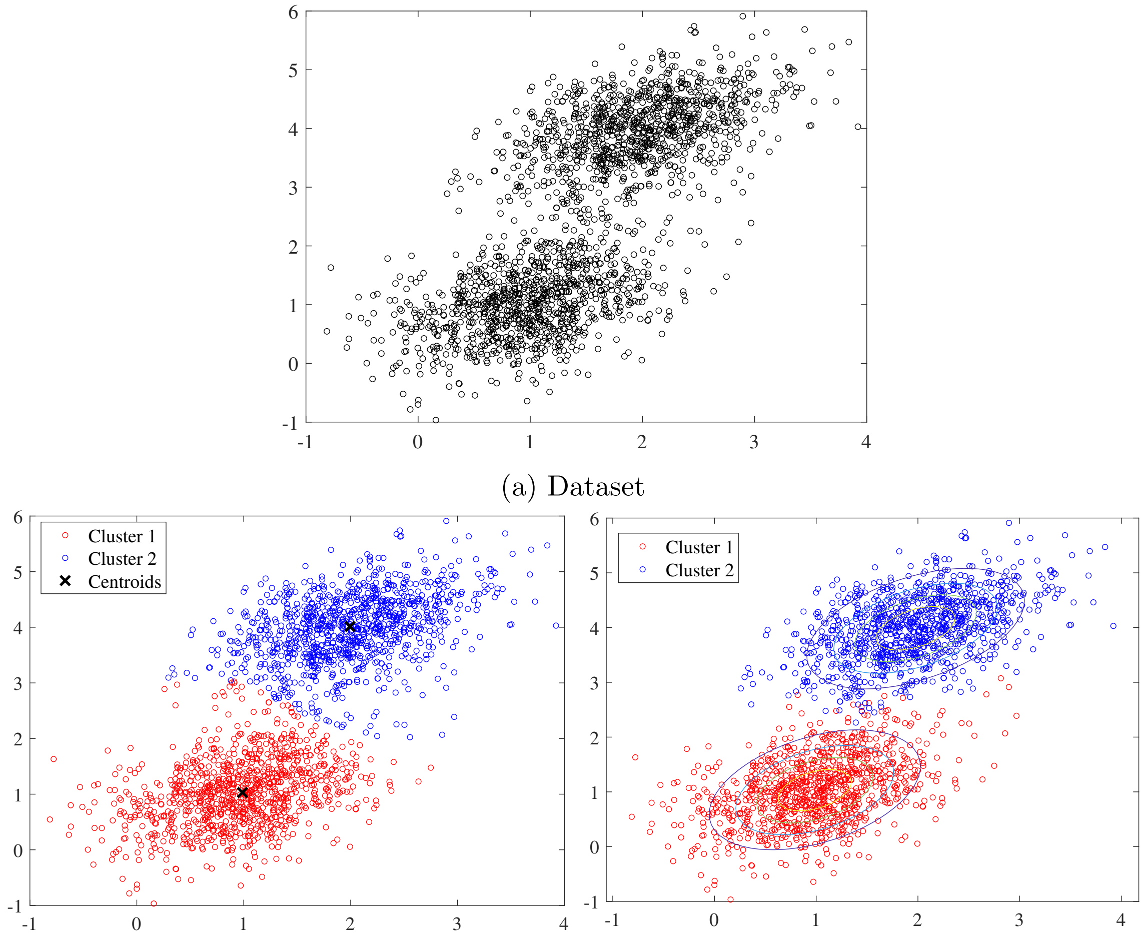
The growth of data traffic demand with varying performance requirements presents a significant challenge for networks in the future. A recent forecast estimates that the number of enterprise and automotive IoT devices will grow from 5.8 billion in 2020 to 41.6 billion in 2025, generating 79.4 zettabytes of data per annum [84]. While 5G is expected to serve many IoT applications, major breakthroughs in designing communication protocols and radio resource management techniques are required to serve applications with a diverse range of requirements in terms of data rate, reliability, availability, end-to-end latency, energy efficiency, security, and privacy [85]. One promising solution to handle big data efficiently is ML, which has demonstrated the capability to facilitate the analysis and prediction of complex scenarios in communication systems [86, 87]. ML algorithms can optimize wireless systems and address intricate problems that cannot be solved through traditional mathematical models, increasing the security and reliability of wireless systems [88]. These algorithms have been applied to various networks and networking technologies to meet the future requirements of communicating devices and services [89]. ML methods, such as those employed in channel estimation, signal detection, and modulation recognition, outperform traditional communication theory and technology, providing excellent performance [90].

ML is capable of rendering effective and efficient services in different layers of networks. In the physical layer, unsupervised learning can be performed for channel estimation and signal detection, convolutional neural network (CNN) algorithms can be applied for channel decoding, and complex CNNs can be adopted to build orthogonal frequency-division multiplexing (OFDM) [91]. In the data link layer, supervised learning and deep neural network (DNN) algorithms can be utilized for user scheduling and operation of automatic repeat request (ARQ) or hybrid ARQ to enhance the reliability. Moreover, in the network layer, reinforcement learning can be applied to improve the network robustness and guarantee service continuity [92, 93].

My research concentrates on the physical layer, and for channel estimation, I will utilize unsupervised learning algorithms. Unsupervised learning techniques, particularly clustering, are employed due to the absence of labeled data at the receiver.

Utilizing supervised learning would necessitate pre-existing information about the mapping from clusters to the correct constellations, information that is not available prior to clustering. Moreover, the application of supervised learning for channel estimation would essentially equate to the use of pilot signals, an approach this study intentionally avoids to enhance efficiency. These algorithms are provided with a set of unlabeled data to accurately predict the output, and they are commonly used for clustering and aggregation problems [94]. However, the utilization of such algorithms in the physical layer domain has been relatively limited. In [95], the investigation focuses on joint user activity and data detection for grant-free random access. The problem of jointly detecting user activity and data is formulated as a clustering problem using the GMM. In [96] and [97], the authors employ GMM in a two-user space-time line coded UL NOMA system. The studies demonstrate that the utilization of GMM allows for superior performance compared to conventional systems in terms of BER.

There are several clustering algorithms, such as K-means [98], DBSCAN [99], OPTICS [100, 101], mean shift [102], and GMM [103], that can be used to estimate the clusters of the received signals. By measuring the distance from the nearest associated centroid, the K-means algorithm assigns each new data point to a cluster. The centroids are then updated based on the previously assigned data point, and this cycle continues until there is no further change in the input data points and centroids [104]. DBSCAN is robust to outliers (noises), and needs to adjust only two parameters for an acceptable result. This algorithm can be applied when clusters have different shapes but similar densities [105]. OPTICS handles the large density difference problem of DBSCAN, and mean shift applies an iterative method, and follows the direction of weight of nearby points, and assigns each point to the closest cluster centroid [106]. Similar to DBSCAN, in the mean shift algorithm, clusters can have arbitrary shapes and do not need to have any knowledge about the number of clusters [107]. Using parameters like prior probability, mean, and variance of each cluster, GMM divides a given dataset into k clusters. The EM algorithm is responsible for determining these parameters [108].



(a) Dataset

(b) K-means

(c) GMM

Figure 2.2: Comparison of GMM and K-means.

To shed additional light on this matter, Fig. 2.2 represents a distinct disparity between the clusters created by K-means and GMM. Fig. 2.2a visually presents the group of data points collected at the BS. Fig. 2.2b and Fig. 2.2c respectively outline the clustering outcomes achieved with K-means and GMM. Notably, with K-means, the clustering process is solely predicated on the distances of points from the centroids, implying that a point with considerable noise can cause a shift in the cluster's form. Yet, GMM bases its clustering on the underlying Gaussian distribution, offering superior performance when contending with noise.

2.3.1 Gaussian Mixture Model

Given a set of data points in a d-dimensional space, I denote the d-dimensional multivariate Gaussian probability density function as $g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and the covariance matrix, respectively, and express it as

$$g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}}. \tag{2.3.1}$$

In GMM clustering, the number of clusters is pre-determined, and the data is assumed to be generated by a mixture of Gaussian distributions. A GMM parameterizes the mean, covariance, and weight of each Gaussian distribution component. When a common M-ary modulation scheme is adopted by all users, there are M_o Gaussian distributions each with a nonnegative mixture weight w_{cj} where $j \in \{1, \cdots, M_o\}$ and $\sum_{j=1}^{M_o} w_{cj} = 1$. Accordingly, the underlying Gaussian mixture distribution can be written as a convex combination of M_o constituent Gaussian distributions (each representing a cluster), i.e.,

$$p(\mathbf{z}; \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{M_o}, \boldsymbol{\Sigma}_1, ...\boldsymbol{\Sigma}_{M_o}) = \sum_{j=1}^{M_o} w_{cj} g_j(\mathbf{z}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \tag{2.3.2}$$

The aim is to estimate $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$, and w_{cj}, $j = 1, \cdots, M_o$, from the observed data. This can be done by maximizing the likelihood function (2.3.2) for all the observed data. To this end, I utilize the EM algorithm [109] that is suitable for solving maximum-likelihood problems containing unobserved latent variables.

The corresponding log-likelihood function can be defined as

$$\begin{aligned} & l^{(t)}(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{M_o}, \boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_{M_o} | \mathbf{z}_1, \cdots, \mathbf{z}_N) \\ & = \sum_{i=1}^{N_s} \left[\sum_{j=1}^{M_o} \Delta_{i,j}^{(t)} \ln \left(w_{cj}^{(t)} g_j \left(\mathbf{z}_i; \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)} \right) \right) \right], \end{aligned} \tag{2.3.3}$$

where N_s is the total number of symbols. I evaluate this function only to verify the convergence of the EM algorithm. Let $\Delta_{i,j}$ symbolize the association of the ith data point to the jth cluster represented by the jth Gaussian distribution. Therefore, I have

$$\Delta_{i,j} = \begin{cases} 1; & \text{if } \mathbf{z}_i \text{ belongs to cluster } g_j, \\ 0; & \text{otherwise.} \end{cases}$$

It is clear that $P(\Delta_{i,j} = 1) = w_{cj}$ and $P(\Delta_{i,j} = 0) = 1 - w_{cj}$. However, both $\Delta_{i,j}$ and w_{cj} are unknown.

In the tth iteration of the EM algorithm, I first estimate the so-called responsibility variable for each model j and every data point i as

$$\hat{\gamma}_{i,j}^{(t)} = \frac{\hat{w_{cj}}^{(t-1)} g_j \left(\mathbf{z}_i; \hat{\boldsymbol{\mu}}_j^{(t-1)}, \hat{\boldsymbol{\Sigma}}_j^{(t-1)} \right)}{\sum_{k=1}^{M_o} \hat{\omega_k}^{(t-1)} g_k \left(\mathbf{z}_i; \hat{\boldsymbol{\mu}}_k^{(t-1)}, \hat{\boldsymbol{\Sigma}}_k^{(t-1)} \right)}. \tag{2.3.4}$$

I then assign each data point to its corresponding cluster. In particular, for each $\mathbf{z}_i$, I find $m_i^{(t)} = \arg\max_j \hat{\gamma}_{i,j}^{(t)}$ and set

$$\Delta_{i,j}^{(t)} = \begin{cases} 1; & \text{if } j = m_i^{(t)}, \\ 0; & \text{otherwise.} \end{cases}$$

In the next step of the EM algorithm, I use the calculated responsibilities to update the mixture weight, mean, and variance for each cluster as follows

$$\hat{w_c}_j^{(t)} = \frac{\sum_{i=1}^{N_s} \hat{\gamma}_{i,j}^{(t)}}{\sum_{i=1}^{N_s} \sum_{k=1}^{M_o} \hat{\gamma}_{i,k}^{(t)}}, \tag{2.3.5}$$

$$\hat{\boldsymbol{\mu}}_j^{(t)} = \frac{\sum_{i=1}^{N_s} \hat{\gamma}_{i,j}^{(t)} \mathbf{z}_i}{\sum_{i=1}^{N_s} \hat{\gamma}_{i,j}^{(t)}}, \tag{2.3.6}$$

$$\hat{\boldsymbol{\Sigma}}_j^{(t)} = \frac{\sum_{i=1}^{N_s} \hat{\gamma}_{i,j}^{(t)} \left[\mathbf{z}_i - \hat{\boldsymbol{\mu}}_j^{(t)}\right] \left[\mathbf{z}_i - \hat{\boldsymbol{\mu}}_j^{(t)}\right]^T}{\sum_{i=1}^{N_s} \hat{\gamma}_{i,j}^{(t)}.}. \tag{2.3.7}$$

After enough iterations, the values of responsibility, mean, covariance, and mixture weight for each cluster converge as the EM algorithm is guaranteed to converge to a local optimum [110]. The number of iterations required for convergence mainly depends on the convergence criterion.

2.4 Communication-assissted AI

Communication-assisted AI, particularly FL, is a new technology that offers computation services at the network edge. This approach minimizes energy consumption and latency, and it also reduces the strain on communication links. There are key distinctions between FL and other forms of distributed learning. Parallel learning, in which the aim is to expedite the learning process and scale up the algorithm, and ensemble learning, in which the goal is to discover the optimum model from a collection of models, assume that the data is distributed in an independent and identically distributed (i.i.d) manner with no consideration for communication constraints [111, 112]. However, in FL, data is distributed in a non-i.i.d fashion, and communication limits are imposed by privacy, security, power, and bandwidth constraints [113]. The objective of traditional distributed learning is to produce an estimate of the parameters under investigation, while the objective of FL is to train the model across a large number of learners. Additionally, with conventional distributed learning, the global model parameters will not be given to local learners [114]. FL enhances user privacy by enabling local model training, thus reducing the transmission of sensitive data to BS.

2.4.1 Federated Learning

Figure 2.3 illustrates the sequential steps involved in FL. In each iteration, the users individually train their respective ML models using their local data. Subsequently, only the updates to the local model parameters are transmitted to the BS, reducing the load on the network links. Next, the BS aggregates the parameter updates by performing a weighted average and computes the new parameters for the global model. Finally, the BS broadcasts the updated global model back to each local learner as the initialization for their local model in the subsequent round [115].

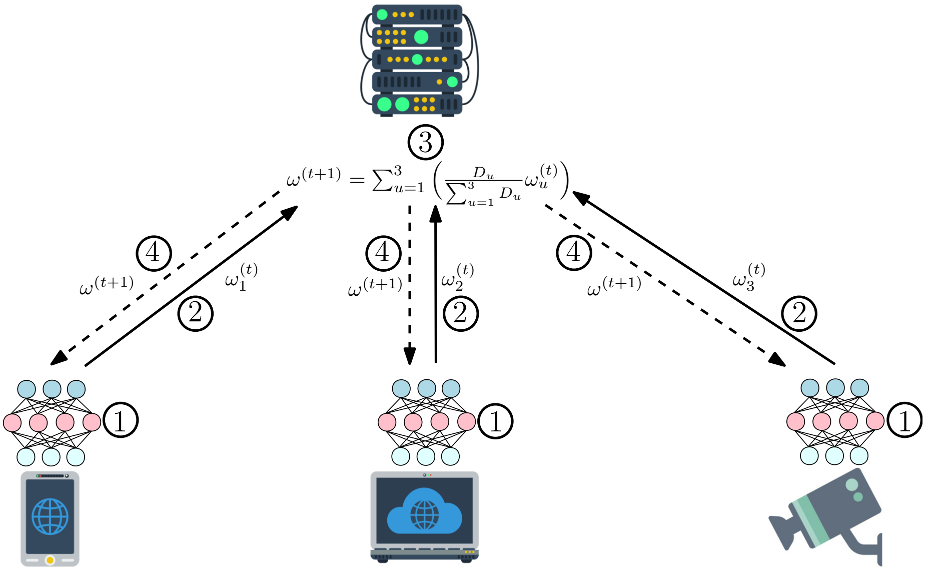


Figure 2.3: Federated Learning framework

It is critical to comprehend the difficulties associated with FL. Since each device's dataset is acquired by the user itself, it depends on the client's local environment. As a result, not only the users' datasets are non-i.i.d throughout the network, but also the size of users' datasets may vary significantly [116]. This statistical heterogeneity has an impact on the convergence mechanism of FL and lowers model accuracy. One can use an adaptive averaging strategy or apply a data-sharing mechanism to reduce the impact of non-i.i.d datasets [117]. To cope with the heterogeneity of systems, FL must handle a variety of devices that have differing amounts of memory and processing power as well as differing battery sizes and storage capacities [113]. Weight-based federated averaging is considered a solution to this problem [118]. Another challenge of FL is the tradeoff between devices' processing power and communication overhead. Local processing power is substantially faster than communication in the network. On the one hand, increasing the number of local computing iterations leads to a decrease in the number of network communications. On the other hand, since users are constrained in terms of power, increasing the number of local iterations depletes the battery, and the device's ability to communicate with the BS is severely limited [119].

In [120], the authors utilized FL to develop a model for distributed resource management in cellular networks, with a specific focus on minimizing UL transmit power. In [121], the authors demonstrate the tradeoff between privacy and FL performance, shedding light on the challenges and considerations in maintaining privacy while achieving optimal FL results. [122] investigates the problem of energy-efficient transmission and computation resource allocation for FL over wireless communication networks, aiming to optimize resource utilization in FL systems. In [123], the authors formulate an optimization problem that jointly considers user selection and resource allocation to minimize the FL training loss, highlighting the importance of effective allocation strategies. [124] presents an analytical study examining the effects of different scheduling policies, such as random scheduling, round-robin, and proportional fair, on the performance of FL in wireless networks.

2.4.2 Gradient Descent

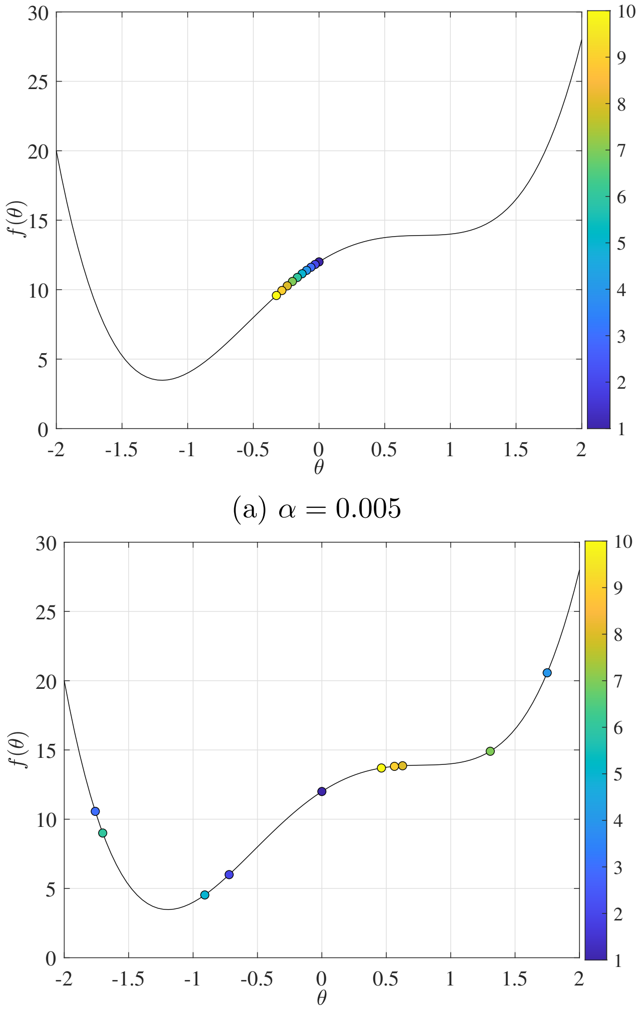
Gradient descent is a powerful iterative optimization algorithm used to locate the local minima or maxima of a given function. It is frequently utilized in ML applications to minimize cost or loss functions. However, it is crucial to note that this method is not universally applicable and is only effective on functions that are differentiable and convex [125].

As is shown in algorithm 1, the gradient descent algorithm operates iteratively, calculating the next point by incorporating the gradient at the current position. This gradient is multiplied by a learning rate and then subtracted from the current position to facilitate a step. The purpose of the subtraction is to minimize the function; conversely, the addition would be employed for maximization [126].

Algorithm 1: Gradient Descent Algorithm.

```
Input: Differentiable Function $f(\theta)$, learning rate $\alpha$
Output: Local optima of $f(\theta)$
Make an initial guess for the model parameter $\theta^0$
repeat
    $\theta^{i+1} \leftarrow \theta^i - \alpha \frac{\partial}{\partial \theta} f(\theta) \Big|_{\theta=\theta^i}$
    $i \leftarrow i + 1$
until converge
```

The learning rate plays a critical role in this process as it scales the gradient and governs the step size, and significantly impacts the algorithm's performance. A small learning rate (Fig. 2.4a) can lead to sluggish convergence or premature termination, failing to reach the optimal point. Conversely, a large learning rate (Fig. 2.4b) may induce unstable behavior, causing the algorithm to diverge or oscillate around the optimal solution [127].



(a) $\alpha = 0.005$

(b) $\alpha = 0.1$

Figure 2.4: Impact of learning rate on the convergence of gradient descent algorithm.

Chapter 3

Design and Analysis of Clustering-based Joint Channel Estimation and Signal Detection for Uplink NOMA

In this chapter, I focus on the design and analysis of a novel joint channel estimation and signal detection algorithm for NOMA. The goal is to enable the BS to jointly estimate the channel and detect the signal from each user, with no or minimum number of pilot symbols. This will significantly reduce the signaling overhead, especially in a mIoT setting, where the BS is unable to estimate the channel to each and every user constantly. I represent the problem as a clustering problem and then use the GMM approach to solve it. I further analyze the convergence of the proposed approach and show that it can achieve almost the same performance as the optimal ML detector which requires full CSI for all users.

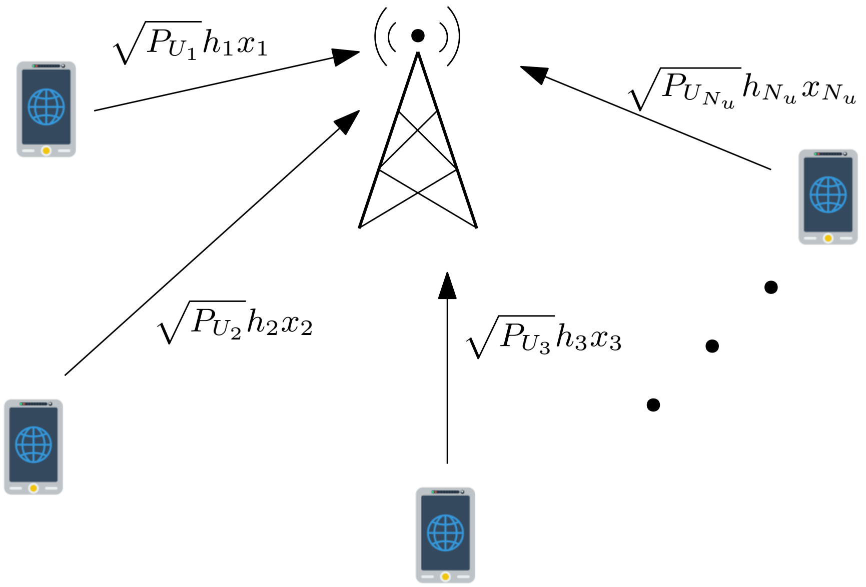


Figure 3.1: System Model.

3.1 System Model

As is shown in Fig. 3.1, I consider a cellular UL NOMA scenario in which N_u active users simultaneously transmit packets of length N_s symbols to a BS. Let x_i denote the signal transmitted by user i, which is drawn from the signal constellation $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{M_o}\}$ with $M_o = |\mathcal{S}|$ being the modulation order. The received signal at the BS, denoted by y, is given by

$$y = \sum_{i=1}^{N_u} h_i \sqrt{P_i} x_i + w, \tag{3.1.1}$$

where P_i is the transmit power of user i and $w \sim \mathcal{CN}(0,1)$ is AWGN. The gain of the channel between user i and the BS is denoted by h_i, which includes both small-scale and large-scale fadings, i.e., $h_i = g_i\sqrt{\ell_0(r_i/r_0)^{-\alpha}\chi}$, where ℓ_0 and r_0 are the reference path-loss and reference distance, respectively, α is the path-loss exponent, r_i is the distance between user i and the BS, χ is the large-scale shadowing modeled by a log-normal distribution with zero mean and variance σ dB, and g_i is the small-scale fading modelled by the Rayleigh, Rician, Nakagami, or any other distribution. The SNR of user i at the BS is given by $\gamma_i = P_i|h_i|^2$. I assume that h_i remains constant for the duration of one packet (N_s symbols), which is a valid assumption for short packets, especially in mIoT applications [128, 129].

Given the user channel gains and knowing that all users utilize the same modulation with constellation set $\mathcal{S}$, the probability distribution of the received signals at the BS can be expressed as a mixture of Gaussian distributions. Let $\mathbf{h} = [h_1, h_2, \cdots, h_{N_u}]$, $\mathbf{p} = [P_1, P_2, \cdots, P_{N_u}]$, and $\mathbf{x} = [x_1, x_2, \cdots, x_{N_u}]$ denote the vectors of channel gains, transmit powers, and user signals, respectively. Therefore, I have

$$\begin{aligned} p\left(y|\mathbf{h},\mathbf{p},\mathcal{S}\right) &= \sum_{\mathbf{u}\in\mathcal{S}^{N_u}} p(\mathbf{x}=\mathbf{u})p(y|\mathbf{h},\mathbf{p},\mathcal{S},\mathbf{x}=\mathbf{u}) && (3.1.2)\\ &= \sum_{\mathbf{u}\in\mathcal{S}^{N_u}} \prod_{i=1}^{N_u} p(x_i=u_i)p(y|\mathbf{h},\mathbf{p},\mathcal{S},\mathbf{x}=\mathbf{u}) && (3.1.3)\\ &= \frac{1}{|\mathcal{S}|^{N_u}} \sum_{\mathbf{u}\in\mathcal{S}^{N_u}} p\left(\sum_{i=1}^{N_u}\sqrt{P_i}h_iu_i + w \,\middle|\, \mathbf{h},\mathbf{p}\right), && (3.1.4) \end{aligned}$$

where I assume that the signals are randomly drawn from $\mathcal{S}$ with equal probabilities, i.e., $p(x_i = u_i) = 1/|\mathcal{S}|$. Since $w \sim \mathcal{CN}(0,1)$, one can write $\sum_{i=1}^{N_u}\sqrt{P_i}h_iu_i + w \sim \mathcal{CN}(\sum_{i=1}^{N_u}\sqrt{P_i}h_iu_i, 1)$. Therefore, it is possible to simplify (3.1.4) as follows

$$y|\mathbf{h} \sim \frac{1}{|\mathcal{S}|^{N_u}} \sum_{\mathbf{u}\in\mathcal{S}^{N_u}} \mathcal{CN}\left(\sum_{i=1}^{N_u}\sqrt{P_i}h_iu_i,\ 1\right), \tag{3.1.5}$$

which means that the distribution of the received signal follows a Gaussian mixture model.

I assume that all users are frame-synchronized, which is achieved by frequently sending beacon signals from the BS [130]. I also assume that the BS does not know the

CSI for any user. Therefore, it attempts to jointly estimate the channels and detect users' signals. However, I assume that the BS knows the number of transmitting users, N_u, and the utilized modulation scheme. Later in Section 3.5, I will show how the assumption of knowing the number of users can be relaxed.

3.2 Clustering-based Joint Channel Estimation and Signal Detection

To better understand the proposed joint channel estimation and signal detection approach, I first discuss Fig. 3.2 that illustrates the signals collected at the receiver in the I-Q plane for a two-user NOMA communication system when both users employ the quadrature phase shift keying (QPSK) modulation[1]. I assume that $|h_1| \geq |h_2|$ without losing generality. Fading causes signal amplification and a rotation with respect to the original constellation. When the received powers for the two users are significantly different, the clusters are distinct from each other. Therefore, the phase and amplitude of the channels can be estimated accurately. However, when channel fading and noise cause the clusters to overlap, estimation of the channels and detection of the signals is more challenging and inevitably less accurate. In Section 3.2.1, I propose an effective algorithm to jointly estimate the user channels by clustering the received signals and detecting the user signals. At the BS, SIC is used to recover the multiplexed user signals from the received superimposed signals in the decreasing order of their received powers. Initially, the receiver detects the signal of the strongest user, that is, the user with the highest received power. It

[1] One can easily show that this is also valid when users use different modulations, and the assumption is made only for a better representation in the chapter.

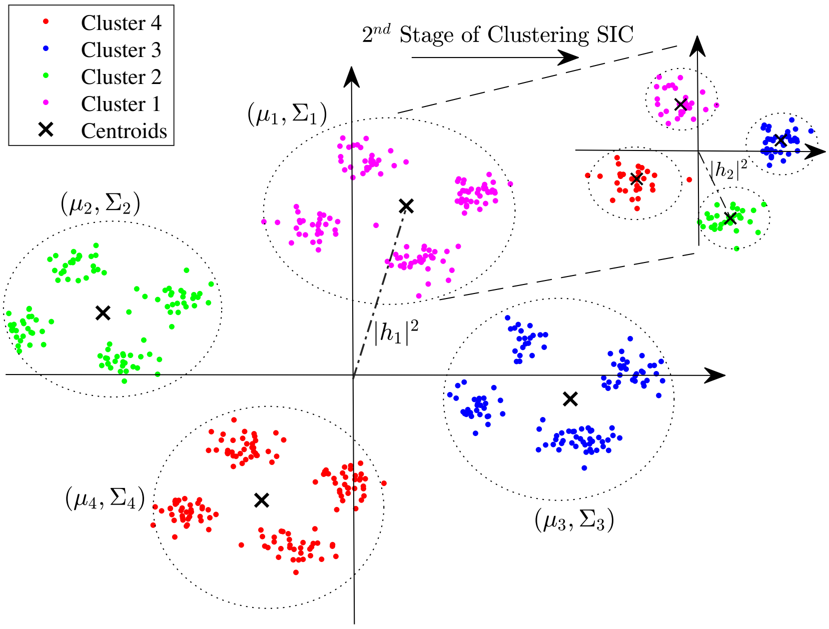


Figure 3.2: Received signal constellation diagram of a two-user NOMA system at the BS for $N_s = 500$, $\gamma_1 = 17$dB, and $\gamma_2 = 11$dB.

then reconstructs and removes it from the received signal. Afterward, it detects the next strongest signal, and so on. In order to implement SIC at the BS, I start by dividing the received signals (data points) into M_o (four with QPSK) clusters, which represent user 1's signals. Next, I further split each of the M_o clusters into M_o smaller clusters representing the user 2's signals and so on. Taking the received signals as the observed data, the considered joint channel estimation and signal detection problem boils down to estimating the unknown latent parameters of the assumed Gaussian mixture distribution in (3.1.5).

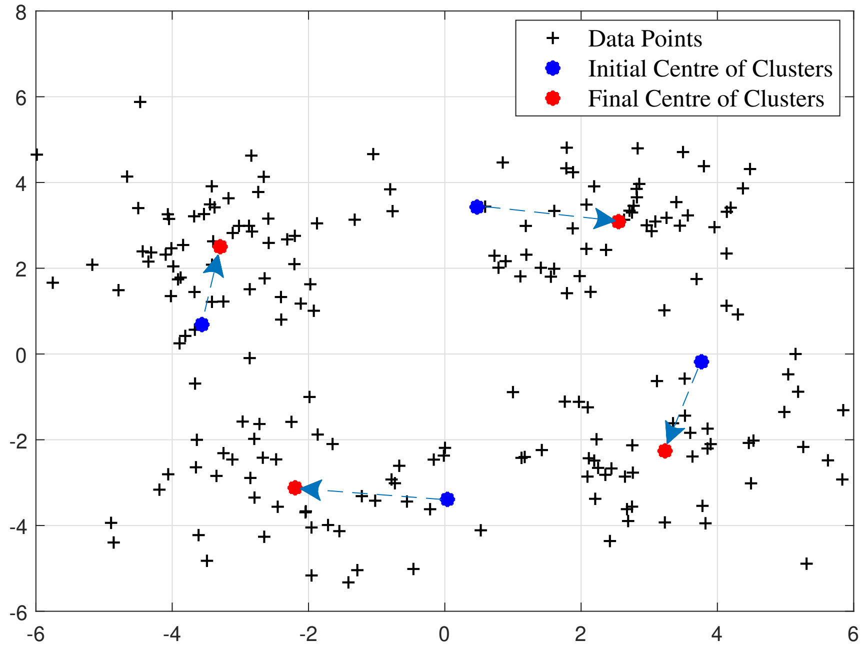


Figure 3.3: EM mean convergence after eight iterations for a single-user scenario when $\gamma = 7$dB and $N_s = 200$.

3.2.1 GMM Clustering of Received Signals at the BS

Owing to the fact that the noise has Gaussian distribution, the received signal can be modeled by a mixture of Gaussian distributions, which makes GMM a natural choice for clustering in mIoT. From a theoretical standpoint, given that the noise affecting the received signals is independent and identically distributed (i.i.d.) AWGN, the GMM clustering is equivalent to the well-known k-means clustering algorithm. However, in practise, due to the limited amount of observations, the noise covariance matrix is not a multiple of the identity matrix, which means that the noise affecting different received signals may be correlated or have different variances. As a result, GMM clustering is more accurate compared to k-means clustering since, unlike k-means, it does not assume the same covariance for all clusters but estimates them from the data.

Fig. 3.3 shows the signals received at the BS for the single user scenario employing the QPSK modulation and blocklength of $N_s = 200$. As seen, the received signals can be effectively clustered into four clusters. The GMM clustering algorithm [103] finds the centroid of these clusters in an iterative manner using the EM algorithm.

As is explained in Section 2.3.1, GMM parameterizes the mean, covariance, and weight of each Gaussian distribution component. The objective is to maximize the likelihood in terms of the parameters given the GMM, and the parameters are means and covariance of the components and the mixing coefficients. This process can be summarized as

1. Initialize the mean $\boldsymbol{\mu}_j$, covariance $\boldsymbol{\Sigma}_j$ and mixing coefficients w_{c_j} and evaluate the initial value of the log likelihood
2. **E step** Evaluate the responsibilities using the current parameter values

$$\hat{\gamma}(v_{ij}) = \frac{w_{cj}\mathcal{N}_j(\mathbf{z}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{l=1}^{M_o} w_{cl}\mathcal{N}_l(\mathbf{z}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{3.2.1}$$

3. **M step** Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_j^{new} = \frac{1}{N_{sj}} \sum_{i=1}^{N_s} \hat{\gamma}(v_{ij})\mathbf{z}_i \tag{3.2.2}$$

$$\boldsymbol{\Sigma}_j^{new} = \frac{1}{N_{sj}} \sum_{i=1}^{N_s} \hat{\gamma}(v_{ij})(\mathbf{z}_i - \boldsymbol{\mu}_j^{new})(\mathbf{z}_i - \boldsymbol{\mu}_j^{new})^T \tag{3.2.3}$$

$$w_{cj}{}^{new} = \frac{N_{sj}}{N_s} \tag{3.2.4}$$

where $N_{sj} = \sum_{i=1}^{N_s} \hat{\gamma}(v_{ij})$ that can be interpreted as the effective number of points assigned to cluster j.

4. Evaluate the log likelihood

$$\ln p(\mathbf{Z}; w_c, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N_s} \ln \left[\sum_{j=1}^{M_o} \left(w_{cj} \mathcal{N}_j(\mathbf{z}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right) \right] \tag{3.2.5}$$

5. Check for the convergence of either the parameters or the log-likelihood. If the convergence is not satisfied, return to step 2.

The number of iterations required for convergence mainly depends on the convergence criterion. In Fig. 3.3, I visualized the convergence of the mean estimates (cluster centroids) after running eight iterations of the EM algorithm.

3.2.2 The Proposed Algorithm

I summarize the proposed algorithm for joint channel estimation and signal detection in Algorithm 2. In the proposed algorithm, at each stage of SIC, I estimate the parameters of only M_o Gaussian distributions. This helps with managing the computational complexity.

Given the modulation order M_o, I fix the mixture weight of each cluster as $w_{cj} = \frac{1}{M_o}$. Next, starting from the first user, I split the received data into M_o clusters and select one point in each cluster as the initial mean, and set the initial variance of each cluster to one. Then, I calculate the responsibility and log-likelihood function values using (3.2.1) and (3.2.5), respectively. Afterward, I calculate the associated cluster centroids and covariance matrices (lines 8 to 13 in Algorithm 2). I continue by evaluating the phase of each cluster. Considering using the QPSK modulation, the phase difference between any two adjacent clusters is $\frac{\pi}{2}$. However, due to noise, the phase of each centroid might differ from $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, or $\frac{7\pi}{4}$. To minimize the effect of phase rotation, I average the phase difference between the centroid of each cluster

Algorithm 2: Joint channel estimation and signal detection using GMM clustering.

```
Input: number of users $N_u$, received signal $\mathbf{y}$, modulation order $M_o$, signal
       constellation $\mathcal{S}$, and convergence threshold $\epsilon$
Output: estimated signals of the users
Set $\omega_j = \frac{1}{M_o}$,  $j = 1, \cdots, M_o$
for user $u = 1$ to $N_u$ do
    Initialize $\hat{\boldsymbol{\mu}}^{(0)}$ and $\hat{\boldsymbol{\Sigma}}^{(0)}$ by dividing the received signals on the reference
      coordinate system into $M_o$ equal sections, select one point in each
      section as the initial mean and set the initial covariance to one
    Calculate $\hat{\gamma}_{i,j}^{(0)}$ according to (3.2.1)
    Calculate the log-likelihood function according to (3.2.5)
    Set $t = 1$
    while $l^{(t)} - l^{(t-1)} \geq \epsilon$ do
        Update $\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\Sigma}}^{(t)}$ using (3.2.2) and (3.2.3)
        Update $\hat{\gamma}_{i,j}^{(t)}$ according to (3.2.1)
        Update log-likelihood function according to (3.2.5)
    end
    Return $\hat{\boldsymbol{\mu}}$=$\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}^{(t)}$
    Calculate the phase of each cluster centroid: $\phi_i = \tan^{-1}\left(\frac{\mathrm{Im}(\hat{\mu}_i)}{\mathrm{Re}(\hat{\mu}_i)}\right)$
    Calculate the average phase as $\phi = \frac{\sum_{i=1}^{M_o} \phi_i - M_o \pi}{M_o}$ and channel amplitude
      $|\hat{h}_u| = \frac{1}{M_o} \sum_{i=1}^{M_o} \mathrm{abs}(\mu_i)$
    Update the decision boundaries based on the $\phi$ and use the pilot symbol
      to map each cluster into the mapping bits
    Demodulate the signal to symbols: $\hat{\mathbf{x}}_u = \mathrm{demod}(\mathbf{y})$
    Re-modulate this user's signal and multiply by the estimated channel gain
      and subtract it from superimposed received signal: $\mathbf{y} \leftarrow \mathbf{y} - |\hat{h}_u| e^{j\phi} \hat{\mathbf{x}}_u$
end
```

and its expected value (line 14 of Algorithm 2) and update the decision boundaries according to the average phase rotation. Finally, I apply SIC and repeat the algorithm for the next strongest user.

This approach is a joint operation due to the inherent interdependence between the channel estimation and signal detection in the proposed algorithm, as the estimated channels are prerequisites for the correct detection and removal of each user's signal, highlighting the mutual reliance of these two processes.

It is important to note that by using SIC, I assume the first M_o clusters to be Gaussian, each consisting of M_o subclusters, although, the clusters corresponding to the strongest user are not strictly Gaussian in practice. However, roughly speaking, the components of the GMM represent clusters that are of similar shapes. Thus, each cluster is implicitly assumed to follow a Gaussian distribution. For cases where this assumption is overly unrealistic, a natural alternative is to assume that each cluster is also a mixture of normally-distributed subclusters [131].

3.2.3 Symbol-to-Bit Demapping

Using GMM clustering, I separate the received signals into M_o clusters in each stage of SIC. I then estimate the channel gain $|h|$ from the coordinates of the cluster centroids (line 14 of Algorithm 2). However, this does not inform us about the exact phase of the channel. In particular, assuming that the user sends QPSK symbols, according to Fig. 3.4, one can consider four different but equally probable choices for the phase of the channel, i.e., $\angle h \in \{\theta - \pi/4, \theta - 3\pi/4, \theta + \pi/4, \theta + 3\pi/4\}$ where $-\pi/4 \leq \theta \leq \pi/4$ is the phase of the cluster that resides in the angular region of $[-\pi/4, \pi/4]$. This implies that I cannot demap the signals to the symbols.

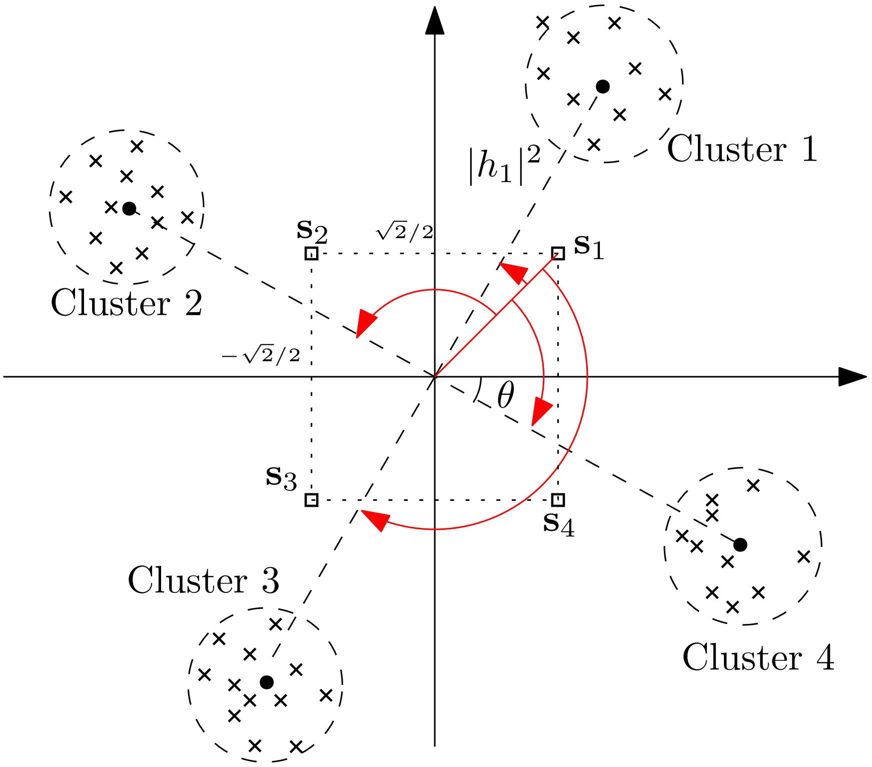


Figure 3.4: Constellation rotation due to channel fading.

To overcome this issue, I sent a few pilot symbols to determine the demapping. Determining the mapping with only one symbol is unambiguous when the SNR is not excessively low. The users' pilot symbols are distinct from one another, and in the scenario when there are two users, each with two pilot symbols, I consider $\text{Pilot}_{u_1} = [s_1, s_1]$ and $\text{Pilot}_{u_2} = [s_1, s_3]$. Furthermore, the BS receives superimposed modulated signals. For a robust and accurate performance, two symbols per user can be sufficient. I use MMSE method to estimate the phase rotation due to the channel. The MMSE estimate is given as

$$\hat{h}_{u,\text{MMSE}} = \left(x_{u,p}^H x_{u,p} + 1\right)^{-1} x_{u,p}^H y_p \tag{3.2.6}$$

where $x_{u,p}$ and $y_{u,p}$ are the transmitted and received pilot symbols, respectively. The phase of the channel is determined to be the one from $\{\theta - \pi/4, \theta - 3\pi/4, \theta + \pi/4, \theta + 3\pi/4\}$ that is the closest to $\hat{h}_{u,\text{MMSE}}$. Using this method, the exact phase rotation due to the channel is not required. As I will show later, this is another advantage of our proposed approach over the conventional channel estimation techniques. Referring to Fig. 3.4, for two-user NOMA, I assume that, in the first time slot, both users send s_1 while, in the second time slot, the first user transmits s_3 and the second user sends s_1 as the pilot symbols. After channel estimation is completed, MLD is applied to detect the data, mathematically represented as

$$\arg\min_{j} \parallel y^{(i)} - \hat{h}_u s_j \parallel^2, \qquad j \in \{1, \cdots, 4\}, \quad \forall i \in \{1, \cdots, N_s\}$$

In the proposed algorithm, the receiver initially lacks any prior knowledge of the channel coefficients. It estimates these coefficients by leveraging the combined information derived from the GMM clustering and the pilot symbols. After estimating the channel gain through GMM clustering and determining the phase rotation using pilot symbols for demapping, the data of the stronger user is decoded. Subsequently, the user's signal is re-modulated and removed from the superimposed signal. This process is repeated for the subsequent users.

