A Learning Framework for Optimized Control of Wireless Links

by

Mostafa HUSSIEN

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, OCTOBER 16, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Mohamed Cheriet, Thesis Supervisor
Department of Systems Engineering, École de technologie supérieure

Mr. Kim Khoa Nguyen, Co-supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr. Abdelouahed Gherbi, President of the Board of Examiners
Department of Software Engineering and IT, École de technologie supérieure

Mr. Eric Granger, Member of the jury
Department of Systems Engineering, École de technologie supérieure

Mr. Walaa Hamouda, External Examiner
Department of Electrical and Computer Engineering, Concordia University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON SEPTEMBER 01, 2023

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This dissertation is submitted for the degree of Doctor of Philosophy at the École de Technologie Supérieure (ÉTS), University of Quebec. The research described herein was conducted under the supervision of Professor **Mohamed Cheriet** in the Department of Systems Engineering and Professor **Kim Khoa Nguyen** in the Department of Electrical Engineering, between Fall 2018 and Summer 2023.

The present dissertation is structured as a compilation of papers published at or submitted to prestigious top-rank journals in the field of wireless communications and artificial intelligence. The papers included in this dissertation are integrated with high fidelity to ensure compliance with the proposed and published articles' structure and shape. Still, only peripheral modifications (e.g., figures framing, repositioning, and rescaling) were made under École de technologie supérieure's thesis guidelines.

ACKNOWLEDGEMENTS

I sincerely thank my esteemed supervisors, Prof. Mohamed Cheriet and Prof. Kim Khoa Nguyen, for their endless guidance and support throughout my Ph.D. journey. I am eternally grateful for their dedication to my academic and personal growth and their patience and encouragement during the most challenging times.

I would also like to acknowledge the invaluable contributions of École de Technologie Supérieure, Synchromedia laboratory, Resilient Machine Learning Institute (ReMI), Samsung AI Center (SAIC-Montreal), and Nokia Bell-Labs, Stuttgart, Germany. Their resources, facilities, and collaborative environments have played an integral role in the success of my Ph.D. journey.

I would like to extend my heartfelt gratitude to my loving family – my mother, sister, brothers, and their families – for their support, encouragement, and belief in me. Their prays, love and sacrifices have been the bedrock of my strength and perseverance.

I am immensely grateful to Marwa, my wonderful wife, and my two kids, Yahia and Youssef, for their patience, understanding, and enduring support throughout my Ph.D. The joy they bring into my life has been indispensable in overcoming challenges and achieving this milestone. Thanks are extended to my nonblood brothers M. Agamy, Weam, and their kids for being in my life. I am truly fortunate to have them by my side.

I would also like to express my sincere appreciation to my close friends Afifi, Abdelmoaty, Amr, M. Othman, M. Omar, A. Khalaf, Al-Emary, and O. Salem for their unwavering support and encouragement. Their friendship and timely advice have been invaluable in navigating the various challenges and celebrating the milestones along this journey.

Un cadre d’apprentissage pour un contrôle optimisé des liaisons sans fil

Mostafa HUSSIEN

RÉSUMÉ

La prolifération des systèmes de communication sans fil a suscité une attention considérable, en raison de la croissance exponentielle des nœuds interconnectés et de l’émergence d’applications ayant des exigences diverses en matière de qualité de service (QoS). Ces cas d’utilisation pris en charge présentent un large éventail d’exigences exigeantes en matière de qualité de service. En particulier, les systèmes de communication de cinquième génération (5G) ont été conçus pour prendre en charge simultanément trois cas d’utilisation différents: les communications ultra-fiables à faible latence (URLLC), les communications massives de type machine (mMTC) et le haut débit mobile amélioré (eMBB). chacun avec ses exigences spécifiques à l’application, tous fonctionnant au sein de ressources réseau partagées. Satisfaire ces exigences nécessite d’exploiter le réseau de manière optimisée, un formidable défi compte tenu du dynamisme inhérent aux conditions des canaux.

L’optimisation des paramètres de transmission (par exemple, schémas de modulation et de codage, intervalles de garde, etc.) est au cœur de l’amélioration de l’efficacité opérationnelle des systèmes de communication modernes. Cependant, l’adaptation dynamique des paramètres de transmission dépend de la connaissance du canal, acquise via un processus de rétroaction. Ce mécanisme de rétroaction est sensible à l’estimation et à la compensation des décalages de fréquence porteuse. Par conséquent, cette thèse se concentre sur trois composants essentiels au sein du pipeline de communication : l’estimation du décalage de fréquence porteuse (CFO), la compression du retour d’information sur l’état du canal (CSI) et l’adaptation de la liaison.

Les complexités intrinsèques associées à la modélisation analytique de ces défis, associées à l’accessibilité d’ensembles de données abondants et à l’extraordinaire efficacité démontrée par les algorithmes d’intelligence artificielle (IA) et d’apprentissage automatique (ML), ont catalysé l’intégration des méthodologies d’IA et de ML pour résoudre ces problèmes. De plus, l’intégration des techniques d’IA et de ML promet de réduire considérablement les temps d’exécution en contournant les algorithmes itératifs conventionnels traditionnellement utilisés dans de tels projets.

Cette thèse propose plusieurs nouvelles solutions basées sur le ML pour les trois problèmes difficiles susmentionnés dans un cadre complet. Les solutions proposées améliorent la précision, la fiabilité et l’efficacité des systèmes de communication. Cette thèse est structurée en trois parties, chacune dédiée à l’examen de l’un des trois problèmes fondamentaux étudiés. Plus précisément, la PARTIE 1 se concentre sur la résolution du problème de l’*estimation CFO*, la PARTIE 2 se penche sur le problème de la *compression de rétroaction CSI* et la PARTIE 3 est consacrée au problème de l’*Adaptation des liens*.

Nos contributions à la PARTIE 1 (l’estimation du CFO) peuvent être résumées comme suit :

- Présentation de l'apprentissage d'ensemble, en particulier de l'algorithme Gradient Boosting Machine (GBM), dans le problème d'estimation CFO. L'adoption du GBM améliore les capacités de généralisation de notre solution proposée tout en l'alignant sur les contraintes de ressources inhérentes rencontrées au niveau de l'équipement utilisateur (UE).
- Présentation de la technique BiModule CFO Estimation (BMCE), qui représente une approche innovante pour combiner les prédictions générées par le module d'estimation avec les résultats dérivés d'un module auxiliaire conçu pour modéliser la corrélation temporelle entre les valeurs CFO. La combinaison de l'estimation directe du CFO dérivée des préambules de synchronisation avec les prévisions du CFO entraîne une amélioration notable de la précision des prévisions, produisant une amélioration de 16% par rapport au seul recours aux méthodes d'estimation directe.

Nos contributions à la PARTIE 2 (la compression de rétroaction CSI) peuvent être résumées comme suit:

- Présentation des auto-encodeurs variationnels (VAE) pour résoudre le problème des canaux de rétroaction bruyants. Il a été démontré empiriquement que les VAE surpassent les auto-encodeurs classiques à estimation ponctuelle en termes de précision de reconstruction, offrant ainsi un moyen plus efficace de gérer l'impact des canaux bruyants.
- Proposer une version personnalisée de la perte de VAE afin d'optimiser davantage les performances de la VAE dans le cadre de problèmes de feedback. Cette fonction de perte personnalisée est précisément conçue pour s'aligner sur les exigences et caractéristiques spécifiques des défis liés au feedback (canaux de feedback bruyants), contribuant ainsi à des reconstructions plus précises et plus significatives.
- Proposer une solution alternative ancrée dans la théorie de l'apprentissage pour répondre aux limites reconnues des approches basées sur l'auto-encodeur. Cette nouvelle approche répond non seulement aux limites des auto-encodeurs conventionnels, mais surpasse également les solutions traditionnelles et basées sur l'apprentissage par une marge substantielle en termes de précision de reconstruction.

Enfin, dans la PARTIE 3 (lien adaptation), nos contributions peuvent être résumées comme suit :

- Proposer une nouvelle modélisation du problème d'adaptation de lien sous la forme d'une classification multiclasse multiétiquette qui fournit un nouveau cadre pour aborder ce problème complexe.
- Proposer une fonction de perte personnalisée pour entraîner les modèles de classification tout en augmentant la fiabilité du système (en minimisant les erreurs faussement positives).
- Proposer un critère de sous-échantillonnage complet pour la formation des modèles d'adaptation de lien au lieu de l'échantillonnage aléatoire adopté dans la littérature. L'utilisation de ce nouveau critère de sous-échantillonnage pour former des modèles sur des ensembles de données limités entraîne des améliorations significatives, avec des améliorations de performances allant jusqu'à 50% observées dans des scénarios spécifiques.

- Développer une architecture neuronale sophistiquée pour résoudre le problème d'adaptation-compression articulaire. Cette architecture est accompagnée d'une fonction de perte et d'une procédure de formation personnalisées, représentant collectivement un cadre de solution complet pour gérer efficacement les complexités des tâches de compression et d'adaptation des articulations.

Mots-clés: Optimisation des canaux, intelligence artificielle, apprentissage automatique, auto-encodeurs variationnels, systèmes de communication de nouvelle génération, 5G, informations sur l'état des canaux, modulation et codage adaptatifs, compression CSI, estimation du décalage de fréquence porteuse, systèmes MIMO-FDD

A Learning Framework for Optimized Control of Wireless Links

Mostafa HUSSIEN

ABSTRACT

The proliferation of wireless communication systems has garnered significant attention, driven by the exponential growth of interconnected nodes and the emergence of applications with diverse Quality-of-Service (QoS) requirements. These supported use cases exhibit a wide spectrum of demanding QoS requirements. In particular, fifth-generation (5G) communication systems have been architected to concurrently support three different use cases: Ultra-Reliable Low-Latency Communications (URLLC), massive Machine-Type Communications (mMTC), and enhanced Mobile BroadBand (eMBB), each with its unique application-specific requirements, all operating within shared network resources. Satisfying these requirements mandates operating the network in an optimized fashion, a formidable challenge given the inherent dynamism of channel conditions.

At the heart of enhancing the operational efficiency of modern communication systems lies the optimization of transmission parameters (e.g., modulation and coding schemes, guard intervals, and more). However, the dynamic adaptation of transmission parameters is contingent upon channel knowledge, which is acquired through a feedback process. This feedback mechanism is sensitive to the estimation and compensation of carrier frequency offsets. Hence, this thesis focuses on three pivotal components within the communication pipeline: Carrier Frequency Offset (CFO) estimation, Channel State Information (CSI) feedback compression, and link adaptation.

The intrinsic complexities associated with analytically modeling these challenges, coupled with the accessibility of abundant datasets and the extraordinary efficacy demonstrated by artificial intelligence (AI) and machine learning (ML) algorithms, have catalyzed the integration of AI and ML methodologies in addressing these issues. Moreover, the incorporation of AI and ML techniques holds the promise of significantly reducing execution times by circumventing the conventional iterative algorithms traditionally employed in such endeavors.

This dissertation proposes several novel ML-based solutions for the aforementioned three challenging problems within a comprehensive framework. The proposed solutions improve the accuracy, reliability, and efficiency of communication systems. This dissertation is structured into three parts, each dedicated to addressing one of the three core problems investigated. Precisely, PART 1 is focused on tackling the *CFO estimation* problem, PART 2 delves into the *CSI feedback compression* problem, and PART 3 is devoted to the *Link Adaptation* problem.

Our contributions to PART 1 (the CFO estimation) can be summarized as follows:

- Introducing ensemble learning, specifically the Gradient Boosting Machine (GBM) algorithm, within the CFO estimation problem. Adopting the GBM enhances the generalization capabilities of our proposed solution while aligning it with the inherent resource constraints encountered at the User Equipment (UE).

- Presenting the BiModule CFO Estimation (BMCE) technique, which represents an innovative approach to combining the predictions generated by the estimation module with the outcomes derived from an auxiliary module designed to model the temporal correlation among CFO values. Combining the direct CFO estimation derived from synchronization preambles with CFO forecasting results in a notable enhancement of prediction accuracy, yielding a 16% improvement over sole reliance on direct estimation methods.

Our contributions to PART 2 (the CSI feedback compression) can be summarized as follows:

- Introducing the Variational Autoencoders (VAE) to tackle the problem of noisy feedback channels. It has been empirically demonstrated that VAEs outperform conventional point estimation autoencoders in terms of reconstruction accuracy, providing a more effective means of managing the impact of noisy channels.
- Proposing a customized version of the VAE loss in order to further optimize the performance of VAE in the context of feedback problems. This customized loss function is precisely designed to align with the specific requirements and characteristics of feedback-related challenges (noisy feedback channels), thereby contributing to more accurate and meaningful reconstructions.
- Proposing an alternative solution rooted in learning theory to address the recognized limitations of autoencoder-based approaches. This novel approach not only addresses the limitations of conventional autoencoders but also outperforms both traditional and learning-based solutions by a substantial margin in terms of reconstruction accuracy.

Finally, in PART 3 (link adaptation), our contributions can be summarized as follows:

- Proposing a novel modeling for the link adaptation problem as a multilabel multiclass classification that provides a new framework for tackling this intricate issue.
- Proposing a customized loss function to train the classification models while increasing the system reliability (by minimizing the false positive errors).
- Proposing a comprehensive subsampling criterion for training link adaptation models instead of the random sampling adopted in the literature. The utilization of this novel subsampling criterion for training models on limited datasets yields significant improvements, with performance enhancements of up to 50% observed in specific scenarios.
- Developing a sophisticated neural architecture for solving the joint compression-adaptation problem. This architecture is accompanied by a customized loss function and training procedure, collectively representing a comprehensive solution framework for efficiently managing the complexities of joint compression and adaptation tasks.

Keywords: Wireless link optimization, artificial intelligence, machine learning, variational autoencoders, next-generation communication systems, 5G, channel state information, adaptive modulation and coding, CSI compression, carrier frequency offset estimation, MIMO-FDD systems

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

LIST OF ABREVIATIONS

3GPP	3rd Generation Partnership Project
5G	Fifth Generation Networks
AI	Artificial Intelligence
AMC	Adaptive Modulation and Coding
ANN	Artificial Neural Networks
AuoML	Automated Machine Learning
AWGN	Additive White Gaussian Noise
B5G	Beyond the fifth generation
BER	Bit-error rate
BO	Bayesian optimization
BS	Base Station
CAGR	Compound Annual Growth Rate
CFO	Carrier Frequency Offset
CNN	Convolutional Neural Network
CP	Cyclic Prefix
CR	Compression Ratio
CRC	Cyclic Redundancy Check
CS	Compressive Sensing
CSI	Channel State Information

CSV	Comma-Separated Values
ELBO	Evidence Lower Bound Objective
FC	Fusion Center
FD	Frequency Domain
FDD	Frequency Division Duplexing
FFT	Fast Fourier Transform
FLOPS	Floating Point Operations
GBM	GBM Gradient Boosting Machines
GI	Guard Interval
ICI	Inter-carrier Interference
IoT	Internet of Things
LTE	Long-Term Evolution
M2M	Machine to Machine
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLP	Multilayer Perceptron
mm-Wave	Millimeter-wave
NLP	Natural Language Processing
NMSE	Normalized Mean Square Error

NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
PBCH	Physical Broadcast Channel (PBCH)
PCA	Principal Component Analysis
PSS	Primary Signal Synchronization
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RB	Resource Block
RBF	Radial Basis Function
RF	Radio Frequency
RL	Reinforcement Learning
SCS	Subcarrier Spacing
SNR	Signal-to-noise Ratio
SSB	Signal Synchronization Block
SSS	Secondary Signal Synchronization
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TDD	Time Division Duplexing
TM	Transmission Mode
UE	User Equipment

VAE	Variational Autoencoders
VQ	Vector Quantization
ZC	Zadoff-Chu

LISTE OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathbf{I}$	Identity matrix
I	An indicator function
T	Number of leaves
$\mathbf{H}$	Channel matrix
N	Number of data points in a dataset
x	An input sample in a larger dataset
y	An output sample from a larger dataset
$\mathbf{w}$	Weight vector
ε	Normalized carrier frequency offset
δ	Drift samples
P_μ	Primary synchronization signal SS
$\mathscr{D}$	A labeled dataset
$\hat{y}$	An approximation for an output sample, typically produced by a predictive model, $f(x)$
R_j	A region in the input space
$\mathscr{L}$	Loss function
Ω	A regularization term
γ	A hyperparameter
$\mathbb{E}$	Expectation
$\mathscr{R}$	Empirical risk of a predictive model

g_i	The first-order gradient statistics of an objective function
q_i	The second-order gradient statistics of an objective function
$\|\cdot\|$	Cardinality of a set
$\|\cdot\|_2$	Ecludian norm
$\|\cdot\|_F$	Forbenius norm
$[n]$	The set of integers from 1 to n, i.e., $\{1,2,\ldots,n\}$
db	Decibel
r(n)	Received signal of an OFDM system
x(n)	Transmitted signal
h(n)	Impulse response of the channel
w(n)	Additive white Gaussian noise
TM	Transmission mode
$KL(\cdot)$	Kullback-Leibler (KL)divergence function
$\mathscr{N}(\mu,\sigma)$	Gaussian distribution with μ mean and σ standard deviation
$diag(x)$	Diagonal matrix with the elements of the diagonal matrix given by a vector x
$\mathbf{vec}(H)$	Vectorization of H in a column-first format

INTRODUCTION

0.1 Context, Background, and Motivations

Wireless communication has revolutionized the way we live and work, providing us with a fast, convenient, and cost-effective way to connect and communicate. In particular, the advent of 5G technology has opened up a whole new world of possibilities, offering faster speeds, lower latency, and more reliable connections than ever before. 5G defines three different classes of applications, namely, Ultra-Reliable Low-Latency Communications (URLLC), enhanced Mobile Broadband (eMBB), and massive Machine-Type Communications (mMTC), Fig. 0.1. This has enabled the development of new and innovative applications and services that are changing the way we interact with the world around us. 5G has the potential to transform a wide range of industries, from healthcare to education, by enabling real-time communication and data transfer on a massive scale. For example, in healthcare, 5G can support telemedicine and remote surgery, allowing medical professionals to diagnose and treat patients from anywhere in the world. In addition, 5G is also set to drive economic growth and job creation by enabling new and innovative business models and services Lin (2022).

According to a report by *Ericsson* Cerwall & et al. (2020), there will be 1.9 billion 5G connections worldwide by the end of 2023, and this number is expected to grow to over 4 billion by the end of 2025. The global 5G market is expected to grow at a compound annual growth rate (CAGR) of over 60% between 2020 and 2025, according to a report by *MarketsandMarkets* Mar (2023). This is expected to result in a market size of over $500 billion by the end of 2025. The race to adopt, advance, and exploit the powers of 5G networks relies mainly on a limited budget of radio resources. In order to make the most of scarce radio resources, it is imperative to develop intelligent technologies and solutions for utilizing them efficiently. These solutions should take into account various factors from the high number of users, harsh quality of service (QoS) requirements, a vast array of services and applications, different propagation environ-

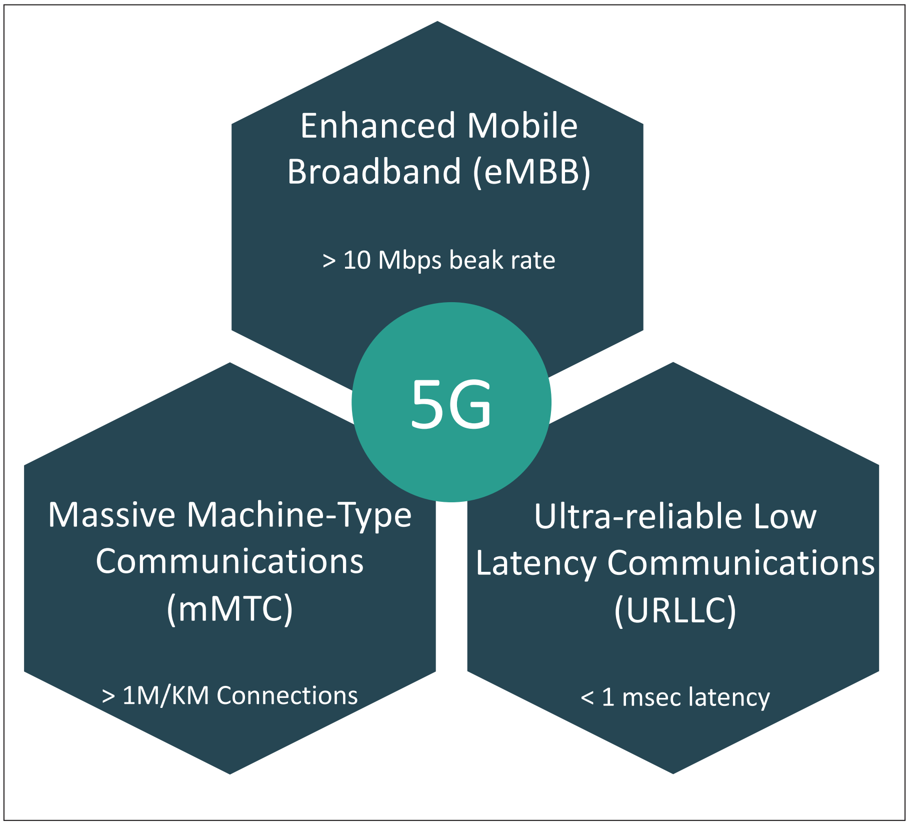

Enhanced Mobile Broadband (eMBB)
> 10 Mbps beak rate
5G
Massive Machine-Type Communications (mMTC)
> 1M/KM Connections
Ultra-reliable Low Latency Communications (URLLC)
< 1 msec latency

Figure 0.1 The application scenarios of 5G

ments, diverging user behaviors, etc. Moreover, some of these factors are stochastic and can not be precisely specified in advance. Accordingly optimizing the radio resource usage under all of the aforementioned factors and constraints becomes a very challenging task Chen *et al.* (2022).

In recent years, the field of artificial intelligence (AI) and machine learning (ML) has seen tremendous advances, leading to breakthroughs in various applications across different domains. With the increasing availability of large amounts of data and computing power, deep learning models have emerged as powerful tools for solving complex problems. In computer vision, deep learning algorithms have surpassed human performance on various benchmark datasets and are being used in a wide range of applications such as object recognition, image classification, and semantic segmentation LeCun *et al.* (2015). Similarly, in natural language processing (NLP), transformers, a type of deep learning model, have revolutionized the field by achieving state-of-the-art performance on various NLP tasks such as machine translation, sentiment analysis, and question-answering Lauriola *et al.* (2022).

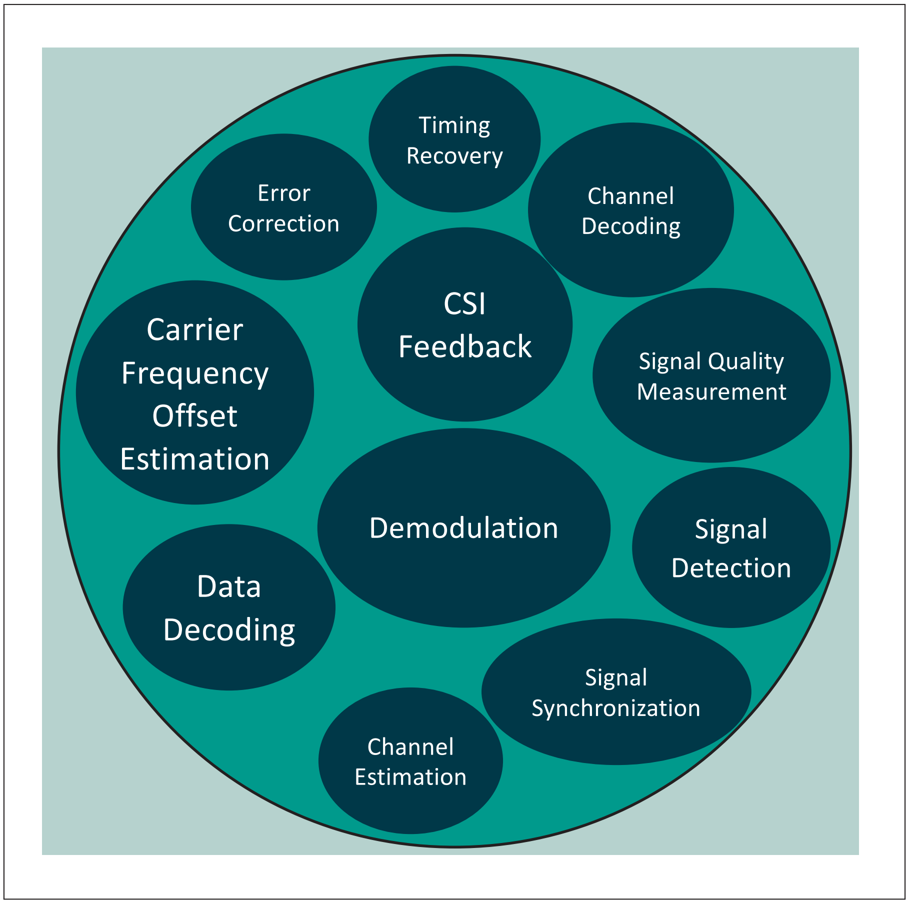


Figure 0.2 A non-exhaustive list of communication functions that can be optimized using AI and ML approaches

In wireless communications, AI and ML have the potential to play a significant role in advancing the physical layer Wang & et al. (2017). One area where AI and ML can have a big impact is in the optimization of wireless networks. With the increasing demand for high data rates and low latency communication, traditional approaches to network optimization are becoming insufficient. AI and ML, on the other hand, can be used to model and analyze the behavior of wireless networks, allowing for real-time optimization and increased network efficiency

Nassef & et al. (2022). Another application of AI and ML in wireless communications is in the design of radio frequency (RF) front-ends, which are critical components of wireless devices such as smartphones and laptops. By using AI and ML techniques, the design of RF front-ends can be automated, reducing the time and cost of development while also improving their performance. In addition, AI and ML can also be used to develop new modulation and coding schemes that can significantly improve the spectral efficiency of wireless networks, enabling the transmission of more data within the same bandwidth, Fig. 0.2 shows a non-exhaustive list of the functions that can benefit from AI and ML models.

In a typical wireless communication system, especially at the physical layer, a sequence of operations takes place between the transmitter and the receiver. These functions are distributed across the various layers of the user and control planes in the 5G protocol stack, see Fig. 0.3. Most of these operations can leverage the power of AI/ML algorithms to optimize their performance. For example, the following operations and processes occur upon the reception of an incoming transmission at the user equipment (UE) side:

- **Signal detection**: The UE detects the presence of the incoming signal using its receiver.
- **Signal synchronization**: The UE synchronizes with the incoming signal to determine its timing and frequency parameters.
- **Demodulation**: The UE demodulates the received signal to extract the data bits from the modulation scheme used for the transmission.
- **Channel decoding**: The UE decodes the channel coding applied to the incoming signal to recover the original data.
- **Error correction**: The UE applies error correction algorithms to correct any errors in the received data.

- **Channel estimation**: The UE estimates the channel parameters, such as the fading, delay spread, and multipath, to improve the performance of the receiver.

- **Carrier recovery**: The UE recovers the carrier signal from the received signal to obtain a stable reference for demodulation.

- **Timing recovery**: The UE recovers the timing information from the received signal to align its receiver with the incoming signal.

- **Data decoding**: The UE decodes the data bits from the received signal to obtain the original information.

- **Signal quality measurement**: The UE measures the quality of the received signal, such as its signal-to-noise ratio (SNR), to determine the reliability of the received data.

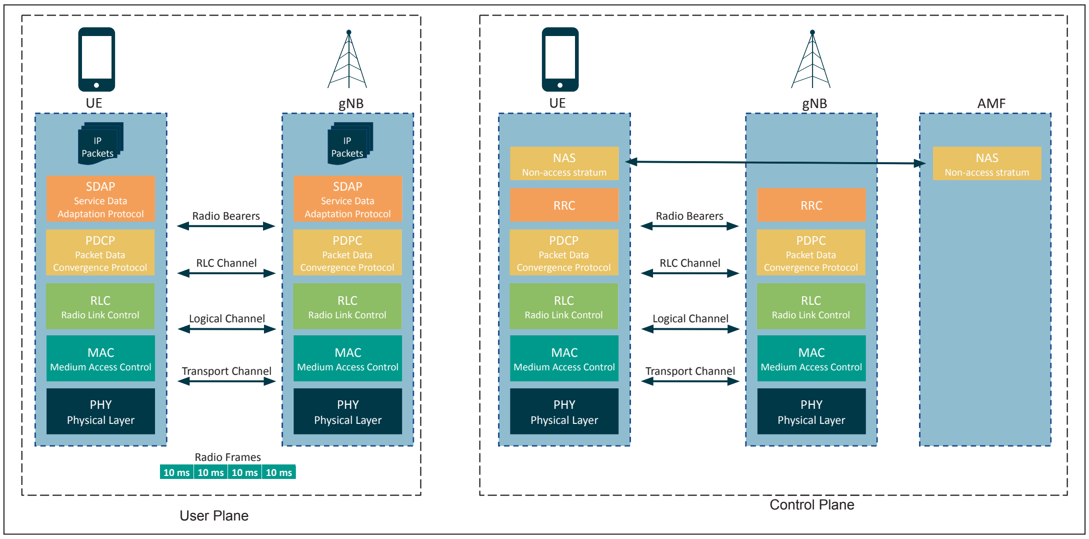


Figure 0.3 The protocol stack of the user plane (left) and control plane (right)

In this thesis, we consider three tasks in the signal pipeline between a UE and BS, see Fig. 0.4. Specifically, we adopt and extend various ML algorithms to optimize the carrier frequency offset (CFO) estimation, channel state information (CSI) feedback compression, and link adaptation. In the next paragraphs, we briefly describe each of these problems.

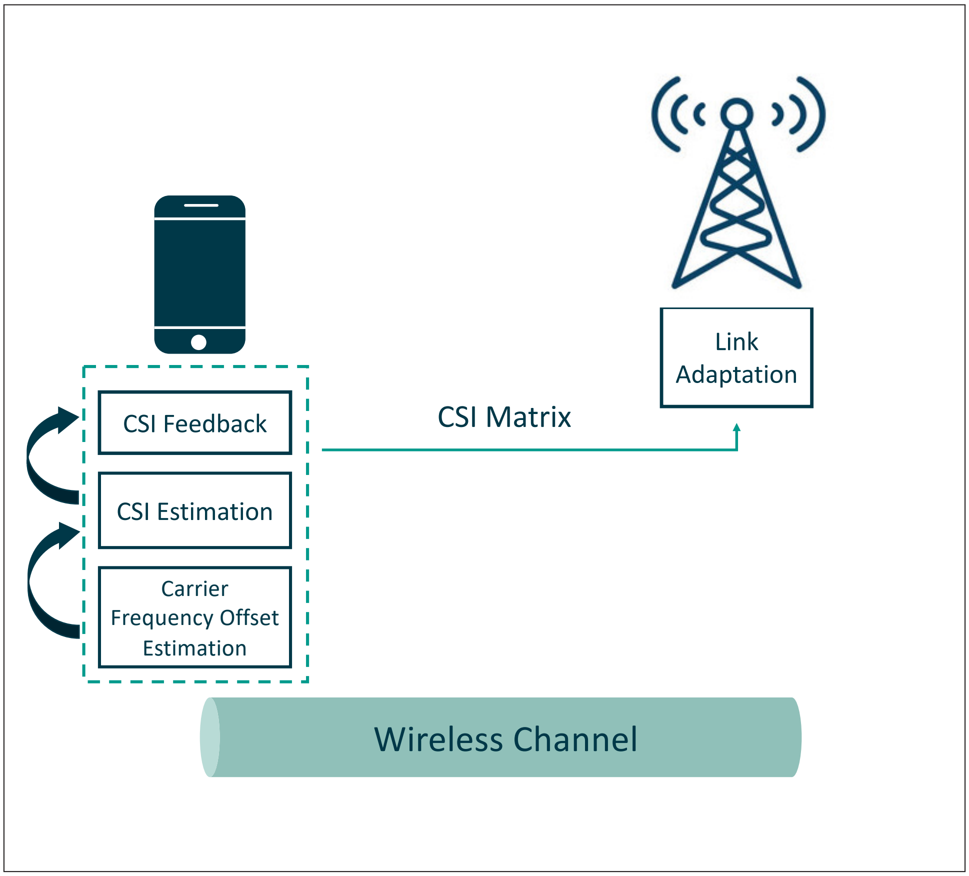


Figure 0.4 A Pipeline for the signal between the BS and the UE

CFO Estimation: CFO refers to the deviation of the carrier frequency of a transmitted signal from its intended frequency. This deviation can occur due to various factors such as errors in the local oscillator or non-idealities in the modulator. The existence of a CFO in a communication system yields a significant degradation in performance, especially in systems that rely on coherent demodulation. For example, in Orthogonal Frequency Division Multiplexing (OFDM) systems, CFO causes inter-carrier interference, reducing the system's overall bit error rate and reducing the signal-to-noise ratio. Additionally, CFO can also lead to incorrect timing and phase recovery, making it difficult to decode the transmitted data. Therefore, it is important to correct or compensate for CFO in order to maintain the desired performance of a communication system. This can be achieved through various methods, including frequency synchronization, which adjusts the carrier frequency to match the intended frequency, or CFO compensation algorithms that estimate and remove the effect of CFO during the demodulation process. We investigate the potential of estimating the CFO from the received primary signal synchronization (PSS) and secondary signal synchronization (SSS) blocks. We leverage the powers of gradient boosting machines (GBM) to regress the value of CFO, which can be used later for signal synchronization.

CSI feedback Compression: In Multiple Input Multiple Output-Frequency Division Duplexing (MIMO-FDD) systems, the uplink and downlink channels are frequency separated, meaning that the CSI at the receiver cannot be directly used to transmit data in the downlink. Therefore, the receiver must estimate the CSI and send it back to the transmitter in a feedback mechanism. This feedback process can cause several issues, including limited feedback bandwidth, quantization errors, and feedback delays, all of which can greatly affect the system's performance. Limited feedback bandwidth is a critical issue as the amount of CSI feedback is limited by the channel bandwidth, and more feedback is required as the number of antennas increases. This can cause a bottleneck in the feedback mechanism, and the lack of information can limit the system's ability to adapt to channel conditions, leading to suboptimal performance. Accordingly, compressing the CSI feedback before fusing it to the BS is essential to enable efficient and timely feedback. Neural compression has played a major role in this problem and achieved an outstanding performance compared with conventional methods such as vector quantization (VQ) and compressive sensing (CS). We propose various ML-based CSI compression and feedback algorithms which boost the accuracy and robustness of the feedback mechanism.

Link Adaptation: Wireless link adaptation is a technique for dynamically adjusting the transmission parameters of a wireless link to match the changing conditions of the wireless channel. The goal of link adaptation is to achieve the best possible performance by adjusting the transmission parameters, such as the modulation and coding scheme (MCS), the transmit power, guard interval, or the antenna configuration, based on the channel conditions. The dynamic nature of wireless channels can cause significant fluctuations in the channel conditions, which can result in varying levels of interference and fading. Link adaptation allows the system to dynamically adjust the transmission parameters to mitigate these effects and maintain a high level of performance. Link adaptation is achieved through the use of channel feedback, where the UE provides information about the channel conditions back to the BS through the CSI feedback mechanism discussed in the previous paragraph. Based on this information, the BS can adjust the transmission parameters to ensure that the data is transmitted with the highest possible quality. One of the main benefits of link adaptation is improved energy efficiency, as the transmitter can adjust its power levels to match the channel conditions, reducing power consumption and extending the battery life of wireless devices. Link adaptation also improves the robustness of the wireless link, as it allows the system to adapt to changes in channel conditions and maintain a high level of performance in challenging environments. The conventional methods are limited in dealing with the fast-growing application requirements and propagation scenarios. Moreover, with the increased number of antennas at the UE and BS sides, the dimensionality of the received CSI increases, and the adaptation mechanism based on the received information becomes more challenging. However, deep learning models are capable of efficiently processing high-dimensional data and extracting useful information about the current state of the channel in a fixed time. In this thesis, we investigate and validate the deep learning approach for link adaptation.

Next, we delve into the motivations behind optimized control, particularly concerning emerging communication systems, especially for 5G.

Motivations

Optimized control of wireless channels is crucial in both modern and legacy communication systems, but it becomes even more critical in modern systems (i.e.,5G) due to the following reasons:

- Higher Frequency Bands: 5G networks, for example, operate in higher frequency bands compared to LTE, such as millimeter waves (mmWave). These high-frequency bands offer wider bandwidths and higher data rates, but they are more susceptible to signal attenuation and obstacles. Optimized control is needed to mitigate these challenges, ensuring better signal quality and coverage.

- Massive MIMO Technology: 5G utilizes advanced antenna technologies like Massive Multiple-Input Multiple-Output (MIMO), which involves using a large number of antennas at both the base station and user devices. Properly managing these multiple antennas for beamforming and spatial multiplexing requires sophisticated channel control algorithms to maximize performance.

- Dynamic Spectrum Sharing: 5G introduces dynamic spectrum sharing capabilities, allowing multiple services to coexist in the same frequency band. This dynamic allocation requires efficient channel control to avoid interference and allocate resources optimally, considering the diverse types of services that 5G supports, including enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC), and massive machine-type communication (mMTC).

- Higher Data Rates and Throughput: 5G aims to provide significantly higher data rates and throughput compared to LTE. Achieving these ambitious goals necessitates precise control of the wireless channel to minimize signal degradation and maintain high-quality connections.

- Lower Latency: 5G targets ultra-low latency, which is critical for applications like virtual reality, augmented reality, autonomous vehicles, and industrial automation. Optimized channel control helps reduce transmission delays and ensures that latency requirements are met.

- Diverse Use Cases: 5G is designed to support a wide range of use cases beyond traditional mobile communications, including IoT applications, industrial automation, smart cities, and healthcare. Each use case has unique requirements, and optimized channel control is essential to deliver the desired performance for different services.
- Energy Efficiency: With the increasing focus on sustainability, next-generation communication systems aim to be more energy-efficient. By optimizing the wireless channels, these systems can reduce unnecessary signaling and transmission power, leading to energy savings.

In summary, while optimized channel control is essential for both 5G and legacy networks, the higher frequency bands, advanced technologies, dynamic spectrum sharing, and diverse use cases in 5G make it even more critical to ensure optimal performance and meet the demanding requirements of next-generation wireless communications.

0.2 Problem Statement and Challenges

Wireless communication systems are subject to several factors of dynamism, including wireless channels, user traffic, and user mobility, among others. Consequently, optimizing wireless networks has become an active area of research aimed at providing each communication scenario with the required quality of service (QoS) and quality of experience (QoE). Fixing certain configuration parameters through the operation time of a network will not be suitable for the changing conditions. This option sacrifices the system's performance in favor of simplicity. On the other hand, responding to continuous changes requires:

- An efficient mechanism to capture the continuous changes in the operation conditions and accurately define the current conditions.
- A mechanism to share this information among the different communicating entities (e.g., users, base stations, cloud centers, etc.).
- A reliable, efficient, and intelligent algorithm to adapt the different configurations to fit the current conditions.

Although this overhead significantly improves the system performance, it adds challenges and complexity to the operation algorithms and protocols. The main problem being addressed in this work is:

"How the various channel parameters be configured under dynamic wireless environments?"

In other words, solving this problem means we optimize the wireless network performance by adapting configurable parameters based on the current system conditions. However, due to the complexity of wireless communication systems, there is a huge number of such parameters and each of them is affected by different factors. Therefore, it is advisable to divide this problem into subproblems that can be solved more efficiently. Consequently, the main problem has been divided into the following subproblems:

0.2.1 SP 1: Subproblem 1

The first subproblem addresses the first processing block for the received signal, i.e., the carrier frequency offset (CFO) estimation, see Fig. 0.4. We can articulate the first subproblem as:

"How can we accurately estimate CFO in NR systems to improve the decoding and channel estimation tasks?"

Accurate CFO estimation is crucial for NR systems, as it directly impacts the system's performance, such as intersymbol interference, throughput, or error rate. The proposed research aims to investigate and compare different CFO estimation techniques, including conventional and AI/ML-based methods, to figure out the most accurate and efficient method. The work also considers various factors affecting CFO estimation, such as signal-to-noise ratio (SNR) to ensure the proposed technique is robust and adaptable to different propagation environments. By solving this subproblem, we aim to enhance the system throughout, leading to improved user experience and network efficiency.

C1: The main challenges related to SP1 are:

- Nonlinear Distortion: Nonlinearities in the transceiver components can introduce distortions in the received signal, making accurate carrier frequency offset estimation difficult. These nonlinearities are hard to model and predict because it changes with manufacturer and working environments. Moreover, it changes with time for the same transceiver.

- Doppler Shift: The movement of the transmitter or receiver can cause a Doppler shift in the received signal, resulting in a frequency offset. Accurately estimating and compensating for this offset is crucial for maintaining reliable communication in mobile scenarios. The Doppler shift changes with the mobility pattern of the UE. For example, a pedestrian UE will suffer a Doppler shift different than a UE in a vehicle moving at 120 Km/h.

- Noise and Interference: The presence of noise and interference in the received signal can affect the accuracy of carrier frequency offset estimation. The challenge lies in distinguishing between the desired signal and unwanted disturbances.

- Channel Estimation Errors: Imperfect estimation of the wireless channel can introduce errors in carrier frequency offset estimation. Estimating and compensating for the channel impairments accurately is crucial for achieving reliable frequency offset estimation.

- Time-Varying Environments: The characteristics of the wireless channel can change over time, leading to variations in the carrier frequency offset. Adapting to these dynamic environments and accurately tracking the offset pose significant challenges.

- Synchronization Overhead: Estimating the carrier frequency offset requires additional overhead in terms of signaling and computational resources. Balancing the accuracy of the estimation with the associated overhead is a challenge for an efficient system design. The overhead in URLLC applications will be different than in eMBB or mMTC.

0.2.2 SP 2: Subproblem 2

After the CFO estimation, the estimated CFO is used to correct the received signal, and the corrected signal is then used to estimate the CSI. MIMO-FDD systems require accurate CSI to achieve the full benefits of spatial multiplexing and beamforming. However, CSI feedback from multiple antennas in MIMO systems can be a significant overhead that affects the system's performance and requires large amounts of data to transmit. Therefore, CSI compression is an essential technique to reduce the overhead of CSI feedback while maintaining the required accuracy for efficient operation. The second subproblem can be articulated as:

> "How can we optimize the overhead of CSI feedback in MIMO-FDD systems while minimizing the impact of reconstruction loss on system performance?"

The proposed research aims to investigate and compare different CSI compression techniques to determine the most efficient and effective approach. The research considers various factors affecting CSI compressions, such as the compression ratio, quantization error, feedback noise, and feedback delay, to ensure the proposed technique is robust and adaptable to different scenarios. Furthermore, the research will evaluate the impact of different compression techniques on the system's performance, such as throughput, error rate, and energy efficiency. By addressing this research question, we aim to enhance the performance of MIMO-FDD systems while reducing the overhead of CSI feedback, leading to improved system efficiency and reduced resource consumption.

C2: The main challenges related to SP2 are:

- Channel Estimation Accuracy: Accurately estimating the wireless channel is essential for reliable CSI feedback. However, various factors such as noise, interference, and multipath propagation can introduce errors in the estimation process, affecting the quality of CSI feedback.

- Feedback Delay: The time required to estimate the channel, process the CSI, and transmit the feedback introduces a delay. This delay can impact the effectiveness of CSI feedback, especially in systems with rapidly changing channels or fast communication protocols. This factor eliminates some estimation/compression techniques that may reach good accuracy only because they are time-consuming.

- Feedback Overhead: Transmitting CSI information from the receiver to the transmitter requires dedicated radio resources and introduces additional overhead in terms of bandwidth and signaling. Optimizing the feedback overhead while ensuring sufficient CSI accuracy is a challenge.

- Channel Correlation: In some scenarios, multiple antennas at the receiver may exhibit correlated channels, meaning the CSI of one antenna can be highly correlated with the CSI of other antennas. Efficiently exploiting such correlations without sacrificing accuracy is a challenge in CSI feedback design.

- Time-Varying Channels: Wireless channels often experience time-varying characteristics due to mobility, environmental changes, and fading. Tracking these variations and providing timely and accurate CSI feedback poses a challenge, particularly in fast-fading or highly dynamic environments. Developing a global technique that works with the same efficiency in various conditions is a challenging task.

- Algorithm Complexity: The complexity and computational requirements for CSI estimation and feedback can be significant, especially in advanced wireless systems with multiple antennas. In light of the available computing resources, managing this complexity while maintaining real-time operations is a challenge.

0.2.3 SP 3: Subproblem 3

Given that now the BS has access to the current channel conditions, then it needs to select the best configuration that suits the current conditions, this process is known as *link adaptation*. Link adaptation is a critical process in wireless communications that adapts the transmission parameters, such as modulation and coding scheme (MCS), to ensure the best possible performance in the current conditions. However, traditional link adaptation techniques rely on accurate channel state information (CSI) feedback, which is challenging to obtain in practice due to channel estimation errors, feedback delays, and other factors. To address these challenges, we articulate the third subproblem as:

"Given the imperfect feedback, how do we select the configuration that best suits the current channel conditions?"

Recently, several research works proposed leveraging ML and AI techniques to enhance the performance of link adaptation algorithms while using imperfect CSI. In this dissertation, we aim to investigate and compare different ML and AI-based approaches for link adaptation in wireless communication systems. Moreover, we consider different factors affecting link adaptation, such as MCS, channel estimation errors, and feedback delay. We look at several possible modelings of the link adaptation problem such as reinforcement learning, classification, etc. The impact of the dataset set selection on the performance of the link adaptation algorithm is also examined to conclude insightful conclusions about the most efficient way of training a link adaptation agent.

C3: The main challenges related to SP3 are:

- Channel Dynamicity: Wireless channels exhibit time-varying characteristics due to factors such as multi-path fading, interference, and environmental conditions. Adapting the link parameters, such as modulation and coding scheme (MCS), to the varying channel conditions in real-time poses a challenge.
- Channel Estimation: Accurate channel state estimation is essential for effective link adaptation. However, channel estimation errors, noise, and interference can impact the reliability of the estimated channel information, making it challenging to select appropriate link parameters.
- Trade-off Between Data Rate and Reliability: Link adaptation involves finding the right balance between achieving higher data rates and maintaining sufficient reliability. Increasing the data rate by employing higher-order modulation schemes may lead to increased vulnerability to channel impairments, affecting the overall reliability of the link.
- Cross-Layer Optimization: Link adaptation is inherently a cross-layer problem, as it involves coordination and optimization across multiple layers of the communication protocol stack. Balancing the trade-offs and interactions between physical layer parameters, medium access control (MAC) protocols, and higher-layer algorithms is a complex challenge.
- Dynamic User Mobility: Link adaptation becomes particularly challenging in scenarios with fast-moving users or varying distances between the transmitter and receiver. Adapt-

ing the link parameters to the changing channel conditions due to user mobility requires efficient algorithms and mechanisms.

0.3 Objectives

The main objectives of this dissertation are:

- To develop a learning framework that maximizes the overall system throughput and reliability of wireless communication systems under dynamic propagation environments. This objective can be divided into the following sub-objectives:

  1. **SO1**: To analyze and design an efficient solution for the problem of CFO estimation in NR with high accuracy and generalization characteristics (i.e., it should work with different signal-to-noise ratios).

  2. **SO2**: To minimize the bandwidth overhead for wireless channel feedback in multiple-input multiple-output (MIMO) frequency division duplexing (FDD) systems. We evaluate the different techniques for minimizing the bandwidth overhead in the channel feedback process in MIMO-OFDM FDD systems. Moreover, we consider more challenging scenarios in which the feedback channel is not assumed to be noise-free.

  3. **SO3**: To efficiently adapt the transmission parameters, based on the algorithms developed in SO1 and SO2, in a way that maximizes the throughput and, at the same time, respects the reliability of the communication systems. After obtaining accurate channel state information (CSI), the transmitter adapts the transmission parameters (e.g., modulation and coding scheme (MCS), guard interval, bandwidth, etc.) accordingly to increase the achievable throughput and channel reliability.

0.4 List of Publications

During the course of his Ph.D. research, the author contributed to the following published and submitted research articles:

1. PEER-REVIEWED JOURNALS:

1.1 Hussien, Mostafa, Kim Khoa Nguyen, and Mohamed Cheriet. "Self-supervised learning for CSI compression in FDD massive MIMO systems." IEEE Communications Letters, IEEE, 26.11 (2022): 2641-2645.

1.2 Hussien, Mostafa, Kim Khoa Nguyen, Ali Ranjha, Moez Krichen, Abdulaziz Alshammari, and Mohamed Cheriet. "Enabling Efficient Data Integration of Industry 5.0 Nodes Through Highly Accurate Neural CSI Feedback." IEEE Transaction on Consumer Electronics, IEEE, DOI: 10.1109/TCE.2023.3294832, (2023).

1.3 Hussien, Mostafa, Ahmed Abdelmoaty, Mahmoud Elsaadany, Mohammed FA Ahmed, Ghyslain Gagnon, Kim Khoa Nguyen, and Mohamed Cheriet. "Carrier Frequency Offset Estimation in 5G NR: Introducing Gradient Boosting Machines." IEEE Access, IEEE, 11 (2023): 34128-34137.

1.4 Hussien, Mostafa, Kim Khoa Nguyen, and Mohamed Cheriet. "A Learning Framework for Bandwidth-Efficient Distributed Inference in Wireless IoT." IEEE Sensors Journal, IEEE, DOI: 10.1109/JSEN.2023.3283923, (2023).

1.5 Hussien, Mostafa, Kim Khoa Nguyen, and Mohamed Cheriet. "Data-Centric AI for Link Adaptation: Opportunities, Challenges, and Open Issues" IEEE Communications Magazine, IEEE (Submitted).

1.6 Hussien, Mostafa, Mohammed FA Ahmed, Kim Khoa Nguyen, and Mohamed Cheriet. "Evolution of MAC Protocols in the Machine Learning Decade: A Comprehensive Survey" Computer Science Reviews, Elsevier. (Submitted).

2. INTERNATIONAL CONFERENCE:

2.1 Mohammed FA Ahmed, Ghassan Dahman, Kim Khoa Nguyen, Mohamed Cheriet, and Gwenael Poitau. "Towards More Reliable Deep Learning-based Link Adaptation for WiFi 6." IEEE International Conference on Communications (ICC). IEEE, 2021.

2.2 Hussien, Mostafa, Kim Khoa Nguyen, and Mohamed Cheriet. "Fault-tolerant 1-bit Representation for Distributed Inference Tasks in wireless IoT." 17th International Conference on Network and Service Management (CNSM). IEEE, 2021.

2.3 Hussien, Mostafa, Kim Khoa Nguyen, and Mohamed Cheriet. "PRVNet: A Novel Partially-regularized Variational Autoencoders for Massive MIMO CSI Feedback." IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022.

2.4 Hussien, Mostafa, Ahmed Abdelmoaty, Mahmoud Elsaadany, Mohammed FA Ahmed, Ghyslain Gagnon, Kim Khoa Nguyen, and Mohamed Cheriet. "BiModule CFO Estimation (BMCE): Augmenting CFO Estimation with Temporal-correlation Modeling" IEEE International Conference on Communications (ICC). IEEE, 2023. (Submitted).

3. COLLABORATIONS:

3.1 Askarizadeh, Mohammad, Mostafa Hussien, Masoumeh Zare, and Kim Khoa Nguyen. "Optimized Transfer Learning for Wireless Channel Selection." IEEE Global Communications Conference (GLOBECOM). IEEE, 2021.

3.2 Hussien, Mostafa, Yi Tian Xu, Di Wu, Xue Liu, and Gregory Dudek. "Efficient Neural Data Compression for Machine Type Communications via Knowledge Distillation." IEEE Global Communications Conference (GLOBECOM). IEEE, 2022.

3.3 Askarizadeh, Mohammad, Mostafa Hussien, Alireza Morsali, and Kim Khoa Nguyen. "Modeling and Optimizing Resource-Constrained Instance-Based Transfer Learning."IEEE Global Communications Conference (GLOBECOM). IEEE, 2022.

3.4 Askarizadeh, Mohammad, Mostafa Hussien, Alireza Morsali, and Kim Khoa Nguyen. "Resource-Constrained Instance-Based Transfer Learning." IEEE Transactions on Neural Networks and Learning Systems. (Submitted).

PATENTS:

3.5 Hussien, Mostafa, Yi Tian Xu, Di Wu, Xue Liu, and Gregory Dudek. “Distillation Encoders: Efficient Neural Data Compression for Machine Type Communications via Knowledge Distillation.” WA-202303-008-1-US0

0.5 Dissertation Outline

This dissertation is composed of several research papers that have been published in or submitted to highly-ranked peer-reviewed journals and flagship conferences in the fields of wireless communications and AI. The focus of these papers is to improve various components and functions in the wireless communication systems pipeline. To facilitate the navigation of readers through this dissertation, Fig. 0.5 shows a roadmap for the dissertation chapters. Following is a detailed description of each part of this dissertation. The chapters are distributed and organized as follows:

PART 0: Background

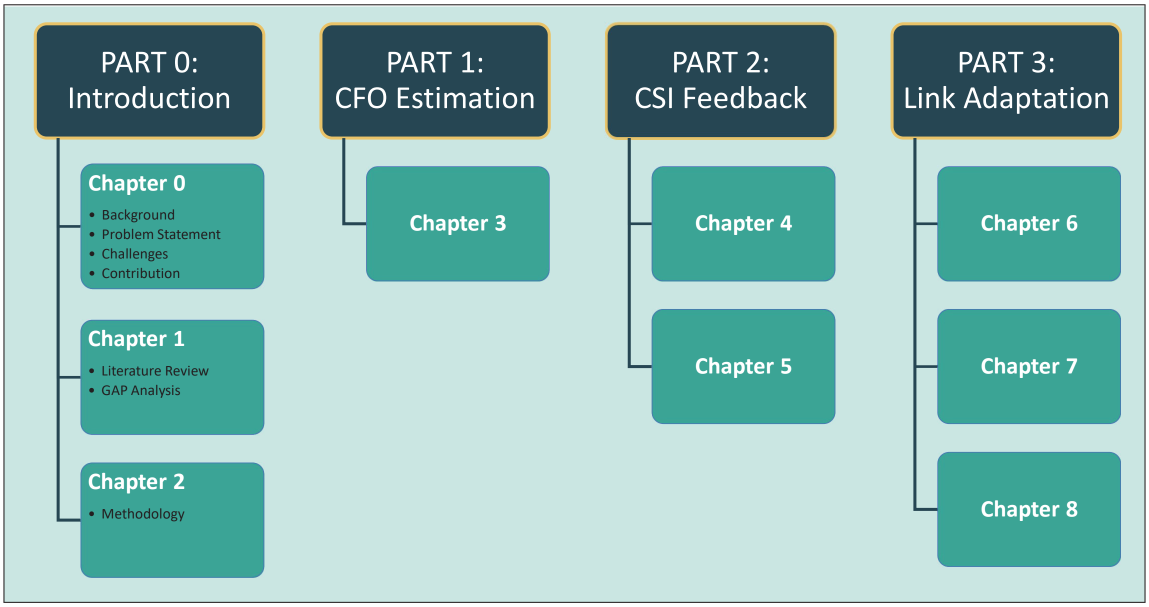


Figure 0.5 A roadmap for the dissertation chapters

- **Chapter 0: Introduction:** starts with a general background on next-generation communications and highlights their main features and characteristics. The problem statement is presented and divided into subproblems each of which addresses one issue. The challenges associated with these subproblems are then discussed. The contribution of the dissertation is then introduced and finally, we illustrate the dissertation outline.

- **Chapter 1: Literature Review:** presents an overview of prior studies on the identified subproblems. In addition, each chapter also provides a detailed description of the related work, emphasizing the challenges associated with the subproblem being addressed.

- **Chapter 2: Methodology:** presents the general methodology highlighting a detailed account of the research design, data collection methods, and data analysis techniques employed to address each subproblem, ensuring the study's validity and reliability.

PART 1: Carrier Frequency Offset Estimation

- **Chapter 3** presents our published work on CFO estimation using ensemble learning (i.e., gradient boosting machines, GBM). As one of the early steps in the pipeline of signal processing, CFO estimation plays a critical role in the subsequent processing blocks such as CSI estimation and feedback.

PART 2: CSI Feedback

- **Chapter 4** presents a step beyond the CFO estimation. Specifically, this chapter presents our published work on the problem of CIS compression leveraging variational autoencoder (VAE) architectures. This work focused on developing a feedback mechanism that is robust against noise in the feedback channel. We customized the VAE loss function and training algorithm to fit the specifications of the CSI compression problem.
- **Chapter 5** presents our work on CSI compression in which we address some of the limitations inherited in autoencoder-based solutions. In this chapter, we proposed a novel technique for CSI compression based on bias/variance tradeoff utilization.

