Machine Learning Applications to Short Packet Communications

A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering.

2023

By

Ahlam Ibrahim Alshukaili

Department of Electrical and Electronic Engineering

Contents

Word Count: 19849

List of Tables

List of Figures

Abstract

Recent machine-type communications represent a significant paradigm shift that will revolutionize the design of wireless communication systems. This shift is driven by the promise of ultra-reliable low-latency communications (URLLC) introduced by 5G and 6G technology. Unlike the primary focus of conventional systems, which is on achieving high transmission rates, URLLC aims to support extremely low latency and high reliability in data transmissions. Thus, Short Packet Communication (SPC) is being introduced as a key enabler for URLLC. This thesis aims to enhance the performance of SPC using machine learning (ML) algorithms. Initially, we study the performance analysis of SPC and develop an accurate evaluation of the packet error probability in the presence of interference. Subsequently, we investigate the performance of sparse recovery algorithms within the context of SPC. Specifically, we propose two algorithms, Compressive Sampling Matching Pursuit (CoSaMP) and Stagewise Matching Pursuit (St-OMP), for sparse recovery. Also, we present a general form of the Symbol Error Rate (SER) utilizing pairwise error probability. Further, we investigate the potential application of ML techniques in SPC. We apply supervised learning, namely Support Vector Machine (SVM) and K-Nearest Neighbours (KNN), and compare them with the application of unsupervised learning, Expectation Maximization (EM), to SPC. To mitigate packet overhead, we employ the Label Assisted Transmission (LAT) method. Additionally, we utilize Silhouette Analysis to determine the optimal clustering number. Finally, we successfully use a supervised learning approach to recover the spreading factor in the Long-Range (LoRa) system using SVM and KNN. The applied algorithms showed significant improvements in the performance of SPC compared to the baseline schemes. Specifically, SVM and KNN algorithms show promising results in signal classification with different signal representations.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

I am deeply grateful to the Almighty God for granting me the opportunity to accomplish this work.

I would like to express my deepest gratitude and appreciation to all those who have supported and contributed to the successful completion of this PhD thesis.

First and foremost, I am sincerely grateful to my supervisor, Dr Khairi Hamdi, for his continual guidance, limitless support, boundless patience and continuous encouragement throughout this research journey. I could not have done this without his help, and I will be forever grateful. I extend my gratitude to all my colleagues and friends in Manchester for their support and for being an integral part of my journey. I am blessed to be surrounded by a remarkable community of colleagues and friends at the University of Manchester.

I would also like to thank the government of the Sultanate of Oman for their generous support in funding my studies at the University of Manchester.

My heartfelt appreciation goes out to my parents for their prayers, patience, and generous understanding. Their constant encouragement and endless support make this work possible. I extend a special thanks to my beloved brothers and sisters for their kindness, support and boundless love.

Dedication

With deepest appreciation and affection,

I dedicate this thesis to

my beloved family.

List of Abbreviations

AP	Access Point
AWGN	Additive White Gaussian Noise
BS	Base Station
BER	Bit Error Rate
CSI	Channel State Information
CSS	Chirp Spread Spectrum
CS	Compressive Sensing
CCDF	Complementary Cumulative Distribution Function
CoSaMP	Compressive Sampling Matching Pursuit
dB	Decibel
DNN	Deep Neural Network
EM	Expectation Maximization
FEC	Forward Error Correction
GMM	Gaussian Mixture Model
ISM	Industrial, Scientific, and Medical
IoH	Internet of Health
IoT	Internet of Things
IoV	Internet of Vehicles
IRS	Intelligent Reflecting Surface
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunication Union