3.3 Theoretical Analysis of the Proposed Clustering Algorithm

In this section, I analyze the performance of the proposed algorithm. In each iteration of EM, the mean, variance, and mixture weight of each cluster are updated. I refer to the difference between the mean of each cluster estimated by EM and its corresponding exact value as the EM error. I start by presenting a theorem that gives

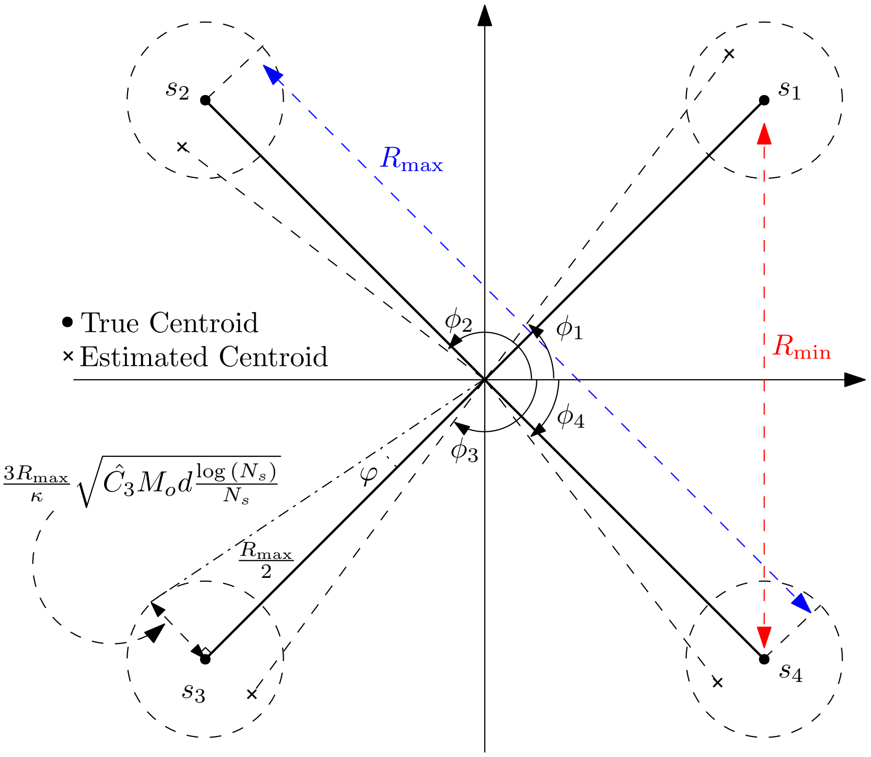


Figure 3.5: The convergence of the EM algorithm when $M_o = 4$ and $d = 2$.

an upper-bound on the EM error. Then, using this bound, I present a mathematical model to predict the SER of my proposed GMM-clustering-based joint channel estimation and signal detection algorithm.

Let $R_{\min}$ and $R_{\max}$ denote the distance between the closest cluster centroids and the distance between the farthest cluster centroids, respectively, as shown in Fig. 3.5. The following theorem characterizes an upper-bound on the EM error.

Theorem 3.3.1. *Assume d-dimensional received signals and M_o isotropic Gaussian distributions with mixture weights w_{cj}, $j \in \{1, \cdots, M\}$. Let $\boldsymbol{\mu}_i^*$ denote the true mean of cluster i and $\boldsymbol{\mu}_i^{(t)}$ represent the estimated mean of the i-th cluster after t iterations of EM. Suppose $\kappa = \min\{w_{cj}\}$, $R_{\min} \geq C_0\sqrt{\min\{d, M_o\}}$, and the initial iterate $\boldsymbol{\mu}^{(0)}$*

satisfies

$$\begin{aligned}\max_{i\in[M_o]} ||\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*||_2 \leq & \frac{R_{\min}}{2} \\ & - \frac{C_1}{2}\sqrt{\min\{d, M_o\}} \log\left(\max\left\{\frac{M_o}{\kappa^2}, R_{\max}, \min\{d, M_o\}\right\}\right)\end{aligned} \tag{3.3.1}$$

where $[M_o] = \{1, 2, \cdots, M_o\}$ *and* $C_0, C_1 > 0$ *are universal constants. For a sufficiently large sample size* N_s *such that that*

$$\frac{\log(N_s)}{N_s} \leq \min\left\{\frac{\kappa^2}{144\hat{C}_2 M_o d}, \frac{\kappa^2 \max_{i\in[M_o]} ||\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*||_2^2}{9\hat{C}_3 R_{\max}^2 M_o d}\right\} \tag{3.3.2}$$

where $C_2, C_3 > 0$ *are universal constants and*

$$\hat{C}_2 = C_2 \log\left(M_o\left(2R_{\max} + \sqrt{d}\right)\right)$$

$$\hat{C}_3 = C_3 \log\left(M_o\left(3R_{\max}^2 + \sqrt{d}\right)\right),$$

the subsequent EM iterates $\{\boldsymbol{\mu}_i^{(t)}\}_{t=1}^{\infty}$ *satisfy*

$$\begin{aligned}\max_{i\in[M_o]} ||\boldsymbol{\mu}_i^{(t)} - \boldsymbol{\mu}_i^*||_2 \leq & \frac{1}{2^t}\max_{i\in[M_o]} ||\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*||_2 \\ & + \frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d \, \log(N_s)}{N_s}}\end{aligned} \tag{3.3.3}$$

with probability at least $1 - \frac{2M_o}{N_s}$.

Proof. See Appendix A. □

Theorem 3.3.1 states that the EM error is bounded. As the iteration number, t, increases, the first term on the right-hand side of (3.3.3) converges to zero and the second term dominates. The following remark characterizes this bound.

Remark 3.3.1. When $t \to \infty$, the EM error characterized in Theorem 3.3.1 is upper bounded by

$$\max_{i \in [M]} ||\boldsymbol{\mu}_i^{(\infty)} - \boldsymbol{\mu}_i^*||_2 \leq \frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d \, \log(N_s)}{N_s}}. \tag{3.3.4}$$

This shows that EM converges to points within balls of radius $\frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d \, \log(N_s)}{N_s}}$ around the true centers. Fig. 3.5 illustrates the essence of Theorem 3.3.1 when $M_o = 4$. As seen in the figure, there is a circle of radius $\frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d \, \log(N_s)}{N_s}}$ around each true center and the EM converges to points within these circles. Moreover, the EM error converges to zero when the sample size, N_s, is arbitrarily large. This follows directly from (3.3.4) by taking N_s to infinity, noting that $\hat{C}_3$ is constant.

Remark 3.3.2. The EM error characterized in Theorem 3.3.1 converges to zero when $t \to \infty$ and $N_s \to \infty$.

The following lemma provides an approximation for the SER of the proposed GMM-clustering-based joint channel estimation and signal detection algorithm when $M_o = 4$, i.e., when QPSK modulation is used.

Lemma 3.3.1. *For a QPSK-modulated signal, i.e., $M_o = 4$ and $d = 2$, at SNR γ with N_s samples, the proposed algorithm achieves an SER approximated by*

$$P_{\text{SER}} \approx Q\left(\sqrt{2\gamma}\sin\left(\frac{\pi}{4} - \varphi\right)\right) + Q\left(\sqrt{2\gamma}\sin\left(\frac{\pi}{4} + \varphi\right)\right) \tag{3.3.5}$$

where

$$\varphi = \tan^{-1}\left(6M_o\sqrt{\frac{\hat{C}_3 M_o d \log(N_s)}{N_s}}\right) \tag{3.3.6}$$

and $Q(z) = \frac{1}{\sqrt{2\pi}}\int_z^\infty e^{-\frac{\nu^2}{2}} d\nu$.

Proof. Considering Remark 3.3.1, each estimated cluster centroid has a distance of at most $\frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d\ \log(N_s)}{N_s}}$ to the true centroid when $t \to \infty$. This means for QPSK-modulated signals ($M_o = 4$), as shown in Fig. 3.5, there is a phase mismatch between the final cluster centroids and their corresponding true centroids given by

$$\begin{aligned} \varphi &= \tan^{-1}\left(\frac{\frac{3R_{\max}}{\kappa}\sqrt{\frac{\hat{C}_3 M_o d \log(N_s)}{N_s}}}{\frac{R_{\max}}{2}}\right) \\ &= \tan^{-1}\left(6M_o\sqrt{\frac{\hat{C}_3 M_o d \log(N_s)}{N_s}}\right) \end{aligned} \tag{3.3.7}$$

where I use $\kappa = 1/M_o$. Therefore, the detection of the QPSK signals is with a rotated phase reference of φ. In [132], it is shown that the SER for a QPSK signal with a noisy phase reference $-\frac{\pi}{4} \leq \varphi \leq \frac{\pi}{4}$ is given by (3.3.5). This completes the proof. □

Remark 3.3.2 states that when the number of samples N_s, i.e., the number of signals received at the BS, is sufficiently large, the EM error converges to zero. This means that the GMM-clustering-based joint channel estimation and signal detection algorithm will find the true locations of the cluster centroids (shown by black dots in Fig. 3.5). Thus, the mismatch in the phase reference φ will be zero (Lemma 3.3.1) and $P_{\text{SER}} \approx 2Q(\sqrt{\gamma})$. Therefore, the SER of the proposed algorithm will be the same as that of the MLD with full CSI.

I can further extend the results in Lemma 3.3.1 to approximate the SER of the proposed algorithm for the two-user NOMA scenario.

Lemma 3.3.2. *For a two-user NOMA scenario with N_s QPSK-modulated signals where the SNR of users 1 and 2 are γ_1 and γ_2, respectively, the proposed algorithm* *yields per-user SERs approximated by*

$$
\begin{aligned}
P_{\mathrm{SER},1} \approx & \frac{1}{4} Q\left(\left(\sqrt{2\gamma_1} + \sqrt{2\gamma_2}\right) \sin\left(\frac{\pi}{4} - \varphi_1\right)\right) \\
& + \frac{1}{4} Q\left(\left(\sqrt{2\gamma_1} + \sqrt{2\gamma_2}\right) \sin\left(\frac{\pi}{4} + \varphi_1\right)\right) \\
& + \frac{1}{4} Q\left(\left(\sqrt{2\gamma_1} - \sqrt{2\gamma_2}\right) \sin\left(\frac{\pi}{4} - \varphi_1\right)\right) \\
& + \frac{1}{4} Q\left(\left(\sqrt{2\gamma_1} - \sqrt{2\gamma_2}\right) \sin\left(\frac{\pi}{4} + \varphi_1\right)\right) \qquad (3.3.8) \\
P_{\mathrm{SER},2} \approx & Q\left(\sqrt{2\gamma_2} \sin\left(\frac{\pi}{4} - \varphi_2\right)\right) + Q\left(\sqrt{2\gamma_2} \sin\left(\frac{\pi}{4} + \varphi_2\right)\right) \qquad (3.3.9)
\end{aligned}
$$

where φ_i*, for* $i \in \{1, 2\}$*, is given by*

$$\varphi_i = \tan^{-1}\left(6M_o \sqrt{\frac{\hat{C}_3^{(i)} M_o d \log(N_s)}{N_s}}\right) \qquad (3.3.10)$$

and $\hat{C}_3^{(i)} = C_3 \log\left(M_o\left(12\gamma_i + \sqrt(d)\right)\right)^2$.

Proof. Authors of [133] characterize the SER of the QPSK constellation for the uplink NOMA scenario. It is shown that the SER for the stronger user, user 1 in our case, is approximated by

$$Q\left(\sqrt{\gamma_1} + \sqrt{\gamma_2}\right) + Q\left(\sqrt{\gamma_1} - \sqrt{\gamma_2}\right). \qquad (3.3.11)$$

In the proposed approach, I have the phase mismatch φ_1 [see (3.3.10) in Lemma 3.3.1] between the final cluster centroids and the true centroids for user 1's signals. Similar to [132], by incorporating this phase difference into each component of (3.3.11), (3.3.8) can be easily derived. After detecting user 1's signal, I apply SIC and subtract the detected signal from the received signal. Detecting user 2's signal is then

[2] *A more accurate approximation can be obtained by considering the phase rotation for each user. However, the approximations in* (3.3.8) *and* (3.3.9) *are sufficiently accurate while finding more accurate approximations is beyond the scope of this chapter.*

straightforward as it is a noisy QPSK signal with the interference of the other user canceled. Therefore, using Lemma 3.3.1, the SER of user 2 can be approximated by (3.3.9). This completes the proof. □

Note that the convergence of GMM as per Theorem 3.3.1 is local since I assume that the EM algorithm is initialized in the neighborhood of the true centroids. As the pilots are in close proximity to the true cluster centroids, they can be utilised to initialise the EM algorithm as they are at the centre of the decision boundaries.When the BS does not have the knowledge of the modulation scheme, other techniques, such as the method of moments [134], can be used for the initialization.

Since the EM algorithm used for GMM clustering comprises two alternating steps of expectation and maximization, to determine its computational complexity, I consider the complexity of each step. For a general case, I assume d dimensions, M_o clusters, and N_s samples. In the expectation step, I calculate the determinant and the inverse of the covariance matrix with the complexity of $\mathcal{O}(M_o N_s d^3)$. The maximizing stage entails calculating the mixture weight, mean, and covariance for each cluster with corresponding $\mathcal{O}(M_o N_s)$, $\mathcal{O}(M_o N_s)$, and $\mathcal{O}(M_o N_s d)$ complexity levels. Given the algorithm convergences after t iterations, the computational complexity of GMM clustering is $\mathcal{O}(t M_o N_s d^3)$, which is most importantly linear in N_s. Since all computations are carried out at the BS, which generally has sufficient computing and energy resources, the mentioned computational complexity does not pose any challenge, particularly considering the substantial gains in throughput. It is worth mentioning that the complexity of SIC-based joint MLD with full CSI is $\mathcal{O}(4M_o N_u + 2(N_u - 1))$ [135].

3.4 Numerical Results

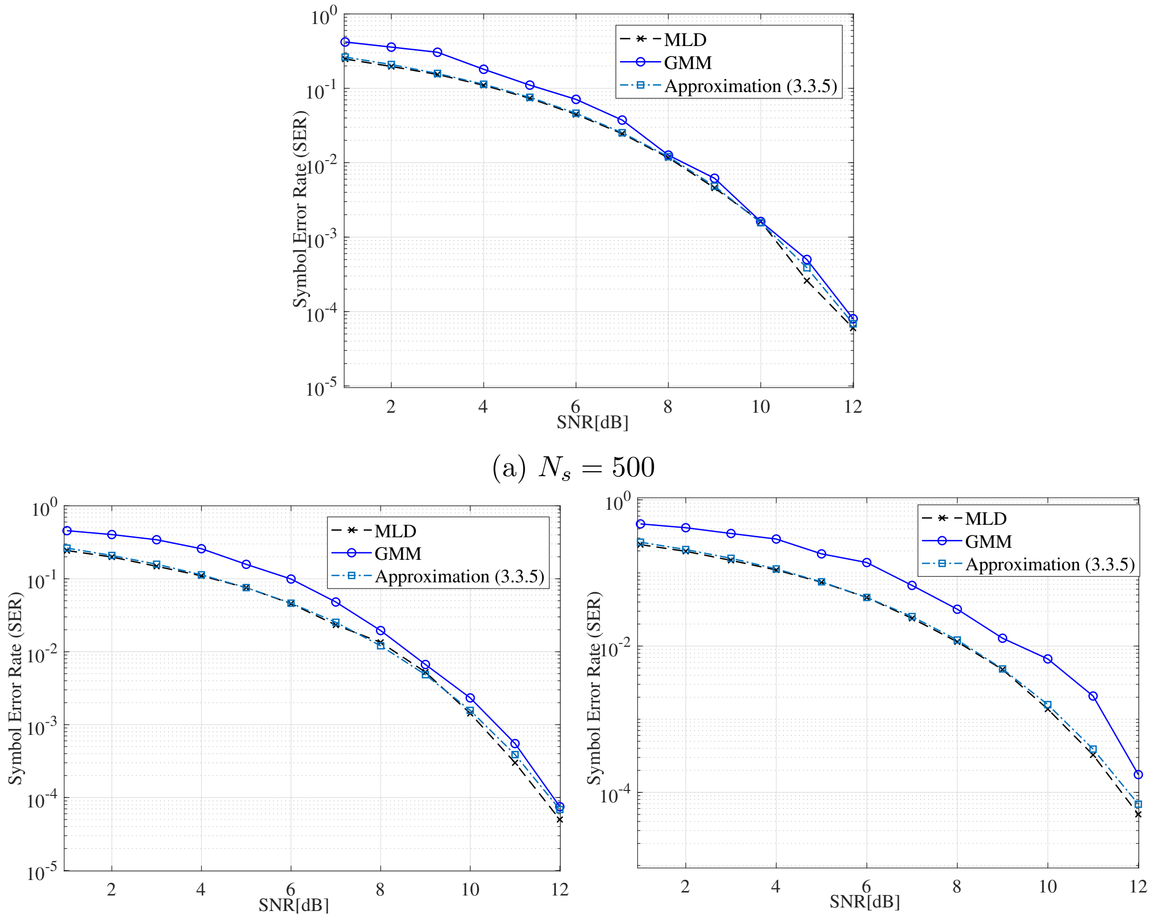
In the simulations, I consider that two pilot symbols are sent for symbol-to-bit demapping, unless I specify otherwise. I also assume that the convergence threshold, ϵ, for the proposed GMM-clustering-based approach is set to 1, unless specified otherwise. It is important to note that the transmit power allocated to both pilot signals and data symbols is identical, ensuring a fair comparison between the MLD and GMM schemes.

3.4.1 Single-user Scenario

In Fig. 3.6, I compare the SER performance of the proposed GMM-clustering-based approach with that of the MLD-based receiver with full CSI knowledge in a single-user scenario with QPSk modulation. As seen in this figure, when the sample size N_s is large, the proposed approach performs close to the optimal MLD-based one with full CSI, while at the high SNR regime, the SERs of the two approaches are almost the same. For small N_s, there is a small difference in the performance of the two approaches, which can mainly be attributed to limited observations making the cluster boundaries of GMM sub-optimal. In addition, Fig. 3.6 shows that the approximation given by (3.3.5) can predict the SER of the proposed approach well.

3.4.2 Two-user NOMA

I show the performance of the proposed GMM-clustering-based approach for a two-user NOMA communication system in Fig. 3.7. As seen in this figure, the proposed approach performs close to the optimal MLD-based approach with full CSI. Similar



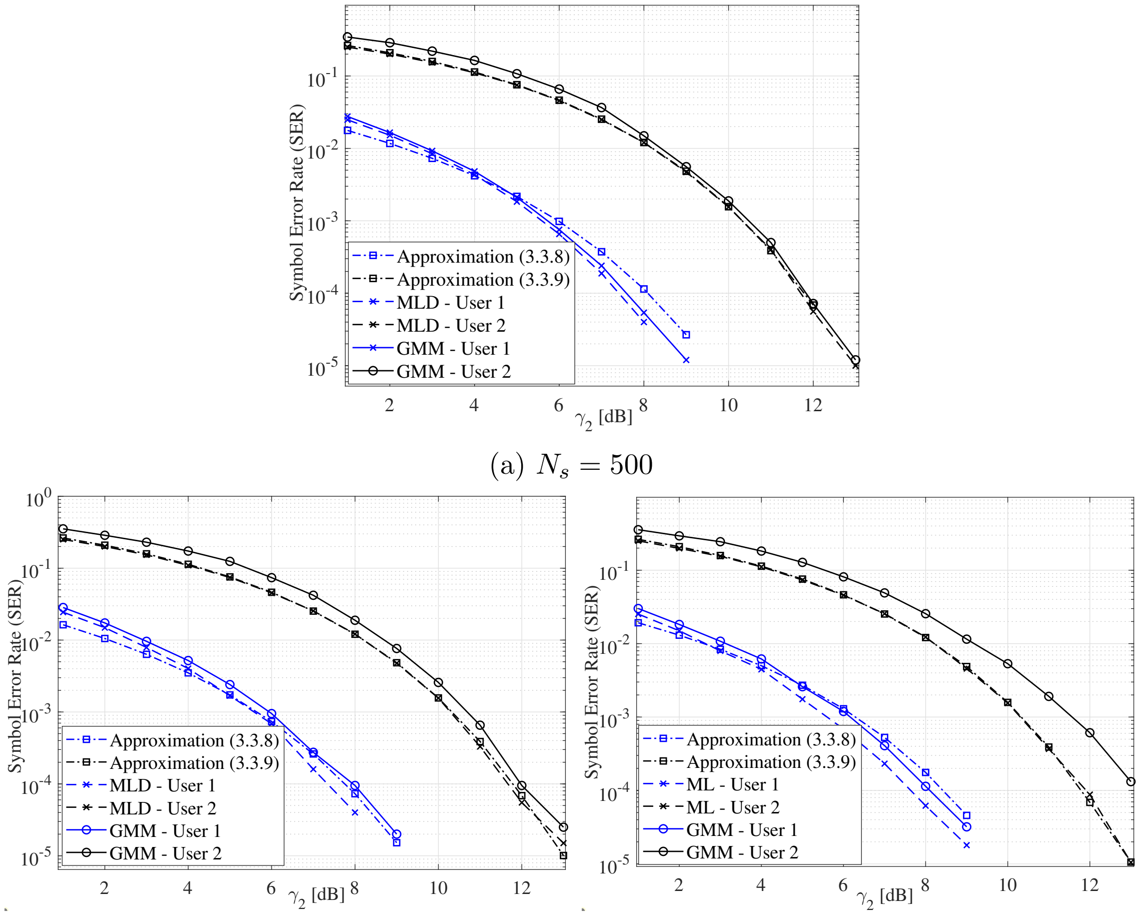
(a) $N_s = 500$

(b) $N_s = 200$

(c) $N_s = 100$

Figure 3.6: SER Comparison of the proposed GMM-clustering-based and optimal MLD-based approaches for point-to-point communication.

to the single-user case, as the number of transmitted symbols increases, the SER performance of the proposed approach reaches that of the MLD-based approach with full CSI. An advantage of the clustering-based approach is that each user needs to send only two pilot symbols for symbol-to-bit demapping. However, in current systems, to acquire sufficiently accurate CSI at the receiver, each user needs to send a long pilot sequence that is usually more than six symbols [136]. This is inefficient when the packet size (number of transmitted symbols) is small [137]. In my proposed approach, the BS does not require a perfect estimation of the channel to eliminate the influence



(a) $N_s = 500$

(b) $N_s = 200$

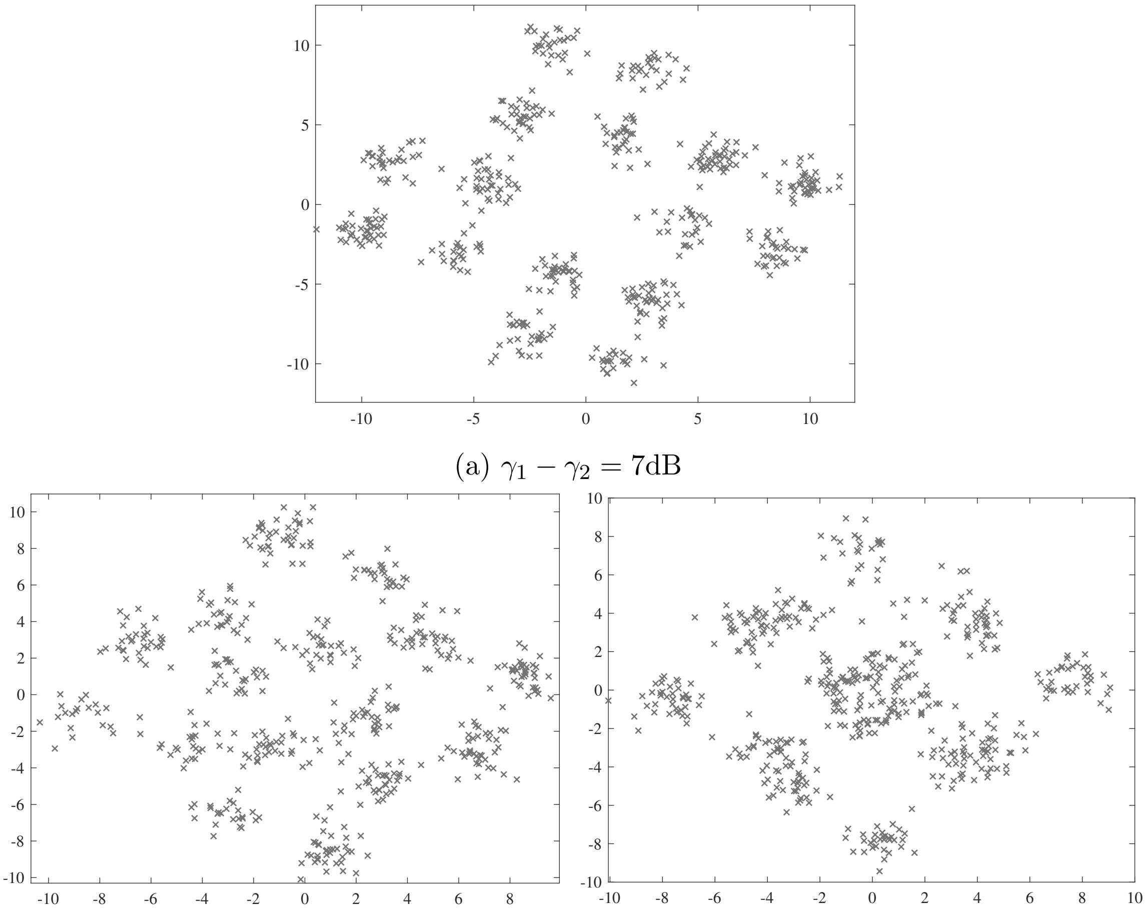
(c) $N_s = 100$

Figure 3.7: SER comparison of the proposed GMM-clustering-based and optimal MLD-based approaches for two-user NOMA when $\gamma_1 - \gamma_2 = 9$dB.

of channel rotation. I can find the exact quadrant of each cluster by using only two pilot symbols. Fig. 3.7 also shows that the theoretical predictions of (3.3.8) and (3.3.9) are reasonably accurate.

Impact of Difference in User Powers

When the received powers from the users are similar, the distance between the clusters associated with the weaker user increases. This can lead to four close clusters around



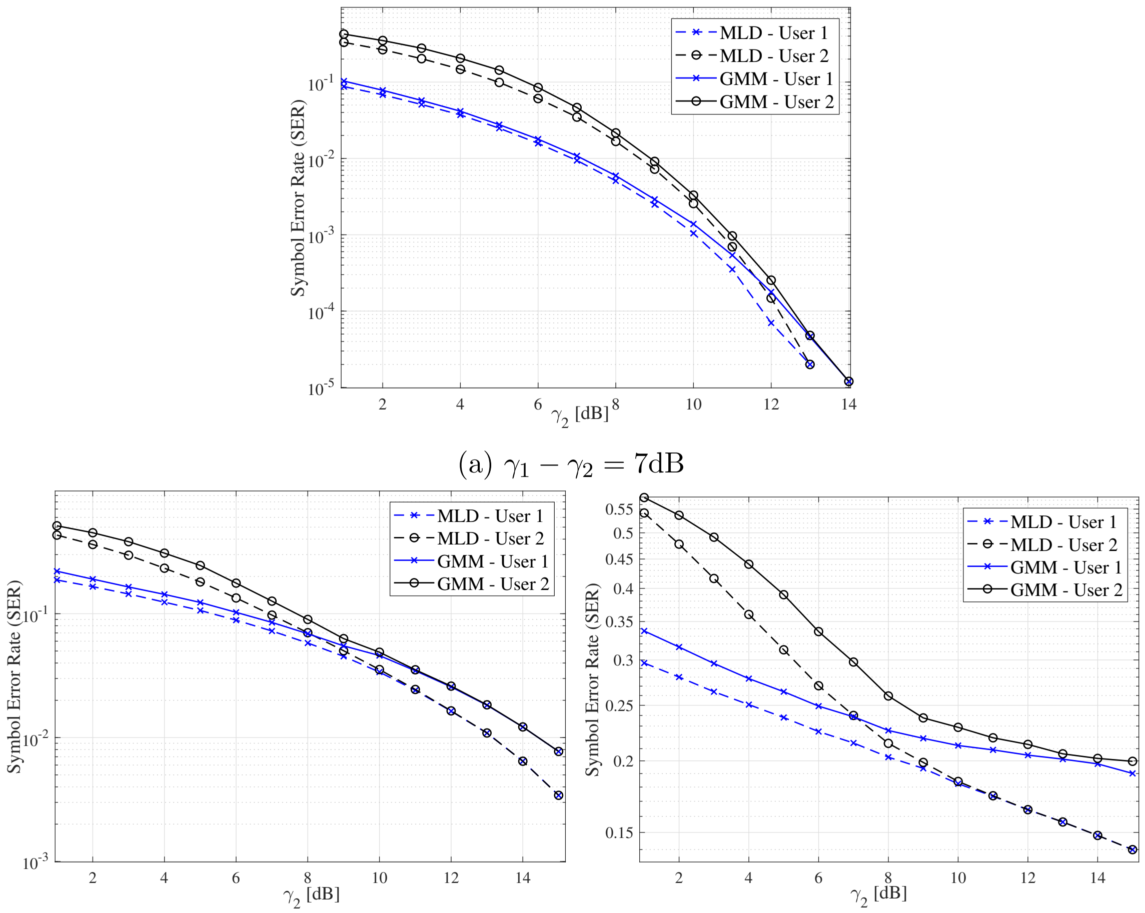
(a) $\gamma_1 - \gamma_2 = 7$dB

(b) $\gamma_1 - \gamma_2 = 5$dB

(c) $\gamma_1 - \gamma_2 = 3$dB

Figure 3.8: Constellation of received signals in a two-user NOMA scenario for three values of user power difference when $\gamma_2 = 10$dB.

the center, each belonging to a different user. Distinguishing these clusters can be hard for the BS as they may be too close to each other or even overlap. Fig. 3.8 illustrates such a case. As seen in Fig. 3.8a, when the difference in the power of two users is adequately large, both users can detect their signals correctly as the received signals form distinct clusters. However, as the power difference between the users decreases (Figs. 3.8b and 3.8c), the clusters associated with the weaker user(s) grow farther from each other. Consequently, four not-so-distinct clusters appear around



(a) $\gamma_1 - \gamma_2 = 7$dB

(b) $\gamma_1 - \gamma_2 = 5$dB

(c) $\gamma_1 - \gamma_2 = 3$dB

Figure 3.9: SER performance of the proposed GMM-clustering-based and optimal MLD-based approaches for the considered two-user NOMA scenario.

the center that belong to different mappings.

In Fig. 3.9, I present the SER performance of the proposed approach in a two-user NOMA scenario for three values of user power difference. One can see from Fig. 3.9a that when the user power difference is sufficiently large, the BS can detect the symbols correctly even with low SNR. However, as the user power difference decreases, the SER deteriorates for both users. The SER performance of a two-user NOMA when the user power difference is relatively low is shown in Figs. 3.9b and Fig. 3.9c. I

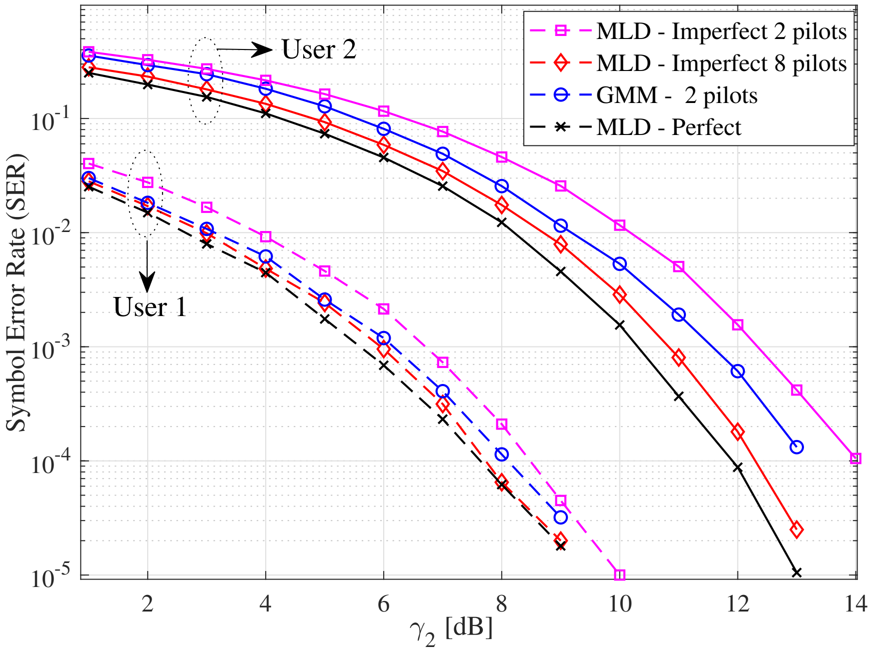


Figure 3.10: SER performance of the proposed GMM-clustering-based approach, MLD-based approach with full CSI, and MLD-based approach with imperfect channel estimation for a two-user NOMA scenario when $\gamma_1 - \gamma_2 = 9$dB and $N_s = 100$.

observe that the proposed algorithm can detect the signals of both users even when the users' received signal powers at the BS are close to each other. It is important to note that in power-domain NOMA, the power of the signals from different users should be substantially different. Otherwise, the BS will not be able to distinguish the signals. This is clear in Fig. 3.9c, which shows that, when the user power difference is 3dB, even the optimal MLD-based receiver performs poorly.

Comparison with Semi-blind Channel Estimation

I have so far compared my proposed joint estimation and detection approach with the optimal approach based on MLD with full CSI. However, in real-world scenarios, obtaining full CSI with only a limited number of symbols is infeasible. Fig. 3.10 shows the performance of the proposed algorithm when two symbols are used for demapping

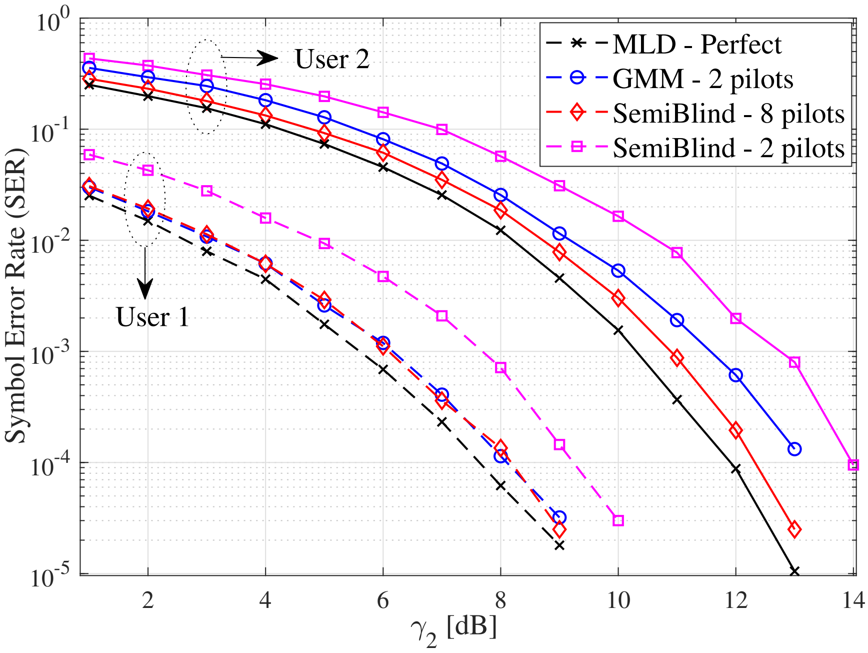


Figure 3.11: SER performance of the proposed GMM-clustering-based approach and the one based on semi-blind channel estimation for a two-user NOMA scenario when $\gamma_1 - \gamma_2 = 9$dB and $N_s = 100$.

in comparison with the MLD-based approach with full CSI, MLD-based approach using two training symbols for channel estimation, and MLD-based approach using eight training symbols for channel estimation. As the results show, the performance of MLD with two symbols used for channel estimation is significantly inferior to the proposed algorithm using the same two pilot symbols. For the MLD-based approach to attain a performance close to that of our proposed algorithm, it requires at least eight pilot symbols to estimate the channel sufficiently accurately. This means for a packet length of 100 symbols, the proposed algorithm offers at least a six percent improvement in throughput.

Fig. 3.11 shows the performance of our proposed algorithm in comparison with a receiver based on the semi-blind channel estimation method proposed in [138]. I observe that, with the same number of pilot symbols, the proposed algorithm offers

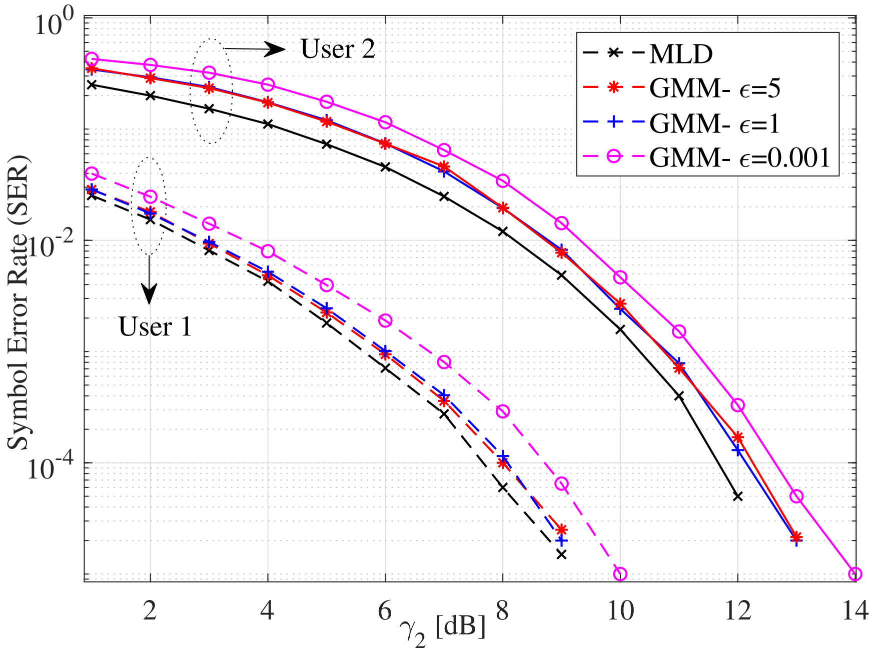


Figure 3.12: SER performance of the proposed approach in a two-user NOMA scenario with different values of ϵ when $\gamma_1 - \gamma_2 = 9$dB and $N_s = 500$.

considerably better performance for all users. For the strongest user, the approach based on semi-blind channel estimation needs at least eight pilot symbols to perform on a par with the proposed algorithm. This means the proposed algorithm can deliver at least six percent higher throughput. Note that the semi-blind approach using eight pilot symbols may offer better SER performance for the weaker user.

Impact of the Convergence Threshold ϵ

One of the parameters that should be considered when using the proposed algorithm is the convergence threshold ϵ. This parameter has a direct impact on the time complexity of the proposed algorithm as well as its SER performance. As shown in Fig. 3.12, when the value of ϵ is small, not only the speed of the algorithm decreases, but also its performance degrades due to possible over-estimation. As the

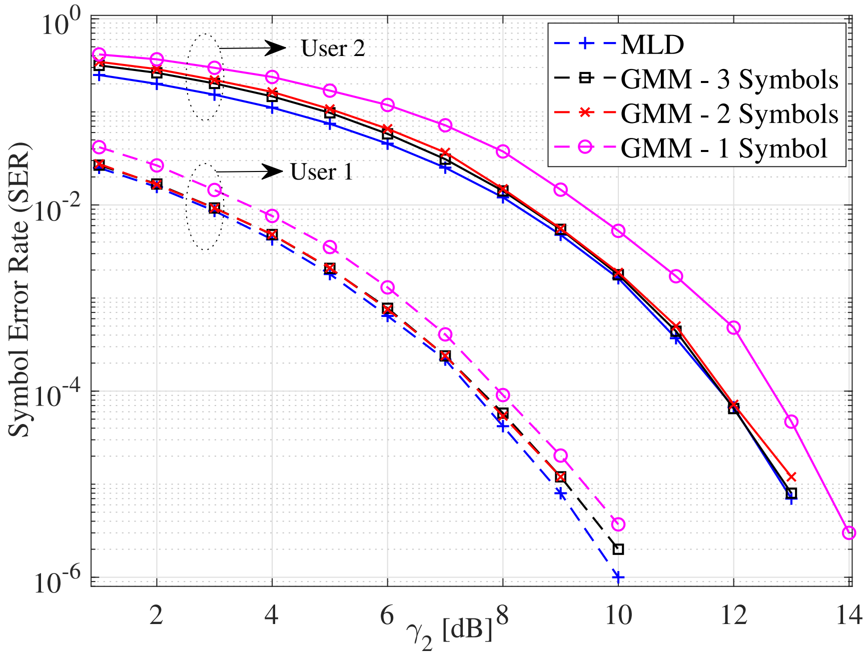


Figure 3.13: SER performance of the proposed approach and the optimal MLD-based approach with full CSI for a two-user NOMA scenario with different numbers of pilot symbols when $\gamma_1 - \gamma_2 = 9$dB and $N_s = 500$.

convergence threshold increases, both the SER performance and convergence speed improve. However, the threshold ought to be set carefully as increasing it excessively may result in under-estimation.

Impact of Pilot Symbol Number on Symbol-to-bit Demapping

As I mentioned earlier, in the high SNR regime and when the received powers from the users are sufficiently different, a single pilot symbol is enough for demapping. With the QPSK modulation and using the one-symbol demapping pilot, I can identify the clusters belonging to each quadrant. Afterward, considering the centroid of each cluster, I can calculate the phase and modulus of each channel. I demonstrate the performance of the one-symbol-based detection in Fig. 3.13. It is evident that, by using one symbol for demapping, the SER performance for the stronger user can

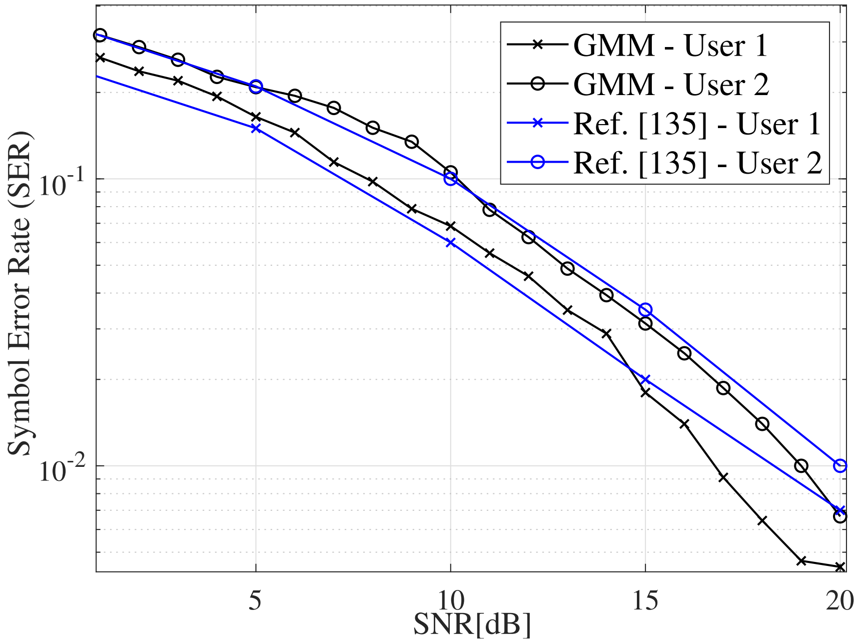


Figure 3.14: SER performance comparison of the proposed algorithm with the algorithm proposed in [135].

approach that of the MLD-based approach with full CSI. In addition, there is less than a 1dB difference between the SER performance for the weaker user and the optimal MLD-based detection.