PART 3: Link Adaptation

- **Chapter 6** presents our work on the link adaptation problem. After optimizing the CSI feedback in the work presented in previous chapters, it is time to use this feedback for optimizing the transmission parameters. This work tries to fill the gap between the CSI feedback and the link adaptation processes and adopt the optimized feedback to obtain a better link adaptation performance.
- **Chapter 7** presents our work that explores the potential of adopting a data-centric approach for the link adaptation problem. It proposes a novel approach for data-centric AI for link adaptation and presents the potential gain from optimizing the dataset selection only while keeping the model fixed.

- **Chapter 8** presents our work to combine the work of CSI compression and link adaptation in one step. Therefore, we jointly learn the CSI compression and the adaptation function in one model under the framework of distributed inference. This work encourages the compress-and-adapt behavior that compresses the CSI and adapts the transmission parameters based on the compressed representation without the need to reconstruct the original data.

CHAPTER 1

LITERATURE REVIEW

In the previous chapter, we introduced the problem of wireless network control at large. Then, we narrowed down our study to include carrier frequency offset (CFO) estimation, channel state information feedback, link adaptation, and communication-efficient distributed inference. In this chapter, we cover the recent research work toward optimal wireless network control, especially in the aforementioned problems. To this end, we divide this chapter into four main subsections and dedicate a section for each of these problems.

1.1 Carrier Frequency Offset Estimation

The problem of CFO synchronization has garnered interest from many researchers. Some have explored the Primary Synchronization Signal (PSS) in the time domain, as reported in Nassralla & et al. (2015), while others have focused on the frequency domain. The authors in Nassralla & et al. (2015) suggested a method for detecting PSS by identifying the highest cross-correlation, but the Integer Carrier Frequency Offset (ICFO) affects the precision of PSS detection. To address this issue, differential correlation-based PSS detection approaches utilizing Discrete Fourier Transform (DFT) or Fast Fourier transform (FFT) have been proposed Morelli & Moretti (2015); Lin *et al.* (2015). Several works have aimed at improving the performance of OFDM systems through the joint optimization of CFO and other parameters. Shaked *et al.* (2017) explored a framework for joint estimation of CFO and the Channel Impulse Response (CIR) in linear periodic channels, utilizing a pilot-aided approach. A Joint Maximum Likelihood Estimator (JMLE) was proposed, offering improved spectral efficiency and reduced computational complexity, as it leverages the periodicity and sparsity of the channel. Abdzadeh-Ziabari *et al.* (2017) tackled the challenge of high mobility systems by performing joint timing, channel, and CFO estimation. To address the high complexity of joint estimation, a computationally efficient algorithm based on Basis Expansion Modeling (BEM)

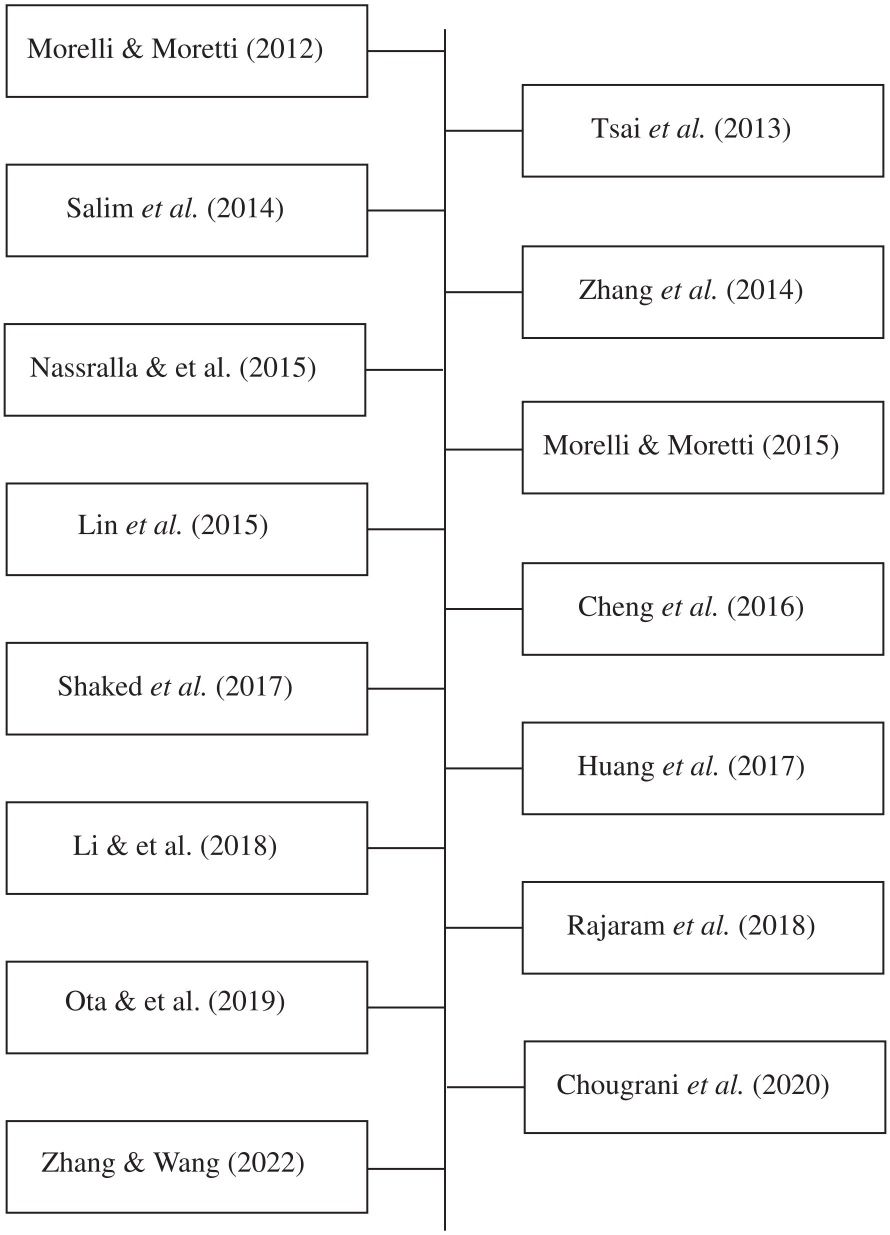


Figure 1.1 A timeline for the literature work in CFO estimation

was introduced. The algorithm tracks channel variations to minimize the number of unknown channel parameters and has been shown to outperform other benchmark algorithms.

In the last few years, the limitations of traditional methods triggered researchers to deploy state-of-the-art data-driven ML techniques for CFO estimation Chougrani *et al.* (2020); Rajaram *et al.* (2018); Ota & et al. (2019); Chougrani *et al.* (2020); Ninkovic & et al. (2021);

Zhang & Wang (2022); Li & et al. (2018). For example, the authors of Rajaram *et al.* (2018) have created a new receiver for a single-carrier modulation that performs joint carrier frequency offset and channel estimation. The receiver uses a frequency-domain equalization technique along with simultaneous wireless information and power transfer (SWIPT) by employing a highly energetic pilot signal. The pilot signal serves two purposes: it transmits power for energy harvesting and also helps estimate the CFO and channel conditions. The receiver has been designed to handle strong interference levels during channel estimation and data detection, offering a flexible and resource-efficient design approach. While the work in Ota & et al. (2019) explores the detection probability of physical-layer cell identity (PCID) in 5G new radio (NR), taking into account frequency offset. It compares three primary synchronization signal (PSS) detection techniques: cross-correlation before frequency offset (FO) estimation and compensation, cross-correlation after FO estimation and compensation, and autocorrelation at a set of user equipment. Simulation results show that cross-correlation PSS detection before FO estimation and compensation performs best in the carrier frequency region below about 14 GHz, with an average received signal-to-noise power ratio of 0 dB and a standard oscillator frequency error of 1 ppm. However, for the frequency region above approximately 14 GHz, cross-correlation PSS detection after FO estimation and compensation demonstrates higher detection probabilities than its counterpart. The PCID detection probability of this latter method is nearly equal to that of the autocorrelation-based PSS detection as the frequency value increases.

The 3rd Generation Partnership Project (3GPP) introduced the narrowband Internet-of-Things (NB-IoT) standard to connect numerous low-cost, low-complexity, and long-life IoT devices with extended coverage. To enhance power efficiency, 3GPP proposed a single-tone frequency-hopping scheme for the random access (RA) waveform in NB-IoT. RA manages the initial connection between user equipment (UE) and the base station (BS), enabling identification and synchronization. However, detecting the new waveform and achieving accurate timing synchronization is challenging due to radio impairments like carrier frequency offset (CFO). To address this issue, the authors in Chougrani *et al.* (2020) proposed a new receiver method for the NB-IoT physical RA channel (NPRACH). The proposed method effectively eliminates CFO without extra computational complexity and supports all NPRACH preamble formats. Performance evaluations under 3GPP conditions show significantly improved detection accuracy and complexity compared to 3GPP requirements and existing state-of-the-art methods. The authors in Aoudia & et al. (2022) present a neural network (NN)-based algorithm for device detection and time of arrival (ToA) and carrier frequency offset (CFO) estimation in the narrowband physical random-access channel (NPRACH) of narrowband internet of things (NB-IoT). The proposed NN architecture uses residual convolutional networks and knowledge of the 5G New Radio (5G NR) preamble structure. When tested against a state-of-the-art baseline on a 3GPP urban microcell (UMi) channel model, the method achieves up to 8 dB gains in false negative rate (FNR) and significant improvements in false positive rate (FPR) and ToA and CFO estimation accuracy across various channel conditions, CFOs, and transmission probabilities. This base station (BS) synchronization method adds no extra complexity to user devices and could potentially extend battery life by reducing preamble length or transmit power.

The work in Li & et al. (2018) addresses the multi-frequency synchronization issue in orthogonal frequency-division multiple access (OFDMA) uplink communications, where each user may have a different carrier frequency offset (CFO) that is difficult to compensate at the receiver side. The main contribution is the development of a novel OFDM receiver that can handle unknown random CFOs using a CFO-compensator bank. The CFO range is divided into sub-ranges, with each supported by a dedicated CFO compensator. Since optimizing the CFO compensator is an NP-hard problem, a machine deep-learning approach is proposed to find a sub-optimal solution. The proposed receiver provides inter-carrier interference-free performance for OFDMA systems across a wide range of SNRs. The work in Dreifuerst & et al. (2020) focuses on developing and analyzing deep learning architectures for estimating the carrier frequency of a complex sinusoid in noise using 1-bit samples of in-phase and quadrature components. This estimation is used in GSM and serves as a starting point for more comprehensive solutions with other signal types. Four different deep learning architectures are trained on eight datasets, considering the impact of signal-to-noise ratios (SNR), quantization, and sequence lengths on estimation error. The architectures are also analyzed for scalability in MIMO receivers. Simulation results show that training with quantized data from signals with SNRs between $[0-10]$ dB improves deep learning estimator performance across the entire SNR range. Convolutional models demonstrate the best performance and faster execution times compared to FFT methods. The approach can accurately estimate carrier frequencies from 1-bit quantized data with fewer pilots and lower SNRs than traditional signal processing methods.

1.2 Channel Feedback in Wireless Networks

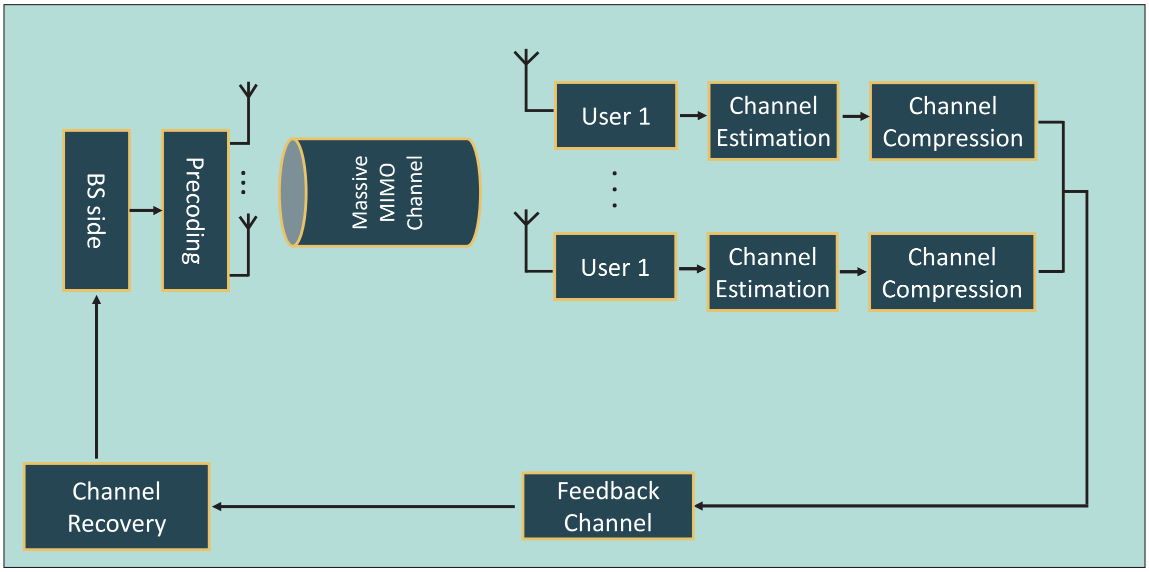


Figure 1.2 A general flow of a typical CSI feedback in MIMO Systems

As discussed in previous chapters, channel reciprocity is achieved in TDD systems. However, in FDD systems, this channel reciprocity is not achieved, and in this case, a feedback mechanism is required to keep the BS updated about the downlink channel, see Fig. 1.2. Unfortunately, in MIMO systems, this information is huge enough to consume considerable bandwidth. To optimize the performance of the wireless system, a compression technique is required to compress the channel before sending it back to the BS. Two major approaches are used to reach this goal: 1) traditional techniques which include compressive sensing, PCA, and vector quantization; 2) encoder-based neural network techniques. In the following two subsections, we are going to present the main work of each technique.

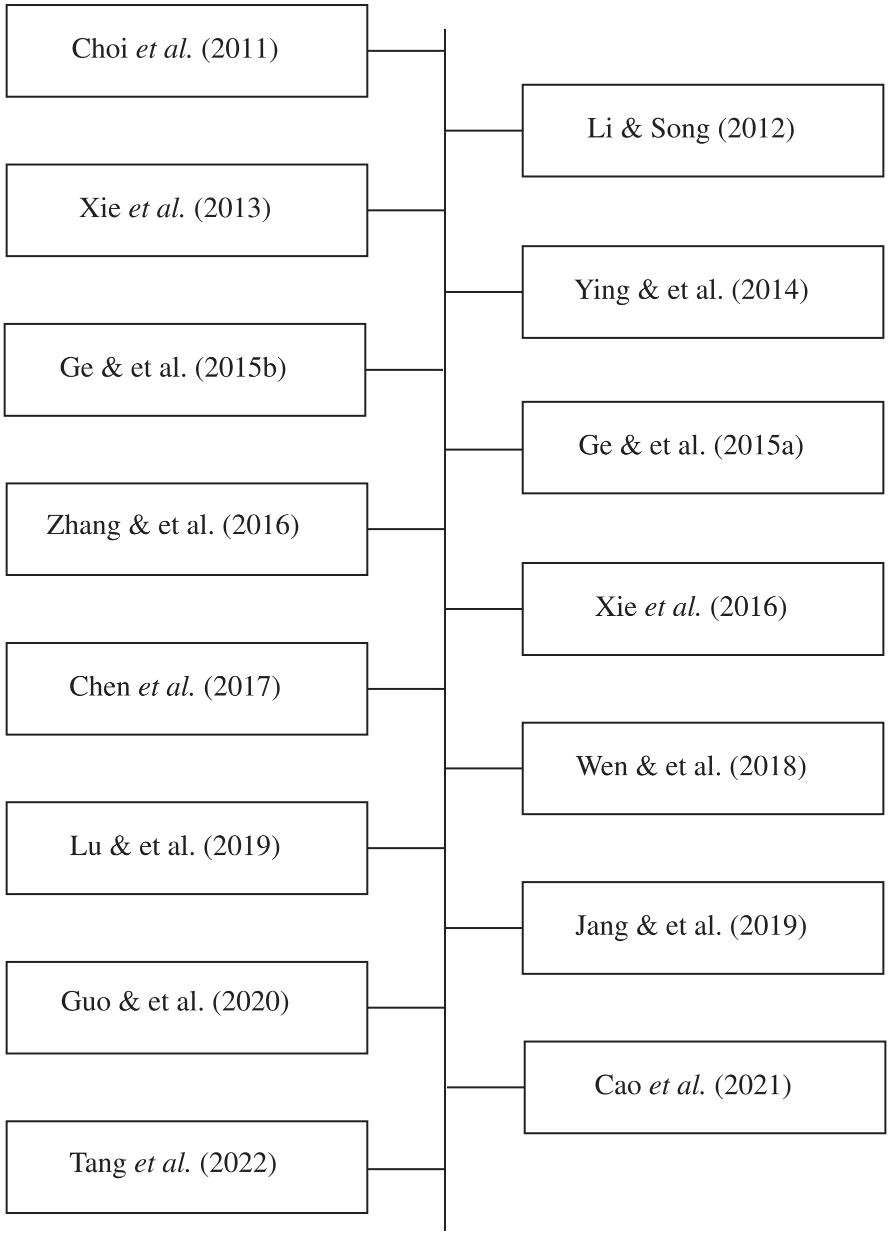


Figure 1.3 A timeline for the literature work in CSI feedback

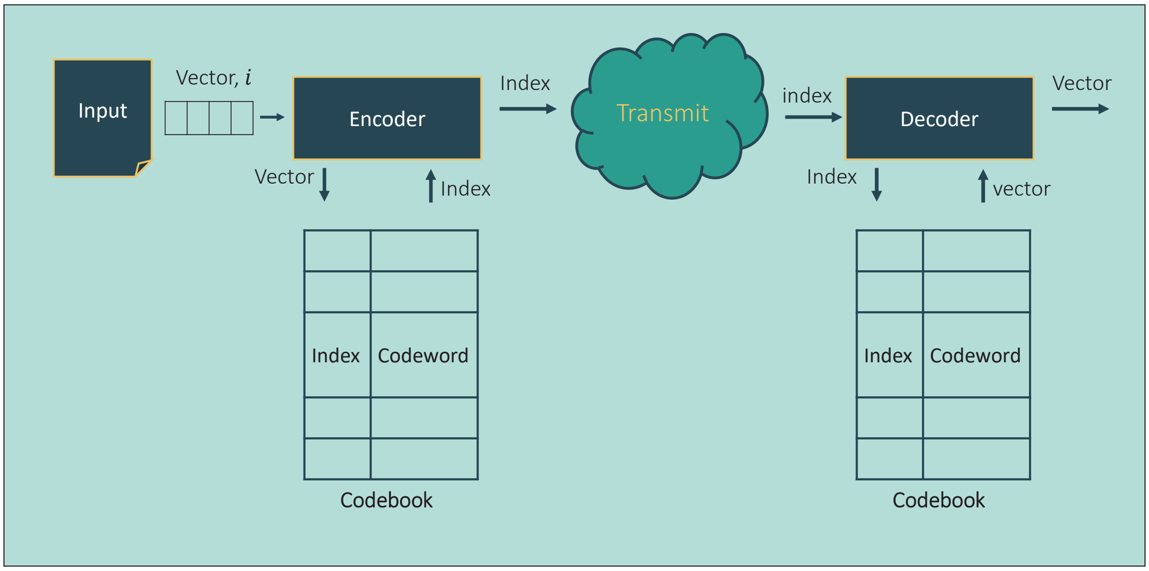


Figure 1.4 The use of vector quantization in CSI compression

1.2.1 Traditional Feedback Techniques

PCA is one of the dimensionality reduction techniques used in machine learning literature Martın-Clemente & Zarzoso (2016). PCA has been used as a tool for compressing CSI before transmission to the BS. The authors in Ge & et al. (2015a,b); Zhang & et al. (2016) employed PCA to reduce the dimension of the feedback in a multi-user scenario. Although the authors achieved a good compromise between the compression rate and reconstruction error, the BS is not able to obtain an accurate CSI reconstruction. This can be explained by the well-known limitations of PCA like the linear assumption between dataset features.

The concept of vector quantization (VQ) is a traditional technique used for data compression, especially, images and videos Si *et al.* (2017). Instead of quantizing scalar values, VQ considers quantizing a complete list of scalars (vector) to its nearest vector from a set of carefully selected vectors, called codebook. The index of the nearest vector is then transmitted to the receiver. The receiver uses the received index to find a vector in the codebook that is the closest approximation to the original vector, see Fig. 1.4. A distance function is required to measure how a certain vector is close to another vector. A variety of distance measures can be applied in this context, Euclidean distance 1.1 is one of the widely used distance functions.

$$d(x,y) = \sqrt{\sum_{i=0}^{N}(x_i - y_i)} \tag{1.1}$$

Vector quantization has been widely used for compressing the CSI channel before feeding it back to the BS. The authors in Love & Heath (2006); Raghavan *et al.* (2007); Mielczarek & Krzymien (2008); Choi *et al.* (2011); Ying & et al. (2014) have investigated the feasibility of VQ technique in CSI compression and feedback. Although the obtained results were promising, VQ suffers from scalability issues. This limitation comes from the fact that the codebook size grows exponentially with the number of feedback bits Chen *et al.* (2017).

Compressive sensing (CS) is another technique widely employed in the compression of channel feedback. Compressive sensing is a sampling technique that exploits signal sparsity to reconstruct a signal from fewer samples. For compressive sensing to work, two conditions should be achieved. First, the signal sparsity, which requires the signals to be sparse in a certain domain. Second, the incoherent nature of observations in the acquisition domain. CS works by solving an undetermined system of linear equations under sparsity constraint. The performance of the CS technique outperforms traditional Nequest sampling theorem Zhang *et al.* (2018).

In Song *et al.* (2009) and Li & Song (2012), the authors applied compressive reconstruction to reconstruct the CSI from a few samples sent by the UE to the BS in a zero-force MIMO network. The feedback channel (uplink) is assumed to be perfectly known at the BS, which means the feedback is assumed to be sent over an error-free channel. This assumption may not be practical in real scenarios, and the amount of time and feedback used to train uplink channels can burden the performance of such techniques.

The CS-based techniques have three main limitations: 1) since the signal reconstruction algorithms for CS-based techniques are iterative algorithms, they suffer from slow reconstruction performance. 2) the performance of CS techniques heavily depends on the sparsity assumption, which may not strictly hold in some MIMO wireless cases. 3) CS-based techniques employ random projection which does not fully exploit channel structures.

1.2.2 Deep Learning-Based Feedback

For one decade and deep learning is consistently achieving incomparable performance in many fields, e.g., computer vision, natural language processing. Due to the inherent problems of compressive sensing techniques, investigating deep learning for CSI feedback problems has gained great attention. Most of the work in this direction employs encoder/decoder architectures, especially autoencoders, to lossy compress/decompress CSI matrices.

The work in Wen & et al. (2018) opened the door for employing deep learning techniques in CSI feedback problems. The authors in Wen & et al. (2018) presented a deep learning architecture called *CsiNet*. *CsiNet*, a convolutional neural network architecture with skip connections in the decoder part. The superiority of CsiNet performance has been proven against traditional CS-based techniques. However, CsiNet is a point estimation model which means that for each dimension in the codeword, the model learns one value. This implies that any noise in this value can largely hurt the reconstruction quality at the BS.

The authors in Lu & et al. (2019) exploited the temporal and frequency correlations of wireless channels. They presented a model called *CSINet-LSTM*, which extends CsiNet with long short-term memory (LSTM) network. LSTM is a classic type of recurrent neural network that is capable of learning long-time dependencies, and temporal correlation, between the input samples.

In Lu *et al.* (2020), the authors proposed a neural network architecture, called *CRNet*, for multi-resolution CSI feedback in massive MIMO. The model is shown to have an improved performance against classic CS-based techniques as well as CsiNet. Another extension to the CsiNet model called *CsiNet+* is introduced in Guo & et al. (2020). However, the number of floating-point operations (flops) in CSINet+ is much larger than the CSINet, which can argue that the improvements come at the cost of complexity.

The authors in Liu *et al.* (2020) went further and integrated a neural quantizer with the standards encoder architecture to quantize the generated codewords. The authors in Lu *et al.* (2019) considered a deep residual convolutional network to compensate for the effect of quantization error after CSI compression. The proposed architecture, *JC-ResNet)*, jointly optimizes the quantization error along with the feature dimensionality reduction. While most of the work in the literature assumes ideal feedback channels, the authors in Jang & et al. (2019) attempted to construct reliable codewords for the CSI feedback problem. The codewords in Jang & et al. (2019) are designed to be robust against the noise introduced in the uplink channel. The authors also applied a uniform quantization for the generated codeword, and the decoder is trained to compensate for the error introduced by the quantization process. Uniform quantization is a simple well-known analog-to-digital conversion technique in which each sample value is rounded to one level in a set of, L, predetermined levels. If each sample will be coded using n bits, then $L = 2^n$. Assume the minimum and maximum levels are denoted by L_{min} and L_{max} respectively. Each sample value, s, is mapped to one of the L levels based on the equation in 1.2.

$$\bar{s} = \delta \left\lfloor \frac{s}{\delta} \right\rceil \; where \;\; \delta = \frac{L_{max} - L_{min}}{2^n - 1} \tag{1.2}$$

The authors in Qing *et al.* (2019) combined the superimposed coding and deep learning for CSI feedback. The downlink CSI matrix is spread and superimposed on the uplink user data sequence toward the base station. The base station then uses a neural network to recover the downlink CSI.

1.3 Wireless Link Adaptation

Wireless channels exhibit rich dynamics. These dynamics are inherent from many sources, such as channel variation due to node mobility, interference from hidden stations, or random channel errors. These dynamics drive the efforts toward the design of robust link adaptation algorithms to accommodate the effect of channel dynamics.

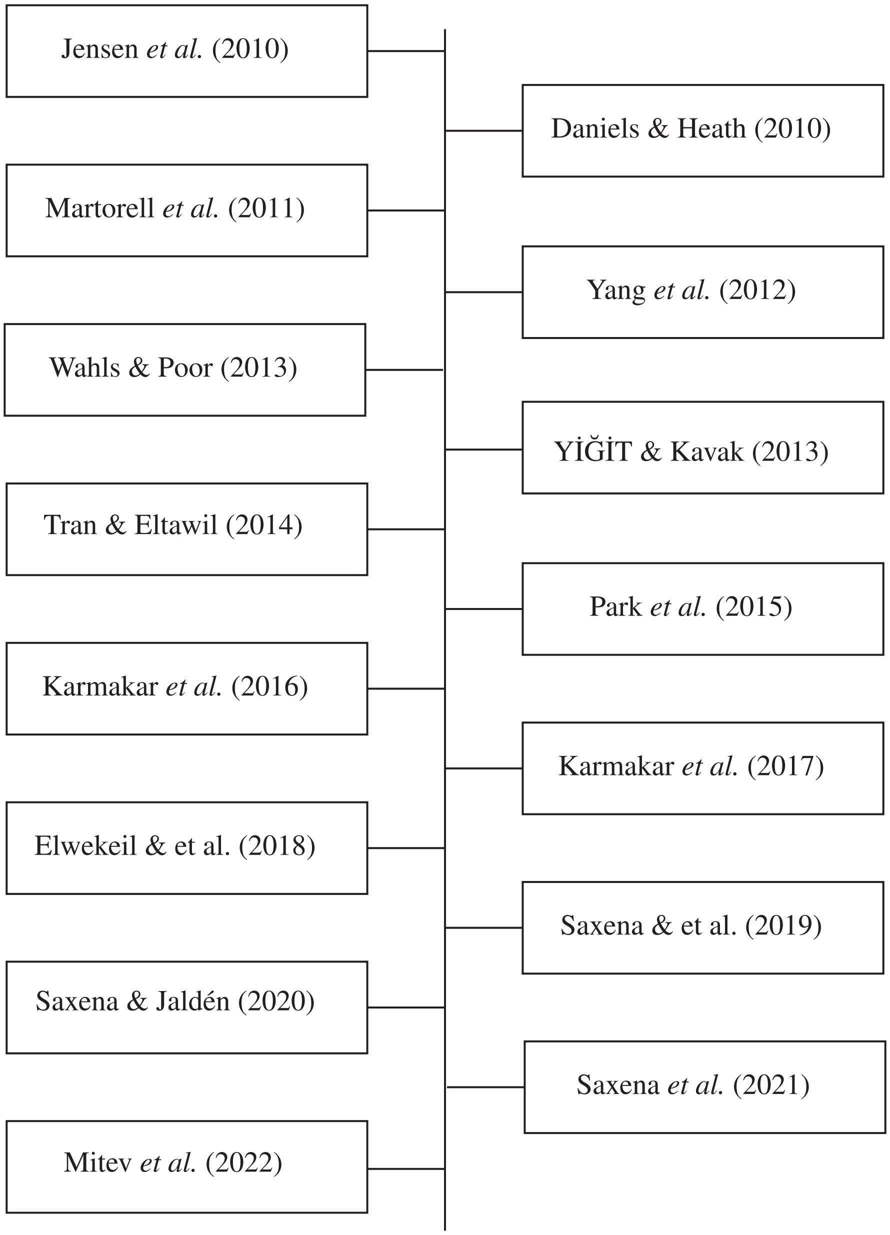


Figure 1.5 A timeline for the literature work in link adaptation

The authors in Judd *et al.* (2008) developed a channel-aware rate adaptation algorithm, *CHARM*, which uses the history of signal strength measurements captured by the wireless card to adapt the transmission rate. The reported results show that CHARM obtained a superior performance over traditional probe-based algorithms in dynamic channels.

In Wong *et al.* (2006), the authors proposed a robust rate adaptation algorithm (RRAA) algorithm that depends on the short-term loss ratio to decide the appropriate rate. They employed an RTS filter to avoid collision losses from rate decrease triggering. The experiments showed the robustness of RRAA compared to other adaptation algorithms.

In Jensen *et al.* (2010) the authors proposed a fast link adaptation algorithm for IEEE 802.11n standards. The algorithm is based on Mutual Information (MI) and the authors conduct a comprehensive evaluation of its PER-estimation accuracy. Through Monte Carlo simulations, a comparative analysis is performed by evaluating the performance of the proposed MI-based against other approaches.

1.3.1 Machine Learning-Based Link Adaptation

A line of work has been performed in the field of AMC. In this section, we will highlight some of the work that has been carried out in recent years. We also present some rational work, before the breakthrough of machine learning, for the sake of completeness. The limitations of each work are highlighted and discussed.

In Daniels & Heath (2010), the authors proposed using support vector machines (SVM) to perform online learning AMC. The authors used an SVM-based algorithm to develop an online learning platform for AMC. This platform can optimize AMC to the unique (potentially dynamic) hardware characteristics of each wireless device in selective channels. Simulation results on IEEE 802.11n protocol show that the proposed algorithm achieves a good rate/reliability tradeoff in each operating device using the error information of each frame. However, the system may waste a portion of bandwidth communicating the frame error information from the transmitter and receiver. Also, SVM is very sensible to the feature vector used in classification.

Most of the work in AMC treats the problem as a classification problem. Supervised learning techniques were used to carry out the AMC task. However, supervised techniques require an offline training phase that may harden the application of such systems in real-life systems. Some work has been proposed to solve the AMC task using reinforcement learning. In Yun & Caramanis (2010) for example, the authors built a reinforcement learning model and introduced a new learning approach based on online kernelized support vector regression (SVR) for link adaptation in MIMO-OFDM wireless systems. The proposed online kernelized SVR-based algorithm requires less memory and executes in order of magnitude to other supervised algorithms. The simulation results show that the proposed system performs as accurately as traditional supervised techniques but with fewer memory requirements. However, the authors assumed perfect channel knowledge at the receiver, which means they did not include the effect of channel estimation error. They also ignored the feedback delay, therefore, the obtained results do not reflect real situations. However, it can give an intuition on the algorithm's performance.

In YİĞİT & Kavak (2013), the authors proposed a neural network-based AMC for optimizing the best modulation and coding scheme (MCS) under a packet error rate (PER) constraint in MIMO-OFDM systems. The authors used a multilayer perceptron (MLP) architecture for the optimization task and compared their results with KNN algorithm in the cases of frequency-flat (1-tap) and frequency-selective (4-tap) wireless channel conditions. The simulation results show that the best performance is obtained with a CNN consisting of two hidden layers with five perceptrons in each layer.

In Dong & et al. (2018), the authors proposed an ML-based method for link adaptation in MIMO-OFDM systems. The method tries to maximize the system throughput and meets certain codeword error ratio (CWER) constraints. The authors proposed the use of an autoencoder for feature extraction and multi-class SVM for classifying the best MCS given the SINRs. The authors proposed another method for adapting MCS and spatial mode jointly through a channel matrix using autoencoder-softmax architecture.

In Elwekeil & et al. (2018), the authors proposed a deep learning approach for AMC in MIMO-OFDM systems. The authors used the estimated channel coefficients and the noise standard deviation to train convolutional neural networks to predict the appropriate modulation and coding scheme. The system obtained good performance even with impairments, such as imperfect timing synchronization, carrier frequency offset, and channel estimation errors at the MIMO-OFDM receivers. The obtained results are better than the KNN, SVM, and DNN approaches in terms of the packet error rate (PER) and the system throughput.

1.4 Distributed Inference

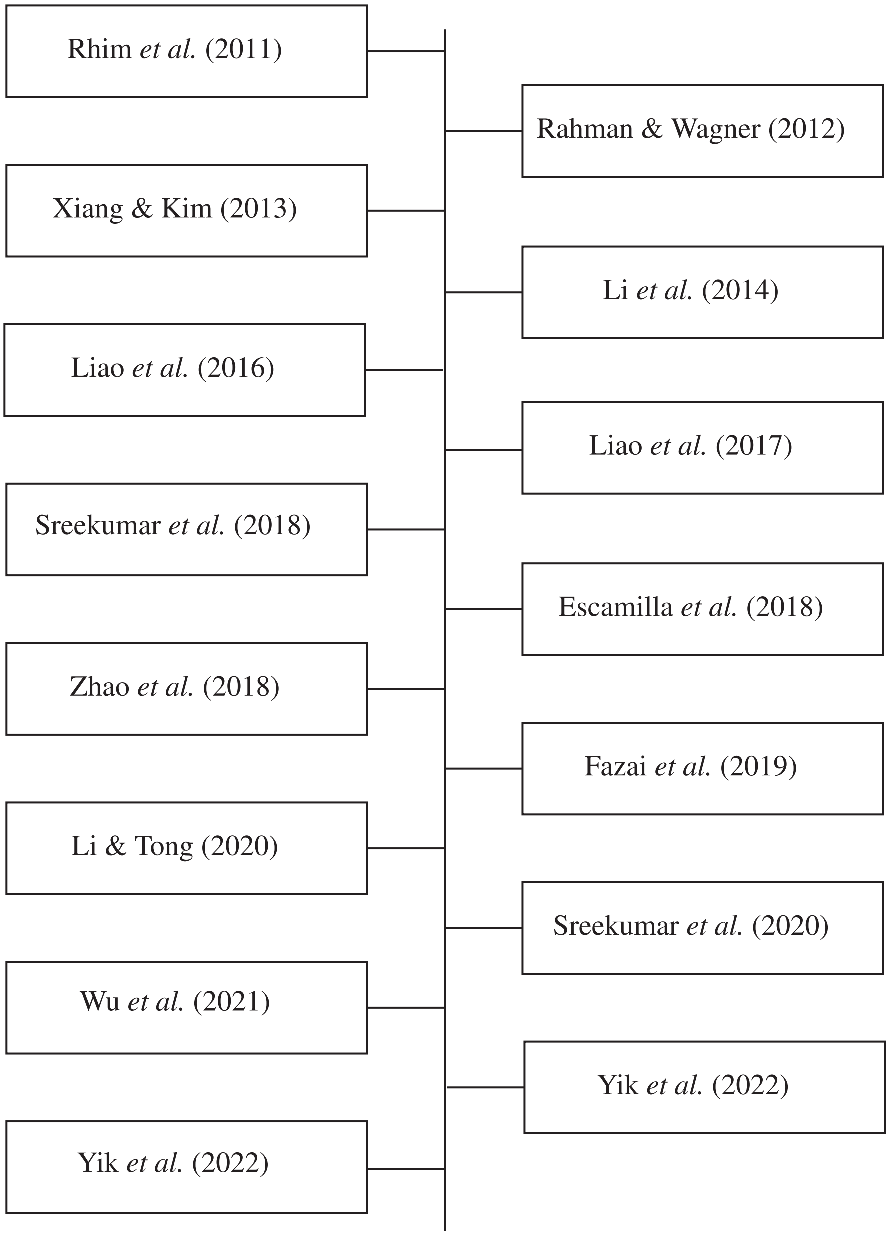


Figure 1.6 A timeline for the literature work in distributed inference

As we discussed earlier, link adaptation can be seen as the choice of a configuration profile that best suits the current channel conditions. Therefore, it is merely a choice. Among the several possible techniques for making such decisions, distributed inference (a.k.a., distributed detection, distributed hypothesis testing, etc.) is a powerful and widely adopted technique. It has been extensively studied in the context of **Statistical Decision Theory** Pratt *et al.* (1995). Because of the rich work in this area as well as distributed inference was one of the techniques we adopted for link adaptation, we dedicate a section to the literature work in this area.

The authors in Escamilla *et al.* (2018) considered a detection system consisting of a single sensor and K detectors. Each terminal is assumed to observe a memoryless source sequence, and the sensor transmits a common message to all the detectors. The communication channel is assumed to be error-free, but the rate is limited. The joint probability mass function of the source sequences observed at the terminals depends on an M-ary hypothesis, where M is greater than or equal to K. The primary objective of their communication system is to enable each detector to determine the underlying hypothesis. Specifically, each detector k strives to maximize the error exponent under hypothesis k while ensuring a low probability of error under all other hypotheses. The authors of Escamilla *et al.* (2018) present an achievable exponents region for the case of a positive communication rate and characterize the optimal exponent's region for the case of a zero communication rate.

The authors in Wu *et al.* (2021) investigated the issue of distributed hypothesis testing in a network of mobile agents with limited communication and sensing capabilities to collaboratively infer the true hypothesis. Specifically, the authors considered a scenario where an unknown subset of agents is compromised and may deliberately disseminate altered information to undermine the team objective. To address this challenge, two distributed algorithms are proposed, wherein each agent maintains and updates two sets of beliefs, namely local beliefs (LB) and actual beliefs (AB), which represent probability distributions over the hypotheses. At each time step, every agent shares its AB with other agents within its communication range, while updating its LB based on local observations. The shared information is then used to update ABs, subject to specific conditions. One algorithm requires a certain number of shared ABs at each time instant, while the other accumulates shared ABs over time and updates them when the number of shared ABs exceeds a prescribed threshold. If the conditions are not met, both algorithms rely on the agent's current LB and AB to update the new AB. The authors prove, under mild assumptions, that the AB for every non-compromised agent converges almost surely to the true hypothesis, without requiring connectivity in the underlying time-varying network topology. The proposed algorithms are demonstrated through a simulation of a team of unmanned aerial vehicles aimed at classifying adversarial agents. The authors compare the two algorithms and show experimentally that the second algorithm outperforms the first algorithm in terms of the speed of convergence.

Privacy-preserving in distributed hypothesis testing has been considered in various works such as Liao *et al.* (2017, 2016); Sreekumar *et al.* (2020, 2018); Abbasalipour & Mirmohseni (2022). In the following paragraphs, we elaborate in more detail on some work that considered the privacy of distributed hypothesis testing.

The work in Sreekumar *et al.* (2018) examined a distributed binary hypothesis testing problem between two entities: a remote observer and a detector. The remote observer is equipped with a discrete memoryless source and transmits its observations to the detector through a noiseless rate-limited communication channel. The detector's objective is to test the independence between its own observations and those of the observer, given some additional side information. The primary goal of the detector is to maximize the type 2 error exponent of the test while adhering to a specified type 1 error probability constraint. Additionally, the detector seeks to preserve the privacy of a private portion, which is correlated with the observer's observations, from the detector. The authors presented a comprehensive single-letter characterization of the rate-error exponent-equivocation and rate-error exponent-distortion tradeoffs while achieving a tight bound.

The work in Abbasalipour & Mirmohseni (2022) examines the issue of distributed binary hypothesis testing in the Gray-Wyner network with side information. An observer has access to a discrete memoryless and stationary source and transmits its observation to two detectors via one common and two private channels, all of which are assumed to be error-free but rate-limited. Additionally, each detector has access to its own discrete memoryless and stationary source, i.e., the side information. The primary objective is to perform two separate binary hypothesis tests on the joint distribution of observations at the detectors. Furthermore, the observer aims to maintain the privacy of a correlated latent source from the detectors. The degree of privacy preserved for the latent source is assessed using equivocation. To address this problem, an achievable inner bound is derived for the general case by incorporating a non-asymptotic analysis of the output statistics of the random binning.

The overlapping between machine learning and statistical hypothesis testing has been explored in a large line of work Li & Tong (2020); Fazai *et al.* (2019); Messner (2023); Mitra *et al.* (2020); Chen *et al.* (2021); Yik *et al.* (2022); Sheffet (2018); Rodriguez-Conde *et al.* (2023); Zhao *et al.* (2018). The following paragraphs explore various works in the overlap area of distributed hypothesis testing and machine learning.

The field of data science commonly employs two related but distinct strategies: hypothesis testing and binary classification. However, selecting the most suitable strategy for a particular analysis can be complex and difficult. The work in Li & Tong (2020) aims to clarify the differences between these two strategies in three areas and offer five practical guidelines to aid data analysts in choosing the most appropriate approach based on specific analytical needs. To demonstrate the practical application of these guidelines, the study presents a cancer driver gene prediction example.

The work in Fazai *et al.* (2019) presents the development of a machine learning approach integrated with statistical hypothesis testing to create an intelligent framework for photovoltaic (PV) fault detection, aiming to improve the efficiency and reliability of PV systems. Fault detection in PV systems is essential to ensure optimal energy harvesting and reliable power production since these systems frequently encounter various faults in harsh outdoor environments. Therefore, this paper focuses on detecting various faults during different modes of operation. The proposed approach combines the advantages of machine learning techniques with statistical hypothesis testing to enhance the fault detection and monitoring of PV systems, including both normal and abnormal conditions. The modeling phase of the framework employs ML techniques, while the generalized likelihood ratio test (GLRT) chart is used for fault detection. Specifically, an ML technique calculates the monitored residuals and GLRT chart analyzes the residuals for fault detection. The proposed ML-based GLRT algorithm is tested using both simulated and real PV data, and the results are evaluated in terms of false alarm rates (FAR), missed detection rates (MDR), and computation time.

The authors in Messner (2023) enhance the existing explainable artificial intelligence (XAI) methods by introducing a model-agnostic hypothesis testing framework for machine learning. To achieve this, the Fisher's variable permutation algorithm is modified to calculate an effect size measure, equivalent to Cohen's f^2 for OLS regression models. Additionally, the *Mann-Kendall* test of monotonicity and *Theil-Sen* estimator are utilized to determine the direction of influence and statistical significance of a variable using Apley's accumulated local effect plots. The effectiveness of this approach is demonstrated on an artificial dataset and a social survey through Python sandbox implementation.

The work in Mitra *et al.* (2020) focuses on a scenario where a group of agents receive partially informative private signals and aim to learn the true underlying state of the world generating their joint observation profiles from a finite set of hypotheses. To solve this problem, a distributed learning rule is proposed, which differs from existing methods as it does not employ any form of "belief-averaging." Instead, agents update their beliefs based on a min-rule. Under standard assumptions on the observation model and network structure, it is established that each agent learns the truth asymptotically almost surely. The main contribution of this study is to prove that each false hypothesis is eliminated by every agent exponentially fast, at a network-independent rate that is higher than the existing rates. The study further introduces a computationally efficient variant of the learning rule that is resistant to agents who do not behave as expected and deliberately spread misinformation, as represented by a Byzantine adversary model.

The authors in Chen *et al.* (2021) introduce a statistical hypothesis test for the deep neural network to learn the implicit representation of CT slices, specifically for COVID-19 CT scan classification. Their proposed approach is called Adaptive Distribution Learning with Statistical Hypothesis Testing (ADLeaST), which can evaluate the significance of each slice in the CT scan. The nonparametric statistics method, the Wilcoxon signed-rank test, is adopted to make the predicted result both explainable and stable, reducing the impact of out-of-distribution (OOD) samples. Additionally, a self-attention mechanism is introduced into the backbone network to explicitly learn the importance of the slices. The experimental results indicate that both proposed methods are stable and effective. Furthermore, the ADLeaST approach significantly outperforms existing state-of-the-art methods.

1.5 Summary and Gap Analysis

In this section, a comprehensive review of the existing literature on the addressed problems is presented. An overall analysis is conducted, examining the merits and shortcomings of each work. As discussed in the previous chapters, the body of research concerning each subproblem can be divided into two primary classes: conventional techniques and AI-based techniques. Table 1.1 provides a succinct overview of the studies encompassed within each category, along with their common benefits and constraints.

Table 1.1 Gap analysis for the literature work

PROBLEM	CATEGORY	WORK	REALTIME PERFORMANCE	ACCURACY	ADOPTED TECHNIQUE	TECHNOLOGY	GENERALIZATION	DATA REQUIREMENT
CFO Estimation	Conventional techniques	Morelli & Moretti (2015)	Low	Average	Maximum Likelihood	LTE	High	Low
		Shaked *et al.* (2017)	Low	Average	Joint Maximum Likelihood Estimator	AWGN	High	Low
		Abdzadeh-Ziabari *et al.* (2017)	Low	Average	Maximum Likelihood	Vehicular Networks	High	Low
	AI-based	Aoudia & et al. (2022)	Average	High	Neural Network	IoT	Average	High
		Dreifuerst & et al. (2020)	Average	High	Neural Networks	GSM	Average	High
		Li & et al. (2018)	Average	High	Neural Networks	AWGN	Average	High
CSI Feedback	Conventional	Martın-Clemente & Zarzoso (2016)	Average	Average	PCA	LTE	High	Low
		Si *et al.* (2017)	Low	Average	Vector Quantization	LTE	High	Low
		Li & Song (2012)	Low	Average	Compressive Sensing	LTE	High	Low
	AI-based	Wen & et al. (2018)	High	High	Convolutional Neural Network	5G	Average	High
		Lu & et al. (2019)	High	High	Long Short-Term Memory	5G	Average	High
		Liu *et al.* (2020)	Average	High	Convolutional Neural Network	5G	Average	High
Link Adaptation	Conventional	Judd *et al.* (2008)	High	Average	Handcrafted Algorithm	IEEE 802.11	High	Low
		Wong *et al.* (2006)	High	Low	Handcrafted Algorithm	IEEE 802.11	High	Low
		Jensen *et al.* (2010)	High	Average	Mutual Information	IEEE 802.11n	High	Low
	AI-based	Daniels & Heath (2010)	High	High	Neural Networks	IEEE 802.11	Average	High
		Yun & Caramanis (2010)	High	High	Reinforcement Learning	IEEE 802.11	Average	High
		Dong & et al. (2018)	High	High	SVM	IEEE 802.11	Low	High
Distributed Inference	Statistical	Escamilla *et al.* (2018)	Average	Average	Likelihood Ratio Test	Sensor Networks	High	Low
		Wu *et al.* (2021)	Average	Average	Likelihood Ratio Test	Sensor Networks	High	Low
		Sreekumar *et al.* (2018)	Average	Average	Likelihood Ratio Test	Sensor Network	High	Low
	AI-based	Fazai *et al.* (2019)	High	High	Neural Network Classifiers	IoT	Average	High
		Messner (2023)	High	High	Neural Network Classifiers	IoT	Average	High
		Mitra *et al.* (2020)	High	High	Neural Network Classifiers	IoT	Average	High

1.6 Conclusion

This chapter provides an overview of the state-of-the-art work on the functions addressed by this research, specifically CFO, CSI feedback, and link adaptation within modern wireless communication systems. In addition, the evolution of the work from conventional to learning-based techniques has been smoothly introduced. Through our discussion in this chapter, we have identified the limitations associated with conventional approaches employed in tackling these challenges and functions. Furthermore, we have emphasized the potential impact that learning techniques can have on enhancing the performance of these functions and, consequently, on improving overall communication systems. Subsequent chapters will delve into each of these issues individually, outlining our contributions and findings in each respective area. Each chapter will present one of our works.

CHAPTER 2

METHODOLOGIES

In this chapter, we explain the adopted methodologies in detail. Then, we iterate over the adopted ML tools. Finally, we present and explain a roadmap that connects each chapter in the thesis with the related gap, subproblem, challenges, objectives, and methodologies.

2.1 Methodologies

In this dissertation, we propose a novel learning framework for optimized wireless links for 5GB communications systems. To tackle the research questions and sub-objectives defined earlier, we followed a comparable methodology across every contribution made in this study. Consequently, this methodology applies to each research question.

1. We carried out a thorough analysis of the existing literature work and methodologies related to the research question. Through this process, we identify the advantages and limitations of each work with a special intention paid for the possible contributions and improvements.

2. Based on the findings in the first step, (especially the identified limitations, and possible improvements), we formulated the problem mathematically and defined the proposed model. Problem formulation is crucial in this work since the same problems can be formulated and solved using different learning models. For example, a given problem can be formulated as a supervised problem (e.g., classification) or a reinforcement learning problem (e.g., multiarmed bandit). This formulation has significant consequences for the performance of the proposed solution. In this step, we deeply analyze all possible formulations and define how much they fit our research question.

3. Based on the formulation defined in the previous phase, we research the possible improvements in each model. For example, we evaluate if using conventional loss functions is enough or if defining a custom loss function with problem-specific terms would be better.

4. Given the proposed improvements we defined in the previous phase, we implement our proposals. Using extensive experiments, we evaluate the performance of our proposed solutions. To get an intuition on our contribution, we compare the obtained performance with state-of-the-art methods and highlight the limitations and strengths of each one. For some research questions, custom evaluation metrics are defined to reflect the performance of our solution in the overall link/network performance. For example, in RQ3, we defined custom evaluation metrics to evaluate the performance of the proposed link adaptation algorithm in terms of packet retransmissions. Such custom-defined metrics take the evaluation beyond just AI model evaluation to a system evaluation that takes into consideration the holistic view of the overall communication system.

5. The final stage involved sharing our findings with the scholarly community. The invaluable insights garnered from the peer-review process significantly aided in refining our reports and enhancing the overall quality of our research. This stage also provided a platform to justify our decisions and engage in reflective contemplation about our solutions, helping us to discern potential limitations and avenues for further improvement.

The previous points summarize the general methodology adopted throughout the whole work in the dissertation. However, a different approach has been adopted for each research question. The following subsections elaborate on each of them in more detail.

2.1.1 M1: Accurate CFO Estimation

CFO refers to the deviation or difference between the actual transmitted carrier frequency and the expected or nominal carrier frequency in a wireless communication system. This offset can occur due to various factors, including oscillator imperfections, Doppler shift, and other environmental conditions. To address RQ1, we developed a learning framework for CFO estimation. Given the known received values of PSS and SSS, this problem can be modeled as a regression problem where a model is trained to predict the target CFO value directly from the PSS and SSS. In addition, it can be modeled as a classification problem where all possible CFO (i.e., the CFO range) values are quantized into a number of classes. A classification model is trained to predict the CFO class from the PSS and SSS values. The problem can be formulated as a reinforcement learning problem where an agent interacts with the propagation environment and it is rewarded on how much accurate is the frequency correction. Furthermore, the training approach for each of these scenarios should be analyzed to obtain outstanding results compared with conventional techniques.

For the CFO problem in NR, we adopted supervised learning modeling. We also consider the problem as a regression problem. Due to the high complexity of the relation between the CFO values and the PSS/SSS, we trained an ensemble model (i.e. gradient boosting machines GBM) for the regression task. In addition, we generated a huge dataset to span several levels of SNRs. Due to the high dimensionality of the PSS/SSS input, we proposed a dimensionality reduction technique to increase the prediction accuracy and avoid the curse-of-dimensionality problem. In chapter two, we explain the proposed solution in more detail.

2.1.2 M2: Robust Communication-Efficient CSI Feedback

The process of transmitting the CSI from the receiver back to the transmitter plays a crucial role in optimizing the performance of MIMO systems by enabling adaptive beamforming, precoding, and other advanced transmission techniques. However, with a large channel matrix, this feedback consumes huge bandwidth and time and a compression technique should be adopted to reduce the bandwidth consumption. Along with the conventional techniques (i.e., compressive sensing, vector quantization, and PCA), several works in the literature adopted autoencoder-based architectures for CSI compression. In addition, they assume noise-free feedback channels. We went beyond the assumption of noise-free feedback channels. To address the more realistic assumption of additive white-Gaussian noise (AWGN) feedback channels, we exploited the power of generative models (i.e., variational autoencoders, VAE) to generate noise-robust codewords. It is worth noting that we not only used VAE for the CSI feedback problem, but also we adapted to the VAE loss function (i.e. ELBO) to fit the problem under consideration. To avoid the limitations of autoencoder-based solutions, we exploited the bias/variance tradeoff for high accuracy and scalable CSI compression technique. In addition, instead of sticking only to the CSI compression, we developed an end-to-end framework for joint CSI compression and link adaptation. Chapters three and four present our proposed solutions in more detail.

2.1.3 M3: Reliable Link Adaptation

Link adaptation refers to the process of dynamically adjusting the transmission parameters in a communication system to optimize the quality and reliability of the wireless link. It involves adapting various parameters, such as modulation scheme, coding rate, transmit power, and channel bandwidth, based on the prevailing channel conditions and system requirements. Unlike other works in the literature that formulated the problem as a reinforcement learning problem, we proposed a novel formulation as a multi-label multi-class classification. The proposed modeling enables the models to predict all the transmission modes that maximize the throughput and minimize the bit error rate (BER). Furthermore, we proposed the first data-centric approach applied to the link adaptation problem. Instead of fine-tuning model hyperparameters (i.e., number of layers, activation, learning rate, etc.), data-centric approaches change the dataset (sampling, size, preprocessing, etc.) to increase the model accuracy. Chapters five and six present the proposed work in more detail.

2.2 AI and ML Tools

Throughout this work, we adopted and customized several ML models and concepts. The following list summarizes these models while Table. 2.1 relates them to the context in which they were adopted.

- **Gradient Boosting Machines (GBM):** is a powerful machine learning technique used for both regression and classification tasks. It builds an ensemble of decision trees sequentially, where each tree corrects the errors of the previous ones. It works by fitting each tree to the residuals (the differences between predicted and actual values) of the previous trees. GBM combines the predictions of multiple weak learners (usually shallow trees) to create a strong predictive model. It's known for its high predictive accuracy and is widely used in various applications and real-world problems.

- **Long Short-Term Memory (LSTM):** is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs. It's particularly well-suited for sequential data, such as time series or natural language processing tasks. LSTM units have memory cells and gating mechanisms that allow them to capture long-range dependencies and prevent the vanishing gradient issue. This makes them effective in modeling and predicting sequences by selectively retaining and updating information over time. LSTMs are widely used in various applications, including speech recognition, machine translation, and sentiment analysis.

- **Support Vector Regressors (SVR):** is an ML learning algorithm used for regression tasks. It aims to find a hyperplane that best fits the data points while minimizing the error or deviation from the true target values. SVR uses support vectors, which are data points closest to the hyperplane, to define the regression model. It seeks to maximize the margin around the hyperplane, and a regularization parameter is used to control the trade-off between model complexity and accuracy. SVR is effective for handling non-linear relationships in data by using kernel functions. It's widely used in applications like stock price prediction and time series forecasting.

- **Multi-Layer Perceptron (MLP):** is a type of artificial neural network composed of multiple layers of interconnected nodes (neurons). It consists of an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to every neuron in the subsequent layer, and each connection has a weight. MLPs are known for their ability to model complex non-linear relationships in data. They are used for various machine learning tasks, including classification and regression, and can learn complex patterns by adjusting the weights through training algorithms like backpropagation.

- **Deep Neural Network (DNN):** is a type of artificial neural network with multiple hidden layers between the input and output layers. These hidden layers enable a DNN to learn increasingly complex and hierarchical representations of data, making it suitable for tasks involving intricate patterns, such as image and speech recognition. Deep learning techniques involve automatic feature extraction and transformation, making them powerful for tasks like deep learning in computer vision, natural language processing, and reinforcement learning. Training deep networks typically requires substantial computational resources but can yield state-of-the-art results in various domains.

- **Variational Autoencoder (VAE):** is a type of generative model used in unsupervised learning and dimensionality reduction. It combines elements of autoencoders and probabilistic modeling. VAEs encode input data into a lower-dimensional latent space while simultaneously learning the distribution of this latent space. This allows for generating new data points that resemble the original data distribution. VAEs are known for their ability to generate diverse and high-quality samples from complex data distributions.

- **Bayesian Optimization (BO):** is a sequential model-based optimization technique used for optimizing complex, expensive-to-evaluate functions. It combines a probabilistic model (typically a Gaussian Process) to model the function and an acquisition function (such as Expected Improvement) to guide the search for the optimal input. BO is useful in scenarios like hyperparameter tuning, where it efficiently explores the parameter space, adapts to the function's behavior, and finds the optimal solution with a relatively small number of function evaluations, reducing the computational cost of optimization tasks.