KNN	K-nearest Neighbour
LAT	Label Assisted Transmission
LAN	Local Area Network
LoRa	Long Range
LTE	Long Term Evolution
LPWANs	Low-Power Wide-Area Networks
ML	Machine Learning
MLD	Maximum Likelihood Detector
mMTC	Massive Machine-Type Communications
mod	Modulo
MC	Modulation Constraints
MIMO	Multiple-Input Multiple-Output
MMP	Multiple Matching Pursuit
NN	Neural Network
NOMA	Non-Orthogonal Multiple Access
OMP	Orthogonal Matching Pursuit
PEP	Pairwise Error Probability
PHY	Physical Layer
QPSK	Quadrature Phase Shift Keying
SER	Symbol Error Rate
SINR	Signal to Interference Plus Noise Ratio
SNR	Signal to Noise Ratio
SPC	Short packet Communications
SSS	Signal Strength Strategy
St-OMP	Stagewise Matching Pursuit algorithms
SVM	Support Vector Machine
SVC	Sparse Vector Coding
SF	Spreading Factor
TDMA	Time Division Multiple Access
5G	The 5th generation mobile networks

6G	The 6th generation mobile networks
UAV	Unmanned Aerial Vehicle
URLLC	Ultra-Reliable Low Latency Communications
WAN	Wide-Area Network

List of Variables

$\mathcal{O}$	Complexity
$\Pr(\vert)$	Conditional probability
$CN(\mu, v)$	Complex Gaussian distribution with the mean μ and variance v.
erfc	Error function
exp	Exponential
$\forall$	For all
log	Logarithm to base 2
lim	Limit
mod	Modulo operation
ln()	Natural logarithm
$\Vert.\Vert$	Norm
$\mathbb{E}$	The expectation operator
$()^T$	The transpose operator
$\Pr(.)$	Probability

Chapter 1

Introduction

This chapter serves as an introduction to the thesis and is structured as follows: Section 1.1 presents the background of the research. Section 1.3 outlines the motivation behind the work. In Section 1.4, the overall objectives of the thesis are introduced, highlighting the intended outcomes and goals. The major contributions of this thesis are summarized in Section 1.5. In Section 1.6, the author's relevant publications are presented. Finally, in Section 1.7, the thesis structure is outlined, providing a roadmap of the subsequent chapters and their respective content.

1.1. Background

Recently, ultra-reliable low latency communication (URLLC) has been established as a service category to support 5G and beyond. To achieve the requirement of this service category, the International Telecommunication Union (ITU) and the 3rd Generation Partnership Project (3GPP) sets strict requirements for ultra-reliability of nearly 100% with very low latency [2], an essential requirement of sensitive applications such as remote robot surgery and autonomous driving. An important observation within these applications is that the transmitted data primarily consists of control commands (such as move left/right, start/stop, .., etc.) [3]. Thus, the volume of data to be transmitted is extremely small. Moreover, these applications often demand a target decoding error probability of less than 10^{-7} while maintaining a latency below 1 ms [4]. This stringent requirement for low latency imposes a new limit on packet size. Consequently, short-packet communications (SPC) have been

proposed as the essential approach to minimize latency for URLLC.
The provision of services for URLLC applications faces a significant challenge when it comes to SPCs. The upcoming systems are expected to differ from current ones that depend on longer blocklengths to achieve higher bandwidth. In the case of SPC, both the packet length and decoding errors need to be carefully considered. Consequently, a novel design must be developed to meet the demands of this new paradigm. Recently, machine learning has emerged as a highly promising means of tackling various issues in wireless communication, offering significant potential in addressing the challenges associated with SPCs.

1.2. Research Gap in Short Packet Communications

In recent years, SPC has emerged as a pivotal aspect of wireless communication, primarily designed to cater to the needs of URLL. While the concept of SPC is relatively recent, its potential for enabling massive machine-type communication (mMTC) and IoT has garnered significant attention. Despite the growing importance of SPC, there exists a noticeable research gap that needs to be addressed. Another noteworthy development in the wireless communication field is the emergence of ML techniques as powerful tools for enhancing system performance. Machine learning has demonstrated promising results in various wireless communication domains, including spectrum sensing, resource allocation, and channel prediction [5]. Given the evolving landscape of wireless communication, exploring the potential benefits of integrating machine learning into SPC is essential. This section aims to elucidate the research gaps in SPC, specifically focusing on its performance in interference environments and its integration with machine learning techniques.