Comparison with existing works

The error performance of uplink NOMA has been investigated in previous research [135, 139, 140]. In Fig. 3.14, I compare the proposed algorithm with the approach of [135], where deep learning is utilized for multi-user detection in uplink NOMA. I consider that each user employs four pilot symbols. In my algorithm, I use a packet length of 100 symbols, while in [135], the authors consider packets of 12 symbols. As seen in Fig. 3.14, my algorithm performs similarly to [135].

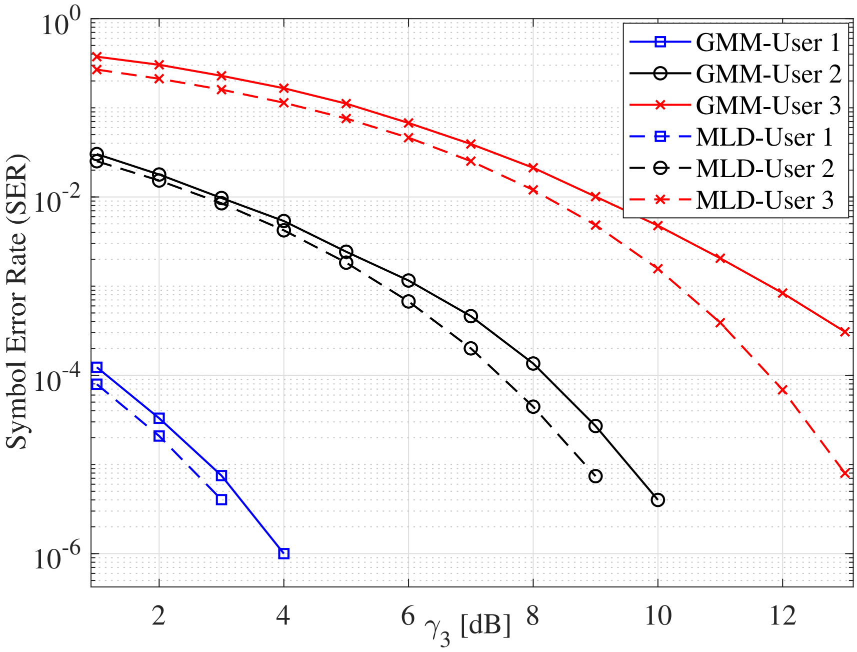


Figure 3.15: SER performance of the proposed GMM-clustering-based approach and the MLD-based approach with full CSI for a three-user NOMA scenario when $\gamma_1 - \gamma_2 =$ 9dB, $\gamma_2 - \gamma_3 =$ 9dB, $N_s = 500$, and $\epsilon = 5$.

3.4.3 Higher Numbers of Users

To evaluate the performance of the proposed algorithm when the number of users increases, I consider a three-user NOMA scenario. Fig. 3.15 shows the SER performance of the proposed algorithm for three-user NOMA when the number of symbols is $N_s = 500$ and three pilot symbols are used for symbol-to-bit demapping. It is clear that the proposed algorithm detects the signals with good accuracy. I note that in power-domain NOMA, usually two or three users are paired. This is because the SIC at the receiver cannot successfully detect user signals, if their powers do not differ sufficiently [141]. Maintaining large power differences among a large number of paired signals is impractical.

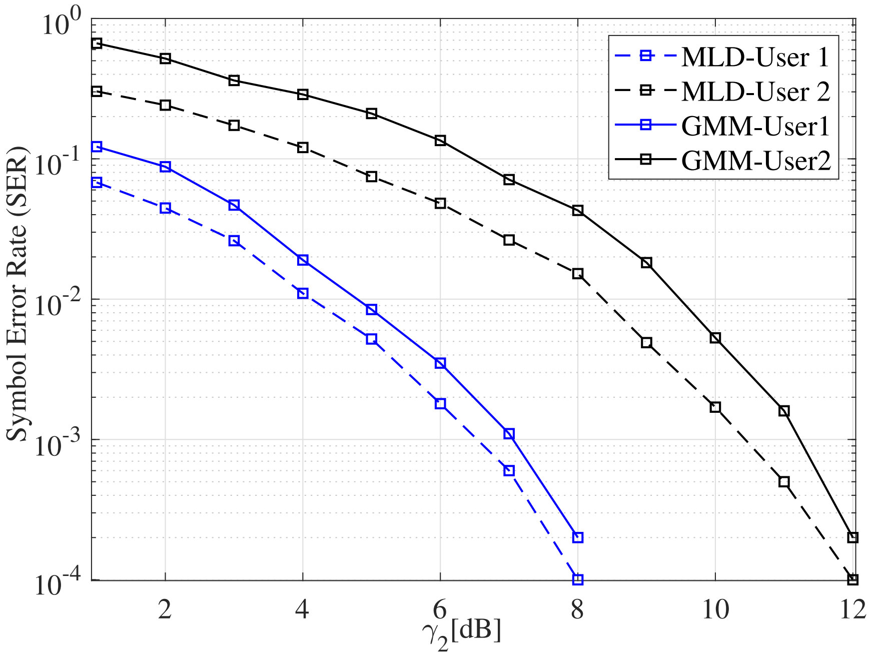


Figure 3.16: SER performance of the proposed algorithm in a two-user NOMA scenario when user 1 and user 2 employ 16-QAM and QPSK modulation schemes, respectively, $\gamma_1 - \gamma_2 = 15$dB, and $N_s = 500$.

3.4.4 Higher-order Modulations

To demonstrate that the proposed algorithm can be used with any modulation scheme, I consider a two-user NOMA scenario where user 1 utilizes the quadrature amplitude modulation (QAM) of the order 16 and user 2 utilizes the QPSK modulation scheme. As before, I benchmark the SER performance of the proposed algorithm, which uses two pilot symbols, against that of the optimal MLD-based approach with full CSI. Fig. 3.16 shows that the proposed algorithm with only two pilot symbols performs nearly as well as the MLD-based approach with full CSI. For the MLD-based approach, the BS needs to obtain the CSI for all users using long pilot sequences. However, the proposed algorithm only needs two pilot symbols, which are used to determine the symbol-to-bit demapping.

3.5 Grant-free Transmission: A Practical Scenario

Thus far, I have assumed that the number of users is fixed, and the receiver knows this number. This information is used to determine the number of clusters in my proposed algorithm. In this section, I show that the proposed algorithm can be implemented in a real-world scenario where the receiver does not have any prior knowledge except for the modulation scheme used by the transmitters and their power levels. To ensure frame synchronization, I assume the BS periodically broadcasts beacon signals, allowing users to align their internal clocks and coordinate transmissions [142].

I slightly modified the proposed GMM-clustering-based algorithm (Algorithm 2) to cope with not knowing the number of users. The modified algorithm is summarized in Algorithm 3. In particular, I first find the average signal power $\mathbb{E}[||\mathbf{y}||^2]$. If it is larger than the noise power (line 2), I perform GMM clustering to assign the data into M_o clusters and detect the user signals. After applying SIC, I reevaluate the signal power and continue the above process until the signal power falls below the noise power[3]. Since BS lacks any prior information about users, I assume the pilot symbols for any transmitting user will be identical, denoted as $\text{Pilot}_{\text{GF}} = [s_0, s_0, s_0]$[4].

I have not considered adaptive modulation and coding (AMC) in my system model. To use AMC, the devices need to have an estimate of the channel to choose

[3] There are different approaches for discovering the number of clusters such as the elbow method [143], X-means clustering [144], and those based on certain information criteria [145]. Two main information-criterion-based approaches use the Bayesian information criterion (BIC) [146] and the Akaike information criterion (AIC) [147]. Both of these approaches are based on the penalized likelihood estimation method. AIC is known to be prone to overfitting, while BIC is prone to underfitting. By determining the modulation technique employed by the users, the BS is also capable of detecting the number of users based on this information.

[4] I acknowledge that the SER performance can potentially be improved through the design of a pool of pilots. However, it falls outside the scope of this chapter and requires substantial further research.

the proper modulation order and code rate. This means that their channels need to be estimated at the BS first and then feedback to the devices, or in the case of TDM, they can estimate their channels themselves. However, this is not realistic for mIoT due to the large number of devices and the fact that they want to send short payload data only. Moreover, in my proposed approach, I assume that the devices and the BS do not know the channels in advance; therefore, it is not possible to perform AMC.

I consider a grant-free scenario where the number of active users in not known to the BS. Each user randomly selects a power level from a pool of possible values $\mathbb{P}_p = \{P_1, P_2, \cdots, P_5\}$, and there is no cooperation. I assume each user performs power control such that its average received power at the BS equals the selected power level. The received signal at the BS at time instant i is then given by

$$y = \sum_{i=1}^{N_u} \sqrt{P_i} g_i x_i + w, \tag{3.5.1}$$

where $p_i \in \mathbb{P}_p$, g_i represents small-scale fading, $w \sim \mathcal{CN}(0, 1)$, and $\mathbb{E}[||x_i||^2] = 1$. The BS does not have any prior knowledge of N_u, g_i, or p_i, but knows $\mathbb{P}_p$, N_0, and the modulation scheme utilized by the users.

I first ignore the small-scale fading, i.e., I assume $g_i = 1$. I also consider five power levels calculated as $P_i = P_1 + (i - 1) \times 7\text{dB}$ for $i = 1, \cdots, 5$. The number of active users in each time frame (with the duration of one packet of length N_s symbols) is uniformly distributed between one and three. At each time slot, a random number of users start transmitting their signals. Fig. 3.17 shows the SER performance of Algorithm 2 when I disregard the small-scale fading and users apply power control. As can be seen in Fig. 3.17, the performance of our proposed model is matched with that of MLD.

I then consider a real-world scenario under the Rayleigh fading scenario ($g_i \sim$

Algorithm 3: GMM-clustering-based joint channel estimation and signal detection for an unknown number of users.

```
Input: Received signal $\mathbf{y}$, noise power $N_0$, modulation order $M_o$, Signal
       Constellation $\mathcal{S}$, and convergence threshold $\epsilon$
Output: Detected user signals
Set $w_{cj} = \frac{1}{M_o}, \;\; j = 1, \cdots, M_o$
while $P_y \succ N_0$ do
    Initialize $\hat{\boldsymbol{\mu}}^{(0)}$ and $\hat{\boldsymbol{\Sigma}}^{(0)}$ by dividing the received signals on the reference
      coordinate system into $M_o$ equal sections and select one point in each
      section as the initial mean and set the initial covariance to one.
    Calculate $\hat{\gamma}_{i,j}^{(0)}$ according to (3.2.1).
    Calculate the log-likelihood function according to (3.2.5).
    Set $t = 1$.
    while $l^{(t)} - l^{(t-1)} \geq \epsilon$ do
        Update $\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\Sigma}}^{(t)}$ using (3.2.2) and (3.2.3).
        Update $\hat{\gamma}_{i,j}^{(t)}$ according to (3.2.1).
        Update the log-likelihood function according to (3.2.5).
    end
    Return optimal $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$.
    Calculate the phase of each cluster centroid as $\phi_i = \tan^{-1}\left(\frac{\mathrm{Im}(\hat{\mu}_i)}{\mathrm{Re}(\hat{\mu}_i)}\right)$.
    Calculate the average phase as $\phi = \frac{\sum_{i=1}^{M_o} \phi_i - M_o \pi}{M_o}$ and channel amplitude
      $|\hat{h}_u| = \frac{1}{M_o} \sum_{i=1}^{M_o} \mathrm{abs}(\mu_i)$.
    Update the decision boundaries using $\phi$ and utilize the pilot symbols to
      map each cluster into the mapping bits.
    Demodulate the signal to symbols: $\hat{\mathbf{x}}_u = \mathrm{demod}(\mathbf{y})$.
    Re-modulate the user signal, multiply by the estimated channel, and
      subtract from the received signal: $\mathbf{y} \leftarrow \mathbf{y} - |\hat{h}_u| e^{j\phi} \hat{\mathbf{x}}_u$ and update $P_y$.
end
```

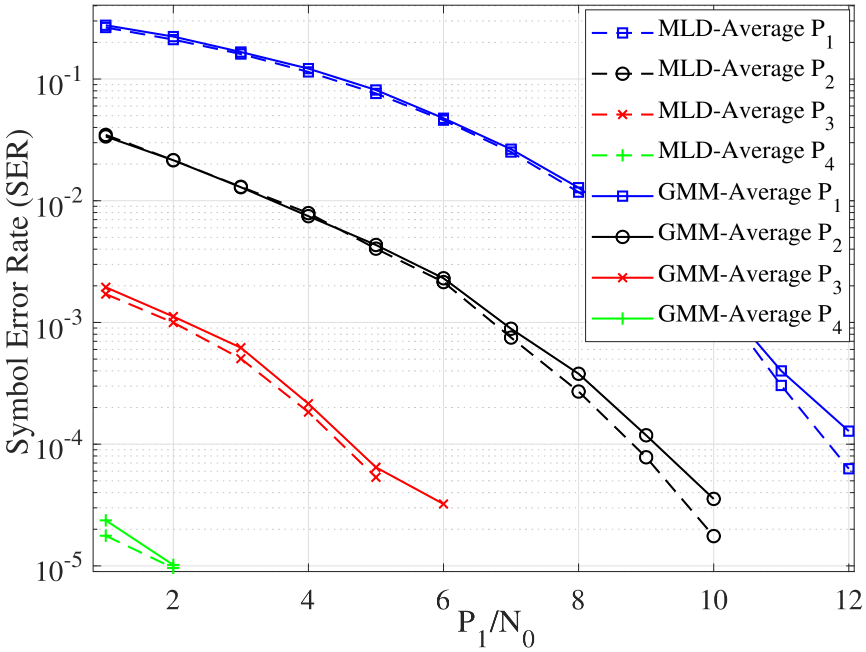


Figure 3.17: SER of the proposed algorithm versus MLD with full CSI for grant-free transmission with AWGN.

$\mathcal{CN}(0,1)$), where the power control does not exist, and users randomly select one of the power levels and start transmitting their packet. To simulate a practical, real-world scenario, I further reduced the power differential between users and considered a 5 dB difference in power between levels. Fig. 3.18 shows the throughput performance of the grant-free transmission with a SIC receiver based on Algorithm 3 and its comparison with the MLD-based receiver with full CSI. At the receiver, I apply Algorithm 3 and calculate the average throughput based on the number of users. The average throughput is the ratio of the number of symbols correctly detected at the receiver and the total number of transmitted symbols. In particular, I assume that a random number of users start transmitting their packets to the BS. BS applies Algorithm 3, detects each user's signals, and calculates the SER of each user. Let me assume the SER of user i as e_i, representing the ratio of erroneous symbols to the packet

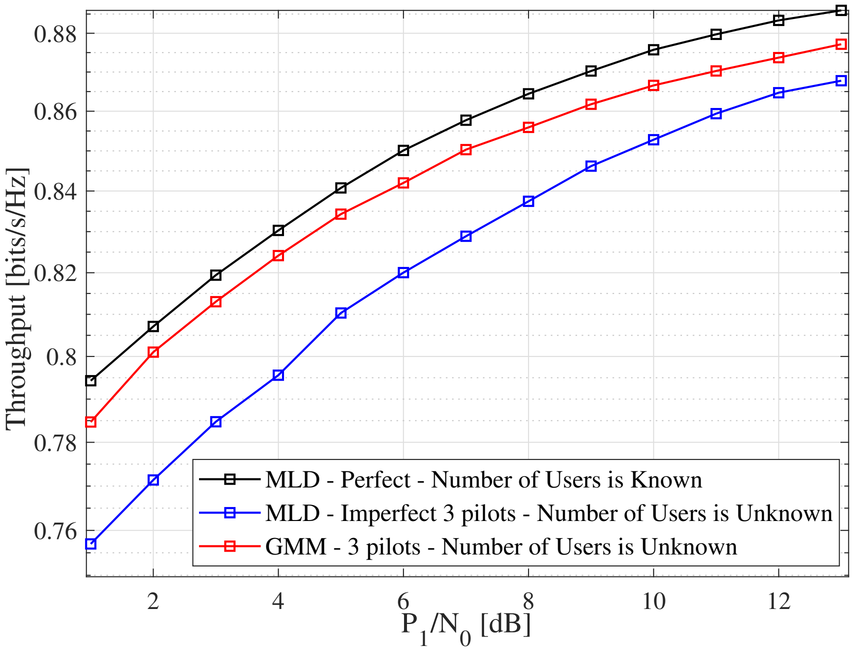


Figure 3.18: Throughput performance of the proposed GMM-clustering-based approach versus the MLD-based one with full CSI for Grant-free transmission with Rayleigh fading when $N_s = 500$, $\epsilon = 5$, and $N_0 = 1$.

length. The system's average SER is then expressed as $\frac{\sum_{i=1}^{N_u} e_i}{N_u}$ where N_u denotes the total number of users. Following this, I calculate the throughput of the system using $1 - \frac{\sum_{i=1}^{N_u} e_i}{N_u}$ [148]. In order to obtain the outcomes depicted in Figure 3.18, I performed a series of 1000 simulation iterations and calculated the average results. As seen in Fig. 3.18, the performance of the proposed algorithm with only three pilot symbols is close to that of the MLD-based approach with full CSI that has the knowledge of the number of users. For a fair comparison, I also simulated the case where MLD is imperfect, BS lacks information regarding the number of users, and the number of pilots is the same as my proposed GMM-based model. As can be seen, my proposed algorithm has significantly superior performance for the identical simulation setup.

3.6 Chapter Conclusion

In this chapter, I proposed to employ a semi-supervised ML algorithm, i.e., GMM clustering, to cluster the signals at the receiver for joint channel estimation and signal detection in grant-free NOMA. I applied a SIC strategy to detect the signals of each user. I compared the performance of my proposed GMM-clustering-based algorithm with that of the optimal detection method based on MLD, which requires full CSI at the receiver. I showed that by using two pilot symbols for each user, the proposed algorithm can reach the performance of the MLD-based approach with full CSI. I took one step further and used a single pilot symbol to estimate the channel of all users. The resulting performance was still close to that of the MLD with full CSI. To make a fairer comparison, I also compared the performance of my proposed algorithm with that of the MLD with imperfect channel estimation relying on a finite number of symbols. I observed that to attain a similar performance, the MLD-based approach needs at least eight pilot symbols for channel estimation to perform as well as my proposed algorithm using only two pilot symbols. Furthermore, to reduce the computational burden at the BS, I proposed a theoretical model to calculate the probability of error based on the maximum error of the EM algorithm utilized for GMM clustering. my simulation results showed that the proposed model predicts the SER of the proposed algorithm well. The simulation results also demonstrated that, when the number of transmitted symbols is moderate or large, the SER performance of the proposed algorithm is on a par with that of the optimal MLD. Finally, since the accuracy of the GMM clustering depends on the sample size, I showed that there exists a tradeoff between the accuracy of the proposed algorithm and the communication block length.

Chapter 4

Channel coding for Clustering-based Channel Estimation and Signal Detection

Building upon the findings from the previous chapter, where I leveraged GMM-based clustering but were still reliant on a minimal number of pilot symbols to ensure performance, this chapter aims to refine my approach further. I introduce the application of channel coding, particularly the use of RI codes, to eliminate the need for any pilot symbols entirely. Through this exploration, I demonstrate not only the feasibility but also the enhanced performance of my approach when contrasted with alternative methods like LDPC codes and the maximum-likelihood receiver. I aim to highlight how my approach, with the assistance of RI codes, can maintain similar error performance while improving the throughput efficiency in mIoT systems.

4.1 System Model

I study a UL NOMA scenario, where N_u users are paired to send their messages simultaneously to the BS. Users and the BS are equipped, each with a single antenna. I assume that each user i has a message $\mathbf{b}_i$ of length k_u bits, which uses a RI code $\mathcal{C}_{\text{RI}}$ with rate R to generate a codeword $\mathbf{c}_i$ of length $n_u = k_u/R$ bits. For the simplicity of notations, the encoding operation is denoted by $\mathbf{c}_i = \mathcal{C}_{\text{RI}}(\mathbf{b}_i)$. Utilizing an M-ary modulation with signal constellation set $\mathcal{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_{M_o}\}$ where $\frac{1}{M_o}\sum_{i=1}^{M_o} |\mathbf{s}_i|^2 = 1$, each user i's codeword will be modulated to generate a packet of length $N_s = n_u/\log_2(M_o)$ symbols, denoted by $\mathbf{x}_i$, and sent to the BS. The received superimposed signal at the BS, denoted by $\mathbf{y} \in \mathbb{C}^{N_s \times 1}$, can be expressed as:

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}, \tag{4.1.1}$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_{N_u}] \in \mathcal{S}^{N_s \times N_u}$ denotes the transmitted signal matrix, $\mathbf{h} = [h_1, \cdots, h_{N_u}] \in \mathbb{C}^{N_u \times 1}$ is the channel vector, $\mathbf{w} \sim \mathcal{CN}(0, \nu \mathbf{I}_{N_s})$ is the multivariate additive white Gaussian noise, $\mathbf{I}_{N_s}$ is the $N_s \times N_s$ identity matrix, and ν is the noise power. The channel between user i and the BS, denoted by h_i, is modeled by a zero-mean circular symmetric complex Gaussian random variable, i.e., $h_i \sim \mathcal{CN}(0, 1)$. I also consider block fading, that is, the channel between each user and the BS remains unchanged for each transmission frame of length N_s symbols. The SNR for user u, denoted by γ_u, is defined as $\gamma_u = |h_u|^2$. I further assume that the BS knows N_u, the total number of active users[1], and their modulation schemes, but the CSI is not available at users and BS, and BS attempts for joint channel estimation and signal detection.

[1] This assumption can be relaxed as the proposed algorithm is iterative and can continue to detect all active users.

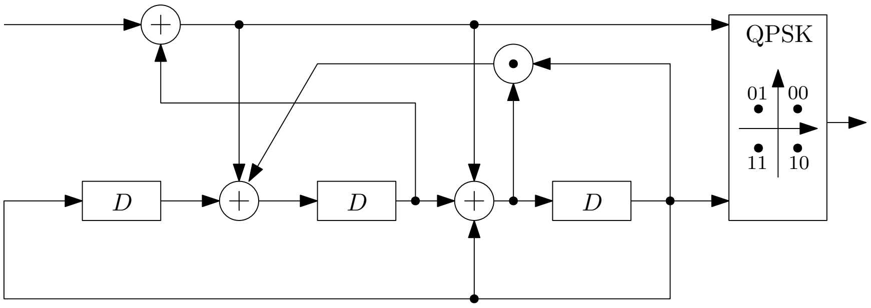


Figure 4.1: 90 degrees rate-1/2 RI encoder

4.2 Rotational Invariant Code

When the signal constellation has rotational symmetry, e.g. QPSK, the presence of channel phase rotation combined with the absence of pilot symbols, results in the receiver having no knowledge of which constellation point has been sent [149]. The time required to reestablish the correct demodulation phase for decoding can cause channel outages for considerable periods of time. In 1983, a full rotational invariant code by using nonlinear convolutional codes was proposed and applied to trellis-coded modulation with a 32° point signal constellation [150]. In RI codes, all rotated versions of a modulated codeword are valid codeword sequences, and all the rotated versions of the same sequence are produced by the same information input and subsequently decode back to the same information. Fig. 4.1 shows the encoder of a rate-1/2 90-degree RI code for QPSK modulation, where D represents the unit delay element.

The general invariant parity check equations (IPCE) for a rate 1/2 2-D trellis-coded modulation code is

$$H^0(D)y^0(D) \oplus (D^b \oplus D^a)y^1(D) \oplus D^b y^0(D) \cdot D^a y^0(D) = 0(D) \tag{4.2.1}$$

where $y^0(D), y^1(D)$ are the binary encoded sequences, $H^0(D)$ is binary polynomial and $0(D)$ is the all zero sequence. The binary polynomial also known as parity-check polynomial is in the form of

$$H^0(D) = h_0^v D^v \oplus \cdots \oplus h_0^1 D \oplus h_0^0 \tag{4.2.2}$$

where $h_0^i \in \{0,1\}$ for $0 \leq i \leq v$, and v is the maximum degree of all the parity check polynomials also known as the constraint length of the code. It is worth mentioning that (4.2.1) has a non-linear term $D^b y^0(D) \cdot D^a y^0(D)$, and as is shown in Fig. 4.1, an AND gate has been considered in the implementation. One can see that the encoder presented in the our paper has $a = 1$, $b = 2$, and $v = 3$. Considering $h^0 = 13_8 \equiv 1011_2 \equiv D^3 \oplus D \oplus 1 = H^0(D)$, the parity check equation will be

$$(D^3 \oplus D \oplus 1)y^0(D) \oplus (D^2 \oplus D)y^1(D) \oplus D^2 y^0(D) \cdot D y^0(D) = 0(D) \tag{4.2.3}$$

The rate 1/2 code generates two outputs given one input. The first output sequence of a systematic encoder is identical to the input sequence, while the second output sequence is generated using the parity-check equation. We choose $y^1(D)$ equal to the input sequence and based on IPCE equation (4.2.1), $y^0(D)$ can be represented in terms of delayed version of $y^1(D)$ and $y^0(D)$ as

$$\begin{aligned} y^0(D) =&(D^b \oplus D^a)y^1(D) \oplus (D^v \oplus h_0^{v-1} D^{v-1} \oplus \cdots \oplus h_0^1 D)y^0(D) \\ &\oplus D^j[D^{b-j} y^0(D) \cdot D^{a-j} y^0(D)] \end{aligned} \tag{4.2.4}$$

The value of $j, 0 \leq j \leq v$, must be selected such that the sequence $D^{b-j}y^0(D)$ and $D^{a-j}y^0(D)$ required to produce the nonlinear term can be generated with the minimum number of additional delay elements. One can see that for the encoder in Fig. 4.1, $j = 0$ and the $y^0(D)$ can be written as

$$y^0(D) = (D^2 \oplus D)y^1(D) \oplus (D^3 \oplus D)y^0(D) \oplus D^2 y^0(D) \cdot D^1 y^0(D) \tag{4.2.5}$$

Let us consider $x^1(D)$ as input. In order for $x^1(D)$ to be unaffected by a phase rotation, the precoding equation for $y^1(D)$ can be written as

$$y^1(D) = x^1(D) \oplus s^a(D) \oplus A(D)y^0(D) \tag{4.2.6}$$

where

$$A(D) = \bar{h}_0^1 D^{-a+1} \oplus h_0^2 D^{-a+2} \oplus \cdots \oplus h_0^a \tag{4.2.7}$$

$$s^a(D) = y^1(D) \oplus (D^{-a} \oplus h_0^1 D^{-a+1} \oplus \cdots \oplus h_0^a)y^0(D) \tag{4.2.8}$$

Since $h^0 = 13$, the value of $A(D)$ for the encoder in our paper is equal to zero. Therefore, $y^1(D)$ can be written as

$$y^1(D) = x^1(D) \oplus s^1(D), \tag{4.2.9}$$

It has been proven that a systematic encoder of rate 1/2 non-linear rotationally invariant sequence codes defined by (4.2.1) can be realized with κ binary delay elements where $\kappa = \max\{v, 2(b-a)\}$ [150].

This RI code is a non-linear convolutional code and can be effectively decoded by using the Viterbi decoder over the code trellis. Fig. 4.2 shows the state transition diagram of the rate 1/2 RI encoder presented in Fig. 4.1. For notation simplicity, I denote the decoding process of the RI code by $\hat{\mathbf{b}} = \mathcal{C}_{\mathrm{RI}}^{-1}(\mathbf{c})$.

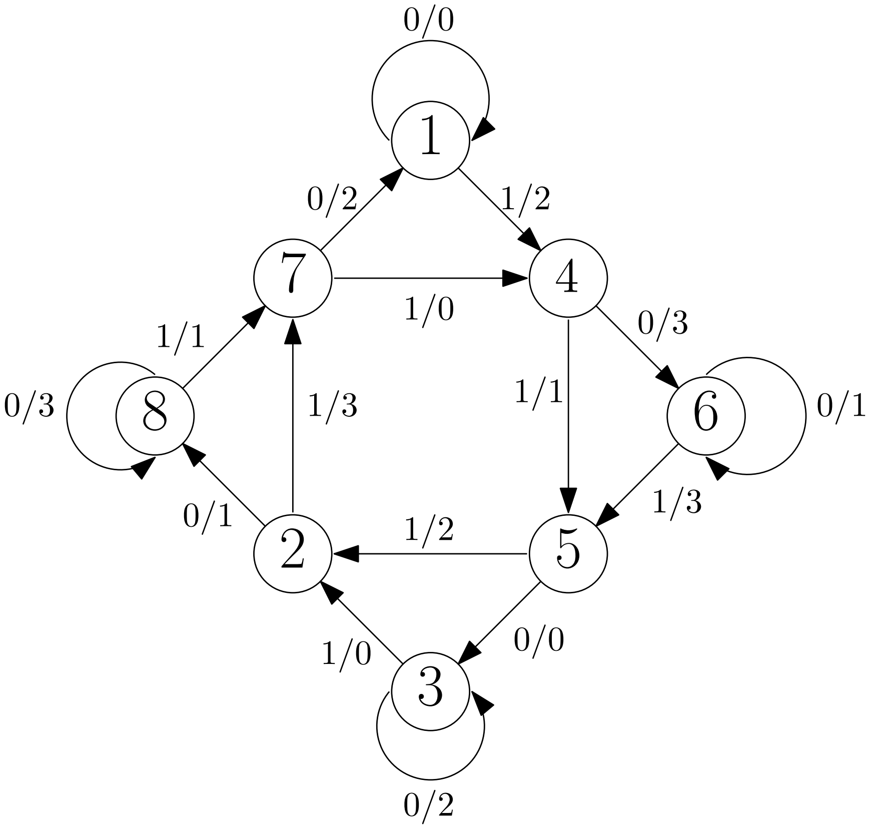


Figure 4.2: State Transition of 90 degrees RI rate 1/2 QPSK encoder (Fig. 4.1).

4.3 GMM-based Clustering algorithm for Joint Channel Estimation and Signal Detection Using RI Coding

Assuming that users' signals are randomly and uniformly drawn from the signal constellation $\mathcal{S}$, one can easily show that conditioned on the channel gains $\mathbf{h}$, the received signal at the BS at time instant i, denoted by y_i, have the following Gaussian mixture distribution:

$$y_i|\mathbf{h} \sim \frac{1}{|\mathcal{S}|^{N_u}} \sum_{\mathbf{s}\in\mathcal{S}^{N_u}} \mathcal{CN}\left(\mathbf{h}^T\mathbf{s}, \nu\right), \tag{4.3.1}$$

where $|\mathcal{S}|$ is the modulation order.

As is explained in the previous chapter, the proposed joint channel estimation and signal detection approach first clusters the received signals into $M_o = 4$ clusters, where M_o is the modulation order. By using the coordinates of cluster centroids, the amplitude and phase of h_1 can be estimated. Then, the signals of user 1 will be detected. By using SIC, user 1's signal will be subtracted from the receive signal, and the algorithm again cluster the residual signal into $M_o = 4$ clusters to estimate h_2 and detect user 2's signal. If more than 2 users are paired, the algorithm continues to detect all users' signals. It is important to note that estimating the channel phase in the above algorithm is not accurate. In the previous chapter, I considered sending a few pilot symbols to determine the demapping. Using such symbols will decrease the throughput and should be avoided if possible.

4.3.1 Applying Clustering Technique at the Receiver

I have adopted the GMM clustering technique for joint channel estimation and signal detection, with detailed explanations provided in earlier chapters. In a nutshell, given the observed data, the distribution's mean, weight, and covariance for each cluster will be determined. One can accomplish this goal by maximizing the likelihood function over all available data at the BS. To do this, the EM technique [109], which is well-suited to addressing maximum likelihood problems involving hidden latent parameters, will be used. Having a finite number of Gaussian mixtures, a closed-form expression for the parameters of the EM algorithm is possible [110].

It is assured that the EM method will converge to a local optima [110]. When users are using the same M-ary modulation and the users' signals are obtained evenly from the same constellation, the clusters' weight will be the same, i.e., $\omega_j = \frac{1}{M_o}$, $j = 1, \cdots, M_o$. Furthermore, utilising QPSK modulation and a SIC receiver, I only need to estimate four Gaussian distributions at each step of the SIC. This contributes to the reduction of computational complexity.

The proposed joint channel estimation and signal detection algorithm using RI coding is summarized in Algorithm 4. First, I initialize the mean and covariance of GMM clustering by running a few iterations (less than five) of the k-means clustering algorithm [109] (line 3). Then, I continue by finding the responsibility and log-likelihood function based on (3.2.1) and (3.2.5), respectively (lines 4 and 5). After that, I attractively calculate the centroids of clusters and covariance matrices (lines 7 to 11 in the Algorithm 4). After convergence, the phase of each cluster is obtained (line 13). To reduce the impact of phase rotation, I average the phase difference between the centroids of each cluster and their expected values (Step 14)[2]. I then adjust the decision boundaries and demodulate the received signal into symbols (line 15). I then use the Viterbi algorithm to decode the user's information (Step 16). Finally, I re-encode this user and modulate and multiply by the estimated channel and subtract it from the superimposed received signal (Step 17) and repeat the algorithm for the next user.

Algorithm 4 can be applied for the NOMA scenario with any number of users. Assuming the GMM converges after t iterations, for the general case of d-dimensional

[2] One can easily show that for a 2-user NOMA scenario, if the channels are known at the BS and $|h_1| \geq |h_2|$, for user 1, I have $\boldsymbol{\mu}_i^{\text{opt}} = h_1\mathbf{s}_i$ and $\sum_i^{\text{opt}} = (|h_2|^2 + \nu)\mathbf{I}_2$, and for user 2, I have $\boldsymbol{\mu}_i^{\text{opt}} = h_2\mathbf{s}_i$, $\sum_i^{\text{opt}} = \nu\mathbf{I}_2$.

Algorithm 4: GMM-based joint channel estimation and signal detection with RI Coding

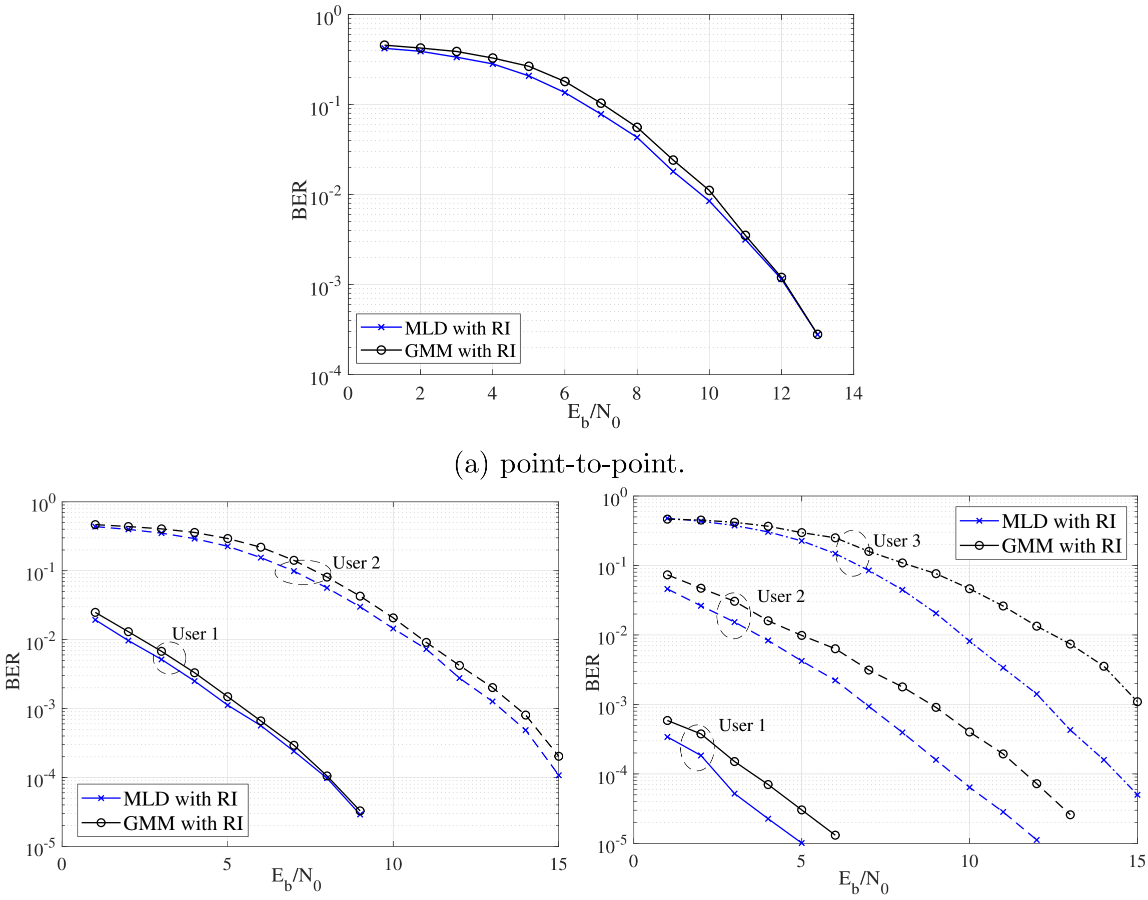
```
Input: Number of users $K$, received signal $\mathbf{y}$, modulation order $M$, Signal
       Constellation $\mathcal{S}$, and convergence threshold $\epsilon$
Output: Decoded information of each user ($\hat{\mathbf{b}}_u, \forall u$ ).
Set $\omega_j = \frac{1}{M}, \quad j = 1, \cdots, M$
for user $u = 1 : K$ do
    Initialize Find $\hat{\boldsymbol{\mu}}^{(0)}$ and $\hat{\boldsymbol{\Sigma}}^{(0)}$ by clustering the received signal into $M$
      clusters by sunning a few iterations of K-means clustering
    Compute $\hat{\gamma}_{i,j}^{(0)}$ (3.2.1), for $i = 1 : N$ and $j = 1 : M$
    Compute log-likelihoods based on (3.2.5)
    Set $t = 1$
    while $l^{(t)} - l^{(t-1)} \geq \epsilon$ do
        Update $\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\Sigma}}^{(t)}$ based on (3.2.2) and (3.2.3)
        Update $\hat{\gamma}_{i,j}^{(t)}$ based on (3.2.1)
        Update log-likelihoods based on (3.2.5)
    end
    Return $\hat{\boldsymbol{\mu}}$=$\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}^{(t)}$
    Find each cluster's phase: $\phi_i = \tan^{-1}\left(\frac{\mathrm{Im}(\hat{\mu_i})}{\mathrm{Re}(\hat{\mu_i})}\right)$
    Find clusters' average phase $\theta = \frac{1}{M}\sum_{i=1}^{M}\phi_i - \pi$ and channel amplitude
      $|\hat{h}_u| = \frac{1}{M}\sum_{i=1}^{M}\mathrm{abs}(\mu_i)$
    Update constellation decision boundaries based on $\theta$ and demodulate the
      signal to symbols: $\hat{\mathbf{x}}_u = demod(\mathbf{y})$
    Use the Viterbi algorithm to decode the user's information bit:
      $\hat{\mathbf{b}}_u = \mathcal{C}_{\mathrm{RI}}^{-1}(\hat{\mathbf{x}}_u)$.
    Re-encode this user, modulate and multiply by the estimated channel and
      subtract from superimposed received signal: $\mathbf{y} \leftarrow \mathbf{y} - |\hat{h}_u|e^{j\theta}\mathcal{C}_{\mathrm{RI}}(\hat{\mathbf{b}}_u)$
end
```

M-ary modulations, the computational complexity in the expectation step, which comprises of calculating the determinant and the inverse of the covariance matrix, is in the order of $\mathcal{O}(MN_s d^3)$. The maximization step of GMM consists of computing the weight, mean, and covariance for each cluster with the associated complexity of $\mathcal{O}(MN_s)$, $\mathcal{O}(MN_s)$, and $\mathcal{O}(MN_s d)$, respectively. The overall complexity of the proposed GMM-based clustering will be in the order of $\mathcal{O}(tM_o d^3)$ operations per symbol. The computational complexities for RI coding and decoding are similar for the GMM and MLD schemes. However, GMM necessitates additional computational steps due to its clustering requirements, while MLD is burdened by the need for extended pilot sequences. RI coding employs a non-linear convolutional code, with decoding performed via the Viterbi algorithm. Given $N_{\text{st}} = 2^3$ states and a truncation depth $T = 15$, the time complexity added by RI coding can be quantified as $\mathcal{O}(N_{\text{st}}^2 \times T)$. Correspondingly, the space complexity is $\mathcal{O}(N_{\text{st}} \times T)$.

4.4 Numerical Results

In this section, I evaluate the performance of the proposed clustering-based joint channel estimation and signal detection algorithm utilizing RI codes. For all simulations, I assume that each user has a packet of length $k_u = 50$ information bits to send to the BS, and a rate-1/2 RI code and QPSK modulation are employed by the users.

Fig. 4.3a shows the BER performance of the proposed joint channel estimation and signal detection approach using RI coding for a point-to-point communication scenario. As can be seen, the proposed approach performs very close to the ideal MLD with full CSI. Fig. 4.3b shows the performance of the proposed approach in a 2-user



(a) point-to-point.

(b) 2-user NOMA, $\gamma_1 - \gamma_2 = 9$dB.

(c) 3-user NOMA, $\gamma_1 - \gamma_2 = \gamma_2 - \gamma_3 = 9$dB.

Figure 4.3: BER performance of the proposed algorithm vs. MLD using rate 1/2 RI code (Fig. 4.1), when $N_s = 50$.

NOMA scenario with a 9dB power difference between the users. As demonstrated in this figure, the proposed approach is capable of accurately determining clusters and performing symbol detection with a BER that is nearly equal to the performance of the optimal MLD with full CSI at the BS. It is important to note that for all power-domain NOMA schemes, even with perfect MLD and perfect CSI at the receiver, there should be a relatively high power differences between users, so that the receiver can successfully detect them. This limits the number of users that can be paired in NOMA scenarios to only 2 or 3 users. Fig. 4.3c shows the performance of the

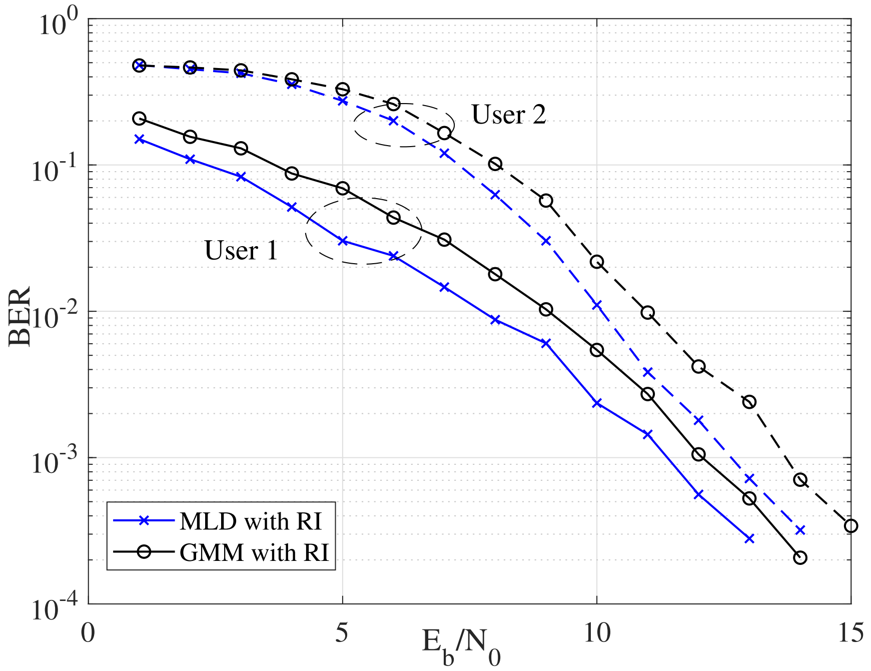


Figure 4.4: BER performance of the 2-user NOMA using rate 1/2 RI code, when $N_s = 50$ and $\gamma_1 - \gamma_2 = 7$ dB.

proposed approach to a 3-user NOMA scenario. The gap between the performance of the proposed approach and that of the optimal MLD is mostly related to the sub-optimal decision boundaries found by GMM due to the limited observation. Using the QPSK modulation for three users results in a total of 64 clusters. As I have considered packets of only 50 symbols, this means that there may exist some clusters without any signal point. In the previous chapter, it has been demonstrated that when the sample size N_s is sufficiently large, even when the number of users increases, the clustering technique performs extremely close to MLD with full CSI.