- **Convolutional Neural Network (CNN):** is a class of deep learning models primarily used for tasks involving grid-like data. CNNs are designed to automatically learn hierarchical representations from input data through a series of layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply learnable filters to capture local patterns while pooling layers downsample the data. CNNs are known for their ability to capture spatial hierarchies and translation-invariant features. They have revolutionized computer vision and are used in various applications across domains.

Table 2.1 The different AI models used in this thesis and the subproblems (SP) in which they have been adopted

Model	Problems	Role
Gradient Boosting Machines (GBM)	SP1: CFO estimation	Proposed solution
Long Short-Term Memory (LSTM)	SP1: CFO estimation	Proposed solution
Support Vector Regressors (SVR)	SP1: CFO estimation	Baseline comprisons
Multilayer Perceptron (MLP)	SP1: CFO estimation	Baseline comparison
	SP2: CSI feedback compression	Proposed solution
Deep Neural Networks	SP2: CSI feedback compression	Proposed solutoin
Variational Autoencoders	SP2: CSI feedback Compression	Proposed solution
Bayesian Optimization	SP2: CSI feedback compression	Optimizing our solutions
	SP3: Link adaptation	
Convolutional Neural Networks	SP3: Link adaptation	Proposed solution

2.3 Thesis Roadmap and Connections

In this section, we illustrate how each chapter in the thesis relates to the research questions, objectives, methodologies employed, and the specific gap within the literature work that is addressed therein. Table. 2.2 maps each chapter to the related GAP, subproblem, challenges, objectives, and methodologies.

Table 2.2 A roadmap that connects each chapter with the related GAP, subproblem, challenges, objectives, and methodologies.

GAPS		
CFO GAPS	G 1.1	Relatively low accuracy of conventional techniques
	G 1.2	Low generalization capabilities for different SNR values
CSI Compression GAPS	G 2.1	Ignoring the noise in the feedback channels
	G 2.2	Limited scalability characteristics for new antenna settings
Link Adaptation GAPS	G 3.1	Dependability on a limited amount of information (i.e., CQI) and limited throughput could be achieved with high reliability
	G 3.2	Limited performance due to the decoupling between the link adaptation and the CSI compression problems

Subproblems (SP)	
SP 1	How to build a low approximation-error CFO-estimator for NR in heterogenous SNR levels?
SP 2	How to build an efficient scalable CSI feedback mechanism for noisy feedback channels?
SP 3	Given the CSI feedback, how to adapt the transmission parameters to achieve a high reliability and throughput?

Challenges	
C 1	Comprehensively explained within the confines of section 0.2.1
C 2	Comprehensively explained within the confines of section 0.2.2
C 3	Comprehensively explained within the confines of section 0.2.3

Objectives	
To develop a learning framework that maximizes the throughput and reliability of wireless systems under dynamic propagation environments. This objective is further divided into subobjectives (SOs)	
SO 1	To develop an accurate CFO-estimator for NR which is robust for different SNR levels
SO 2	To minimize overhead (bandwidth & time) of the feedback process while maintaining a high feedback accuracy
SO 3	To develop a reliable adaptation module to maximize the throughput

Methodologies	
M1	Comprehensively explained within the confines of subsection 2.1.1.
M2	Comprehensively explained within the confines of subsection 2.1.2
M3	Comprehensively explained within the confines of subsection 2.1.3

Roadmap and Interconnections					
	GAP	SP	C	SO	M
Chapter 3	G 1.2	SP 1	C 1	SO 1	M 1
Chapter 4	G 2.1	SP 2	C 2	SO 2	M 2
Chapter 5	G 2.2	SP 2	C 2	SO 2	M 2
Chapter 6	G 3.1	SP 3	C 3	SO 3	M 3
Chapter 7	G 3.1	SP 3	C 3	SO 3	M 3
Chapter 8	G 3.2	SP 3	C 3	SO 3	M 3

2.4 Conclusion

Based on the limitations identified in the literature work given in the previous chapter, this chapter introduced our adopted methodology to achieve the intended objectives by addressing the identified challenges of each subproblem. In the following chapters, we present our published/submitted research papers, one in each chapter. To facilitate the navigation through each of our work, we cluster them into parts (PART 1, PART 2, and PART 3) where each part presents the papers tackling the same subproblem.

PART 1: CFO Estimation

CHAPTER 3

CARRIER FREQUENCY OFFSET ESTIMATION IN 5G NR: INTRODUCING GRADIENT BOOSTING MACHINES

Mostafa Hussien, Ahmed Abdelmoaty, Mahmoud Elsaadany, Mohammed F. A. Ahmed,
Ghyslain Gagnon, Kim Khoa Nguyen and Mohamed Cheriet

École de technologie supérieure (ÉTS), Univeristy of Québec, Canada, H3C 1K3

Paper published in IEEE Access, March 2023

Abstract : The beyond fifth generation (B5G) communication systems imposed several challenges on radio designers. For example, a machine is required to set up a call at a low signal-to-noise ratio (SNR), as low as -10 dB, in the extended coverage mode. Moreover, only one receive antenna will be available, and virtually no frequency diversity. Such requirements present major challenges to maintaining timing and frequency synchronization. Carrier frequency offset (CFO) estimation is at the heart of these challenges. Different ways have been proposed for CFO estimation such as maximum likelihood based on a cyclic prefix. Nevertheless, these methods remain limited in various ways. At the same time, machine learning (ML) techniques showed outstanding performance in several wireless communication problems. In this work, we propose an ML-based approach for CFO estimation in OFDM systems. Specifically, we propose a gradient-boosting machine (GBM)-based solution to predict the CFO given the received primary synchronization signal (PSS) and secondary synchronization signal (SSS). Furthermore, we make our dataset available for public access to encourage other researchers to pursue this promising direction. We compare our results with different baseline models (i.e., artificial neural networks and support vector machines). The experimental results show that our model outperforms other baseline models due to its ensemble nature which enables ensemble models to obtain a better generalization behavior.

keywords :

New radio (NR), Signal synchronization, carrier frequency offset estimation, gradient boosting machines, ensemble learning

3.1 Introduction

Massive integration of connected devices with emerging services provisions a successive increase in traffic demand and higher data rates. It is expected that data rates to be exploded by deploying 5G new radio (NR). Globally, mobile data traffic is projected to margin 226 Exabytes (EB) per month in 2026. To cope with these requirements and to provide better user experience and services, it is anticipated for future communication networks (e.g., 6G) to integrate multiple advanced technologies, such as edge computing and machine learning (ML). This integration is stimulated by the increase in traffic and data requirements. Indeed, it is anticipated that the connected devices will margin 80 million by 2025 Abdelmoaty & et al. (2022).

Orthogonal frequency division multiplexing (OFDM) is one of the adopted technologies in 4G long-term evolution (LTE), and it is expected to continue supporting the 5G NR. OFDM is proven to have the ability to work in harsh fading environments due to multipath. Furthermore, relying on 5G NR and IEEE 802.11ax is provisioned to exploit quadrature-amplitude modulation (QAM) with higher orders of up to 1024. Additionally, different modulation schemes with higher orders (e.g., 64-QAM and 256-APSK) are expected to be supported by millimeter-wave (mm-Wave) technologies and satellite TV standards, respectively.

Higher-order modulation schemes are susceptible to phase errors due to their highly dense constellation mapping. These errors arise from residual carrier frequency offset (CFO) that results from an intrinsic mismatch between the transmitter and the receiver oscillators. Interestingly, it is linearly increasing during frame reception and eventually translated to a large phase offset that causes a considerable declination in spectral efficiency and increases the bit-error-rate

(BER). CFO resulting from Doppler shift may reach up to 2 KHz at 4.2 GHz band, this is equivalent to 13% of the sub-carrier spacing Siyari *et al.* (2019).

The CFO destroys the orthogonality between subcarriers and induces inter-carrier interference (ICI). Consequently, there are degradations in the OFDM system performance. Therefore, estimating the CFO is very critical for future communication networks. In general, estimating the CFO can be classified into two categories: data-driven estimation and blind estimation. The blind estimation of the CFO can be performed with algorithms such as the cyclic prefix (CP)-based maximum likelihood. Zadoff-Chu (ZC)-based cross-correlation or auto-correlation algorithm is an example of the data-driven CFO estimation Ramadan & et al. (2020).

Recently, state-of-the-art machine learning (ML) algorithms go all the way from data mining techniques, and resource allocation problems to tackle most of the issues in cellular networks. Interestingly, there is a significant trend for implementing powerful ML-based solutions for many complex problems in wireless communications such as link adaptation Hussien & et al. (2021), resource allocation Lee *et al.* (2019), CSI compression Hussien *et al.* (2022), beamformingChen & et al. (2020), among others. ML algorithms can be categorized into three main categories namely supervised, unsupervised, and reinforcement learning (RL). Basically, supervised learning requires labeled data in order to train the system. Hence, the system learns from these labels to predict the target output. Conversely, unsupervised learning does not have the luxury of accessing labeled data. The expected output is not known priorly and the system needs to learn in a blind fashion. Finally, in the RL regime, an agent learns the best actions by itself, but with enforced guidance by a reward mechanism. The actions in RL are made by the agent toward the environment, which in turn replies by changing its state and sending back a reward value depending on how good is the agent's actions.

3.2 System Model and NR Signal Synchronization

3.2.1 System Model

For a typical NR OFDM system, the received signal $r(n)$ can be represented by:

$$r(n) = y(n) + w(n) = \Big[x(n) * h(n)\Big] e^{-j2\pi\varepsilon n} + w(n) \tag{3.1}$$

where:

$x(n)$ is the transmitted signal.

$h(n)$ is the impulse response of the channel.

$w(n)$ is the additive white Gaussian noise (AWGN).

ε is the normalized CFO.

$*$ represents the convolution operator.

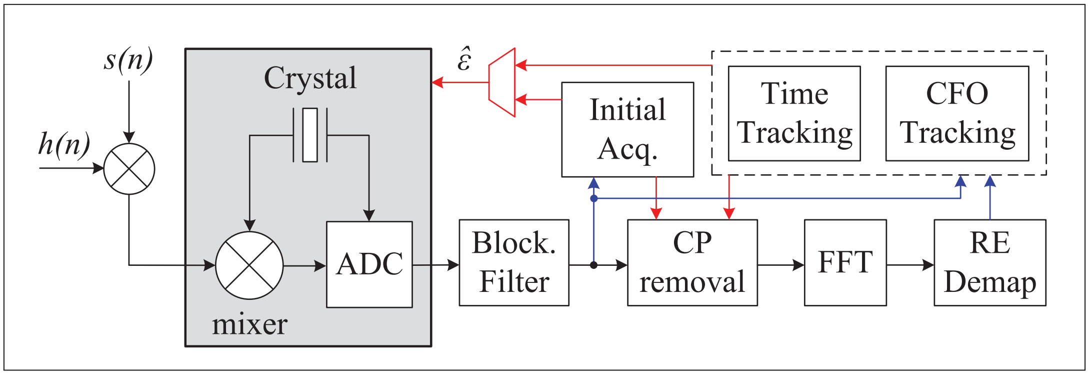


Figure 3.1 CFO tracking in NR UE receiver

Fig. 3.1 illustrates the block diagram of the processing chain of the NR user equipment (UE) receiver. We are assuming an OFDM system where a unified crystal is exploited to lock the carrier frequency to the sampling clock through the RF processing. Then, the initial acquisition phase starts by estimating the frequency/timing offsets experienced by both the RF crystal and the channel. However, a residual CFO caused by temperature changes and Doppler effects always exists and needs to be estimated. Hence, it is normal to consider the tracked CFO a fractional part of the subcarrier spacing (SCS). Lastly, the received symbols are transferred to the frequency domain (FD) using FFT as given by:

$$R_l^{(\delta)}(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r_l^{(\delta)}(n) e^{-j2\pi kn/N}, \quad 0 \leq k < N \tag{3.2}$$

where:
$r_l^{(\delta)}(n)$ is the l^{th} OFDM symbol after removing the cyclic prefix (CP). The length of the OFDM symbol is denoted by N, and δ represents the drift samples.

3.2.2 NR Signal Synchronization

Based on the 3GPP specifications release 15 3rd Generation Partnership Project (3GPP) (2020), the NR system is defined by multiple SCS and a CP overhead. CP can be either normal or extended. The basic SCS 15 KHz is used as a base for obtaining any other SCS by the scale of 2^{μ}, where $\mu \in \{0,1,2,3,4\}$. The frame structure in the time domain consists of 10 subframes with a fixed duration of 1 ms. Regardless of the CP overhead, each SCS is aligning on symbol boundaries in every subframe. The period of each time slot is equal to $1/2^{\mu}$. For each slot, there are 14 and 12 OFDM symbols for normal and extended CP, respectively. While in the frequency domain, similarly to LTE; a resource block (RB) is defined by 12 consecutive subcarriers. An RB grid in the NR system is shown in Fig. 3.2.

The procedure of signal synchronization includes cell search, frame boundaries detection, and signal quality measurements. In NR, downlink synchronization signals are classified into two types:

a) Primary Synchronization Signal (PSS):
PSS sequences can be denoted by $P_{\mu}[m]$ and composed of 127 samples of n-sequences that given by:

$$P_{\mu}[m] = 1 - 2n[(m + 43\mu) \mod 127] \text{ for } 0 \leq m \leq 126 \tag{3.3}$$

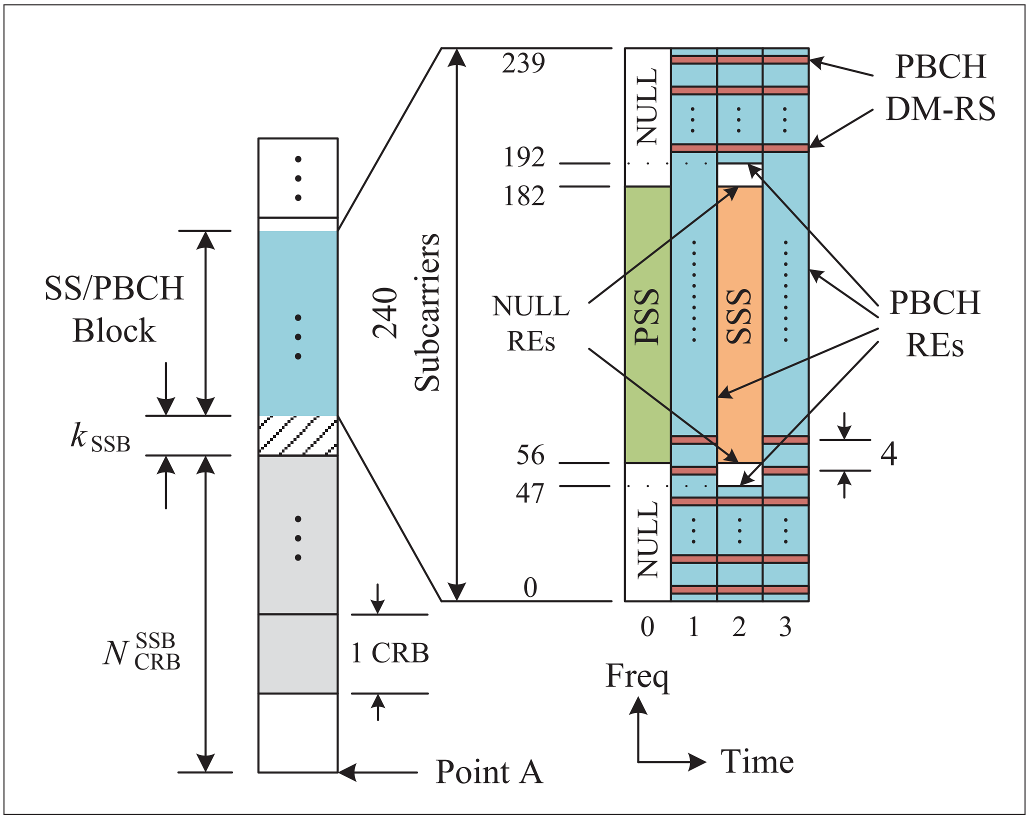


Figure 3.2 Frame structure of SS block

where $\mu \in \{0,1,2,3,4\}$ stands for cell sector ID and

$$n[m+7] = (n[m+4]+n[m]) \mod 2 \tag{3.4}$$

where the first 7 samples of n[m] can be given by $\{0,1,1,0,1,1,1\}$. In the frequency domain, PSS channel consists of 240 subcarriers, and using Eq. 3.3 it can be given as:

$$D_\mu[f] = \begin{cases} P_\mu[f-56], & 56 \le f \le 126 \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

PSS is located in the first OFDM symbol of the synchronization block and occupies subcarriers with indexes from 57 to 183.

b) Secondary Synchronization Signal (SSS):

SSS is a result of the combination of two n-sequences with a duration of 127 samples. SSS is generated depending on group ID $\in [0,355]$. SSS occupies subcarriers with the same indices as PSS but it is located in the third OFDM symbol of the signal synchronization block (SSB).

Additionally, another type of signal is mapped to the SSB, namely the physical broadcast channel (PBCH). 56 information bits representing four information fields are transmitted by the PBCH in each SSB. The first 24 bits are used for cell configuration, while the last 24 bits are reserved for cyclic redundancy check (CRC). The remaining bits are used by the UE to detect the radio frame's beginning; accordingly, it starts the procedure of synchronization.

3.3 Proposed Model

In this section, we introduce the GBM-based framework in subsection 3.3.1. The description of the dataset is presented in subsection 3.3.2.

3.3.1 Gradient Boosting Machines (GBM)

Gradient boosting machine (GBM) is a widely-adopted efficient classification and regression model. Given a dataset $\mathscr{D} = \{x_i, y_i\}_{i=1}^{N}$ that consists of N observation-label pairs (x_i, y_i). GBM iteratively constructs M weak learners (usually decision trees), $h(x,a_1), h(x,a_2), \ldots, h(x,a_M)$. Assume the labels are generated from an underlying function such that $y_i = f(x_i)$ is the true function to be approximated. A GBM model approximates $f(x)$ by a prediction function, $\hat{y} = g(x)$. The prediction function could be expressed as an additive expansion of a basis function $h(x,a_m)$ such that:

$$\begin{cases} \hat{y} = \sum_{m=1}^{M} \beta_m h(x;a_m) \\ \\ h(x;a_m) = \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}), \end{cases} \tag{3.6}$$

where $I = 1$ if $x \in R_{jm}$ and zero otherwise. The input space is divided into J different non-overlapping regions $R_{1m}, R_{2m}, \ldots, R_{Jm}$. Each decision tree predicts a constant-value γ_{jm} for a region R_{jm}. In a certain decision tree, the mean values of each splitting variable are given by the parameter a_m. The hyperparameter β_m controls the contribution of each node to the final prediction Babajide Mustapha & Saeed (2016). The values of these hyperparameters are selected to minimize a specified loss function. Specifically, the mean square error function is a typical choice for regression problems. For a better approximation with small chances of overfitting, a regularization parameter is added to the loss function.

$$\mathscr{L} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{m}^{M} \Omega(h_m) \tag{3.7}$$

The second term is a regularization term that counts for the complexity of the model to prevent overfitting. The regularization term, Ω, can be given by:

$$\Omega(h_m) = \gamma T + \frac{1}{2}\lambda ||\mathbf{w}||^2, \tag{3.8}$$

where T is the number of leaves and $\mathbf{w}$ is the vector of leaf weights. The hyperparameters γ and λ control the hardness of the regularization, and accordingly, the complexity of the model. It is worth noting that there are different techniques that can be adopted to prevent overfitting during the training phase. Column subsampling and shrinkage are two examples of such techniques Chen & Guestrin (2016).

Following the principle of *empirical risk minimization*, we train a GBM model, h_M, to minimize the empirical risk given by:

$$\mathscr{R}(h_M) = \mathbb{E}[\mathscr{L}(y, h_M(x))], \tag{3.9}$$

where $\mathscr{R}(\cdot)$ is the empirical risk of the input model and $\mathscr{L}(\cdot)$ is the adopted loss function. Given an N-points dataset, we can compute the empirical risk given in (3.9) by:

$$\mathscr{R}(h_M) = \frac{1}{N}\sum_{i=1}^{N}\mathscr{L}(y_i, h_M(x_i)) \tag{3.10}$$

Since we adopt the additive training approach to train the GBM model, the prediction of the i^{th} data point at a time step, t, is given by:

$$\hat{y}_i^{(t)} = \sum_{i=1}^{M} h(x_i; a_m) = \hat{y}_i^{(t-1)} + h^t(x_i), \tag{3.11}$$

which means that we add the t^{th} learner to minimize the objective in (3.7) in a greedily fashion. Therefore, the objective in (3.7) at the t^{th} time step can be expressed as:

$$\mathscr{L}^t = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + h^t(x_i)) + \Omega(h^t). \tag{3.12}$$

The second-order Taylor expansion can be used to get a fast optimization for the objective function.

$$\mathscr{L}^t = \sum_{i=1}^{N}[l(y_i, \hat{y}_i^{(t-1)}) + g_i h^t(x_i) + \frac{1}{2}q_i {h^t}^2(x_i)] + \Omega(h^t), \tag{3.13}$$

where $g_i = \delta_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ and $q_i = \delta^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ represent the first and second order gradient statistics on the objective function. To further simplify the objective in (3.13), we can remove the constants to reduce the objective to the following form:

$$\mathscr{L}^t = \sum_{i=1}^{N}[g_i h^t(x_i) + \frac{1}{2}q_i {h^t}^2(x_i)] + \Omega(h^t). \tag{3.14}$$

Algorithm 3.1 : GMB Training Procedure

```
Input: A training dataset, $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$;
Initialize a model with a constant value, $h_0(x) = \underset{\gamma}{argmin} \sum_{i=1}^{N} \mathscr{L}(y_i, \gamma)$;
for m=1 to M do
    Compute residuals, $r_{m,i}$ such that for $i = 1, \ldots, N$, compute:
      $r_{m,i} = -[\frac{\delta \mathscr{L}(y_i, h_{(}x_i))}{\delta h(x_i)}]_{h(x_i)=h_{m-1}(x_i)}$ ;
    Train a regression tree with features $x$ and labels $r$ and create terminal regions $R_jm$
      for $j = 1, \ldots, J_m$;
    Compute $\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \in R_{jm}} \mathscr{L}(y_i, h_{m-1}(x_i) + \gamma)$, for $j = 1, \ldots, J_m$;
    Update the model: $h_m(x) = h_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm})$
end
Return the trained ensemble model, $h_M$;
```

The regularization term can be further expanded as follows:

$$\begin{aligned} \mathscr{L}^t &= \sum_{i=1}^{N} [g_i h^t(x_i) + \frac{1}{2} q_i {h^t}^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2, \\ &= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} q_i + \lambda) w_j^2] + \gamma T, \end{aligned} \tag{3.15}$$

where $I_j = \{i | z(x_i) = j\}$ is the instance set of all leaf nodes. For a certain structure $z(x)$, the optimal value for a leaf j is denoted by w_j^* and given by:

$$w_j^* = -\frac{G_j}{Q_j + \lambda}, \tag{3.16}$$

where $G_j = \sum_{i \in I_j} g_i$ and $Q_j = \sum_{i \in I_j} q_i$. We can calculate the corresponding optimal objective by:

$$\tilde{\mathscr{L}}^t(z) = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{Q_j + \lambda} + \gamma T, \tag{3.17}$$

where the equation in (3.17) can be used to measure the quality of a tree structure z. Algorithm 3.1 describes the training process of a GBM model.

3.3.2 Dataset Description

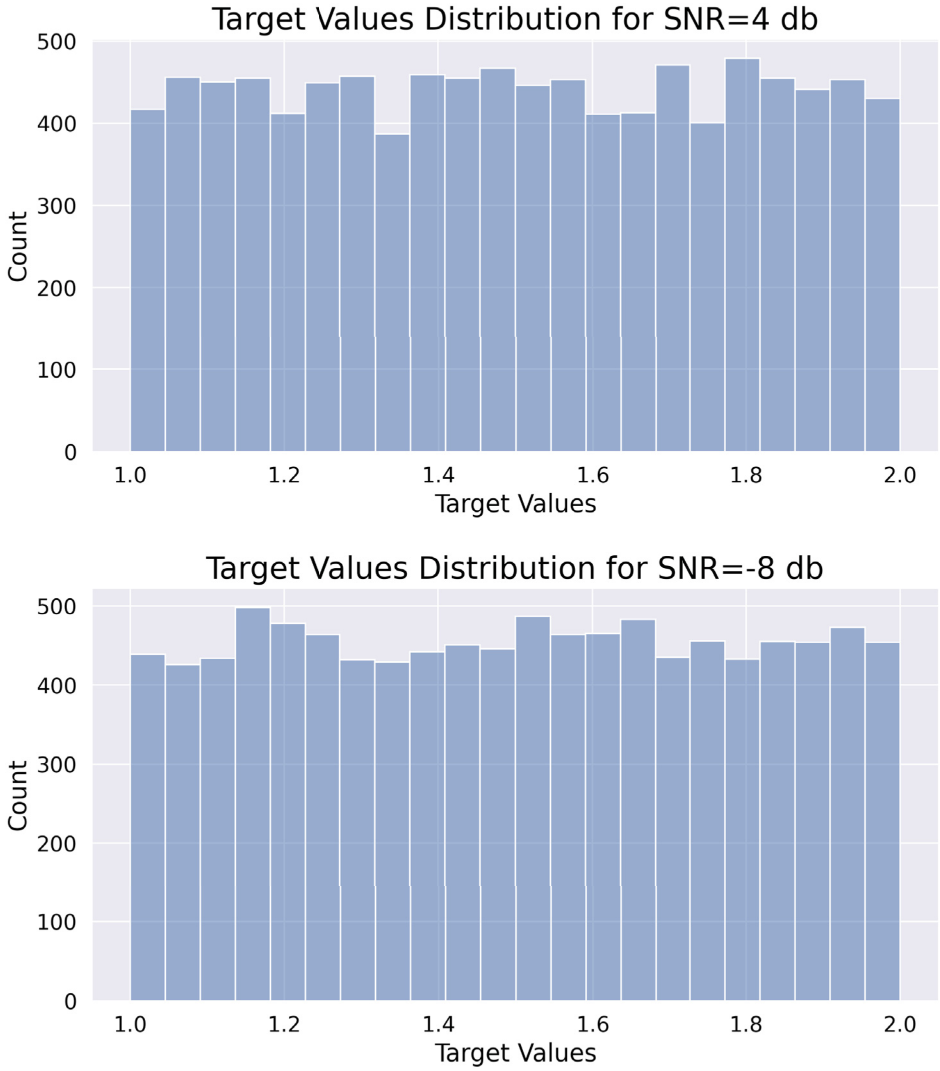


Figure 3.3 The target distribution for SNR=4 and SNR=-8. We can see that the target follows a uniform distribution

It is of crucial importance to have a large and well-annotated dataset to build any predictive model. Although it is at the heart of any communication system, this standardized publicly available dataset is not available for the problem of CFO estimation. This dataset is presented here in order to stimulate further studies in this area. We built a dataset to cover a wide range of SNRs, namely, from $SNR = -10$ db to $SNR = 10$ db with a step of 2 db. For each SNR$\in \{-10, -8, \ldots, 10\}$, we generated uniformly distributed CFO. We generated a file for each SNR separately. This facilitates designing a model for each SNR value. A larger collection of randomly generated CFO with different SNR values has been also generated that could be used for training a global model that predicts the CFO for any SNR value. The data has been formatted as comma-separated values (CSV) files. Each row in any file consists of 509 columns representing the real and imaginary parts of the PSS (127×2), the real and imaginary parts of the SSS (127×2), and the final column represents the target CFO value to be predicted.

Table 3.1 The description of dataset files

File Name	**File Size (MByte)**	**SNR Value (db)**
(1)_SNR=-10	90.4	-10
(2)_SNR=-8	90.8	-8
(3)_SNR=-6	91.2	-6
(4)_SNR=-4	91.6	-4
(5)_SNR=-2	92	-2
(6)_SNR=0	92.3	0
(7)_SNR=2	92.6	2
(8)_SNR=4	90.2	4
(9)_SNR=6	92.9	6
(10)_SNR=8	93	8
(11)_SNR=10	11.5	10

To motivate the community to further explore this interesting research direction, we released the dataset as an open-source at the *GitHub* repository of this work[1]. The dataset consists of 12 CSV files. Each SNR value has one file as well as one file for the aggregated data from different SNR. Providing the data of each SNR separately helps in building ensemble models

[1] https://github.com/Mostafa-Korashy/ML-based-Frequency-Offset-Estimation-in-NR

with the help of a multiplexer to select the model corresponding to the estimated SNR. Table 3.1 shows the details of each file.

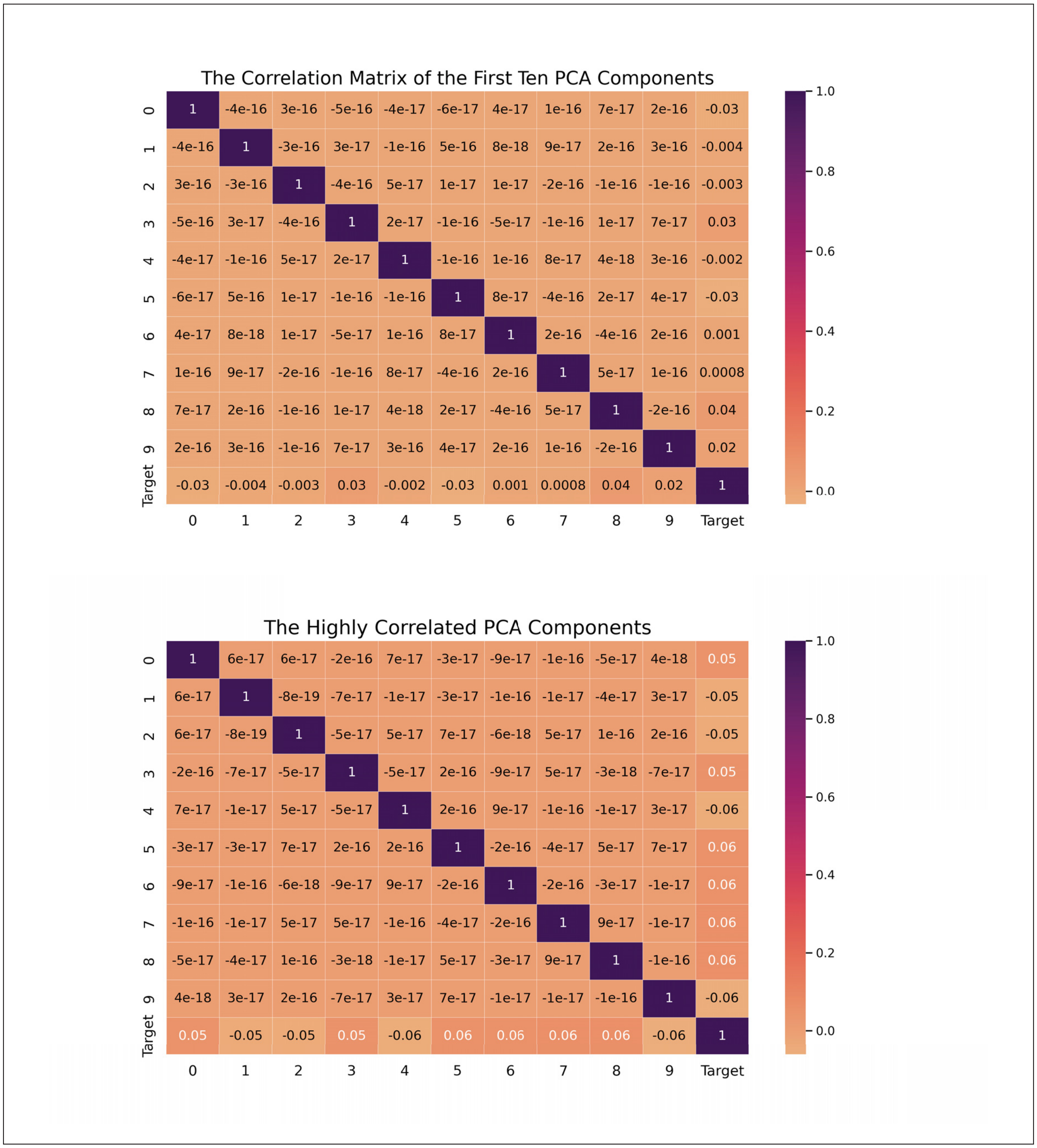
3.4 Results and Discussion

3.4.1 Dimensionality Reduction

The curse of dimensionality is one of the most common problems that raise from dealing with high-dimensional data. Sampling from a high-dimensional space makes the data sparse. Consequently, deriving important conclusions from such a sparse sample becomes more challenging. Unsurprisingly, the curse of dimensionality is presented in the problem under investigation de Bodt & et al. (2018). Different dimensionality reduction techniques can be adopted in this problem such as principal component analysis (PCA), singular value decomposition (SVD), autoencoders, etc. Among those techniques, autoencoder is a nonlinear neural network-based technique that achieved outstanding performance in the prior art. However, we employed a PCA-based dimensionality reduction technique in our study for the sake of simplicity and explainability.

PCA is a statistical procedure introduced by *Karl Pearson* in his pioneering paper Pearson (1901), that uses an orthogonal transformation to convert a group of correlated variables into a group of uncorrelated variables Reddy & et al. (2020). It has been widely used for many applications such as visualizing high-dimensional data, and dimensionality reduction for downstream tasks (e.g. regression or classification). Algorithm 3.2 summarizes the steps of PCA.

A typical way of adopting PCA for dimensionality reduction is to use the first k components for the downstream tasks. When we analyzed the correlation of the first k components and the target value, we figured out that some later components maintain higher Pearson factors than the early components. This implies that these later components could be more relevant for the downstream task than the early ones. Therefore, we perform the PCA analysis using the highest possible dimension (i.e., the same dimension as the input). Then, we analyze the correlation



The Correlation Matrix of the First Ten PCA Components

	0	1	2	3	4	5	6	7	8	9	Target
0	1	-4e-16	3e-16	-5e-16	-4e-17	-6e-17	4e-17	1e-16	7e-17	2e-16	-0.03
1	-4e-16	1	-3e-16	3e-17	-1e-16	5e-16	8e-18	9e-17	2e-16	3e-16	-0.004
2	3e-16	-3e-16	1	-4e-16	5e-17	1e-17	1e-17	-2e-16	-1e-16	-1e-16	-0.003
3	-5e-16	3e-17	-4e-16	1	2e-17	-1e-16	-5e-17	-1e-16	1e-17	7e-17	0.03
4	-4e-17	-1e-16	5e-17	2e-17	1	-1e-16	1e-16	8e-17	4e-18	3e-16	-0.002
5	-6e-17	5e-16	1e-17	-1e-16	-1e-16	1	8e-17	-4e-16	2e-17	4e-17	-0.03
6	4e-17	8e-18	1e-17	-5e-17	1e-16	8e-17	1	2e-16	-4e-16	2e-16	0.001
7	1e-16	9e-17	-2e-16	-1e-16	8e-17	-4e-16	2e-16	1	5e-17	1e-16	0.0008
8	7e-17	2e-16	-1e-16	1e-17	4e-18	2e-17	-4e-16	5e-17	1	-2e-16	0.04
9	2e-16	3e-16	-1e-16	7e-17	3e-16	4e-17	2e-16	1e-16	-2e-16	1	0.02
Target	-0.03	-0.004	-0.003	0.03	-0.002	-0.03	0.001	0.0008	0.04	0.02	1

The Highly Correlated PCA Components

	0	1	2	3	4	5	6	7	8	9	Target
0	1	6e-17	6e-17	-2e-16	7e-17	-3e-17	-9e-17	-1e-16	-5e-17	4e-18	0.05
1	6e-17	1	-8e-19	-7e-17	-1e-17	-3e-17	-1e-16	-1e-17	-4e-17	3e-17	-0.05
2	6e-17	-8e-19	1	-5e-17	5e-17	7e-17	-6e-18	5e-17	1e-16	2e-16	-0.05
3	-2e-16	-7e-17	-5e-17	1	-5e-17	2e-16	-9e-17	5e-17	-3e-18	-7e-17	0.05
4	7e-17	-1e-17	5e-17	-5e-17	1	2e-16	9e-17	-1e-16	-1e-17	3e-17	-0.06
5	-3e-17	-3e-17	7e-17	2e-16	2e-16	1	-2e-16	-4e-17	5e-17	7e-17	0.06
6	-9e-17	-1e-16	-6e-18	-9e-17	9e-17	-2e-16	1	-2e-16	-3e-17	-1e-17	0.06
7	-1e-16	-1e-17	5e-17	5e-17	-1e-16	-4e-17	-2e-16	1	9e-17	-1e-17	0.06
8	-5e-17	-4e-17	1e-16	-3e-18	-1e-17	5e-17	-3e-17	9e-17	1	-1e-16	0.06
9	4e-18	3e-17	2e-16	-7e-17	3e-17	7e-17	-1e-17	-1e-17	-1e-16	1	-0.06
Target	0.05	-0.05	-0.05	0.05	-0.06	0.06	0.06	0.06	0.06	-0.06	1

Figure 3.4 The correlation matrix of the first ten PCA components

between the target and all PCA components using the Pearson matrix. Finally, we consider the PCA components that show the highest correlation with the target label. Fig. 3.4 shows the correlation matrix for both the first ten PCA components and the ten PCA components with

Algorithm 3.2 : Dimensionality Reduction using PCA for Carrier Frequency Estimation

```
Input: A training dataset, $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$. Each point, $X_i$ is an n-dimensional vector
    such that $x_i = [x_i^{(1)}, x_i = [x_i^{(2)}, \ldots, x_i = [x_i^{(n)}]$;
Standardize the raw data: $x_i^{(j)} = \frac{x_i^{(j)} - \bar{x}^{(j)}}{\sigma^j} \forall j$;
Calculate the covariance matrix for the standardized data: $\sum = \frac{1}{N} \sum_1^N (x_i)(x_i)^T$, where
    $\sum \in \mathbb{R}^{n \times n}$;
Compute the eigenvectors and eigenvalues of the covariance matrix, $\sum$;
```

the highest correlation factors. The top figure shows the correlation matrix of the first ten PCA components where we can see the 8^{th} component has a much higher correlation factor than the 2^{nd} component. This motivated us to consider the 15 PCA components with the highest Pearson factors among the 508 components resulting from the PCA analysis.

3.4.2 Prediction Accuracy

We trained a GBM to predict the CFO. For each SNR value, we trained a GBM model using 80% of the data. The remaining 20% has been used for testing. The training set is further divided into train and validation sets. Fig. 3.5 shows the prediction accuracy for a 100-point sample from the test set. We can see that the GBM model is capable of predicting the target CFO with a considerable level of accuracy. Again, this can be attributed to the power of the ensemble model and the benefit of the adopted boosting technique.

To illustrate the distribution of prediction errors, we plot the histogram of the error bins for different SNR values. For example, we plot the histogram of the prediction error as shown in Fig. 3.6. We can see that most of the prediction errors lie in the early bins that correspond to the smaller error bins. Note that we normalize our target to be in the range of $[2, 3]$ which means the early error bins have a small percentage compared with the target values.

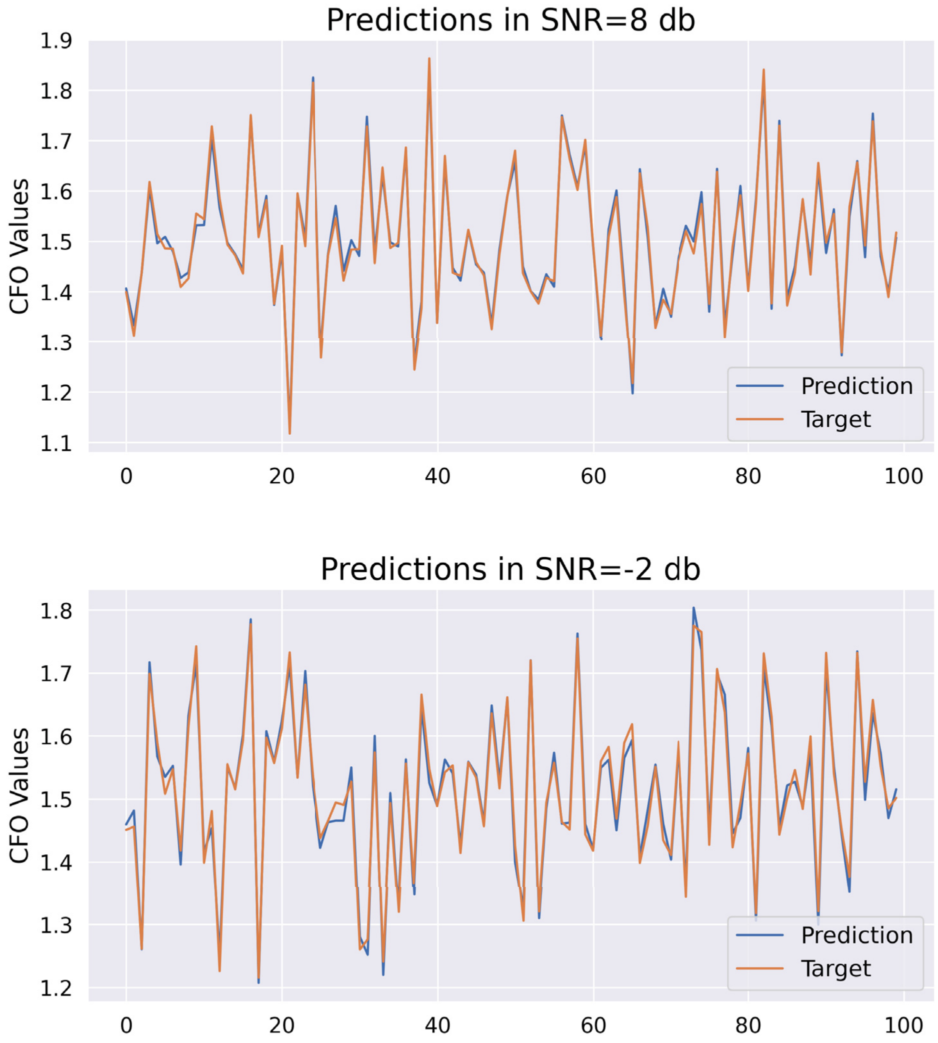


Figure 3.5 The prediction of the CFO using GBM

3.4.3 Baseline Comparisons

In this section, we compare our proposed GBM model with two widely adopted models for regression, namely artificial neural networks (ANN) and support vector machines (SVM).

Artificial neural network (ANN): ANN models have shown outstanding performance in many fields and problems such as computer vision, link adaptation, CSI compression, etc Schmid-

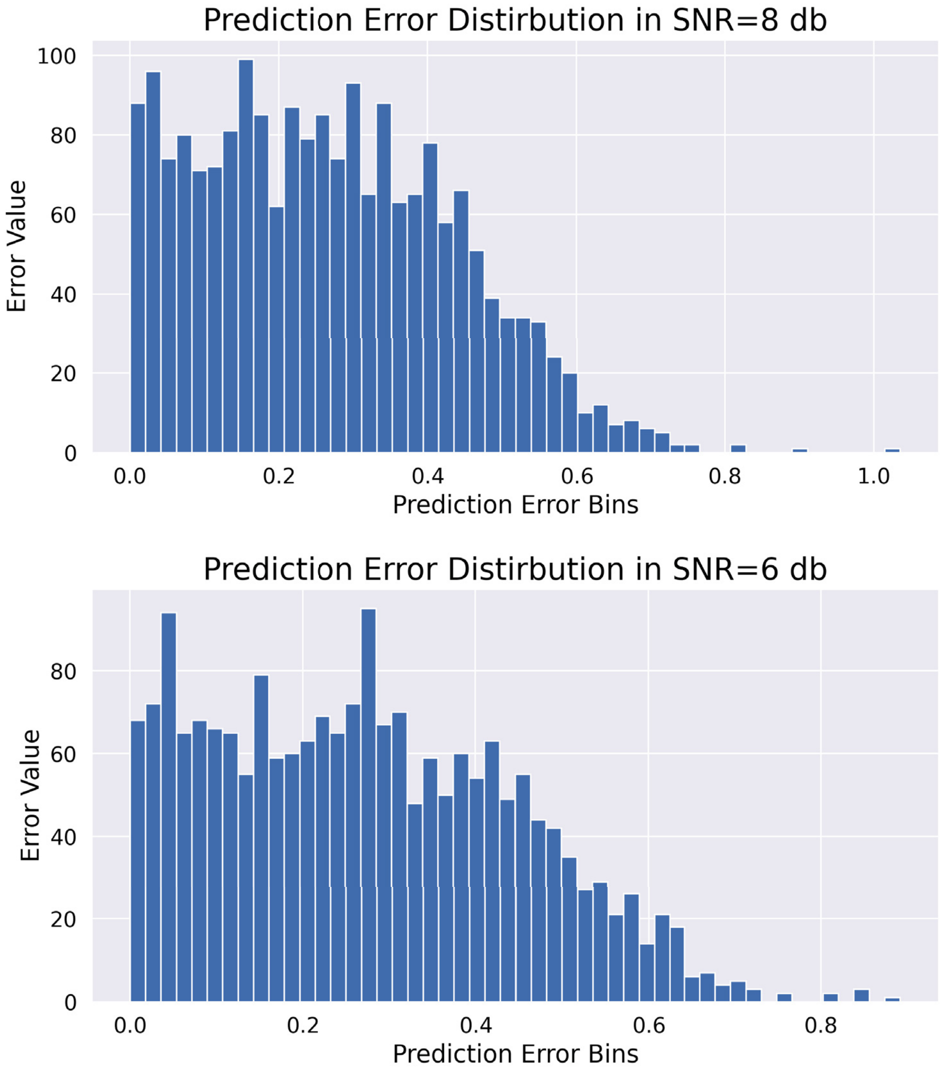


Figure 3.6 The histogram of the prediction error for SNR=8 and SNR=6

huber (2015). Multilayer perceptron (MLP) is one of the widely adopted models, especially for processing tabular data such as the problem under consideration. In this work, we used an architecture that consists of one input layer, one hidden layer with *ReLU* activation, and one output layer with *linear* activation. The model is trained to minimize the mean square error (8.6) using the *Adam* optimizer Kingma & Ba (2014) with a learning rate, $lr = 0.1$. We set the batch size to 128. An $l2$ regularization has been adopted to prevent the high-bias/low-variance behavior (overfitting). The model has been trained for 500 epochs.

$$\mathscr{L} = \frac{1}{N} \|y - \hat{y}\|_2 \tag{3.18}$$

Support Vector Machines (SVM): For several decades, SVMs dominated most predictive tasks due to their powerful modeling capabilities and inherited simplicity and explainability. In the literature, the term SVM has been used to refer to a classification model, while support vector regressor (SVR) referred to regression models. In this work, we use the term *SVM* to denote the adopted regression model. SVM is a margin maximization technique that assumes the data is linearly separable. We adopt a kernelized version of SVM since the linear separability assumption is not guaranteed in practice, especially in our dataset that exhibits a high degree of nonlinearity. Several kernels can be adopted such as Polynomial Kernel, Gaussian Kernel, Radial Basis Function (RBF), Laplace RBF Kernel, Sigmoid Kernel, and Anove RBF Kernel Nguyen (2017). In this work, we adopted an RBF kernel (3.19).

$$\mathscr{K}(x,y) = \exp\left[\frac{\|x-y\|^2}{2\sigma^2}\right] \tag{3.19}$$

The prediction results of the GBM and the two baselines (i.e., ANN, SVM) are shown in Fig. 3.7. We can see that the prediction accuracy of our proposed GBM outperforms the predictions of the other two models. We can see the predictions of the ANN model are closer to the target compared with the predictions of the SVM model. However, the predictions of the GBM are the closest to the target values. This can be attributed to the ensemble nature of the GBM model. Many weak learners can perform better than a single stronger learner. Ensemble models exploit multiple weak learners to produce weak predictions based on features extracted through various data projections. The produced results are then fused with any voting mechanisms to achieve better performances than that obtained by any standalone learner. This gives the GBM an extra advantage over the other two models Dong & et al. (2020).

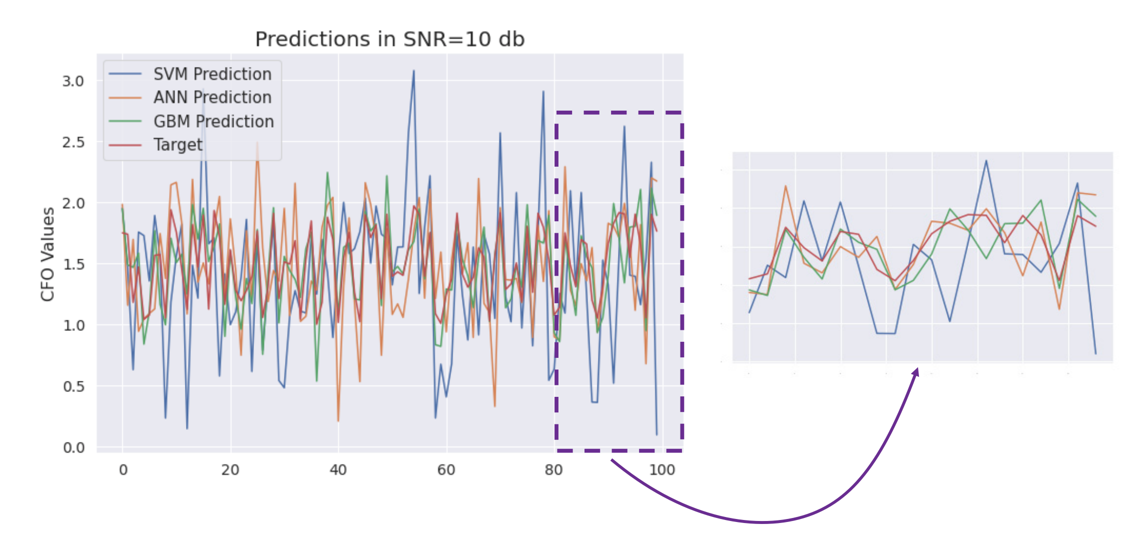


Figure 3.7 The prediction accuracy of our proposed GBM with two baseline models (i.e., ANN and SVR)

3.5 Future Work

As shown previously, the adoption of ML techniques for learning the CFO is a promising direction that can bring plenty of advantages, especially for B5G communication systems. However, challenges such as dimensionality reduction, hyperparameter optimization, and building universal predictive models require further investigation. In this section, we propose various research directions for future exploration such as:

- In this work, we proposed and validated the use of PCA as a dimensionality reduction technique. Other dimensionality reduction techniques are worth exploring. Specifically, neural network architectures such as autoencoders have been widely used for dimensionality reduction and they showed an outstanding performance in this regard. Additionally, they are capable of capturing the nonlinearities inherent in communication systems through nonlinear dimensionality reduction.

- Towards the goal of zero-touch network management, Fine-tuning the different hyperparameter values to obtain the best model becomes a challenging task, especially when we need to retrain the model (e.g., after data drift). We proposed Bayesian optimization as a solution for our work. The authors believe that more contributions in this area will be appreciated.

- A dedicated model for each SNR value has been proposed in this work. Another direction that looks more appealing is the use of a universal predictive model that can be applied to all SNR values. This solution is of more interest, and more challenging as well.
- In order to avoid wasting resources training models from scratch for each new device being installed in a new environment, transfer learning techniques can be utilized and optimized to reduce the required resources for training CFO predictive models. The impact of such techniques on the performance of the CFO predictive models should be evaluated.
- The problem can be extended to span different channel models (slow versus fast-fading, etc.) and different numerology settings.

We believe that releasing our dataset for public access can encourage other researchers to investigate more in these directions, among others.

3.6 Conclusion

In this work, we proposed a machine-learning approach for carrier frequency offset (CFO) estimation using gradient-boosting machines (GBM). Compared with various baseline models, our proposed model achieved a competitive performance in terms of prediction accuracy. Moreover, we released our dataset as open source to motivate other researchers to continue investigating data-driven solutions for CFO. We also proposed several promising research directions for further investigating the feasibility of data-driven CFO for new radio.

PART 2: CSI Feedback Compression

CHAPTER 4

PRVNET: A NOVEL PARTIALLY-REGULARIZED VARIATIONAL AUTOENCODERS FOR MASSIVE MIMO CSI FEEDBACK

Mostafa Hussien, Kim Khoa Nguyen, and Mohamed Cheriet

École de technologie supérieure (ÉTS), Univeristy of Québec, Canada, H3C 1K3

Paper published in IEEE Wireless Communications and Networking Conference (WCNC), April 2022

Abstract: In a multiple-input multiple-output frequency-division duplexing (MIMO-FDD) system, the user equipment (UE) sends the downlink channel state information (CSI) to the base station to report the link status. Due to the complexity of MIMO systems, the overhead incurred in sending this information negatively affects the system bandwidth. Although this problem has been widely considered in the literature, prior work generally assumes an ideal feedback channel. In this paper, we introduce PRVNet, a neural network architecture inspired by variational autoencoders (VAE) to compress the CSI matrix before sending it back to the base station under noisy channel conditions. Moreover, we propose a customized loss function that best suits the special characteristics of the problem being addressed. We also introduce an additional regularization hyperparameter for the learning objective, which is crucial for achieving competitive performance. In addition, we provide an efficient way to tune this hyperparameter using KL-annealing. Experimental results show the proposed model outperforms the benchmark models including two deep learning-based models in a noise-free feedback channel assumption. In addition, the proposed model achieves an outstanding performance under different noise levels for additive white Gaussian noise feedback channels.

keywords :

Generative models, variational autoencoders, CSI compression, MIMO-FDD

4.1 Introduction

Multiple-input multiple-output (MIMO) system is considered a key enabling technology for fifth-generation, 5G, wireless systems. One of the active research areas in MIMO systems is channel state information (CSI) compression for feedback. In modern MIMO systems, a base station (BS) can be equipped with several antennas to reduce the multiuser interference and increase the cell throughput. In this setting, the BS is required to perform the precoding at its side. Therefore, the BS should have access to the current CSI. In time division duplexing (TDD) systems, the downlink CSI can be estimated using any channel estimation technique at the BS side. This is achievable because channel reciprocity holds for TDD systems, (Fig. 4.1). However, in frequency division duplexing (FDD) systems, the uplink and downlink channels use different frequency bands, so channel reciprocity no longer holds. In this case, the downlink CSI should be sent to the BS by the user equipment (UE). In modern MIMO systems, this channel matrix is huge, and the bandwidth overhead incurred for sending this matrix can heavily degrade the system's performance.

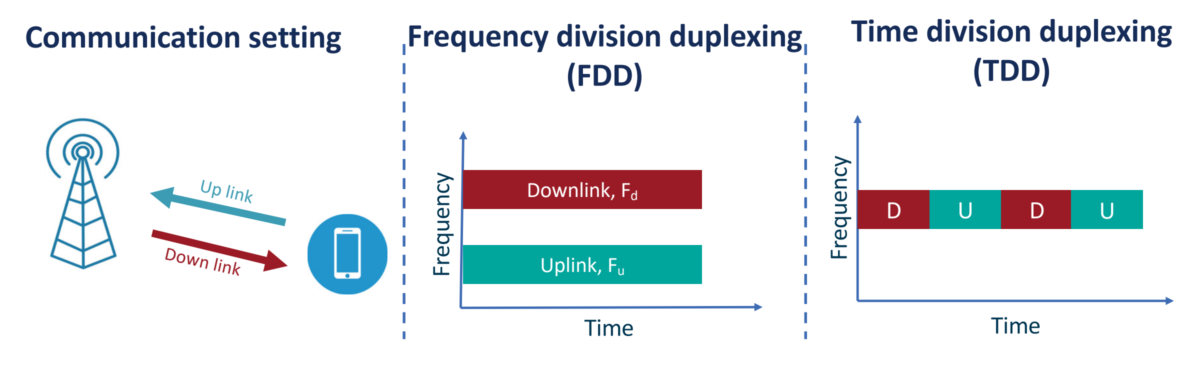


Figure 4.1 Channel reciprocity in TDD and FDD systems

To alleviate this problem, the UE can compress the CSI matrix before sending it back to the BS. The compressed CSI matrix, however, should maintain enough information about the original CSI matrix in order for the BS to accurately reconstruct the original CSI matrix. Obtaining a low-fidelity reconstruction for the original CSI matrix results in poor system performance. The problem then is how to optimally compress the CSI matrix while preserving its salient features and information. At the same time, the compression/decompression processes should be completed in real-time. Moreover, the encoder part should not consume much space or power since it resides at the UE which might have limited space and power resources. Lastly, and more importantly, this compressed data will be sent to the BS over a wireless channel (uplink channel) which suffers from traditional wireless transmission impairments such as noise, fading, or path loss. How the compression technique is resilient against the varying noise conditions of the uplink channel adds one more challenge to the CSI feedback problem.

The problem of compressing CSI feedback has been considered in the literature. Traditional methods Kuo & et al. (2012); Lu & et al. (2015) considered compressive sensing (CS) technique to compress the CSI matrix before feeding it back to the BS. However, the CSI matrix should maintain a high degree of sparsity for these methods to work. This condition is not always guaranteed in complex communication systems. In addition, many of these techniques depend on iterative approaches for solving a system of equations which makes them suffer a relatively slow performance.