- In existing SPC literature, the main focus is on the performance of SPC without considering its performance in an interference environment. A lack of research considers the performance of SPC with two static interferences, as mentioned in Section 3.2. To bridge this gap, this research provides a realistic system model where random interference with different distributions is considered.

- One of the recent solutions for SPC design is to adopt compressive sensing and sparse coding to transmit short packets. This technique is sensitive to sparse recovery at the receiver side. This work applies two new sparse recovery algorithms and investigates their performance to meet the requirements of URLL.

- In digital communications, signals are received in groups by nature. One suggested solution is using unsupervised machine learning, such as EM, to cluster the received signals according to the used modulation type and then find the maximum likelihood to recover the desired signal. To reduce the complexity of this method, Silhouette Analysis has been applied to predict the optimal clustering solution. Moreover, supervised machine learning algorithms, namely SVM and KNN, have been developed to recover the transmitted signal directly.

- In the LoRa system, interference is highly proportional to the spreading factor. The exciting research focuses on assigning the spreading factor at the transmitter side, as mentioned in Section 6.2. However, in practice, signals are affected by various natural effects, such as fading. Thus, SVM and KNN have been applied to recover the spreading factor at the receiver to mitigate the interference effect.

1.3. Research Motivation

Machine learning (ML) algorithms have demonstrated encouraging results in addressing various communication system challenges. Recently, SPC has emerged as a new research area in wireless communication systems, driven by the ever-increasing demand for URLLC in numerous applications. With the rapid advancements in wireless technologies and the rapidly increasing number of devices communicating with the Internet of Things (IoT), there is an essential need to develop efficient and robust communication protocols capable of delivering short packets of critical information with minimal latency. Current communication systems, designed primarily for handling larger data packets, struggle to meet the stringent latency requirements of URLLC applications. As a result, novel approaches and techniques are required to address the unique challenges posed by SPCs. Although there have been many efforts to investigate and improve the system performance of SPC, there are still important areas that need attention due to significant deficiencies. The motivation behind this PhD thesis is to address critical challenges associated with SPC performance. Also, the successful applications of ML in wireless communication motivate us to explore its applications to SPC.

Interference poses a significant challenge in wireless communication, affecting the reliability and latency of short-packet transmissions. However, in the literature, there is a noticeable absence of studies that address the impact of interference on SPC performance. To help bridge this gap, SPC performance in an interference environment across different wireless scenarios is investigated. Interference poses a significant challenge in wireless communication, affecting the reliability and latency of short-packet transmissions. By analyzing the impact of interference on SPC performance, this research aims to develop an accurate mathematical expression for error probability that improves the system's performance. The expression will be applied to real wireless communication scenarios.

Recently, sparse vector coding (SVC) has played a pivotal role in SPC. This algorithm enables efficient encoding and decoding of sparse data, which is a characteristic feature of short-packet transmissions. This method converts the transmitted signal to the sparse vector that transmits the non-zero position; as a result, it reduces the packet length without optimizing the packet overhead. However, it is crucial to recover the non-zero position at the receiver. Therefore, the second part of this research comprehensively examined the sparse recovery algorithms.

Furthermore, integrating ML techniques into SPC holds great promise for optimizing system performance. The classification and clustering algorithms also show promising results in wireless communication. When it comes to designing communication systems, the conventional approach is to rely on the receiver having known channel state information (CSI) and known bit-symbol mapping. To achieve this, pilot symbols are transmitted to estimate the CSI. Once this is done, the transmitted symbols can be recovered using maximum likelihood or other low complexity detection schemes. In a simple wireless communication system, symbols are modulated and selected from a pre-defined set of constellation points. This means that the received signals naturally fall into clusters. To take advantage of this, the received signal can be grouped into a cluster with correspondent constellation points, and the bit recovery problem can be formed as a clustering/classification problem.