Fig. 4.4 shows the performance of the proposed scheme when the gap between the received signal powers is 7 dB. While the gap to the MLD with full CSI is slightly increased compared to that in Fig. 4.3b, the proposed scheme can achieve a reasonable performance without the need for any pilot symbol. The gap can be reduced even if

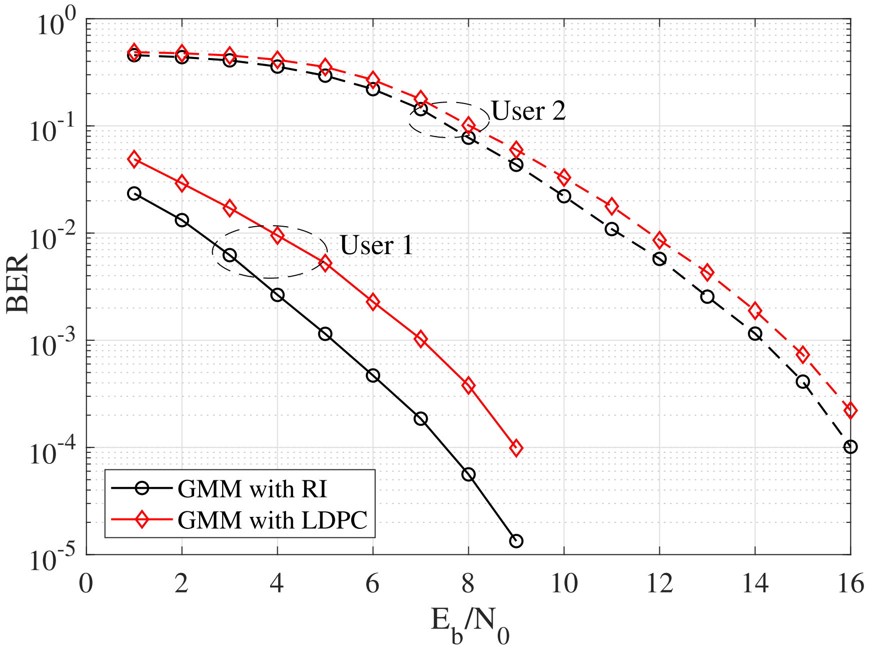


Figure 4.5: BER comparison of RI code with rate 1/2 (Fig. 4.1) versus LDPC code with rate 1/2 using GMM.

the power difference is lower, when N_s is increased.

For the sake of comparison, I consider another approach based on LDPC codes paired with the GMM-based clustering approach. The detection algorithm remains the same up until step 14 of Algorithm 4. I consider all possible (4 in the case of QPSK) channel rotations and perform LDPC decoding and find the syndrome. The proper mapping may be determined by picking the mapping with the most syndrome checked. This method has a very high computational complexity, particularly when the number of users increases. Fig. 4.5 shows the performance of the proposed joint channel estimation and signal detection algorithm with different coding techniques with rate 1/2. As can be seen, joint channel estimation and signal detection with RI coding outperforms that with LDPC coding in terms of BER. For the strong user, the gap between the proposed RI coding technique and LDPC coding with GMM is

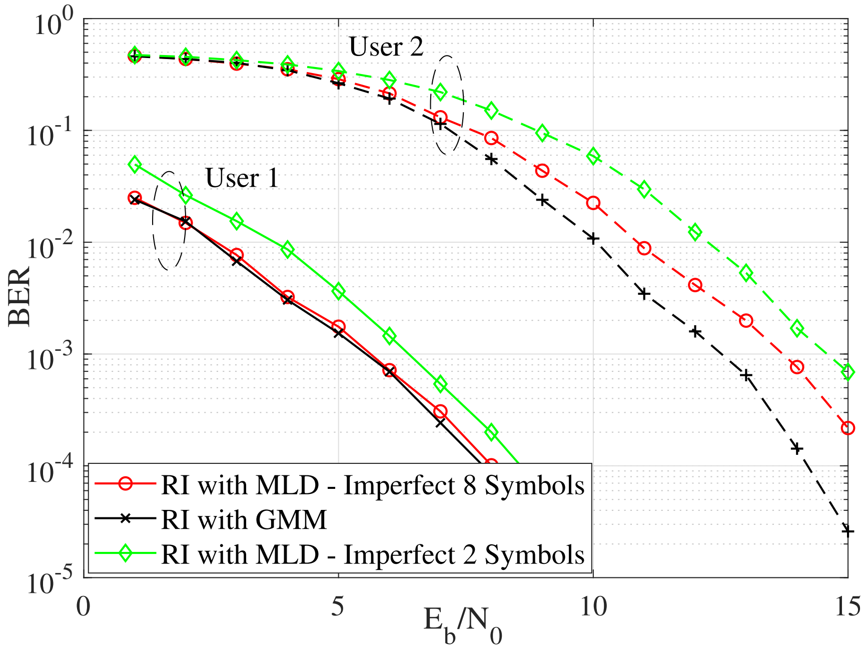


Figure 4.6: BER comparison of the GMM-based scheme and imperfect-MLD using rate 1/2 RI code (Fig. 4.1).

more than 2dB at the BER of 10^{-3}.

For MLD, attaining accurate CSI with a limited number of symbols is not viable in practice. As can be seen in Fig. 4.6, the MLD with only 2 training symbols performs poorly compared with our proposed technique without any pilot symbol. For MLD to have a BER performance similar to our proposed scheme, at least 8 training symbols should be used. However, this results in about 16% loss in throughput. Moreover, even with 8 training symbols, my proposed approach has better performance for the weak user.

Fig. 4.7 shows the BER performance of 2-user NOMA with RI coding for GMM clustering versus K-means clustering. As opposed to GMM clustering in Algorithm 1, K-means assumes that the covariance matrix for all clusters is the same. As can be seen, the GMM-based clustering outperforms the K-mean clustering approach.

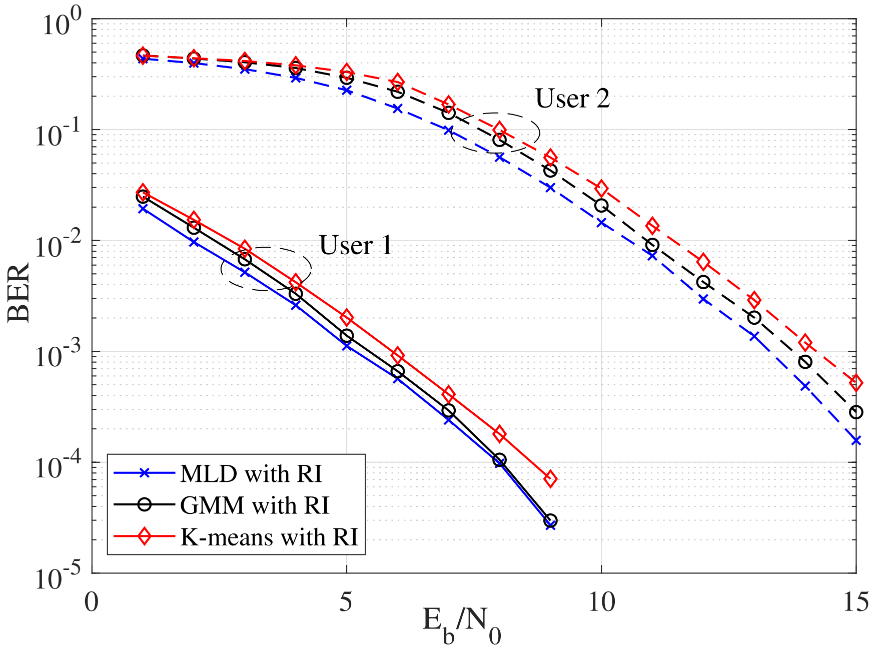


Figure 4.7: BER comparison of MLD vs GMM vs K-means for Two user NOMA using rate 1/2 RI code (Fig. 4.1).

4.5 Chapter Conclusion

By combining the clustering technique with rotational-invariant coding, this paper investigated the joint channel estimation and signal detection for the uplink of non-orthogonal multiple access without the use of any pilot symbols for channel estimation. In order to counteract the effects of channel rotation, I employ rotational invariant coding, which allows us to communicate without the use of pilot signals. I employ the GMM to cluster incoming signals without supervision and optimise decision boundaries in accordance with the clustering results in order to improve the BER. Results showed that the proposed scheme without any pilot symbols can achieve the same performance as the maximum-likelihood detector that needs to obtain full CSI to operate well.

Chapter 5

Federated Learning for Massive IoT

The focus of this chapter is to investigate the performance of distributed ML approaches in a realistic scenario in which communication channels are erroneous. This is motivated by mIoT applications, where in most cases, the goal is not to collect each and every data point at the server but to learn from data that is collected by many sensors. A distributed ML, or in particular the FL approach, performs this task in an iterative manner, where each device performs learning on its own local dataset using the global parameter shared by the BS and then shares the learning parameters with the BS to update the global parameter. This is repeated until convergence. I will show that even in the presence of communication errors, I can design effective approaches to maximize learning accuracy.

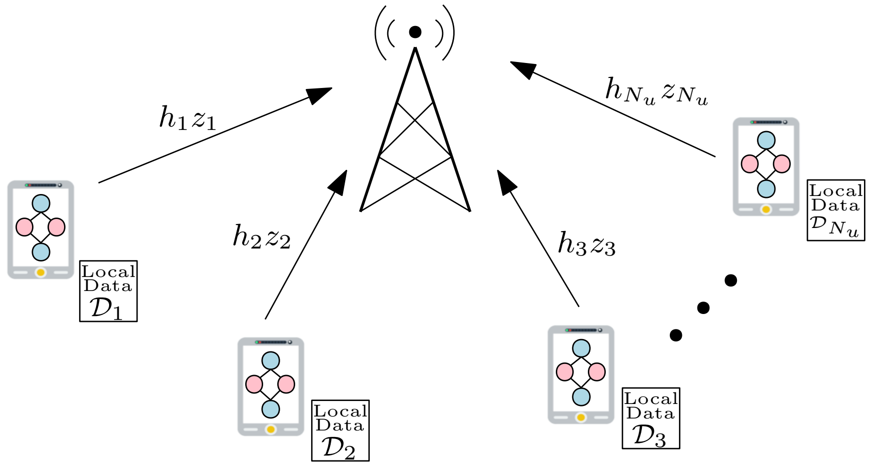

$h_1 z_1$
$h_2 z_2$
$h_3 z_3$
$h_{N_u} z_{N_u}$
Local Data $\mathcal{D}_1$
Local Data $\mathcal{D}_2$
Local Data $\mathcal{D}_3$
Local Data $\mathcal{D}_{N_u}$

Figure 5.1: System Model.

5.1 System Model

I consider an FL system comprising of one BS and a set $\mathcal{U}$ of N_u IoT devices with local datasets, $\mathcal{D}_1$, $\mathcal{D}_2$, ..., $\mathcal{D}_{N_u}$, as is shown in Fig. 5.1. The dataset of each device u is defined as $\mathcal{D}_u = \{(x_u^1, y_u^1), (x_u^2, y_u^2), \cdots, (x_u^{D_u}, y_u^{D_u})\}$, where D_u is the size of the user u's dataset. The total amount of training data stored by all users is stated as $D = \sum_{u=1}^{N_u} D_u$. The loss function, which varies depending on the learning model, is used to evaluate the FL algorithm's performance. For a linear regression learning model, the loss function can be written as $f(\boldsymbol{\omega}, \boldsymbol{x}, y) = \frac{1}{2} \parallel y - \boldsymbol{\omega}^T \boldsymbol{x} \parallel^2$, while in the case of neural network it is $f(\boldsymbol{\omega}, \boldsymbol{x}, y) = \frac{1}{2} \parallel y - f_{nn}(\boldsymbol{x}; \boldsymbol{\omega}) \parallel^2$, where $f_{nn}(\boldsymbol{x}; \boldsymbol{\omega})$ is the learning output of the neural network and $\boldsymbol{\omega}$ is the weight vector to be learned.

Suppose IoT devices wish to send their local updates in messages with a length of k bits to the BS. An encoder is used to map these messages to codewords of length n using a channel code of rate $R = k/n$. The received signal of user u at the BS is

$$r_u = h_u z_u + w, \tag{5.1.1}$$

where r_u is the received signal, z_u is the transmitted signal of user u, and w is zero-mean AWGN with variance σ^2 and h_u is the channel gain that follows a zero-mean circularly symmetric Gaussian distribution, i.e., $h_u \sim \mathcal{CN}(0, 1)$. I presume that all packets experience i.i.d Rayleigh fading. I further assume the channel is block fading, which is constant over each packet duration and independently varies across packets. For short packets, particularly in mIoT applications, the assumption of block fading is valid [128, 129]. Assuming the user's transmit power as P, i.e., $\mathbb{E}[|z_u|^2] = P$, the received SNR at the BS is $\gamma_u = \gamma_0 ||h_u||^2$, where $\gamma_0 = P/\sigma^2$.

Packet errors will arise as a consequence of block fading and channel noise. Here, I further simplify the communication channel as a packet erasure channel in which devices' packets are either erased with a probability of ϵ or successfully received at the BS with a probability of $1 - \epsilon$. I examine two distinct scenarios of short and long-packet communication. In the case of long packet transmission, the packet erasure rate can be accurately estimated by using the outage probability as follows:

$$\epsilon(\gamma, R) = \Pr(\gamma < \gamma_{\text{th}}) \tag{5.1.2}$$

where $\gamma_{\text{th}} = 2^R - 1$. For the short packet communication, authors of [46] showed that the packet error rate at the receiver can be written as

$$\epsilon(\gamma, n, R) \approx Q\left(\frac{nC(\gamma) - k + 0.5\log_2(n)}{\sqrt{nV(\gamma)}}\right) \tag{5.1.3}$$

where $C(\gamma) = \log_2(1 + \gamma)$ is the channel capacity, $V(\gamma) = \log_2^2(e)\left(1 - (1 + \gamma)^{-2}\right)$ is the channel dispersion, and $Q(.)$ is the standard Q-function [46].

While uplink transmissions are erroneous, I assume the BS utilizes the whole spectrum and transmits with high power on the downlink; therefore the error in the downlink direction is negligible.

5.1.1 Principles of FL in an Error-free Scenario

In each iteration of FL, each device u computes its local update $\boldsymbol{\omega}_u$ (usually using a few epochs of gradient descent), and transmits its updated parameter of the trained model to the BS. Next, the BS calculates the average weight, $\boldsymbol{\omega}$, by aggregating all the local updates. Then, the BS broadcasts the updated global parameter to be used by devices for the next iteration of FL.

Let us consider the loss function of device u, which calculates the model error on its data set $\mathcal{D}_u$ as

$$F_u(\boldsymbol{\omega}) = \frac{1}{D_u} \sum_{i=1}^{D_u} f(\boldsymbol{\omega}, \boldsymbol{x}_u^i, y_u^i) \tag{5.1.4}$$

Employing the gradient descent (GD) approach, the local parameter of device u at time t can be computed as

$$\boldsymbol{\omega}_u^{(t)} = \boldsymbol{\omega}^{(t-1)} - \eta \nabla F_u \left(\boldsymbol{\omega}^{(t-1)} \right) \tag{5.1.5}$$

where η is the learning rate. Once the IoT device has computed its own local parameter, it will transmit the updated parameter to the BS through an error-free channel, and the BS will aggregate all of the received local parameters to compute the global update using

$$\boldsymbol{\omega}^{(t)} = \frac{1}{D} \sum_{u=1}^{N_u} D_u \boldsymbol{\omega}_u^{(t)}. \tag{5.1.6}$$

After calculating the global parameter, the BS will broadcast it throughout the network. One could combine the last two steps of FL and calculate the global update as

$$\boldsymbol{\omega}^{(t)} = \boldsymbol{\omega}^{(t-1)} - \frac{\eta}{D} \sum_{u=1}^{N_u} D_u \nabla F_u \left(\boldsymbol{\omega}^{(t-1)} \right). \tag{5.1.7}$$

5.1.2 FL in the Presence of Communication Errors

Given the SNR, γ, blocklength, n, and code rate, R, the probability that the BS does not receive the local parameters can be calculated. I will utilize this to formulate the FL updating rules in the presence of communication errors.

FL with Erroneous Communications without the BS Memory

The number of local parameters received by the BS during each communication round may vary depending on the channel quality. In general, when the BS does not have a memory to store past local/global updates, the global parameter at the $(t+1)$th iteration can be calculated as

$$\boldsymbol{\omega}^{(t+1)} = \frac{\sum_{u=1}^{N_u} I_u^{(t)} D_u \boldsymbol{\omega}_u^{(t)}}{\sum_{u=1}^{N_u} I_u^{(t)} D_u}, \tag{5.1.8}$$

where $I_u^{(t)} \in \{0,1\}$ indicates whether the BS receives the local update of user u at time t correctly, which follows a Bernoulli distribution:

$$I_u^{(t)} = \begin{cases} 1; & \text{with probability } 1 - \epsilon(\gamma_u, n, R), \\ 0; & \text{with probability } \epsilon(\gamma_u, n, R). \end{cases}$$

In this scenario, the number of local updates at the BS may vary across iterations[1].

FL with Erroneous Communications with the BS Memory

I will assume in this part that the BS has a memory to store model parameters.

BS Caches each User's Past Local Parameter In this setup, I suppose the BS has a memory dedicated to storing each user's most recent local parameter. To

[1] Partial recovery can be considered as an extension to the model. In such a case, the update rule would need to be altered. However, in the current model of this paper, I consider that if I am unable to recover the original k bits, I declare an error in the system.

compute the global parameter at each communication round, the BS employs the fresh local parameter for users with successful transmissions and reuses the stored local parameters for users with an erroneous channel, i.e., in the case of channel erasure, $\boldsymbol{\omega}_u^{(t)} = \boldsymbol{\omega}_u^{(t-1)}$. Therefore, the number of users that participate in the update of global parameter would be fixed. The global update can be computed as

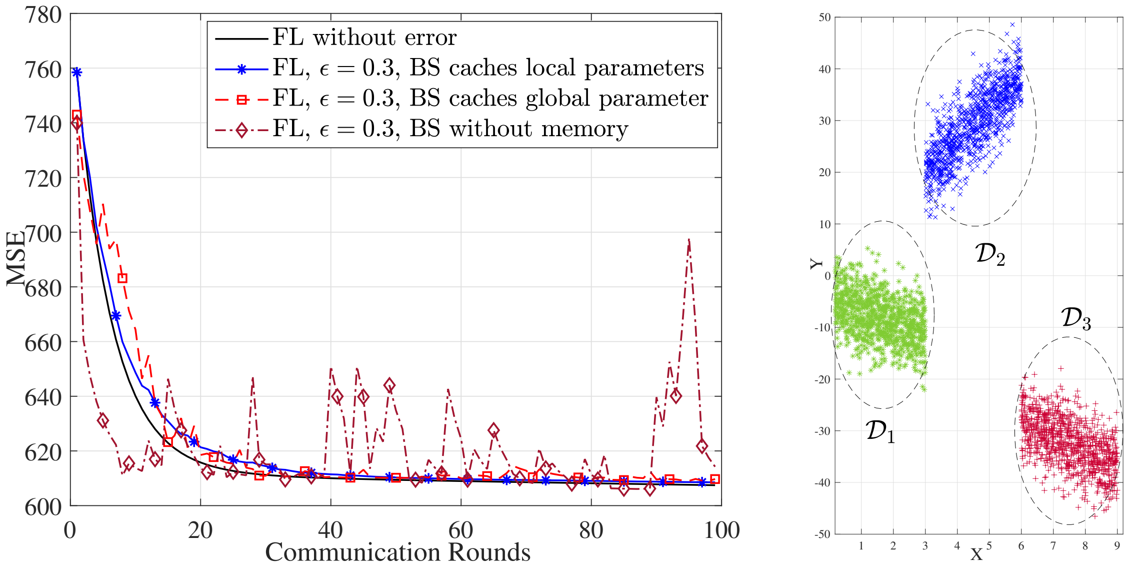
$$\boldsymbol{\omega}^{(t+1)} = \frac{1}{D} \sum_{u=1}^{N_u} D_u \left(\boldsymbol{\omega}_u^{(t)} I_u^{(t)} + \boldsymbol{\omega}_u^{(t-1)} \left(1 - I_u^{(t)}\right)\right). \tag{5.1.9}$$

BS Caches Global Parameters In an mIoT scenario, where a large number of users transmit their local updates to the BS, it may be problematic for the BS to maintain a dedicated memory pool for storing all users' past local updates. In this case, I assume that the BS has a limited memory that it utilises to preserve the previous m global updates. In this case, the global update at the $(t+1)$th iteration of FL can be calculated using:

$$\boldsymbol{\omega}^{(t+1)} = \frac{1}{D} \sum_{u=1}^{N_u} D_u \Big(\boldsymbol{\omega}_u^{(t)} I_u^{(t)} + \Big(1 - I_u^{(t)}\Big) \sum_{i=1}^{m} \alpha_{t-i} \boldsymbol{\omega}^{(t-i)} \Big), \tag{5.1.10}$$

where $\sum_{i=1}^{m} \alpha_{t-i} \boldsymbol{\omega}^{(t-i)}$ is the weighted average of last m global updates, and α_{t-i} represents the weight of the global parameter at time instant $t-i$, where $\sum_{i=1}^{m} \alpha_{t-i} = 1$.

Fig. 5.2 compares FL approaches in the presence of communication errors, where I assumed that $\epsilon_u = \epsilon$ for all users. It can be seen that the best performance is for the case where the BS has memory to store the past local parameters. When the BS has a restricted memory space, instead of saving all local parameters of IoT devices, the BS may store the previous m global updates (5.1.2), however, the performance deteriorates in comparison to the scenario when the BS memorises local values.



(a) MSE vs. Communication round

(b) Non-i.i.d. Dataset

Figure 5.2: Performance of FL with/without communication errors, when $U = 3$, $|D_u| = 100$, $\eta = 0.05$ and 2 iterations of gradient descent is used at each devices in iteration of FL.

5.2 Performance Analysis

5.2.1 FL in Erroneous Communication without the BS Memory

For the case of erroneous communications, where the BS does not have memory, the global update can be written as

$$\boldsymbol{\omega}^{(t+1)} = \frac{\sum_{u=1}^{N_u} I_u^{(t)} \boldsymbol{\omega}_u^{(t)}}{\sum_{u=1}^{N_u} I_u^{(t)}}, \tag{5.2.1}$$

where, I assume that all devices have the same training dataset size, i.e., $D_u = D/N_u$[2]. Let us consider the numerator of (5.2.1) as $Y = \sum_{u=1}^{N_u} I_u^{(t)} \omega_u^{(t)}$ where $I_u^{(t)} \sim \text{Bern}(1 - \epsilon_u)$, i.e. $\Pr(I_u^{(t)} = 1) = 1 - \epsilon_u$ and $\Pr(I_u^{(t)} = 0) = \epsilon_u$. The moment generating function

[2] The imbalance in dataset size does not have a considerable impact on the system's performance.

(MGF) of Bern$(1-\epsilon_u)$ can be written as

$$M_u(t) = \mathbb{E}[e^{tI_u}] = \epsilon_u + (1-\epsilon_u)e^t. \tag{5.2.2}$$

Subsequently, one can find the MGF of Y as

$$M(t) = \mathbb{E}\left[e^{tY}\right] = \prod_{u=1}^{N_u} \mathbb{E}\left[e^{t\omega_u^{(t)} I_u^{(t)}}\right] = \prod_{u=1}^{N_u} M_u\left(t\omega_u^{(t)}\right). \tag{5.2.3}$$

The cumulant generating function (CGF) of Y is given by

$$K(t) = \ln M(t) = \sum_{u=1}^{N_u} \ln\left(\epsilon_u + (1-\epsilon_u)e^{t\omega_u^{(t)}}\right). \tag{5.2.4}$$

I apply the saddle point approximation of the probability density function (PDF) of Y that can be written as [151]

$$\begin{aligned} f(t) &\approx e^{K(t)-tx_t}\sqrt{\frac{1}{2\pi K''(t)}} \\ &= \left(2\pi \sum_{u=1}^{N_u} \frac{\epsilon_u(1-\epsilon_u){\omega_u^{(t)}}^2 e^{t\omega_u^{(t)}}}{\left(\epsilon_u + (1-\epsilon_u)e^{t\omega_u^{(t)}}\right)^2}\right)^{-0.5} \\ &\times \prod_{u=1}^{N_u}\left(\epsilon_u + (1-\epsilon_u)e^{t\omega_u^{(t)}}\right)\exp\left(\frac{-t\epsilon_u(1-\epsilon_u)\omega_u^{(t)} e^{t\omega_u^{(t)}}}{\epsilon_u + (1-\epsilon_u)e^{t\omega_u^{(t)}}}\right), \end{aligned} \tag{5.2.5}$$

where $x_t = K'(t)$ and $K'(t), K''(t)$ denote the first and second differentiation of $K(t)$ relative to t, respectively.

Using the following lemma, one can easily calculate the moments and distribution of the BS's update (5.2.1).

Lemma 5.2.1. *Consider random variables O and H where H either has no mass at 0 (discrete) or has support $[0,\infty)$. The approximation of the first and second moments of the distribution of $G = O/H$ using the second order Taylor expansion, can be* *written as*

$$\mathbb{E}(G) \equiv \mathbb{E}[O/H] \approx \frac{\mu_O}{\mu_H} - \frac{\mathrm{cov}(O,H)}{\mu_H^2} + \frac{\mu_O \sigma_H}{\mu_H^3}, \tag{5.2.6}$$

$$\mathrm{var}(O/H) \approx (\frac{\mu_O}{\mu_H})^2 \left[\frac{\sigma_O^2}{\mu_O^2} - 2\frac{\mathrm{cov}(O,H)}{\mu_O \mu_H} + \frac{\sigma_H^2}{\mu_H^2} \right]. \tag{5.2.7}$$

Proof. Please refer to [152, 153] □

For the special case of mIoT, the distribution of (5.2.1) can be approximated with a Cauchy distribution.

Lemma 5.2.2. *In the mIoT scenario, the global updates* (5.2.1) *can be approximated with a standard Cauchy distribution.*

Proof. Let $\{I_1, \cdots, I_U, \cdots\}$ be a sequence of independent, but not necessarily identically distributed random variables, each with expected value of $1 - \epsilon_u$ and variance of $\epsilon_u(1-\epsilon_u)$. Taking into account $S_{N_u}^2 = \sum_{u=1}^{N_u} \epsilon_u(1-\epsilon_u)$, based on Lyapunov's central limit theorem (CLT), as N_u goes to infinity, the distribution of $\frac{1}{S_{N_u}} \sum_{u=1}^{N_u}(I_u - 1 + \epsilon_u)$ converges to a standard normal distribution $\mathcal{N}(0,1)$. In other words, the numerator and denominator of (5.2.1) can be approximated by zero mean Gaussian distributions. The distribution of the ratio of two independent random variables that both follow a Gaussian distribution with zero mean takes on the shape of a Cauchy distribution. As a result, a Cauchy distribution characterises the global update in massive IoT scenarios when the BS lacks memory. □

Now let us turn our focus on the convergence of FL in the presence of communication errors, when the BS does not have memory. Since $I_u^{(t)}$ follows a Bernoulli distribution, i.e., $I_u^{(t)} \sim \mathrm{Bern}(1-\epsilon_u), \forall u \in \mathcal{U}$, and the channels are independent, the probability mass function (PMF) of global parameter can be calculated as [154, 155]

$$\Pr\left\{\boldsymbol{\omega}^{(t+1)} = \frac{\sum_{u=1}^{N_u} I_u^{(t)}\boldsymbol{\omega}_u^{(t)}}{\sum_{u=1}^{N_u} I_u^{(t)}}\right\} = \prod_{u=1}^{N_u} \epsilon_u^{1-I_u^{(t)}}(1-\epsilon_u)^{I_u^{(t)}}. \tag{5.2.8}$$

One can easily see that the random characteristic of erasure occurrences is critical to (5.2.8), and results in fluctuation of the global parameter. Even though the instantaneous loss of the FL algorithm with erroneous communications without BS memory fluctuates, I will prove that the expected value of the loss converges.

I assume that $F_u(.)$, for all $u \in \{1, \cdots, N_u\}$, is strongly convex with a parameter μ and L-smooth. For function $F_u : \mathbb{R}^d \to \mathbb{R}$ that is L-smooth, $\forall v_1, v_2 \in \mathbb{R}^d$ [156]:

$$F_u(v_1) \leq F_u(v_2) + \nabla F_u(v_2)^T(v_1 - v_2) + \frac{L}{2} \parallel v_1 - v_2 \parallel_2^2 . \tag{5.2.9}$$

One can easily show that, $F(\omega)$ is also L-smooth [155], where

$$F(\omega) = \frac{1}{N_u}\sum_{u=1}^{N_u} F_u(\omega). \tag{5.2.10}$$

Lemma 5.2.3. *Assuming $F(.)$ is convex and L-smooth, $\forall \omega \in \mathbb{R}^d$, and $F(\boldsymbol{\omega}) = \frac{1}{N_u}\sum_{u=1}^{N_u}\frac{1}{D_u}\sum_{i=1}^{D_u} f(\boldsymbol{\omega}, \boldsymbol{x}_u^i, y_u^i)$, there exist constants M and N such that*

$$\parallel \nabla f(\boldsymbol{\omega}, \boldsymbol{x}, y) \parallel_2^2 \leq M \parallel \nabla F(\boldsymbol{\omega}) \parallel_2^2 + N,$$

for all $(\boldsymbol{x}, y) \in \{(\boldsymbol{x}_u^i, y_u^i) : u \in \mathcal{U} \text{ and } i \in \mathcal{D}_u)\}$.

Proof. The proof of existence of such upper bound has been given in [157, Theorem 4.7]. □

It is worth mentioning that there are numerous articles that assumed the stochastic gradients are uniformly bounded which is called bounded dissimilarity, i.e. $\mathbb{E}[\parallel \nabla f(\omega) \parallel^2] \leq \sigma^2$. However, in [156], it has been shown that this claim is in contradiction with strong convexity.

Theorem 5.2.1. *Let me consider a FL scenario with N_u devices with the data set $\mathcal{D}_u$ of size D_u, where the channel from device u to the BS is modeled by an erasure channel with erasure probability ϵ_u. I assume that the loss function $F(\omega)$ is strongly convex with a parameter μ and L-smooth. For $\mu \leq L$, $M < \frac{D}{\sum_{u=1}^{N_u} D_u \epsilon_u}$, and $\eta^{(t)} = \frac{1}{L}$, the expected total loss of the FL algorithm (5.2.1) converges as follows:*

$$\mathbb{E}\left[F\left(\omega^{(t)}\right) - F\left(\omega^{*}\right)\right] \leq CA^t + B\frac{1-A^t}{1-A}, \tag{5.2.11}$$

where

$$A = 1 - \frac{\mu}{L}\left(1 - \frac{M\sum_{u=1}^{N_u} D_u \epsilon_u}{D}\right), \tag{5.2.12}$$

$$B = \frac{N}{2L}\frac{\sum_{u=1}^{N_u} D_u \epsilon_u}{D}, \tag{5.2.13}$$

$$C = \mathbb{E}\left[F\left(\omega^{(0)}\right) - F\left(\omega^{*}\right)\right]. \tag{5.2.14}$$

and $\mathbb{E}(.)$ is the expectation with respect to the packet error.

Proof. The proof is provided in Appendix B.1. □

Remark 5.2.1. Since $A < 1$, when t goes to infinity, the expected loss will converge to

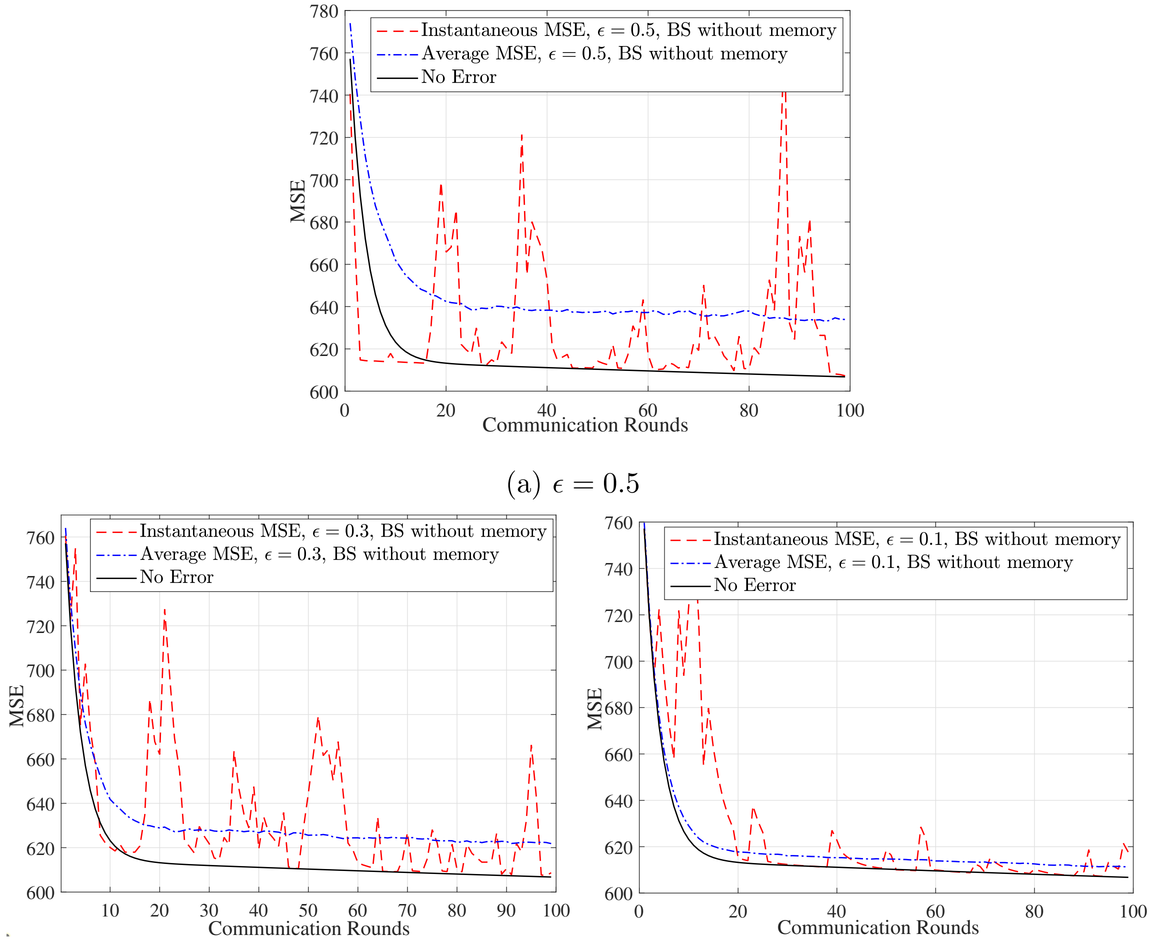
$$\mathbb{E}\left[F\left(\omega^{(\infty)}\right)\right] \leq F\left(\omega^{*}\right) + \frac{B}{1-A}. \tag{5.2.15}$$

Remark 1 shows that as t goes to infinity, the global loss converges to a ball around the optimal value with a radius of $\frac{B}{1-A}$. Let us assume that $D_u = D/N_u$ and $\epsilon_u = \epsilon$, $\forall u \in \mathcal{U}$. I then have $A = 1 - \frac{\mu}{L}(1 - M\epsilon)$ and $B = \frac{N}{2L}\epsilon$. Subsequently,

$$\mathbb{E}\left[F\left(\omega^{(\infty)}\right)\right] \leq F\left(\omega^{*}\right) + \frac{N\epsilon}{2\mu(1-M\epsilon)}, \tag{5.2.16}$$

where $M < 1/\epsilon$.

Remark 5.2.2. The gap to the optimal global loss increases with ϵ, since the radius of the ball around the minimum global loss, i.e., $\frac{N\epsilon}{2\mu(1-M\epsilon)}$, is an increasing function of ϵ. It is also clear that in the noiseless case, i.e., $\epsilon = 0$, the expected loss will converge to the optimal global loss.



(a) $\epsilon = 0.5$

(b) $\epsilon = 0.3$

(c) $\epsilon = 0.1$

Figure 5.3: Performance of FL in erroneous communication without the BS memory for non-i.i.d. dataset (Fig. 5.2b), when $N_u = 3$, $|D_u| = 100$, $\eta = 0.05$ and 3 iterations of GD are applied at each device.

Fig. 5.3 shows the global loss of the FL scenario with communication errors, when the BS does not have memory. As can be seen, the instantaneous MSE will fluctuate significantly with the increase of error probability ϵ, whereas the average MSE will converge. Moreover, the gap between the average MSE and the optimal global loss increases with ϵ, which confirms our finding in Remark 2.

5.2.2 FL in Erroneous Communication when the BS has Memory

In the preceding section, I demonstrated that the expectation of loss converges when the BS lacks memory. Although the expected loss converges, the instantaneous loss and the global parameters fluctuates, which means that the FL algorithm does not necessarily converge. In the following, I explore two instances in which the BS has memory and demonstrate that even in the presence of communication errors, the FL algorithm converges to the optimal global parameters.

BS Caches Users' Local Parameters

Now I consider the case where the BS has memory to store past local updates of all users. In such a scenario, the distribution of the global gradient is the weighted sum of two Bernoulli random variables. In particular, when $D_u = D/N_u$ for all users, I have

$$\begin{aligned}\omega^{(t+1)} &= \frac{1}{N_u}\sum_{u=1}^{N_u}\left(\omega_u^{(t)} I_u^{(t)} + \omega_u^{(t-1)}\left(1 - I_u^{(t)}\right)\right) \\ &= \sum_{u=1}^{N_u}\frac{\omega_u^{(t-1)}}{N_u} + I_u^{(t)}\frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u},\end{aligned} \tag{5.2.17}$$

where $I_u^{(t)} \sim \text{Bern}(1-\epsilon_u)$. Having the MGF of $I_u^{(t)}$ as $M_u(t)$, the MGF of $\alpha I_u^{(t)} + \beta$, which is a linear transformation of $I_u^{(t)}$, can be written as $e^{\beta t} M_u(\alpha t)$. First, I find $M_u(t)$ as follows:

$$M_u(t) = \mathbb{E}[e^{tI_u}] = \epsilon_u + (1-\epsilon_u)e^t. \tag{5.2.18}$$

Next, I calculate the MGF of the weighted sum of Bernoulli random variables, i.e., $\omega^{(t+1)}$, as

$$\begin{aligned} M(t) &= \prod_{u=1}^{N_u} \mathbb{E}\left(e^{t\left(\frac{\omega_u^{(t-1)}}{N_u} + I_u^{(t)} \frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u}\right)}\right) \\ &= \prod_{u=1}^{N_u} e^{t\frac{\omega_u^{(t-1)}}{N_u}} M_u(t\frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u}) \\ &= \prod_{u=1}^{N_u} e^{t\frac{\omega_u^{(t-1)}}{N_u}} \left(\epsilon_u + (1-\epsilon_u)e^{t\frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u}}\right). \end{aligned} \tag{5.2.19}$$

Hence, the CGF of $\omega^{(t+1)}$ can be computed as

$$K(t) = \sum_{u=1}^{N_u} \left(t\frac{\omega_u^{(t-1)}}{N_u} + \ln\left(\epsilon_u + (1-\epsilon_u)e^{t\frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u}}\right)\right). \tag{5.2.20}$$

Applying the saddle point approximation, the PDF of $\omega^{(t+1)}$ can be estimated as

$$\begin{aligned} f(t) \approx & \left[2\pi \sum_{u=1}^{N_u} \frac{\epsilon_u(1-\epsilon_u){\Psi_u^{(t)}}^2 e^{t\Psi_u^{(t)}}}{(\epsilon_u + (1-\epsilon_u)e^{t\Psi_u^{(t)}})^2}\right]^{-0.5} \\ & \times \prod_{u=1}^{N_u} \left(\epsilon_u + (1-\epsilon_u)e^{t\Psi_u^{(t)}}\right) \exp\left(-t\frac{(1-\epsilon_u)\Psi_u^{(t)} e^{t\Psi_u^{(t)}}}{\epsilon_u + (1-\epsilon_u)e^{t\Psi_u^{(t)}}}\right), \end{aligned} \tag{5.2.21}$$

where $\Psi_u^{(t)} = \frac{\omega_u^{(t)} - \omega_u^{(t-1)}}{N_u}$.

The following theorem establishes the convergence of the FL algorithm to the optimal global parameter, even in the presence of communication errors. By incorporating past local updates in cases of communication error, the BS successfully reaches the global minimum of the global loss function.

Theorem 5.2.2. *Let me consider a FL problem with N_u devices, where the channel from each device to the BS is modeled by an erasure channel with erasure probability ϵ. I assume that the local loss function $F_i(x)$ at device i is convex and L-smooth. I further assume that $||\nabla F(x) - \nabla F(y)||_2 \geq \mu||x-y||_2$, for all $x, y \in \mathbb{R}^d$, where $F(x) = \frac{1}{N_u}\sum_{i=1}^{N_u} F_i(x)$. Let $\delta_t = ||w^{(t)} - w^*||_2^2$, where $w^* = \arg\min_w F(w)$, and $\bar{\delta}_{t+1} = \frac{1}{t+1}\sum_{i=0}^{t}\delta_i$. For the FL algorithm (5.2.17), when $\epsilon \leq \frac{\mu}{2L}$ and $\eta = \frac{1}{L}$, $\bar{\delta}_k$ is upper bounded by:*

$$\bar{\delta}_t \leq \frac{F(w^{(0)}) - F(w^*)}{t\beta^2}, \quad \textit{for } t > 0, \tag{5.2.22}$$

where $\beta^2 = \frac{\mu^2}{2L} - 2L\epsilon^2$.

Proof. The proof is provided in Appendix B.2. □

In a mIoT setup, however, where the density of IoT devices might exceed $20,000$ per square kilometre, assuming such memory at the BS is impractical [158]. To tackle this scaling problem, I will propose a model in which, given a small number of memory units at the BS, the global parameter $\omega^{(t)}$ converges to its optimum value, ω^*.

BS Caches Global Parameters

Assume the server has limited memory; hence, it will store the previous m global updates. In such a scenario, the distribution of global gradient is again the weighted sums of Bernoulli distributions. For the case of a BS with m memory units, I can write

$$\begin{aligned}\omega^{(t+1)} &= \frac{1}{N_u}\sum_{u=1}^{N_u}\left(\omega_u^{(t)} I_u^{(t)} + \left(1 - I_u^{(t)}\right)\sum_{i=0}^{m-1}\alpha_{t-i}\boldsymbol{\omega}^{(t-i)}\right) \\ &= \sum_{i=0}^{m-1}\alpha_{t-i}\boldsymbol{\omega}^{(t-i)} + \sum_{u=1}^{N_u}\frac{I_u^{(t)}}{N_u}\left(\omega_u^{(t)} - \sum_{i=0}^{m-1}\alpha_{t-i}\boldsymbol{\omega}^{(t-i)}\right),\end{aligned} \tag{5.2.23}$$

where $\sum_{i=0}^{m-1} \alpha_{t-i} = 1$, and $I_u^{(t)} \sim \text{Bern}(1 - \epsilon_u)$. By calculating the MGF and CGF of $Y^{(t)}$, one can see that the saddle point approximation of the distribution is the same as (5.2.21) with $\Psi_u^{(t)} = \frac{\omega_u^{(t)} - \sum_{i=0}^{m-1} \alpha_{t-i} \boldsymbol{\omega}^{(t-i)}}{N_u}$.