On the other hand, artificial intelligence (AI) and deep learning (DL) have shown outstanding performance in solving different complex problems in wireless communications Wang & et al. (2017); Hussien & et al. (2021). A line of work has utilized AI and DL techniques to solve the CSI feedback problem. The authors in Wen & et al. (2018) opened the door for applying DL techniques in the CSI feedback problem. They presented *CsiNet*, a convolutional neural network architecture with skip connections in the decoder part. The advantage of CsiNet performance has been proven against traditional CS-based techniques. However, CsiNet employs a point estimation architecture in which the model learns one scalar value for each dimension in the codeword. This results in noise-sensitive codewords, and any noise level can largely hurt the reconstruction fidelity at the BS. Unlike CsiNet, our proposed model in this paper approximates distribution parameters for each dimension. In particular, mean and variance for a Gaussian latent space. This makes our codewords more robust against noises, and the decoder has the capacity to reconstruct the received codewords even in the presence of relatively large noise levels.

The authors in Lu & et al. (2019) exploited the temporal and frequency correlations of wireless channels. They presented *CSINet-LSTM*, which extends CsiNet with long short-term memory (LSTM) network. LSTM is a classic type of recurrent neural network capable of capturing long-term dependencies (temporal correlation) between input sequences. In Lu *et al.* (2020), the authors proposed a neural network architecture, called *CRNet*, for multi-resolution CSI feedback in massive MIMO. Their model achieves better performance than classical CS-based techniques as well as CsiNet. Another extension to CsiNet called *CsiNet+* has been introduced in Guo & et al. (2020). However, the floating point operations (flops) in CSINet+ is much larger than CsiNet, therefore the improvements come at the cost of complexity.

In general, the limitations of prior work consist of a) most prior work assumes an ideal control channel and pays less attention to the more practical assumption of noisy feedback channels, and b) no prior work has deeply investigated the power of generative models, especially variational autoencoders (VAE) Kingma & Welling (2013), in the context of CSI compression despite their proven performance in many applications.

In this work, we propose a VAE-based framework for CSI feedback compression in MIMO-OFDM systems. The proposed framework customizes the VAE loss function to suit the special characteristics of the CSI feedback problem while, at the same time, benefiting from the robustness of the VAE-generated codewords against noise. The main contributions of this work can be summarized as follows:

- A novel partially-regularized VAE model, named PRVNet, for CSI feedback problem with a new objective function that reflects the specific characteristics of CSI compression.
- A seminal algorithm inspired by Kullback-Leibler (KL)-annealing to fine-tune the additional hyperparameter introduced in the objective function.
- We consider the CSI feedback in an additive white Gaussian noise (AWGN) channel. Since we employ a distribution-estimation model, our proposed model is shown to be capable of reconstructing the CSI matrices with high accuracy even under high noise levels.

The rest of this paper is organized as follows: in section 4.2, we present the system model. Section 4.3 presents a detailed description of the proposed PRVNet model, its architecture, and the training algorithm. The results of the proposed model along with comparisons against state-of-the-art works are presented in Section 4.4, followed by a conclusion.

4.2 System Model

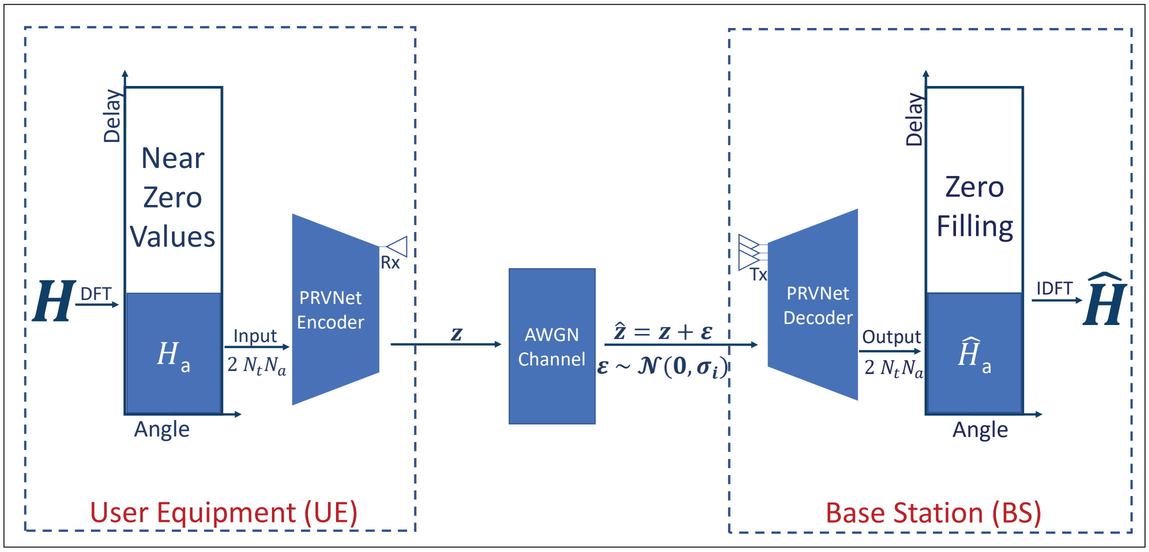


Figure 4.2 An overview of the PRVNet for CSI feedback compression through an AWGN channel

We consider a simple single-cell downlink massive MIMO system with $N_t \gg 1$ transmit antennas at the BS and a single receive antenna at the UE. The system employs OFDM with $\tilde{N}_c$ subcarriers. The received signal at the n_{th} subcarrier, y_n, is given by:

$$y_n = \tilde{\mathbf{h}}_n^{\mathbf{H}} \mathbf{v_n} x_n + z_n, \tag{4.1}$$

where $\tilde{\mathbf{h}}_n \in \mathbb{C}^{N_t \times 1}$, $\mathbf{v}_n \in \mathbb{C}^{N_t \times 1}$, $x_n \in \mathbb{C}$, and $z_n \in \mathbb{C}$ denote the channel vector, precoding vector, data-bearing symbol, and additive noise of the n_{th} subcarrier, respectively. Also, assume $\tilde{\mathbf{H}} = \left[\tilde{\mathbf{h}}_1 \ldots \tilde{\mathbf{h}}_{\tilde{N}_c}\right] \in \mathbb{C}^{\tilde{N}_c \times N_t}$ be the CSI stacked in the spatial frequency domain. The BS can design the precoding vectors $\{\mathbf{v}_n,\ n = 1, \ldots \tilde{N}_c\}$ once it receives $\tilde{\mathbf{H}}$ feedback.

In FDD systems, the BS continually receives the channel matrix, $\tilde{\mathbf{H}}$, through feedback links. This feedback has an $N_t \times \tilde{N}_c$ dimension. Estimating this channel at the UE side is out of the scope of this work. We assume that a perfect CSI has been acquired through pilot-based training Choi *et al.* (2014) and this work focuses on the feedback scheme.

To reduce the feedback overhead, we propose that $\tilde{\mathbf{H}}$ can be sparsified in the angular-delay domain using a 2D discrete Fourier transform (DFT) as follows:

$$\mathbf{H} = \mathbf{F_d}\tilde{\mathbf{H}}\mathbf{F_a^H}, \tag{4.2}$$

where $\mathbf{F_d}$ and $\mathbf{F_a^H}$ are $\tilde{N}_c \times \tilde{N}_c$ and $N_t \times N_t$ DFT matrices, respectively. Only a small fraction of the elements of $\mathbf{H}$ are large components, and the remainders are close to zero. In the delay domain, only the first N_a rows of $\mathbf{H}$ contain values because the time delay between multipath arrivals lies within a limited period. Therefore, we can retain the first N_a rows of $\mathbf{H}$ and ignore the remaining. We will use $\mathbf{H}_a$ to denote the $N_a \times N_t$ truncated matrix. The dimension of the channel matrix then reduces to $2N_aN_t$, which remains a large number in the massive MIMO regime. For classical CS-based methods, $\mathbf{H}_a$ is sparse enough when $N_t \to \infty$, in other words, $\mathbf{H}_a$ does not meet the sparsity requirement with the limited N_t.

We are interested in designing an encoder:

$$\mathbf{s} = F_{enc}(\mathbf{H_a}), \tag{4.3}$$

which can transform the channel matrix into an M-dimensional vector (codeword), where $M < N$ and $N = 2N_aN_t$. In this case, we can define the data compression ratio, $\gamma = M/N$. In addition, we have to design the inverse transformation (decoder) from the codeword to the original channel matrix such that:

$$\hat{\mathbf{H}}_a = F_{dec}(\mathbf{s}). \tag{4.4}$$

The CSI feedback approach works as follows. Once the channel matrix $\tilde{\mathbf{H}}$ is acquired at the UE side, we perform 2D DFT in (4.2) to obtain the truncated matrix $\mathbf{H}_a$ and then use the encoder

(4.3) to generate a codeword $\mathbf{s}$. The generated code word, $\mathbf{s}$, is sent to the BS over an AWGN control channel. The BS receives a noisy version of the codeword denoted by $\hat{\mathbf{s}}$ such that:

$$\hat{\mathbf{s}} = \mathbf{s} + \mathbf{z}, \tag{4.5}$$

where $\mathbf{z}$ is a noise vector sampled from a standard Gaussian. Then the BS uses the decoder (4.4) to obtain an approximation for the truncated channel matrix $\hat{\mathbf{H}}_a$. The final channel matrix in the spatial-frequency domain can be obtained by performing an inverse DFT as depicted in Fig. 4.2.

4.3 Proposed PRVNet for CSI Feedback

In the following sections, we refer to a set of CSI channel matrices as a dataset X consisting of C different CSI matrices indexed by $c \in \{1, 2, \ldots C\}$.

4.3.1 Variational Autoencoders (VAE)

The VAE consists of two models, namely encoder and decoder models. These models are trained jointly to maximize the standard VAE objective in (4.6).

$$\mathscr{L}(x, \phi, \theta) = \mathbb{E}_{z \sim q_{(z|x)}} \left[log\, p_\theta(x|z) \right] - KL(q_\phi(z|x) || p(z)), \tag{4.6}$$

where x is the input, z is the latent code, ϕ and θ are the encoder and decoder parameters, respectively. The output of the encoder model, also known as the inference model, is given by:

$$f_\phi(x_c) \equiv [\mu_\phi(x_c), \sigma_\phi(x_c)] \in \mathbb{R}^{2K} \tag{4.7}$$

where the non-linear function $f_\phi(\cdot)$ is a neural network with parameters ϕ. Both $\mu_\phi(x_c)$ and $\sigma_\phi(x_c)$ are K-dimensional vectors representing the mean and variance of a Gaussian distribution. The latent representation, code word, $\mathbf{z}_c$ is a K-dimensional vector sampled from this distribution such that:

$$q_{\phi}(z_c|x_c) = \mathcal{N}(\mu_{\phi}(x_c), diag\{\sigma_{\phi}^{2}(x_c)\}). \tag{4.8}$$

That is, for each data point, x_c, in the dataset, the inference model outputs the corresponding variational parameters of a variational distribution, $q_{\phi}(z_c|x_c)$. When optimized, this distribution approximates the intractable posterior $p(z_c|x_c)$.

The decoder model, also known as the generative model, takes the sampled codeword as input. It uses this codeword, $\mathbf{z}_c$, to reconstruct the original input, x_c using a nonlinear function $p_{\theta}(x_c|z_c)$. The model is then trained to maximize the function given by (4.6) in an end-to-end fashion.

The first term in (4.6) represents the reconstruction loss between the original input and its reconstructed image. While the second term represents the KL divergence between the encoder's distribution, $q_{\phi}(z|x)$, and the true distribution, $p(z)$. This divergence measures how much information is lost when using q to represent a prior over $\mathbf{z}$ and encourages its values to follow a Gaussian distribution. Since the function in (4.6) is a lower bound for the log marginal likelihood, it is referred to as the evidence lower bound (ELBO) function. We can note that ELBO is a function in both ϕ and θ.

Taxonomy of Autoencoders

Variational autoencoders are generative models that learn a latent representation for the input data. Unlike classic autoencoders which employ a deterministic latent space (i.e., estimating a point for each dimension in the latent space), VAE employs a stochastic latent space (i.e., samples form a tractable distribution usually assumed to be a Gaussian distribution) Hussien (2021).

Maximum-likelihood estimation in a regular autoencoder takes the following form:

$$\begin{aligned}\theta^{AE}, \phi^{AE} &= \arg\max_{\theta,\phi} \textstyle\sum_c \mathbb{E}_{\delta}(z_c - g_{\phi}(x_c))\,[log(p_{\theta})(x_c|z_c)] \\ &= \arg\max_{\theta,\phi} \textstyle\sum_c log(p_{\theta})(x_c|g_{\phi}(x_c))\end{aligned} \tag{4.9}$$

We can note from (4.9) that the classical autoencoder effectively optimizes the first term in the VAE objective using a delta variational distribution. This means that $q_{\phi}(z_c|x_c) = \delta(z_c - g_{\phi}(x_c))$, and hence it does not regularize $q_{\phi}(z_c|x_c)$ toward any distribution like VAE. We can also note that $\delta(z_c - g_{\phi}(x_c))$ is a delta distribution with mass only at the output $g_{\phi}(x_c)$. Contrast this to what happens in VAE, where the learning is done using a variational distribution (i.e., $g_{\phi}(x_c)$ generates the parameters of a certain tractable distribution, the mean and variance in the case of Gaussian distribution). This implies that VAE has the ability to capture per-data-point variances in the latent space, $\mathbf{z_c}$. One of the main issues of autoencoders is the high possibility of overfitting which is due to the fact that the network learns to put all the probability mass to the non-zero entries in x_c. By introducing dropout Srivastava & et al. (2014) at the input layer, the classical autoencoder is less prone to overfitting. Fig. 4.3 shows the main difference between point estimate autoencoders and VAE.

Algorithm 4.1 PRVNet training for CSI feedback with stochastic gradient descent

```
Randomly initialize $\theta$ and $\phi$;
while not convergeed do
    Sample a batch of CSI channels $\mathscr{B}$;
    forall $c \in \mathscr{B}$ do
        Sample $\varepsilon \in \mathcal{N}(0, I)$;
        Compute $z_c$ using the reparameterization trick;
        Compute noisy gradient $\nabla_{\theta}\mathscr{L}$ and $\nabla_{\phi}\mathscr{L}$ using the sampled $z_c$;
    end
    Average noisy gradient for a batch;
    Update $\theta$ and $\phi$ using stochastic gradient descent;
end
Return $\theta$ and $\phi$
```

4.3.2 The proposed model (PRVNet)

As discussed in subsection 4.3.1, the second term in the loss function (4.6) introduces a compromise between how close the approximate posterior stays to the prior during learning and our ability to reconstruct the original data from the codeword. Therefore, we introduce a new hyperparameter β, where $\beta \neq 1$. Note that by using this hyperparameter, we are no longer optimizing a lower bound on the log marginal likelihood.

Setting $\beta < 1$ means that we force the model to learn better data reconstruction and to pay less attention to the prior constraint $\frac{1}{C}\sum_{c=0}^{C} q(z|x_c) \approx p(z) \approx \mathcal{N}(z;0,I_K)$. In other words, a model trained with $\beta < 1$ will be less able to generate novel CSI matrices by ancestral sampling. On the other hand, setting $\beta > 1$ emphasizes the importance of the prior distribution constraint over the ability to reconstruct the input from the codeword. Note that setting β to zero eliminates the prior distribution constraint and reduces the loss function to that of the classical point estimate autoencoders.

Recall that our goal is to make a good reconstruction at the BS side without generating novel imagined CSI matrices. Treating β as a free hyperparameter, with $\beta < 1$, therefore can significantly improve the reconstruction results without any additional cost in terms of time or the number of model parameters. Therefore, we propose an objective function in (4.10). Since we can interpret the second term as a regularization term, we coin a model trained with (4.10) by a partially regularized VAE network (PRVNet).

$$-\mathbb{E}_{z\sim q_(z|x)}\left[log\, p_\theta(x|z)\right] + \beta \cdot KL(q_\phi(z|x)||p(z)) \tag{4.10}$$

Selecting a value for β

We propose an algorithm for selecting the best value of β. At the beginning of the training phase, we set $\beta = 0$ and gradually increase its value to 1. We linearly anneal the KL term slowly over a large number of gradient updates to ϕ and θ and record the best value of β when the performance reaches the peak Liang & et al. (2018). After figuring out the best value of β, which we denote here as β_*, we retrain the model with the values of β starting from 0 to β_*. If the computation power is limited, we can stop increasing β once we notice a degradation in the validation metric. In this way, training our model does not incur any additional cost compared with training traditional VAE models.

Training PRVNet

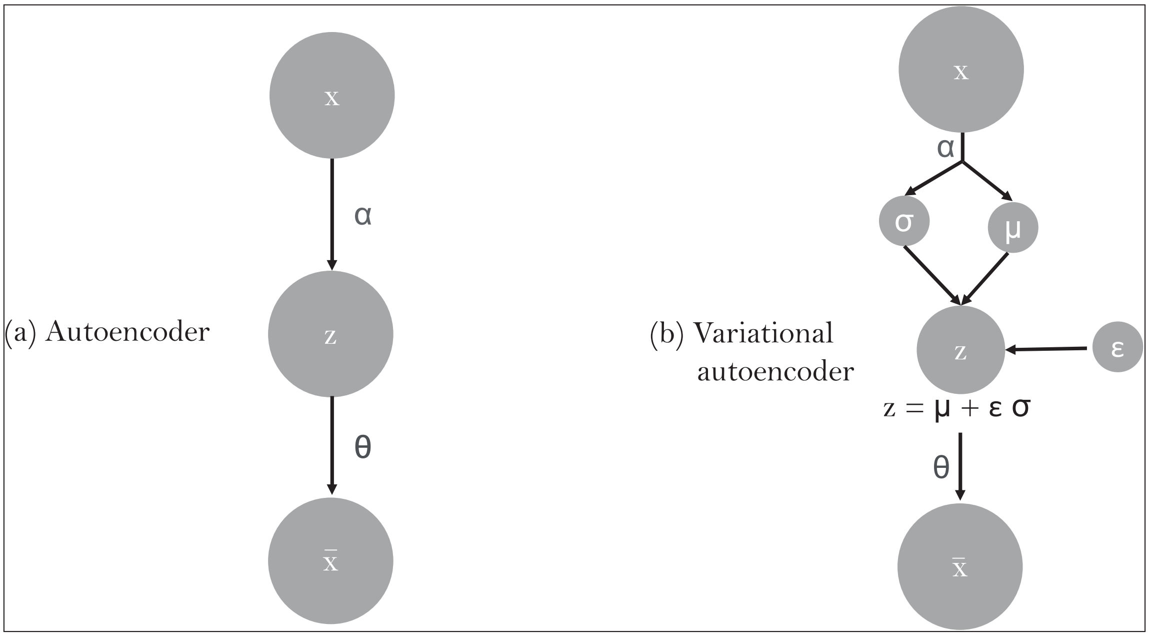


Figure 4.3 Variatinal autoencoder with reparameterization trick versus classical point estimate autoencoders

Recall that the proposed model optimizes the function in (4.10) while VAE is trained to optimize the standard ELBO function given in (4.6). We can obtain an unbiased estimate of (4.10) by sampling $\mathbf{z_c} \sim q_\phi$ and optimize it by stochastic gradient descent. However, the challenge is that we cannot trivially take gradients with respect to ϕ through this sampling. The reparameterization trick solves this challenge by sampling $\varepsilon \sim \mathcal{N}(0, I_k)$ and reparametrize the generated latent code such that, $\mathbf{z_c} = \mu_\phi(x_c) + \varepsilon \odot \sigma_\phi(x_c)$ Kingma & Welling (2013). This way, the stochasticity in the sampling process is eliminated and the gradient with respect to ϕ

now can be back-propagated through the sampled latent code $\mathbf{z_c}$. A detailed description for the training process is given in Algorithm 4.1.

4.4 Simulation Results and Analysis

4.4.1 Experiment Setup

Architecture Details

The encoder and decoder models are convolutional neural networks (CNN)-based architectures. We set the batch size to 128. The model weights are initialized according to *He* initialization He & et al. (2015). We optimize the model using *Adam* optimizer Kingma & Ba (2014) with 0.1 learning rate for 1000 epochs. The function proposed in (4.10) is used as the model loss function. To alleviate the effect of overfitting, we employ a weight decay of 1^{-4} for kernel and bias weights.

Dataset

We consider two types of scenarios as given in Wen & et al. (2018): the outdoor scenario at 300MHz and the indoor scenario at 5.3GHz. The channels are generated following the default settings of COST 2100 Liu & et al. (2012). At the BS, a uniform linear array (ULA) with $N_t = 32$ is considered. For the FDD system, we set $N_c = 1024$ in the frequency domain and $N_a = 32$ in the angular domain. The dataset contains $150,000$ independently generated channels divided into three parts. The train, validation, and testing parts consist of $100,000$, $30,000$, and $20,000$ channel matrices, respectively.

4.4.2 Performance of PRVNet

We compare the performance of PRVNet with three CS-based methods, namely, Lasso L_1-solver Daubechies *et al.* (2004), TVAL-3 Chengbo & et al. (2009), and BM3D-AMP Metzler & et al. (2016). Moreover, two recent deep learning-based methods, namely, CsiNet

Table 4.1 Comparison of NMSE (db) for different methods

CR	Methods	**NMSE (db)**	
		Indoor	**Outdoor**
1/4	LASSO	-7.59	-5.08
	BM3D-AMP	-4.33	-1.33
	TVAL3	-14.87	-6.90
	CsiNet	-17.36	-8.75
	CRNet	-26.99	-12.71
	PRVNet (our)	**-27.7**	**-13.9**
1/16	LASSO	-2.72	-1.01
	BM3D-AMP	0.26	0.55
	TVAL3	-2.61	-0.43
	CsiNet	-8.65	-4.51
	CRNet	-11.35	-5.44
	PRVNet (our)	**-13**	**-6.1**
1/32	LASSO	-1.03	-0.24
	BM3D-AMP	24.72	22.66
	TVAL3	-0.27	0.46
	CsiNet	-6.24	-2.81
	CRNet	-8.93	-3.51
	PRVNet (our)	**-9.52**	**-4.23**
1/64	LASSO	-0.14	-0.06
	BM3D-AMP	0.22	25.45
	TVAL3	0.63	0.76
	CsiNet	-5.84	-1.93
	CRNet	-6.49	-2.22
	PRVNet (our)	**-6.9**	**-2.53**

Wen & et al. (2018) and CRNet Lu *et al.* (2020), are also considered in the comparison. To evaluate the performance of different methods, we measure the distance between the original CSI matrix, $\mathbf{H}_a$, and the reconstruction image, $\hat{\mathbf{H}}_a$, using the normalized mean square error (5.12).

$$\text{NMSE (db)} = 10\log\,\mathbb{E}\left(\frac{\left\|\mathbf{H}_a - \hat{\mathbf{H}}_a\right\|_2^2}{\|\mathbf{H}_a\|_2^2}\right). \tag{4.11}$$

Table 5.4 and Fig. 5.6 show the performance of the proposed PRVNet against different benchmark models. We can see that PRVNet outperforms all classical CS-based methods as well as recent deep learning-based methods. PRVNet with the proposed loss function in (4.10) is capable of capturing CSI features to increase the reconstruction accuracy at the BS. Unlike the other benchmark models which do not consider channel noise in their model design, an advantage of the proposed model is its robustness against different noise levels. This property will be further investigated shortly.

Table 4.2 The effect of annealing β to different values on indoor scenario for a compression ratio 1/4

β annealing strategy	NMSE (db)
No annealing	-25.83
Annealing β to the maximum (β=1)	-26.32
Annealing β to 0.3	**-27.7**

The effect of β annealing has been studied and demonstrated in Table 4.2. We can see that the model achieved the highest NMSE when no β-annealing has been applied. Under the same dataset and compression ratio, the model achieved lower NMSE when β has been annealed to 1. The best NMSE has been achieved by annealing β from 0 to 0.3 and completing the training epochs without further increase in the value of β. Although, this value might be sub-optimal compared to a thorough grid search. The proposed algorithm is much more efficient and achieves a similar performance.

Table 4.3 The robustness of the proposed PRVNet under different signal-to-noise ratios.

SNR (db)	NMSE (db)
35	-27.7
32	-26.56
29	-25.81
26	-25.6
23	-24.95

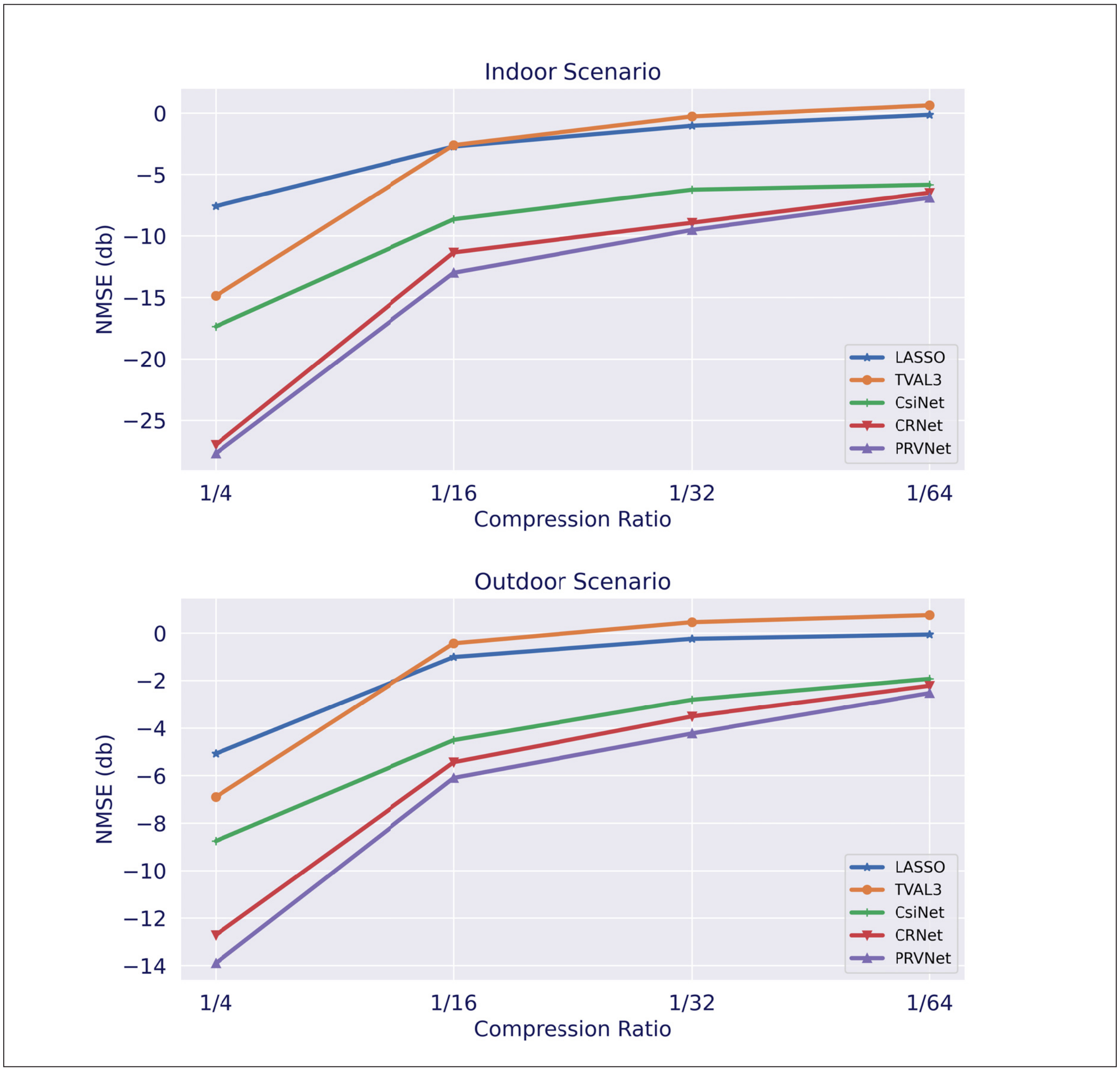


Figure 4.4 Top: indoor, bottom: outdoor. Comparison (in terms of the reconstruction loss measured in NMSE) between the proposed model and other works in literature

To further evaluate the robustness of the proposed model under different noise conditions, we simulate the AWGN feedback channel by adding random Gaussian noise to the codeword and passing it through the decoder model such that:

$$\bar{\mathbf{z}} = \mathbf{z} + \varepsilon, \tag{4.12}$$

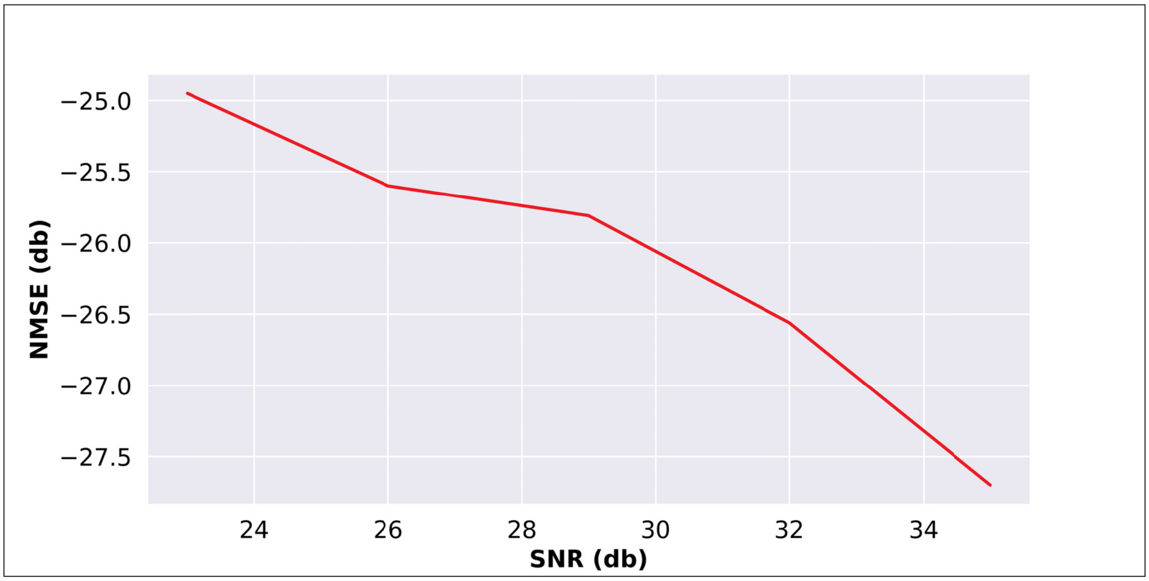


Figure 4.5 The NMSE(db) under different SNR(db) values

where $\varepsilon \sim \mathcal{N}(0, \sigma_n)$. The degradation of the NMSE with different SNR values is shown in Table 4.3. We can see that the proposed PRVNet model shows outstanding robustness against different noise levels, see Fig. 4.5. We notice a slow degradation in the NMSE when the noise level increases, which indicates that the codewords generated by the proposed PRVNet model can still convey relevant features about the original CSI matrix even under noisy conditions. This can be explained by the fact that PRVNet, unlike other models in the literature, learns a distribution for each dimension in the codeword. This makes the effect of the noise much less severe than in point estimate models because even with the noise, a value in the codeword may still look as being sampled from the same learned distribution.

4.5 Conclusion

In this paper, a novel deep learning model, PRVNet, has been proposed for downlink channel state information (CSI) feedback in MIMO-FDD systems. The PRVNet customizes the traditional variational autoencoder objective to incorporate the special characteristics of the CSI feedback problem. Unlike prior work that assumes an ideal feedback channel, we modeled an AWGN feedback channel and proved that the codewords generated by PRVNet are more robust against varying noise conditions. The proposed model outperforms state-of-the-art deep learning-based and compressive-sensing-based models in both noise-free and noisy channel conditions.

CHAPTER 5

ENABLING EFFICIENT DATA INTEGRATION OF INDUSTRY 5.0 NODES THROUGH HIGHLY ACCURATE NEURAL CSI FEEDBACK

Mostafa Hussien, Kim Khoa Nguyen, Ali Ranjha, Moez Krichen, Abdulaziz Alshammari, and Mohamed Cheriet

École de technologie supérieure (ÉTS), Univeristy of Québec, Canada, H3C 1K3

Paper published in IEEE Transactions on Consumer Electronics, July 2023

Abstract: Industry 5.0 refers to the fifth industrial revolution that leverages advanced technologies such as the Internet of Things (IoT) and artificial intelligence (AI) to increase efficiency, productivity, and flexibility in manufacturing and other industries. Wireless IoT devices help collect and transmit real-time data to support intelligent decision-making, while AI algorithms process and analyze the data to optimize production processes, predict equipment failure, and enhance supply chain management. To achieve efficient integration for the data fused by various sensors, the data should be perfectly synchronized and out-of-errors. In cellular-based sensors, this requires the base station (BS) to know the state of the channel at each node. In this work, we propose a novel method for CSI compression by learning an approximation for a sufficient statistics function. Our method establishes a new category of compression techniques based on the theory of sufficient statistics. Moreover, We present a detailed analysis of the upper bound of the prediction error in our specific scenario. We develop a Bayesian optimization framework to optimally select the adopted neural network architecture. The experimental results confirm that our solution outperforms both conventional and learning-based solutions in terms of reconstruction error, model size, and scalability.

Keywords:

CSI feedback, MIMO-FDD systems, Industry 5.0

5.1 Introduction

Consumer electronics have become miniaturized powerful computers with a wide range of functions due to the integration of information technology. These devices are often used in interconnected networks of products and services. In this regard, Industry 5.0 is considered to combine human expertise along with intelligent, efficient and precise machines enabling futuristic technologies such as Internet of things (IoT), artificial intelligence (AI), industrial automation, health informatics, and collaborative robots to name a few. In this regard, data fusion from multiple wireless IoT nodes is one of the main challenges. Moreover, accurate synchronization and error-free transmission are crucially required to achieve efficient data integration. To this end, a base station (BS) should obtain timely updates on the channel state of each node. This process is known as the channel state information (CSI) feedback process. The BS uses this feedback to adjust the transmission parameters for each node according to its current channel status. Therefore, the need for an accurate CSI feedback mechanism for data fusion and data integration in industry 5.0 becomes clear.

Massive multiple-input multiple-output (MIMO) technology is considered as one of the main enablers of beyond 5G (B5G) systems. Using beamforming technologies, MIMO systems enhance channel capacity and throughput by using hundreds of antennas Wen & et al. (2018). To improve the performance gain of MIMO, the BS requires access to downlink CSI, which is available in time division duplexing (TDD) systems where channel reciprocity is achieved Liang & et al. (2020). In frequency division duplexing (FDD), only the UE can estimate and transmit the channel back to the BS Guo & et al. (2020). However, transmitting a large CSI matrix to the BS consumes a significant amount of bandwidth. Therefore, the UE should compress the matrix before transmitting it to the BS. According to the rate/distortion theory Blau & Michaeli (2019), the cost of increasing the compression ratio is increasing the reconstruction distortion.

Different techniques have been employed to achieve a good rate/distortion balance in CSI compression. Principal component analysis, vector quantization, compressive sensing, and deep

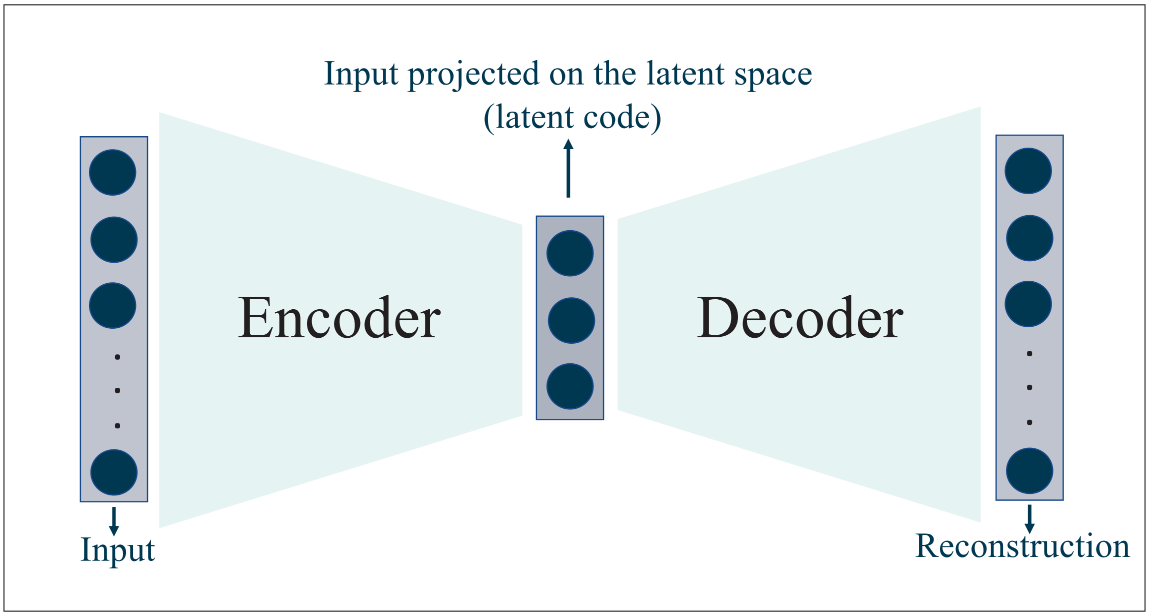


Figure 5.1 The autoencoder architecture widely adopted in the literature for CSI compression

learning are among the widely used techniques in this task. Deep learning techniques achieved a significant improvement in the reconstruction distortion for a given compression ratio compared with other conventional methods. These methods mainly work by nonlinearly projecting its input to a smaller space (i.e., *latent space*) using encoder architecture parameterized by parameters ϕ, as shown in Fig. 5.1. These methods, jointly learn the inverse mapping from the latent space to the input space by a decoder parameterized by parameters θ. Although they achieved a competitive performance compared with conventional techniques, some limitations are inherited in the applicability of such systems, such as:

- Autoencoders have a fixed size input. Therefore, the BS should maintain a separate model for each supported compression ratio. Although some work has proposed adaptive rate models Wang & et al. (2021), they achieve this by a padding module that pads the codewords. This padding trick could harm the reconstruction accuracy.
- Autoencoders work by learning the deep features of their input. These features significantly change between different propagation environments (e.g., indoor and outdoor). Therefore,

a different model should be trained for different environments. Specifically, a separate model should be trained for each setting of the compression ratio, propagation scenario, and the number of antennas. For example, 24 models are required at the UE side to support 2 propagation scenarios, 3 configurations of BS antennas, and 4 settings of compression ratios.

- Some prior work proposed complex architectures (e.g., attention-based models, LSTM-based models) with large model sizes and higher computational complexity. Unfortunately, the complexity of such models keeps them far away from the current UE capabilities.

Table 5.1 Summary of the notation adopted in the text.

Notation	Meaning
$\mathbf{H}$	Bold-faced capital letters denote matrices
$\mathbf{a}$	Bold-faced small letters denote vectors
$\|\cdot\|$	Cardinality (i.e., the number of elements in a matrix)
$\|\|\cdot\|\|_2$	Euclidean norm
$\|\|\cdot\|\|_F$	Frobenius norm
$\text{vec}(\mathbf{H})$	Vectorization of the matrix $\mathbf{H}$ in column-first order
$\lambda_{min}(\mathbf{H})$	The minimum eigenvalue of a symmetric matrix $\mathbf{H}$
$\mathbf{I}$	The identity matrix
$[n]$	The set $\{1,2,\ldots,n\}$
$\mathcal{N}(\mu,\sigma)$	A Gaussian distribution with mean μ and variance σ
$\sigma(\cdot)$	ReLU activation in the form of $\sigma(x)=\max(0,x)$
$\mathbb{I}(E)$	The indicator function of an event E

This work presents the first attempt for neural CSI compression beyond the nonlinear projection using autoencoders. We propose a novel technique for CSI feedback by instance-aware optimization. This method has its roots in the learning theory, specifically in the bias/variance tradeoff. Simply, bias and variance count for how much good the model is behaving in seen and unseen data, respectively. These two error terms are inversely proportional such that de-

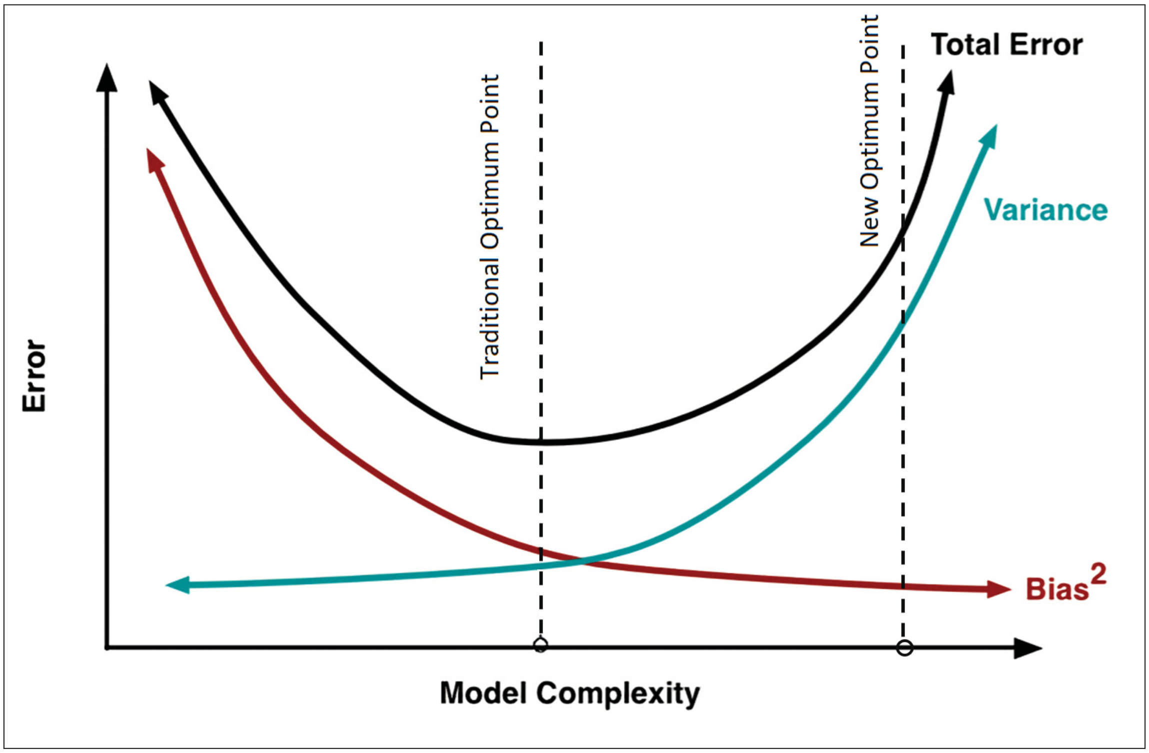


Figure 5.2 The total error of a neural network model as the combination of the bias and variance error terms Neal & et al. (2018)

creasing one term will increase the other as indicated in Fig. 5.2. To the best of our knowledge, this is the first work to answer the following question:

Question: What is the optimal combination of bias/variance errors for the CSI compression problem? and what is the optimality of such a method in terms of error upper bound and the employed architecture?

To answer these questions, we propose an instance-aware optimization technique. In this case, the optimal bias/variance combination is no longer a midpoint between the two error terms. Rather, it is a point with a low bias (even with high variance). In this case, we do not consider the variance due to the fact that we optimize a model per each instance. Therefore, for each sampled channel, we approximate a nonlinear sufficient statistics function using a neural network as a function approximator. Given an instance, $\mathbf{H}$, a sufficient statistic function, $\tau(\mathbf{H})$, is a function that contains all the information in $\mathbf{H}$ required to compute any estimate. The parameters of such a function are interpreted as the codeword. Extensive experiments prove that our reconstruction distortion and model size are significantly reduced compared to other state-of-the-art solutions. Moreover, it is a modular solution that works with different system configurations. We can summarize the contributions of our work as follows:

- An efficient enabler for fused data integration in IoT and industry 5.0 through a highly accurate CSI feedback mechanism.
- Proposing a new modeling for the CSI compression by approximating a sufficient statistics function for each channel matrix. The compressed representation of the CSI matrix is, simply, the weights of the trained neural networks.
- A detailed error upper bound analysis for the specific proposed scenario, which allows us to better understand the proposed solution behavior.
- A Bayesian framework for optimizing the neural network model for each instantaneous channel to adapt the changes in each channel distribution.
- Taking a fundamental step towards opening a new direction for CSI compression other than nonlinear projection. This work combines concepts from learning theory (bias/variance tradeoff) Yang & et al. (2020), information theory (sufficient statistics) MacKay & Mac Kay (2003), and approximation theory (error upper bounds) Elbrächter & et al. (2019) for CSI compression.

The rest of this paper is organized as follows: Section 5.3 introduces the system model. Section 5.4 describes the proposed method. Section 5.5 provides an upper bound analysis for the error. The experimental results are shown in Section 5.6. An overview of the literature work is given in Section 5.2. Section 5.7 concludes the work.

5.2 Literature Review

Prior work is categorized into two primary classes: conventional-based and deep learning-based techniques, see Table 5.2. Conventional techniques contain principal component anal-

Table 5.2 A Comprehensive taxonomy for the literature work based on the applied technique

	Applied Technique	Publications
Conventional Techniques	Principal Component Analysis (PCA)	Joung et al. Joung *et al.* (2016), Joung et al. Joung (2016)
	Vector Quantization (VQ)	Kang et al. Kang & Choi (2021), Ying et al. Ying & et al. (2014)
	Compressive Sensing	Rao et al. Rao & Lau (2014), Lu et al. Lu & et al. (2015), Gao et al. Gao & et al. (2018)
Deep Learning	Convolutional Autoencoders	Wen et al. Wen & et al. (2018), Lu et al. Lu *et al.* (2020)
	Recurrent Autoencoders	Liu et al. Liu *et al.* (2021), Wang et al. Wang & et al. (2018a)

ysis, vector quantization, and compressive sensing techniques. In vector quantization, the BS builds a codebook of channels Kang & Choi (2021), where a new CSI matrix is encoded by determining its nearest point in the codebook and assigning it to the corresponding index. Euclidean distance and Cosine similarity are commonly employed metrics to measure the distance between two data points. The distance between the nearest codebook entry and the current channel is used as a measure of reconstruction loss. However, these methods have two major drawbacks: 1) increasing antenna numbers results in a larger codebook, and 2) longer encoding/decoding time due to the incorporated linear search in the codebook. Another research avenue explores the use of compressive sensing (CS) for CSI compression Rao & Lau (2014); Gao & et al. (2018); Lu & et al. (2015). While CS-based methods have demonstrated improved performance in comparison to earlier methods, they mandate a sparsity constraint on the channel matrix, which may not be satisfied in modern MIMO systems. Moreover, these methods use iterative algorithms for solving a system of equations, which can pose challenges in terms of real-time requirements.

Inspired by the exceptional performance of deep learning, various proposals have been made for deep learning-based CSI compression Lu *et al.* (2020); Hussien *et al.* (2020); Guo *et al.* (2020). This approach endeavors to overcome the limitations of conventional techniques while capitalizing on the superior proven capabilities of learning methods. The common approach employed by these methods is the utilization of autoencoder-based architectures, as illustrated in Fig. 5.1. The encoder part of the autoencoder transforms the input channel into a lower dimensional space. Using compressed latent codes, the decoder reconstructs the original input. One of the early proposals was presented in Wen & et al. (2018), where a convolutional neural network (CNN) was used to learn the features of the CSI matrices. The authors introduced a model named *CsiNet*, which significantly outperformed state-of-the-art models. The exceptional performance of *CsiNet* spurred further research to enhance its effectiveness and application. For instance, in Hussien *et al.* (2020), partially regularized variational autoencoders were proposed to tackle the noise in the feedback channel, while in Lu *et al.* (2020), a novel architecture was presented to extract CSI features from multiple resolutions.

5.3 Problem Definition and Setup

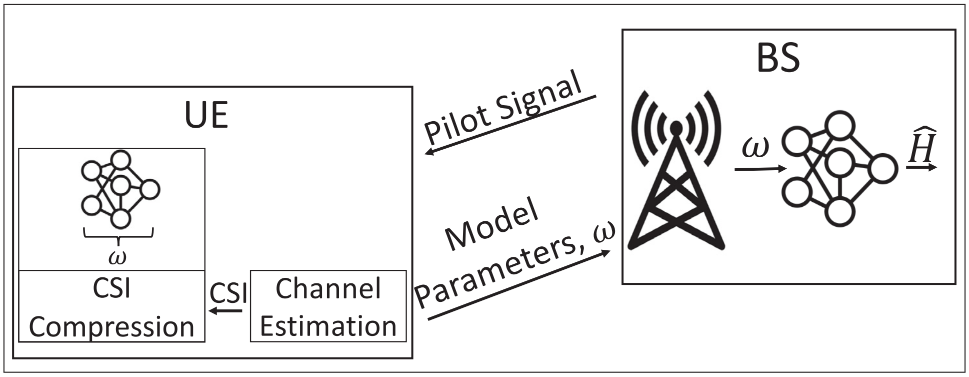


Figure 5.3 An overview of the system model

We consider a single-cell massive MIMO system consisting of a base station (BS) and user equipment (UE), as shown in Fig. 5.3. The BS has N_t transmit antennas, where N_t is significantly greater than 1. The UE is equipped with a single receive antenna. The system operates using an orthogonal frequency-division multiplexing (OFDM) scheme, with N_c orthogonal subcarriers. The received signal at the n^{th} subcarrier is described by the following equation:

$$y_n = \mathbf{h}_n^T \mathbf{v}_n x_n + z_n \tag{5.1}$$

where $\mathbf{h}_n$ is the channel vector, $\mathbf{v}_n$ is the transmit beamforming vector, x_n is the transmitted signal, and z_n is additive white Gaussian noise.

The CSI matrix $\mathbf{H}$ consists of the concatenation of the instantaneous channel vectors at all subcarriers and can be expressed as $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}2, \ldots, \mathbf{h}N_c] \in \mathbb{C}^{N_c \times N_t}$. The complete downlink instantaneous CSI matrix has a dimension of $N_c \times N_t$ complex numbers, which is equal to $2N_cN_t$ real numbers when taking into account the real and imaginary components of each complex value. The increasing number of antennas and subcarriers leads to a significant increase in the size of the matrix, resulting in a substantial amount of bandwidth being consumed for the feedback process. Furthermore, the extended transmission time for this large volume of information increases the likelihood of the feedback being outdated.

The CSI matrix is transformed into the angular-delay domain through the application of a 2D Discrete Fourier Transform (2D-DFT), as represented by the equation:

$$\mathbf{H}' = \mathbf{F}_d \mathbf{H} \mathbf{F}_a \tag{5.2}$$

where $\mathbf{F}_d \in \mathbb{C}^{N_c \times N_c}$, and $\mathbf{F}_a \in \mathbb{C}^{N_t \times N_t}$ are the two DFT matrices. The result of this transformation, $\mathbf{H}'$, contains elements that represent a specific path delay and angle of arrival (AoA). Only the first N_a rows of $\mathbf{H}'$ contain valuable information, while the remaining rows, with near-zero values, can be discarded without causing a significant loss of information. By retaining only the first N_a rows, a new matrix, $\mathbf{H}_a \in \mathbb{C}^{N_a \times N_t}$, is created with a reduced size of $2N_aN_t$ where $N_a < N_c$.

The real and imaginary parts of the complex values in $\mathbf{H}_a$ are concatenated along the third dimension of $\mathbf{H}_a$ to form a tensor. The elements of the resulting tensor, $\mathbf{H}_a \in \mathbb{R}^{N_a \times N_t \times 2}$, are then normalized to fall in the range of [0,1].

5.4 Proposed Solution

5.4.1 Background on Learning Theory

In neural network functions, a balance is usually sought between approximation error and estimation error. Approximation error measures the difference between the true target function and the closest neural network of a given architecture, while estimation error measures the difference between that closest function and the estimated function. The approximation error, also known as statistical risk or variance, affects a network's ability to generalize to new data points. Whereas the estimation error, also known as the model bias, controls the model behavior on the training dataset. It is well known that low bias comes at the cost of high variance, but rethinking the modeling of CSI compression could lead to significant improvements.

In CSI compression, the goal is to find a compressed representation of the CSI matrix that retains enough information from the original data to allow for efficient and minimally distorted reconstruction. The method used to achieve this representation, whether it be a universal model or a specifically trained model, is not important. The focus is on reducing reconstruction loss at a given compression rate. Using instance-based optimization eliminates variance error and focuses the optimizer on reducing bias error, leading to lower reconstruction loss at a given compression ratio, as shown by the results.

5.4.2 Model Description

Prior work modeled the problem of neural CSI compression as jointly optimizing an encoder and decoder parameters, ϕ and θ respectively, to minimize the $l2$ loss between a ground truth

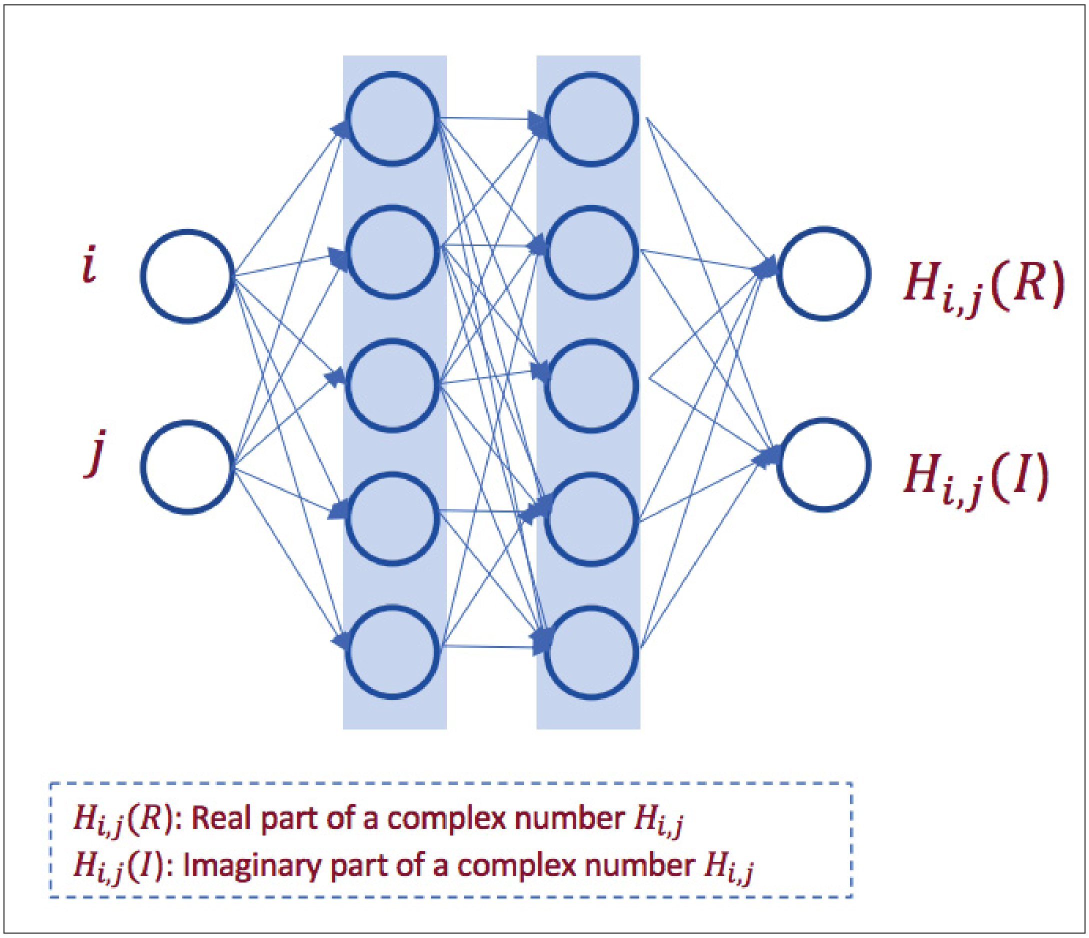

i
j
$H_{i,j}(R)$
$H_{i,j}(I)$
$H_{i,j}(R)$: Real part of a complex number $H_{i,j}$
$H_{i,j}(I)$: Imaginary part of a complex number $H_{i,j}$

Figure 5.4 The proposed shallow network for learning a highly biased representation for a CSI channel $\mathbf{H}$

channel and its reconstruction such that:

$$\underset{\phi,\theta}{min} \left\| \mathbf{H} - f_\theta(f_\phi(\mathbf{H})) \right\|_2 . \tag{5.3}$$

The limitations of this modeling have been discussed earlier in section 8.1. In this paper, we present a new modeling that learns a function parameterized by parameters ω, $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, such that:

$$f_\omega(i, j) \cong [R(\mathbf{H}_{i,j}), I(\mathbf{H}_{i,j})] \quad \forall i \in [N_a], j \in [N_t], \tag{5.4}$$

where i, j are the row, column indices in the channel matrix. The real and imaginary components of complex numbers are denoted by $R(\cdot)$ and $I(\cdot)$. Finally, $[n] = \{1, 2, \ldots, n\}$. When the training completes, the trained weights, ω, is treated as the compressed codeword.