Additionally, LoRa is implemented to transmit short packets over long-range with low-power. LoRa adopts a unique modulation scheme called chirp spreading

modulation. The transmitted LoRa signal has defined spreading factor values. The spreading factor is highly correlated with the interference. For example, the signal can interfere with another signal that has a different spreading value. Inspired by the unique feature of LoRa and the spreading factor parameter, the spreading factor recovery problem is considered a multiclass problem solved by supervised learning. The system performance can be improved, and the recovery of the target spreading factor can reduce the interference's effect.
By addressing these research areas comprehensively, this PhD thesis aims to contribute to the advancement of SPC performance.

1.4. Aims and Objectives

This research aims to enhance the performance of SPC using ML algorithms. The central focus of this thesis is the application of ML in SPC, addressing a range of complex challenges. This includes the performance of SPC in the presence of random interference, minimizing SPC overheads, recovering non-zero elements of (SVC), and precisely reconstructing spreading factors in LoRa communication systems. The main objectives are summarized as follows:

- To provide an overview of SPC system design and its main challenges.
- To cover advanced applications of ML to SPC.
- To provide a performance analysis of SPC in the presence of interference.
- To investigate the performance analysis of SPC in different real scenarios: binomial and Poisson distribution.
- To study and investigate the performance of different sparse recovery algorithms and show their impacts on SPC.
- To develop supervised and unsupervised machine learning applications to improve the performance of an SPC system.
- To develop a supervised machine learning scheme that aims to further reduce the packet overhead.
- To propose supervised learning schemes to improve the performance of the LoRa system and reduce interference.

1.5. Contributions

- C_1 (Chapter 3): A new mathematical expression of packet error probability for SPC in the presence of random interference is proposed. Then this expression is investigated using two real scenarios where the interference is considered first as a binomial and then as a Poisson distribution.

- C_2 (Chapter 4): Two new algorithms for sparse coding recovery are applied and compared with existing algorithms. The performance of all algorithms is comprehensively examined in various aspects, including recovery error and recovery time. Subsequently, a general error probability expression is introduced using pairwise error probability.

- C_3 (Chapter 5): A supervised ML scheme for symbol detection is developed and compared with unsupervised learning. Then, the Silhouette Analysis method is proposed to determine the optimal number of clusters.

- C_4 (Chapter 5): A new method for spreading factor recovery at the receiver is proposed. This is accomplished by employing the supervised ML algorithms SVM and KNN.

1.6. Author's Publications

- P1. (Chapter 3): Ahlam Alshukaili, and Khairi A. Hamdi. "Accurate Evaluation of Packet Error Probability for Short Packet Communications." IEEE Communications Letters (submitted).

- P.2 (Chapter 4): Ahlam Alshukaili, and Khairi A. Hamdi. "Sparse Recovery Algorithms Implementations for Short Packet Communications." 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring). IEEE, 2022, Finland.

- P.3 (Chapter 6): Ahlam Alshukaili, and Khairi A. Hamdi. "Spreading Factor Recovery in LoRa Using Machine Learning." 2023 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, 2023, Italy.

1.7. Thesis Organization

The thesis is organized as follows:

Chapter 2 provides the fundamental details of SPCs, ML algorithms and the LoRa system. The chapter begins by explaining the differences between finite and infinite blocklengths, highlighting the unique features of SPCs. Subsequently, it describes various types of ML algorithms, providing the fundamental principles and concepts of the specific algorithms employed in this research: Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Sparse Coding and Expectation-Maximization (EM) algorithms. Lastly, the chapter introduces LoRa modulation and its relevant parameters while addressing the distinct types of interference encountered with LoRa systems.

Chapter 3 introduces a new mathematical analysis of the performance of SPC in the presence of interference. The system model is covered in detail, providing a thorough description of the system's condition and the assumptions made. Subsequently, the distribution of Signal-to-Interference-plus-Noise Ratio (SINR) is presented. Following that, the packet error probability is derived. Finally, the analytical expression is examined under two distinct wireless communication scenarios, namely the binomial and Poisson distributions.