In what follows, I show that the FL algorithm in the presence of communications errors, converges to the optimal parameters, when the BS has memory, even a single unit.

Theorem 5.2.3. *Consider a FL scenario with N_u devices, where the channel between each device and the BS is modelled by an erasure channel with erasure probability ϵ, and the local loss function $F_u(x)$ at device u is convex and L-smooth. Let us assume $||\nabla F(x) - \nabla F(y)||_2 \geq \mu ||x - y||_2$, for all $x, y \in \mathbb{R}^d$, where $F(x) = \frac{1}{N_u} \sum_{u=1}^{N_u} F_u(x)$. Considering $\delta_t = ||\omega^{(t)} - \omega^*||_2^2$, where $\omega^* = \arg\min_\omega F(\omega)$, and $\bar{\delta}_{t+1} = \frac{1}{t+1} \sum_{i=0}^{t} \delta_i$, for the FL algorithm* (5.1.10) *with single memory unit, when $\epsilon \leq \frac{\mu}{2L}$ and $\eta = \frac{1}{L}$, $\bar{\delta}_k$ is upper bounded by:*

$$\bar{\delta}_t \leq \frac{F(\omega^{(0)}) - F(\omega^*)}{t\beta^2}, \quad \text{for } t > 0, \tag{5.2.24}$$

where $\beta^2 = \frac{\mu^2}{2L} - 2L\epsilon^2$.

Proof. The proof is provided in Appendix B.3. □

Remark 5.2.3. According to Theorem 5.2.3 as the number of iterations t increases, the gap between the global parameter $\omega^{(t)}$ and the global minima ω^* decreases (since otherwise if $\delta_t \geq \epsilon_0$, where $\epsilon_0 > 0$, $\bar{\delta}_t$ will be always bounded above ϵ_0).

I emphasize that Theorem 5.2.3 and Remark 3 do not state that δ_t is a decreasing function of t; instead, it shows $\omega^{(t)}$ converges to w^* when t is sufficiently large even in the presence of communication errors.

5.3 Numerical Results

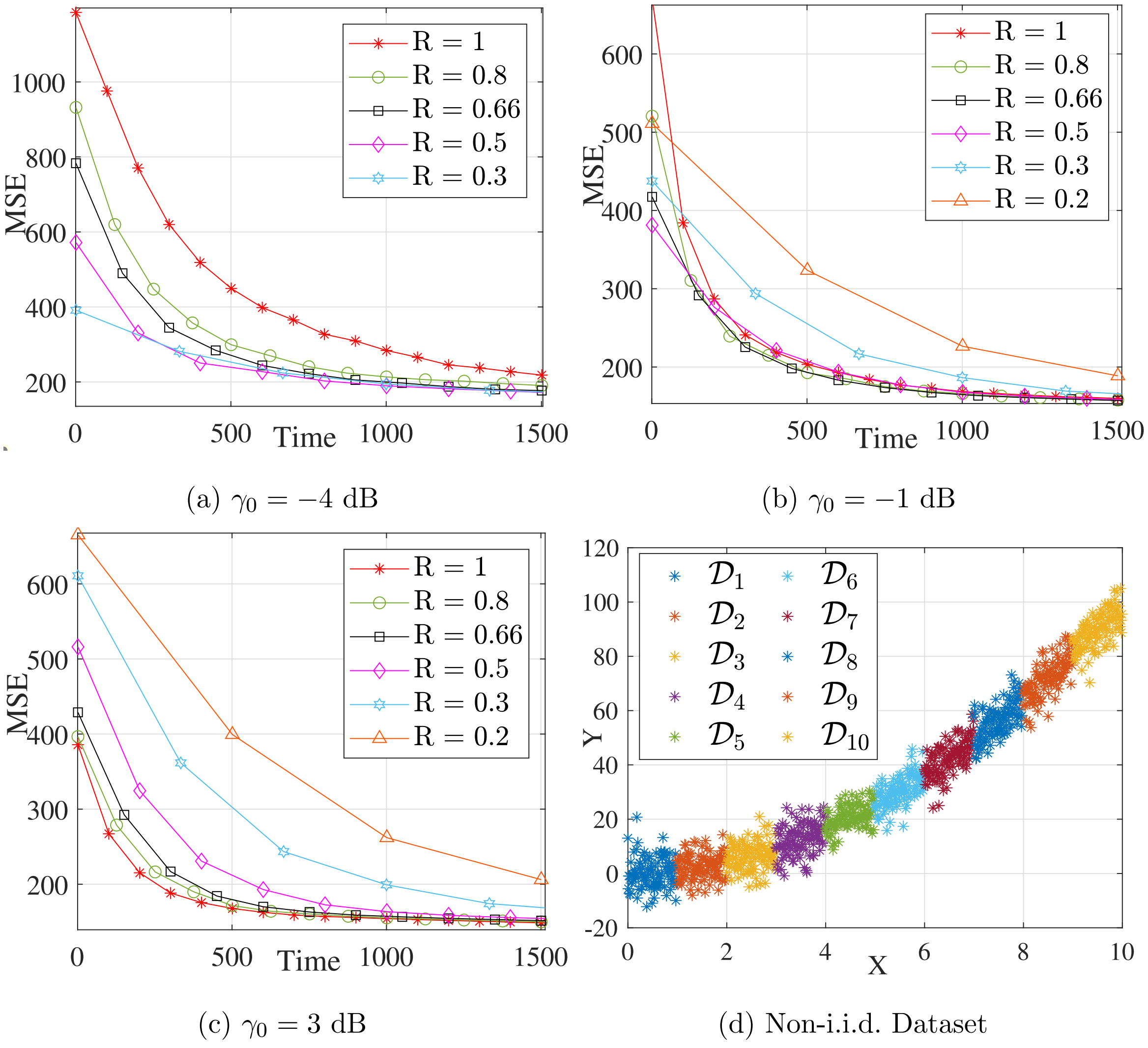
I start by investigating the performance of FL with short packet communications over Rayleigh fading channel. I choose a message length of $k = 100$ bits due to the fact that the normal approximation for AWGN channels is relatively good for packet lengths of $k \geq 100$ bits and $R \geq 0.5$ [46]. I consider there are $N_u = 10$ users in the mIoT scenario. As shown in Fig. 5.4d, the non-i.i.d. datasets are created using the non-linear model $y = x^2 + \lambda$, where $\lambda \sim \mathcal{N}(0, 5)$. The data set for each sub-figure in Fig. 5.4 and Fig. 5.5 has been regenerated.

I consider the symbol duration as the time unit. I assume for all approaches the devices use the same modulation and that the symbol duration is the same. When the rate R is reduced, the packet length increases and the time required to transmit the packet increases. The duration of each communication round is considered to be equivalent to the packet duration, i.e., n symbols. Thus, for example, 1000 time units corresponds to 10 rounds for a rate 1 code (assuming one symbol corresponds to 1 bit and each message is 100 bits, so 10 messages of 100 bits can be communicated) but only 5 rounds for a rate 1/2 code (as 200 symbols are sent in each round). Furthermore, Monte-Carlo simulations are used in short-packet communications, i.e. I ran 100 simulations and averaged the results. I assume all users are allocated with orthogonal radio resources; therefore, their simultaneous transmissions do not interfere with each other[3].

Fig. 5.4 and Fig. 5.5 show the global loss of FL based on the coding-rate R in short packet erroneous communication, when the BS stores local update of IoT

[3] A more general case in which radio resource allocation, with the possibility of non-orthogonal transmissions, can be considered to optimize the learning accuracy or the convergence time. This is, however, out of the scope of this chapter and will be considered by the authors in future works.

devices and global updates, respectively. Considering fading channel, given R and SNR, I use normal approximation bound (5.1.3) to calculate the probability of an erroneous communication.



(a) $\gamma_0 = -4$ dB

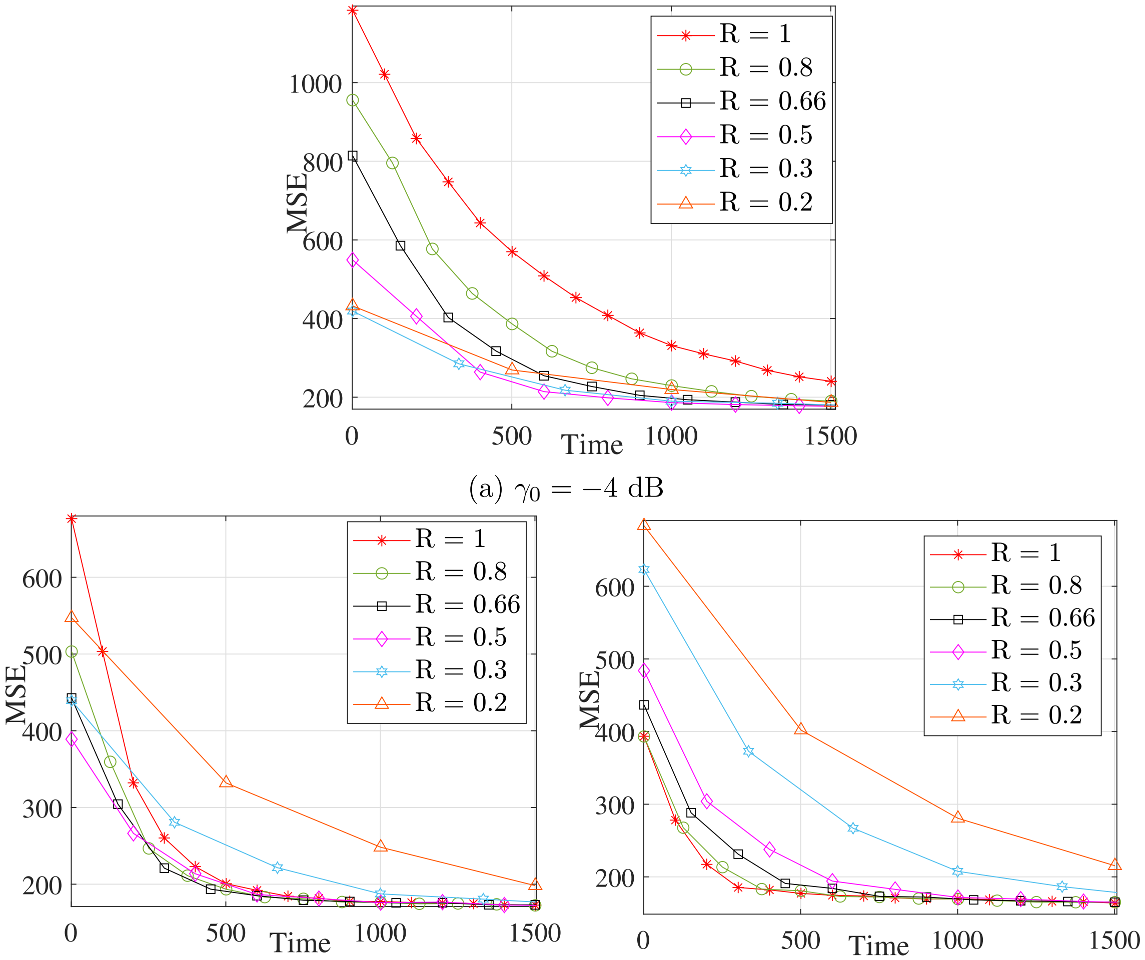
(b) $\gamma_0 = -1$ dB

(c) $\gamma_0 = 3$ dB

(d) Non-i.i.d. Dataset

Figure 5.4: Impact of Rate and SNR on convergence of FL when the BS stores local parameters of devices in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\eta = 0.05$ and 1 iteration of GD at device is applied.

Fig.5.4c and Fig. 5.5c illustrate that in the high SNR regime, increasing the code rate results in a lower convergence time. Because the users' power is sufficient to overcome the fading and noise, devices with higher rates will communicate more often than those with lower rates over a given learning period, leading the MSE to converge faster.



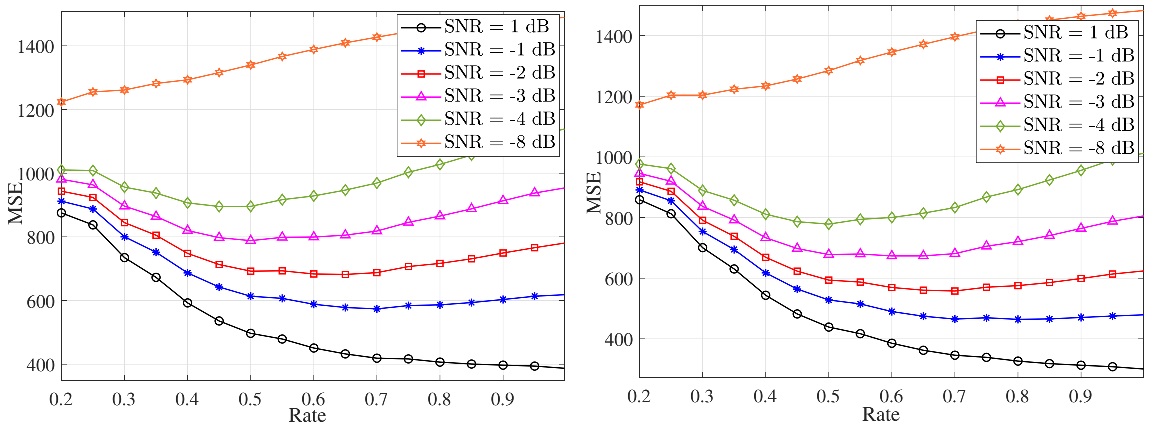
(a) $\gamma_0 = -4$ dB

(b) $\gamma_0 = -1$ dB

(c) $\gamma_0 = 3$ dB

Figure 5.5: Impact of Rate and SNR on the convergence of FL when the BS store global parameters in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\eta = 0.05$, $m = 2$ and 1 iteration of GD at device is applied.

In the low SNR regime, lower code rates are preferable for a shorter convergence time since one must compensate for the channel noise and fading by adding more



(a) BS stores global updates, $m = 2$

(b) BS stores local update of users

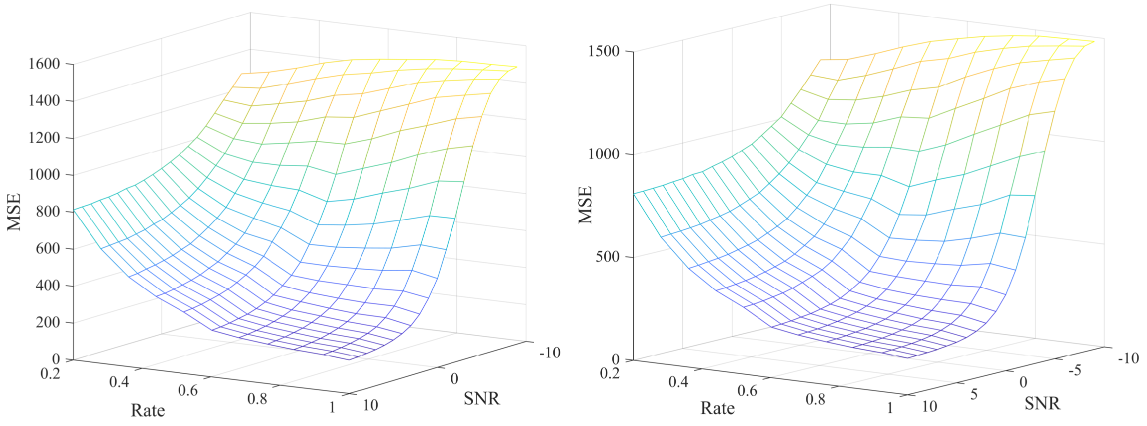
Figure 5.6: Rate vs. MSE of FL in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\eta = 0.005$, time = 1500 T_s and 1 iteration of GD at device is applied.

parity bits to the message. However, the minimum code rate does not necessarily result in the minimum MSE. Fig. 5.4b and Fig. 5.5b demonstrate that for low SNRs, a coding rate of $R = 0.66$ yields the best performance. It is important to note that, in this scenario, the performance of FL when the code rate is $R = 0.2$ is significantly inferior to that of $R = 1$. When the SNR is further reduced, as illustrated in Fig. 5.4a and Fig. 5.5a, the optimal coding rate R is decreased[4].

Fig. 5.6 shows the MSE performance for various coding rates and multiple SNRs, assuming a fixed convergence time. It is evident that, for SNRs greater than 1 dB, the highest coding rate results in the minimum MSE. Nevertheless, as the SNR decreases, the optimal coding rate changes, necessitating careful selection of the coding rate based on the SNR regime.

Fig. 5.7 shows the loss performance of erroneous short packet communication as a function of R and SNR. I set the time to 1500 T_s, where T_s is the symbol duration.

[4] I also investigated the impact of adding new data to the dataset of each user at each communication round. Given that IoT measurements do not drastically change over time, the gradient value at each round remains relatively stable. Consequently, this does not noticeably affect the results. Therefore, the results have not been added to this chapter.



(a) BS stores global updates, $m = 2$

(b) BS stores local update of users

Figure 5.7: Rate vs. SNR vs. MSE of FL in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\eta = 0.005$, time $= 1500\ T_s$ and 1 iteration of GD at device is applied.

As can be seen, when the SNR is low, for a descent loss performance, I need lower R to mitigate the impact of high noise and fading. However, in high SNR scenarios, it is better to increase the code rate R. This is due to the fact that when R is low, the packet length n increases, and subsequently the number of communication rounds between devices and the BS decreases, leaving users with little time to achieve ideal performance. Fig. 5.7 clearly demonstrates that when R grows, the number of communications for given times will increase and as a result, the MSE will decrease.

When the BS does not have enough storage capacity, instead of storing the local parameters of devices, it will store the past m global updates (5.1.2). Fig. 5.8 shows the impact of the available memory when the BS uses memory to cache the previous m global updates. Here, I considered all the past stored global updates having similar weight ($\alpha = 1/m$). As can be seen, the performance converges with just one unit of memory ($m = 1$) and no fluctuations are visible. Additionally, it is evident that the implementation of multiple memory units ($m > 2$) and utilizing equal weight

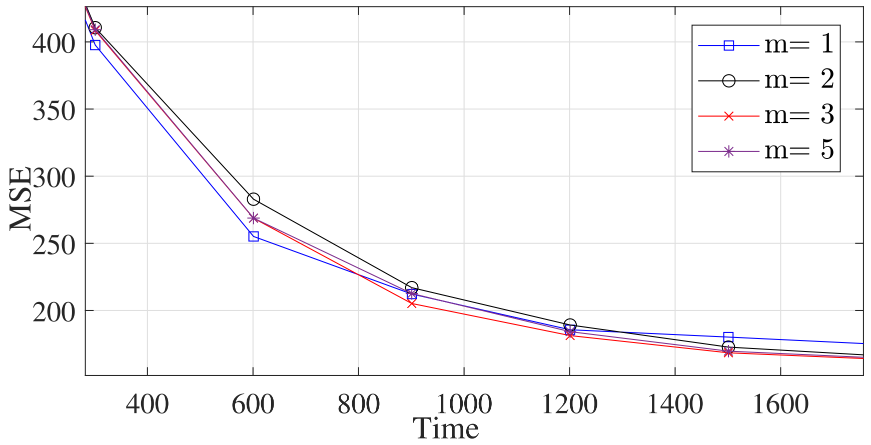


Figure 5.8: Impact of memory capacity, m, on convergence, when the BS store the global parameter in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\gamma_0 = 7$ dB, $\eta = 0.05$, $R = 0.5$ and similar α for all global parameters (equal weighted).

averaging does not significantly enhance the performance of the system. This finding shows that beyond a certain threshold, the utilization of additional memory units may not yield substantial improvement.

Fig. 5.9 examines the effect of averaging model on the FL performance when I apply a FL scheme where the BS caches the past $m = 6$ global updates. As is shown, I compare the performance of equal weighted averaging with the exponentially weighted moving average (EWMA). The EWMA can be written as

$$\boldsymbol{\omega}^{(t)} = \alpha \sum_{i=1}^{m} (1-\alpha)^{i-1} \omega^{(t-i)}. \tag{5.3.1}$$

A larger α accelerates the process of discounting earlier observations. As can be seen, the EWMA with $\alpha \geq 0.5$ has better performance compare to equal weighted averaging due to the fact that it gives higher weight to the recent global parameter. Our findings show that for $\alpha > 0.95$, the performance increase is minimal.

Despite several studies [159–161] addressing packet erasure in FL over wireless communication through packet retransmission strategies, this approach has significant drawbacks. Primarily, the usage of packet retransmission strategies can result in

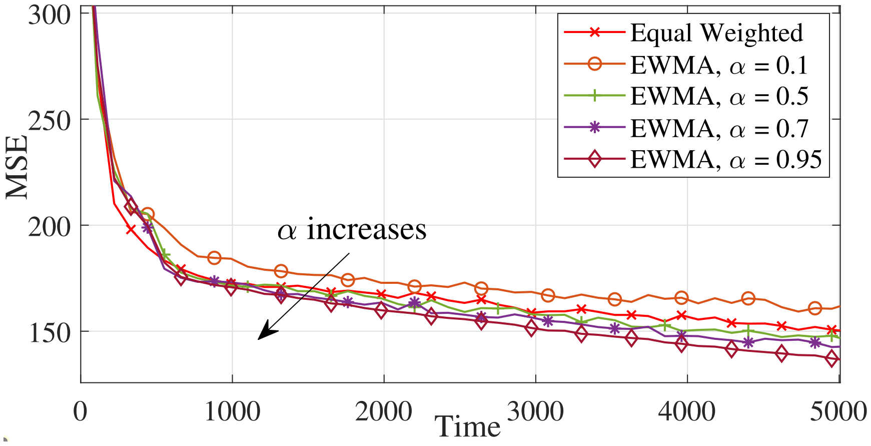
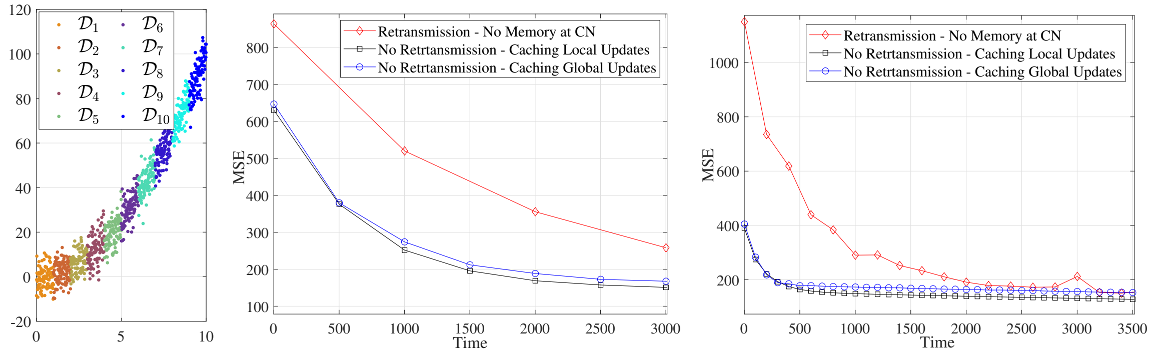


Figure 5.9: Impact of various averaging techniques on convergence when the BS store the global parameter in erroneous communication, for $N_u = 10$, $|D_u| = 100$, $\gamma_0 = 3$ dB, $\eta = 0.05$, $R = 0.9$ and $m = 6$.

a prolongation of the FL convergence time. This delay arises as the BS is required to wait for the receipt of erroneous packets prior to computing the average parameters from all users. Furthermore, there is an inherent lack of assurance that the retransmitted packet will be decoded accurately. This uncertainty factor becomes markedly significant in a scenario of low SNR, where incorrect retransmissions can incite variability in the FL convergence. As a consequence, the retransmission methods, as explored in prior works, may not yield consistently reliable performance under conditions of low SNR. Figure 5.10 showcases that our proposed model outperforms the retransmission strategies cited in previous studies, regardless of high or low coding rate conditions. Moreover, as observed in Figure 5.10c where the SNR is set at 2 dB, there is an inherent inconsistency in the MSE performance, despite the single retransmission of the erroneous packet under the retransmission strategy. This observation underscores the potential instability associated with this approach.



(a) Dataset (b) $R = 0.2$ (c) $R = 1$

Figure 5.10: Convergence of FL when the CN has memory to stores local/global parameters of devices versus the case of retransmission in erroneous communication, for $U = 10$, $\eta = 0.05$, $|D_u| = 100$, $\gamma_0 = 2$ dB, $m = 2$ and 1 iteration of GD at device is applied.

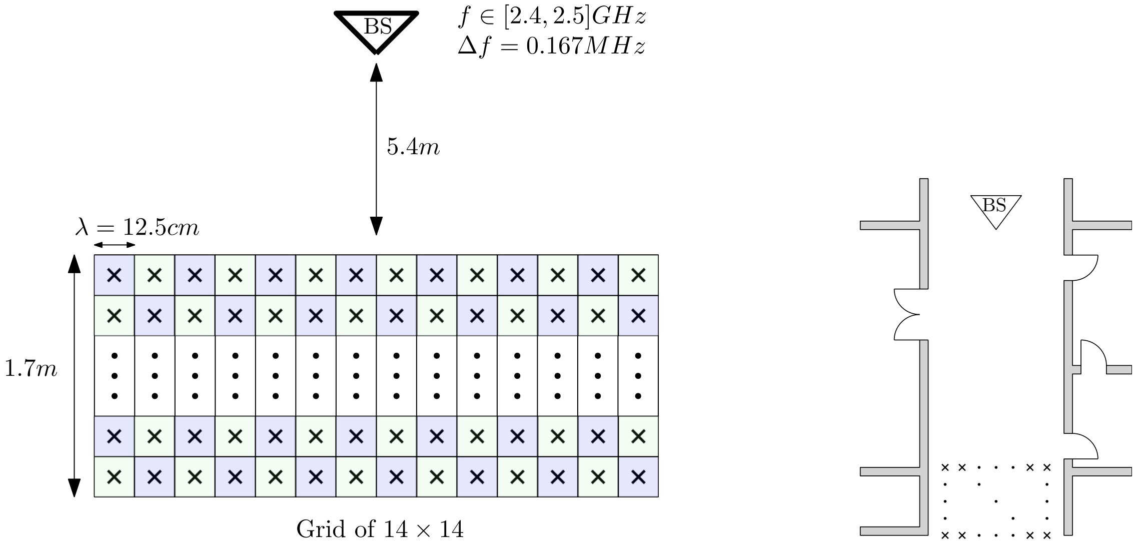
5.4 Massive IoT: Practical Scenarios

In this section I consider two real world scenarios to show the tradeoff between the rate, SNR and probability of error and the impact on the convergence time and accuracy. I discuss both short packet and long packet communication.

5.4.1 Short Packet Communication

I examine the effectiveness of the proposed FL methods in a practical mIoT scenario. Our study makes use of the data set in [45], which is the measurements of the channel transfer function in the frequency domain using a Rhode & Schwartz ZVB14 vector network analyser (VNA). This measurement system employs omni-directional antennas at both the BS and IoT devices, with both the transmitter and receiver positioned at a height of $1.5m$. The VNA collects 601 frequency samples at 0.167 MHz intervals in a 100 MHz spectrum beginning from 2.4 GHz. As shown in Fig. 5.11, 196 IoT devices are positioned on a 14×14 grid separated by one wavelength ($\lambda = 12.5cm$).

It is important to note that the real-time measurements were performed with the transmitter and receiver in a completely static environment.



(a) Grid of users (14×14)

(b) Floor Plan

Figure 5.11: Massive IoT setup.

I assume the data of each user can be estimated using

$$g(x, y, f) = \sum_{i=1}^{L_1} \kappa_i f^{-i+1} + \sum_{i=1}^{L_2} \nu_i x^{-i+1} + \sum_{i=1}^{L_3} \xi_i y^{-i+1}, \tag{5.4.1}$$

where f is the frequency, and x and y are the coordinates of the user. In the case of centralized learning, IoT devices send the whole data to the BS for estimating κ, ν and ξ. This means that the BS needs $601 \times 196 \times (L_1 + L_2 + L_3)$ parameters for fitting. In the case of FL, each user applies the GD on its own data and transmit the learning parameters to the BS. Therefore, the maximum number of parameters transmitted by devices is $196 \times (L_1 + L_2 + L_3)$. I consider $L_1 = 3$, $L_2 = 2$, $L_3 = 2$. Therefore, at each iteration of FL, users have 7 coefficients to transmit. Owing to the fact that IoT devices are power limited, I assume that they are using IEEE-754 half precision standard for quantization, therefore, each parameter will be converted to a

16 bit digit that has up to 8 floating point number precision. As a result, each user have a packet of 112 bits to transmit to the BS.

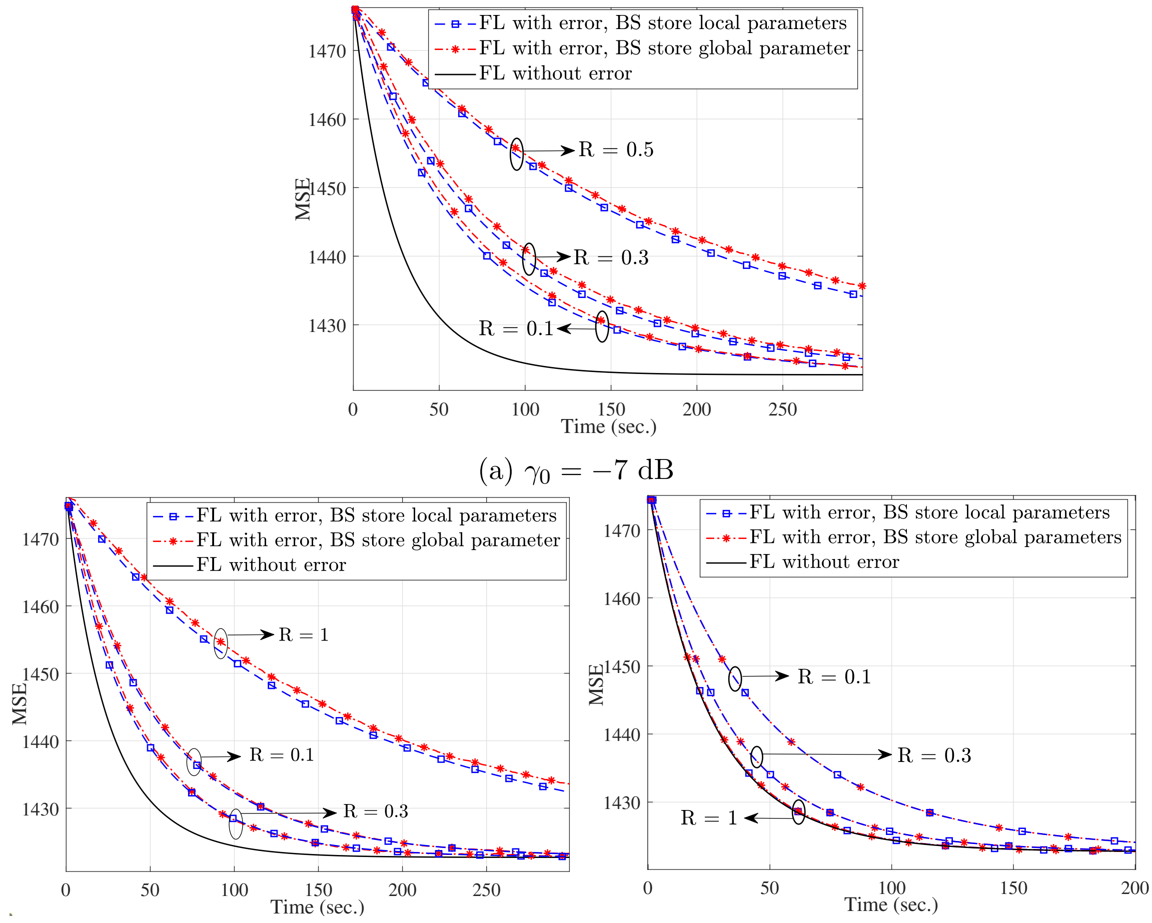
The majority of IoT devices use a low-power micro-controller unit in which algorithms with minimal computation complexity can be executed [162, 163]. I assume that users have access to the Particle Electron board, a popular board that has seen extensive use in IoT applications. It runs at a speed of 120 MHz thanks to its ARM Cortex M3 processor and has a max gain of 5 dBi at 2.45 GHz. Particle Electron is equipped with a flexible ultra wideband polymer antenna that covers all common LTE frequencies [164]. Therefore, the transmission of users is assumed to be orthogonal, and they undergo Rayleigh fading. I am interested in determining the overall FL system delay given the SNR and code rate. I further disregard the BS's communication and processing time since it has a high SNR, a large bandwidth, and high computing capacity. Therefore, it is reasonable to conclude that the delay results from the processing and transmission times of individual IoT nodes. Each user's computation time during local training can be determined using

$$t^u_{comp} = \frac{\rho \mu_u |D_u|}{CC_u}, \tag{5.4.2}$$

where ρ denotes the number of CPU cycles required to process a single sample of data which is set to 10^4 cycle/sample [165], CC_u is the computational capacity of user u which is 1.2×10^8 for Particle Electron board, μ_u is the number of iterations of GD for local training of user u in a communication round, and $|D_u|$ is the number of samples of user u. Furthermore, the communication time can be computed using

$$t^u_{comm} = \frac{L_u}{R_u}, \tag{5.4.3}$$

where L_u is the size of model parameters of user u and R_u is the transmission rate



(a) $\gamma_0 = -7$ dB

(b) $\gamma_0 = -3$ dB

(c) $\gamma_0 = 15$ dB

Figure 5.12: Impact of Rate and SNR on convergence of FL for BS with memory in erroneous communication, for massive IoT short packet communication with $\eta = 0.05$, $\alpha = 0.95$, $m = 5$ and 1 iteration of GD at device is applied.

from user u to the BS.

I use the information of the blue users in Fig. 5.11a for training purposes. To ensure that the dataset is non-i.i.d. and imbalanced across all the devices, I assume that each user in each row measures the channel transfer function at a fraction of frequencies and that the size of each user's dataset is a random number between 80 and 120. For the test, I use 30% of the data of green users in each row.

The results are presented in Fig. 5.12 which shows the convergence of MSE in a massive IoT scenario for various coding rates across different SNR regimes. I consider $R = 1$ in the case of FL without error. Fig. 5.12a shows that in the low SNRs, the selection of high coding rates results in a longer convergence time. Conversely, Fig. 5.12c demonstrates that in high SNRs, higher coding rates are recommended for faster convergence. However, as is shown in Fig. 5.12b, it is apparent that in the mid-SNR regime, the selection of the code rate requires careful consideration, as neither a high code rate nor a low code rate is optimal.

5.4.2 Long Packet Communication

To investigate the performance of the proposed FL schemes in long packet communication, I consider image classification and apply the FL to train a neural network using a highly non-i.i.d. dataset. In order to compute the PER for long-packet communication across a fading channel, given R and the transmitted SNR γ_0, I use (5.1.2) and check to see if the instantaneous received SNR is above the SNR threshold required for error-free communication. I consider the MNIST digits dataset, which consists of handwritten images of each number 0 to 7. I considered eight IoT devices ($U = 8$), each of them with 1000 images of one of the numbers between 0 and 7. I employed a MATLAB parallel pool with eight workers and allocated 70% of the dataset to training, 15% to test and 15% to validation. In my simulation, I deployed a CNN model consisting of nine layers designed to accommodate the MNIST data set. This model includes an image input layer, a series of alternating convolutional layers, ReLU activation layers for feature extraction, and max pooling layers for spatial dimension downsampling. The model's convolutional layer depth scales from 32 to 64, with each layer having a kernel size 5. The model concludes with a fully connected layer, which

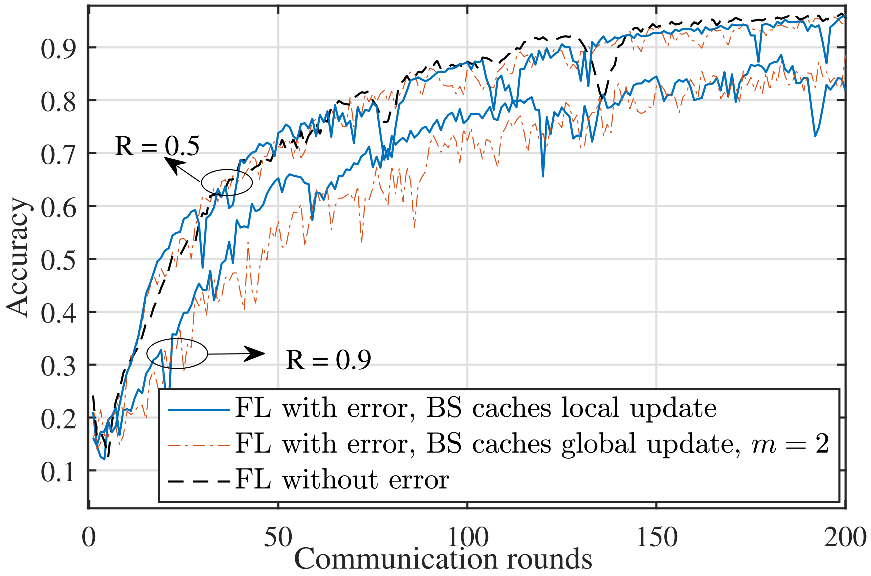


Figure 5.13: Accuracy vs. communication round for the FL algorithm with MNIST digits dataset, when $U = 8$, $\gamma_0 = 0$ dB, and long packet communications.

correlates to our data set's classes, and a softmax layer to derive output probabilities [37]. Fig. 5.13 shows the accuracy of FL schemes for error-free and erroneous communication for two distinct coding rates. As can be seen for $\gamma_0 = 0$ dB, applying a low coding rate results in better overall performance. One can see that erroneous communication with $R = 0.5$ has a similar performance to error-free communication, and the performance will reach 90% in less than 100 communication rounds. However, when I increase the coding rate to $R = 0.9$, after 200 communication rounds the accuracy will be around 84%.

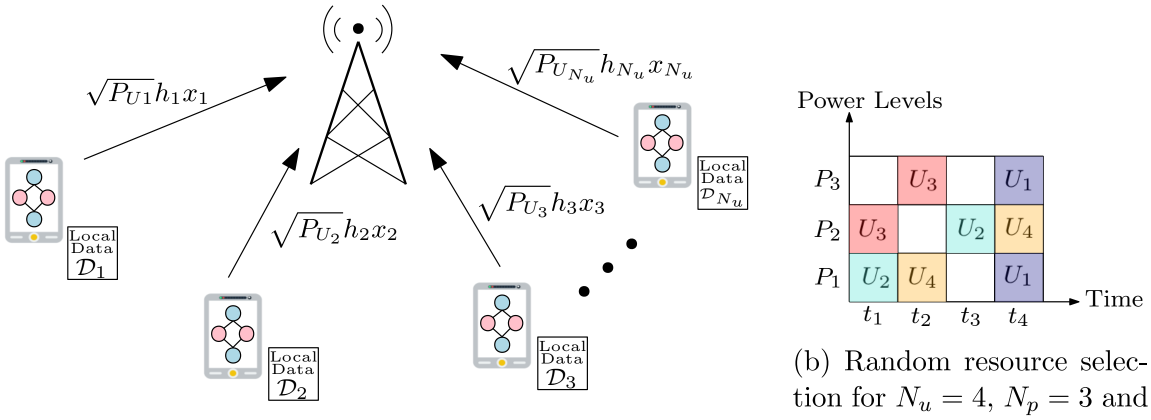
5.5 Conclusion

In this chapter, I investigated the performance of the federated learning algorithm in the presence of communication errors and studied the impact of coding rate and block length on accuracy and convergence. I modeled the communication channels as packet erasure channels, with block length, coding rate, and SNR determining the erasure probability. I proposed two schemes to improve the performance of FL under erroneous communications. I demonstrated the effect of coding rate on the convergence of FL for both short packet and long packet communications considering erroneous transmission. I illustrated that a single memory unit has a significant effect on the performance of FL. While the communication errors are deleterious to the reliability of the packets, the effect can be easily compensated by reusing past local or global parameters, in case of communication errors. This is of significant importance for mIoT systems, as one can relax the reliability requirement, and still achieve the desired level of accuracy withing the required time.

Chapter 6

Unified Framework of Clustering, Channel Coding, and Federated Learning in Massive IoT

Drawing upon the insights gained in prior chapters, this chapter converges the preceding methods and principles, situating them in a practical mIoT scenario. The chapter emphasizes that, under the FL framework, it is possible to achieve optimal learning outcomes even with moderate communication reliability, underscoring the flexibility of my approach. Moreover, I introduce a mathematical model for estimating the error probability of a mIoT system, enabling a detailed assessment of the system's learning performance. Through the integration of the learning model, communication reliability, and error probability analysis, this chapter provides a thorough and inventive viewpoint on the implementation and optimization of mIoT systems within the scope of an FL framework.


$\sqrt{P_{U1}}h_1x_1$
$\sqrt{P_{U_2}}h_2x_2$
$\sqrt{P_{U_3}}h_3x_3$
$\sqrt{P_{U_{N_u}}}h_{N_u}x_{N_u}$
Local Data $\mathcal{D}_1$
Local Data $\mathcal{D}_2$
Local Data $\mathcal{D}_3$
Local Data $\mathcal{D}_{N_u}$
Power Levels
Time

	t_1	t_2	t_3	t_4
P_3		U_3		U_1
P_2	U_3		U_2	U_4
P_1	U_2	U_4		U_1

(a) System Model.

(b) Random resource selection for $N_u = 4$, $N_p = 3$ and $N_r = 2$.

Figure 6.1: System Model and Resource Selection

6.1 System Model

As depicted in Fig. 6.1a, I consider an FL system consists of a BS and N_u active users, each equipped with their unique local dataset, expressed as $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{N_u}$. As part of the FL process, users transmit the learning parameters to the BS in each communication round. I elaborate on the methodology adopted for users' transmission in the following.

I consider a UL NOMA scenario in which N_u active users transmit their packet in a time frame that consists of N_t time-slots to the BS. As is shown in Fig. 6.1b, I assume that each user, in a random selection process, chooses one among the N_p power levels and N_r time-slots for transmission, which effectively means each user initiates the transmission of N_r repetitions of its packet in the selected time slots. Additionally, I assume that users employ QPSK modulation alongside a channel coding of rate 1/2 for their transmissions. In each time-slot j, the received signal is

$$y_j = \sum_{i \in \mathcal{S}_j} h_i \sqrt{P_i} x_i + w, \tag{6.1.1}$$

Here, $\mathcal{S}_j$ represents the set of active users in the jth time slot. The transmit power of user i is symbolized by P_i. The term w denotes AWGN, which follows a complex normal distribution $\mathcal{CN}(0, 1)$, and h_i represents the fading channel gain existing between user i and the BS, modeled using the Rayleigh distribution. I consider block fading, that is, the channel between each user and the BS remains unchanged for each transmission frame.

I consider a SIC receiver at the BS. This SIC receiver is uniquely adapted to leverage the GMM clustering algorithm, the specifics of which are comprehensively detailed in Algorithm 2 of Chapter 3. My utilization of the GMM clustering algorithm within the SIC receiver enables efficient and effective joint channel estimation and signal detection. Once the users' transmissions are successfully decoded, their data is integrated into the learning process facilitated by the FL model. It is worth noting that the detection process here aligns with the approach delineated in Chapter 3. As such, I continue the decoding process until the end of each time frame, leveraging the data from successfully decoded users to compute updated learning parameters using the FL model.