The proposed formulation considers the channel matrix, $\mathbf{H}$, with dimensions $N_a \times N_t$ as a dataset $\mathscr{D}$ such that $\mathscr{D} = \{((i, j), \mathbf{H}_{i,j})\}_{i=1,j=1}^{i=N_a,j=N_t}$. The input to the model is constant and does not depend on the current channel being compressed, with only the labels changing. According to Shannon's theory Shannon *et al.* (1959), this results in zero entropy for the degenerate distribution of the input, meaning there is no need to transmit the input matrix to the BS. The original matrix can be reconstructed using only the model parameters, ω, and the matrix dimensions (i.e., N_a and N_t) by plugging the index vectors into the function f_ω such that:

$$\hat{\mathbf{H}} = f_\omega(\mathbf{i}, \mathbf{j}) \tag{5.5}$$

where $\mathbf{i} = [N_a]$ and $\mathbf{j} = [N_t]$. Approximating this function with a neural network is easier than approximating the true underlying channel distribution, making it possible to achieve good performance with a shallow architecture (one hidden layer with few nodes, as shown in Fig. 5.4). Algorithm 1 shows the details of the encoding process at the UE and the decoding process at the BS side.

Recall that the total error of a neural network consists of two terms namely the estimation error and approximation error:

$$\mathscr{L}(f_\omega) = \mathscr{L}_a(f_\omega) + \mathscr{L}_e(f_\omega) \tag{5.6}$$

where $\mathscr{L}_a(f_\omega)$ is the approximation error of the function f_ω and $\mathscr{L}_e(f_\omega)$ is the estimation error. We employ a neural network model for approximating f_ω with an instance-aware optimization. In this case, we eliminate the contribution of the approximation error, and the optimizer is dedicated to minimizing the estimation error. This helps achieve better results compared with traditional neural network optimization methods. Fig. 8.2 shows the overall system and how it works on both sides.

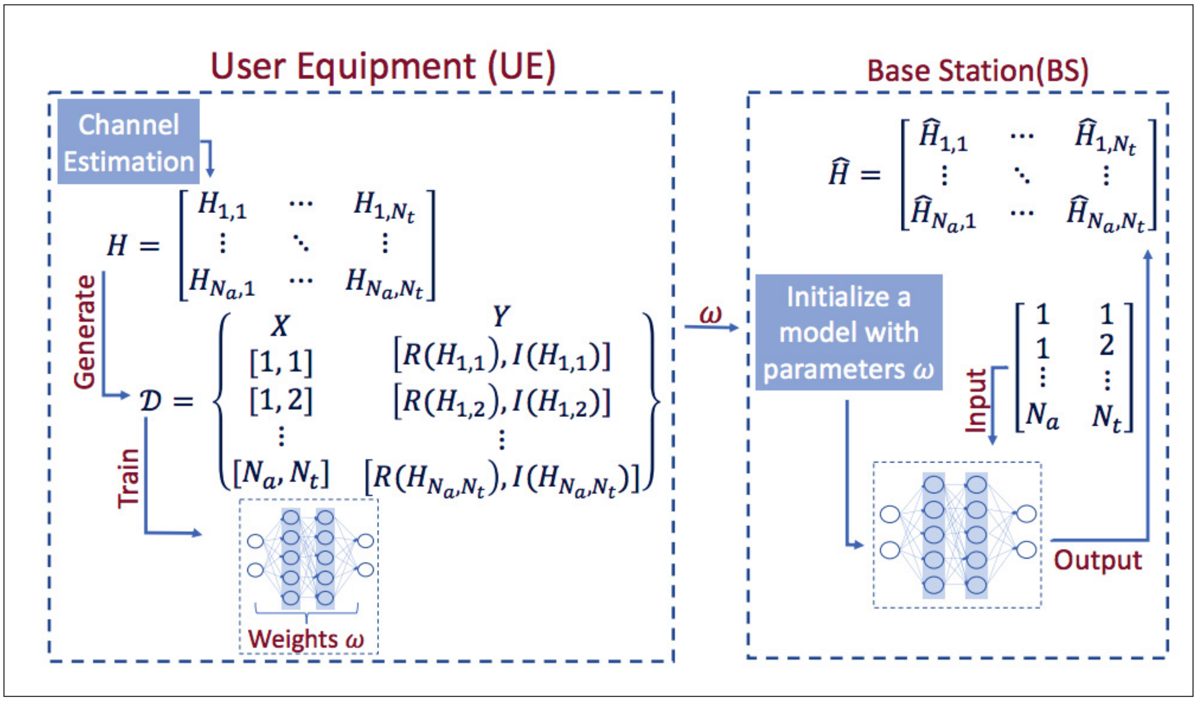

User Equipment (UE)
Channel Estimation
$H = \begin{bmatrix} H_{1,1} & \cdots & H_{1,N_t} \\ \vdots & \ddots & \vdots \\ H_{N_a,1} & \cdots & H_{N_a,N_t} \end{bmatrix}$
Generate
$\mathcal{D} = \left\{ \begin{array}{cc} X & Y \\ [1,1] & [R(H_{1,1}), I(H_{1,1})] \\ [1,2] & [R(H_{1,2}), I(H_{1,2})] \\ \vdots & \vdots \\ [N_a, N_t] & [R(H_{N_a,N_t}), I(H_{N_a,N_t})] \end{array} \right\}$
Train
Weights ω
ω
Base Station(BS)
$\widehat{H} = \begin{bmatrix} \widehat{H}_{1,1} & \cdots & \widehat{H}_{1,N_t} \\ \vdots & \ddots & \vdots \\ \widehat{H}_{N_a,1} & \cdots & \widehat{H}_{N_a,N_t} \end{bmatrix}$
Initialize a model with parameters ω
Input
$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ N_a & N_t \end{bmatrix}$
Output

Figure 5.5 A general overview of the proposed framework at the BS and UE sides

Algorithm 5.1 CSI compression using the proposed method.

```
Input: A new channel matrix, $\mathbf{H}$
Encoding at the UE
    - Generate a dataset, $\mathscr{D}$, from $\mathbf{H}$, such that: $\mathscr{D} = \{((i,j), \mathbf{H}_{i,j})\}_{i=1,j=1}^{i=N_a,j=N_t}$.
    - Initialize a neural network model with initial weights $\omega'$.
    - Train the model, $f_{\omega'}$, for $n$ epochs to get a trained-model parameters $\omega$.
    - Transmit the trained-model parameters $\omega$ to the BS.
Decoding at the BS
    - Initialize a new model with the received weights $\omega$.
    - Plug the fixed input matrix (i.e., the indices of the channel matrix) to obtain a
      reconstruction for the original channel, $\hat{\mathbf{H}}$.
```

The proposed modeling approach introduces a new type of compression algorithm that utilizes learning internal representations of the data instead of traditional projection techniques. This offers a new way to reduce approximation error for compression applications. A main advantage of the proposed solution is being not constrained to certain propagation scenarios or MIMO settings, eliminating the need for the UE to keep multiple models for different scenarios.

5.4.3 Information Theoretic Background

Information theory serves as a theoretical foundation of our work. The weights of the shallow network are interpreted as the parameters of a parameterized function f_ω. The function is an approximation for the sufficient statistics of the sampled channel.

Definition 5.1. *For a family of distributions, $f_\theta(X)$, a function $T(X)$ is said to be sufficient statistics if X is independent of θ given $T(X)$ for any distribution on θ.*

According to definition 5.1, this means that a sufficient statistics function should contain enough information about the sample, X, such that:

$$P(\theta|X) \leq P(\theta|T(X)), \tag{5.7}$$

with the equality is only satisfied when the function $T(X)$ keeps all information in X without any loss. For tractable distributions, such as Gaussian, a closed-form equation for the sufficient statistics can be derived. However, for distributions with unknown parameters, θ, it is not feasible to derive such an equation. To tackle this issue, we leverage the capabilities of neural networks for function approximation to learn a good approximation for $T(X)$.

5.5 Bounds Analysis

Notation: During this text, we refer to vectors by small bold-face letters and matrices by capital bold-face letters. The Euclidean norm of a matrix $\mathbf{A}$ is denoted by $||\cdot||_2$. The vectorization of a symmetric matrix, $\mathbf{H}$, in a column-first order is denoted by $\text{vec}(\mathbf{H})$. A Gaussian distribution with mean μ and variance $\blacksquare$ is denoted by $\mathcal{N}(\mu,\blacksquare)$. We denote the indicator function of an event E by $\mathbb{I}\{E\}$. The identity matrix is referred to as $\mathbf{I}$ and the set $\{1,2,3,\ldots,n\}$ is referred to as $[n]$. The ReLU activation function is referred to as $\sigma(\cdot)$ such that $\sigma(x) = \max(0,x)$. Table. 5.1 summarizes the used notation; some notations are used in the appendix.

5.5.1 Defining A Shallow Neural Network

We considered a two-layers neural network with ReLU activation. The hidden layer has h nodes. The neural network function is defined as:

$$f_{\mathbf{W},\mathbf{a}}(\mathbf{x}) = \frac{1}{\sqrt{h}}\sum_{i=1}^{h} a_i\sigma(\mathbf{W}_i^T\mathbf{x}), \tag{5.8}$$

where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional input vector, $\mathbf{W} \in \mathbb{R}^{d\times h}$ is the weight matrix of the first layer such that W_i is the weight vector associated with the node i in the hidden layer. The weight vector of the second layer is given by $\mathbf{a} = (a_1,a_2,\ldots,a_h)^T \in \mathbb{R}^h$. A set of n labeled samples $\mathscr{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is available for training. These samples are i.i.d, with an underlying data distribution $\mathscr{D}$ over $\mathbb{R}^d \times \mathbb{R}$. For each sample (x_i, y_i), we assume $||x_i||_2 = 1$ and $|y| \leq 1$. The weights are randomly initialized. The model is trained over the dataset $\mathscr{S}$ using gradient descent to minimize the $l2$-loss given by:

$$\Phi(\mathbf{W},\mathbf{a},\mathscr{S}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{\mathbf{W},\mathbf{a}}(\mathbf{x}_i))^2 \tag{5.9}$$

5.5.2 Bounding the Generalization Error for Our Model

Given the dataset $\mathscr{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, we define a new matrix $\mathbf{H}^\infty \in \mathbb{R}^{n\times n}$ called *Gram matrix* such that:

$$\begin{aligned} \mathbf{H}^\infty &= \mathbb{E}_{w\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\mathbf{x}_i^T\mathbf{x}_j\mathbb{I}\{\mathbf{w}^T\mathbf{x}_i \geq 0, \mathbf{w}^T\mathbf{x}_j \geq 0\}] \\ &= \mathbf{x}_i^T\mathbf{x}_j(\pi - \arccos\big(\mathbf{x}_i^T\mathbf{x}_j\big))/2\pi, \quad \forall i,j \in [n]. \end{aligned} \tag{5.10}$$

This matrix has been extensively studied in Arora *et al.* (2019); Du (2018). It can be interpreted as a Gram matrix from a kernel associated with the ReLU function. For a sufficiently large number of nodes in the hidden layer, h, if $\mathbf{H}^\infty$ is positive definite, it is shown that gradient descent optimization converges to zero loss Du (2018).

Theorem 1. *Consider* $\varepsilon \in (0,1)$ *as a fixed failure probability. Assume a sample* $\mathscr{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ *is i.i.d. sampled from a* $(\lambda_o, \varepsilon/3, n)$*-nondegenrate distribution* $\mathscr{D}$*,* $k = O(\lambda_o\varepsilon/n)$*,* $h \geq k^{-2}\,\mathrm{Poly}(n, \lambda_0^{-1}, \varepsilon^{-1})$*. Consider a mean square error loss function,* $l : \mathbb{R}\times\mathbb{R} \to [0,1]$ *Then, with a probability at least* $(1-\varepsilon)$*, a two-layers NN* $f_{W,a}$ *has a population loss of* $L_{\mathscr{D}}(f_{W,a}) = \mathbb{E}_{(x,y)\sim\mathscr{D}}[l(f_{W,a}(x), y)]$ *bounded as:*

$$L_{\mathscr{D}}(f_{W,a}) \leq \sqrt{\frac{2\mathbf{y}^T(\mathbf{H}^\infty)^{-1}\mathbf{y}}{n}} + O(\sqrt{\frac{log\frac{n}{\lambda_0\varepsilon}}{n}}) \tag{5.11}$$

To compute this bound, the Gram matrix $\mathbf{H}^\infty$ should be invertible. However, this condition is not always satisfied. Furthermore, the model input in our method is fixed to the indices of the channel matrix. This input does not change with every channel. Therefore, this matrix should be replaced by an identity matrix $\mathbf{I}$ of size n where $n = 2N_aN_t$.

5.6 Experimental Results

Table 5.3 Experimental Setup

Channel Model	COST 2100 channel model Liu & et al. (2012)
Number of antennas in the the BS	ULA with 32 antennas
Number of antennas in the UE	1
Development environment	TensorFlow2.1.0 + python 3.6
Size of the dataset	150,000 channels
Propagation environments	Indoor and outdoor

This section presents the different results of evaluating our proposed method. The results of channel reconstruction for different propagation environments (e.g., indoor and outdoor environments) compared with state-of-the-art learning methods are shown in 5.6.1. The performance of our method in end-systems is given in 5.6.2. The results of Bayesian optimization are presented in 5.6.3. Finally, we show and discuss when and why our method does not work in 5.6.4. The used dataset has 150,000 CSI realizations for indoor and outdoor scenarios generated using the settings in table 5.3. For the complete description of the dataset, we refer the readers to Hussien *et al.* (2020). The reconstruction quality is measured by the NMSE given in (5.12).

$$\text{NMSE (db)} = 10\log \mathbb{E}\left(\frac{\left\|\mathbf{H}_a - \hat{\mathbf{H}}_a\right\|_2^2}{\left\|\mathbf{H}_a\right\|_2^2}\right). \tag{5.12}$$

5.6.1 Results and Discussion

The plots in Fig. 5.6 show how well different methods perform in indoor and outdoor environments. The figure shows a comparison between our method with different conventional techniques (i.e., LASSO Daubechies *et al.* (2004), BM3D-AMPMetzler & et al. (2016), TVAL3Li *et al.* (2009)) and deep learning-based techniques (i.e., CsiNetWen & et al. (2018), CRNetLu *et al.* (2020)). We can see that our method significantly outperforms other methods in terms of the reconstruction distortion measured by NMSE. The obtained results confirm the superiority of our proposed method compared with other methods. Due to its focus on minimizing one source of errors (i.e., estimation errors), the optimizer is able to achieve superior performance. Moreover, sufficient statistics functions learn informative features only and remove any redundancies in the considered sample. Table. 5.4 shows the NMSE values for each method.

Time Complexity: Despite the adopted per-instance training strategy, the time complexity of our method is still on-the-bar with other methods. This is a straight consequence of using a model with a small number of parameters. As a result, our model's training time is comparable

Table 5.4 Comparison of NMSE (db) for different methods

CR	**Methods**	**NMSE (db)**	
		Indoor	**Outdoor**
1/4	LASSO	-7.59	-5.08
	BM3D-AMP	-4.33	-1.33
	TVAL3	-14.87	-6.90
	CsiNet	-17.36	-8.75
	CRNet	-26.99	-12.71
	(Our method)	**-36.66**	**-22.69**
1/16	LASSO	-2.72	-1.01
	BM3D-AMP	0.26	0.55
	TVAL3	-2.61	-0.43
	CsiNet	-8.65	-4.51
	CRNet	-11.35	-5.44
	(Our method)	**-32.10**	**-19.13**
1/32	LASSO	-1.03	-0.24
	BM3D-AMP	24.72	22.66
	TVAL3	-0.27	0.46
	CsiNet	-6.24	-2.81
	CRNet	-8.93	-3.51
	(Our method)	**-28.95**	**-15.94**
1/64	LASSO	-0.14	-0.06
	BM3D-AMP	0.22	25.45
	TVAL3	0.63	0.76
	CsiNet	-5.84	-1.93
	CRNet	-6.49	-2.22
	(Our method)	**-23.61**	**-12.32**

Table 5.5 FLOPS for Different Models Compared with our Model.

	CR=$\frac{1}{4}$	CR=$\frac{1}{8}$	CR=$\frac{1}{16}$	CR=$\frac{1}{32}$
CsiNet-LSTM	412.9 M	410.8 M	409.8 M	409.2 M
MarkovNet	44.5 M	42.4 M	41.3 M	40.8 M
MarkovNet CNN	41.2 M	40.7 M	40.5 M	40.4 M
CsiNet	7.8 M	5.7 M	4.7 M	4.1 M
CRNet	7.7 M	5.6 M	4.5 M	4.0 M
Deep AE	6.3 M	5.8 M	5.5 M	5.4 M
Our Method (500 Epochs)	**520 K**	**283 K**	**128 K**	**66 K**

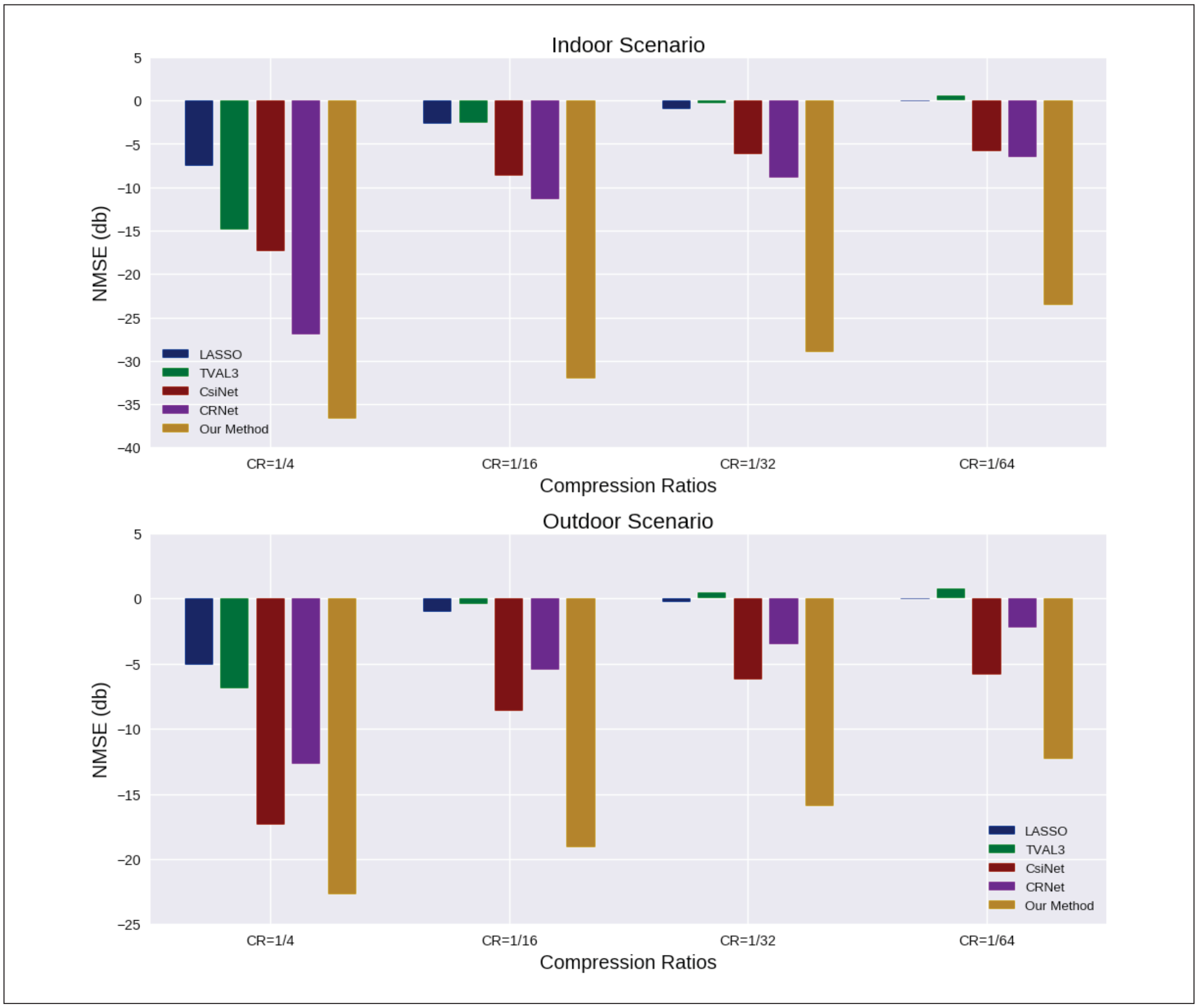


Figure 5.6 Evaluating the proposed model in comparison with other state-of-the-art models

to other models' inference time. Table 5.5 shows the floating-point operations (FLOPS) for different models. Furthermore, Fig. 5.7 shows the error for a 300 epochs training session. We can see that the error slightly improves after a relatively small number of epochs, therefore we can stop the training earlier to reduce the encoding time. Note that in certain modern communications scenarios, such as vehicular networks, the UE has strong computational capabilities. For example, the computation power of Tesla vehicles can run the auto-pilot that can process 2300 frames per second Talpes & et al. (2020). The computation resources of such vehicles are

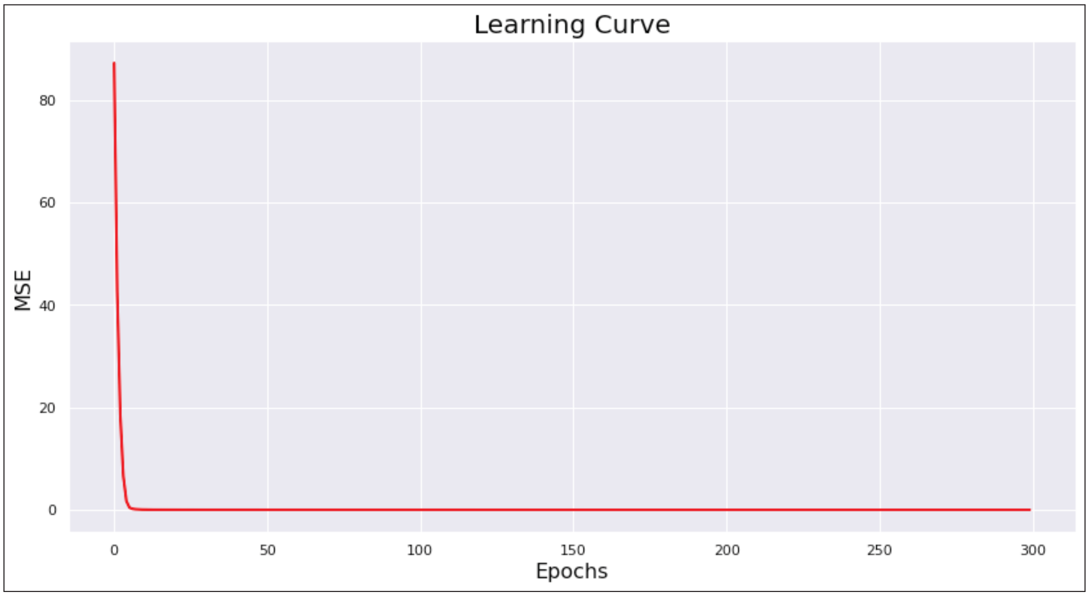


Figure 5.7 The error decay with training epochs

powerful enough to be used in Cryptocurrency mining, which requires extensive computations. This implies that training a shallow-net for CSI compression will be much more feasible.

Table 5.6 The number of parameters for different models in the literature

	1/4	1/16	1/32	1/64
CsiNet-LSTM Wang & et al. (2018b)	132.7 M	123.2 M	118.5 M	116.1 M
Deep AE Jang & et al. (2019)	3.2 M	2.9 M	2.8 M	2.7 M
CsiNet CsiNetWen & et al. (2018)	2.1 M	1.1 M	0.5 M	0.3 M
CRNet Lu *et al.* (2020)	2.1 M	1.1 M	0.5 M	0.3 M
MarkovNet Liu *et al.* (2021)	2.1 M	1.1 M	0.5 M	0.3 M
MarkovNet-CNN Liu *et al.* (2021)	34.9 K	27.8 K	24.2 K	22.4 K
Our method	**512**	**128**	**64**	**32**

Space Complexity: Table 5.6 displays the number of parameters in different studies in the literature. Our model has a notably smaller number of parameters compared to other methods. The model parameters in our approach serve as the compressed representation of the channel, while other methods require a larger number of weights to perform nonlinear projections onto a lower dimensional space, particularly for complex architectures like LSTM and attention-based models.

Scalability: A problem inherited in traditional autoencoder-based solutions (almost all neural network solutions) is that a given model is designed with a certain input size. For example, for a MIMO system with 32 transmit antennas, single receive antennas, and 1024 subcarriers, the model input is 1024×32. However, during the handover, the UE may connect to another BS with a different number of transmit antennas, 64 for example. Subsequently, the channel size will be 1024×64, and a new model is required. To handle this handover, the UE should maintain a set of models for each possible input size and compression ratio, which is not affordable in the context of current UE resources. On the other hand, the proposed method uses the weights of the model as a compressed representation for the input channel. A model will be trained for each channel and no need to maintain different models for different input sizes. This makes our solution more scalable and efficient. Furthermore, it bridges the gap between academic work and its applicability in real-life scenarios.

5.6.2 Performance Evaluation for Downstream Tasks

After being received at the BS, the CSI information is used in many possible tasks such as link adaptation Kim *et al.* (2015); Ku & Walsh (2014); Hussien & et al. (2021) or beam forming Yue *et al.* (2015); Liu & Lau (2016). These tasks are called **downstream tasks** in the machine learning regime. Therefore, using only the reconstruction distortion (measured by NMSE as given in Eq. 5.12) to evaluate certain CSI compression techniques is not enough. This concern becomes more clear when we obtain high NMSE, while the lost information is very sensitive to the accuracy of the downstream task. In this case, using the reconstruction distortion only as a performance measure can lead to severe degradation in the accuracy of these tasks. To illustrate this point, we evaluated our method in a link adaptation task at the BS. Link adaptation is the process of adaptively tuning various transmission parameters (e.g., modulation and coding scheme (MCS), guard interval, etc.) to maximize the system throughput.

We follow the same experimental setup applied in Hussien & et al. (2021). We tested three propagation environments (namely, Suburban Macro-cell, Urban Macro-cell, and Rural Macro-cell). For each scenario, we used two datasets with 10K and 50K data points each, as indicated in Hussien & et al. (2021). Moreover, we tested four different compression ratios (namely, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$). The accuracy of the link adaptation task is measured by the percentage of retransmissions to the total number of transmissions. We compare the performance in the cases of using the original CSI (without compression) and using the reconstruction generated by our method using the indicated compression ratios. Fig. 5.8 shows the results of each scenario. We can see that the percentage of retransmissions increases by a small margin with increasing the compression ratio. For example, in the worst scenario of *Urban Macro-cell*, the percentage of retransmissions has increased from 14% with the original CSI to 16.15% with a compression ratio of $\frac{1}{32}$. This represents around 15% degradation in the downstream task to save around 97% of the consumed bandwidth in the feedback process.

5.6.3 Bayesian Hyperparameters Optimization

Algorithm 5.2 Bayesian hyperparameters optimization

```
Input: Set of evaluation points $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$
model ← $\mathscr{G}\mathscr{P}$.fit($X, Y$)
While Not converged do
    $x_{new}$ ← acquisition.optimize()
    $y_{new}$ ← evaluate($x_{new}$)
    $\mathscr{D}$.add($x_{new}, y_{new}$)
    model.update($\mathscr{D}$)
End-while
return model.result()
```

The proposed solution has two hyperparameters, namely the number of layers, L, and the number of nodes in each layer $\{n_l\}_{l=1}^{l=L}$. These hyperparameters largely affect the performance of the proposed method. These hyperparameters are constrained by the condition that the total number of weights in the model should equal the dimension of the reduced data. Specifically, given a channel matrix, $\mathbf{H}$, and a compression ratio γ, then the dimension of the reduced data is $d = \|\mathbf{H}\| \times \gamma$, where $\|\cdot\|$ is the cardinality of a matrix (i.e., the total number of elements in

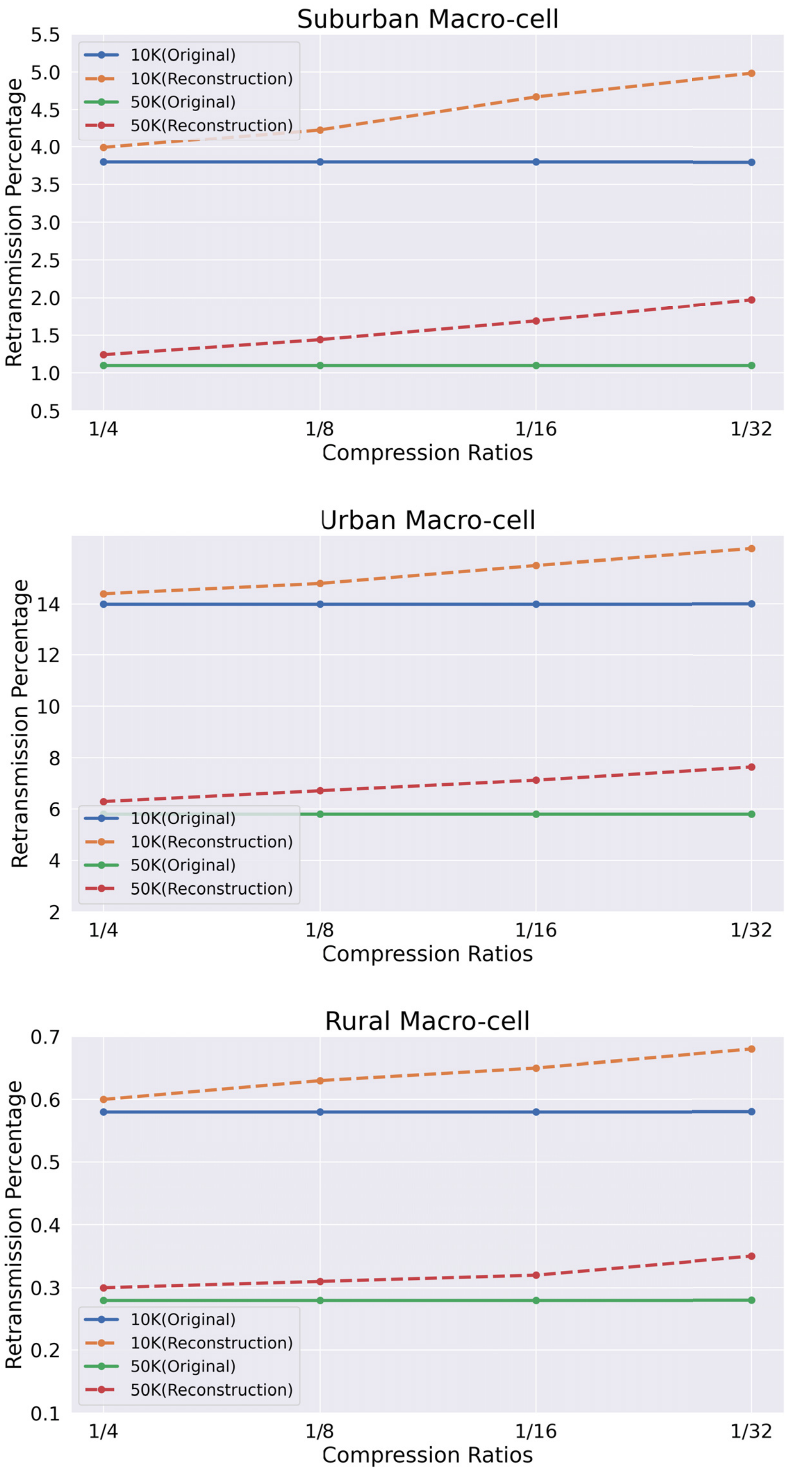


Figure 5.8 The accuracy of using the reconstruction in link adaptation (as one of the downstream tasks) at the BS

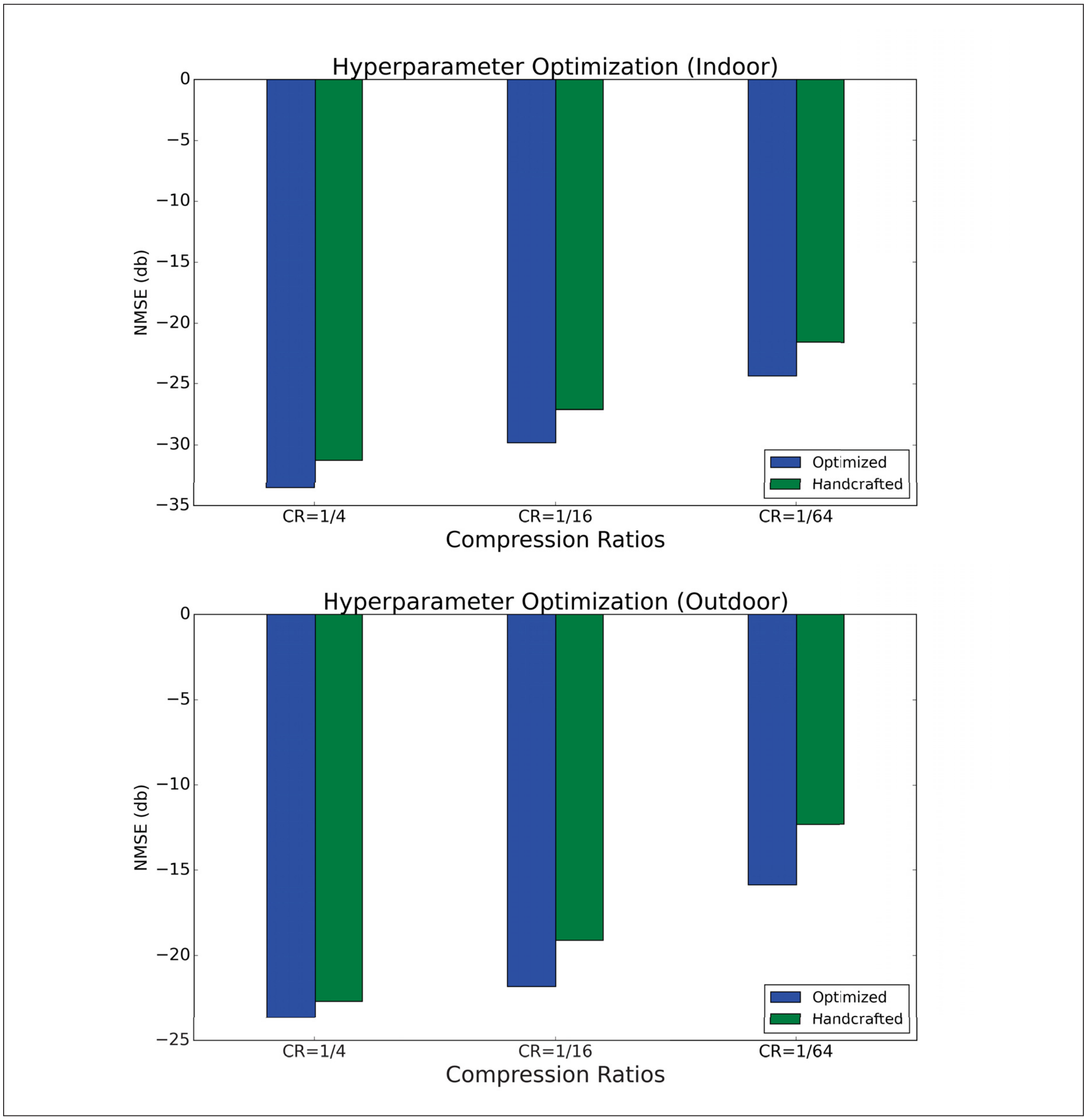


Figure 5.9 A comparison between optimized versus handcrafted models. For optimized models, we used Bayesian optimization to optimize the free hyperparameters (i.e., the number of layers and the number of neurons in each layer)

the matrix). Therefore, the constraint on the values of the hyperparameters can be given as:

$$d = \sum_{l=1}^{L-1} n_l \times n_{l+1}. \tag{5.13}$$

Specifically, using the model weights as the compressed representation for the input channel imposes this constraint. Based on the model architecture, this constraint may be difficult to satisfy. Therefore, we relax it by replacing the equality with inequality such that:

$$\sum_{l=1}^{L-1} n_l \times n_{l+1} \leq d. \tag{5.14}$$

In this case, we guarantee the total number of weights is less than or equal to the target dimension.

The main challenge of hyperparameter optimization is that we do not have a closed form for the objective. However, Bayesian optimization (BO) is a seminal tool for solving these black-box optimization problems Shahriari & et al. (2015). BO does not impose any assumptions on the form of the objective function. It has been widely used to optimize hard-to-evaluate functions and proved to be more efficient than its rivals such as random or grid search. For more information on BO, we refer to Snoek *et al.* (2012).

The existing automated machine learning (AutoML) libraries (e.g., *AutoKeras* Jin *et al.* (2019)) do not support constraining the number of parameters. Therefore, we used *Scikit-optimize* Head *et al.* (2018), a sequential model-based optimization library for BO, to build our own framework. We used the expected improvement (Eq. 5.15) as our acquisition function. We set the number of calls to 10, this means that ten different combinations will be evaluated. We evaluated the BO on an indoor channel.

$$EI(x) = \mathbb{E}(f(x) - f(x^+)), \tag{5.15}$$

where x^+ is the best point observed so far.

Fig. 5.9 compares the performance of optimized models with handcrafted models. We measure the performance of each model using NMSE given in (5.12). We can see the advantage of hyperparameter optimization in the achieved NMSE. More improvements can be achieved by increasing the number of calls, which means we try more combinations.

An increasing trend in the improvement achieved by the optimized model can be observed in Fig. 5.9 (outdoor scenario). Specifically, we can see the optimized model outperforms the handcrafted model by a large margin in the outdoor scenario compared with the indoor scenario. This margin increases with increasing the compression ratio. This means that the advantage of the Bayesian optimization framework increases by increasing the complexity of the compression problem (i.e., using more complex propagation scenarios or increasing the compression ratio).

5.6.4 Analysis of Failure Cases

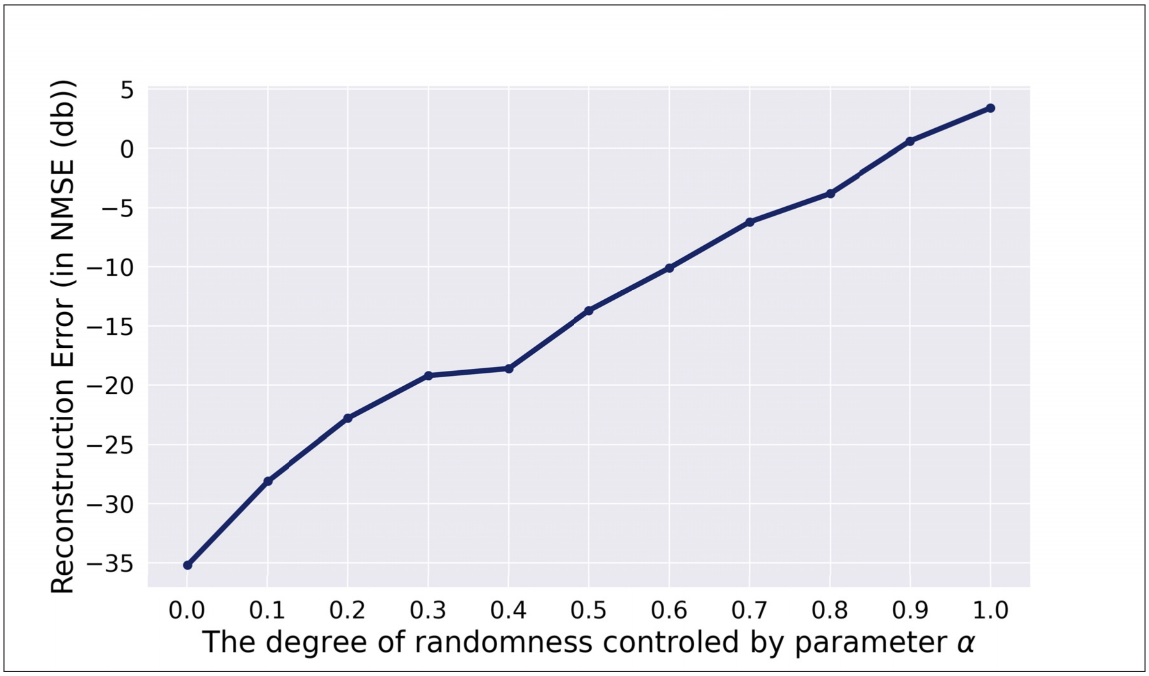


Figure 5.10 The reconstruction error for various randomness degrees

It is well known that neural networks learn the patterns present in the data. Projecting this to our problem, this requires the CSI matrix to have some correlation between its dimensions for the method to be effective. As a result, the proposed method is expected to fail for channels composed of completely random numbers, especially with a shallow network. However, a deeper neural network that can memorize the input data may perform better, although it may result in a number of weights larger than the input channel, compromising the concept of compression. In CSI compression, the indices of the channel represent the physical characteristics of the underlying wireless environment. Therefore, there is a strong correlation between the channel indices and the elements in the channel matrix (i.e., the channel gain of each subcarrier).

To test this idea, we generate a convex combination between one real and one random channel, namely **H** and **R** respectively, according to Eq. (5.16):

$$\mathbf{C} = \alpha \mathbf{R} + (1 - \alpha)\mathbf{H}. \tag{5.16}$$

Both channels are normalized to fall between [0,1]. The parameter α determines the level of randomness in the resulting channel, **C**. We tried 10 different values for α, ranging from 0 to 1 with a step size of 0.1. The results are displayed in Fig. 5.10, which shows that the reconstruction error increases as randomness in the channel increases. The shallow network successfully learned a good representation of the real channel ($\alpha = 0$), but failed to do so for a completely random channel ($\alpha = 1$).

5.7 Conclusion

To achieve efficient data integration for industry 5.0 applications, the captured data should maintain a high degree of synchronization for error-free transmissions. Satisfying these requirements necessitates the base station (BS) to acquire the instantaneous channel state information (CSI) of each node in a process called CSI feedback. To save bandwidth and power, the CSI should be compressed before transmission. In this work, we present a new learning-based compression technique for CSI in massive MIMO systems. Unlike prior work which uses autoencoder-based architectures for compression via nonlinear projection, we introduce a novel approach based on information theory. Our method leverages neural networks to learn a nonlinear sufficient statistic for the input channel. We use instance-aware optimization to train a shallow network to approximate the statistic for each channel sample which reduces the approximation error (variance) and increases the estimation error (bias). Moreover, we provide an error-bound analysis for the proposed method. We also implement a Bayesian optimization framework to determine the best model for the given channel distribution. Our method outperforms other state-of-the-art works in terms of reconstruction loss and model size. It also demonstrated robust performance in downstream tasks such as link adaptation.

PART 3: Link Adaptation

CHAPTER 6

TOWARDS MORE RELIABLE DEEP LEARNING-BASED LINK ADAPTATION FOR WIFI 6

Mostafa Hussien, Mohammed F. A . Ahmed, Ghassan Dahman, Kim Khoa Nguyen,
Mohamed Cheriet, and Gwenael Poitau
École de technologie supérieure (ÉTS), Univeristy of Québec, Canada

Paper published in IEEE Conference on Communications (ICC), June 2021

Abstract : The problem of selecting the modulation and coding scheme (MCS) that maximizes the system throughput, known as link adaptation, has been investigated extensively, especially for IEEE 802.11 (WiFi) standards. Recently, deep learning has widely been adopted as an efficient solution to this problem. However, in failure cases, predicting a higher-rate MCS can result in a failed transmission. In this case, a retransmission is required, which largely degrades the system throughput. To address this issue, we model the adaptive modulation and coding (AMC) problem as a multi-label multi-class classification problem. The proposed modeling allows more control over what the model predicts in failure cases. We also design a simple, yet powerful, loss function to reduce the number of retransmissions due to higher-rate MCS classification errors. Since wireless channels change significantly due to the surrounding environment, a huge dataset has been generated to cover all possible propagation conditions. However, to reduce training complexity, we train the CNN model using part of the dataset. The effect of different subdataset selection criteria on the classification accuracy is studied. The proposed model adapts the IEEE 802.11ax communications standard in outdoor scenarios. The simulation results show the proposed loss function reduces up to 50% of retransmissions compared to traditional loss functions.

keywords :

Link adaptation, multi-class classification, WiFi 6

6.1 Introduction

Nowadays, dynamic resource allocation and link adaptation techniques have been incorporated into different wireless standards to support the quality of service (QoS) requirements while serving the increased number of users Xu & et al. (2019). Link adaptation represents a key element in determining the system's latency and throughput performanceShariatmadari & et al. (2016). Fortunately, machine learning is anticipated to provide viable solutions to the link adaptation challenges in wireless systemsO'Shea & Hoydis (2017).

In the literature, the link adaptation problem has been modeled either as a reinforcement learning problem Saxena & et al. (2019); Mismar *et al.* (2019), or as a multiclass classification problem where the class labels represent different modulation and coding scheme (MCS) combinations Elwekeil & et al. (2018); Karmakar & et al. (2019); Dong & et al. (2018); Li & et al. (2019); Blanquez-Casado & et al. (2019). According to this modeling, each data point can belong to a single class and a supervised machine learning model can be trained to select the ideal MCS based on the training data. However, supervised models, generally, have a certain level of accuracy Jagannath & et al. (2018). In this case, failing to predict the ideal MCS has unpredictable implications on the system throughput. In fact, predicting a higher-rate MCS will result in a failed transmission and, consequently, a retransmission is required which largely degrades the system throughput. These problems come from the fact that modeling the problem as a multiclass classification has no control over what the model can predict in failure cases. Now the question is, if the model failed to predict the optimal MCS, can we train it to predict a suboptimal one?

To answer this question, we model the link adaptation problem, for the first time, as a multi-label multi-class classification. In this modeling, a data point is allowed to belong to more than one class at the same time (all the successful MCS in the AMC problem). Therefore, the model learns to predict not only the optimal MCS but also all suboptimal ones. Such a modeling approach gives more control to what the model learns from the training phase and what it can predict in failure cases. However, we need to enforce the model to avoid predicting higher-rate MCSs that may produce retransmissions. To solve this issue, we propose a new loss function that adds more penalization to such cases. The proposed loss function reduces the number of retransmissions compared to the traditional crossentropy loss function, which is widely employed in the literature. Fig. 6.1 shows an overview of the proposed system.

As wireless channels vary significantly according to the surrounding environment, a huge dataset is required to cover all possible channel variations. However, it is computationally expensive to utilize all the samples for training. In this work, we examine different selection criteria for the training dataset. The selection criteria are based on domain knowledge and our understanding of the nature of wireless channels. For orthogonal frequency-division multiplexing (OFDM)-based systems, we assume an interference-free, noise-free, single-user, and single-input single-output setup. In this case, the delay dispersion of the channel is the decisive factor on the MCS selection. Hence, instead of randomly selecting the training subdataset, we select the subdataset that comprises a uniform (or as close as possible to a uniform) distribution of the channel's delay dispersion behaviors. Given that the *channel dispersion behavior* is not easy to be *fully* characterized, for such selection to take place, we employ well-known criteria characterizing the delay dispersion such as root-mean-square delay spread and window delay spread.

The contributions of this work can be summarized as follows:

- We modeled the problem of AMC as a multi-label multi-class classification problem. The model is trained to predict all the possible labels for successful transmission (including the optimal MCS and suboptimal ones).
- We employed a convolutional neural network (CNN) with an innovative loss function. The proposed model allows controlling what transmission parameters combination to predict when failing to predict the optimal one.
- We studied the impact of training subdataset selection criteria on the AMC problem and highlighted the corresponding effect on classification accuracy.

6.2 Problem Formulation, Dataset Generation, and Training Subdataset Selection

6.2.1 Problem Formulation

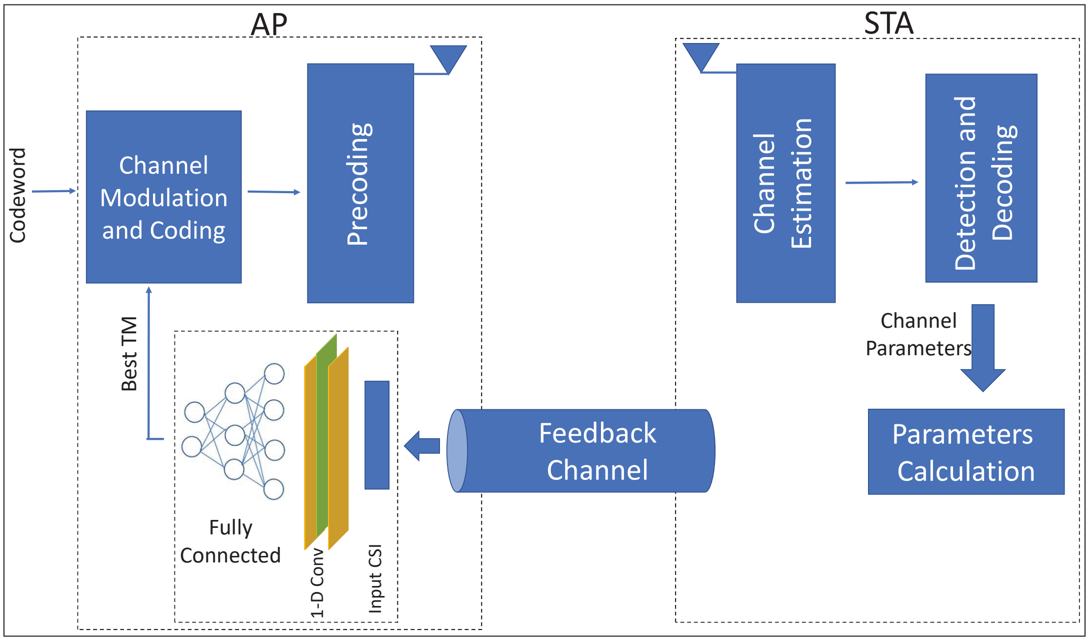


Figure 6.1 System Overview

Table 6.1 IEEE 802.11ax bitrate for different single user TMs.

MCS	N_s	Modulation	Coding	20MHz	
				0.8 GI	3.2 GI
0	1	BPSK	1/2	8.6	7.3
1	1	QPSK	1/2	17.2	14.6
2	1	QPSK	3/4	25.8	21.9
3	1	16-QAM	1/2	34.4	29.3
4	1	16-QAM	3/4	51.6	43.9
5	1	64-QAM	2/3	68.8	58.5
6	1	64-QAM	3/4	77.4	65.8
7	1	64-QAM	5/6	86	73.1
8	1	256-QAM	3/4	103.2	87.8
9	1	256-QAM	5/6	114.7	97.5
10	1	1024-QAM	3/4	129	109.7
11	1	1024-QAM	5/6	143.4	121.9

Assume we have **C** different combinations of **MCS** and guard intervals, **GI**, each of them called a transmission mode, (**TM**). The TMs are indexed as $i \in I \subset \mathbb{N}$, where the cardinality of I is the number of available combinations. The index, i, hereafter referred to as the class distinctly maps to a combination of **MCS** and **GI**. We adopt the IEEE 802.11ax standard for a single-input single-output system at 0.8 and 3.2 guard intervals with a fixed bandwidth of 20 MHz as shown in table 6.1. Therefore, in terms of multi-label multi-class classification, link adaptation is the problem of selecting all the class labels, i, to which a certain channel realization belongs. Thus, for a certain channel realization ch_n, the classifier selects all the labels, i, corresponding to all valid transmission modes TM_i. Then, we can express the classifier function as a function F that maps a channel realization ch_n to a set of labels $y \subset \{1,2,\ldots,C\}$ as:

$$F(ch_n) = y = \{i : TX(ch_n, TM_i) = 1\}, \tag{6.1}$$

where $TX(ch_n, TM_i) = 1$ when transmitting a packet through a channel given by ch_n with transmission configuration given by TM_i is successful, and zero otherwise. From the predicted TMs, we select the TM corresponding to the highest data rate. As shown in Fig. 6.1, a user station (STA) sends the estimated channel state information (CSI) to the access point (AP). The AP then uses the received CSI to adapt the transmission parameters for the next transmission.

6.2.2 Datasets Generation

We selected four scenarios with diverse delay dispersion characteristics: urban micro-cell, suburban macro-cell, urban macro-cell, and rural macro-cell. Using the *Matlab* WINNER II toolboxBultitude & Rautiainen (2007), for each scenario, 50,000 channels are generated. For each channel, we use the *Matlab* IEEE 802.11ax toolbox to simulate transmitting a packet using all available TMs. We split the generated channels into 80% training and 20% testing.

6.2.3 Selection of Training Subdatasets using Different Delay Dispersion Criteria

The training subdatasets are constructed using two approaches: random selection criteria and different delay-spread-based selection criteria. Based on the random approach, Cases 1 & 2 are identified, and based on the delay-spread approach, Cases 3, 4, & 5 are identified.

The random selection criteria (Cases 1 & 2)

The random approach is applied in the following two ways:

- Case 1, Random Full Dataset (RandomFD): all data points (i.e., a total of 160,000 data points; 40,000 points from each of the four scenarios) are used for training.
- Case 2, Random Partial Dataset (RandomPD): the training subdataset is composed of data points selected randomly and equally from each scenario.

RandomFD represents a reference case where all data points are used for training, and RandomPD is the typical widely-used way of reducing the number of data points through random selection.

The delay-spread-based criteria (Cases 3, 4, & 5)

The delay-spread-based selection approach is applied to select different training subdatasets each of which has the same number of data points as RandomPD. Unlike RandomPD, the data points of the built subdatasets are selected to represent the full delay dispersion behavior of RandomFD. Using this approach, from the total 160,000 available data points, we select the subdataset points such that the distribution of the delay dispersion metric will be as close as possible to uniform.

Let's assume $RandomFD_i$ to be the i^{th} data point in the RandomFD dataset; $\mathscr{S}(RandomFD_i)$ is its corresponding delay dispersion evaluated based on a specific metric of interest, $\mathscr{S}$; $i = 1,2,...,I$ (where I is the total number of data points in RandomFD), and $\min \mathscr{S}(RandomFD)$ &

$\max \mathscr{S}(RandomFD)$ are the minimum and maximum obtained delay dispersion values, respectively, among all the data points of RandomFD. We assume the interval $[\min \mathscr{S}(RandomFD)$, $\max \mathscr{S}(RandomFD)]$ to be divided into Z equal disjoint sub-intervals. We define the histogram of $\mathscr{S}(RandomFD)$ as the function that counts the number of delay-spread observations, n_z, that fall into the z^{th} sub-interval, where $z = 1, 2, \ldots, Z$, and n_{min} & n_{max} are the minimum and maximum number of observations, respectively, obtained per sub-interval using the full dataset i.e., RandomFD.

Our proposed delay-spread-based approach to select a subdataset from RandomFD given a histogram, m_z, is as follows.

$$\begin{aligned} \max_{n_{max} \geq x \geq n_{min}} \quad & \sum_{z=1}^{Z} m_z \\ m_z = {} & min(x, n_z) \\ \text{s.t.} \quad & \sum_{z=1}^{Z} m_z \leq T, \end{aligned} \tag{6.2}$$

where T is the total number of data points in the selected subdataset.

The value of x determines the maximum number of data points at each of the Z intervals, which results in selecting a subdataset with a histogram that exhibits a tendency toward having a uniform distribution of the delay dispersion behavior over the $[minFD, maxFD]$ range. The possibility of ending up with a perfect uniform distribution increases as the number of data points in RandomFD increases.

Based on the applied delay-spread metric (i.e., $\mathscr{S}$), which is our design criterion, we can now define the differences among Case 3, Case 4, and Case 5 of the studied cases.

- Case 3, **r**oot-**m**ean-**s**quare delay spread Partial Dataset (rmsPD). In this case, the training dataset is selected using the delay-spread metric defined as the normalized second-order moment of the delay profile of the channels.

- Case 4, **w**indow (40%) delay spread Partial Dataset (W40%PD). In this case, we characterize the delay dispersion using the delay window parameter which is defined as "the length of the middle portion of the power delay profile containing a certain percentage of the total power found in that impulse response" (p. 4, ITUR-R). Here we use the 40% as our design criterion.

- Case 5, **w**indow (70%) delay spread Partial Dataset (W70%PD). In this case, we use the same definition of the delay dispersion metric as in Case 4; however, here we use the window that contains 70% of the power of the delay profile.

6.3 Proposed Deep-Learning Approach for AMC

The convolutional neural networks (CNNs) have shown superior performance in different domains including computer vision, natural language processing, speech synthesis, etc O'Shea & Hoydis (2017). One main advantage of CNNs is their proven capabilities in processing raw data. This advantage eliminates the burdens of data pre-processing. Inspired by this, we propose a CNN-based approach for AMC in IEEE 802.11ax.