6.2 Numerical Results

In Section 5.4.1, I put the proposed FL methods to the test within a mIoT system context. Drawing on the dataset from Alhajri's study [45], I assessed the channel transfer function using a Rhode & Schwartz ZVB14 VNA. In a static environment, the system captured 601 frequency samples from 196 IoT devices arranged in a 14×14 grid (see Fig. 5.11). Each device's data was modeled using an equation that incorporated frequency and device coordinates (Eq.5.4.1). In this section, I replicate the same setup, using data from half of the users (represented as blue users in Fig. 5.11a) to train the model. For the testing phase, I employ 30% of the data from the remaining users (illustrated as green users in Fig. 5.11a) who were not considered for the training phase.

I consider a scenario where each user selects a power level randomly from a set of possible values represented as $\mathbb{P}_p = P_1, P_2, \cdots, P_5$. Moreover, the user chooses N_r time-slots from the total N_t time-slots available within the designated time frame. Users use a code with a rate of $R = 1/2$ to encode their messages and initiate transmission using QPSK modulation. In this scenario, I account for small-scale fading, which is modeled by a Rayleigh distribution. Five power levels are taken into consideration, calculated as $P_i = P_1 + (i-1) \times 5\text{dB}$ for $i = 1, \cdots, 5$. At the BS, I implement a SIC receiver that uses the GMM clustering algorithm. Once users are successfully decoded, they are incorporated into the learning process.

For the learning procedure, I employ FL as detailed in Chapter 5. I evaluate two approaches that involve the BS using a memory system: in one strategy, I have the BS retain previous local updates from each user, and in the other, I keep records of global updates at the BS.

6.2.1 Implementing FL with the Clustering-based Detection Algorithm Introduced in Chapter 3

In this part, I presume that the receiver deploys the algorithm 2 that I proposed in Chapter 3. I further consider that users transmit five pilot symbols for channel estimation and use a rate 1/2 systematic convolutional encoder. Moreover, I assume that the system under examination possesses five memory units, represented as $m = 5$,

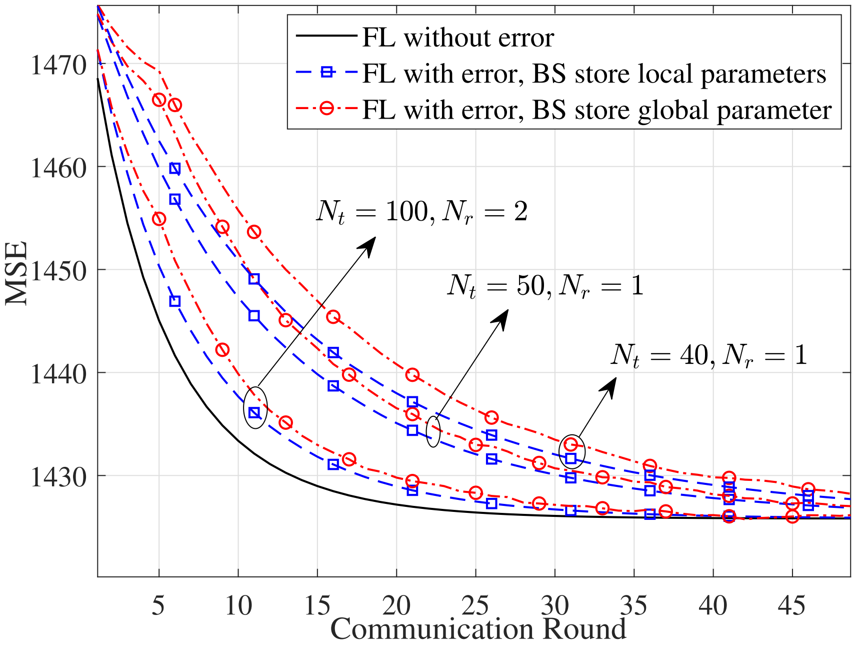


Figure 6.2: Convergence of FL for BS with memory, for mIoT short packet communication with rate 1/2 systematic convolutional encoder, $\gamma_0 = 1dB$, $\eta = 0.05$, $\alpha = 0.95$, $m = 5$ and 1 iteration of GD at device is applied.

which are utilized for storing the global parameter. The computation of the updated global parameter is carried out through the use of the EWMA technique with a weighting factor of α as shown in Equation 5.3.1.

Figure 6.2 showcases the results, displaying the MSE convergence in a mIoT context across various transmission arrangements. When considering error-free FL, I work with a rate of $R = 1$. The findings suggest that extending transmission opportunities (that is, increasing the number of time-slots, N_t) shortens the convergence period. This benefit arises from the lowered probability of time-slot overlaps among the 98 transmitting users, which in turn means users have an increased likelihood of successful transmission. This reduces the error probability, leading to more active participation in the learning process and consequently accelerating algorithm convergence.

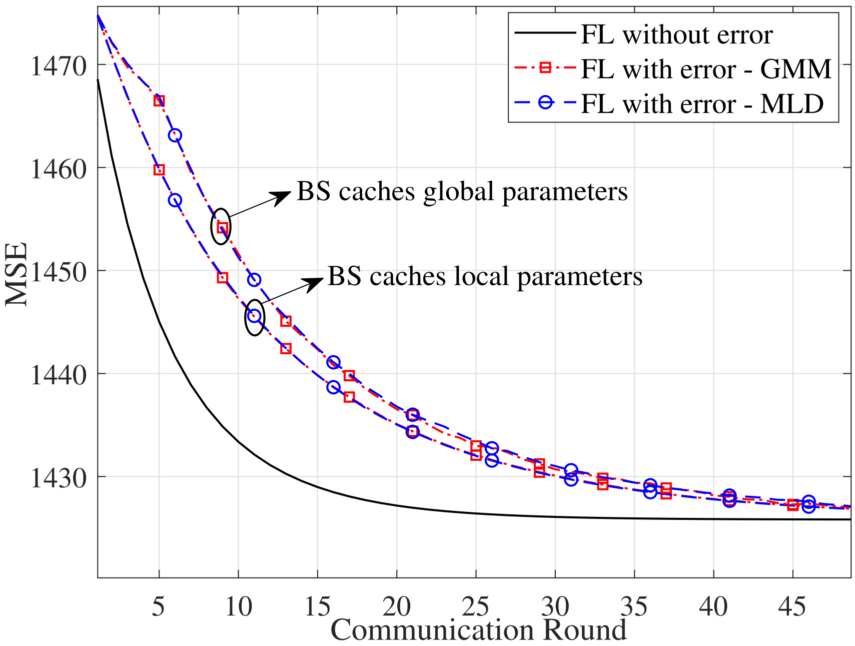


Figure 6.3: Comparison of convergence of FL when BS applies GMM clustering versus MLD, for mIoT short packet communication with rate 1/2 systematic convolutional encoder, $\gamma_0 = 1dB$, $\eta = 0.05$, $\alpha = 0.95$, $m = 5$ and 1 iteration of GD at the device is applied.

To provide further insight, I simulated a scenario where each user selects two time-slots for transmission. This strategy affords the BS a higher decoding success rate. Once the BS has successfully decoded a user, it can eliminate any duplicate signals from other time-slots. As shown in Figure 6.2, the FL algorithm converges to the optimal value swiftly in this situation.

It is essential to emphasize that when $N_t = 40$, there is an average of $\zeta = \frac{98}{40} = 2.45$ devices transmitting in each time-slot, which can induce a high probability of error (as discussed in Chapter 3). However, since the goal here is to learn a model, I can relax the communication reliability to a moderate degree, while still maintaining robust performance via FL.

Figure 6.3 provides a comparative view of the FL convergence performance under two conditions: one where I employ GMM clustering at the BS for channel estimation and signal detection, and the other where I implement MLD at the BS. For this comparison, I assume that the communication frame comprises $N_t = 50$ time-slots and each user sends its packet once ($N_r = 1$). Upon examining the results, it becomes clear that the performance of the mIoT system when utilizing GMM at the BS closely parallels the performance when employing MLD at the BS. This outcome is consistent with the conclusions drawn in Chapter 3, where I demonstrated that the SER performance of GMM is closely akin to that of MLD.

6.2.2 Implementing FL with the Clustering-based Detection Algorithm Introduced in Chapter 4

Here, I assume that the receiver deploys algorithm 4, as recommended in Chapter 4. I assume that users do not transmit any pilot symbols and engage a rate 1/2 RI encoder, as demonstrated in Fig. 4.1.

The performance of the mIoT system, under the condition of $N_t = 50$ time-slots and a single transmission per user ($N_r = 1$), is showcased in Fig.6.4. It is important to acknowledge that despite the presence of Rayleigh fading and the absence of pilot symbols transmitted by users, the utilization of GMM clustering algorithm in combination with the RI code (as elaborated in Chapter 4) can yield comparable results to those obtained when pilots are transmitted.

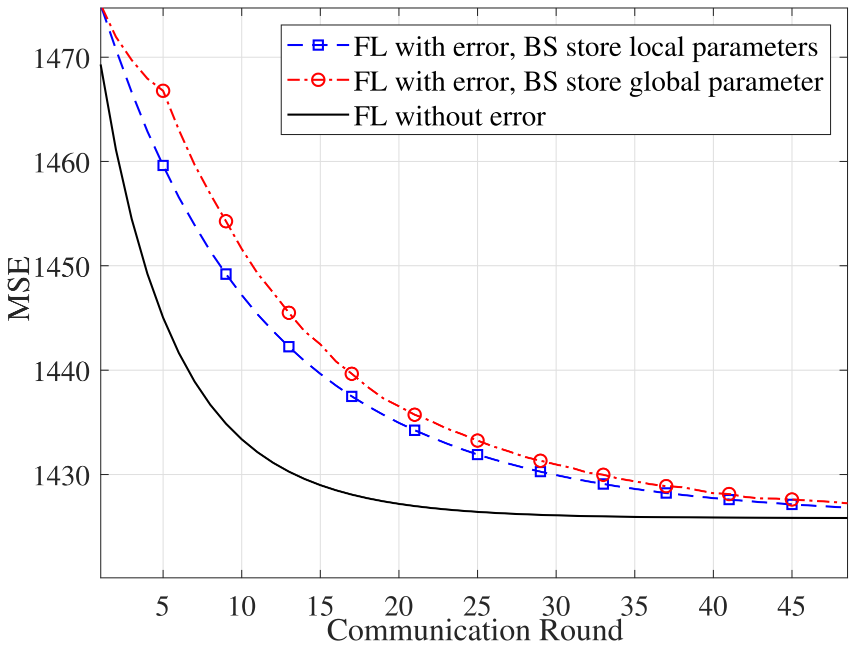


Figure 6.4: Convergence of FL for BS with memory, for mIoT short packet communication with rate 1/2 RI encoder (Fig. 4.1), $\gamma_0 = 1dB$, $\eta = 0.05$, $\alpha = 0.95$, $m = 5$, $N_t = 50$, $N_r = 1$ and 1 iteration of GD at device is applied.

6.3 Mathematical Analysis of Packet Error Rate in the Proposed System Model

In this section, I develop a mathematical model specifically designed to calculate the PDF of the packet error rate in a mIoT scenario described in Section 6.1. I start by calculating the PDF of the signal-to-interference-plus-noise ratio (SINR) for each time slot i, considering the number of active users in that slot. This PDF is represented as $f_{SINR}(\gamma \mid N_u^i)$. I then use order statistics to find the PDF of the maximum SINR for a given number of users, which is denoted as $f_{SINR}(\gamma_{\max} \mid N_u^i)$. From here, I apply the Poisson distribution to find $f(N_u^i; \zeta)$, which represents the probability of observing a certain number of active users in each slot. As a result, I determine $f_{SINR}(\gamma_{\max})$, the PDF of the maximum SINR, regardless of the number of users in each slot. Following this, I compute the PDF of the packet error rate, denoted as $f_\epsilon(\epsilon)$. With these calculations done, I determine the number of users successfully decoded in time slot i and apply SIC. I repeat this process for each time slot until the end of the time frame. The following subchapters will provide a thorough breakdown and explanation of each of these steps.

6.3.1 PDF of SINR

Over a Rayleigh channel, I can write the PDF of the received SNR, γ, as

$$f_\gamma(\gamma) = \frac{1}{\bar{\gamma}} \exp\Big[-\frac{\gamma}{\bar{\gamma}}\Big], \quad \gamma > 0 \tag{6.3.1}$$

where $\bar{\gamma} = 2\sigma^2 E_s/N_0$ is the average received SNR [166]. Without loss of generality, I consider $N_0 = 1$ and $|h|^2 \sim \exp(1)$; hence, PDF of received SNR can be written as [167]

$$f_\gamma(\gamma) = \exp(-\gamma), \quad \gamma > 0 \tag{6.3.2}$$

In NOMA with SIC, I account for possible interference from other users. For this, I calculate the SINR, which serves as a crucial metric. I express this calculation mathematically as follows

$$SINR_i = \frac{P_i|h_i|^2}{\sum_{j=i+1}^{N_u} P_j|h_j|^2 + \sigma_w^2} \tag{6.3.3}$$

Lemma 6.3.1. *Let me define the PDF of the SINR given the number of users as $f_{SINR}(\gamma|N_u)$. Assuming $\sigma_w^2 = 1$, and $\lambda_i = \frac{1}{|P_i|}$, the $f_{SINR}(\gamma \mid N_u)$ for $N_u \in \{1, 2, 3, 4\}$ is*

$$f_{SINR}(\gamma \mid N_u = 1) = \frac{1}{N_p} \sum_{i=1}^{N_p} \lambda_i \exp\Big(-\lambda_i \gamma\Big), \tag{6.3.4}$$

$$f_{SINR}(\gamma \mid N_u = 2) = \frac{1}{N_p^2} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \lambda_i \lambda_j \frac{\exp\left(\lambda_i\right)}{\left(\lambda_i + \lambda_j \gamma\right)^2}, \tag{6.3.5}$$

$$\begin{aligned} f_{SINR}(\gamma \mid N_u = 3) =& \frac{1}{N_p^3} \sum_{\substack{i_1=1 \\ i_1 \neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_j \left(\frac{\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_2} - \lambda_{i_1}\right)\left(\lambda_{i_1} + \lambda_j \gamma\right)^2} \right. \\ & \left. + \frac{\exp\left(\lambda_{i_2}\right)}{\left(\lambda_{i_1} - \lambda_{i_2}\right)\left(\lambda_{i_2} + \lambda_j \gamma\right)^2} \right) \\ & + \frac{1}{N_p^2} \sum_{i_1=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1}^2 \lambda_j \frac{2 \exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_1} + \lambda_j \gamma\right)^3}, \end{aligned} \tag{6.3.6}$$

$$\begin{aligned} f_{SINR}(\gamma \mid N_u = 4) =& \frac{1}{N_p^4} \sum_{\substack{i_1=1 \\ i_1 \neq i_2 \neq i_3}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{i_3=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_{i_3} \lambda_j \left(\frac{\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_2} - \lambda_{i_1}\right)\left(\lambda_{i_3} - \lambda_{i_1}\right)\left(\lambda_{i_1} + \lambda_j \gamma\right)^2} \right. \\ & + \frac{\exp\left(\lambda_{i_2}\right)}{\left(\lambda_{i_1} - \lambda_{i_2}\right)\left(\lambda_{i_3} - \lambda_{i_2}\right)\left(\lambda_{i_2} + \lambda_j \gamma\right)^2} \\ & \left. + \frac{\exp\left(\lambda_{i_3}\right)}{\left(\lambda_{i_1} - \lambda_{i_3}\right)\left(\lambda_{i_2} - \lambda_{i_3}\right)\left(\lambda_{i_3} + \lambda_j \gamma\right)^2} \right) \\ & + \frac{3}{N_p^3} \sum_{\substack{i_1=1 \\ i_1 \neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_j \left(\frac{-\exp\left(\lambda_{i_1}\right)\left(\lambda_{i_1} + \lambda_j \gamma - 2\right)}{\left(\lambda_{i_2} - \lambda_{i_1}\right)\left(\lambda_{i_1} + \lambda_j \gamma\right)^3} \right. \\ & + \frac{\exp\left(\lambda_{i_2}\right)}{\left(\lambda_{i_2} - \lambda_{i_1}\right)^2\left(\lambda_{i_2} + \lambda_j \gamma\right)^2} \\ & \left. - \frac{\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_2} - \lambda_{i_1}\right)^2\left(\lambda_{i_1} + \lambda_j \gamma\right)^2} \right) \\ & + \frac{1}{2! N_p^2} \sum_{i_1=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1}^3 \lambda_j \frac{6 \exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_1} + \lambda_j \gamma\right)^4}, \end{aligned} \tag{6.3.7}$$

Proof. The proof is provided in Appendix C.1. □

I should highlight that although the PDF of the SINR can be readily calculated for a larger number of users, it is unusual in NOMA settings to pair more than three users together. As a result, to align with prevalent NOMA implementations and for practical considerations, in this study, I have confined the calculation of the PDF of the SINR to a maximum of four users.

6.3.2 Applying Order Statistics

Order statistics involves the study of distributions and statistical characteristics of newly created random variables that result from arranging the realizations of a set of random variables. I Consider γ_j's, where $j \in 1, 2, \cdots, N_u$, to denote N_u i.i.d non-negative random variables with a PDF $f_{SINR}(\gamma)$ and a CDF $F_{SINR}(\gamma)$. Here, I define $\gamma_{i:N_u}$ as the random variable corresponding to the i^{th} largest value within the original set of N_u random variables, such that the sequence $\gamma_{1:N_u} \geq \gamma_{2:N_u} \geq \cdots \geq \gamma_{N_u:N_u}$ holds. The term $\gamma_{i:N_u}$ is what I recognize as the i^{th} order statistics. As demonstrated in [168], I can express the PDF of $\gamma_{i:N_u}$, for $i \in 1, 2, \cdots, N_u$, in the following manner:

$$f_{SINR}(\gamma_{i:N_u}) = \frac{N_u!}{(N_u - i)!(i-1)!}\Big(F_{SINR}(\gamma)\Big)^{N_u - i}\Big(1 - F_{SINR}(\gamma)\Big)^{i-1} f_{SINR}(\gamma), \tag{6.3.8}$$

By integrating Lemma 6.3.1 and equation 6.3.8, I derive the PDF of $\gamma_{\max}$, given the number of users N_u, in the following way:

$$f_{SINR}(\gamma_{\max} \mid N_u) = N_u\Big(F_{SINR}(\gamma \mid N_u)\Big)^{N_u - 1} f_{SINR}(\gamma \mid N_u), \tag{6.3.9}$$

6.3.3 Deriving PDF of $f_{SINR}(\gamma_{\mathbf{max}})$

In order to model the number of users present at each individual time slot, I utilize the Poisson distribution. Given that there are a total of N_u users and N_t transmission opportunities, I compute the average number of users selecting a specific time slot as

$\zeta = \frac{N_u}{N_t}$. This results in the following PDF:

$$f(N_u; \zeta) = \frac{\zeta^{N_u}}{N_u!} \exp(-\zeta), \tag{6.3.10}$$

Upon obtaining the PDF of the number of users, I can express $f_{SINR}(\gamma_{\max})$ in the following manner:

$$f_{SINR}(\gamma_{\max}) = \sum_{n_u=1}^{N_u} f_{SINR}(\gamma_{\max} \mid n_u) f(n_u; \zeta), \tag{6.3.11}$$

6.3.4 PDF of Packet Error Rate

In this subsection, I derive the PDF of the error probability, leveraging the known PDF of $\gamma_{\max}$. In FBL regime, I can compute the packet error rate at the receiver as follows:

$$\epsilon(\gamma_{\max}) \approx Q\left(\frac{nC(\gamma_{\max}) - k + 0.5\log_2(n)}{\sqrt{nV(\gamma_{\max})}}\right) \tag{6.3.12}$$

Given the monotonic nature of ϵ (similar to the Q-function), I can apply the following theorem to determine the PDF of ϵ based on the known $f_{SINR}(\gamma_{\max})$:

Theorem 6.3.1. *I define $\epsilon = g(\gamma)$ where function g is strictly monotonic and differentiable of ϵ (i.e. $\gamma = g^{-1}(\epsilon)$). By applying the chain rule of differentiation, I can derive the PDF of error as follows:*

$$f_\epsilon(\epsilon) = f_\gamma\Big(g^{-1}(\epsilon)\Big)\left|\frac{d}{d\epsilon}g^{-1}(\epsilon)\right| \tag{6.3.13}$$

Where

$$g^{-1}(\epsilon) = -1 + \sqrt{\frac{\alpha Q^{-1}(\epsilon)}{W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)}} \tag{6.3.14}$$

$$\frac{dg^{-1}(\epsilon)}{d\epsilon} = 0.5\left[\frac{\alpha Q^{-1}(\epsilon)}{W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)\Big)}\right]^{-0.5}\frac{A}{W^2\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)\Big)} \tag{6.3.15}$$

and

$$\begin{aligned} A = &\frac{\alpha W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)\Big)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)} \\ &- \alpha Q^{-1}(\epsilon)\left[\frac{\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)} + \alpha Q^{-1}(\epsilon)\frac{\alpha\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)}\right. \\ &\left.+\frac{1}{\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)\Big)\exp\Big(W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta+\alpha Q^{-1}(\epsilon)\right)\Big)\Big)}\right] \end{aligned} \tag{6.3.16}$$

Here, W refers to the Lambert function, Φ symbolizes the CDF of the standard normal Gaussian distribution, and α is equivalent to $-\sqrt{\frac{1}{n}}\log_2 e/\ln 2$, where e represents Euler's number. Additionally, β is defined as $-2\frac{R}{\ln 2}$, with R signifying the code rate.

Proof. The proof is provided in Appendix C.2. □

6.3.5 Probability of Error

I calculate the average error probability of the mIoT system using the following formula:

$$P_e = \int \epsilon(\gamma_{\max})f_\epsilon(\epsilon)d\epsilon \tag{6.3.17}$$

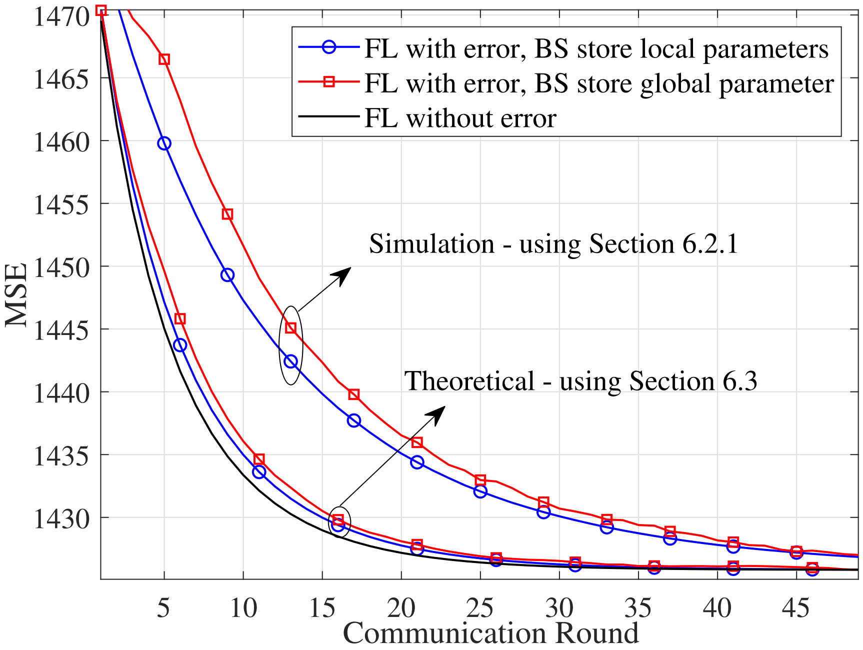


Figure 6.5: Comparing the FL convergence when the BS uses the theoretical model (Eq. 6.3.17) versus when it utilizes GMM clustering (section 6.2.1), in mIoT short packet communication with a rate 1/2 systematic convolutional encoder, learning rate $\eta = 0.05$, $\alpha = 0.95$, memory capacity $m = 5$, and with 1 iteration of GD performed at the device level.

6.4 Performance Evaluation of mIoT Network based on the Mathematical Model

In this section, I evaluate the performance of the mIoT system, leveraging the mathematical analysis presented in section 6.3. Subsequently, I compare these results with a scenario wherein users are transmitting data, and the system's performance is evaluated through simulation as outlined in section 6.2.1. The environment considered for this study involves a Rayleigh fading channel, with five power levels computed as $P_i = P_1 + (i - 1) \times 5\text{dB}$ for $i = 1, \cdots, 5$. The weakest user's power level is fixed at $\gamma_0 = 1dB$. Users, in a randomized manner, select one of the $N_t = 50$ time-slots along with one of the power levels, following which they commence transmission.

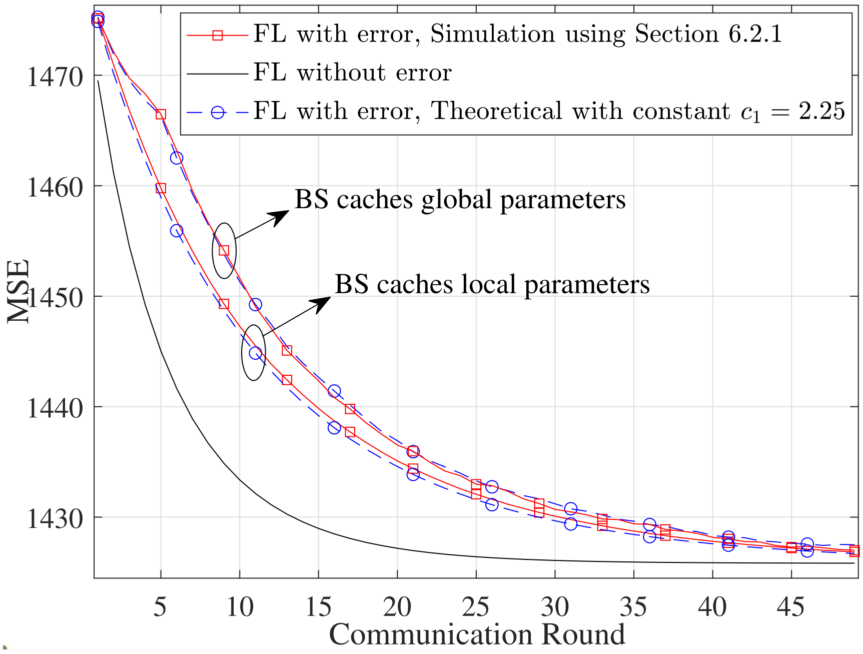


Figure 6.6: Comparing the FL convergence when the BS uses the theoretical model (Eq. 6.3.17) versus when it utilizes GMM clustering (section 6.2.1), in mIoT short packet communication with a rate 1/2 systematic convolutional encoder, learning rate $\eta = 0.05$, $\alpha = 0.95$, memory capacity $m = 5$, and with 1 iteration of GD performed at the device level.

In the theoretical analysis, I apply theorem 6.3.1 to compute the average probability of error via Equation 6.3.17. As established in Chapter 5, the efficacy of the FL algorithm can be gauged through the average error probability of the mIoT system. In Figure 6.5, I have analyzed the performance of FL through the average error probability as determined by the theoretical model. This is compared to the scenario where users are transmitting their packets. In the scenario wherein users are transmitting packets, the BS deploys GMM clustering, decodes the users, and employs the successfully decoded users in the learning process, as detailed in section 6.2.1.

As depicted in Fig. 6.5, a comparison between the evaluation of Theorem 6.3.1 and the simulation results provided in section 6.2.1 reveals a slight offset between the performance of the mathematical model and that of the GMM clustering model. This offset can be moderated by introducing a universal constant to Theorem 6.3.1. Specifically, by scaling β by a universal constant c_1, it is possible to align the performance of the mIoT system calculated from Theorem 6.3.1 precisely with the performance derived from the simulation in section 6.2.1.

In Figure 6.6, considering $N_t = 50$, $N_r = 1$, and $c_1 = 2.25$, I show that the performance of the proposed mathematical model aligns perfectly with the simulation results. This exact alignment provides robust validation for the analytical model proposed in this chapter.

6.5 Chapter Conclusion

In this chapter, I have simulated a real-world mIoT system, using the methodologies and themes introduced in earlier chapters. The aim of this chapter is to consolidate and expand upon the concepts presented in the previous chapters. I explored two distinct channel coding techniques. First, I analyzed a scenario in which users employ a pre-existing convolutional code and transmit a minimum number of pilot symbols for channel estimation. This approach is in line with the one discussed in Chapter 3. In contrast, the second scenario addressed a situation where users adopt RI coding and forgo the transmission of any pilot symbols, echoing the discourse from Chapter 4. Furthermore, I proposed a mathematical framework to estimate the error probability in a mIoT system. This framework enables the FL model to evaluate the learning performance of the mIoT system. The system's error probability can be predicted based on the average number of transmitting users. Subsequently, the learning performance of the mIoT system can be evaluated by utilizing the model proposed in

Chapter 5. The chapter revealed that when learning is prioritized, the need for absolute communication reliability can be tempered. The incorporation of the FL model achieves optimal learning outcomes, thereby demonstrating that the FL framework can mitigate the necessity for stringent communication reliability requirements.

Chapter 7

Conclusions and Future Work

This thesis provides a comprehensive exploration of the integration of communication and learning within the context of mIoT networks. The work leverages GMM clustering for channel estimation and signal detection, thus reducing the reliance on conventional CSI-based methods while achieving comparable or better performance with fewer pilot symbols. Moreover, it incorporates RI coding, eliminating the need for pilot symbols and showcasing robustness against channel rotation effects. Beyond communication, this thesis delves into the realm of learning, introducing FL within mIoT systems and exploring the optimization of various factors impacting FL's accuracy and convergence. It also investigates the use of past local or global parameters to compensate for the reliability issues in mIoT systems. Ultimately, this study synthesizes these findings in a simulated real-world mIoT system, demonstrating how innovative communication techniques and FL can create a balanced, efficient, and resilient mIoT network, significantly reducing the need for stringent communication reliability requirements. This thesis exemplifies a novel approach to bridging the gap between communication and learning within mIoT networks.

7.1 Summary

Chapter 2 provides the essential groundwork for the thesis by acquainting the reader with key concepts and techniques pivotal to the thesis. The chapter begins with an overview of NOMA and SIC receivers, establishing a foundational understanding of the communication framework employed in this work. It proceeds to review a range of channel estimation techniques, situating the research within the broader context of communication strategies. To further bolster the thesis context, an introduction to various clustering algorithms is provided, with a particular emphasis on the GMM clustering technique, which plays a central role in the research. Lastly, the chapter delves into the concept of FL as a method of distributed learning, setting the stage for the subsequent exploration of learning integration within mIoT networks.

Chapter 3 introduces an innovative approach for joint channel estimation and signal detection in grant-free NOMA by utilizing a semi-supervised ML algorithm, specifically GMM clustering. The method employs a SIC strategy for the detection of signals from multiple users. Importantly, it significantly reduces the need for pilot symbols without compromising performance, as demonstrated through comparisons with maximum likelihood detection that requires lengthy pilots at the receiver. The proposed GMM-based clustering algorithm presents a novel way to minimize computational burden, with a theoretical model developed to calculate the probability of error.

Chapter 4 extends the study to the application of RI coding in NOMA, negating the need for any pilot symbols for channel estimation. A combination of the clustering technique and RI coding is used for joint channel estimation and signal detection.

The technique optimizes decision boundaries based on clustering results, thereby mitigating the effects of channel rotation and improving the BER. The performance is shown to match that of maximum-likelihood detectors that require full CSI, further strengthening the argument for efficient, unsupervised methods.

Chapter 5 delves into the performance evaluation of FL in mIoT systems under the influence of communication errors. It models communication channels as packet erasure channels, proposing two strategies to improve FL performance amidst these challenges. The research examines the impact of factors like coding rate, block length, and SNR on the erasure probability. The significance of memory unit and the effect of reusing past local or global parameters in case of communication errors is also explored, offering practical ways to balance reliability requirements and accuracy in mIoT systems.

Finally, chapter 6 brings together all preceding work by simulating a real-world mIoT system, illustrating the practical application of the research. This chapter examines the system's performance in two scenarios: one employing a pilot-based GMM clustering technique, and the other a pilot-less approach, drawing from the methods proposed in Chapters 3 and 4. The chapter presents a mathematical framework for estimating the error probability in mIoT systems, allowing the FL model to evaluate the learning performance of the mIoT system. The findings underscore the possibility of relaxing communication reliability without compromising learning performance, offering a promising path forward for mIoT systems.

7.2 Future Work

The body of this thesis presented a novel communication approach tailored for mIoT systems, specifically focusing on the uplink through power-domain NOMA for single antenna users. The proposed integration of joint channel estimation and signal detection, leveraging clustering techniques, offers room for expansion and could be extended to scenarios involving downlink, BS equipped with multiple antennas, and even the application of code-domain NOMA. Following the trajectory set by this thesis, several potential extensions for future exploration emerge.

7.2.1 Utilizing GMM-based Detector in a Combined Code-domain and Power-domain NOMA

The integration of code-domain and power-domain NOMA techniques is suggested as a promising strategy to potentially accommodate a larger user base. Specifically, with code-domain NOMA serving as the foundation, additional users could be incorporated using power-domain NOMA. At the receiver end, the BS employs SIC and leverages the GMM clustering method to jointly estimate the channel and detect the signal. This proposition seeks to optimize user support within the NOMA framework.

7.2.2 Employment of Artificial Neural Networks for Estimating Updates of Erroneous Users

Chapter 5 introduced the concept of utilizing a memory cache at the BS to store global or local parameters from the previous communication round. A potential extension of this approach is to apply artificial neural networks (ANNs) to discern a pattern among users' local FL model parameters. Establishing this correlation would empower the BS to estimate the local model parameters for users unable to transmit their local models to the BS at each learning stage.

In this setup, the ANN's input for predicting user j's local FL model is a vector, ω_i, representing the local model of user i. User i maintains a consistent connection with the BS, providing input data for the multilayer perceptron to forecast the local FL models of other users. The output of the ANN for predicting user j's local FL model is a vector $o = \omega_i - \omega_j$, indicating the difference between user i's local model and user j's local parameter. Based on the predicted output o and user i's local FL model, one can ascertain the local FL model of user j, i.e., $\hat{\omega}_j = \omega_i - o$, with $\hat{\omega}_j$ denoting the predicted local parameter of user j. The hidden layer within an ANN facilitates the learning of non-linear relationships between the input vector ω_i and the output vector o.

7.2.3 Prioritizing Users with the Highest System Impact

During the learning process, different users contribute information with varying degrees of importance. For instance, when two users are located in close proximity, there is a high likelihood that the data they collect is similar. Under such circumstances, it might be feasible to disregard the input from one of them. Identifying these high-impact users would enable the system to streamline the process by focusing on crucial data and minimizing redundancy. A potential approach to pinpointing these key contributors involves plotting accuracy against MSE for all users. As illustrated in figure 7.1, users located on the lowest Pareto exert the most significant influence on learning. Upon the identification of these pivotal users, the BS could allocate them higher power to ensure accurate receipt of their packet.

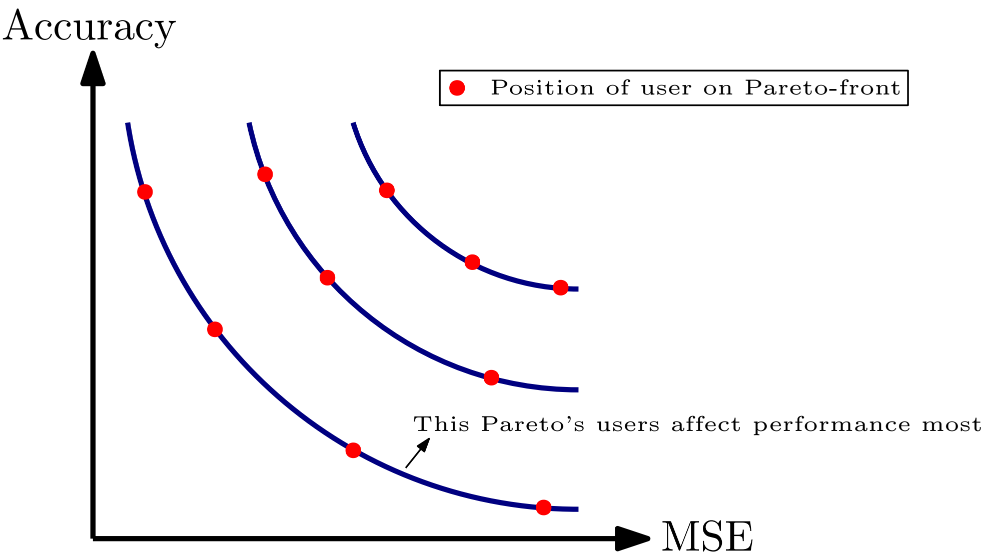

Accuracy
● Position of user on Pareto-front
This Pareto's users affect performance most
MSE

Figure 7.1: Pareto-front of Accuracy versus MSE

7.2.4 Non-orthogonal HARQ for Federated Learning in Wireless Communication

In chapter 5, it was demonstrated that in a mIoT scenario where users transmit data wirelessly, communication errors occur due to the limited power and insufficient bandwidth of the users. To address this issue, a memory element at the BS to aid in convergence was introduced. It was found that careful selection of the coding rate based on the SNR regime is crucial for achieving fast convergence of the FL algorithm. However, since wireless channels are dynamic, the SNR of each user varies over time, making the use of fixed coding rates inefficient. Hence, rate adaptation must be considered. Additionally, the research indicated that the updates from certain users could have a significant impact on the system, and failure of their packets can significantly impair system performance.

The reliability of most communication strategies is improved through the use of hybrid automatic repeat request (HARQ), but this comes at the cost of increased packet latency resulting from the retransmission of failed packets. In the case of mIoT, requesting the retransmission of unsuccessful packets is impractical due to the power limitations of users and the high rate of packet failure caused by a low SNR regime. In order to make HARQ a viable technique in mIoT, and to take advantage of its benefits, incremental redundancy HARQ along with NOMA can be utilized. This approach not only eliminates the need for retransmitting the entire packet but also removes packet delay.

The objective is to reduce the convergence time of FL in a mIoT system that suffers from erroneous communication. To achieve this, one can use an incremental redundancy HARQ technique. The proposed technique allows users who have failed to transmit their packets to send variable lengths of redundancy. It eliminates the need to retransmit the entire packet for a power-limited user while enabling the BS to decode the user's packet. Consequently, the negative impact of packet failure on the system's performance is eliminated. Additionally, one can consider a NOMA setup, where users superimpose their packets in the power domain, thus preventing queuing and excess delay at the BS.

Appendix A

Appendix for Chapter 3

A.1 Proof of Theorem 3.3.1

The proof is based on the findings of [169–171]. Without loss of generality, we focus on the update rule for one of the centers. We start by writing the update rule for the mean as

$$\boldsymbol{\mu}_1^+ - \boldsymbol{\mu}_1^* = \frac{\mathbb{E}[\gamma_1(X, \boldsymbol{\mu})(X - \boldsymbol{\mu}_1^*)]}{\mathbb{E}[\gamma_1(X, \boldsymbol{\mu})]}. \tag{A.1.1}$$

Since the vector of the true centers $\boldsymbol{\mu}^*$ is fixed, we have

$$\mathbb{E}[\gamma_1(X, \boldsymbol{\mu}_1^*)(X - \boldsymbol{\mu}_1^*)] = 0. \tag{A.1.2}$$

Hence, we can write

$$\boldsymbol{\mu}_1^+ - \boldsymbol{\mu}_1^* = \frac{\mathbb{E}\left[\left(\gamma_1(X, \boldsymbol{\mu}) - \gamma_1(X, \boldsymbol{\mu}^*)\right)(X - \boldsymbol{\mu}_1^*)\right]}{\mathbb{E}[\gamma_1(X, \boldsymbol{\mu})]}. \tag{A.1.3}$$

We find an upper bound on the norm of the expectation in the numerator. Therefore, we define

$$\boldsymbol{\mu}^t := \boldsymbol{\mu}^* + t(\boldsymbol{\mu} - \boldsymbol{\mu}^*)$$

$$g_X(t) := \gamma_1(X, \boldsymbol{\mu}^t).$$

Subsequently, we have

$$\begin{aligned}\gamma_1(X, \boldsymbol{\mu}) - \gamma_1(X, \boldsymbol{\mu}^*) &= \int_0^1 g_X'(t)dt \\ &= \int_0^1 \nabla_{\boldsymbol{\mu}} \gamma_1(x, \boldsymbol{\mu}_1^t)^T (\boldsymbol{\mu}_1^t - \boldsymbol{\mu}_1^*)dt. \qquad \text{(A.1.4)}\end{aligned}$$

Computing the integration and applying the expectation, the upper bound can be written as

$$\begin{aligned}&\|\mathbb{E}\left[(\gamma_1(X, \boldsymbol{\mu}) - \gamma_1(X, \boldsymbol{\mu}^*))(X - \boldsymbol{\mu}_1^*)\right]\|_2 \\ &\leq V_1 \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^*\|_2 + \sum_{i \neq 1} V_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^*\|_2 \qquad \text{(A.1.5)} \\ &\leq M \left(\max_i V_i\right)\left(\max_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^*\|_2\right) \qquad \text{(A.1.6)}\end{aligned}$$

where

$$V_1 = \sup_{t \in [0,1]} \|\mathbb{E}\left[\gamma_1(X; \boldsymbol{\mu}^t)(1 - \gamma_1(X; \boldsymbol{\mu}^t))(X - \boldsymbol{\mu}_1^*)(X - \boldsymbol{\mu}_1^t)^T\right]\|_{op} \qquad \text{(A.1.7)}$$

$$V_i = \sup_{t \in [0,1]} \|\mathbb{E}\left[\gamma_1(X; \boldsymbol{\mu}^t)\gamma_i(X; \boldsymbol{\mu}^t)(X - \boldsymbol{\mu}_1^*)(X - \boldsymbol{\mu}_i^t)^T\right]\|_{op}. \qquad \text{(A.1.8)}$$

Considering Z as the label of X, one can write

$$\mathbb{E}\left[\gamma_1(X; \boldsymbol{\mu}^*)\right] = \mathbb{E}\left[\mathbb{P}_{\boldsymbol{\mu}^*}(Z = 1 \mid X)\right] = \omega_1 > \kappa. \qquad \text{(A.1.9)}$$

When $\boldsymbol{\mu}$ is in the vicinity of $\boldsymbol{\mu}^*$, we have $\mathbb{E}\left[\gamma_1(X; \boldsymbol{\mu})\right] \approx \mathbb{E}\left[\gamma_1(X; \boldsymbol{\mu}^*)\right] > \kappa$. According to [170, Lemma 5.2], we know that, as long as $R_{\min} \geq 30 \min\{M_o, d\}^{0.5}$ and

$$a \geq \frac{1}{2} R_{\min} - \min\{M_o, d\}^{0.5} \max\{4\sqrt{2}[\log(\frac{R_{\min}}{4})]_+^{0.5}, 8\sqrt{3}, 8\log(\frac{4}{\kappa})\}, \qquad \text{(A.1.10)}$$

for any $\boldsymbol{\mu}_i \in \mathcal{B}(\boldsymbol{\mu}_i^*, a)$ $i \in [M_o]$, we have $\mathbb{E}\left[\gamma_i(X; \boldsymbol{\mu})\right] \geq \frac{3}{4}\kappa$ $i \in [M_o]$.