6.3.1 CNN Model

The proposed deep convolutional neural network (DCNN) includes convolutional layers, average pooling layers, and fully-connected layers. Typically, the first hidden layer is a convolutional layer with 20 filters. The second hidden layer is a convolutional layer of 32 filters, followed by an average pooling layer with a pool size of 4. Then, another convolutional layer is added with 64 filters followed by an average pooling layer with a pool size of 2. A convolutional layer consisting of 32 filters is added, followed by an average pooling layer with a pool size of 2. For all convolutional layers, every filter has a size of 10 × 2, with ReLU activation, $F(x) = \max(x, 0)$. After the 4 convolutional layers, there are 2 fully-connected layers. The fully-connected layers contain 50 and C neurons respectively, where C is the number of available TMs. Since one channel can belong to many classes at the same time, we used *Sigmoid* activation function (6.3) in the output layer to approximate the multinomial distribution of the class labels. To relieve the effect of overfitting, an $l2$ regularizer is added to the last two layers.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6.3}$$

For training the model, an *Adam* optimizer Kingma & Ba (2014) is adopted along with our customized loss function (section 6.4). The DCNN is trained for 1000 epochs with a batch size of 128. After training the DCNN, it is deployed for predicting the appropriate TMs.

6.3.2 Dataset Description

Consider a labeled dataset consisting of pairs of x and y where x represents different CSI in different selection cases described in subsection 6.2.3. The label vector y is a vector in $\{0,1\}^C$ where C is the number of the available TMs (i.e., the number of classes). If the i^{th} position in the label vector of the j^{th} data instance is set to one, this indicates that transmission over a channel with CSI equal to j^{th} CSI in the dataset using the i^{th} transmission mode will result in a successful transmission. In the same way, 0 indicates a failed transmission. In our experiments, the label vector is 24^{th}-dimensional vector representing the different available combinations of MCS and GI.

6.3.3 Evaluation Metrics

To evaluate the proposed model in the context of communication systems efficiency, we applied two system-specific evaluation metrics, namely, data-rate loss (DRL) and the number of retransmissions (NR). We define δ as:

$$\delta = R(\overline{TM}_i) - R(TM_i), \tag{6.4}$$

where $R(\cdot)$ is a function that maps a TM to the data rate associated with this TM, TM_i is the optimal TM given in the dataset, and $\overline{TM}_i$ is the predicted TM. A positive value of δ means predicting TM with a rate higher than the optimal one. This implicitly incurs a retransmission. The number of retransmissions is given by NR metric. A negative value of δ implies that the model predicts suboptimal TM, which leads to a rate loss. The difference between the data rates of TM_i and $\overline{TM}_i$ is given by DRL.

6.4 Proposed Customized Loss

6.4.1 Why we need a customized loss

The traditional loss function used in multi-label multi-class classification problems is crossentropy (6.5).

$$\mathbf{CE}(y,\hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i), \tag{6.5}$$

where C is the total number of classes, which equals to the dimension of y. We can see that the function in (6.5) treats all wrong predictions equally which is not relevant to the considered AMC problem. We can see that equation (6.5) pushes the model toward learning the true distribution of class labels. Although this is the ultimate goal of any classifier, in some cases we aim to emphasize a certain types of errors (false positives or false negatives).

Definition 6.1. *The label vector of a data instance i is denoted as y_i. The set of positive indices in y_i denoted as $y_i^+ = \{j : y_i(j) = 1\}$, the set of negative indices denoted as $y_i^- = \{j : y_i(j) = 0\}$ where $y_i(j)$ is the j^{th} index in the vector y_i and $j \in \{1,2,\ldots,C\}$.*

Definition 6.2. *The predicted label vector for a data instance i is $\hat{y}_i$. The set of predicted positive indices in $\hat{y}_i$ is denoted by $\hat{y}_i^+ = \{j : \hat{y}_i(j) = 1\}$, and the set of predicted negative indices is $\hat{y}_i^- = \{j : \hat{y}_i(j) = 0\}$ where $\hat{y}_i(j)$ is the j^{th} index in the vector $\hat{y}_i$ and $j \in \{1,2,\ldots,C\}$.*

Definition 6.3. *Given a classifier f, the false positive, and false negative are defined as:*

$$fp(f) = \sum_{j \in \hat{y}_i^+} 1 : j \in y_i^-$$

$$fn(f) = \sum_{j \in \hat{y}_i^-} 1 : j \in y_i^+$$

In the problem under consideration, a false positive in a higher-rate MCS may lead to retransmission, which is very costly in terms of bandwidth resources. However, a false negative indicates selecting a lower-rate TM, which can be tolerated than retransmission. For this reason, we aim to design a loss function that emphasizes on false positives more than false negatives.

6.4.2 Proposed Loss

We propose a new customized loss function that adds more penalization on false positive predictions. Since the proposed loss function emphasizes on false positives, we named it Crossentropy+, $\mathbf{CE}_+$. The new loss is given by:

$$\mathbf{CE}_+ (y, \hat{y}) = \mathbf{CE} (y, \hat{y}) + \phi (y, \hat{y}), \tag{6.6}$$

where $\mathbf{CE} (y, \bar{y})$ is the traditional crossentropy given in (6.5) and $\phi (y, \hat{y})$ is an extra penalization term for false positive predictions given by:

$$\phi (y, \hat{y}) = \beta \times \sum_{i=1}^{C} (y_i - 1)^2 \times \hat{y}_i, \tag{6.7}$$

where C is the total number of classes and β is a weight term added to control the credit assigned for the traditional crossentropy term and the newly added term. Setting β to a large value may lead the model to predict $\hat{y} = \{0\}^C$ vector which minimizes the second term and completely ignores the first term. On the other hand, if we set $\beta \leq 1$, the model may ignore it and learns parameters that minimize only the first term of (6.6). We set $\beta = 1.3$ for all the experiments in this work. However, in the future, we can learn a value for β to meet different QoS requirements (may be different for a WiFi public network than for a 5G URLLC network).

6.5 Experimental Results

We organize this section into two subsections: the prediction results of the DCNN model using the different proposed delay-spread-based subdataset selection criteria, and the improved prediction results achieved by adapting the proposed loss function.

6.5.1 Results of AMC using DCNN Model

To evaluate the effect of the training set size, we trained the model with varying set sizes, namely, 10K, 20K, 30K, 40K, and 50K channels, for each selection criterion. We also consider a larger *RandomFD* dataset. For each training set, we test the model using three different scenarios, namely, suburban macro-cell (C1), urban macro-cell (C2), and rural macro-cell (D1).

Fig. 7.3 shows the percentage of retransmissions to the total data points in each test scenario. We can see that, among the different selection criteria, W40%PD obtained the best performance in all the test scenarios. Also note that for all criteria, scenario D1 obtained a higher retransmission rate compared to both C1, and C2. This figure also shows that RandomPD and rmsPD training subdatasets always obtain higher retransmission percentages compared to W40%PD and W70%PD. We observe that the performance is largely improved with increasing the size of the training dataset. However, little or no improvement has been recorded when the size increases from 40K to 50K. According to the VC-dimension theorem Bishop *et al.* (1995), this saturation happens when the number of training data points reaches a threshold, N_{vc}, after which adding more data points does not improve the learning anymore.

Fig. 6.3 shows the percentage of data rate loss obtained using the DCNN-model with different training subdataset selection criteria. As explained in section 6.4, a data rate loss happens when the model predicts a false negative in the index of the ideal TM. The figure shows an inverse trend between the retransmission rate and the data rate loss. However, it is worth noting that since the overall system performance is decided by both: rate loss and retransmission rate, it is more likely to tolerate a reasonable rate loss rather than repeated retransmissions. We can see that W40%PD, which results in the best performance in terms of retransmissions,

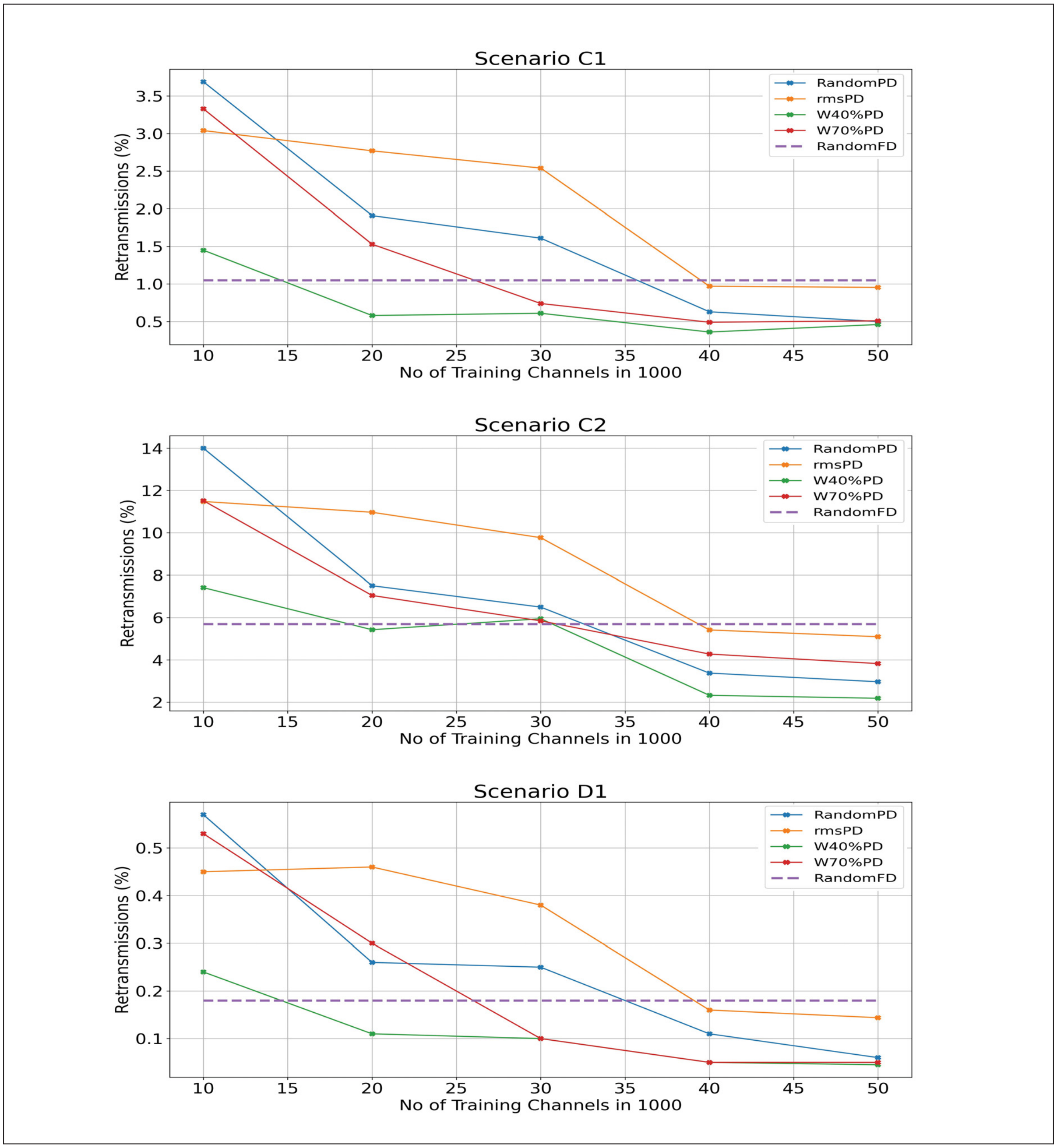
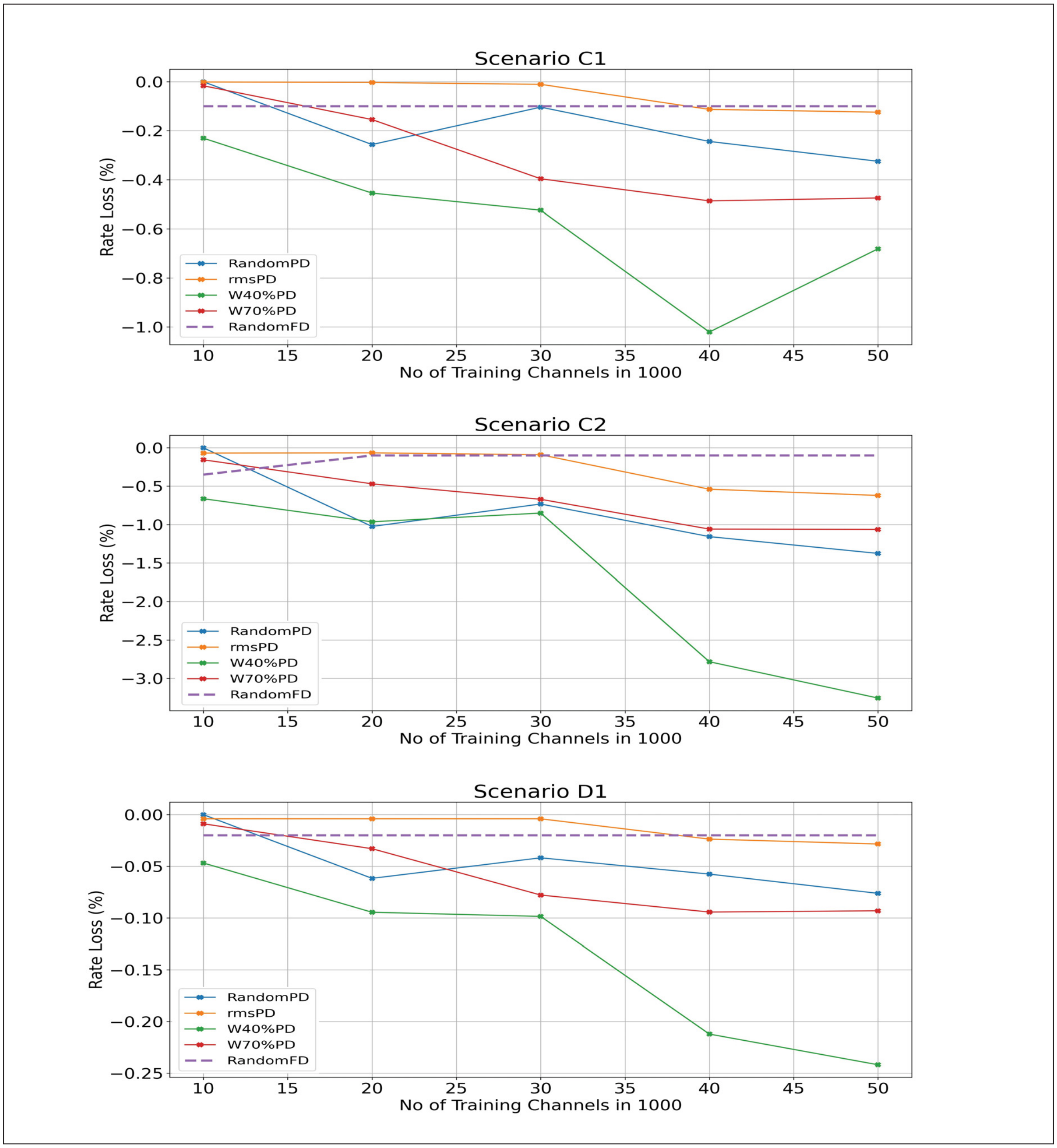


Figure 6.2 The percentage of retransmissions in each test scenario

obtained around -3.1% rate loss in the worst case (scenario C2). Based on these observations, we can conclude that training a model based on W40%PD gives the best performance in the retransmission with acceptable rate loss. Also, the proposed DCNN approach obtained nearoptimal TM selection. However, we can further improve the model performance by introducing the proposed loss function as described in the next subsection.


Scenario C1

Scenario C2

Scenario D1

Figure 6.3 The percentage of data-rate loss in each test scenario

6.5.2 The Performance of the Proposed Loss-Function

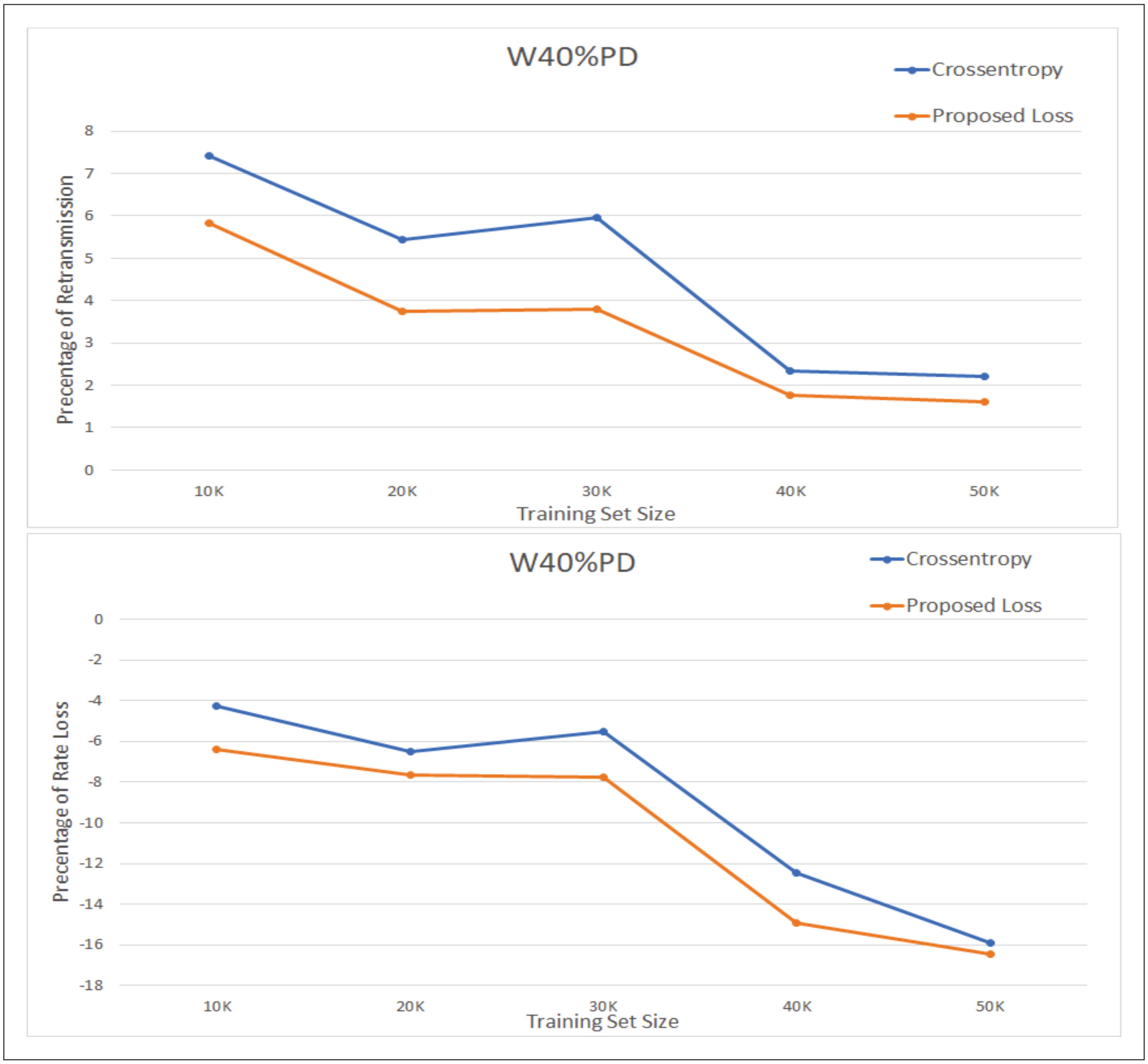


Figure 6.4 The percentage of retransmissions and rate loss for W40%PD in scenario C2 for models trained with crossentropy and our proposed loss function (6.6)

To evaluate the performance of the proposed loss function, we trained a model with traditional crossentropy and our proposed loss functions. To obtain a fair comparison, we used the same model capacity in the two cases. We also fixed all other hyperparameters (e.g., the same number of epochs, initialization, activation, regularizer, optimizer, and learning rate).

Table 6.2 Percentage of retransmission and rate loss for models trained with classical crossentropy loss function (CE) and the proposed loss function (PLoss)

Percentage of Retransmission								
	RandomPD		rmsPD		W40%PD		W70%PD	
	CE	PLoss	CE	PLoss	CE	Ploss	CE	Ploss
10K	14.00	**14.00**	11.48	**10.93**	7.42	**5.84**	11.52	**9.26**
20K	7.51	**6.74**	10.79	**7.47**	5.43	**3.75**	7.05	**6.36**
30K	6.50	**3.72**	9.77	**9.06**	5.95	**3.80**	5.85	**4.40**
40K	3.38	**3.03**	**5.24**	5.26	2.33	**1.76**	4.28	**3.88**
50K	6.63	**3.47**	9.70	**3.57**	2.20	**1.62**	3.98	**3.64**
Percentage of Rate Loss								
10K	0.0	0.0	0.48	0.52	4.29	6.40	1.12	2.82
20K	5.38	6.91	0.51	1.20	6.49	7.68	3.63	4.46
30K	4.76	6.39	0.75	1.06	5.52	7.76	4.69	6.71
40K	7.61	8.81	4.38	4.03	12.46	14.94	7.91	5.75
50K	3.42	5.21	1.01	5.60	15.88	16.46	7.60	8.75

The results of training the model using the two loss functions are shown in Table 6.2. The table shows the number of retransmissions in scenario C2. We selected this test scenario since it has the largest percentage of retransmissions compared to other scenarios, as shown in Fig. 7.3. We can see that the proposed loss function has largely reduced the number of retransmissions under all selection criteria and dataset sizes. The proposed loss function obtained more than 50% improvement over traditional crossentropy in some cases.

Table 6.2 shows the percentage of rate loss for each training set size. We can see that the rate loss using our proposed loss function is larger than that of traditional crossentropy. Given that the model capacity is the same, this can be explained by the fact that reducing the false positives may result in increased false negatives. However, depending on the specifications of the used communication system (specifically the cost of retransmissions compared to rate loss), varying the value of β in (6.7) provides a wide range of fine-tuning to meet different performance requirements.

6.6 Conclusion

A convolutional neural network framework for adaptive modulation and coding (AMC) in IEEE 802.11ax has been presented. We modeled the problem of AMC as a multi-label multi-class problem. The results showed that traditional loss functions are limited in solving such problems. We proposed a new loss function that increases the reliability of the adaptation framework. The proposed loss function proved to outperform the traditional crossentropy function. We also studied the impact of subdataset selection on the model performance. Empirically, we concluded that the window delay 40% subdataset selection criterion and the proposed loss function give the best throughput/reliability compromise.

CHAPTER 7

DATA-CENTRIC AI FOR LINK ADAPTATION: OPPORTUNITIES, CHALLENGES, AND OPEN ISSUES

Mostafa Hussien, Kim Khoa Nguyen, and Mohamed Cheriet

École de technologie supérieure (ÉTS), Univeristy of Québec, Canada, H3C 1K3

Paper Submitted to IEEE Communications Magazine

Abstract: Over the past decade, artificial intelligence (AI) has demonstrated remarkable performance across various domains, including wireless communications. The predominant approach in this field has been the utilization of a conventional workflow known as model-centric AI. Under this paradigm, a dataset is prepared in advance and kept fixed, while a model is developed and iteratively improved to reach a satisfactory performance. Although this approach has yielded promising results in several domains, it primarily emphasizes one component, namely the AI model. Thereby neglecting an essential yet often underestimated resource: the data. The significance of data as a resource cannot be overstated, as it represents 50% of the overall resources involved in an AI model, which consists of both the algorithm and the data. In contrast, data-centric AI approaches adopt a different perspective by assigning a higher priority to data. In this paradigm, the model is held constant, and the dataset is continuously modified and enhanced until a desirable performance is reached. This article aims to explore the emerging data-centric paradigm in AI development. Specifically, we propose a data-centric approach for addressing the wireless link adaptation problem in the IEEE 802.11ax standards. Leveraging our domain knowledge, we introduce relevant metrics for constructing the datasets in this scenario. Our findings demonstrate that the data-centric approach surpasses the conventional model-centric approach, yielding promising results.

Keywords: Data-centric AI, link adaptation, model-centric AI

7.1 Introduction

Over the past decade, the field of artificial intelligence (AI) has experienced a paradigm shift, particularly in the realm of deep learning. The remarkable capabilities of deep learning models have become evident through their exceptional performance across various domains and problem sets. Notably, these models have surpassed human performance in intricate and demanding tasks, such as ChatGPT Du *et al.* (2023). The catalysts behind this revolution can be attributed to advancements in computational power and the growth of vast amounts of data. Consequently, AI techniques have found widespread application in diverse domains, consistently delivering increasingly extraordinary performance. However, it is noteworthy that many of these AI success stories primarily stem from advancements in algorithmic modifications and learning techniques within AI models. Nevertheless, it is imperative to recognize that an AI model comprises two fundamental components: data and model. In the conventional model-centric AI approach, an AI solution is developed by gathering a collection of data points, which undergo minimal preprocessing. Subsequently, the dataset is held fixed, and a model is constructed and trained using this dataset. The model then undergoes multiple iterations of refinement and hyperparameter-tuning until a satisfactory level of accuracy or loss is reached. However, this approach often underestimates the critical role that data can play in enhancing the performance of the model Sambasivan (2021).

Recently, the distinguished AI pioneer, *Andrew Ng*, introduced a novel AI development approach known as *”data-centric AI”*, accompanied by the launch of an international competition centered around this methodology Andrew Ng. This approach represents a complete inversion of the traditional model-centric approach, wherein the focus lies on refining and enhancing the data while keeping the model fixed, thereby improving the obtained performance. Notably, this data-centric paradigm has exhibited remarkable advancements in model performance compared to the conventional model-centric approach. The model-centric approach has exerted significant influence over various aspects of research, competitions, development tools, and even the AI job market. Consequently, most development tools have predominantly emphasized model development, with relatively less attention dedicated to data collection and preparation. However, a current trend is emerging, aiming to reach a balance between the efforts expended on data and model development. It is believed that models have now attained a level of maturity enabling them to learn complex functions proficiently. Conversely, data still lacks this level of maturity, necessitating further investigation to uncover its untapped potential. During a recent presentation, *Andrew Ng* highlighted the potential of the data-centric approach in addressing the steel-sheet defect detection problem. Ng and his team reported a significant 16.9% improvement in accuracy when employing the data-centric approach, surpassing the performance of the baseline model. In contrast, the traditional model-centric approach failed to yield any improvement beyond the capabilities of the baseline model, likely due to the complexity already achieved by the baseline model itself. This outcome effectively underscores the inherent power of data in conjunction with a specific AI model Bansal *et al.* (2019); Daniels & Heath (2010).

This article delves into the potential of data within the wireless communications domain, specifically in the context of link adaptation (LA) in IEEE 802.11ax protocol standards. The focus lies on introducing a convolutional neural network (CNN) model that adheres to the principles of the data-centric approach. In this endeavor, we leverage domain expertise to propose and evaluate diverse criteria for generating datasets, encompassing both the widely employed random criterion and novel alternatives for LA problems. By training the same model on datasets constructed using these various criteria, we highlight the efficacy of this approach in AI development. Additionally, we shed light on key areas of research that warrant further investigation in future endeavors.

The subsequent sections of this article are structured as follows: an introductory overview of the link adaptation problem is provided, offering insights into its significance and challenges. Subsequently, a comprehensive discussion is presented on two distinct approaches to AI development, highlighting the conventional model-centric approach alongside the proposed data-centric approach. Within the context of link adaptation, the data-centric approach is explored, accompanied by a thorough presentation of the obtained results, clarifying its effectiveness. Lastly, the article delves into a discussion of the principal opportunities and challenges associ-

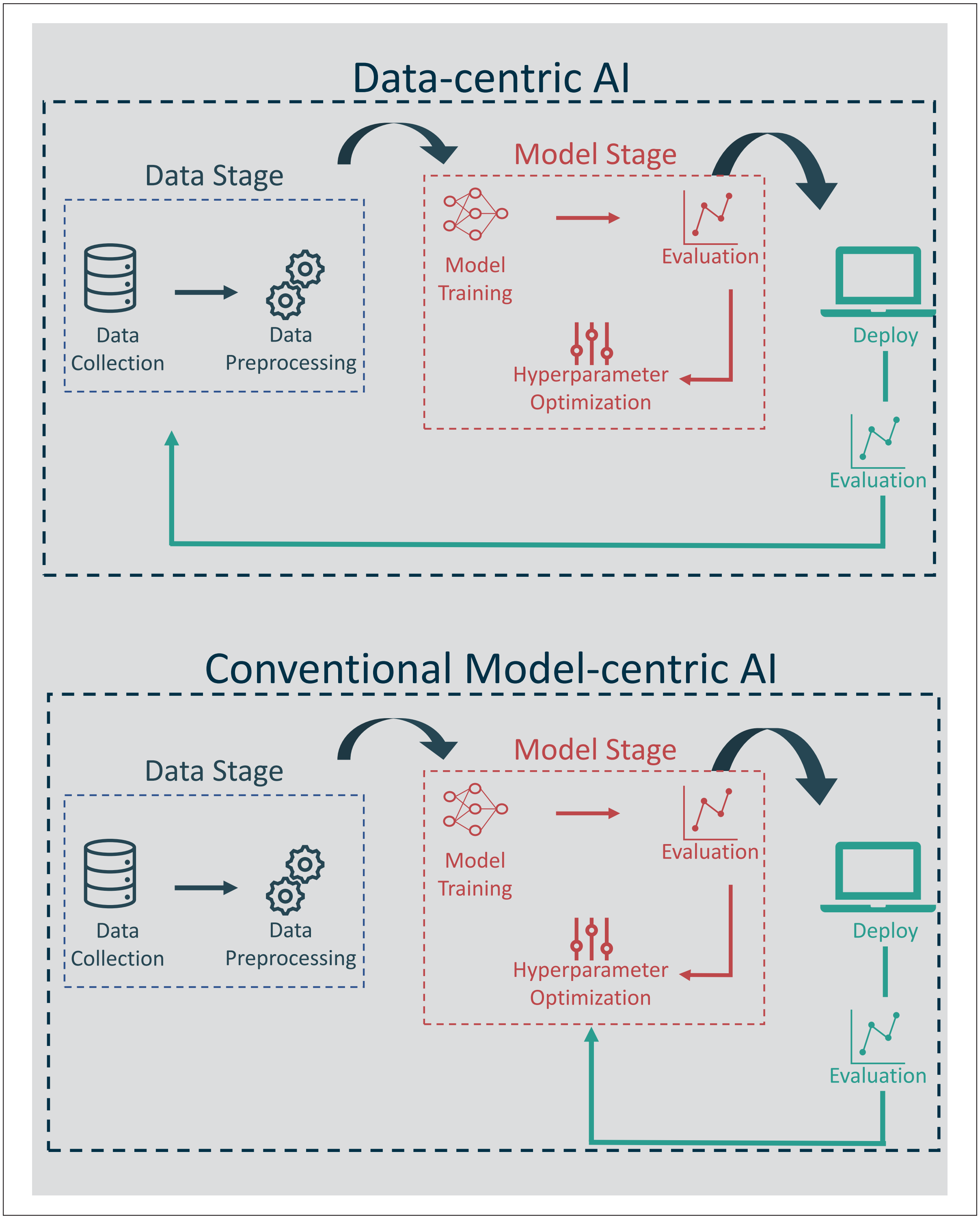


Figure 7.1 Data-centric Vs model-centric AI

ated with the data-centric approach, providing valuable insights into its practical implications. Moreover, potential avenues for future research are suggested, aiming to foster further investigation and advancements in this field.

7.2 Model-centric Vs Data-centric AI

As previously mentioned, the model-centric AI approach is primarily focused on exploring the depths of the model itself, primarily through meticulous examination of potential areas for improvement within the algorithm, see Fig. 7.1. This can involve exploring diverse hyperparameter configurations or introducing novel components to the learning approach. Throughout this process, the dataset remains unchanged and undergoes no further processing. However, we contend that the development of effective AI solutions necessitates a certain level of compromise between the model-centric and data-centric approaches. Moreover, considering the greater potential inherent in the data, it becomes imperative to dedicate increased attention to the critical aspects of data collection, labeling, and cleaning. In numerous real-world scenarios, it has been observed that significant enhancements in model performance can be achieved through diligent data processing, surpassing the gains obtained through model-centric improvements.

Prior to the advent of the era of "Big Data," the primary concern in building AI models revolved around the size of the dataset. However, with the proliferation of the Internet of Things (IoT) and advancements in sensing technologies, the quantity of data has ceased to be a major obstacle. Instead, the focus has shifted towards the quality of the data, as label consistency and the absence of noise can significantly enhance the performance of models. While achieving this objective may pose challenges in certain domains, such as segmentation or detection, where data collection can be costly, it is relatively simpler in domains like wireless communications. This is due to the availability of a multitude of simulators that facilitate data generation and labeling. Furthermore, the role of domain knowledge becomes crucial once again, playing a vital part in constructing AI models. Given that effective data cleaning necessitates a strong understanding of the problem domain, the involvement of domain experts is imperative in this process, as they can contribute to the development of more robust and powerful AI models.

7.3 Link Adaptation for IEEE 802.11ax

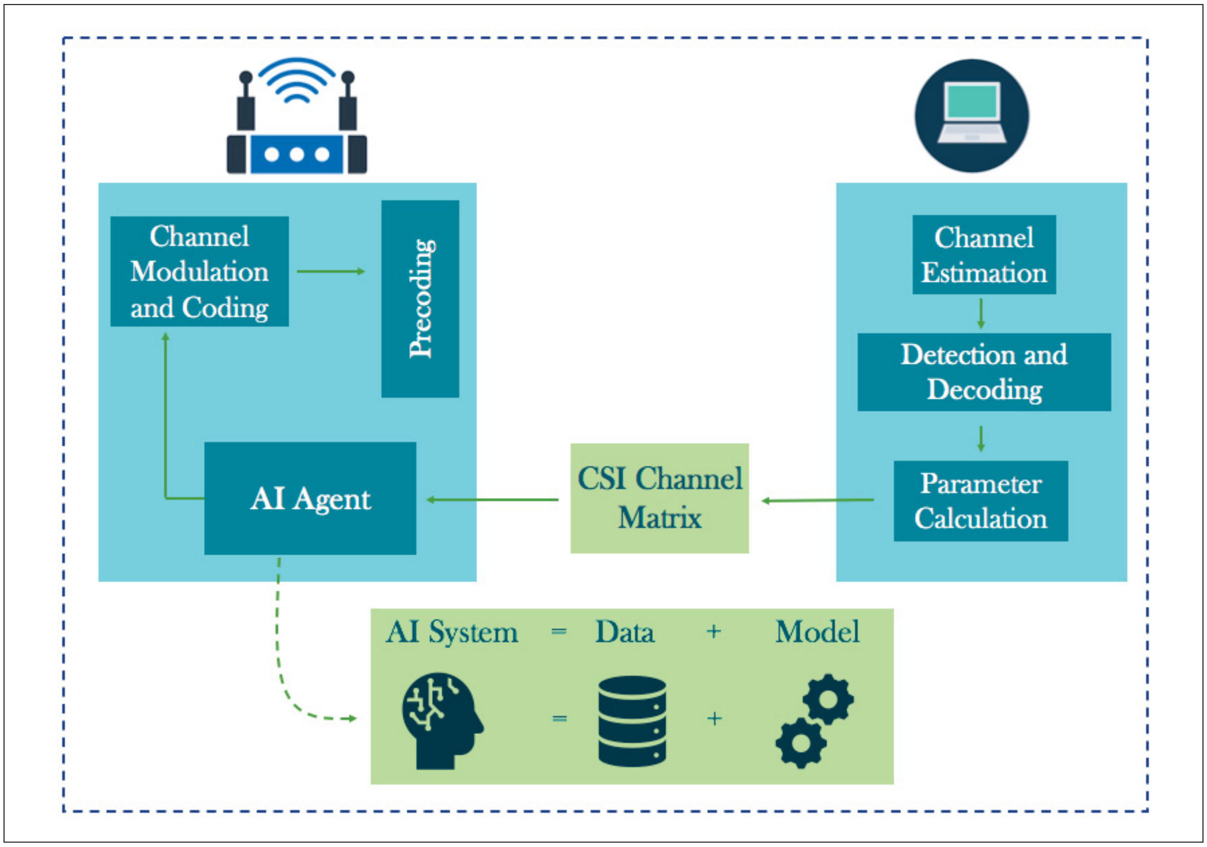


Figure 7.2 A data-centric AI framework for link adaptation in WiFi 6.

With the growth of connected nodes and the rapid expansion of IoT applications, the demand for intelligent communications has become increasingly inevitable. To meet the requirements of quality of service (QoS) in such environments, the adoption of techniques like link adaptation and dynamic resource allocation is essential Tong *et al.* (2023). In a typical wireless communication scenario, it is crucial to adapt various transmission parameters based on the prevailing channel conditions. Examples of such parameters include modulation and coding schemes (MCS), guard interval (GI), number of spatial streams, etc. However, wireless channels are subject to continual variations over time, making the selection of these parameters on a global basis a challenging task. Thus, a dynamic adaptation mechanism is essential. Link adaptation, which involves the selection of appropriate transmission parameters in response to dynamic channel conditions, plays a critical role in optimizing the latency and throughput performance of the system Mota *et al.* (2019).

Traditional approaches to link adaptation have often relied on simple techniques such as lookup tables, where a basic channel quality metric is mapped to specific parameter configurations. Typically, the channel quality indicator (CQI) has been widely employed as a metric to evaluate channel quality. In earlier WiFi protocol standards like IEEE 802.11b, 802.11c, or 802.11g, these conventional techniques yielded satisfactory performance improvements. However, in more complex standards like IEEE 802.11ax, which encompass a vast number of parameter settings, these basic approaches fall short of meeting expectations Yang *et al.* (2019). Moreover, utilizing a basic channel quality metric like CQI, which represents a scalar value, restricts the ability of adaptation techniques to perceive the dynamics of the wireless channel. Instead, channel state information (CSI) offers a more comprehensive understanding of the channel characteristics. Consequently, there is a growing demand for more sophisticated techniques that can overcome the limitations of these conventional approaches Elwekeil & et al. (2018). Fig. 7.2 shows a typical design of a link adaptation system based on the CSI feedback.

In light of these considerations, there arises an inherent necessity for more effective techniques to address the emerging challenges. Machine learning (ML) techniques have demonstrated their effectiveness in tackling various complex problems, such as semantic segmentation, automatic image captioning, sentiment analysis, and others. Within the domain of wireless communications, ML has found widespread application in tasks such as channel classification, CSI compression, beam selection, etc. Consequently, supervised and reinforcement learning methods have been extensively adopted to develop more efficient link adaptation techniques for modern communication systems. Unsurprisingly, ML-based approaches have exhibited superior performance compared to conventional lookup tables in terms of throughput and reliability. However, it is worth noting that all existing techniques described in the literature have adhered to a model-centric approach for AI, commonly referred to as model-centric AI. Within this framework, the primary focus and contributions are directed towards constructing a high-performing model, with the notion of a "good" model being one that achieves the highest accuracy on the testing dataset Dong & et al. (2018).

7.4 Data-centric AI Design Issues

Although data-centric AI has several benefits, some design issues must also be considered. Here are some of them:

- Data Quality: Data-centric AI heavily relies on the availability of high-quality and relevant data. However, ensuring data quality can be challenging. Issues such as data incompleteness, inaccuracies, biases, and data drift can affect the performance and reliability of AI systems. Data cleaning, preprocessing, and continuous monitoring are essential to mitigate these issues.
- Data Bias and Fairness: Biases present in training data can lead to biased predictions and discriminatory outcomes. Data-centric AI systems may inadvertently learn and perpetuate biases present in the data, leading to unfair or unethical decision-making. Addressing bias and promoting fairness in AI systems require careful consideration of the data collection process, diverse and representative datasets, and algorithmic fairness techniques.
- Data Privacy and Security: Data-centric AI involves collecting, storing, and analyzing vast amounts of data. This raises concerns about data privacy and security. Organizations must implement robust data protection measures, comply with relevant regulations (e.g., GDPR), and ensure secure data handling practices to protect sensitive information and maintain user trust.
- Data Governance and Ownership: With data-centric AI, questions arise about data ownership, access, and control. Organizations must establish clear data governance frameworks, including data rights, data sharing agreements, and accountability mechanisms. Trans-

parency and consent in data collection and usage become crucial to address concerns related to data control and ownership.

- Data Scalability and Infrastructure: Managing large volumes of data requires scalable and efficient infrastructure. Data-centric AI systems need to handle real-time data streams, process high-dimensional data, and support distributed computing for training and inference. Designing robust and scalable data pipelines, storage systems, and computing infrastructure is essential for effective data-centric AI.

- Interpretability and Explainability: Data-centric AI models, such as deep learning neural networks, are often considered black boxes, making it challenging to understand the rationale behind their decisions. Interpretability and explainability techniques aim to provide insights into the model's internal workings and enable humans to understand and trust AI-generated outcomes. Ensuring transparency and interpretability can be crucial, especially in high-stakes domains like healthcare or finance.

- Channel and Environment Dynamics: Wireless channels exhibit variability due to factors such as fading, interference, and mobility. Dynamic environments pose challenges for data-centric AI models that rely on historical data and may struggle to adapt to changing channel conditions. Designing AI models that can adapt in real-time to varying channel conditions and environmental dynamics is essential for reliable and robust wireless communication systems.

- Ethical Considerations: Data-centric AI raises several ethical considerations, including issues related to privacy, fairness, accountability, and bias. Ethical design frameworks should be employed to guide the development and deployment of AI systems. Stakeholder engagement, interdisciplinary collaboration, and adherence to ethical guidelines can help address these concerns.

It is important to employ a holistic and multidisciplinary approach that includes domain experts, data scientists, ethicists, and policymakers when addressing these design issues.

7.5 Data-centric AI for Link Adaptation

7.5.0.1 Dataset Description

To embrace the data-centric approach, the model was trained on diverse datasets, each selected based on distinct criteria while maintaining a common structure. A typical dataset utilized for model training consists of a collection of paired instances denoted as x and y, where x represents various channel matrices generated from different communication scenarios. The label vector y is an element of the set $\{0,1\}^C$, where C denotes the number of classes and the classes represent the different transmission modes. Specifically, if the transmission of a packet through a channel described by x_i, using transmission mode TM_j, is successful, the corresponding entry $y_i(j)$ is assigned a value of 1. Conversely, $y_i(j) = 0$ indicates a failed transmission when using the j-th transmission mode with the channel state x_i. Consequently, the label vector maintains a 24-dimensional structure, representing the success or failure of various transmission modes under specific CSI conditions Deng & et al. (2017). Notably, an exhaustive search strategy is employed to construct the ground truth data. Each dataset is divided into an 80% training subset and a 20% testing subset to facilitate model evaluation LeCun *et al.* (2015).

7.5.1 Data-centrism for Link Adaptation

The training subdatasets have been generated utilizing two distinct approaches: *the random* selection criterion and *the delay-spread-based* selection criteria. Delay-spread refers to the difference in arrival times between the earliest and latest arriving signals in a wireless communication channel, indicating the spread or dispersion of signal propagation delays. Employing the random approach, we identified datasets DS1 and DS2 as subdatasets. Conversely, by employing the delay-spread approach, we identified datasets DS3, DS4, and DS5 as additional subdatasets.

7.5.1.1 Datasets DS1 and DS2: Random Sampling

To generate datasets DS1 and DS2, we adopted the random approach as follows:

- **DS1: Complete Random DS**: This dataset represents the complete dataset and contains a total of 160,000 channels; 40,000 channels are sampled from each of the four scenarios. DS1 represents the aggressive scenario where the whole dataset is used for training. Although it is usually better to train on large datasets, this imposes certain restrictions on the capacity of the trained model to prevent overfitting/underfitting problems. Moreover, the required computation resources may also be a bottleneck in the training phase.
- **DS2: Partial Random Dataset**: This is a subset of DS1 and composed of channels selected randomly and equally from each scenario. Specifically, it contains 40,000 channels sampled as 10,000 random channels from each scenario. Random sampling with an equal likelihood for each scenario is a widely adopted technique for reducing the dataset size when the size of the complete dataset is huge and it is difficult to use the whole dataset for training.

7.5.1.2 The Delay-spread-based Criteria (DS3, DS4, & DS5)

This subsection shows how domain knowledge can be exploited to sample subdatasets in a more efficient way. The delay-spread-based selection approach is employed to create distinct training subdatasets, each containing the same number of data points as DS2. In contrast to DS2, the data points in these subdatasets are carefully chosen to capture the complete range of delay dispersion characteristics exhibited by DS1. By utilizing this approach, a subset of data points is selected from the pool of 160,000 available points, ensuring that the distribution of the delay dispersion metric closely approximates a uniform distribution Alamgir *et al.* (2020).

Let $DS1_i$ represent the i^{th} data point in the DS1 dataset, and let $\mathscr{S}(DS1_i)$ denote its corresponding delay dispersion evaluated using a specific metric of interest $\mathscr{S}$. Here, i ranges from 1 to I, where I represents the total number of data points in DS1. The minimum and maximum delay dispersion values obtained among all data points in DS1 are denoted as $\min \mathscr{S}(DS1)$

and $\max \mathscr{S}(DS1)$, respectively. We partition the interval $[\min \mathscr{S}(DS1), \max \mathscr{S}(DS1)]$ into Z equally spaced and non-overlapping sub-intervals. The histogram of $\mathscr{S}(DS1)$ is defined as a function that tallies the number of delay dispersion observations, denoted as n_z, falling within the z^{th} sub-interval. Here, z ranges from 1 to Z. Additionally, n_{min} and n_{max} represent the minimum and maximum number of observations per sub-interval, respectively, obtained using the full dataset DS1. The proposed approach for selecting a subdataset from DS1, based on the histogram m_z, relies on the following delay-spread-based methodology: By setting the value of x, we can determine the maximum number of data points allotted to each of the Z intervals, consequently leading to the selection of a subdataset with a histogram showcasing a tendency toward a more uniform distribution of the delay dispersion behavior across the range $[minFD, maxFD]$. As the number of data points in the DS1 dataset increases, the likelihood of achieving a perfect uniform distribution also increases.

Using the delay-spread as a selection criterion, we sampled three different datasets, namely DS3, DS4, and DS5, as follows:

- DS3: **r**oot-**m**ean-**s**quare delay-spread Partial Dataset (rmsPD). Here, the training dataset is sampled using the delay-spread metric defined as the normalized second-order moment of the delay profile of the channels.

- DS4: **w**indow (40%) delay spread Partial Dataset (W40%PD). In this case, the characterization of delay dispersion revolves around the utilization of the delay window parameter, defined as "the length of the central segment within the power delay profile that encompasses a specific percentage of the total power present in the impulse response" (p. 4, ITUR-R). For our particular investigation, we employed a design criterion of 40% to determine the length of the delay window.

- DS5: **w**indow (70%) delay spread Partial Dataset (W70%PD). We employ the same definition of the delay dispersion metric as that utilized in DS4. However, in this case, we consider the window that encompasses 70% of the power within the delay profile.

7.5.2 Datasets Generation

WINNER II (Wireless World Initiative New Radio) is an international research project conducted under the framework of FP6 (Framework Program 6) initiated by the European Commission. Its primary objective is to develop the radio interface for systems beyond 3G. Operating within the frequency ranges up to 6 GHz, WINNER II encompasses diverse indoor and outdoor propagation scenarios, ranging from urban to rural environments, including a combination of these settings. The project adopts a Geometry-based Stochastic Channel Model (GSCM) classification, whereby the propagation channel is represented as a collection of multipath clusters, with each cluster containing multiple (typically 10 to 30) multipath components. Utilizing extensive measurement campaigns, these multipath components are extracted, analyzed, and subsequently clustered based on their angle-of-arrival, angle-of-departure, delay, and power attributes. Consequently, each propagation scenario is characterized by distinct parameters, such as the number of clusters, their spatial distribution, power levels, and the characteristics of their constituent multipath components. WINNER II offers models for 12 propagation scenarios. In this study, we selected four communication scenarios, namely urban micro-cell, suburban macro-cell, urban macro-cell, and rural macro-cell, each exhibiting diverse delay dispersion characteristics. Employing the MATLAB WINNER II toolbox, we generated 50,000 channels for each scenario. We adopted an exhaustive search strategy to build the dataset labels using *MATLAB IEEE 802.11ax toolbox*.

7.5.3 Model Setup

Since we follow a data-centric approach, we fix the model architecture and change the dataset pipeline to improve the model performance. The fixed architecture includes convolutional layers, average pooling layers, and fully-connected layers. The first hidden layer is a convolutional layer with 20 filters followed by a 32-filters convolutional layer. An average pooling layer with a pool size of 4 is then applied. Another convolutional layer with 64 filters is added followed by an average pool operation with a pool size of 2. Lastly, a convolutional layer with 32 filters is added followed by an average pool with a pool size of 2. A *ReLU* activation has been used in all convolutional layers. This convolutional architecture is followed by 2 fully-connected layers. The fully-connected layers contain 50 and C neurons respectively, where C is the number of classes (available TMs). The output layer in the link adaptation problem approximates a multinomial distribution. Therefore, a *Sigmoid* activation has been applied at the output layer. An $l2$ regularization term has been adopted to avoid the overfitting effect. Overfitting in neural networks refers to a condition where the model learns the training data well, resulting in poor generalization to new, unseen data. The model has been trained to 1000 epochs using *Adam* optimizer Jais *et al.* (2019) with 0.01 learning rate to minimize the *Crossentropy* function. The weight updates were propagated after each of the 128 channels. The best-performing model, in terms of reliability measured by the PER, has been deployed to infer the best TM.

7.5.4 Performance Evaluation

Fig. 7.3 shows the performance of the same model trained on different datasets. A significant improvement has been obtained by training the model on a dataset selected according to the W40% selection criteria in different communication scenarios. The performance of the model trained on a randomly selected subdataset has shown higher PER compared with other models. This emphasis is on the potential of a data-centric approach in wireless problems in general and especially for link adaptation.

7.5.4.1 High-Quality Data or Larger Dataset Size?

An imprecise belief has been established from the literature work that more data is always better. Although large data sizes are one of the strongest points of any AI model, concentrating on collecting larger datasets is not always the magic solution. The quality of the data is an important factor that is consistently underestimated in the literature. Fig. 7.3 shows that, in different scenarios, we can reach the same performance of the 35K dataset (selected according to RMS) by only the 10K dataset (selected according to W40%). This suggests that investing enough time to understand the problem specifics and obtaining clean data can be equivalent to/or more important than generating more data points.

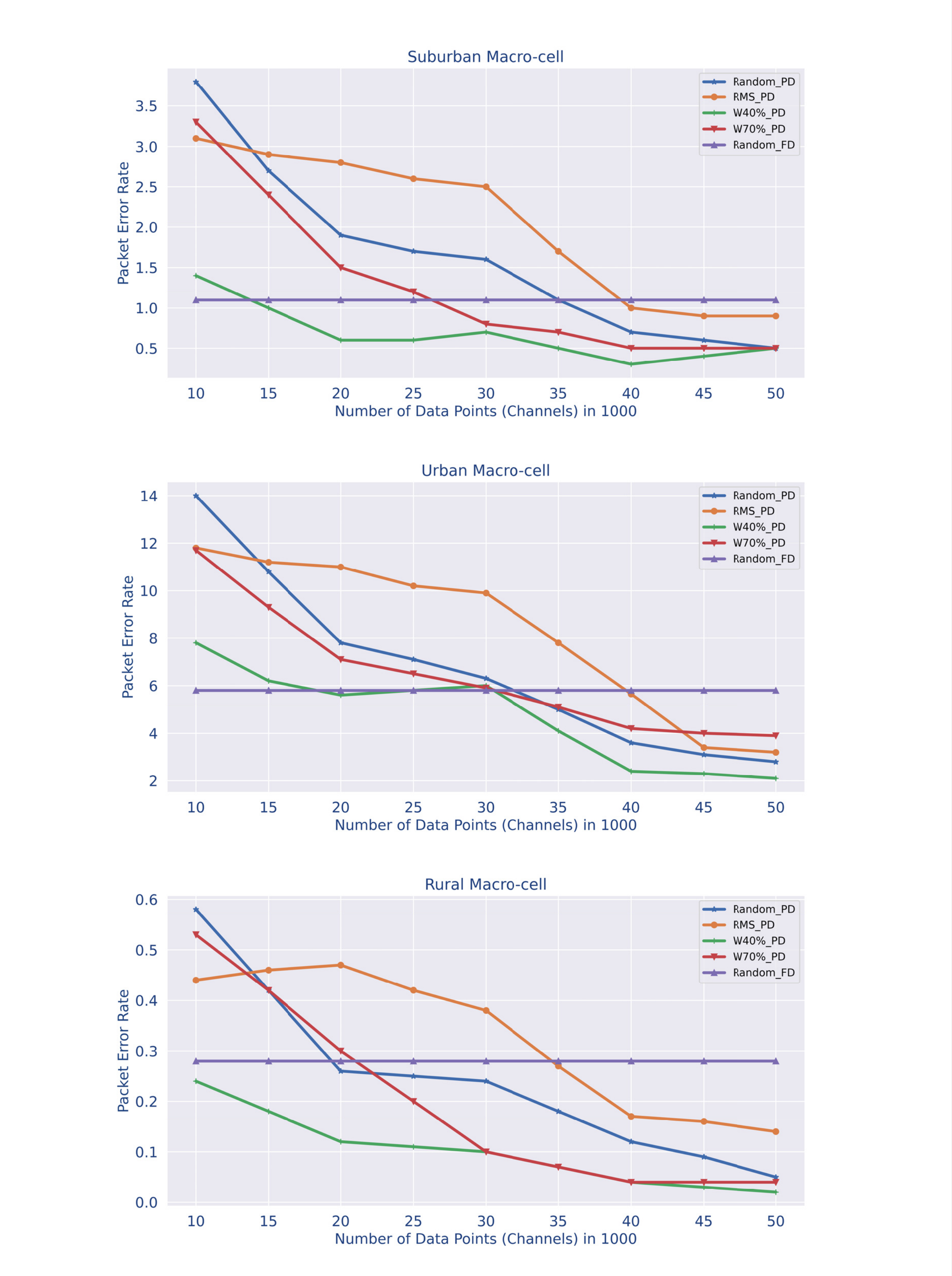


Figure 7.3 The system’s reliability measured in packet error rate (PER)

7.6 Opportunities and Challenges

Since its recent launch, data-centric AI has attracted great attention. A huge contribution room is available for researchers from both industry and academia for improving and shaping the future of data-centric AI in wireless communications. This work just opened the door for, and highlighted the power of, data-centric AI solutions. However, a long line of wireless communication problems still waiting to be explored in the same way, such as wireless channel selection, CSI feedback, beamforming, or transmit antenna selection.

Although the recent advances in mode-centric AI could be considered metrics-driven, there is a considerable lack of metrics for metrics to evaluate the goodness of data in data-centric AI. Furthermore, the nature of data is different for different problems. This opens the door for a long path of research on metric development for evaluating the goodness of data for each problem.

Since the data-driven approach mainly depends on data collection, labeling, and wrangling pipeline, a large contribution room is opened for domain experts, without a solid background in AI, to involve in building and improving the performance of AI models. Although some enterprises do not have an AI expert within their staff, they definitely have some data experts who have solid domain experience. Those domain experts can participate in building powerful AI solutions for their enterprises. The domain knowledge will be a strong point added to the power of the AI models. This will help in widening the spread of AI and making it available to more sectors and businesses.

One of the main challenges facing this new approach to AI development is the lack of tools designed for collecting, monitoring, and evaluating the goodness of data. Another barrier facing all data-driven solutions in wireless communications is the lack of a standardized publicly available dataset for each problem. This diverts the formulations of the problem between the different works and complicates the comparison between different models and formulations. To facilitate the comparison of different proposed works, we should agree on a certain problem formulation. For example, a link adaptation problem should be treated as a classification problem or as an MDP? Standardizing such formulations makes the efforts of different researchers aligned to the same objective and increases the comparability between different proposed techniques.

7.7 Conclusion

This article presents a novel data-centric framework for addressing the wireless link adaptation problem within the context of the IEEE 802.11ax protocol standard. We introduce a set of specific criteria for the collection and preparation of datasets that are tailored to the unique requirements of link adaptation. Through a comprehensive evaluation, we demonstrate the potential enhancements in model performance achieved by adopting the principles of data-centric artificial intelligence, with a particular emphasis on leveraging domain knowledge to gain valuable insights. Furthermore, we analyze the various opportunities and challenges that arise with the emergence of the data-centric approach in the field of wireless communications. Our findings underscore the promising prospects of this approach in developing intelligent communication systems capable of meeting the increasing demands for quality of service. The observed improvements showcased in this study serve as a compelling motivation for further exploration of the data-centric approach by other researchers, who can investigate its effectiveness in addressing diverse communication problems. Finally, we outline several noteworthy research directions that merit attention in future endeavors, building upon the contributions presented in this article.