Using the above result and the upper bound (A.1.5), we have

$$\begin{aligned}\|\boldsymbol{\mu}_1^+ - \boldsymbol{\mu}_1^*\|_2 &= \frac{\|\mathbb{E}\left[(\gamma_1(X, \boldsymbol{\mu}) - \gamma_1(X, \boldsymbol{\mu}^*))(X - \boldsymbol{\mu}_1^*)\right]\|_2}{\mathbb{E}[\gamma_1(X, \boldsymbol{\mu})]} \\ &\leq \frac{4M_o}{3\kappa}\left(\max_i V_i\right)\left(\max_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^*\|_2\right). \qquad \text{(A.1.11)}\end{aligned}$$

Defining the event $\mathcal{E}_{1,i}$ as

$$\sup_{\boldsymbol{\mu}\in\mathcal{U}} \| \frac{1}{N_s}\sum_{j=1}^{N_s} \gamma_i(X_j;\boldsymbol{\mu})(X_j-\boldsymbol{\mu}_i^*) - \mathbb{E}[\gamma_i(X;\boldsymbol{\mu})(X-\boldsymbol{\mu}_i^*)]\|_2$$
$$\leq 1.5R_{max}\left(\frac{\hat{C}_3 M_o d\log N_s}{N_s}\right)^{0.5} \tag{A.1.12}$$

and the event $\mathcal{E}_{2,i}$ as

$$\sup_{\boldsymbol{\mu}\in\mathcal{U}} \mid \frac{1}{N_s}\sum_{j=1}^{N_s} \gamma_i(X_j;\boldsymbol{\mu}) - \mathbb{E}[\gamma_i(X;\boldsymbol{\mu})] \mid \leq \left(\frac{\hat{C}_2 M_o d\log N_s}{N_s}\right)^{0.5} \tag{A.1.13}$$

where

$$\mathcal{U} = \prod_{i=1}^{M_o}\mathcal{B}(\boldsymbol{\mu}^*, R_{\max})$$
$$\hat{C}_2 = C_2\log\left(M_o\left(2\sqrt{2}R_{\max}+\sqrt{d}\right)\right)$$
$$\hat{C}_3 = C_3\log\left(M_o\left(6R_{\max}^2+\sqrt{d}\right)\right),$$

and C_2 and C_3 are universal constants.

In view of [170, Lemmas 5.3 and 5.4], the event $\{\cap_{i\in[M_o]}\mathcal{E}_{1,i}\}\cap\{\cap_{i\in[M_o]}\mathcal{E}_{2,i}\}$ for all $i\in[M_o]$ occurs with the probability at least $1-\frac{2M_o}{N_s}$. Due to the sample size condition (3.3.2), we have

$$R_{\max}\left(\frac{\hat{C}_3 M_o d\log N_s}{N_s}\right)^{0.5} \leq \frac{\kappa}{3}\max_{i\in[M_o]}\|\boldsymbol{\mu}_i^0-\boldsymbol{\mu}_i^*\|_2 \tag{A.1.14}$$
$$\left(\frac{\hat{C}_2 M_o d\log N_s}{N_s}\right)^{0.5} \leq \frac{\kappa}{12}. \tag{A.1.15}$$

Using the definition of the event $\mathcal{E}_{2,i}$ for all $i\in[M_o]$, the second inequality (A.1.15) can be written as

$$\sup_{\boldsymbol{\mu}\in\mathcal{U}} \mid \frac{1}{N_s}\sum_{j=1}^{N_s} \gamma_i(X_j;\boldsymbol{\mu}) - \mathbb{E}[\gamma_i(X;\boldsymbol{\mu})] \mid \leq \frac{\kappa}{12}. \tag{A.1.16}$$

Taking $\boldsymbol{\mu}^{(0)}$ as the initial value of the mean, we can write

$$\|\boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_i^*\|_2 = \frac{\|\frac{1}{N_s}\sum_{j=1}^{N_s} \gamma_i(X_j;\boldsymbol{\mu}^{(0)})(X_j - \boldsymbol{\mu}_i^*)\|_2}{\frac{1}{N_s}\sum_{j=1}^{N_s} \gamma_i(X_j;\boldsymbol{\mu}^{(0)})} \tag{A.1.17}$$

$$\leq \frac{\|\mathbb{E}[\gamma_i(X;\boldsymbol{\mu}^{(0)})(X - \boldsymbol{\mu}_i^*)]\|_2 + R_{\max}\left(\frac{\hat{C}_3 M_o d \log N_s}{N_s}\right)^{0.5}}{\mathbb{E}[\gamma_i(X;\boldsymbol{\mu}^{(0)})] - \frac{\kappa}{12}} \tag{A.1.18}$$

$$\leq \frac{\|\mathbb{E}[\gamma_i(X;\boldsymbol{\mu}^{(0)})(X - \boldsymbol{\mu}_i^*)]\|_2 + R_{\max}\left(\frac{\hat{C}_3 M_o d \log(N_s)}{N_s}\right)^{0.5}}{\frac{2\kappa}{3}} \tag{A.1.19}$$

$$\leq \frac{1}{2}\max_{i\in[M_o]} \|\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*\|_2 + \frac{3R_{\max}\left(\hat{C}_3 M_o d \frac{\log N_s}{N_s}\right)^{0.5}}{2\kappa}. \tag{A.1.20}$$

Using the above inequality and the first sample size condition, we have

$$\|\boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_i^*\|_2 \leq \frac{1}{2}\max_{i\in[M_o]} \|\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*\|_2 + \frac{3R_{\max}\left(\hat{C}_3 M_o d \frac{\log(N_s)}{N_s}\right)^{0.5}}{2\kappa} \tag{A.1.21}$$

$$\leq \max_{i\in[M_o]} \|\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*\|_2. \tag{A.1.22}$$

By applying (A.1.17)-(A.1.20) over t iterations, we have

$$\max_{i\in[M_o]}\|\boldsymbol{\mu}_i^{(t)} - \boldsymbol{\mu}_i^*\|_2$$

$$\leq \frac{1}{2^t}\max_{i\in[M_o]} \|\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*\|_2 + \left(1 + \frac{1}{2} + \cdots + \frac{1}{2^{t-1}}\right)\frac{3R_{\max}\left(\hat{C}_3 M_o d \frac{\log(N_s)}{N_s}\right)^{0.5}}{2\kappa} \tag{A.1.23}$$

$$\leq \frac{1}{2^t}\max_{i\in[M_o]} \|\boldsymbol{\mu}_i^{(0)} - \boldsymbol{\mu}_i^*\|_2 + \frac{3R_{\max}\left(\hat{C}_3 M_o d \frac{\log(N_s)}{N_s}\right)^{0.5}}{\kappa}. \tag{A.1.24}$$

Appendix B

Appendix for Chapter 5

B.1 Proof of Theorem 5.2.1

The global model update can be written as

$$\omega^{(t+1)} = \omega^{(t)} - \eta^{(t)} \frac{\sum_{u=1}^{N_u} D_u \nabla F_u \left(\omega^{(t)}\right) I_u^{(t)}}{\sum_{u=1}^{N_u} D_u I_u^{(t)}}, \tag{B.1.1}$$

that is equivalent to $\omega^{(t+1)} = \omega^{(t)} - \eta^{(t)} \left(\nabla F(\omega^{(t)}) - Q\right)$. where

$$Q = \nabla F \left(\omega^{(t)}\right) - \frac{\sum_{u=1}^{N_u} D_u \nabla F_u \left(\omega^{(t)}\right) I_u}{\sum_{u=1}^{N_u} D_u I_u}. \tag{B.1.2}$$

Assuming that $v_1 = \omega^{(t+1)}$ and $v_2 = \omega^{(t)}$, we have $v_1 - v_2 = -\eta^{(t)} \left(\nabla F(\omega^{(t)}) - Q\right)$. Subsequently, since $F(.)$ is L-smooth, based on the Lipschitz's gradient (5.2.9), we have

$$\begin{aligned} F\left(\omega^{(t+1)}\right) \leq F\left(\omega^{(t)}\right) + \nabla F\left(\omega^{(t)}\right)^T \left(-\eta^{(t)}\left(\nabla F\left(\omega^{(t)}\right) - Q\right)\right) \\ + \frac{L}{2} \left\| -\eta^{(t)} (\nabla F(\omega^{(t)}) - Q) \right\|_2^2. \end{aligned} \tag{B.1.3}$$

Expanding the aforementioned equation, we have

$$\begin{aligned} F\left(\omega^{(t+1)}\right) \leq & F\left(\omega^{(t)}\right) - \eta^{(t)} \left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + \eta^{(t)} \nabla F\left(\omega^{(t)}\right)^T Q \\ & + \frac{L{\eta^{(t)}}^2}{2} \left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 - L{\eta^{(t)}}^2 \nabla F\left(\omega^{(t)}\right)^T Q \\ & + \frac{L{\eta^{(t)}}^2}{2} \parallel Q \parallel_2^2 . \end{aligned} \tag{B.1.4}$$

By fixing the learning rate to $\eta^{(t)} = \frac{1}{L}$, we have

$$F\left(\omega^{(t+1)}\right) \leq F\left(\omega^{(t)}\right) - \frac{1}{2L}\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + \frac{1}{2L} \parallel Q \parallel_2^2 . \tag{B.1.5}$$

Now, we calculate $\parallel Q \parallel^2$.

$$\begin{aligned}
\parallel Q \parallel_2^2 &= \left\|\nabla F\left(\omega^{(t)}\right) - \frac{\sum_{u=1}^{N_u} D_u \nabla F_u\left(\omega^{(t)}\right) I_u^{(t)}}{\sum_{u=1}^{N_u} D_u I_u^{(t)}}\right\|_2^2 \\
&= \left\|\nabla F\left(\omega^{(t)}\right) - \frac{\sum_{u=1}^{N_u} I_u^{(t)} \sum_{i=1}^{D_u} \nabla f(\omega, \boldsymbol{x}_u^i, y_u^i)}{\sum_{u=1}^{N_u} D_u I_u^{(t)}}\right\|_2^2 \\
&= \left\|\frac{1}{D}\sum_{u=1}^{N_u}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)\right. \\
&\qquad \left. - \frac{\sum_{u=1}^{N_u}\sum_{i=1}^{D_u} I_u^{(t)} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)}{\sum_{u=1}^{N_u} D_u I_u^{(t)}}\right\|_2^2 \\
&\overset{(i)}{=} \left\|\left(\sum_{u\in\mathcal{S}^{(t)}} D_u\right) \frac{\sum_{u\in\mathcal{S}^{(t)}}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)}{D\sum_{u\in\mathcal{S}^{(t)}} D_u}\right. \\
&\qquad + \left(\sum_{u\in\mathcal{S}^{(t)}} D_u\right) \frac{\sum_{u\in\mathcal{F}^{(t)}}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)}{D\sum_{u\in\mathcal{S}^{(t)}} D_u} \\
&\qquad \left. - \frac{D\sum_{u\in\mathcal{S}^{(t)}}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)}{D\sum_{u\in\mathcal{S}^{(t)}} D_u}\right\|_2^2 ,
\end{aligned} \tag{B.1.6}$$

where in step (i), we assumed that $\mathcal{S}^{(t)} = \{u|u \in \mathcal{U} \text{ and } I_u^{(t)} = 1\}$ and $\mathcal{F}^{(t)} = \{u|u \in \mathcal{U} \text{ and } I_u^{(t)} = 0\}$. This can be further expanded as follows:

$$
\begin{aligned}
\parallel Q \parallel_2^2 = & \left\| - \frac{\left(D - \sum_{u\in\mathcal{S}^{(t)}} D_u\right)\left(\sum_{u\in\mathcal{S}^{(t)}}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)\right)}{D\sum_{u\in\mathcal{S}^{(t)}} D_u} \right. \\
& \left. + \frac{\sum_{u\in\mathcal{F}^{(t)}}\sum_{i=1}^{D_u} \nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)}{D} \right\|_2^2 \\
\le & \frac{\left(D - \sum_{u\in\mathcal{S}^{(t)}} D_u\right)^2 \left(\sum_{u\in\mathcal{S}^{(t)}}\sum_{i=1}^{D_u} \left\|\nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)\right\|_2^2\right)}{\left(D\sum_{u\in\mathcal{S}^{(t)}} D_u\right)^2} \\
& + \frac{\sum_{u\in\mathcal{F}^{(t)}}\sum_{i=1}^{D_u} \left\|\nabla f(\omega^{(t)}, \boldsymbol{x}_u^i, y_u^i)\right\|_2^2}{D^2} \\
\overset{(ii)}{\le} & \left(D - \sum_{u\in\mathcal{S}^{(t)}} D_u\right)^2 \frac{\left(M\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + N\right)\left(\sum_{u\in\mathcal{S}^{(t)}} D_u\right)}{\left(D\sum_{u\in\mathcal{S}^{(t)}} D_u\right)^2} \\
& + \frac{\left(D - \sum_{u\in\mathcal{S}^{(t)}} D_u\right)\left(M\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + N\right)}{D^2} \\
= & \left(\frac{D - \sum_{u=1}^{N_u} I_u^{(t)} D_u}{D\sum_{u=1}^{N_u} I_u^{(t)} D_u}\right)\left(M\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + N\right) \\
\le & \frac{1}{D}\left(D - \sum_{u=1}^{N_u} D_u I_u^{(t)}\right)\left(M\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + N\right),
\end{aligned}
\tag{B.1.7}
$$

where in step (ii), we considered a bound on the $\mathbb{E}[\parallel \nabla f(\omega) \parallel^2]$, i.e., $\mathbb{E}[\parallel \nabla f(\omega) \parallel^2] \le M \parallel \nabla F(\omega) \parallel^2 + N$. By using the bound obtained above for $||Q||^2$, (B.1.5) can be rewritten as follows:

$$
\begin{aligned}
F\left(\omega^{(t+1)}\right) \le & F\left(\omega^{(t)}\right) - \frac{1}{2L}\left\|\nabla F\left(\omega^{(t)}\right)\right\|^2 \\
& + \frac{D - \sum_{u=1}^{N_u} D_u I_u^{(t)}}{2LD}\left(M\left\|\nabla F\left(\omega^{(t)}\right)\right\|^2 + N\right) \\
= & F\left(\omega^{(t)}\right) + \frac{N}{2LD}\left(D - \sum_{u=1}^{N_u} D_u I_u^{(t)}\right) \\
& - \frac{1 - M\frac{D - \sum_{u=1}^{N_u} D_u I_u^{(t)}}{D}}{2L}\left\|\nabla F\left(\omega^{(t)}\right)\right\|^2
\end{aligned}
\tag{B.1.8}
$$

Subtracting $F\left(\omega^{*}\right)$ from both sides and taking expectation, we have

$$\begin{aligned}
&\mathbb{E}\left[F\left(\omega^{(t+1)}\right)-F(\omega^{*})\right]\\
&\leq \mathbb{E}\left[F\left(\omega^{(t)}\right)-F(\omega^{*})\right]+\frac{N}{2L}\frac{\sum_{u=1}^{N_u} D_u\epsilon_u}{D}\\
&+\frac{1}{2L}\left(\frac{M\sum_{u=1}^{N_u} D_u\epsilon_u}{D}-1\right)\mathbb{E}\left[\left\|\nabla F\left(\omega^{(t)}\right)\right\|^2\right], \qquad \text{(B.1.9)}
\end{aligned}$$

which follows from the fact that $\mathbb{E}[I_u^{(t)}]=1-\epsilon_u$. Since the global loss function is strongly convex with a parameter μ, according to [172, Lemma 2.1], we have

$$\left\|\nabla F\left(\omega^{(t)}\right)\right\|^2 \geq 2\mu\left(F\left(\omega^{(t)}\right)-F(\omega^{*})\right) \qquad \text{(B.1.10)}$$

It should be noted that constants should satisfy $\mu \leq L$ [157]. Hence, if $\frac{M\sum_{u=1}^{N_u} D_u\epsilon_u}{D}-1<0$, we have:

$$\begin{aligned}
&\mathbb{E}\left[F\left(\omega^{(t+1)}\right)-F\left(\omega^{*}\right)\right]\\
&\leq \left(1-\frac{\mu}{L}\left(1-\frac{M\sum_{u=1}^{N_u}\epsilon_u}{D}\right)\right)\mathbb{E}\left[F\left(\omega^{(t)}\right)-F\left(\omega^{*}\right)\right]\\
&+\frac{N}{2L}\frac{\sum_{u=1}^{N_u}\epsilon_u}{D}. \qquad \text{(B.1.11)}
\end{aligned}$$

Considering $A=1-\frac{\mu}{L}\left(1-\frac{M\sum_{u=1}^{N_u} D_u\epsilon_u}{D}\right)$ and $B=\frac{N}{2L}\frac{\sum_{u=1}^{N_u} D_u\epsilon_u}{D}$, with recursion, we have

$$\begin{aligned}
&\mathbb{E}\left[F\left(\omega^{(t+1)}\right)-F\left(\omega^{*}\right)\right]\leq A\mathbb{E}\left[F\left(\omega^{(t)}\right)-F\left(\omega^{*}\right)\right]+B\\
&\leq \mathbb{E}\left[F\left(\omega^{(0)}\right)-F\left(\omega^{*}\right)\right]A^{t+1}+B\sum_{i=0}^{t}A^i\\
&=\mathbb{E}\left[F\left(\omega^{(0)}\right)-F\left(\omega^{*}\right)\right]A^{t+1}+B\frac{1-A^{t+1}}{1-A}. \qquad \text{(B.1.12)}
\end{aligned}$$

This completes the proof.

B.2 Proof of Theorem 5.2.2

Assuming that $D_i = D/N_u$, (5.2.17) can be written as follows:

$$\begin{aligned} w^{(t+1)} &= \frac{1}{N}\sum_{i\in\mathcal{S}(t)} w_i^{(t)} + \frac{1}{N}\sum_{j\in\mathcal{F}(t)} w_j^{(t-1)} \qquad \text{(B.2.1)} \\ &= \frac{1}{N}\sum_{i\in\mathcal{S}(t)} \left(w^{(t)} - \eta\nabla F_i(w^{(t)})\right) \\ &+ \frac{1}{N}\sum_{j\in\mathcal{F}(t)} \left(w^{(t-1)} - \eta\nabla F_j(w^{(t-1)})\right) \\ &= w^{(t)} - \eta\nabla F(w^{(t)}) + \frac{|\mathcal{F}(t)|}{N}\left(w^{(t-1)} - w^{(t)}\right) \\ &+ \eta\nabla F_{\mathcal{F}}(w^{(t)}) - \eta\nabla F_{\mathcal{F}}(w^{(t-1)}), \qquad \text{(B.2.2)} \end{aligned}$$

where $F_{\mathcal{F}}(x) = \frac{|\mathcal{F}(x)|}{N}\sum_{j\in\mathcal{F}(t)} F_j(x)$. Since $F(x)$ is convex and L-smooth, by using (5.2.9), we have:

$$\begin{aligned} &F(w^{(t+1)}) \leq F(w^{(t)}) + \nabla F(w^{(t)})\left(w^{(t+1)} - w^{(t)}\right)' \\ &+ \frac{L}{2}||w^{(t+1)} - w^{(t)}||_2^2 \\ &\overset{\text{(B.2.2)}}{=} F(w^{(t)}) - \eta||\nabla F(w^{(t)})||_2^2 \\ &+ \frac{|\mathcal{F}(t)|}{N}\nabla F(w^{(t)})(w^{(t-1)} - w^{(t)})' \\ &+ \eta\nabla F(w^{(t)})(\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)}))' \\ &+ \frac{L}{2}\eta^2||\nabla F(w^{(t)})||_2^2 + \frac{L|\mathcal{F}(t)|^2}{2N^2}||w^{(t-1)} - w^{(t)}||_2^2 \\ &+ \frac{L}{2}\eta^2||\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)})||_2^2 \\ &- \frac{L|\mathcal{F}(t)|}{N}\eta\nabla F(w^{(t)})(w^{(t-1)} - w^{(t)})' \\ &- L\eta^2\nabla F(w^{(t)})(\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)}))' \\ &- \frac{\eta L|\mathcal{F}(t)|}{N}(\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)}))(w^{(t)} - w^{(t-1)})' \end{aligned}$$

Now, assuming that $\eta = \frac{1}{L}$ and due to the fact that $|\mathcal{F}_F| \approx \epsilon N$, when N is sufficiently large, this can be simplified to:

$$\begin{aligned}
&F(w^{(t+1)}) \leq F(w^{(t)}) - \frac{1}{2L}||\nabla F(w^{(t)})||_2^2 \\
&+ \frac{L\epsilon^2}{2}||w^{(t)} - w^{(t-1)}||_2^2 + \frac{1}{2L}||\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)})||_2^2 \\
&- \epsilon(\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)}))(w^{(t)} - w^{(t-1)})' \\
&\overset{(a)}{\leq} F(w^{(t)}) - \frac{1}{2L}||\nabla F(w^{(t)})||_2^2 + \frac{L\epsilon^2}{2}||w^{(t)} - w^{(t-1)}||_2^2 \\
&- \frac{1}{2L}||\nabla F_{\mathcal{F}}(w^{(t)}) - \nabla F_{\mathcal{F}}(w^{(t-1)})||_2^2 \\
&\leq F(w^{(t)}) - \frac{1}{2L}||\nabla F(w^{(t)})||_2^2 + \frac{L}{2}\epsilon^2||w^{(t)} - w^{(t-1)}||_2^2, \qquad \text{(B.2.3)}
\end{aligned}$$

where step (a) follows from $||\nabla F(\omega^{(t)})||_2^2 \leq L^2||\omega^{(t)} - \omega^*||_2^2 - L^2||\omega^{(t+1)} - \omega^*||_2^2$ and [155, Lemma 1], which indicates that $F_{\mathcal{F}}(.)$ is convex and $L\epsilon$-smooth. Since we assumed that $||\nabla F(x) - \nabla F(y)||_2 \geq \mu||x - y||_2$, for all $x, y \in \mathbb{R}^d$, we have:

$$\begin{aligned}
F(w^{(t+1)}) \leq F(w^{(t)}) &- \frac{\mu^2}{2L}||w^{(t)} - w^*||_2^2 \\
&+ \frac{L}{2}\epsilon^2||w^{(t-1)} - w^{(t)}||_2^2. \qquad \text{(B.2.4)}
\end{aligned}$$

It is easy to show that $||w^{(t-1)} - w^{(t)}||_2^2 \leq 2\left(\delta_t + \delta_{t-1}\right)$. We can further simplify (B.2.4) as follows:

$$F(w^{(t+1)}) \leq F(w^{(t)}) + \left(L\epsilon^2 - \frac{\mu^2}{2L}\right)\delta_t + L\epsilon^2\delta_{t-1}. \qquad \text{(B.2.5)}$$

Summing up both sides over $t = 1, \cdots, k$, and using telescopic cancellation, we have:

$$\begin{aligned}
F(w^{(k+1)}) \leq F(w^{(0)}) &+ (2L\epsilon^2 - \frac{\mu^2}{2L})\sum_{i=1}^{k-1}\delta_i \\
&+ (L\epsilon^2 - \frac{\mu^2}{2L})(\delta_k + \delta_0), \qquad \text{(B.2.6)}
\end{aligned}$$

where in (B.2.6), we have assume that the first global update, i.e., when $t = 1$, is calculated without any communications error. That is $F(w^{(1)}) \leq F(w^0) - \frac{\mu}{2L}\delta_0$. Assuming that $\epsilon \leq \frac{\mu}{2L}$, we have $\frac{\mu^2}{2L} - 2L\epsilon^2 < \frac{\mu^2}{2L} - L\epsilon^2$. Therefore, (B.2.6) can be simplified as follows:

$$F(w^{(k+1)}) \leq F(w^{(0)}) - \beta^2(k+1)\bar{\delta}_{k+1}, \tag{B.2.7}$$

where $\beta^2 = \frac{\mu^2}{2L} - 2L\epsilon^2$. By rearranging the above inequality, we have:

$$\bar{\delta}_{k+1} \leq \frac{F(w^{(0)}) - F(w^{(k+1)})}{(k+1)\beta^2}. \tag{B.2.8}$$

Since $F(w^*) \leq F(w^{(k+1)})$, We will have:

$$\bar{\delta}_{k+1} \leq \frac{F(w^{(0)}) - F(w^*)}{(k+1)\beta^2}. \tag{B.2.9}$$

B.3 Proof of Theorem 5.2.3

Assuming that $D_u = D/N_u$, (5.1.10) can be written as follows:

$$\begin{aligned}\omega^{(t+1)} &= \frac{1}{N_u}\sum_{u\in\mathcal{S}(t)}\omega_u^{(t)} + \frac{1}{U}\sum_{u\in\mathcal{F}(t)}\omega^{(t-1)}\\ &= \frac{1}{N_u}\sum_{u\in\mathcal{S}(t)}\left(\omega^{(t)} - \eta\nabla F_u\left(\omega^{(t)}\right)\right) + \frac{|\mathcal{F}(t)|}{N_u}\omega^{(t-1)}\\ &= \omega^{(t)} - \eta\nabla F\left(\omega^{(t)}\right) + \frac{|\mathcal{F}(t)|}{N_u}\left(\omega^{(t-1)} - \omega^{(t)}\right)\\ &\quad + \eta\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right),\end{aligned} \tag{B.3.1}$$

where $F_{\mathcal{F}}(x) = \frac{|\mathcal{F}(x)|}{N_u}\sum_{u\in\mathcal{F}(t)}F_u(x)$. Since $F(x)$ is convex and L-smooth, by using (5.2.9), we have:

$$\begin{aligned}F\left(\omega^{(t+1)}\right) &\leq F\left(\omega^{(t)}\right) + \nabla F\left(\omega^{(t)}\right)\left(\omega^{(t+1)} - \omega^{(t)}\right)^T\\ &+ \frac{L}{2}\left\|\omega^{(t+1)} - \omega^{(t)}\right\|_2^2\\ &= F\left(\omega^{(t)}\right) - \eta\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 + \eta\nabla F\left(\omega^{(t)}\right)\left(\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right)\right)^T\\ &+ \frac{|\mathcal{F}(t)|}{N_u}\nabla F\left(\omega^{(t)}\right)\left(\omega^{(t-1)} - \omega^{(t)}\right)^T\\ &+ \frac{L}{2}\eta^2\left\|\nabla F(\omega^{(t)})\right\|_2^2 + \frac{L|\mathcal{F}(t)|^2}{2{N_u}^2}\left\|\omega^{(t-1)} - \omega^{(t)}\right\|_2^2\\ &+ \frac{L}{2}\eta^2\left\|\nabla F_{\mathcal{F}}(\omega^{(t)})\right\|_2^2 - \frac{L|\mathcal{F}(t)|}{N_u}\eta\nabla F\left(\omega^{(t)}\right)\left(\omega^{(t-1)} - \omega^{(t)}\right)^T\\ &- L\eta^2\nabla F\left(\omega^{(t)}\right)\left(\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right)\right)^T\\ &- \frac{\eta L|\mathcal{F}(t)|}{N_u}\left(\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right)\right)\left(\omega^{(t)} - \omega^{(t-1)}\right)^T.\end{aligned} \tag{B.3.2}$$

Now, assuming that $\eta = \frac{1}{L}$ and due to the fact that $|\mathcal{F}_F| \approx \epsilon N_u$, when N_u is sufficiently large, this can be simplified to:

$$\begin{aligned}
&F\left(\omega^{(t+1)}\right) \leq F\left(\omega^{(t)}\right) - \frac{1}{2L}\left\|\nabla F\left(\omega^{(t)}\right)\right\|_2^2 \\
&+ \frac{L\epsilon^2}{2}\left\|\omega^{(t)} - \omega^{(t-1)}\right\|_2^2 + \frac{1}{2L}\left\|\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right)\right\|_2^2 \\
&- \epsilon\left(\nabla F_{\mathcal{F}}\left(\omega^{(t)}\right)\right)\left(\omega^{(t)} - \omega^{(t-1)}\right)^T \\
&\overset{(a)}{\leq} F\left(\omega^{(t)}\right) - \frac{1}{2L}\left\|\nabla F(\omega^{(t)})\right\|_2^2 + \frac{L\epsilon^2}{2}\left\|\omega^{(t)} - \omega^{(t-1)}\right\|_2^2 \\
&+ \frac{L\epsilon^2}{2}\left\|\omega^{(t)} - \omega^*\right\|_2^2 - \frac{L\epsilon^2}{2}\left\|\omega^{(t+1)} - \omega^*\right\|_2^2
\end{aligned} \tag{B.3.3}$$

where step (a) follows from $||\nabla F(\omega^{(t)})||_2^2 \leq L^2||\omega^{(t)} - \omega^*||_2^2 - L^2||\omega^{(t+1)} - \omega^*||_2^2$ and [155, Lemma 1], which indicates that $F_{\mathcal{F}}(.)$ is convex and $L\epsilon$-smooth. Since we assumed that $||\nabla F(x) - \nabla F(y)||_2 \geq \mu||x - y||_2$, for all $x, y \in \mathbb{R}^d$, we have:

$$\begin{aligned}
F\left(\omega^{(t+1)}\right) \leq F\left(\omega^{(t)}\right) &- \frac{\mu^2}{2L}\left\|\omega^{(t)} - \omega^*\right\|_2^2 \\
&+ \frac{L}{2}\epsilon^2\left\|\omega^{(t-1)} - \omega^{(t)}\right\|_2^2 + \frac{L\epsilon^2}{2}\left\|\omega^{(t)} - \omega^*\right\|_2^2 \\
&- \frac{L\epsilon^2}{2}\left\|\omega^{(t+1)} - \omega^*\right\|_2^2
\end{aligned} \tag{B.3.4}$$

It is easy to show that $||\omega^{(t-1)} - \omega^{(t)}||_2^2 \leq 2\left(\delta_t + \delta_{t-1}\right)$. We can further simplify (B.3.4) as follows:

$$\begin{aligned}
F(\omega^{(t+1)}) \leq & F(\omega^{(t)}) + \left(L\epsilon^2 - \frac{\mu^2}{2L}\right)\delta_t + L\epsilon^2\delta_{t-1} \\
&+ \frac{L\epsilon^2}{2}\delta_t - \frac{L\epsilon^2}{2}\delta_{t+1}.
\end{aligned} \tag{B.3.5}$$

Summing up both sides over $t = 1, \cdots, k$, and using telescopic cancellation, we have:

$$\begin{aligned}
F\left(\omega^{(k+1)}\right) &\leq F\left(\omega^{(0)}\right) + \left(2L\epsilon^2 - \frac{\mu^2}{2L}\right)\sum_{i=1}^{k-1}\delta_i \\
&+ \left(L\epsilon^2 - \frac{\mu^2}{2L}\right)\left(\delta_k + \delta_0\right) + \frac{L\epsilon^2}{2}\delta_0 &\text{(B.3.6)} \\
&\leq F\left(\omega^{(0)}\right) + \left(2L\epsilon^2 - \frac{\mu^2}{2L}\right)\sum_{i=0}^{k}\delta_i, &\text{(B.3.7)}
\end{aligned}$$

where in (B.3.6), we assumed that the first global update, i.e., when $t = 1$, $\omega^{(1)}$ is calculated without any communications error. That is $F\left(\omega^{(1)}\right) \leq F\left(\omega^{(0)}\right) - \frac{\mu}{2L}\delta_0$. Assuming that $\epsilon \leq \frac{\mu}{2L}$, we have $\frac{\mu^2}{2L} - 2L\epsilon^2 < \frac{\mu^2}{2L} - L\epsilon^2$. Therefore, (B.3.7) can be simplified as follows:

$$F\left(\omega^{(k+1)}\right) \leq F\left(\omega^{(0)}\right) - \beta^2(k+1)\bar{\delta}_{k+1}, \tag{B.3.8}$$

where $\beta^2 = \frac{\mu^2}{2L} - 2L\epsilon^2$. By rearranging the above inequality, we have:

$$\bar{\delta}_{k+1} \leq \frac{F\left(\omega^{(0)}\right) - F\left(\omega^{(k+1)}\right)}{(k+1)\beta^2}. \tag{B.3.9}$$

Since $F(\omega^*) \leq F\left(\omega^{(k+1)}\right)$, We will have:

$$\bar{\delta}_{k+1} \leq \frac{F\left(\omega^{(0)}\right) - F(\omega^*)}{(k+1)\beta^2}. \tag{B.3.10}$$

This completes the proof.

Appendix C

Appendix for Chapter 6

C.1 Proof of Lemma 6.3.1

The goal is to calculate the distribution of

$$SINR_i = \frac{P_i|h_i|^2}{\sum_{j=i+1}^{N_u} P_j|h_j|^2 + 1}, \tag{C.1.1}$$

It is worth acknowledging that obtaining a direct derivation of the PDF of equation C.1.1 may pose substantial challenges. Therefore, I will independently calculate the PDF of the numerator and denominator. Then, employing the below-provided lemma, I will determine the PDF of the SINR given the number of users. This approach simplifies the computation by breaking down the problem into smaller, more manageable tasks.

Lemma C.1.1. *If $Y = U/V$ where U and V are are two independent random variables, the distribution of Y can be computed as*

$$f_Y(y) = \int_{-\infty}^{\infty} f_U(yz) f_V(z)|z|dz, \tag{C.1.2}$$

Proof. Please refer to [173]. □

In the following, I will calculate the PDF of the numerator and denominator of equation C.1.1 and then I will apply lemma C.1.1 and derive the PDF of SINR.

C.1.1 PDF of Numerator of Equation C.1.1

The channel power gain $|h(t)|^2$ which is proportional to instantaneous SNR, follows exponential distribution. The PDF of received SNR, γ can be written as

$$f_\gamma(\gamma) = \exp(-\gamma), \quad \gamma > 0 \tag{C.1.3}$$

The PDF of $Z = P|h|^2$ can be written as

$$F_Z(z) = \mathbb{P}(Z < z) = \mathbb{P}(P|h|^2 < z) = \sum_{i=1}^{N_p} \mathbb{P}(|h|^2 < \frac{z}{P}|P = P_i)\mathbb{P}(P = P_i) \tag{C.1.4}$$

$$= \sum_{i=1}^{N_p} \frac{1}{N_p} F_{|h|^2}(\frac{z}{P_i}) = \frac{1}{N_p}\sum_{i=1}^{N_p} 1 - \exp\Big(-\frac{\gamma}{P_i}\Big), \tag{C.1.5}$$

and the PDF of numerator can be written as

$$f_{\text{Num}}(\gamma) = \frac{1}{N_p}\sum_{i=1}^{N_p} \frac{1}{P_i}\exp\Big(-\frac{\gamma}{P_i}\Big), \tag{C.1.6}$$

C.1.2 PDF of Denominator of Equation C.1.1

The PDF of the sum of random variables is the convolution of the PDF of each variable. PDF of the denominator is a shifted PDF of $\sum_{i=1}^{N_u-1}\gamma_i$. I will start by calculating PDF of $\sum_{i=1}^{N_u-1}\gamma_i$.

$$f_{\sum_{i=1}^{N_u-1}\gamma_i}(\gamma \mid N_u) = f_{\gamma_1} * f_{\gamma_2} * \cdots * f_{\gamma_{N_u-1}} \tag{C.1.7}$$

Once I derived the PDF of $\sum_{i=1}^{N_u-1}\gamma_i$, I will use the fact that $y = \gamma + 1 \Rightarrow f_Y(y) = f_\gamma(y-1)$ and calculate the PDF of $\sum_{i=1}^{N_u-1}\gamma_i + 1$ that is the PDF of denominator or equation C.1.1. I will assume $\lambda_i = \frac{1}{|P_i|}$, and in the following paragraphs, I will calculate the PDF of $\sum_{i=1}^{N_u-1}\gamma_i$ given the number of users.

$\mathbf{N_u = 2}$:

In this case, since $f_{\sum_{i=1}^{1}\gamma_i}(\gamma \mid N_u = 2) = f_{\gamma_1}$, the PDF would be the same as equation C.1.6.

$\mathbf{N_u = 3}$:

I consider $x_3 = \gamma_1 + \gamma_2$, therefore the PDF can be derived as

$$f_{\sum_{i=1}^{2}\gamma_i}(x_3\,|N_u = 3) = \int_0^{x_3} f(\gamma_1) f(x_3 - \gamma_1) d\gamma_1 \tag{C.1.8}$$

$$\overset{\text{(C.1.6)}}{=} \int_0^{x_3} \frac{1}{{N_p}^2} \sum_{i_1=1}^{N_p} \lambda_{i_1} e^{-\lambda_{i_1}\gamma_1} . \sum_{i_2=1}^{N_p} \lambda_{i_2} e^{-\lambda_{i_2}(x_3-\gamma_1)} d\gamma_1 \tag{C.1.9}$$

$$= \frac{1}{{N_p}^2} \int_0^{x_3} \sum_{i_1=1}^{N_p} \sum_{i_2=1}^{N_p} \lambda_{i_1}\lambda_{i_2} e^{-\lambda_{i_2}x_3} e^{(\lambda_{i_2}-\lambda_{i_1})\gamma_1} d\gamma_1 \tag{C.1.10}$$

$$= \frac{1}{{N_p}^2} \sum_{i_1=1}^{N_p} \sum_{i_2=1}^{N_p} \lambda_{i_1}\lambda_{i_2} e^{-\lambda_{i_2}x_3} \int_0^{x_3} e^{(\lambda_{i_2}-\lambda_{i_1})\gamma_1} d\gamma_1 \tag{C.1.11}$$

$$= \frac{1}{{N_p}^2} \sum_{i_1=1}^{N_p} \sum_{i_2=1}^{N_p} \lambda_{i_1}\lambda_{i_2} e^{-\lambda_{i_2}x_3} \frac{1}{\lambda_{i_2} - \lambda_{i_1}} \left[e^{(\lambda_{i_2}-\lambda_{i_1})\gamma_1} \right]_0^{x_3} \tag{C.1.12}$$

$$\overset{(a)}{=} \frac{1}{{N_p}^2} \sum_{\substack{i_1=1\\ i\neq j}}^{N_p} \sum_{i_2=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}}{\lambda_{i_2} - \lambda_{i_1}} e^{-\lambda_{i_1}x_3} + \frac{\lambda_{i_1}\lambda_{i_2}}{\lambda_{i_1} - \lambda_{i_2}} e^{-\lambda_{i_2}x_3} + \frac{1}{N_p} \sum_{i_1=1}^{N_p} \lambda_{i_1}^2 x_3 e^{-\lambda_{i_1}x_3} \tag{C.1.13}$$

where in step (a), I use L'Hôpital's rule, that follows

$$\lim_{\lambda_{i_2}\to\lambda_{i_1}} \frac{e^{-\lambda_{i_1}x_3} - e^{-\lambda_{i_2}x_3}}{\lambda_{i_2} - \lambda_{i_1}} \overset{\text{Hop.}}{=} \frac{-x_3 e^{-\lambda_{i_1}x_3}}{-1} = x_3 e^{-\lambda_{i_1}x_3} \tag{C.1.14}$$

$\mathbf{N_u = 4}$:

I consider $x_4 = \gamma_1 + \gamma_2 + \gamma_3 = x_3 + \gamma_3$, then

$$f_{\sum_{i=1}^{3}\gamma_i}(x_4 \mid N_u = 4) = \int_0^{x_4} f(x_3) f(x_4 - x_3) dx_3 \tag{C.1.15}$$

$$\overset{\text{(C.1.13)}}{=} \int_0^{x_4} \Bigg[\frac{1}{{N_p}^2} \sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}}{\lambda_{i_2} - \lambda_{i_1}} e^{-\lambda_{i_1}x_3} + \frac{\lambda_{i_1}\lambda_{i_2}}{\lambda_{i_1} - \lambda_{i_2}} e^{-\lambda_{i_2}x_3} + \frac{1}{N_p} \sum_{i_1=1}^{N_p} \lambda_{i_1}^2 x_3 e^{-\lambda_{i_1}x_3} \Bigg] . \frac{1}{N_p} \sum_{i_3=1}^{N_p} \lambda_{i_3} e^{-\lambda_{i_3}(x_4-x_3)} dx_3 \tag{C.1.16}$$

$$= \int_0^{x_4} \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}}{\lambda_{i_2}-\lambda_{i_1}} e^{-\lambda_{i_3}x_4}\left(e^{(\lambda_{i_3}-\lambda_{i_1})x_3} - e^{(\lambda_{i_3}-\lambda_{i_2})x_3}\right)dx_3$$

$$+ \int_0^{x_4}\frac{1}{{N_p}^2}\sum_{i_1=1}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_3}\lambda_{i_1}^2 x_3 e^{-\lambda_{i_3}x_4}e^{(\lambda_{i_3}-\lambda_{i_1})x_3}dx_3 \tag{C.1.17}$$

Now I will start with the first term of (C.1.17).

$$\int_0^{x_4} \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}}{\lambda_{i_2}-\lambda_{i_1}} e^{-\lambda_{i_3}x_4}\left(e^{(\lambda_{i_3}-\lambda_{i_1})x_3} - e^{(\lambda_{i_3}-\lambda_{i_2})x_3}\right)dx_3$$

$$= \int_0^{x_4} \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}}{\lambda_{i_2}-\lambda_{i_1}} e^{-\lambda_{i_3}x_4}\left(e^{(\lambda_{i_3}-\lambda_{i_1})x_3} - e^{(\lambda_{i_3}-\lambda_{i_2})x_3}\right)dx_3 \tag{C.1.18}$$

$$= \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p} \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}}{\lambda_{i_2}-\lambda_{i_1}}\left[\frac{1}{\lambda_{i_3}-\lambda_{i_1}}\left(e^{(\lambda_{i_3}-\lambda_{i_1})x_3}\right) - \frac{1}{\lambda_{i_3}-\lambda_{i_2}}\left(e^{(\lambda_{i_3}-\lambda_{i_2})x_3}\right)\right]_{x_3=0}^{x_4} \tag{C.1.19}$$

$$= \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2\neq i_3}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\left[\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_1}x_4}}{(\lambda_{i_2}-\lambda_{i_1})(\lambda_{i_3}-\lambda_{i_1})} + \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_2}x_4}}{(\lambda_{i_1}-\lambda_{i_2})(\lambda_{i_3}-\lambda_{i_2})}\right.$$

$$\left. + \frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}}{(\lambda_{i_1}-\lambda_{i_3})(\lambda_{i_2}-\lambda_{i_3})}\right]$$

$$+ \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1=i_3\neq i_2}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}\left[\frac{x_4}{\lambda_{i_2}-\lambda_{i_1}} + \frac{e^{(\lambda_{i_3}-\lambda_{i_2})x_4}-1}{(\lambda_{i_3}-\lambda_{i_2})(\lambda_{i_1}-\lambda_{i_2})}\right]$$

$$+ \frac{1}{{N_p}^3} \sum_{\substack{i_1=1\\ i_1\neq i_2=i_3}}^{N_p} \sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}\left[\frac{x_4}{\lambda_{i_1}-\lambda_{i_2}} + \frac{e^{(\lambda_{i_3}-\lambda_{i_1})x_4}-1}{(\lambda_{i_2}-\lambda_{i_1})(\lambda_{i_3}-\lambda_{i_1})}\right] \tag{C.1.20}$$

The second term of (C.1.17) can be computed as

$$\int_0^{x_4}\frac{1}{{N_p}^2}\sum_{i_1=1}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_3}\lambda_{i_1}^2 x_3 e^{-\lambda_{i_3}x_4}e^{(\lambda_{i_3}-\lambda_{i_1})x_3}dx_3 \tag{C.1.21}$$

$$=\frac{1}{{N_p}^2}\sum_{\substack{i_1=1\\ i_1=i_2\neq i_3}}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_1}^2\lambda_{i_3}e^{-\lambda_{i_1}x_4}\Big[\frac{x_4}{\lambda_{i_3}-\lambda_{i_1}}+\frac{e^{(\lambda_{i_1}-\lambda_{i_3})x_4}-1}{(\lambda_{i_3}-\lambda_{i_1})^2}\Big]+\frac{1}{2!}\frac{1}{N_p}\sum_{i_1=1}^{N_p}\lambda_{i_1}^3{x_4}^2e^{-\lambda_{i_1}x_4} \tag{C.1.22}$$

Therefore the PDF for the case of $N_u = 4$ can be written as

$$\begin{aligned}
f(x_4 \mid N_u = 4) = \frac{1}{{N_p}^3}\sum_{\substack{i_1=1\\ i_1\neq i_2\neq i_3}}^{N_p}\sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\Big[&\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_1}x_4}}{(\lambda_{i_2}-\lambda_{i_1})(\lambda_{i_3}-\lambda_{i_1})}+\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_2}x_4}}{(\lambda_{i_1}-\lambda_{i_2})(\lambda_{i_3}-\lambda_{i_2})}\\
&+\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}}{(\lambda_{i_1}-\lambda_{i_3})(\lambda_{i_2}-\lambda_{i_3})}\Big]\\
&+\frac{1}{{N_p}^2}\sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p}\sum_{i_2=1}^{N_p}\lambda_{i_1}\lambda_{i_2}^2e^{-\lambda_{i_2}x_4}(\frac{x_4}{\lambda_{i_1}-\lambda_{i_2}}+\frac{e^{(\lambda_{i_2}-\lambda_{i_1})x_4}-1}{(\lambda_{i_1}-\lambda_{i_2})^2})\\
&+\frac{1}{{N_p}^2}\sum_{\substack{i_2=1\\ i_2\neq i_3}}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_2}\lambda_{i_3}^2e^{-\lambda_{i_3}x_4}(\frac{x_4}{\lambda_{i_2}-\lambda_{i_3}}+\frac{e^{(\lambda_{i_3}-\lambda_{i_2})x_4}-1}{(\lambda_{i_2}-\lambda_{i_3})^2})\\
&+\frac{1}{{N_p}^2}\sum_{\substack{i_1=1\\ i_1\neq i_3}}^{N_p}\sum_{i_3=1}^{N_p}\lambda_{i_3}\lambda_{i_1}^2e^{-\lambda_{i_1}x_4}(\frac{x_4}{\lambda_{i_3}-\lambda_{i_1}}+\frac{e^{(\lambda_{i_1}-\lambda_{i_3})x_4}-1}{(\lambda_{i_3}-\lambda_{i_1})^2})\\
&+\frac{1}{2!}\frac{1}{N_p}\sum_{i_1=1}^{N_p}\lambda_{i_1}^3{x_4}^2e^{-\lambda_{i_1}x_4}
\end{aligned} \tag{C.1.23}$$

As can be seen by changing $\lambda_{i_1} \to \lambda_{i_2}$, $\lambda_{i_2} \to \lambda_{i_3}$, and $\lambda_{i_3} \to \lambda_{i_1}$, the second, third, and fourth terms will be equal. Therefore, one can summarize (C.1.23) as

$$\begin{aligned}
f(x_4 \mid N_u = 4) = \frac{1}{{N_p}^3}\sum_{\substack{i_1=1\\ i_1\neq i_2\neq i_3}}^{N_p}\sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\Bigg[&\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_1}x_4}}{(\lambda_{i_2}-\lambda_{i_1})(\lambda_{i_3}-\lambda_{i_1})}+\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_2}x_4}}{(\lambda_{i_1}-\lambda_{i_2})(\lambda_{i_3}-\lambda_{i_2})}\\
&+\frac{\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}e^{-\lambda_{i_3}x_4}}{(\lambda_{i_1}-\lambda_{i_3})(\lambda_{i_2}-\lambda_{i_3})}\Bigg]\\
+\frac{3}{{N_p}^2}\sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p}\sum_{i_2=1}^{N_p}\lambda_{i_1}^2\lambda_{i_2}e^{-\lambda_{i_1}x_4}&\left(\frac{x_4}{\lambda_{i_2}-\lambda_{i_1}}+\frac{e^{(\lambda_{i_1}-\lambda_{i_2})x_4}-1}{(\lambda_{i_2}-\lambda_{i_1})^2}\right)+\frac{1}{2!}\frac{1}{N_p}\sum_{i_1=1}^{N_p}\lambda_{i_1}^3{x_4}^2e^{-\lambda_{i_1}x_4}
\end{aligned} \tag{C.1.24}$$

C.1.3 PDF of Equation C.1.1

In this subsection, I will use the lemma C.1.1 and based on the PDF of numerator and denominators, compute the PDF of SINR.