CHAPTER 8

A LEARNING FRAMEWORK FOR BANDWIDTH-EFFICIENT DISTRIBUTED INFERENCE IN WIRELESS IOT

Mostafa Hussien, Kim Khoa Nguyen, and Mohamed Cheriet

École de technologie supérieure (ÉTS), Univeristy of Québec, Canada, H3C 1K3

Paper published in IEEE Sensors Journal, June 2023

Abstract: In the Internet of Things (IoT) regime, the sensors usually have limited bandwidth and power resources. Therefore, in a distributed setup, each sensor should compress and quantize the sensed observations before transmitting them to a fusion center (FC) where a global decision is inferred. Most of the existing compression techniques and entropy quantizers consider only the reconstruction fidelity as a metric, which means they decouple the compression from the sensing goal. In this work, we argue that data compression mechanisms and entropy quantizers should be co-designed with the sensing goal, specifically for machine-consumed data and machine-to-machine (M2M) communications. To this end, we propose a novel deep learning-based framework for compressing and quantizing the observations of correlated sensors. Instead of maximizing the reconstruction fidelity, our objective is to compress the sensor observations in a way that maximizes the accuracy of the inferred decision (i.e., sensing goal) at the FC. Unlike prior work, we do not impose any assumptions about the observations' distribution which emphasizes the wide applicability of our framework. We also propose a novel loss function that keeps the model focused on learning complementary features at each sensor. The results show the superior performance of our framework compared to other benchmark models.

keywords :

Distributed inference, statistical hypothesis testing, neural networks, discrete-representation autoencoders

8.1 Introduction

Many wireless Internet of things (IoT) applications employ a distributed inference mechanism e.g., radar systems, multi-view surveillance systems, or multi-sensory human activity recognition. In the later system, for example, a human wears multiple, spatially-distributed, sensors (e.g., gyroscope and accelerometer). A decision about a human activity (e.g., walking, running, etc.) is inferred from the received sensor signals. In such a scenario, if each sensor considered only its local observations for decision inference, the error probability would be much higher compared to the scenario in which a global decision is inferred from the aggregated sensor data Salehkalaibar & et al. (2018).

To tackle this problem, a distributed setup may be employed in which the sensed data (a.k.a, environment observations) are sent to a central node, called **fusion center (FC)**. The FC infers a global decision based on the aggregated data received from all sensors. However, the sensors usually have limited power and bandwidth resources. For example, each sensor may have a fixed data rate of, $\mathscr{R}$ bps. Therefore, it should compress and quantize its sensed observation to fit the assigned bit rate. The FC, then, performs a specific inference task (i.e., the sensing goal). However, the FC infers the decision from only partial information due to the compression and quantization steps. This may result in a reduced decision accuracy at the FC Salehkalaibar & et al. (2018). Optimally processing the observations at each sensor can minimize the degradation of the decision accuracy Abdi & Ristaniemi (2020). For conditionally-independent sensor observations, an optimal decision can be easily reached using Bayesian inference theory Zhu & et al. (2019). However, the conditional-independence assumption does not hold for many real-life problems. In prior work, Chamberland & Veeravalli (2007); Tay & et al. (2009), the authors assume that the statistical distribution of sensor observations is priorly known. In this case, the goal is to design an optimal decision rule that maximizes the likelihood of the correct decisions. Unfortunately, in many practical applications, this distribution is not priorly known which increases the problem's complexity. In this case, data-driven solutions provide practical and efficient alternatives.

Although different works in the literature propose compression and quantization techniques for sensor data, their goal was mainly obtaining a high-fidelity reconstruction at the FC. This seems relevant for human-consumed data such as images and videos. However, for machine-consumed data, adopting reconstruction fidelity as a metric is doubtful. Indeed, the accuracy of the inferred decisions is more crucial than having a good reconstruction.

In this work, we tackle the problem of compressing and quantizing correlated-sensor observations for distributed inference tasks. Our main objective is to maximize the accuracy of the inferred decision rather than minimizing the reconstruction loss. While most of the literature work assumes sensor independence for mathematical tractability, we address the more challenging scenario of correlated sensors. We argue that this correlation can be exploited to obtain higher compression ratios without a considerable reduction in decision accuracy. Typically, this can be achieved by transmitting the unique features of each sensor and avoiding transmitting redundant features which are likely to be transmitted by other nodes in the network. In other words, we can formulate our research question as: can we distributively screen redundancies in sensor observations to transmit only informative data without imposing any assumptions on the distribution of the observations?

To answer this question, we exploit the recent advances in statistical learning techniques, especially deep learning. We propose a novel deep-learning framework for compressing and quantizing the observations at each sensor. In addition, the framework is jointly trained with the decision rule at the FC in an end-to-end fashion to maximize the accuracy of the inferred decision. End-to-end learning refers to training a complex system by applying gradient-based learning to the system as a whole Glasmachers (2017). We also propose a new loss function that helps the sensors in learning decision-aware representations of observations. Furthermore, we propose a training algorithm to efficiently train the proposed framework. Extensive results show the robustness and superiority of our proposed framework compared with different benchmark models.

8.1.1 Related Work

Similar work has been proposed for specific problems. For example, a line of work has been proposed for the problem of human activity recognition Yang & et al. (2008); Guo & et al. (2012); Huynh (2008); He & et al. (2012). In this problem, the hypotheses are the different human actions, while the data comes from multiple sensors fixed on the actor's body (e.g., gyroscope, accelerometer, etc.). The authors in Yang & et al. (2008) aimed to achieve a high action-classification accuracy with the minimum bandwidth consumption. At each sensor, the decision is inferred from the local information. The FC then takes a global decision using a majority-voting mechanism. Although they obtained good results, this approach ignores any complementary information captured by other sensors. Another work has been proposed for the problem of earthquake detection from wireless IoT sensors network Faulkner & et al. (2011). They presented a distributed approach for rapid detection of earthquakes using cell phone accelerometers, consumer USB devices, and cloud computing-based sensor fusion. The work in Faulkner & et al. (2011) learns a threshold for each sensor involved in the network in a way that maximizes the performance of the anomaly detection algorithm employed at the FC. Experimental results showed that this approach successfully distinguished between seismic motion and acceleration due to normal daily activities.

The work in Raghavan & Baras (2019) studied the problem of binary hypothesis testing with two observers, where the collected observations are assumed to be statistically correlated. To infer a decision, one of three solutions can be adopted. The first is a centralized solution in which the observations collected by both observers are sent to the FC. A global decision is inferred at the FC from the received sensor observations. The main concern of this solution is the huge bandwidth incurred in fusing the raw observations to the FC. The second solution makes each observer rely on their own locally collected observation. Then, each node exchanges its, locally inferred, decision with the other sensor to reach a global decision. The main limitation of this solution is that each sensor depends only on its local information and ignores any complementary information captured by the other sensor. In the last solution, each observer formulates the problem as a sequential hypothesis-testing problem. The authors in

Bouchoucha & et al. (2015) proposed a framework for exploiting the correlation between observations to reduce the mean square error of the distributed estimation. Specifically, each node predicts its next observation and transmits the quantized prediction errors to the FC instead of the quantized observations.

In the context of task-aware compression, a similar problem has been addressed in Chinchali & et al. (2018); Hu & et al. (2020); Amer (2020). For example, the authors in Chinchali & et al. (2018) used a reinforcement learning (RL) agent at each sensor to compress the observations before fusing them to the FC. The reward function at each agent considers its commitment to the assigned bandwidth. Although they achieved a good performance, there is a probability that the agent does not meet the bandwidth constraints after deployment. While in Hu & et al. (2020), the authors proposed *Starafish*, an image compression framework that outperforms JPEG by up to 3X in terms of bandwidth consumption and up to 2.5X in power consumption. The authors in Hu & et al. (2020) used an AutoML technique to search for tiny ML models that can work on power AIoT accelerators.

We can summarize the limitations of the literature work, which we addressed in our work, as 1) the conditional-independence assumption of the sensor observations is not always held; 2) the conditional-independence assumption ignores the potential opportunity to benefit from complementary features captured by different sensors; 3) the compression algorithms are designed independently from the sensing goal; 4) the limited power of analytical-based techniques in dealing with a large number of possible decisions and correlated sensors.

Contribution

This paper presents a novel deep learning-based compression framework for correlated-sensors data compression and quantization. Discrete representation autoencoders are adopted at each sensor to generate the compressed quantized form of the observations. At the FC, a multi-layer perceptron (MLP) architecture is adopted to jointly learn the decision rule with the sensor encoders. The main contribution of this work comes in three folds:

1. Extending autoencoders to learn a compressed and quantized representation for correlated-sensor observations. This learned representation conveys the complementary features at each sensor observation which helps maximize the likelihood of the correct decision at the FC while satisfying a communication constraint. This representation is jointly learned with the decision rule at the FC in an end-to-end fashion to maximize the decision accuracy.

2. Proposing a novel loss function that encourages the model to learn the unique features of each sensor. The function learns the soft probabilities of a baseline model trained using the raw observations. Moreover, we present a training algorithm that efficiently works in a wide range of applications.

3. Eliminating the conditional-independence assumption between sensor observations which has been widely adopted for mathematical tractability. Moreover, we consider a multi-hypothesis problem, which is more complex and realistic than the simple binary hypothesis problem assumed in most of the literature work.

The rest of this paper is organized as follows: Section 8.2 formulates the problem. In Section 8.3, we describe the various elements of the proposed framework. The discussion and the experimental results are given in Section 8.4. Section 8.5 concludes our work.

8.2 Problem Statement

Notation: Through this text, we refer to random variables by italic capital letters (e.g. X). Small letters refer to one realization of a random variable (e.g., x). Superscripts denote the sensor number. For example, x^i denotes the observation at sensor i. The observations are referred to by X while Y refers to the random variable of the labels (i.e., the target decisions at the FC). The parameters of the encoder at the i^{th} sensor is referred to as ϕ_i. The parameters of the decision rule at the FC is referred to by ω. The $\log(\cdot)$ function uses a base of 2. Table 8.1 summarizes the used symbols and notations. Through the text, we use the terms *observations* and *data-points* interchangeably. The terms *decision* and *sensing goal* have the same meaning.

Table 8.1 The notation used through the text

Symbols	**Description**
$\mathbf{x}^i$	The current observation of the i^{th} sensor.
$\mathbf{z}^i$	The compressed and quantized representation for the current observation at the i^{th} sensor.
ϕ_i	The encoder parameters of the i^{th} sensor.
f_{ϕ_i}	The encoder function at the i^{th} sensor given by a neural network parameterized by parameters ϕ_i.
y_j	The label of the j^{th} data point.
$\hat{y}_j$	The predicted label of the j^{th} data point.
θ	The parameters of the decision function (i.e., decision rule) at the FC.
ω	The parameters of the decision function given the raw-observations.
$\mathscr{S}$	The total number of sensors.
$\mathscr{C}$	The number of possible classes (i.e., decisions) to be predicted at the FC.
d	The dimension of the raw observations.
n	The dimension of the compressed and quantized observations.
$\mathscr{R}$	The bandwidth (in bps) assigned to each sensor.
χ	The observation space, $\mathbb{R}^d$.
Z	The latent space, $\{0,1\}^n$.
g_θ	The decision rule at the FC given by a neural network parameterized by parameters θ.
$\mathbb{S}^n$	An n-dimensional vector where each element belongs to the set $\mathbb{S}$.
$CE(\cdot)$	Crossentropy loss function, given in Eq. 8.7.
$KL(\cdot)$	KL-Divergence loss given in Eq. 8.3.

Suppose Y is a discrete random variable, representing a hypothesis about an environment. The variable takes values: $y \in \{1,2,\ldots,\mathscr{C}\}$ where $\mathscr{C}$ is the number of possible hypotheses or classes. Our goal is to form an estimate, $\hat{Y}$, of the true hypothesis, based on a set of observations collected from a set of $\mathscr{S}$ sensors. Accordingly, for each $t = 1,\ldots,\mathscr{S}$, let $\mathbf{x}^t$ represents the observation at node t, where $\mathbf{x}^t \in \mathbb{R}^d$ in some space χ known as the observation space. The set of all observations corresponds to an $\mathscr{S}$-dimensional random vector $X = (\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^{\mathscr{S}}) \in \chi^{\mathscr{S}}$ drawn from the conditional distribution $P(X|Y)$.

Our objective is to reach an optimal estimate $\hat{Y}$ for the true labels Y at the FC. If the FC has access to the distribution of the observations, $P(X|Y)$, then an optimal decision rule can be easily formulated. For example, with a binary hypothesis, an optimal decision rule can be reached by means of a likelihood ratio test: $P(X|Y=1)/P(X|Y=-1)$. However, in real-world problems, the FC does not know the distribution of the observation a priori, and it has access to only summarized forms of the original observations, $\mathbf{z}^t$, for all values of t. More specifically, we assume that each sensor, t, is restricted to a given bandwidth of, $\mathscr{R}$, bps. Therefore, each sensor is allowed to transmit an *n-dimensional* message, $\mathbf{z}^t \in \{0,1\}^n$, taking values in some space Z, such that $n \leq \mathscr{R}$. The conversion from the observation space, χ, to Z-space is carried out by an encoder $q : \chi \rightarrow Z$. The encoder, q, maps an input observation, x, in χ-space, to a codeword, $\mathbf{z}$, in Z-space. This encoded observation, $\mathbf{z}$, will be sent to the FC. To compute the estimate $\hat{Y}$, the FC applies a certain decision rule, g_θ, on the aggregated received messages such that $\hat{Y} = g_\theta(\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^{\mathscr{S}})$. It is known from the rate-distortion theory that the rate, $\mathscr{R}$, and the distortion at the receiver (in terms of reconstruction loss) are inversely proportional Blau & Michaeli (2019). Therefore, a larger rate, $\mathscr{R}$, implies better reconstruction fidelity at the receiver end. However, in our problem, we are not concerned about reconstruction fidelity as our main objective. Rather, we are more interested in maximizing the accuracy of the inferred decisions.

Inherently, increasing the rate, $\mathscr{R}$, will increase the information included in a message, $\mathbf{z}^t$, which increases the FC accuracy. In other words, increasing the rate, $\mathscr{R}$, increases the mutual information, I, between the joint distributions $P(\hat{Y}|Z)$ and $P(\hat{Y}|X)$. However, for limited bandwidth systems, increasing the bandwidth is not a practical option and each sensor should respect the assigned bandwidth. In this case, for correlated sensor observations, the redundancy between the different sensor observations can be exploited to obtain more efficient compression with minimal loss in the decision accuracy at the FC. This can be reached by optimizing the function given in (8.1). Note that the same objective of Eq. (8.1) can be received from Eq. (8.2) by minimizing the KL-divergence between the two distributions, as described below.

$$
\begin{aligned}
\overset{min}{\phi, \theta} \quad & \frac{1}{\mathscr{N}} \sum_{j=1}^{\mathscr{N}} -\log\big(g_\theta(z_j) = y_j\big) \\
\text{s.t.} \quad & \phi = [\phi_1, \phi_2, \ldots, \phi_{\mathscr{S}}], \\
& z_j = (f_{\phi_1}(\mathbf{x}^1), f_{\phi_2}(\mathbf{x}^2), \ldots, f_{\phi_S}(\mathbf{x}^{\mathscr{S}})), \\
& f_{\phi_i} \in \{0,1\}^n \;\; \forall i \in \{1, 2, \ldots, \mathscr{S}\}, \\
& n \leq \mathscr{R}
\end{aligned}
\tag{8.1}
$$

where $\mathscr{N}$ is the total number of points in a test set, g_θ is the decision function at the FC parameterized by parameters θ, f_{ϕ_i} is the encoder function at the i^{th} sensor parameterized by ϕ_i, and $\mathscr{R}$ is the bandwidth assigned for each sensor. The function in (8.1) optimizes 1) the encoder parameters $\phi = [\phi^1, \phi^2, \ldots, \phi^S]$ and 2) the decision rule parameters θ, to minimize the negative log likelihood loss. $f_{\phi^i}(\mathbf{x}^i)$ denotes the compressed and quantized version of the observation at the ith sensor. For example, $f_{\phi^1}(\mathbf{x}^1)$ is the compressed and quantized version of the observation at the 1st sensor. Since the output of the quantizer is binary quantized, it belongs to $\{0,1\}^n$ where n is the dimensionality of the compressed representation, which should be less than or equal to the assigned bandwidth $\mathscr{R}$.

The same objective can be formulated in terms of the *Kullback–Leibler* divergence between the two conditional distributions of the decision given the raw observations and the compressed messages as given in (8.2) and (8.4).

$$
\begin{aligned}
\overset{\text{min}}{\omega, \theta, \phi_i} \quad & KL\big(P(\hat{Y}|X)\,||\,P(\hat{Y}|Z)\big) \\
\text{s.t.} \quad & \phi = [\phi_1, \phi_2, \ldots, \phi_{\mathscr{S}}], \\
& P(\hat{Y}|X) = f_\omega\left(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{\mathscr{S}}\right), \\
& P(\hat{Y}|Z) = g_\theta\left(f_{\phi_1}(\mathbf{x}^1), f_{\phi_2}(\mathbf{x}^2), \ldots, f_{\phi_{\mathscr{S}}}(\mathbf{x}^{\mathscr{S}})\right), \\
& f_{\phi_i} \in \{0,1\}^n \;\; \forall i \in \{1, 2, \ldots, \mathscr{S}\}, \\
& n \leq \mathscr{R}
\end{aligned}
\tag{8.2}
$$

where ω is the parameters of a benchmark model (i.e., a larger neural network model trained to classify the raw observations without compression). But the *KL-divergence* is given by:

$$KL(P||Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{8.3}$$

Substituting the *KL* term in (8.2) by (8.3), we get Eq. (8.4).

$$\begin{aligned} \overset{\min}{\omega,\theta,\phi_i} \quad & \sum_i P(\hat{Y}_i|X_i) \log\left(\frac{P(\hat{Y}_i|X_i)}{P(\hat{Y}_i|Z_i)}\right) \\ \text{s.t.} \quad & \phi = [\phi_1, \phi_2, \ldots, \phi_{\mathscr{S}}], \\ & P(\hat{Y}|X) = f_\omega\left(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{\mathscr{S}}\right), \\ & P(\hat{Y}|Z) = g_\theta\left(f_{\phi_1}(\mathbf{x}^1), f_{\phi_2}(\mathbf{x}^2), \ldots, f_{\phi_{\mathscr{S}}}(\mathbf{x}^{\mathscr{S}})\right), \\ & f_{\phi_i} \in \{0,1\}^n \ \forall i \in \{1, 2, \ldots, \mathscr{S}\}, \\ & n \leq \mathscr{R} \end{aligned} \tag{8.4}$$

The following two points should be considered. Firstly, the message space $\{0,1\}$ is significantly smaller than the observation space $\mathbb{R}$. Secondly, the required dimension for the message, n, is substantially smaller than that of the raw observation, d (i.e., $n \ll d$). Therefore, the problem can be thought of as finding, for each sensor, t, an optimal encoder/quantizer q: $q(\mathbf{x}^t) = \mathbf{z}^t$ that maximizes the mutual information between the two distributions $P(Y|X)$ and $P(\hat{Y}|Z)$ under a certain communication rate $\mathscr{R}$. Note that although Eq. (8.4), which is mainly proposed for correlated observations, can also be used for independent observations, this will not be really beneficial in terms of either the compression ratios or the accuracy of the inferred decision.

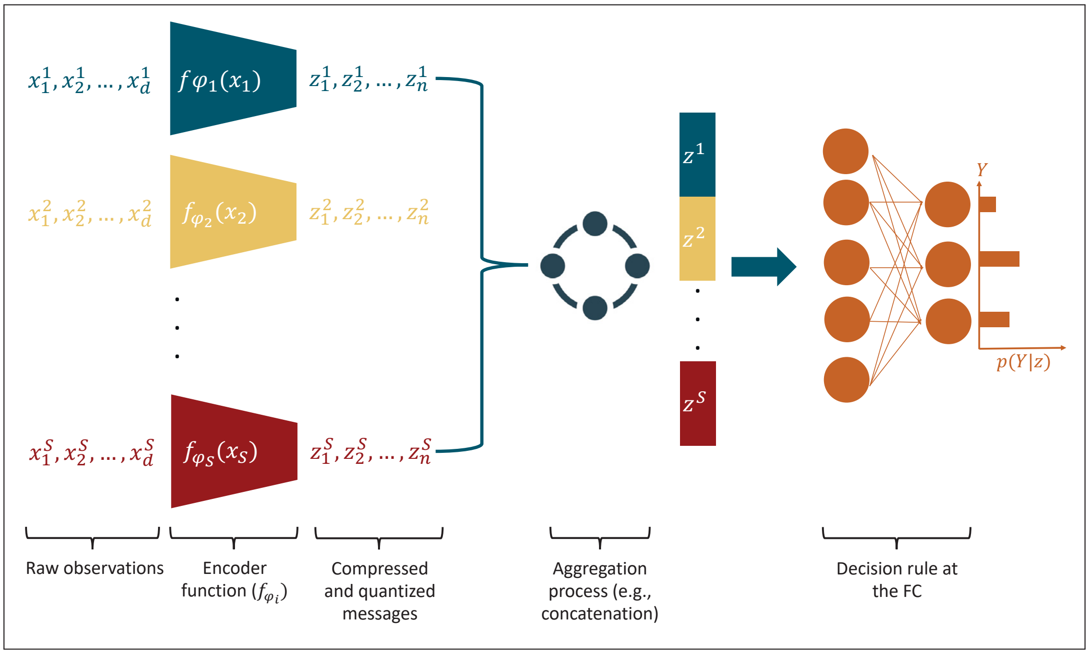

$x_1^1, x_2^1, \dots, x_d^1$
$f\varphi_1(x_1)$
$z_1^1, z_2^1, \dots, z_n^1$
$x_1^2, x_2^2, \dots, x_d^2$
$f_{\varphi_2}(x_2)$
$z_1^2, z_2^2, \dots, z_n^2$
$x_1^S, x_2^S, \dots, x_d^S$
$f_{\varphi_S}(x_S)$
$z_1^S, z_2^S, \dots, z_n^S$
z^1
z^2
z^S
Y
$p(Y|z)$
Raw observations
Encoder function (f_{φ_i})
Compressed and quantized messages
Aggregation process (e.g., concatenation)
Decision rule at the FC

Figure 8.1 Diagram demonstrating the system model. On the left, we see the sensor observations going through the discrete encoders to obtain the compressed quantized form of the observations. Then these messages are sent to the FC which passes the aggregated message to the neural network architecture to get a hypothesis estimation

8.3 Proposed Framework

8.3.1 Autoencoders

One of the powerful deep-learning architectures that achieved state-of-the-art results in different contexts is the autoencoder (AE). AE is a neural-network architecture consisting of two models namely, *encoder* and *decoder* models. The encoder maps an *I-dimensional* input to an *O-dimensional* codeword, where $O \ll I$. The decoder then reconstructs the input from this, compressed, codeword. This codeword is usually referred to as the **latent representation** and it belongs to a space called the **latent space**. This process is performed in an end-to-end fashion which implies that the encoder learns to compress the data in a way that helps the decoder in the reconstruction process. If the codeword is quantized (binary or multi-level), then the

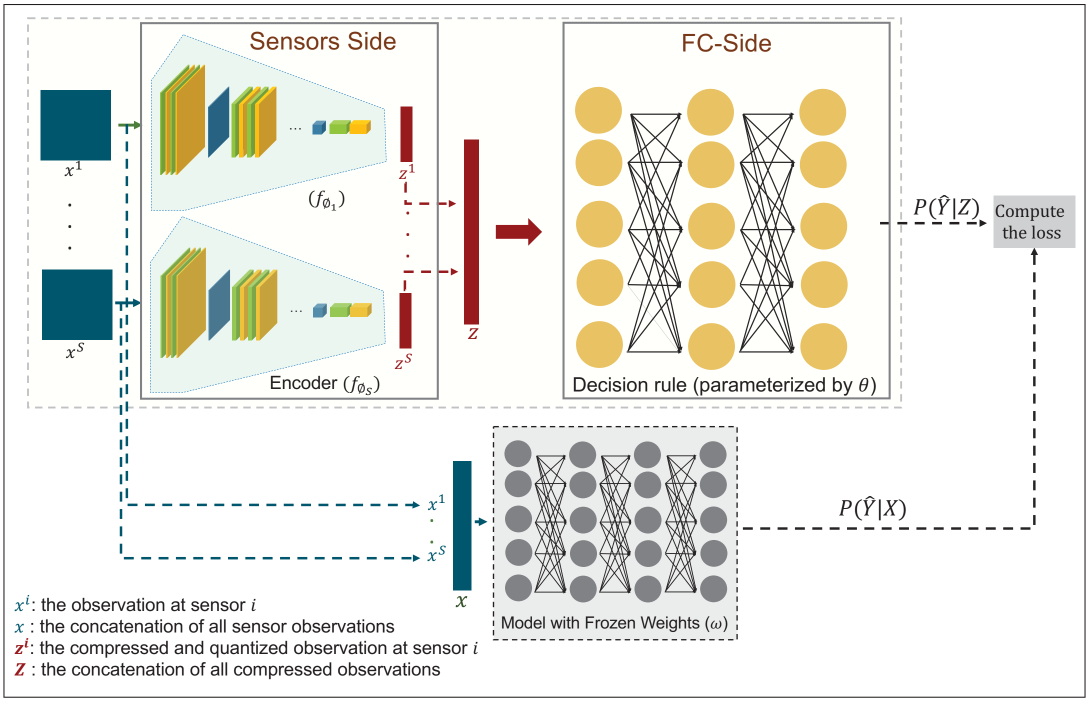

Sensors Side
FC-Side
x^1
x^S
(f_{ϕ_1})
Encoder (f_{ϕ_S})
z^1
z^S
Z
Decision rule (parameterized by θ)
$P(\hat{Y}|Z)$
Compute the loss
x^1
x^S
x
Model with Frozen Weights (ω)
$P(\hat{Y}|X)$
x^i: the observation at sensor i
x : the concatenation of all sensor observations
z^i: the compressed and quantized observation at sensor i
Z : the concatenation of all compressed observations

Figure 8.2 The proposed framework for deep distributed inference in wireless sensor networks

architecture is referred to as **discrete representation's autoencoder**. For further details on autoencoder architecture, we refer to Majumdar (2018).

According to the aforementioned problem formulation, our objective is to jointly learn an optimal encoder and quantizer at each sensor, q^t: $q^t(\mathbf{x}_i) = \mathbf{z}_i$, and an optimal decision rule at the FC $g_\theta(\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^S)$. To this end, we adopt a discrete-representation autoencoder at each sensor node to compress and quantize the sensor observations, see Fig. 8.1. It is worth differentiating between compression and quantization in this context. By compression, we mean the mapping from a higher-dimensional to a lower-dimensional space, $f : \mathbb{S}^d \to \mathbb{S}^n$, where $n \ll d$ and $\mathbb{S}$ is a certain set. On the other hand, quantization is mapping the values of individual dimensions from a set $\mathbb{S}_1$ to a set $\mathbb{S}_2$, where the cardinality of $\mathbb{S}_1$ is smaller than that of $\mathbb{S}_2$, (i.e., $|\mathbb{S}_1| < |\mathbb{S}_2|$).

Each sensor transmits the output of its *encoder model* to the FC. The output of the encoder model at sensor i is given by: $f_{\phi_i}(\cdot)$ where ϕ_i is the parameters of the i^{th} sensor. At the FC, an MLP neural network parameterized by parameters, θ, is used to approximate the optimal decision rule as depicted in Fig. 8.2. The decision rule at the FC is given by:

$$g_\theta([f_{\phi_1}(\mathbf{x}^1), f_{\phi_2}(\mathbf{x}^2), \ldots, f_{\phi_S}(\mathbf{x}^{\mathscr{S}})]). \tag{8.5}$$

where $\mathbf{x}^i$ is the current observation at the i^{th} sensor.

8.3.2 Implementation Details

The encoder architecture at each sensor is an MLP of three fully-connected layers with *ReLU* activations. In the output layer of the encoder, a *QSigmoid* activation is used Moons & et al. (2017). In the FC, we used six fully connected layers with *ReLU* activations in the hidden layers and *Softmax* activation in the output layer.

The model weights are initialized using *He* initializer He & et al. (2015). The models are trained using *Adam* optimizer Kingma & Ba (2014), with (0.01) learning rate and optimized to minimize our proposed loss function given in Eq. (8.8). Due to the adopted end-to-end training, the encoders will learn to encode the unique features at each sensor that help the FC to infer the correct decisions. Furthermore, the FC model optimizes its weights to maximize the likelihood of the correct decision given the encoded observations. Therefore, we can interpret the optimization of the classifier weights at the FC as learning an optimized threshold function for the decision rule.

8.3.3 Training Procedure

We propose a three-phase training algorithm for the proposed framework. In the first phase, we train an autoencoder at each sensor. The autoencoders are trained for input reconstruction from compressed codewords by minimizing the l2-loss given in Eq. (8.6).

Algorithm 8.1 The training procedure for the proposed framework, $\mathscr{S}$, sensors

```
Input: Dataset $D$, consisting of $\mathscr{N}$ observation/label tuples acquired from $\mathscr{S}$ sensors
Output: Model parameters, $\theta$, and $\phi_i$ for $i \in \{1,2,\ldots,\mathscr{S}\}$
At each sensor, $s_i$, train an AE to reconstruct its input using observations in $D$
Train an inference model, $I_1$, to approximate the conditional distribution $p(\hat{Y}|X)$
Freeze the weights of $I_1$, known as $\omega$
Train an inference model, $I_2$, (jointly with the encoders weights, $\phi_i$ for
  $i \in \{1,2,\ldots,\mathscr{S}\}$) to approximate the conditional distribution $p(\hat{Y}|Z)$
Return the learned parameters of $I_2$ (i.e., $\omega$), along with $\phi_i$ for $i \in \{1,2,\ldots,\mathscr{S}\}$
```

$$\underset{\phi,\theta}{min} \ \frac{1}{\mathscr{N}} \sum_{i=1}^{\mathscr{N}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2, \tag{8.6}$$

where $\mathbf{x}_i$ is the i^{th} observation, $\hat{\mathbf{x}}_i$ is the reconstruction, and $\mathscr{N}$ is the total number of observations in a dataset. In the second phase, we, independently, train an inference model, I_1, that takes as input the raw observations, X = $[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{\mathscr{S}}]$, and outputs the corresponding decision. Note that the inputs to this model are the raw observations without compression or quantization. The model is trained to optimize the classical Crossentropy function (8.7).

$$\underset{\theta}{min}[-\sum_{i=1}^{\mathscr{C}} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)], \tag{8.7}$$

where y_i and $\hat{y}_i$ are the true and predicted one-hot encoded label vectors of the i^{th} data point, and $\mathscr{C}$ is the number of classes. The output of the model I_1 approximates the conditional distribution $P(\hat{Y}|X)$. The set of parameters in the model I_1, denoted by vector ω, is then frozen and it will be used only for computing the value of the inference model loss function at the FC. This model represents the benchmark model that we aim to mimic after the compression and the quantization take place. We elaborate more on this part in subsection 8.3.4.

In the third and last phase, we use the encoder model (of the AE trained in the first phase) at each sensor to compress the captured observations. The output of the encoder at sensor i is denoted by $\mathbf{z}_i$. The parameters of an encoder model of an AE at the sensor, i, are denoted by ϕ_i. Accordingly, $\mathbf{z}^i = f_{\phi_i}(\mathbf{x}^i)$. The outputs of all the encoders are concatenated and fed to an inference model, I_2, with parameters θ to predict the output. In this case, the output of I_2 approximates the conditional distribution $P(\hat{Y}|Z)$. It is worth noting that the weights of I_2, θ, are trained jointly with the encoder's weights, ϕ_i, at each sensor. This means that the training in the last phase is done in an end-to-end fashion between the encoder weights and the parameters of the decision rule at the FC. Algorithm 8.1 summarizes the training procedure.

8.3.4 Proposed Loss Function

In the first and second phases of the training, we optimize the MSE and Crossentropy loss functions, respectively. As we move to the third phase of the training, where the sensor encoders and the FC inference model are jointly trained, it is observed that minimizing traditional Crossentropy is insufficient in tackling the problem at hand. Recall from the previous section that the objective of the proposed framework is to make the encoders benefit from the redundancies (between the sensor observations) to obtain high compression ratios without harming the decision accuracy. This implies that encoders should learn to encode the complementary features of their observation. To this end, we propose a novel loss function given by Eq. (8.8).

$$\mathscr{L}(Y,\hat{Y}) = CE(Y,\hat{Y}) + KL(P(\hat{Y}|X)||P(\hat{Y}|Z)) \tag{8.8}$$

The proposed function helps the model to learn a joint conditional distribution for the decision given the compressed observations, $P(\hat{Y}|Z)$, which is as similar as possible to the joint conditional distribution for the decision given the uncompressed observations, $P(\hat{Y}|X)$. This term reduces the loss in decision accuracy due to the compression of the sensor observations. Given a limited budget of bit rate to encode the observations, we argue the proposed function encourages the encoders to encode only the relevant features that help in maximizing the likelihood of the correct decision at the FC.

During the end-to-end optimization of the proposed loss function, the encoders will tend to eliminate the mutual information between the correlated sensors' observations and encode only the relevant features to satisfy the bandwidth constraint. Although this setting is derived from correlated data, it works also for independent sensors. However, the inter-observation redundancy is much less and the compression can hurt the inferred decision accuracy.

Note that we handcrafted a model for each dataset to achieve the highest possible accuracy. The models have been selected according to the proposed loss function, Eq. (8.8), such that it emphasizes learning complementary features at each sensor. To this end, the second term in Eq. (8.8) adds a regularization term based on the KL-divergence between the conditional probability distribution of the decision given the full observation $P(\hat{Y}|X)$, and the distribution of the decision given the compressed and quantized version of the observations $P(\hat{Y}|Z)$. Moreover, our model jointly learns a quantizer function (entropy encoder) along with the source encoder. Jointly learning the encoders with the decision rule encourages the model to learn only the complementary features at each sensor. The proposed models work well with each problem without overwhelming the framework with complex architectures such as *AlexNet*, *ResNet*, *GoogleNet*, etc LeCun *et al.* (2015). The power of these complex models is required mainly for high-dimensional observation space, such as surveillance cameras. In this case, the hidden (deep) convolutional layers can extract spatial features in the observations in an efficient way. However, in lower-dimensional observation space, as in our case, handcrafted models are good enough. This conclusion is compatible with the findings reported in Suto & Oniga (2019).

8.3.5 Dataset Preparation

The proposed framework is general and widely applicable in different problems. For the framework to be employed in a certain distributed inference task, a dataset should be prepared for training purposes. A typical dataset consists of $\mathscr{N}$ data points along with the associated labels $\{\mathbf{x}_i, y_i\}_{i=1}^{\mathscr{N}}$. Each data point, $\mathbf{x}_i$, represents the concatenation of simultaneous readings from $\mathscr{S}$ sensors such that $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^{\mathscr{S}}]$. The label $y_i \in \{1, 2, \ldots, \mathscr{C}\}$ is the target hypothesis (i.e., class) associated with these sensor readings. It is worth noting that these readings are assumed to be perfectly synchronized and each data point represents the readings at the same time step.

8.4 Results and Discussion

Various datasets have been used to evaluate the performance of the proposed framework. Each dataset represents a different environment setting and generating distribution.

8.4.1 Distributed Inference Accuracy

Comparative Evaluation

Table 8.2 The classification accuracy of the proposed framework under different compression ratios compared with benchmark models on the WARD dataset

Method	**CR=2**	**CR=4**	**CR=8**
Cheng et al. (ASRCM) Cheng & et al. (2017)	94%	88%	83%
Cheng et al. (NN) Cheng & et al. (2017)	82%	78%	75%
Zhang et al. Zhang & Sawchuk (2013)	87%	83%	80%
Our Framework	**99.7%**	**97.4%**	**95.6%**

Table 8.3 The classification accuracy of the proposed framework at compression ratio (CR=2) compared with benchmark models on the WARD dataset

Method	**Detection Accuracy**
Zhu et al. Zhu & et al. (2019)	99.00%
Yang et al. Yang & et al. (2009)	93.60%
Huynh Huynh (2008)	96.97%
He et al. + PCA He & et al. (2012)	76.31%
He et al. + LDA He & et al. (2012)	40.30%
He et al. + GDA He & et al. (2012)	99.20%
Guo (Majority voting) Guo & et al. (2012)	94.96%
Guo (Maximum) Guo & et al. (2012)	96.20%
Guo (WLOP) Guo & et al. (2012)	98.02%
Guo (WLOGP) Guo & et al. (2012)	98.78%
Sheng et al. Sheng & et al. (2016)	95.90%
Oniga and Jozef Oniga & Jozsef (2015)	98.10 %
Our Framework	**99.7**%

To evaluate the effectiveness of the proposed framework, we used a publicly available dataset called *Wearable Action Recognition Database (WARD)* presented in Yang & et al. (2009). The obtained performance is compared against three other baseline models applied to the same dataset. The dataset is designed for human activity recognition from sensors' data. This dataset is collected from five sensor boards attached to different points in the human body. Each sensor board has a tri-axial accelerometer and a bio-axial gyroscope with three and two-dimensional outputs respectively. Each human operator performs 13 different actions which represent the labels (classes) to be predicted by the classifier at the FC.

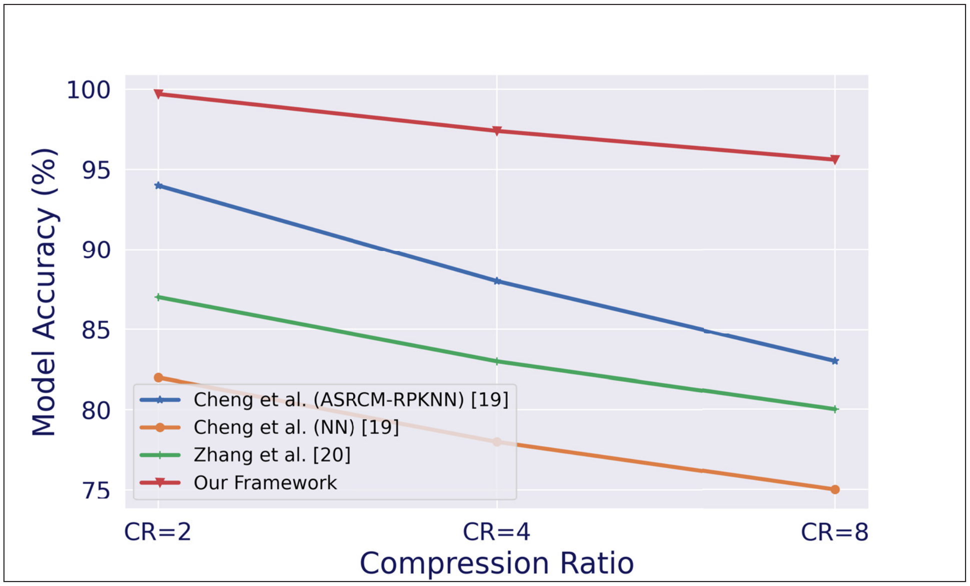


Figure 8.3 Comparing model accuracy under different compression ratios

Table 8.2 and Fig. 8.3 show a comparison between the performance of the proposed framework and the baseline models under different compression ratios. We can see from table 8.2 that the performance of our framework outperforms other models under all compression ratios. We can see that our framework preserves high accuracy even under high compression ratios. For example, increasing the compression ratio from 2 to 8 decreased the accuracy by 4.1% only (i.e., from 99.7% to 95.6%). This is a small margin compared with 11% loss in Cheng et al. (ASRCM) Cheng & et al. (2017), and 7% in Cheng et al. (NN) Cheng & et al. (2017) and Zhang et al. Zhang & Sawchuk (2013).

Table 8.3 shows the classification accuracy of the proposed framework compared with the accuracy of other works in the literature. The table reports results for Zhu et al. Zhu & et al. (2019), Yang et al. Yang & et al. (2009), Huynh Huynh (2008), He et al. He & et al. (2012), Guo et al. Guo & et al. (2012), Oniga et al. Oniga & Jozsef (2015), and Sheng et al. Sheng & et al. (2016). It is clear from the table that the proposed framework achieves state-of-the-art accuracy compared with the aforementioned works. In addition, the proposed framework involves the minimal required bit rate, $\mathscr{R}$, from the sensors to the FC, which highly contributes to power saving and prolongs the sensors' lifetime. These results can be attributed to the fact that we learn complementary features between correlated sensors that highly contribute to improving the decision accuracy rather than learning local features for each sensor. This learning behavior is motivated by the proposed loss function, Eq. (8.8). Moreover, our framework jointly learns a quantizer function $q : \chi \to Z$ with the encoder function which minimizes the end-to-end error and improves the accuracy of the sensing task. Note that the work in Zhu & et al. (2019) explores the correlation between the sensor observations to disable the transmission on the sensors that did not capture new relevant features and thus save the consumed bandwidth. Comparatively, in our work, we exploit this correlation to transmit only the relevant complementary features. Consequently, we contribute in two directions, namely, saving the consumed bandwidth and, at the same time, improving the decision accuracy.

Artificial Problem

We tested the proposed framework with four datasets, which are: 1) MNIST LeCun (1998), 2) Fashion-MNIST Xiao & et al. (2017), 3) Street View Houses (SVH) Netzer & et al. (2011), 4) CIFAR-10 Krizhevsky & et al. (2009). For each dataset, we used different Compression Ratios, *CR*. *CR* is defined as the ratio between the uncompressed dimension and compressed dimension Sayood (2017). It is worth noting that the compression ratios of the literature work consider only compression by dimensionality reduction (i.e., any input or output dimension $\in \mathbb{R}$). Based on that, the input and output space remains the same. Unlike prior methods, we go beyond to counts for the quantization (since an input dimension is $\in \mathbb{R}$ while an output dimension is quantized $\in \{0,1\}$). In these experiments, we simulate two sensors (s_1,s_2) sending

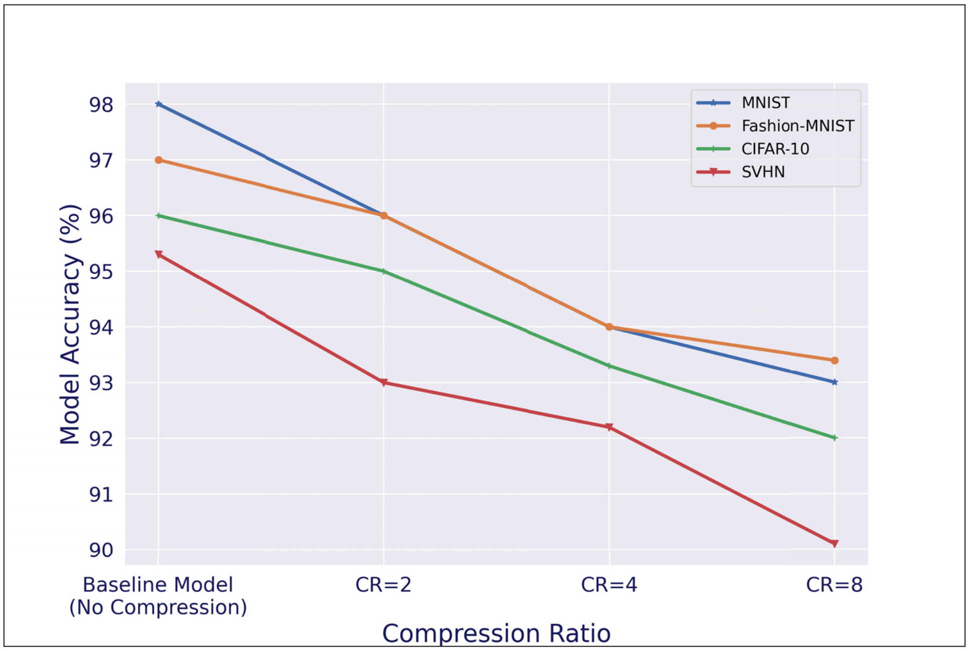


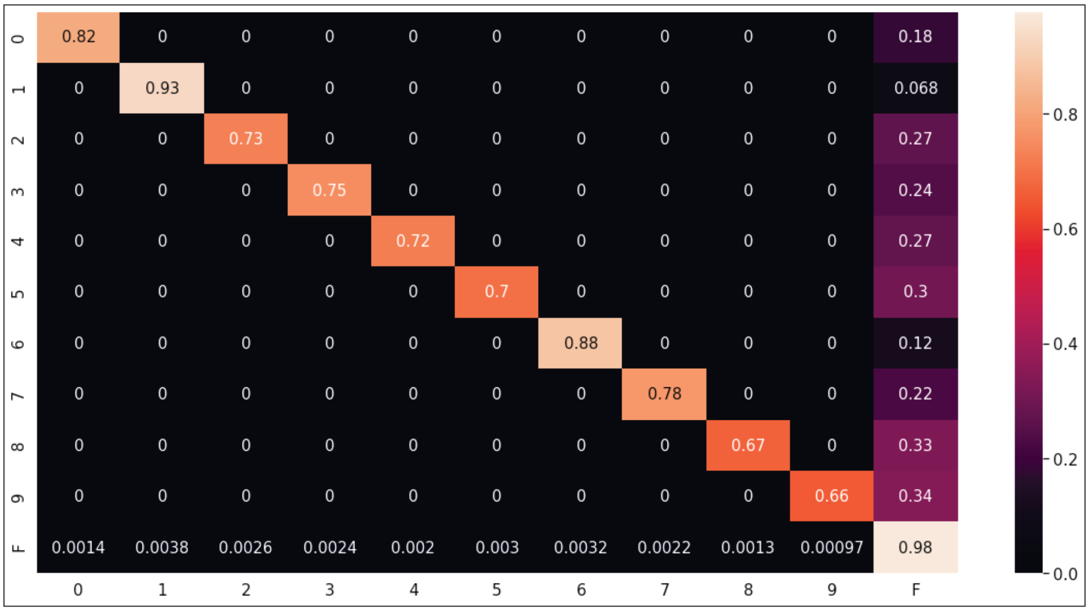
Figure 8.4 The decision accuracy of the proposed framework with four datasets under three compression ratios. The baseline model represents the case in which we fuse the raw observations to the FC without compression

their data to a FC. Assume the observations at sensor s_1 belong to a class C_i and at sensor s_2 belong to a class C_j. The decision rule at the FC can be defined as:

$$\psi(z^1,z^2) = \begin{cases} i & ifi = j \\ -1 & ifi \neq j \end{cases} \tag{8.9}$$

In other words, the decision will be the class label if the two observations belong to the same label, and -1 otherwise. Since each dataset consists of images belonging to one out of 10 total classes, we expect the classifier to have 11 classes. In order to make a fair comparison, we used the same classifier capacity (e.g., number of layers, number of nodes in each layer, activation functions, etc.) for each dataset. We compared the obtained results with the baseline model accuracy. The baseline model is defined as a neural network that takes the raw observations as input. In this case, the FC has complete knowledge of sensed data, which represents the optimal case in terms of data availability.

Fig. 8.4 shows the obtained results in each case. We can notice that the framework performance approaches the baseline with the lowest compression ratio, $CR = 2$. A small loss in accuracy is reported with higher CRs (i.e., $CR = 4$ and 8). However, the obtained accuracy is still high even with the highest CR. For example, we obtained 95.3% of the baseline with $CR = 8$ in the MNIST dataset. This means compressing the observations to only 12.5% of their original dimension with quantization, results in a 4.7% of accuracy reduction.



	0	1	2	3	4	5	6	7	8	9	F
0	0.82	0	0	0	0	0	0	0	0	0	0.18
1	0	0.93	0	0	0	0	0	0	0	0	0.068
2	0	0	0.73	0	0	0	0	0	0	0	0.27
3	0	0	0	0.75	0	0	0	0	0	0	0.24
4	0	0	0	0	0.72	0	0	0	0	0	0.27
5	0	0	0	0	0	0.7	0	0	0	0	0.3
6	0	0	0	0	0	0	0.88	0	0	0	0.12
7	0	0	0	0	0	0	0	0.78	0	0	0.22
8	0	0	0	0	0	0	0	0	0.67	0	0.33
9	0	0	0	0	0	0	0	0	0	0.66	0.34
F	0.0014	0.0038	0.0026	0.0024	0.002	0.003	0.0032	0.0022	0.0013	0.00097	0.98

Figure 8.5 A heatmap representation for the confusion matrix of an MNIST classifier with 98-dimension latent code corresponding to a compression ratio of 8. The class label F here represents class label -1

In the reconstruction of the training dataset, we randomly shuffle the datasets at each sensor. Consequently, most of the observation combinations fall in the class of -1 (i.e., the two observations are not in the same class). This produces an imbalanced class distribution. Due to this imbalance, we report the confusion matrix of the framework classifier for the MNIST dataset and 98-dimension latent code in Fig. 8.5. We can see from Fig. 8.5 that the proposed framework is capable of inferring the right decision with high accuracy even with imbalanced data.

The key idea of compressing correlated sensors' data is extracting complementary information from correlated observations and ignoring any redundancies. Our proposed loss function (Eq. 8.8) achieves this goal by incorporating a KL divergence term to the loss function between the soft labels generated by a baseline model (e.g., a large model trained on raw observations to predict $P(Y|X)$) and the decision function at the FC, $P(Y|Z)$. To minimize this term, we encode only the complementary features that help the FC to mimic the behavior of the baseline model. As described in Algorithm. 8.1, we jointly train the encoder models at each sensor with the decision function at the FC in an end-to-end fashion. This end-to-end training makes the encoders jointly learn these features with the decision function as they receive penalization based on the distance between the predicted distribution and that of the baseline model.

We also consider a convex combination between these two terms in Eq. (8.8) as follows:

$$\mathscr{L}(Y,\hat{Y}) = \phi\, CE(Y,\hat{Y}) + (1-\phi)(KL(P(\hat{Y}|X)||P(\hat{Y}|Z))) \tag{8.10}$$

We experimented different values for ϕ in the range $[0,1]$ with 0.1 step size. We note that the best results are obtained at different values for ϕ for different datasets. For example, using ϕ equals 0.4 gave the best results with MNIST while 0.5 and 0.7 gave the best results with Fashion-MNIST and CIFAR10. This can be attributed to the different distribution of the observation and the class weights in each dataset. Therefore, we recommend trying different values of ϕ and using the value with the best results.

Indoor Localization

We test our proposed framework on a dataset for indoor localization using WiFi fingerprint. The dataset consists of 7175 fingerprints collected from 489 different locations (almost 15 fingerprints per location). The training dataset was compiled by taking samples at every 3 meters on average with 15 samples per location. The time at each location was approximately 40 seconds performing consecutive scans with a bq Aquaris E5 4G device using Android stock 6.0.1 without making any movements during the process. For a complete description of the dataset and the dataset collection protocol, we refer the reader to González & et al. (2019).

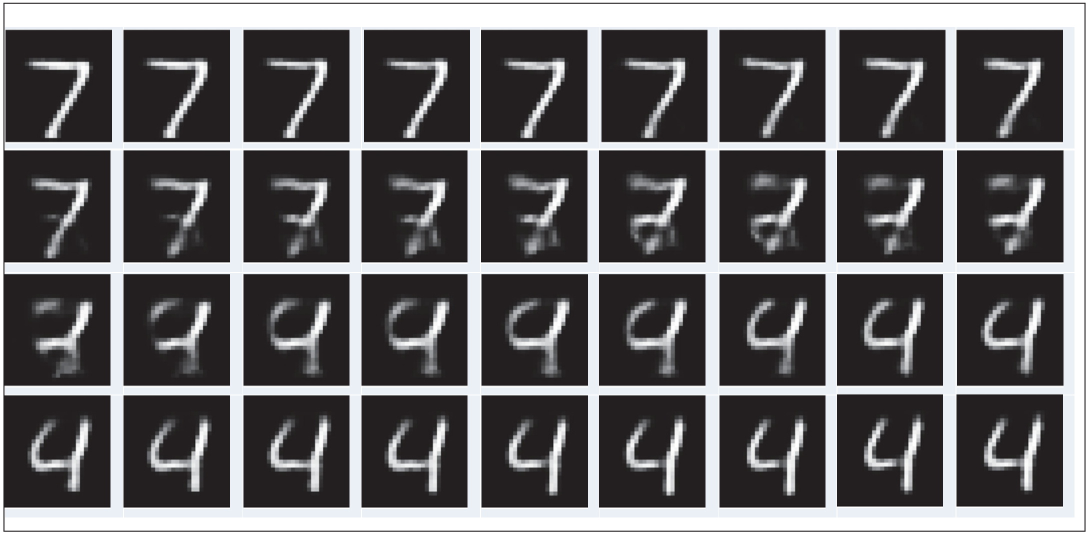


Figure 8.6 Interpolation between two points in the latent space. We choose a start and end point, then we gradually flip a bit each time along the different bits between the two vectors. The starting point is shown in the top-left corner, and the endpoint is in the bottom right

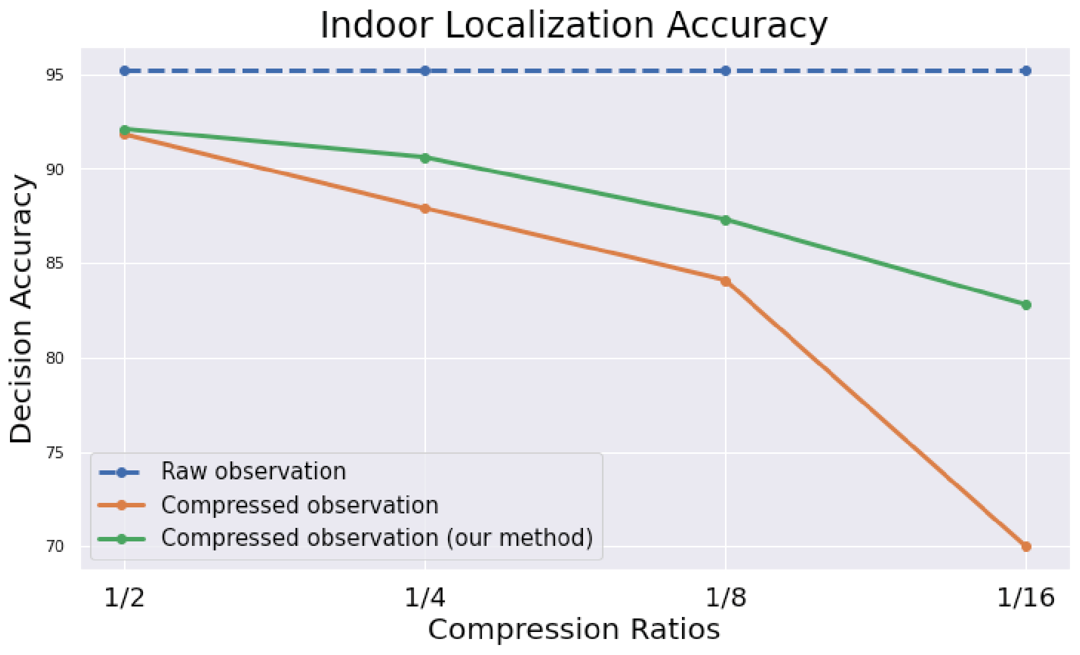


Figure 8.7 The accuracy of indoor localization problem using the proposed framework

Fig. 8.7 shows the degradation in the accuracy due to the increase in the compression ratio. Note that the objective of this experiment is not to achieve state-of-the-art results in the classification. Rather, we aim to highlight how much accuracy we can lose due to the observation compression. It is clear from Fig. 8.7 that the decrease in the accuracy due to observation compressing using our proposed framework is much less compared with compression using traditional autoencoders.

8.4.2 Semantics of the Latent Representation

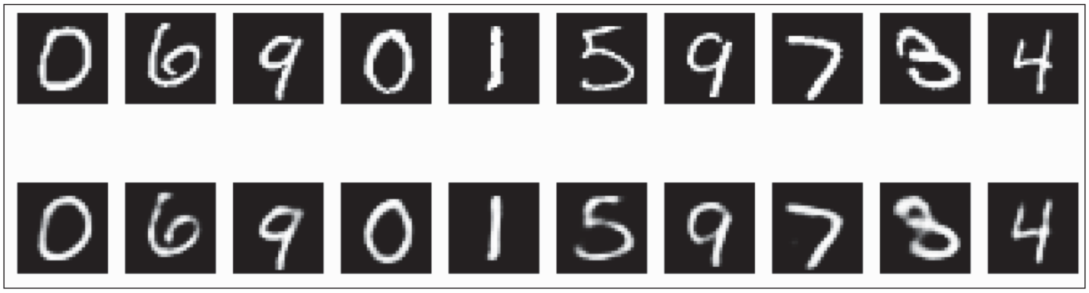


Figure 8.8 Reconstruction of MNIST data images using a 98-dimensional discrete latent code. The top row shows the original input, while the reconstruction is shown in the bottom row

In AE-based architectures for dimensionality reduction, a special interest is paid to the robustness of the learned codewords in the latent space Choi & et al. (2019). To evaluate the robustness of such codewords, we interpolate between different points in the latent space and qualitatively observe the gradual changes in the reconstructed data. This experiment verifies that the model: (a) has injected enough redundancies into the codewords and consequently the model is capable of reconstructing the input even with codeword errors, (b) has learned relevant features for the underlying structure of the data.