For the case of $N_u = 1$, since there is no interference, the PDF of SINR is the same as PDF of SNR (Eq. C.1.6)

For the case of $N_u = 2$, the PDF of SINR can be calculated as

$$f_{SINR}(\gamma \mid N_u = 2) = \int_{-\infty}^{\infty} f_{\text{Num}}(\gamma y) f_{\text{Denom}}(y-1)|y|dy, \tag{C.1.25}$$

At this point, it is worth highlighting why I have adopted $f_{\text{Denom}}(y-1)$. This arises from the fact that $f_{\sum_{i=1}^{1}\gamma_i}(\gamma) = f_{\text{Denom}}(y)$ and since there is an additional one in the denominator, it is necessary to represent it as $f_{\sum_{i=1}^{1}\gamma_i+1}(\gamma) = f_{\text{Denom}}(y-1)$. This adjustment accommodates the additional factor in the original distribution. The PDF of SINR can be written as

$$f_{SINR}(\gamma \mid N_u = 2) = \int_{-\infty}^{\infty} f_{\text{Num}}(\gamma y) f_{\text{Denom}}(y-1 \mid N_u = 2)|y|dy \tag{C.1.26}$$

$$= \int_{0}^{\infty} y \frac{1}{N_p} \sum_{i=1}^{N_p} \lambda_i e^{-\lambda_i \gamma y} \frac{1}{N_p} \sum_{j=1}^{N_p} \lambda_j e^{-\lambda_j y - \lambda_j} dy \tag{C.1.27}$$

$$= \frac{1}{{N_p}^2} \sum_{i=1}^{N_p} \lambda_i \lambda_j \int_{0}^{\infty} y e^{-\lambda_i \gamma y - \lambda_j y + \lambda_j} dy \tag{C.1.28}$$

$$= \frac{1}{N_p^2} \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \lambda_i \lambda_j \frac{\exp(\lambda_i)}{(\lambda_i + \lambda_j \gamma)^2} \tag{C.1.29}$$

For the case of $N_u = 3$, the PDF of SINR can be calculated as

$$\begin{aligned} f_{SINR}(\gamma \mid N_u = 3) &= \int_{-\infty}^{\infty} f_{\text{Num}}(\gamma y) f_{\text{Denom}}(y - 1 \mid N_u = 3)|y|dy \qquad \text{(C.1.30)} \\ &= \frac{1}{N_p^3} \sum_{\substack{i_1=1 \\ i_1 \neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_j \int_0^{\infty} \frac{y e^{-\lambda_j \gamma y - \lambda_{i_1} y + \lambda_{i_1}}}{\lambda_{i_2} - \lambda_{i_1}} + \frac{y e^{-\lambda_j \gamma y - \lambda_{i_2} y + \lambda_{i_2}}}{\lambda_{i_1} - \lambda_{i_2}} dy \\ &+ \frac{1}{N_p^2} \sum_{i_1=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1}^2 \lambda_j \int_0^{\infty} y^2 e^{-\lambda_j \gamma y - \lambda_{i_1} y + \lambda_{i_1}} dy \qquad \text{(C.1.31)} \\ &= \frac{1}{N_p^3} \sum_{\substack{i_1=1 \\ i_1 \neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_j \left(\frac{\exp(\lambda_{i_1})}{(\lambda_{i_2} - \lambda_{i_1})(\lambda_{i_1} + \lambda_j \gamma)^2} \right. \\ &\qquad \left. + \frac{\exp(\lambda_{i_2})}{(\lambda_{i_1} - \lambda_{i_2})(\lambda_{i_2} + \lambda_j \gamma)^2} \right) \\ &+ \frac{1}{N_p^2} \sum_{i_1=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1}^2 \lambda_j \frac{2 \exp(\lambda_{i_1})}{(\lambda_{i_1} + \lambda_j \gamma)^3}, \qquad \text{(C.1.32)} \end{aligned}$$

For the case of $N_u = 4$, the PDF of SINR can be calculated as

$$\begin{aligned} f_{SINR}(\gamma \mid N_u = 4) &= \frac{1}{N_p^4} \sum_{\substack{i_1=1 \\ i_1 \neq i_2 \neq i_3}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{i_3=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_{i_3} \lambda_j \int_0^{\infty} \left[\frac{y e^{-\lambda_j \gamma y - \lambda_{i_1} y - \lambda_{i_1}}}{(\lambda_{i_2} - \lambda_{i_1})(\lambda_{i_3} - \lambda_{i_1})} \right. \\ &\qquad \left. + \frac{y e^{-\lambda_j \gamma y - \lambda_{i_2} y - \lambda_{i_2}}}{(\lambda_{i_1} - \lambda_{i_2})(\lambda_{i_3} - \lambda_{i_2})} + \frac{y e^{-\lambda_j \gamma y - \lambda_{i_3} y - \lambda_{i_3}}}{(\lambda_{i_1} - \lambda_{i_3})(\lambda_{i_2} - \lambda_{i_3})} \right] dy \\ &+ \frac{3}{N_p^3} \sum_{\substack{i_1=1 \\ i_1 \neq i_2}}^{N_p} \sum_{i_2=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1} \lambda_{i_2} \lambda_j \int_0^{\infty} y e^{-\lambda_j \gamma y - \lambda_{i_1} y - \lambda_{i_1}} \left(\frac{y - 1}{\lambda_{i_2} - \lambda_{i_1}} \right. \\ &\qquad \left. + \frac{e^{(\lambda_{i_1} - \lambda_{i_2})(y-1)} - 1}{(\lambda_{i_2} - \lambda_{i_1})^2} \right) dy \\ &+ \frac{1}{2! N_p^2} \sum_{i_1=1}^{N_p} \sum_{j=1}^{N_p} \lambda_{i_1}^3 \lambda_j \int_0^{\infty} y^3 e^{-\lambda_j \gamma y - \lambda_{i_1} y - \lambda_{i_1}} dy \qquad \text{(C.1.33)} \end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{N_p^4}\sum_{\substack{i_1=1\\ i_1\neq i_2\neq i_3}}^{N_p}\sum_{i_2=1}^{N_p}\sum_{i_3=1}^{N_p}\sum_{j=1}^{N_p}\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}\lambda_j\left(\frac{\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_2}-\lambda_{i_1}\right)\left(\lambda_{i_3}-\lambda_{i_1}\right)\left(\lambda_{i_1}+\lambda_j\gamma\right)^2}\right.\\
&\qquad+\frac{\exp\left(\lambda_{i_2}\right)}{\left(\lambda_{i_1}-\lambda_{i_2}\right)\left(\lambda_{i_3}-\lambda_{i_2}\right)\left(\lambda_{i_2}+\lambda_j\gamma\right)^2}\\
&\qquad\left.+\frac{\exp\left(\lambda_{i_3}\right)}{\left(\lambda_{i_1}-\lambda_{i_3}\right)\left(\lambda_{i_2}-\lambda_{i_3}\right)\left(\lambda_{i_3}+\lambda_j\gamma\right)^2}\right)\\
&+\frac{3}{N_p^3}\sum_{\substack{i_1=1\\ i_1\neq i_2}}^{N_p}\sum_{i_2=1}^{N_p}\sum_{j=1}^{N_p}\lambda_{i_1}\lambda_{i_2}\lambda_j\left(\frac{-\exp\left(\lambda_{i_1}\right)\left(\lambda_{i_1}+\lambda_j\gamma-2\right)}{\left(\lambda_{i_2}-\lambda_{i_1}\right)\left(\lambda_{i_1}+\lambda_j\gamma\right)^3}\right.\\
&\qquad+\frac{\exp\left(\lambda_{i_2}\right)}{\left(\lambda_{i_2}-\lambda_{i_1}\right)^2\left(\lambda_{i_2}+\lambda_j\gamma\right)^2}\\
&\qquad\left.-\frac{\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_2}-\lambda_{i_1}\right)^2\left(\lambda_{i_1}+\lambda_j\gamma\right)^2}\right)\\
&+\frac{1}{2!N_p^2}\sum_{i_1=1}^{N_p}\sum_{j=1}^{N_p}\lambda_{i_1}^3\lambda_j\frac{6\exp\left(\lambda_{i_1}\right)}{\left(\lambda_{i_1}+\lambda_j\gamma\right)^4},
\end{aligned}
\tag{C.1.34}
$$

C.2 Proof of Theorem 6.3.1

The goal is to calculate the PDF of the packet error rate that can be written as

$$f_\epsilon(\epsilon) = f_\gamma\Big(g^{-1}(\epsilon)\Big)\left|\frac{d}{d\epsilon}g^{-1}(\epsilon)\right| \tag{C.2.1}$$

The PDF for f_γ has been previously computed. Subsequently, I will tackle the calculation of $g^{-1}(\epsilon)$ and the derivative of $g^{-1}(\epsilon)$ with respect to ϵ in two separate subsections.

C.2.1 Calculating $\gamma = g^{-1}(\epsilon)$

In the following, I will calculate γ as a function of ϵ. Owing to the fact that

$$R \approx \log_2(1+\gamma) - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon)\log_2 e \tag{C.2.2}$$

where $V = 1 - \frac{1}{(1+\gamma)^2}$.

Writing Taylor series for $\sqrt{1-x}$, one can derive

$$\sqrt{1-x} = 1 + \sum_{j=1}^{\infty}\binom{1/2}{j}(-x)^j = 1 - \frac{x}{2} - \frac{x^2}{8} - \frac{x^3}{16} - \cdots, \tag{C.2.3}$$

where $b_j = \left|\binom{1/2}{j}\right| = \left|\frac{1/2(1/2-1)\cdots(1/2-j+1)}{j!}\right|$; hence,

$$\sqrt{V} = 1 - \sum_{j=1}^{\infty}\frac{b_j}{(1+\gamma)^{2j}} \tag{C.2.4}$$

In [174], it has been shown that considering only the first term in the summation will lead to a very tight approximation when the instantaneous SNR is more than −2dB. It is worth mentioning that for SNR more than 10dB, one can approximate $V = 1$, and for SNR's less than −2dB, three terms of summation should be considered for a tight approximation. In a mIoT scenario with power domain NOMA, one can safely say that SNR is more than −2dB. Therefore, in the following I calculate the SNR as a function of probability of error (ϵ) by considering only one term in the summation of (C.2.4).

I consider $x = 1 + \gamma$, $\alpha_1 = \sqrt{\frac{1}{n}} Q^{-1}(\epsilon) \log_2 e/\ln 2$ and $\beta_1 = \frac{R}{\ln 2}$.

$$\beta_1 = \ln x - \alpha_1 (1 - \frac{1}{2x^2}) \tag{C.2.5}$$

$$\overset{\alpha_1 + \beta_1 = \beta_2}{\Longrightarrow} \ln x + \frac{\alpha_1}{2} x^{-2} = \beta_2 \tag{C.2.6}$$

$$\Longrightarrow \frac{-1}{2} \ln(x^{-2}) + \frac{\alpha_1}{2} x^{-2} = \beta_2 \tag{C.2.7}$$

Assuming $\alpha_2 = -\alpha_1, \beta_3 = -2\beta_2$, one can write

$$e^{\ln x^{-2}} . e^{\alpha_2 x^{-2}} = e^{\beta_3} \tag{C.2.8}$$

$$\Longrightarrow \alpha_2 x^{-2} e^{\alpha_2 x^{-2}} = \alpha_2 e^{\beta_3} \tag{C.2.9}$$

$$\Longrightarrow \alpha_2 x^{-2} = W(\alpha_2 e^{\beta_3}) \Longrightarrow x = \sqrt{\frac{\alpha_2}{W(\alpha_2 e^{\beta_3})}}, \tag{C.2.10}$$

where W is the Lambert function. Subsequently,

$$g^{-1}(\epsilon) = -1 + \sqrt{\frac{-\sqrt{\frac{1}{n}} Q^{-1}(\epsilon) \log_2 e/\ln 2}{W\Bigg(- \left(\sqrt{\frac{1}{n}} Q^{-1}(\epsilon) \log_2 e/\ln 2\right) \exp\left(\frac{-2R}{\ln 2} - 2\sqrt{\frac{1}{n}} Q^{-1}(\epsilon) \log_2 e/\ln 2\right)\Bigg)}} \tag{C.2.11}$$

C.2.2 Calculating $\frac{dg^{-1}(\epsilon)}{d\epsilon}$

Next, $\frac{d}{d\epsilon} g^{-1}(\epsilon)$ in (C.2.1) should be calculated. Using the following definitions

1. $\left(f^{-1}\right)'(x) = \dfrac{1}{f'\left(f^{-1}(x)\right)} \Rightarrow \left(Q^{-1}\right)'(x) = \dfrac{1}{\Phi'\left(\Phi^{-1}(x)\right)}$

   where $\Phi'(x) = \dfrac{1}{\sqrt{2\pi}} \exp(-\dfrac{x^2}{2})$, $\Phi^{-1} = \sqrt{2}\mathrm{erf}^{-1}(2x - 1)$

2. $\dfrac{df\left(g(x)\right)}{dx} = f'\left(g(x)\right).g'(x)$

3. $\dfrac{dW(x)}{dx} = \dfrac{1}{x + e^{W(x)}} \Rightarrow \dfrac{dW\left(f(x)\right)}{dx} = \dfrac{f'(x)}{f(x) + e^{W\left(f(x)\right)}}$

Assuming $\alpha = -\sqrt{\frac{1}{n}}\log_2 e/\ln 2$, $\beta = -2\frac{R}{\ln 2}$, then

$$\frac{dg^{-1}(\epsilon)}{d\epsilon} = 0.5\left[\frac{\alpha Q^{-1}(\epsilon)}{W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)}\right]^{-0.5} \frac{A}{W^2\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)} \tag{C.2.12}$$

where

$$\begin{aligned} A = & \frac{\alpha W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)} \\ & - \alpha Q^{-1}(\epsilon)\left[\frac{\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)} + \alpha Q^{-1}(\epsilon)\frac{\alpha\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)}{\Phi'\Big(\Phi^{-1}(\epsilon)\Big)}\right. \\ & \left. + \frac{1}{\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)\exp\Big(W\Big(\alpha Q^{-1}(\epsilon)\exp\left(\beta + \alpha Q^{-1}(\epsilon)\right)\Big)\Big)}\right] \end{aligned} \tag{C.2.13}$$

Bibliography

[1] V. Dala Pegorara Souto, P. S. Dester, M. Soares Pereira Facina, D. Gomes Silva, F. A. P. de Figueiredo, G. Rodrigues de Lima Tejerina, J. C. Silveira Santos Filho, J. Silveira Ferreira, L. L. Mendes, R. D. Souza *et al.*, "Emerging MIMO technologies for 6G networks," *Sensors*, vol. 23, no. 4, p. 1921, 2023.

[2] S. Alraih, I. Shayea, M. Behjati, R. Nordin, N. F. Abdullah, A. Abu-Samah, and D. Nandi, "Revolution or evolution? technical requirements and considerations towards 6G mobile communications," *Sensors*, vol. 22, no. 3, p. 762, 2022.

[3] R. Dangi, P. Lalwani, G. Choudhary, I. You, and G. Pau, "Study and investigation on 5G technology: A systematic review," *Sensors*, vol. 22, no. 1, p. 26, 2022.

[4] M. Kamel, W. Hamouda, and A. Youssef, "Uplink coverage and capacity analysis of mMTC in ultra-dense networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 746–759, 2019.

[5] K. Mikhaylov, V. Petrov, R. Gupta, M. A. Lema, O. Galinina, S. Andreev, Y. Koucheryavy, M. Valkama, A. Pouttu, and M. Dohler, "Energy efficiency of multi-radio massive machine-type communication (MR-MMTC): Applications, challenges, and solutions," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 100–106, 2019.

[6] J. Liu and X. Wang, "A grant-based random access scheme with low latency for mMTC in IoT networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[7] B. Manzoor, B. Al Homssi, and A. Al-Hourani, "IoT coverage enhancement using repetition in energy constrained devices: An analytic approach," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 1122–1131, 2021.

[8] A. M. Escolar, J. M. Alcaraz-Calero, P. Salva-Garcia, J. B. Bernabe, and Q. Wang, "Adaptive network slicing in multi-tenant 5G IoT networks," *IEEE Access*, vol. 9, pp. 14 048–14 069, 2021.

[9] C. Bockelmann, N. K. Pratas, G. Wunder, S. Saur, M. Navarro, D. Gregoratti, G. Vivier, E. De Carvalho, Y. Ji, Č. Stefanović *et al.*, "Towards massive connectivity support for scalable mMTC communications in 5G networks," *IEEE access*, vol. 6, pp. 28 969–28 992, 2018.

[10] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[11] U. M. Malik, M. A. Javed, S. Zeadally, and S. u. Islam, "Energy-efficient fog computing for 6G-enabled massive IoT: Recent trends and future opportunities," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 572–14 594, 2022.

[12] K. Sheth, K. Patel, H. Shah, S. Tanwar, R. Gupta, and N. Kumar, "A taxonomy of AI techniques for 6G communication networks," *Computer communications*, vol. 161, pp. 279–303, 2020.

[13] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11 891–11 915, 2021.

[14] M. Adhikari and A. Hazra, "6G-enabled ultra-reliable low-latency communication in edge networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 67–74, 2022.

[15] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, 2021.

[16] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, pp. 1–74, 2021.

[17] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[18] G. Zhu, Z. Lyu, X. Jiao, P. Liu, M. Chen, J. Xu, S. Cui, and P. Zhang, "Pushing AI to wireless network edge: an overview on integrated sensing, communication, and computation towards 6G," *Science China Information Sciences*, vol. 66, no. 3, p. 130301, 2023.

[19] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, I. Chih-Lin, and H. V. Poor, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 185–191, 2017.

[20] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-Lin, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, 2015.

[21] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1805–1838, 2020.

[22] X. Yi and C. Zhong, "Deep learning for joint channel estimation and signal detection in OFDM systems," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2780–2784, 2020.

[23] W. Zhang, X.-G. Xia, and P.-C. Ching, "Optimal training and pilot pattern design for OFDM systems in Rayleigh fading," *IEEE Transactions on Broadcasting*, vol. 52, no. 4, pp. 505–514, 2006.

[24] L. Bariah, A. Al-Dweik, and S. Muhaidat, "Blind channel estimation technique for OFDM systems over time varying channels," *IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1–7, 2018.

[25] K. Liu, J. P. C. Da Costa, H.-C. So, and A. L. De Almeida, "Semi-blind receivers for joint symbol and channel estimation in space-time-frequency MIMO-OFDM systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5444–5457, 2013.

[26] J. Kim, H. Ro, and H. Park, "Deep learning-based detector for dual mode OFDM with index modulation," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1562–1566, 2021.

[27] P. Tang, M. Steinbach, and V. Kumar, "Cluster analysis: Basic concepts and algorithms. introduction to data mining," 2005.

[28] E. Patel and D. S. Kushwaha, "Clustering cloud workloads: K-means vs Gaussian mixture model," *Procedia Computer Science*, vol. 171, pp. 158–167, 2020.

[29] A. Salari, M. Shirvanimoghaddam, M. B. Shahab, R. Arablouei, and S. Johnson, "Clustering-based joint channel estimation and signal detection for grant-free NOMA," in *2020 IEEE Globecom Workshops*, 2020.

[30] A. Salari, M. Shirvanimoghaddam, M. B. Shahab, Y. Li, and S. Johnson, "NOMA joint channel estimation and signal detection using rotational invariant codes and GMM-based clustering," *IEEE Communications Letters*, vol. 26, no. 10, pp. 2485–2489, 2022.

[31] A. Salari, M. Shirvanimoghaddam, M. B. Shahab, R. Arablouei, and S. Johnson, "Design and analysis of clustering-based joint channel estimation and signal detection for NOMA," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[32] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1117–1174, 2022.

[33] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[34] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[35] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Transactions on Wireless Communications*, 2022.

[36] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless IoT networks with optimized communication and resources," *IEEE Internet of Things Journal*, 2022.

[37] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[38] A. Salari, S. Johnson, B. Vucetic, and M. Shirvanimoghaddam, "Rate-convergence tradeoff of federated learning over wireless channels," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[39] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Non-orthogonal multiple access for 5G and beyond," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2347–2381, 2017.

[40] Z. Liu and L.-L. Yang, "Sparse or dense: A comparative study of code-domain NOMA systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 4768–4780, 2021.

[41] F. Ghanami, G. A. Hodtani, B. Vucetic, and M. Shirvanimoghaddam, "Performance analysis and optimization of NOMA with HARQ for short packet communications in massive IoT," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4736–4748, 2020.

[42] R. Kotaba, C. N. Manchón, T. Balercia, and P. Popovski, "How URLLC can benefit from NOMA-based retransmissions," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1684–1699, 2020.

[43] A. Maatouk, M. Assaad, and A. Ephremides, "Minimizing the age of information: NOMA or OMA?" *IEEE Conference on Computer Communications Workshops (INFOCOM)*, pp. 102–108, 2019.

[44] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive non-orthogonal multiple access for cellular IoT: Potentials and limitations," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 55–61, 2017.

[45] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Classification of indoor environments for IoT applications: A machine learning approach," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 12, pp. 2164–2168, 2018.

[46] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] Y. Liu, S. Zhang, X. Mu, Z. Ding, R. Schober, N. Al-Dhahir, E. Hossain, and X. Shen, "Evolution of NOMA toward next generation multiple access (NGMA) for 6G," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 4, pp. 1037–1071, 2022.

[49] I. Budhiraja, N. Kumar, S. Tyagi, S. Tanwar, Z. Han, M. J. Piran, and D. Y. Suh, "A systematic review on NOMA variants for 5G and beyond," *IEEE Access*, vol. 9, pp. 85 573–85 644, 2021.

[50] N. P. Le and K. N. Le, "Uplink NOMA short-packet communications with residual hardware impairments and channel estimation errors," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4057–4072, 2022.

[51] M. Aldababsa, M. Toka, S. Gökçeli, G. K. Kurt, and O. Kucur, "A tutorial on non-orthogonal multiple access for 5G and beyond," *Wireless communications and mobile computing*, vol. 2018, 2018.

[52] Z. Ma and J. Bao, "Sparse code multiple access (SCMA)," *Multiple Access Techniques for 5G Wireless Networks and Beyond*, pp. 369–416, 2019.

[53] E. Catak, F. Tekce, O. Dizdar, and L. Durak-Ata, "Multi-user shared access in massive machine-type communication systems via superimposed waveforms," *Physical Communication*, vol. 37, p. 100896, 2019.

[54] Z. Wu, K. Lu, C. Jiang, and X. Shao, "Comprehensive study and comparison on 5G NOMA schemes," *IEEE Access*, vol. 6, pp. 18 511–18 519, 2018.

[55] M. Vaezi, Z. Ding, and H. V. Poor, *Multiple access techniques for 5G wireless networks and beyond.* Springer, 2019, vol. 159.

[56] H. Duan, Y. Zhang, and J. Song, "Optimized resource allocation in scalable video broadcasting using LDM and BDM," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, 2020.

[57] X. Dai, Z. Zhang, B. Bai, S. Chen, and S. Sun, "Pattern division multiple access: A new multiple access technology for 5G," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 54–60, 2018.

[58] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017.

[59] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive non-orthogonal multiple access for cellular IoT: Potentials and limitations," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 55–61, 2017.

[60] S. Ghosh, A. Al-Dweik, and M.-S. Alouini, "On the performance of end-to-end cooperative NOMA-based IoT networks with wireless energy harvesting," *IEEE Internet of Things Journal*, 2023.

[61] Q. Gao, M. Jia, Q. Guo, X. Gu, and L. Hanzo, "Jointly optimized beamforming and power allocation for full-duplex cell-free NOMA in space-ground integrated networks," *IEEE Transactions on Communications*, 2023.

[62] N. K. Breesam, W. A. Al-Hussaibi, F. H. Ali, and I. M. Al-Musawi, "Efficient resource allocation for wireless-powered MIMO-NOMA communications," *IEEE Access*, vol. 10, pp. 130 302–130 313, 2022.

[63] M. B. Shahab, S. J. Johnson, M. Shirvanimoghaddam, and M. Dohler, "Receiver design for uplink power domain NOMA with discontinuous transmissions," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2738–2742, 2021.

[64] W. Yu, C. H. Foh, A. U. Quddus, Y. Liu, and R. Tafazolli, "Throughput analysis and user barring design for uplink NOMA-enabled random access," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6298–6314, 2021.

[65] A. Kumar, "ICA-ResNet-based signal detection and optimal power allocation for CRN-NOMA system," *International Conference on Mathematical and Statistical Physics, Computational Science, Education, and Communication (ICMSCE)*, vol. 12616, pp. 56–74, 2023.

[66] S. Hu, H. Huang, G. Gui, and H. Sari, "An analysis of the power imbalance on the uplink of power-domain NOMA," *IEEE Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–5, 2022.

[67] R. Abbas, T. Huang, B. Shahab, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Grant-free non-orthogonal multiple access: A key enabler for 6G-IoT," *arXiv preprint arXiv:2003.10257*, 2020.

[68] L.-H. Shen, K.-T. Feng, and L. Hanzo, “Five facets of 6G: Research challenges and opportunities,” *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–39, 2023.

[69] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, “Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO,” *IEEE transactions on signal processing*, vol. 68, pp. 764–779, 2020.

[70] Y. Ma, Z. Yuan, W. Li, and Z. Li, “PDRS: A fast non-iterative scheme for massive grant-free access in massive MIMO,” *IEEE Wireless Communications Letters*, vol. 10, no. 4, pp. 760–764, 2020.

[71] Z. Zhang, Y. Liang, and Y. Li, “Exploiting classifier diversity for efficient grant-free random access,” *IEEE International Conference on Communications*, pp. 4025–4030, 2022.

[72] C. Zhengy, F.-C. Zhengy, J. Luoy, X. Xiongy, and D. Feng, “Uplink performance analysis of grant-free NOMA networks,” *IEEE Vehicular Technology Conference:(VTC2022-Spring)*, pp. 1–5, 2022.

[73] Z. Shi, W. Gao, J. Liu, N. Kato, and Y. Zhang, “Distributed Q-learning-assisted grant-free NORA for massive machine-type communications,” *IEEE Global Communications Conference*, pp. 1–5, 2020.

[74] C. Zhang, Y. Liu, and Z. Ding, “Semi-grant-free NOMA: A stochastic geometry model,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1197–1213, 2021.

[75] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, “Deep reinforcement learning for throughput improvement of the uplink grant-free NOMA system,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6369–6379, 2020.

[76] T. Peken, G. Vanhoy, and T. Bose, “Blind channel estimation for massive MIMO,” *Analog Integrated Circuits and Signal Processing*, vol. 91, no. 2, pp. 257–266, 2017.

[77] A. Hyvarinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[78] E. Nayebi and B. D. Rao, “Semi-blind channel estimation for multiuser massive MIMO systems,” *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 540–553, 2017.

[79] X. Yuan, C. Fan, and Y. J. Zhang, “Fundamental limits of training-based uplink multiuser MIMO systems,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7544–7558, 2018.

[80] D. Ap, “Maximum likelihood from incomplete data via EM algorithm,” *J. Royal Stat. Soc. B.*, vol. 39, no. 1, pp. 1–38, 1977.

[81] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning.* Springer, 2006, vol. 4, no. 4.

[82] C. H. Aldana, E. de Carvalho, and J. M. Cioffi, "Channel estimation for multicarrier multiple input single output systems using the EM algorithm," *IEEE Transactions on Signal Processing*, vol. 51, no. 12, pp. 3280–3292, 2003.

[83] S. Gulomjon, F. Yongqing, J. Sangirov, F. Ye, and A. Olmasov, "A performance analysis of optimized semi-blind channel estimation method in OFDM systems," *International Conference on Advanced Communication Technology (ICACT)*, pp. 907–912, 2017.

[84] M. Framingham, "Growth in connected IoT device," *International Data Corporation (IDC)*, 2019.

[85] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1117–1174, 2022.

[86] S. M. Aldossari and K.-C. Chen, "Machine learning for wireless communication channel modeling: An overview," *Wireless Personal Communications*, vol. 106, pp. 41–70, 2019.

[87] X.-L. Huang, X. Ma, and F. Hu, "Machine learning and intelligent communications," *Mobile Networks and Applications*, vol. 23, pp. 68–70, 2018.

[88] J. Chen, Y. Gao, Y. Zhou, Z. Liu, D. p. Li, and M. Zhang, "Machine learning enabled wireless communication network system," *International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1285–1289, 2022.

[89] I. Ahmad, S. Shahabuddin, H. Malik, E. Harjula, T. Leppänen, L. Lovén, A. Anttonen, A. H. Sodhro, M. Mahtab Alam, M. Juntti, A. Ylä-Jääski, T. Sauter, A. Gurtov, M. Ylianttila, and J. Riekki, "Machine learning meets communication networks: Current trends and future challenges," *IEEE Access*, vol. 8, pp. 223 418–223 460, 2020.

[90] F. Liao, S. Wei, and S. Zou, "Deep learning methods in communication systems: A review," *Journal of Physics: Conference Series*, vol. 1617, no. 1, p. 012024, aug 2020. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1617/1/012024

[91] B. Ozpoyraz, A. T. Dogukan, Y. Gevez, U. Altun, and E. Basar, "Deep learning-aided 6G wireless networks: A comprehensive survey of revolutionary PHY architectures," *arXiv preprint arXiv:2201.03866*, 2022.

[92] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2019.

[93] M. Siddiqui, K. Khurshid, I. Rashid, and A. A. Khan, "Artificial intelligence based 6G intelligent IoT: Unfolding an analytical concept for future hybrid communication systems," *WCSE*, pp. 122–129, 2020.

[94] V. P. Rekkas, S. Sotiroudis, P. Sarigiannidis, S. Wan, G. K. Karagiannidis, and S. K. Goudos, "Machine learning in beyond 5G/6G networks—state-of-the-art and future trends," *Electronics*, vol. 10, no. 22, p. 2786, 2021.

[95] Z. Zhang, Q. Guo, Y. Li, M. Jin, and C. Huang, "Variational bayesian inference clustering based joint user activity and data detection for grant-free random access in mMTC," *IEEE Internet of Things Journal*, 2023.

[96] K.-H. Lee and B. C. Jung, "A novel expectation-maximization-based blind receiver for low-complexity uplink STLC-NOMA systems," *Sensors*, vol. 22, no. 20, p. 8054, 2022.

[97] K.-H. Lee, J. S. Yeom, J. Joung, and B. C. Jung, "Performance analysis of uplink NOMA with constellation-rotated STLC for IoT networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 705–717, 2022.

[98] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the K-means clustering algorithm," *Expert systems with applications*, vol. 40, no. 1, pp. 200–210, 2013.

[99] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." *KDD*, vol. 96, no. 34, pp. 226–231, 1996.

[100] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[101] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[102] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[103] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using Gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 29–37, 2009.

[104] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities," *Computer Networks*, vol. 183, p. 107556, 2020.

[105] H. V. Singh, A. Girdhar, and S. Dahiya, “A literature survey based on DBSCAN algorithms,” *International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 751–758, 2022.

[106] E. Schubert and M. Gertz, “Improving the cluster structure extracted from OPTICS plots.” *LWDA*, pp. 318–329, 2018.

[107] K. Virupakshappa and E. Oruklu, “Unsupervised machine learning for ultrasonic flaw detection using Gaussian mixture modeling, K-means clustering and mean shift clustering,” *IEEE International Ultrasonics Symposium (IUS)*, pp. 647–649, 2019.

[108] J. Wang and J. Jiang, “Unsupervised deep clustering via adaptive GMM modeling and optimization,” *Neurocomputing*, vol. 433, pp. 199–211, 2021.

[109] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[110] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, 1977.

[111] S. Niknam, H. S. Dhillon, and J. H. Reed, “Federated learning for wireless communications: Motivation, opportunities, and challenges,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[112] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtepe, H. Kim, and S. Nepal, “End-to-end evaluation of federated learning and split learning for internet of things,” *arXiv preprint arXiv:2003.13376*, 2020.

[113] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[114] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, “Cost-effective federated learning design,” *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2021.

[115] Z. Zhao, Y. Mao, Y. Liu, L. Song, Y. Ouyang, X. Chen, and W. Ding, “Towards efficient communications in federated learning: A contemporary survey,” *Journal of the Franklin Institute*, 2023.

[116] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-IID data,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[117] Z. Zhao, C. Feng, W. Hong, J. Jiang, C. Jia, T. Q. Quek, and M. Peng, "Federated learning with non-IID data in wireless networks," *IEEE Transactions on Wireless Communications*, 2021.

[118] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3394–3409, 2020.

[119] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[120] Z. Ji and Z. Qin, "Federated learning for distributed energy-efficient resource allocation," *IEEE International Conference on Communications*, pp. 1–6, 2022.

[121] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[122] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[123] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[124] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.

[125] P. Baldi, "Gradient descent learning algorithm overview: A general dynamical systems perspective," *IEEE Transactions on neural networks*, vol. 6, no. 1, pp. 182–195, 1995.

[126] W. A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique," *Signal processing*, vol. 6, no. 2, pp. 113–133, 1984.

[127] V. Plagianakos, G. Magoulas, and M. Vrahatis, "Learning rate adaptation in stochastic gradient descent," *Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873–1950)*, pp. 433–444, 2001.

[128] B. Yu, Y. Cai, and D. Wu, "Joint access control and resource allocation for short-packet-based mMTC in status update systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 851–865, 2020.

[129] S. Lv, X. Xu, S. Han, X. Tao, and P. Zhang, "Energy-efficient secure short-packet transmission in NOMA-assisted mMTC networks with relaying," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1699–1712, 2021.

[130] K. F. Hasan, C. Wang, Y. Feng, and Y.-C. Tian, "Time synchronization in vehicular ad-hoc networks: A survey on theory and practice," *Vehicular communications*, vol. 14, pp. 39–51, 2018.

[131] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 155–176, 1996.

[132] Y. Some and P. Kam, "Bit-error probability of QPSK with noisy phase reference," *IEE Proceedings-Communications*, vol. 142, no. 5, pp. 292–296, 1995.

[133] X. Wang, F. Labeau, and L. Mei, "Closed-form BER expressions of QPSK constellation for uplink non-orthogonal multiple access," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2242–2245, 2017.

[134] D. Hsu and S. M. Kakade, "Learning mixtures of spherical Gaussians: moment methods and spectral decompositions," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013.

[135] A. Emir, F. Kara, H. Kaya, and H. Yanikomeroglu, "Deepmud: Multi-user detection for uplink grant-free NOMA IoT networks via deep learning," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1133–1137, 2021.

[136] M. Dong, L. Tong, and B. M. Sadler, "Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1403–1418, 2004.

[137] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, "Short block-length codes for ultra-reliable low latency communications," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, 2019.

[138] C. R. Murthy, A. K. Jagannatham, and B. D. Rao, "Training-based and semiblind channel estimation for MIMO systems with maximum ratio transmission," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2546–2558, 2006.

[139] H. Semira, F. Kara, H. Kaya, and H. Yanikomeroglu, "Multi-user joint maximum-likelihood detection in uplink NOMA-IoT networks: Removing the error floor," *IEEE Wireless Communications Letters*, vol. 10, no. 11, pp. 2459–2463, 2021.

[140] ——, "Error performance analysis of multiuser detection in uplink-NOMA with adaptive M-QAM," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1654–1658, 2022.

[141] J. S. Yeom, H. S. Jang, K. S. Ko, and B. C. Jung, "BER Performance of Uplink NOMA With Joint Maximum-Likelihood Detector," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 295–10 300, 2019.

[142] J. A. Ansere, G. Han, and H. Wang, "A novel reliable adaptive beacon time synchronization algorithm for large-scale vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 565–11 576, 2019.

[143] P. Bholowalia and A. Kumar, "EBK-means: A clustering technique based on elbow method and K-means in WSN," *International Journal of Computer Applications*, vol. 105, no. 9, 2014.

[144] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending K-means with efficient estimation of the number of clusters." in *Icml*, vol. 1, 2000.

[145] U. D. Gupta, V. Menon, and U. Babbar, "Detecting the number of clusters during expectation-maximization clustering using information criterion," in *2010 Second International Conference on Machine Learning and Computing.* IEEE, 2010.

[146] A. A. Neath and J. E. Cavanaugh, "The Bayesian information criterion: background, derivation, and applications," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 2, pp. 199–203, 2012.

[147] H. Bozdogan, "Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[148] A. Adel Aly, H. M. ELAttar, H. ElBadawy, and W. Abbas, "Aggregated throughput prediction for collated massive machine-type communications in 5G wireless networks," *Sensors*, vol. 19, no. 17, p. 3651, 2019.

[149] S. S. Pietrobon, G. Ungerboeck, L. C. Pérez, and D. Costello, "Rotationally invariant nonlinear trellis codes for two-dimensional modulation," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1773–1791, 1994.

[150] G. Ungerboeck and S. S. Pietrobon, "Codes for QPSK modulation with invariance under 90 degrees rotation," in *Jet Propulsion Lab., Proceedings of the Mobile Satellite Conference*, 1988.

[151] R. W. Butler, *Saddlepoint approximations with applications.* Cambridge University Press, 2007, vol. 22.

[152] A. Stuart and K. Ord, "Kendalls advanced theory of statistics Arnold," 1994.

[153] R. C. Elandt-Johnson and N. L. Johnson, *Survival models and data analysis.* John Wiley & Sons, 1980, vol. 110.

[154] D. N. Ayyala, "High-dimensional statistical inference: Theoretical development to data analytics," *Elsevier Handbook of Statistics*, vol. 43, pp. 289–335, 2020.

[155] M. Shirvanimoghaddam, A. Salari, Y. Gao, and A. Guha, "Federated learning with erroneous communication links," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1293–1297, 2022.

[156] L. Nguyen, P. H. Nguyen, M. Dijk, P. Richtárik, K. Scheinberg, and M. Takác, "SGD and Hogwild! convergence without the bounded gradients assumption," *PMLR International Conference on Machine Learning*, vol. 80, pp. 3750–3758, 2018.

[157] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[158] Ericsson. (2017) Ericsson mobility report. [Online]. Available: https://www.ericsson.com/4ac635/assets/local/reports-papers/mobility-report/documents/2017/emr-november-2017-massive-iot-coverage-in-the-city.pdf

[159] X. XU, S. LIU, and G. YU, "Adaptive retransmission design for wireless federated edge learning," *ZTE Communications*, vol. 21, no. 1, pp. 3–14, 2023.

[160] F. Foukalas, "Federated-learning-driven radio access networks," *IEEE Wireless Communications*, vol. 29, no. 4, pp. 48–54, 2022.

[161] H. Hellström, V. Fodor, and C. Fischione, "Federated learning over-the-air by retransmissions," *IEEE Transactions on Wireless Communications*, 2023.

[162] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, "Squeezing deep learning into mobile and embedded devices," *IEEE Pervasive Computing*, vol. 16, no. 3, pp. 82–88, 2017.

[163] V. Prutyanov, N. Melentev, D. Lopatkin, A. Menshchikov, and A. Somov, "Developing IoT devices empowered by artificial intelligence: Experimental study," *IEEE Global IoT Summit (GIoTS)*, pp. 1–6, 2019.

[164] *Electron Datasheet*, Particle, March 2022. [Online]. Available: https://docs.particle.io/reference/datasheets/e-series/electron-datasheet/

[165] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, "System optimization of federated learning networks with a constrained latency," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 1095–1100, 2021.

[166] J. Proakis and M. Salehi, *Digital communications.* 5th Edition McGraw-Hill, New York, 2008.

[167] W. C. Lee, *Mobile communications engineering: theory and applications.* McGraw-Hill Education, 1998.

[168] H.-C. Yang and M.-S. Alouini, *Order statistics in wireless communications: diversity, adaptation, and scheduling in MIMO and OFDM systems.* Cambridge University Press, 2011.

[169] R. Vershynin, *High-dimensional probability: An introduction with applications in data science.* Cambridge university press, 2018.

[170] R. Zhao, Y. Li, Y. Sun *et al.*, “Statistical convergence of the EM algorithm on Gaussian mixture models,” *Electronic Journal of Statistics*, vol. 14, no. 1, pp. 632–660, 2020.

[171] B. Yan, M. Yin, and P. Sarkar, “Convergence of gradient EM on multi-component mixture of Gaussians,” in *Advances in Neural Information Processing Systems*, 2017.

[172] M. P. Friedlander and M. Schmidt, “Hybrid deterministic-stochastic methods for data fitting,” *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1380–A1405, 2012.

[173] J. Curtiss, “On the distribution of the quotient of two chance variables,” *The Annals of Mathematical Statistics*, vol. 12, no. 4, pp. 409–421, 1941.

[174] S. Schiessl, J. Gross, and H. Al-Zubaidy, “Delay analysis for wireless fading channels with finite blocklength channel coding,” *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 13–22, 2015.