We randomly select two test points to represent the start and end points. In each step, we flip a bit in the latent codeword, fed the new codeword to the decoder model, and observe the gradual changes in the reconstruction. Algorithm 8.2, describes this experiment in more detail. Fig. 8.6 shows the gradual transition in the digit shape with the gradual bit flipping. We can

Algorithm 8.2 The procedure for evaluating the semantics of the latent codewords

```
Randomly select two random points $\mathbf{x}_1$, $\mathbf{x}_2$
Encode each data point using the encoder function, $f_\phi$:
$\mathbf{z}_1, \mathbf{z}_2 = f_\phi(\mathbf{x}_1), f_\phi(\mathbf{x}_2)$;
$\mathbf{h} = \mathbf{z}_1 \oplus \mathbf{z}_2$;
i=0;
while $i < len(\mathbf{h})$ do
    if h[i] equals 1 then
        Flip the bit at $\mathbf{z}_1$[i];
        i = i + 1;
        $\bar{\mathbf{x}}_{1i} = f_\theta(\mathbf{z}_1)$; where $f_\theta$ is the decoder function.
        Plot $\bar{\mathbf{x}}_{1i}$;
    End-if
End-while
```

observe that decrementing the hamming distance between the start and end points, by flipping bits, slowly alters the digit characteristic until it reaches the endpoint.

8.4.3 Rate/Computation Tradeoff

While compressing the observations reduces bandwidth consumption for transmission, it comes at a cost in accuracy and computation. The required computation resources (measured by the floating-point operations (FLOPS)) increase according to the model complexity (measured by the number of weights). Moreover, increasing the model complexity leads to improved compression, and consequently improved decision accuracy at the FC. Therefore, a design decision should compromise between the model complexity on the one hand, and the consumed computations and FC accuracy on the other hand. However, the training phase can be done offline (before the deployment of the sensors), and only the inference will take place during the operation which requires only one forward pass (a very small number of FLOPS) to predict the encoded messages. Fig. 9 shows this trade-off trend between the computation requirement (measured by FLOPS) and the model accuracy. In this figure, we can see that increasing the model accuracy requires adopting smaller compression ratios which imply higher data transmission. On the other hand, a smaller compression ratio requires transmitting more data and

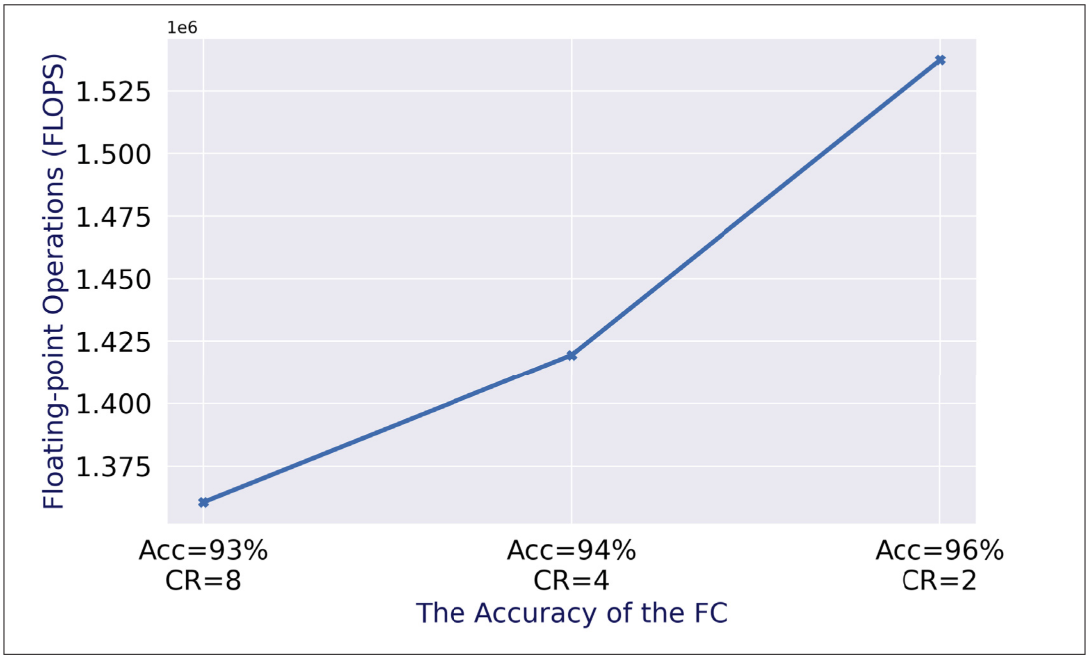


Figure 8.9 Tradeoff between model accuracy (achieved at different compression ratios) and the consequent increase of computation requirement

requires more computational resources at each sensor. The optimization of compression ratios is out of the scope of this paper and will be explored in our future work.

8.4.4 On Quantizer Design

Different techniques for quantizer design have been proposed on the literature. Some work proposed analytical techniques for the quantizer design. These techniques give a precise description for the optimality of the quantizer and, in some cases, they specify the quantizer performance (in the form of error bound or other criteria). However, to derive this mathematical analysis, these techniques assume certain statistical properties in the sensor data. There are many cases in which we do not have prior knowledge of this information. For example, the quantizer proposed on Vempaty *et al.* (2014) studied Gaussian observations. On the other hand, our proposed quantizer design does not impose any restrictions or assumptions on the distribution of the raw observations. Other learning-based quantizer designs either do not address the case of correlated observations or do not consider the accuracy of the inferred decision on the loop. The authors in Choi & et al. (2019) proposed a learning-based quantizer in the channel coding regime. However, they did not consider the accuracy of the decisions inferred from this compressed data. In this work, we address this gap.

8.4.5 Applicability

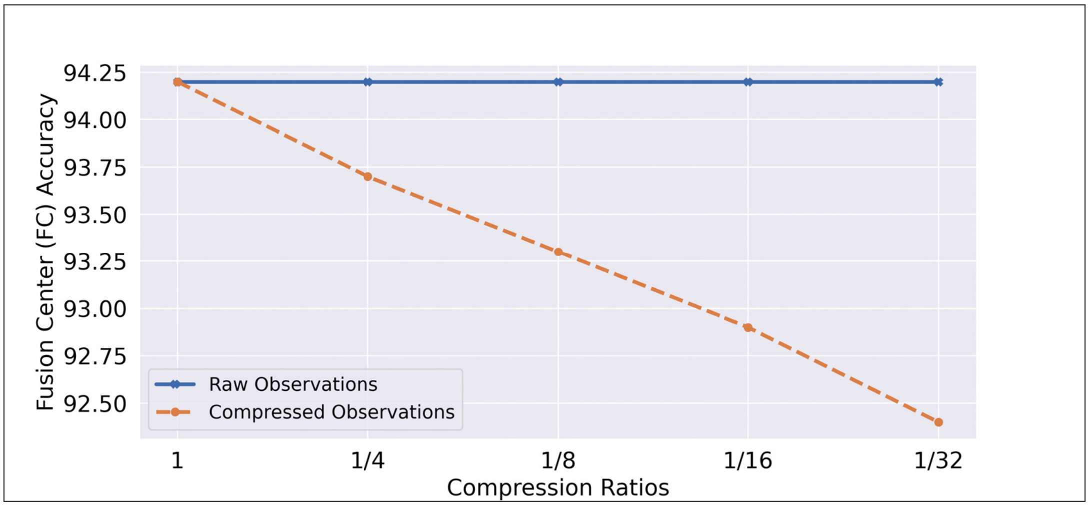


Figure 8.10 The decision accuracy for a wireless link adaptation problem under different compression ratios. The results confirm the general applicability of the proposed system to problems from different domains

Our framework along with the proposed loss function, Eq. (8.8), and the training procedure given in Algorithm. 1 can work with any type of parallel distributed detection network. This type of setting has various applications in wireless IoT. Although minor customizations are required to fit each specific problem, the framework is still widely applicable to different problems from various domains. In this paper, we reported the experimental results on various types of sensors and applications (e.g., image classification, human activity recognition, etc.). Specifically, we experimented with 5 different datasets (MNSIT, Fashion MNIST, SVHN, CIFAR-10, WARD) representing three different types of sensors (cameras, gyroscope, and accelerometer). To further evaluate the generality of our framework, we evaluated a completely different domain (i.e., wireless link adaptation) using three datasets combined in a global one Hussien & et al. (2021). In this scenario, the sensors are the antennas at each mobile node, the observations are the channel state information (CSI) captured at each mobile, and the environment is the wireless channel Hussien *et al.* (2020). The sensors send their observations to an FC to infer a global decision. The base station (BS) acts as an FC in this case, and the decision is the selected modulation and coding scheme (MCS). The results shown in Fig. 8.10 show a minor loss in the adaptation decision at the FC with the increase in the adopted CR. For example, when compressing the original raw observations (i.e., CSI in this case), the accuracy only drops from 94.25% to 93.7%. This means only 0.55% loss in accuracy is achieved while saving 75% of the original bandwidth. The obtained results confirm the general applicability of our proposed method in different domains and problems.

8.4.6 Results of Input Reconstruction

A task of input reconstruction was performed to evaluate the robustness of the learned features. In this experiment, MNIST and Fashion-MNIST datasets are used in the evaluation. We used a $CR = 8$, which corresponds to a latent code of 98-bit. Fig. 8.8 shows the result of the input reconstruction.

8.5 Conclusion

In this paper, we proposed a deep-learning framework for compressing correlated sensor observations in distributed inference problems. The proposed framework employs discrete representation autoencoders to encode the observations at each sensor. A novel loss function has been proposed to improve the accuracy of the framework. A multi-layer perceptron architecture has been used at the FC to jointly learn the decision rule. The proposed framework addresses the hard-to-tackle problem of correlated sensor observations and does not assume any prior knowledge about the distribution of the observations. The framework has been extensively tested using different datasets and has demonstrated significant performance improvements.

CONCLUSION AND RECOMMENDATIONS

9.1 Conclusions

In this dissertation, we presented a learning framework for optimized control of wireless links. The work started by building an efficient technique for carrier frequency offset (CFO) estimation. The predicted CFO is used for signal correction to compensate for the predicted CFO. The signal is then used for channel estimation to calculate the channel state information (CSI). The user equipment (UE) should send the estimated CSI to the gNB, which is a time and bandwidth-expensive process. At this time, the second part of the proposed framework then comes to the scene to optimize the time and bandwidth consumed in this feedback process. Part 2 of the dissertation discusses the problem of CSI compression in MIMO-FDD systems along with our proposed solutions. We proposed two main solutions: 1) the first solution is an autoencoder-based architecture based on the variational autoencoders (VAE), and 2) the second solution is a neural compression technique based on leveraging the bias/variance tradeoff for CSI compression. The gNB uses the CSI feedback in several functionalities such as link adaptation, beamforming, mobility management, resource allocation, etc. The proposed framework builds on the received CSI feedback from part 2 to develop a reliable link adaptation technique. This represents the third module in our proposed learning framework. This module exploits the CSI feedback to predict the best modulation and coding scheme (MCS), guard interval (GI), and other PHY transmission parameters that best suits the current channel conditions. As a part of this module in the framework, we developed a custom loss function that focuses more on minimizing the false positives in the classes corresponding to retransmissions. This approach has dual benefits for reliability and power aspects. Furthermore, we investigated the potential of data-centric AI in the problem of link adaptation. We studied and analyzed several selection criteria on the performance of a link adaptation model and concluded that good dataset selection can lead to improvements in the model's accuracy.

From the earlier chapters, we can conclude that the integration of AI and ML techniques holds great potential for enhancing next-generation networks across various aspects. The incorporation of AI and ML methods enables realizing heightened Quality of Service (QoS) standards, facilitates precise prediction of future network conditions, and aids in the planning and management of network resources, including wireless links, well in advance. Our dissertation demonstrates that by adopting and customizing ML techniques, higher levels of accuracy can be achieved. The advantages of our proposed framework can be summarized as follows:

- Enhanced Accuracy: The proposed framework exhibits high accuracy in diverse communication tasks, such as CFO estimation, CSI feedback, and link adaptation.
- Real-Time Performance: By leveraging the proposed framework, substantial time savings are realized compared to conventional techniques, enabling real-time execution of numerous tasks.
- Scalability: The proposed framework demonstrates high scalability, accommodating high-dimensional inputs and outputs. This adaptability enables the framework to be effectively employed with various transmit and receive antenna configurations.
- Task Integration: Our framework facilitates the integration of multiple tasks into a single step. For instance, we exemplified this capability by combining CSI compression with link adaptation in an end-to-end model. This approach offers several advantages to modern communications systems, ranging from time efficiency to comprehensive resource management and provisioning.

Despite the aforementioned advantages of the proposed framework, certain areas for improvement warrant attention, including:

- Incorporating Channel Estimation: The current framework does not encompass channel estimation in the loop. Integrating channel estimation into the system can significantly enhance reliability and performance, especially when jointly optimized with other stages in the pipeline.

- Comprehensive CFO Estimation: While we addressed CFO estimation as a standalone problem, further exploration could involve integrating CFO estimation with other tasks. Designing an end-to-end model that includes CFO estimation alongside CSI feedback and link adaptation could be of great interest.

- Energy Considerations: The proposed framework primarily focused on system throughput and reliability, but it neglected to consider the energy consumption aspect. Considering energy efficiency is crucial in the design of modern communication systems.

9.2 Recommendations and Future Work

Based on our previous discussion, we can define some directions for future work such as:

- Although ensemble models obtained a competitive performance in the CFO estimation problem as shown in Chapter 1, more efficient dimensionality reduction techniques can be exploited. Moreover, automatically adapting the model architecture through automated ML techniques can be a promising direction to explore.

- Joint optimization brings several advantages to the overall system performance. In this work, we considered joint optimization of CSI compression and link adaptation. However, other functionalities that also depend on the CSI have not been considered such as beamforming. We think jointly optimizing CSI compression and beamforming could be a very promising direction.

- Instead of the fixed compression ratio considered in the CSI compression problem, we can gain a lot of flexibility, bandwidth, and performance gains if we considered an adaptive approach for specifying compression ratios. Adapting compression ratios can depend on the current CSI matrix, feedback channel, and the correlation between the old and new CSI matrices.

- Although our objective in the link adaptation work is maximizing throughout and link reliability, we can incorporate more factors such as power consumption. A large margin of improvement can be reached if we considered the power cost of these methods.

- The lifetime of the proposed models has not been verified after deployment. Some models can give very good performance on the training/testing datasets, but this performance is degraded over time due to many factors including but not limited to data drift.

APPENDIX I

PROOF OF THEOREM 1

In this appendix, we provide the derivation of theorem 1 as given by Du (2018). Assume the distribution $\mathscr{D}$ is a $(\lambda_0, \varepsilon/3, n)$-non-degenerate distribution. Then, with a probability $(1-\varepsilon/3)$, the minimum eigenvalue of the Gram matrix is greater than or equal λ_0 such that: $\lambda_{min}(\mathbf{H}^{\infty}) \geq \lambda_0$. For any sample $\mathscr{S}$, the following points are always satisfied:

- First:

$$\Phi(\mathbf{W}(k)) \leq (1 - \frac{\eta \lambda_0}{2})^k \, O(\frac{n}{\varepsilon}) \leq \frac{1}{2} \tag{11}$$

  where $\mathbf{W}(k)$ is the weight matrix at the k^{th} iteration. Accordingly, the upper bound for the training error can be derived as:

$$\begin{aligned} L_{\mathscr{S}}(f_{\mathbf{W}(k),\mathbf{a}}) &= \frac{1}{n}\sum_{i=1}^{n} L(f_{\mathbf{W}(k),\mathbf{a}}(x_i), y_i) && \text{(12a)} \\ &= \frac{1}{n}\sum_{i=1}^{n} L(u_i(k), y_i) && \text{(12b)} \end{aligned}$$

  where $u_i(k)$ is the prediction of the neural network at the k^{th} epoch for the i^{th} observation.

$$\begin{aligned} L_{\mathscr{S}}(f_{\mathbf{W}(k),\mathbf{a}}) &= \frac{1}{n}\sum_{i=1}^{n} [L(u_i(k), y_i), l(y_i, y_i)] && \text{(13a)} \\ &\leq \frac{1}{n}\sum_{i=1}^{n} |L(u_i(k) - y_i)| && \text{(13b)} \\ &\leq \frac{1}{\sqrt{n}} ||L(\mathbf{u}(k) - \mathbf{y})||_2 && \text{(13c)} \\ &= \sqrt{\frac{2\Phi(\mathbf{W}(k))}{n}} && \text{(13d)} \\ &\leq \frac{1}{\sqrt{n}} && \text{(13e)} \end{aligned}$$

- Second: $||\mathbf{W}_r(k) - \mathbf{W}_r(0)||_2 \leq R \quad (\forall r \in [m])$ and $||\mathbf{W}(k) - \mathbf{W}(0)||_F \leq B$, where:

$$R = O(\frac{n}{\lambda_0 \sqrt{m}\sqrt{\varepsilon}}), \tag{14}$$

$$B = \sqrt{y^T(\mathbf{H}^\infty)^{-1}y} + O(\frac{nk}{\lambda_0 \varepsilon}) + \frac{\text{poly}(n, \lambda_0^{-1}, \varepsilon^{-1})}{k^{1/2}m^{1/4}} \tag{15}$$

It is worth to note that $B \leq O(\sqrt{\frac{n}{\lambda_0}})$.

- Third: Let B_i = i (i = 1, 2, . . .). Simultaneously for all i, the function class $F_{R_i,B_i}^{W(0),a}$ has Rademacher complexity bounded as:

$$\begin{aligned} &R_S(F_{R_i,B_i}^{W(0),a}) \leq \\ &\frac{B_i}{\sqrt{2n}}(1 + (\frac{2\log\frac{10}{\varepsilon}}{m})^{1/4}) + \frac{2R^2\sqrt{m}}{k} + R\sqrt{2\log\frac{10}{\varepsilon}}. \end{aligned} \tag{16}$$

Let i^* be the smallest integer such that: $B \leq B_{i^*}$. Then we have $i^* \leq O(\sqrt{\frac{n}{\lambda_0}})$ and $B_{i^*} \leq B+1$. From above we know $f_{W(k),a}$ ε $F_{R_i,B_{i^*}}^{W(0),a}$, and:
$R_S(F_{R_i,B_{i^*}}^{W(0),a})$

$$\leq \frac{B+1}{\sqrt{2n}}(1 + (\frac{2\log\frac{10}{\varepsilon}}{m})^{1/4}) + \frac{2R^2\sqrt{m}}{k} + R\sqrt{2\log\frac{10}{\varepsilon}} \tag{17}$$

$$\begin{aligned} &= \sqrt{\frac{y^T(H^\infty)^{-1}y}{2n}}(1 + (\frac{2\log\frac{10}{\varepsilon}}{m})^{1/4}) + \frac{1}{\sqrt{n}} + O(\frac{\sqrt{n}k}{\lambda_0\varepsilon}) \\ &+ \frac{\text{Poly}(n, \lambda_0^{-1}, \varepsilon^{-1})}{m^{1/4}k^{1/2}} + \frac{2R^2\sqrt{m}}{k} + R\sqrt{2\log\frac{10}{\varepsilon}} \end{aligned} \tag{18}$$

$$\begin{aligned} &\leq \sqrt{\frac{y^T(H^\infty)^{-1}y}{2n}} + \sqrt{\frac{\sqrt{n}\lambda_0^{-1}\sqrt{n}}{2n}}(\frac{2\log\frac{10}{\varepsilon}}{m})^{1/4}) + \frac{1}{\sqrt{n}} \\ &+ O(\frac{\sqrt{n}k}{\lambda_0\varepsilon}) + \frac{\text{Poly}(n, \lambda_0^{-1}, \varepsilon^{-1})}{m^{1/4}k^{1/2}} \end{aligned} \tag{19}$$

$$= \sqrt{\frac{y^T (H^\infty)^{-1} y}{2n}} + \frac{1}{\sqrt{n}} + O(\frac{\sqrt{n}k}{\lambda_0 \varepsilon}) + \frac{\text{Poly}(n, \lambda_0^{-1}, \varepsilon^{-1})}{m^{1/4} k^{1/2}} \tag{20}$$

$$\leq \sqrt{\frac{y^T (H^\infty)^{-1} y}{2n}} + \frac{2}{\sqrt{n}} \tag{21}$$

Next, from the theory of *Rademacher* complexity and a union bound over a finite set of different i's, for any random initialization $(\mathbf{W}_{(0),a})$, with probability at least $(1-\varepsilon/3)$ over the sample $\mathscr{S}$, we have:

$$\begin{aligned} &\text{Sup}_{f \varepsilon F_{R_i,B_i}^{W(0),a}} \{L_D(f) - L_S(f)\} \leq 2R_S(F_{R_i,B_i}^{W(0),a}) \\ &+ O\left(\sqrt{\frac{\log \frac{n}{\lambda_0 \varepsilon}}{n}}\right), \forall_i \varepsilon \left\{1, 2, \ldots, O\left(\sqrt{\frac{n}{\lambda_0}}\right)\right\} \end{aligned} \tag{22}$$

Finally, taking a union bound, we know that with probability at least $(1-\frac{2}{3}\varepsilon)$ over the sample $\mathscr{S}$ and the random initialization $(\mathbf{W}_{(0),a})$ a), the followings are all satisfied (for some i^*):

$$L_s\left(f_{W(k),a}\right) \leq \frac{1}{\sqrt{n}}, \tag{23a}$$

$$f_{W(k),a} \in F_{R_i,B_{i^*}}^{W(0),a} \tag{23b}$$

$$R_S(F_{R_i,B_{i^*}}^{W(0),a}) \leq \sqrt{\frac{y^T (H^\infty)^{-1} y}{2n}} + \frac{2}{\sqrt{n}} \tag{23c}$$

$$\begin{aligned} &\text{Sup}_{f \varepsilon F_{R_i,B_{i^*}}^{W(0),a}} \{L_D(f) - L_S(f)\} \leq 2R_S(F_{R_i,B_{i^*}}^{W(0),a}) \\ &+ O\left(\sqrt{\frac{\log \frac{n}{\lambda_0 \varepsilon}}{n}}\right) \end{aligned} \tag{24}$$

These together imply:

$$L_D\left(f_{W(k),a}\right) \leq \frac{1}{\sqrt{n}} + 2R_S(F^{W(0),a}_{R_i,B_{i^*}}) + O\left(\sqrt{\frac{\log\frac{n}{\lambda_0\varepsilon}}{n}}\right) \tag{25a}$$

$$\leq \frac{1}{\sqrt{n}} + 2\left(\sqrt{\frac{y^T(H^\infty)^{-1}y}{2n}}\right) + O\left(\sqrt{\frac{\log\frac{n}{\lambda_0\varepsilon}}{n}}\right) \tag{25b}$$

$$= \sqrt{\frac{2y^T(H^\infty)^{-1}y}{n}} + O\left(\sqrt{\frac{\log\frac{n}{\lambda_0\varepsilon}}{n}}\right) \tag{25c}$$

This completes the proof.

BIBLIOGRAPHY

(2023). MarketsandMarkets Report [Format]. Consulted at https://www.marketsandmarkets.com/.

3rd Generation Partnership Project (3GPP). (2020). NR; Physical Channels and Modulation. *TS*, 15.8.0, 38.211.

Abbasalipour, R. & Mirmohseni, M. (2022). Privacy-aware Distributed Hypothesis Testing in Gray-Wyner Network with Side Information. *arXiv preprint arXiv:2202.02307*.

Abdelmoaty, A. & et al. (2022). Resilient Topology Design for Wireless Backhaul: A Deep Reinforcement Learning Approach. *IEEE Wireless Communications Letters*, 1-1. doi: 10.1109/LWC.2022.3207358.

Abdi, Y. & Ristaniemi, T. (2020). The Max-Product Algorithm Viewed as Linear Data-Fusion: A Distributed Detection Scenario. *IEEE Transactions on Wireless Communications*, 19(11), 7585–7597.

Abdzadeh-Ziabari, H., Zhu, W.-P. & Swamy, M. (2017). Joint Maximum Likelihood Timing, Frequency-offset, and Doubly Selective Channel Estimation for OFDM Systems. *IEEE Transactions on Vehicular Technology*, 67(3), 2787–2791.

Alamgir, M., Sultana, M. N. & Chang, K. (2020). Link adaptation on an underwater communications network using machine learning algorithms: Boosted regression tree approach. *IEEE access*, 8, 73957–73971.

Amer, H. (2020). *Image/Video Compression: Human and Computer Vision Perspectives*. (Ph.D. thesis, University of Waterloo, Canada).

(2021). Data-centric AI. Consulted at https://https-deeplearning-ai.github.io/data-centric-comp/.

Aoudia, F. A. & et al. (2022). Deep Learning-Based Synchronization for Uplink NB-IoT. *IEEE Global Communications Conference (GLOBECOM)*, pp. 1478–1483.

Arora, S., Du, S., Hu, W., Li, Z. & Wang, R. (2019). Fine-grained Analysis of Optimization And Generalization for Overparameterized Two-layer Neural Networks. *International Conference on Machine Learning (ICML)*, pp. 322–332.

Babajide Mustapha, I. & Saeed, F. (2016). Bioactive Molecule Prediction using Extreme Gradient Boosting. *Molecules*, 21(8), 983.

Bansal, G., Nushi, B., Kamar, E., Lasecki, W. S., Weld, D. S. & Horvitz, E. (2019). Beyond accuracy: The role of mental models in human-ai team performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1), 2–11.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Blanquez-Casado, F. & et al. (2019). Link adaptation mechanisms based on logistic regression modeling. *IEEE Communications Letters*, 23(5), 942–945.

Blau, Y. & Michaeli, T. (2019). Rethinking lossy compression: The rate-distortion-perception tradeoff. *International Conference on Machine Learning (ICML)*, pp. 675–685.

Bouchoucha, T. & et al. (2015). Distributed Estimation Based on Observations Prediction in Wireless Sensor Networks. *IEEE Signal Processing Letters*, 22(10), 1530–1533.

Bultitude, Y. d. J. & Rautiainen, T. (2007). IST-4-027756 WINNER II D1. 1.2 V1. 2 WINNER II channel models. *EBITG, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep., Tech. Rep.*

Cao, Z., Shih, W.-T., Guo, J., Wen, C.-K. & Jin, S. (2021). Lightweight convolutional neural networks for CSI feedback in massive MIMO. *IEEE Communications Letters*, 25(8), 2624–2628.

Cerwall, P. & et al. (2020). Ericsson Report [Format]. Consulted at https://www.ericsson.com/assets/local/reports-papers/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf.

Chamberland, J.-F. & Veeravalli, V. V. (2007). Wireless Sensors in Distributed Detection Applications. *IEEE Signal Processing Magazine*, 24(3), 16–25.

Chen, G.-L., Hsu, C.-C. & Wu, M.-H. (2021). Adaptive distribution learning with statistical hypothesis testing for COVID-19 CT scan classification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 471–479.

Chen, J. & et al. (2020). Hybrid Beamforming/combining for Millimeter Wave MIMO: A Machine Learning Approach. *IEEE Transactions on Vehicular Technology*, 69(10), 11353–11368.

Chen, J., Yin, H., Cottatellucci, L. & Gesbert, D. (2017). Feedback mechanisms for FDD massive MIMO with D2D-based limited CSI sharing. *IEEE Transactions on Wireless Communications*, 16(8), 5162–5175.

Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 785–794.

Chen, W., Montojo, J., Lee, J., Shafi, M. & Kim, Y. (2022). The standardization of 5G-Advanced in 3GPP. *IEEE Communications Magazine*, 60(11), 98–104.

Cheng, L. & et al. (2017). Accelerated Sparse Representation for Human Activity Recognition. *International Conference on Information Reuse and Integration*.

Cheng, P., Chen, Z., de Hoog, F. & Sung, C. K. (2016). Sparse blind carrier-frequency offset estimation for OFDMA uplink. *IEEE Transactions on Communications*, 64(12), 5254–5265.

Chengbo, L. & et al. (2009). TV Minimization by Augmented Lagrangian and Alternating Direction Algorithms.

Chinchali, S. P. & et al. (2018). Neural Networks Meet Physical Networks: Distributed Inference Between Edge Devices and the Cloud. *ACM Workshop on Hot Topics in Networks*.

Choi, J., Clerckx, B., Lee, N. & Kim, G. (2011). A new design of polar-cap differential codebook for temporally/spatially correlated MISO channels. *IEEE Transactions on Wireless Communications*, 11(2), 703–711.

Choi, J., Love, D. J. & Bidigare, P. (2014). Downlink Training Techniques for FDD Massive MIMO Systems: Open-loop and Closed-loop Training with Memory. *IEEE Journal of Selected Topics in Signal Processing*, 8(5), 802–814.

Choi, K. & et al. (2019). Neural Joint Source-channel Coding. *International Conference on Machine Learning (ICML)*.

Chougrani, H., Kisseleff, S. & Chatzinotas, S. (2020). Efficient preamble detection and time-of-arrival estimation for single-tone frequency hopping random access in NB-IoT. *IEEE Internet of Things Journal*, 8(9), 7437–7449.

Daniels, R. & Heath, R. W. (2010). Online adaptive modulation and coding with support vector machines. *EWC*.

Daubechies, I., Defrise, M. & De Mol, C. (2004). An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. *Communications on Pure and Applied Mathematics*, 57(11), 1413–1457.

de Bodt, C. & et al. (2018). Nonlinear Dimensionality Reduction with Missing Data using Parametric Multiple Imputations. *IEEE Transactions on Neural Networks and Learning Systems*, 30(4), 1166–1179.

Deng, D.-J. & et al. (2017). IEEE 802.11 ax: highly efficient WLANs for intelligent information infrastructure. *IEEE Communications Magazine*, 55(12), 52–59.

Dong, X. & et al. (2020). A survey on Ensemble Learning. *Frontiers of Computer Science*, 14(2), 241–258.

Dong, Z. & et al. (2018). Machine learning based link adaptation method for MIMO system. *PIMRC*.

Dreifuerst, R. M. & et al. (2020). Deep Learning-based Carrier Frequency Offset Estimation with One-bit ADCs. *International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5.

Du, H., Teng, S., Chen, H., Ma, J., Wang, X., Gou, C., Li, B., Ma, S., Miao, Q., Na, X. et al. (2023). Chat with ChatGPT on Intelligent Vehicles: An IEEE TIV Perspective. *IEEE Transactions on Intelligent Vehicles*.

Du, S. S. e. a. (2018). Gradient Descent Provably Optimizes Over-parameterized Neural Networks. *arXiv preprint arXiv:1810.02054*.

Elbrächter, D. & et al. (2019). Deep Neural Network Approximation Theory. *arXiv preprint arXiv:1901.02220*.

Elwekeil, M. & et al. (2018). Deep Convolutional Neural Networks for Link Adaptations in MIMO-OFDM Wireless Systems. *IEEE Wireless Communications Letters*, 8(3), 665–668.

Escamilla, P., Wigger, M. & Zaidi, A. (2018). Distributed hypothesis testing with concurrent detections. *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 166–170.

Faulkner, M. & et al. (2011). The Next Big One: Detecting Earthquakes and Other Rare Events From Community-based Sensors. *ACM/IEEE International Conference on Information Processing in Sensor Networks*.

Fazai, R., Mansouri, M., Abodayeh, K., Trabelsi, M., Nounou, H. & Nounou, M. (2019). Machine learning-based statistical hypothesis testing for fault detection. *2019 4th Conference on Control and Fault Tolerant Systems (SysTol)*, pp. 38–43.

Gao, Z. & et al. (2018). Compressive Sensing Techniques for Next-generation Wireless Communications. *IEEE Wireless Communications*, 25(3), 144–153.

Ge, A. & et al. (2015a). PCA based limited feedback scheme for massive MIMO with Kalman filter enhancing performance. *IEEE International Conference on Communications in China (ICCC)*.

Ge, A. & et al. (2015b). Principal component analysis based limited feedback scheme for massive MIMO systems. *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 326–331.

Glasmachers, T. (2017). Limits of End-to-End Learning. *Asian Conference on Machine Learning (ACML)*, pp. 17–32.

González, J. L. S. & et al. (2019). Energy-efficient Indoor Localization WiFi-fingerprint System: An Experimental Study. *IEEE Access*, 7, 162664–162682.

Guo, J. & et al. (2020). Convolutional Neural Network-based Multiple-rate Compressive Sensing for Massive MIMO CSI Feedback: Design, Simulation, and Analysis. *IEEE Transactions on Wireless Communications*, 19(4), 2827–2840.

Guo, J., Wen, C.-K. & Jin, S. (2020). Deep Learning-based CSI Feedback for Beamforming in Single-and Multi-cell Massive MIMO Systems. *IEEE Journal on Selected Areas in Communications*.

Guo, Y. & et al. (2012). Human Activity Recognition by Fusing Multiple Sensor Nodes in the Wearable Sensor Systems. *Journal of Mechanics in Medicine and Biology*, 12(05), 1250084.

He, K. & et al. (2015). Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification. *International Conference on Computer Vision (ICCV)*, pp. 1026–1034.

He, W. & et al. (2012). Recognition of Human Activities with Wearable Sensors. *EURASIP Journal on Advances in Signal Processing*, 2012(1), 108.

Head, T., MechCoder, G. L., Shcherbatyi, I. et al. (2018). Scikit-optimize/scikit-optimize: v0. 5.2. *Zenodo*.

Hu, P. & et al. (2020). Starfish: Resilient Image Compression for AIoT Cameras. *Conference on Embedded Networked Sensor Systems*, pp. 395–408.

Huang, M., Huang, L., Guo, C., Zhang, P., Zhang, J. & Yang, L.-L. (2017). Carrier frequency offset estimation in uplink OFDMA systems: An approach relying on sparse recovery. *IEEE Transactions on Vehicular Technology*, 66(10), 9592–9597.

Hussien, M. & et al. (2021). Towards More Reliable Deep Learning-based Link Adaptation for WiFi 6. *International Conference on Communications*, pp. 1–6.

Hussien, M., Nguyen, K. K. & Cheriet, M. (2020). PRVNet: Variational Autoencoders for Massive MIMO CSI Feedback. *arXiv preprint arXiv:2011.04178*.

Hussien, M., Nguyen, K. K. & Cheriet, M. (2022). PRVNet: A Novel Partially-Regularized Variational Autoencoders for Massive MIMO CSI Feedback. *Wireless Communications and Networking Conference (WCNC)*, pp. 2286–2291.

Hussien, M. e. a. (2021). Fault-Tolerant 1-bit Representation for Distributed Inference Tasks in Wireless IoT. *International Conference on Network and Service Management (CNSM)*, pp. 427–431.

Huynh, D. T. G. (2008). *Human Activity Recognition with Wearable Sensors*. (Ph.D. thesis, Technische Universität).

ITUR-R. Recommendation ITU-R P.1407-6: multipath propagation and parameterization of its characteristics. ITU-R.

Jagannath, J. & et al. (2018). Artificial neural network based automatic modulation classification over a software defined radio testbed. *IEEE International Conference on Communications (ICC)*.

Jais, I. K. M., Ismail, A. R. & Nisa, S. Q. (2019). Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, 2(1), 41–46.

Jang, Y. & et al. (2019). Deep Autoencoder Based CSI Feedback with Feedback Errors and Feedback Delay in FDD Massive MIMO Systems. *IEEE Wireless Communications Letters*, 8(3), 833–836.

Jensen, T. L., Kant, S., Wehinger, J. & Fleury, B. H. (2010). Fast Link Adaptation for MIMO OFDM. *IEEE Transactions on Vehicular Technology*, 59(8), 3766–3778.

Jin, H., Song, Q. & Hu, X. (2019). Auto-keras: An Efficient Neural Architecture Search System. *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1946–1956.

Joung, J. (2016). Principal-Component-Analysis-Inspired Channel Feedback Framework: Sorting-and-Sampling and Interpolation-and-Rearrangement. *IEEE Communications Letters*, 20(10), 2043–2046.

Joung, J., Kurniawan, E. & Sun, S. (2016). Channel Correlation Modeling and its Application to Massive MIMO Channel Feedback Reduction. *IEEE Transactions on Vehicular Technology*, 66(5), 3787–3797.

Judd, G., Wang, X. & Steenkiste, P. (2008). Efficient channel-aware rate adaptation in dynamic environments. *International Conference on Mobile Systems, Applications, and Services*.

Kang, J. & Choi, W. (2021). Novel Codebook Design for Channel State Information Quantization in MIMO Rician Fading Channels with Limited Feedback. *IEEE Transactions on Signal Processing*, 69, 2858–2872.

Karmakar, R. & et al. (2019). Intelligent MU-MIMO user selection with dynamic link adaptation in IEEE 802.11 ax. *IEEE Transactions on Wireless Communications*, 18(2), 1155–1165.

Karmakar, R., Chattopadhyay, S. & Chakraborty, S. (2016). Dynamic link adaptation in IEEE 802.11 ac: A distributed learning based approach. *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pp. 87–94.

Karmakar, R., Chattopadhyay, S. & Chakraborty, S. (2017). SmartLA: Reinforcement learning-based link adaptation for high throughput wireless access networks. *Computer Communications*, 110, 1–25.

Kim, S. M., Jung, B. C. & Sung, D. K. (2015). Joint Link Adaptation and User Scheduling with HARQ in Multicell Environments. *IEEE Transactions on Vehicular Technology*, 65(3), 1292–1302.

Kingma, D. P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. & Welling, M. (2013). Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A. & et al. (2009). Learning Multiple Layers of Features From Tiny Images.

Ku, G. & Walsh, J. M. (2014). Resource allocation and link adaptation in LTE and LTE advanced: A tutorial. *IEEE communications Surveys & Tutorials*, 17(3), 1605–1633.

Kuo & et al. (2012). Compressive Sensing Based Channel Feedback Protocols for Spatially-correlated Massive Antenna Arrays. *Wireless Communications and Networking Conference (WCNC)*.

Lauriola, I., Lavelli, A. & Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470, 443–456.

LeCun, Y. (1998). The Mnist Database of Handwritten Digits. Consulted at http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444.

Lee, H., Lee, S. H. & Quek, T. Q. (2019). Deep Learning for Distributed Optimization: Applications to Wireless Resource Management. *IEEE Journal on Selected Areas in Communications*, 37(10), 2251–2266.

Li, A. & et al. (2018). A carrier-frequency-offset resilient OFDMA receiver designed through machine deep learning. *IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6.

Li, C., Yin, W. & Zhang, Y. (2009). TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Agorithm. November 2021.

Li, J. J. & Tong, X. (2020). Statistical hypothesis testing versus machine learning binary classification: distinctions and guidelines. *Patterns*, 1(7), 100115.

Li, L. & et al. (2019). An SDN agent-enabled rate adaptation framework for WLAN. *ICC*.

Li, Y. & Song, R. (2012). Novel scheme of CSI feedback compression for multi-user MIMO-OFDM system. *Procedia Engineering*, 29, 3631–3635.

Li, Y., Nitinawarat, S. & Veeravalli, V. V. (2014). Universal outlier hypothesis testing. *IEEE Transactions on Information Theory*, 60(7), 4066–4082.

Liang, D. & et al. (2018). Variational Autoencoders for Collaborative Filtering. *WWW Conference*, pp. 689–698.

Liang, P. & et al. (2020). Deep Learning and Compressive Sensing-based CSI Feedback in FDD Massive MIMO Systems. *IEEE Transactions on Vehicular Technology*, 69(8), 9217–9222.

Liao, J., Sankar, L., Tan, V. Y. & Calmon, F. P. (2016). Hypothesis testing in the high privacy limit. *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 649–656.

Liao, J., Sankar, L., Calmon, F. P. & Tan, V. Y. (2017). Hypothesis testing under maximal leakage privacy constraints. *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 779–783.

Lin, J.-C., Sun, Y.-T. & Poor, H. V. (2015). Initial Synchronization Exploiting Inherent Diversity for the LTE Sector Search Process. *IEEE Transactions on Wireless Communications*, 15(2), 1114–1128.

Lin, X. (2022). An overview of 5G advanced evolution in 3GPP release 18. *IEEE Communications Standards Magazine*, 6(3), 77–83.

Liu, A. & Lau, V. K. (2016). Impact of CSI Knowledge on the Codebook-based Hybrid Beamforming in Massive MIMO. *IEEE Transactions on Signal Processing*, 64(24), 6545–6556.

Liu, L. & et al. (2012). The COST 2100 MIMO Channel Model. *IEEE Wireless Communications*, 19(6), 92–99.

Liu, Z., Zhang, L. & Ding, Z. (2020). An efficient deep learning framework for low rate massive MIMO CSI reporting. *IEEE Transactions on Communications*, 68(8), 4761–4772.

Liu, Z., Del Rosario, M. & Ding, Z. (2021). A Markovian Model-driven Deep Learning Framework for Massive MIMO CSI Feedback. *IEEE Transactions on Wireless Communications*.

Love, D. J. & Heath, R. W. (2006). Limited feedback diversity techniques for correlated channels. *IEEE Transactions on Vehicular Technology*, 55(2), 718–722.

Lu, C. & et al. (2019). MIMO Channel Information Feedback using Deep Recurrent Network. *IEEE Communications Letters*, 23(1), 188–191.

Lu, C., Xu, W., Jin, S. & Wang, K. (2019). Bit-level optimized neural network for multi-antenna channel quantization. *IEEE Wireless Communications Letters*, 9(1), 87–90.

Lu, L. & et al. (2015). Sparsity-enhancing Basis for Compressive Sensing Based Channel Feedback in Massive MIMO Systems. *Global Communications Conference (GLOBECOM)*.

Lu, Z., Wang, J. & Song, J. (2020). Multi-resolution CSI Feedback with Deep Learning in Massive MIMO System. *IEEE International Conference on Communications (ICC)*, pp. 1–6.

MacKay, D. J. & Mac Kay, D. J. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Majumdar, A. (2018). Blind Denoising Autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1), 312–317.

Martın-Clemente, R. & Zarzoso, V. (2016). On the link between L1-PCA and ICA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3), 515–528.

Martorell, G., Riera-Palou, F. & Femenias, G. (2011). Cross-layer fast link adaptation for MIMO-OFDM based WLANs. *Wireless Personal Communications*, 56, 599–609.

Messner, W. (2023). Hypothesis Testing and Machine Learning: Interpreting Variable Effects in Deep Artificial Neural Networks using Cohen's f2. *arXiv preprint arXiv:2302.01407*.

Metzler, C. A. & et al. (2016). From Denoising to Compressed Sensing. *IEEE Transactions on Information Theory*, 62(9), 5117–5144.

Mielczarek, B. & Krzymien, W. A. (2008). Vector quantized CSI prediction in linear multi-user MIMO systems. *VTC Spring 2008-IEEE Vehicular Technology Conference*, pp. 852–857.

Mismar, F. B., Evans, B. L. & Alkhateeb, A. (2019). Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination. *IEEE Transactions on Communications*, 68(3), 1581–1592.

Mitev, M., Butt, M. M., Sehier, P., Chorti, A., Rose, L. & Lehti, A. (2022). Smart link adaptation and scheduling for IIoT. *IEEE Networking Letters*, 4(1), 6–10.

Mitra, A., Richards, J. A. & Sundaram, S. (2020). A new approach to distributed hypothesis testing and non-bayesian learning: Improved learning rate and byzantine resilience. *IEEE Transactions on Automatic Control*, 66(9), 4084–4100.

Moons, B. & et al. (2017). Minimum Energy Quantized Neural Networks. *Asilomar Conference on Signals, Systems, and Computers*.

Morelli, M. & Moretti, M. (2012). Carrier frequency offset estimation for OFDM direct-conversion receivers. *IEEE transactions on wireless communications*, 11(7), 2670–2679.

Morelli, M. & Moretti, M. (2015). A Robust Maximum Likelihood Scheme for PSS Detection and Integer Frequency Offset Recovery in LTE systems. *IEEE Transactions on Wireless Communications*, 15(2), 1353–1363.

Mota, M. P., Araujo, D. C., Neto, F. H. C., de Almeida, A. L. & Cavalcanti, F. R. (2019). Adaptive modulation and coding based on reinforcement learning for 5g networks. *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6.

Nassef, O. & et al. (2022). A survey: Distributed Machine Learning for 5G and beyond. *Computer Networks*, 207, 108820.

Nassralla, M. H. & et al. (2015). A Low-complexity Detection Algorithm for the Primary Synchronization Signal in LTE. *IEEE Transactions on Vehicular Technology*, 65(10), 8751–8757.

Neal, B. & et al. (2018). A Modern Take on the Bias-variance Tradeoff in Neural Networks. *arXiv preprint arXiv:1810.08591*.

Netzer, Y. & et al. (2011). Reading Digits in Natural Images with Unsupervised Feature Learning.

Nguyen, L. (2017). Tutorial on Support Vector Machine. *Applied and Computational Mathematics*, 6(4-1), 1–15.

Ninkovic, V. & et al. (2021). Deep learning-based Packet Detection and Carrier Frequency Offset Estimation in IEEE 802.11 ah. *IEEE Access*, 9, 99853–99865.

Oniga, S. & Jozsef, S. (2015). Optimal Recognition Method of Human Activities using Artificial Neural Networks. *Measurement Science Review*, 15(6), 323.

Ota, K. & et al. (2019). Performance of physical cell ID detection probability considering frequency offset for NR radio interface. *IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–6.

O'Shea, T. & Hoydis, J. (2017). An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4), 563–575.

Park, S., Daniels, R. C. & Heath, R. W. (2015). Optimizing the target error rate for link adaptation. *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Pearson, K. (1901). LIII: On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559–572.

Pratt, J. W., Raiffa, H. & Schlaifer, R. (1995). *Introduction to Statistical Decision Theory*. MIT press.

Qing, C., Cai, B., Yang, Q., Wang, J. & Huang, C. (2019). Deep learning for CSI feedback based on superimposed coding. *IEEE Access*, 7, 93723–93733.

Raghavan, A. & Baras, J. S. (2019). Binary Hypothesis Testing By Two Collaborating Observers: A Fresh Look. *Mediterranean Conference on Control and Automation (MED)*.

Raghavan, V., Heath, R. W. & Sayeed, A. M. (2007). Systematic Codebook Designs for Quantized Beamforming in Correlated MIMO Channels. *IEEE Journal on Selected Areas in Communications*, 25(7), 1298–1310.

Rahman, M. S. & Wagner, A. B. (2012). On the optimality of binning for distributed hypothesis testing. *IEEE Transactions on Information Theory*, 58(10), 6282–6303.

Rajaram, A., Dinis, R., Jayakody, D. N. K. & Kumar, N. (2018). Receiver design to employ simultaneous wireless information and power transmission with joint CFO and channel estimation. *IEEE Access*, 7, 9678–9687.

Ramadan, K. & et al. (2020). Joint Low-complexity Equalization and CFO Estimation and Compensation for UWA-OFDM Communication Systems. *International Journal of Communication Systems*, 33(3), e3972.

Rao, X. & Lau, V. K. (2014). Distributed Compressive CSIT Estimation and Feedback for FDD Multi-user Massive MIMO Systems. *IEEE Transactions on Signal Processing*, 62(12), 3261–3271.

Reddy, G. T. & et al. (2020). Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access*, 8, 54776–54788.

Rhim, J. B., Varshney, L. R. & Goyal, V. K. (2011). Conflict in distributed hypothesis testing with quantized prior probabilities. *2011 Data Compression Conference*, pp. 313–322.

Rodriguez-Conde, I., Campos, C. & Fdez-Riverola, F. (2023). Horizontally Distributed Inference of Deep Neural Networks for AI-Enabled IoT. *Sensors*, 23(4), 1911.

Salehkalaibar, S. & et al. (2018). On Hypothesis Testing Against Conditional Independence with Multiple Decision Centers. *IEEE Transactions on Communications*, 66(6), 2409–2420.

Salim, O. H., Nasir, A. A., Xiang, W. & Kennedy, R. A. (2014). Joint channel, phase noise, and carrier frequency offset estimation in cooperative OFDM systems. *2014 IEEE International Conference on Communications (ICC)*, pp. 4384–4389.

Sambasivan, N. e. a. (2021). "Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI". *CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Saxena, V. & et al. (2019). Contextual Multi-armed Bandits for Link Adaptation in Cellular Networks. *Workshop on Network Meets AI & ML.*

Saxena, V. & Jaldén, J. (2020). Bayesian link adaptation under a BLER target. *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5.

Saxena, V., Tullberg, H. & Jaldén, J. (2021). Reinforcement learning for efficient and tuning-free link adaptation. *IEEE Transactions on Wireless Communications*, 21(2), 768–780.

Sayood, K. (2017). *Introduction to Data Compression*. Morgan Kaufmann.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.

Shahriari, B. & et al. (2015). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Shaked, R., Shlezinger, N. & Dabora, R. (2017). Joint Estimation of Carrier Frequency Offset and Channel Impulse Response for Linear Periodic Channels. *IEEE Transactions on Communications*, 66(1), 302–319.

Shannon, C. E. et al. (1959). Coding Theorems for a Discrete Source with a Fidelity Criterion. *IRE, International Convention Record*, 4(142-163), 1.

Shariatmadari, H. & et al. (2016). Link adaptation design for ultra-reliable communications. *IEEE International Conference on Communications (ICC).*

Sheffet, O. (2018). Locally private hypothesis testing. *International Conference on Machine Learning*, pp. 4605–4614.

Sheng, M. & et al. (2016). Short-time Activity Recognition with Wearable Sensors using Convolutional Neural Network. *ACM SIGGRAPH Conference on Virtual Reality Continuum and its Applications in Industry*, pp. 413–416.

Si, H., Ng, B. L., Rahman, M. S. & Zhang, J. (2017). A novel and efficient vector quantization based CPRI compression algorithm. *IEEE Transactions on Vehicular Technology*, 66(8), 7061–7071.

Siyari, P., Rahbari, H. & Krunz, M. (2019). Lightweight Machine Learning for Efficient Frequency-offset-aware Demodulation. *IEEE Journal on Selected Areas in Communications (JSAC)*, 37(11), 2544–2558.

Snoek, J., Larochelle, H. & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Neural Information Processing Systems (NIPS)*, 25.

Song, H., Seo, W. & Hong, D. (2009). Compressive feedback based on sparse approximation for multiuser MIMO systems. *IEEE Transactions on Vehicular Technology*, 59(2), 1017–1023.

Sreekumar, S., Gündüz, D. & Cohen, A. (2018). Distributed hypothesis testing under privacy constraints. *2018 IEEE Information Theory Workshop (ITW)*, pp. 1–5.

Sreekumar, S., Cohen, A. & Gündüz, D. (2020). Privacy-aware distributed hypothesis testing. *Entropy*, 22(6), 665.

Srivastava, N. & et al. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.

Suto, J. & Oniga, S. (2019). Efficiency Investigation from Shallow to Deep Neural Network Techniques in Human Activity Recognition. *Cognitive Systems Research*, 54, 37–49.

Talpes, E. & et al. (2020). Compute Solution for Tesla's Full Self-driving Computer. *IEEE Micro*, 40(2), 25–35.

Tang, S., Xia, J., Fan, L., Lei, X., Xu, W. & Nallanathan, A. (2022). Dilated convolution based CSI feedback compression for massive MIMO systems. *IEEE Transactions on Vehicular Technology*, 71(10), 11216–11221.

Tay, W. P. & et al. (2009). Bayesian Detection in Bounded Height Tree Networks. *IEEE Transactions on Signal Processing*, 57(10), 4042–4051.

Tong, J., Fu, L. & Han, Z. (2023). Model-Based Thompson Sampling for Frequency and Rate Selection in Underwater Acoustic Communications. *IEEE Transactions on Wireless Communications*.

Tran, S. V. & Eltawil, A. M. (2014). Link adaptation for wireless systems. *Wireless Communications and Mobile Computing*, 14(16), 1509–1521.

Tsai, Y.-R., Huang, H.-Y., Chen, Y.-C. & Yang, K.-J. (2013). Simultaneous multiple carrier frequency offsets estimation for coordinated multi-point transmission in OFDM systems. *IEEE transactions on wireless communications*, 12(9), 4558–4568.

Vempaty, A., He, H., Chen, B. & Varshney, P. K. (2014). On Quantizer Design for Distributed Bayesian Estimation in Sensor Networks. *IEEE Transactions on Signal Processing*, 62(20), 5359–5369.

Wahls, S. & Poor, H. V. (2013). An outer loop link adaptation for BICM-OFDM that learns. *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 719–723.

Wang, T. & et al. (2017). Deep Learning for Wireless Physical Layer: Opportunities and Challenges. *China Communications*, 14(11), 92–111.

Wang, T. & et al. (2018a). Deep learning-based CSI feedback approach for time-varying massive MIMO channels. *IEEE Wireless Communications Letters*, 8(2), 416–419.

Wang, T. & et al. (2018b). Deep learning-based CSI Feedback Approach for Time-varying Massive MIMO Channels. *IEEE Wireless Communications Letters*, 8(2), 416–419.

Wang, Y. & et al. (2021). A Novel Compression CSI Feedback Based on Deep Learning for FDD Massive MIMO Systems. *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5.

Wen, C.-K. & et al. (2018). Deep Learning for Massive MIMO CSI Feedback. *IEEE Wireless Communications Letters*, 7(5), 748–751.

Wong, S. H., Yang, H., Lu, S. & Bharghavan, V. (2006). Robust rate adaptation for 802.11 wireless networks. *International Conference on Mobile Computing and Networking*.

Wu, B., Carr, S., Bharadwaj, S., Xu, Z. & Topcu, U. (2021). Byzantine-resilient distributed hypothesis testing with time-varying network topology. *IEEE Transactions on Automatic Control*, 67(7), 3243–3258.

Xiang, Y. & Kim, Y.-H. (2013). Interactive hypothesis testing against independence. *2013 IEEE International Symposium on Information Theory*, pp. 2840–2844.

Xiao, H. & et al. (2017). Fashion-MNIST: A Novel Image Dataset For Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, H., Gao, F. & Jin, S. (2016). An overview of low-rank channel estimation for massive MIMO systems. *IEEE Access*, 4, 7313–7321.

Xie, X., Zhang, X. & Sundaresan, K. (2013). Adaptive feedback compression for MIMO networks. *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 477–488.

Xu, W. & et al. (2019). Intelligent link adaptation in 802.11 vehicular networks: challenges and solutions. *IEEE Communications Standards Magazine*, 3(1), 12–18.

Yang, A. Y. & et al. (2008). Distributed Segmentation and Classification of Human Actions using a Wearable Motion Sensor Network. *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Yang, A. Y. & et al. (2009). Distributed Recognition of Human Actions using Wearable Motion Sensor Networks. *Journal of Ambient Intelligence and Smart Environments*, 1(2), 103–115.

Yang, P., Xiao, Y., Li, L., Tang, Q., Yu, Y. & Li, S. (2012). Link adaptation for spatial modulation with limited feedback. *IEEE Transactions on Vehicular Technology*, 61(8), 3808–3813.

Yang, P., Xiao, Y., Xiao, M., Guan, Y. L., Li, S. & Xiang, W. (2019). Adaptive spatial modulation MIMO based on machine learning. *IEEE Journal on Selected Areas in Communications*, 37(9), 2117–2131.

Yang, Z. & et al. (2020). Rethinking Bias-variance Trade-off for Generalization of Neural Networks. *International Conference on Machine Learning (ICML)*, pp. 10767–10777.

YİĞİT, H. & Kavak, A. (2013). A learning approach in link adaptation for MIMO-OFDM systems. *Turkish Journal of Electrical Engineering & Computer Sciences*, 21(5), 1465–1478.

Yik, W., Serafini, L., Lindsey, T. & Montañez, G. D. (2022). Identifying Bias in Data Using Two-Distribution Hypothesis Tests. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 831–844.

Ying, D. & et al. (2014). Kronecker Product Correlation Model and Limited Feedback Codebook Design in a 3D Channel Model. *International Conference on Communications (ICC)*, pp. 5865–5870.

Yue, D.-W., Zhang, Y. & Jia, Y. (2015). Beamforming Based on Specular Component for Massive MIMO Systems in Ricean Fading. *IEEE Wireless Communications Letters*, 4(2), 197–200.

Yun, S. & Caramanis, C. (2010). Reinforcement learning for link adaptation in MIMO-OFDM wireless systems. *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5.

Zhang, J. & Wang, H. (2022). Learning the Optimal LLR under Carrier Frequency Offset. *Wireless Communications and Networking Conference (WCNC)*, pp. 441–446.

Zhang, M. & Sawchuk, A. A. (2013). Human Daily Activity Recognition with Sparse Representation using Wearable Sensors. *Journal of Biomedical and Health Informatics*, 17(3), 553–560.

Zhang, T. & et al. (2016). A limited feedback scheme for massive MIMO systems based on principal component analysis. *EURASIP Journal on Advances in Signal Processing*, 2016(1), 1–12.

Zhang, W., Yin, Q. & Wang, W. (2014). Blind closed-form carrier frequency offset estimation for OFDM with multi-antenna receiver. *IEEE Transactions on Vehicular Technology*, 64(8), 3850–3856.

Zhang, Y., Xiang, Y., Zhang, L. Y., Rong, Y. & Guo, S. (2018). Secure wireless communications based on compressive sensing: a survey. *IEEE Communications Surveys & Tutorials*, 21(2), 1093–1111.

Zhao, Z., Barijough, K. M. & Gerstlauer, A. (2018). Deepthings: Distributed adaptive deep learning inference on resource-constrained iot edge clusters. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11), 2348–2359.

Zhu, P. & et al. (2019). Cost Aware Inference for IoT Devices. *International Conference on Artificial Intelligence and Statistics*.