Chapter 4 presents four recovery algorithms for SVC. First, an overview of compressive sensing in SPC is provided. Then, the process of each algorithm is explained. Finally, the performance of these algorithms is comprehensively examined.

Chapter 5 presents a ML approach for symbol recovery in SPC. The method for reducing the packet overhead is described. Then, the clustering framework is provided for both supervised and unsupervised ML. Finally, the performance of the algorithms is illustrated by the bit error rate (BER).

Chapter 6 presents a new ML framework aimed at recovering the spreading factor in LoRa systems. The problem of spreading factor recovery is approached as a classification problem, and the two supervised learning algorithms, SVM and KNN, are employed to recover the desired spreading factor. The proposed framework leverages the classification capabilities of SVM and KNN to predict and recover the spreading factor in LoRa systems.

Chapter 7 concludes this thesis and suggests future research that can be carried out based on the work presented in this thesis.

Chapter 2

Background Theory

This chapter covers the background of several of the key concepts presented in this thesis. This includes an overview of short packet communication (SPC), machine learning (ML), and long-range (LoRa).

2.1. Short Packet Communications

The upcoming 5G and future advances networks are designed to enable SPC for critical applications such as industrial automation in which the information commands are expected to be small [6, 7]. However, existing wireless frameworks are unsuitable to be adopted directly for SPC. Current systems tend to support long-packet transmission. However, it is worth mentioning that, in SPC, the control information is not negligible as with the current system of long-packet transmission.

2.1.1. Infinite Blocklength

Before touching on the details of the finite blocklength scheme, we first start with the infinite blocklength because wireless communication systems are normally assumed to work with infinite blocklength scenarios. In this system, the two main performance factors are the capacity [8, 9] and the outage capacity [9, 10]. The channel capacity is defined as the maximal achievable rate at which the signal can be transmitted reliably, and the outage capacity is the maximal achievable rate of the transmitted signal with error probability less than ϵ ($\epsilon > 0$). The two factors are not restricted by the blocklength, and the outage capacity can resolve the capacity by letting ϵ tend to zero and the blocklength L to infinity as

$$C = \lim_{\epsilon \to 0 \mathrm{L} \to \infty} C_\epsilon \tag{2.1}$$

Furthermore, in an infinite blocklength system, the overhead (the additional information for correction and estimation operation called the metadata) is relatively negligible. Thus the overhead does not affect the system performance efficiency, see Section (2.1.3).

2.1.2. Finite Blocklength

On the other hand, transmitting small messages requires designing short packets suitable for this type of information. Current performance parameters are insufficient for short packets to achieve maximum rate. The reason is the overhead must be taken into consideration as the payload data size may be comparable to the metadata [7], as shown in Eq. (2.2). Thus, the overhead and blocklength are important in SPC design. Polyanskiy et al. [11] provided a closed form and theoretical principle of the maximum achievable rate for a finite blocklength system

$$R(\mathrm{L}, \epsilon, SNR) \approx C(SNR) - \sqrt{\frac{V(SNR)}{\mathrm{L}}} Q^{-1}(\epsilon) + \mathcal{O}(\frac{\log \mathrm{L}}{\mathrm{L}}) \tag{2.2}$$

Where L is the blocklength, ϵ is the decoding error probability, SNR is the signal to noise ratio, $C(SNR) = \log(1 + SNR)$ is the Shannon capacity, $V(SNR)$ is the channel dispersion defined as [11]

$$V(SNR) = (\log e)^2 \left(1 - \frac{1}{(1 + SNR)^2}\right) \tag{2.3}$$

and $Q^{-1}(.)$ is the inverse Gaussian of the Q-function [12]

$$Q^{-1}(x) = \int_x^\infty \frac{1}{2\pi} e^{-t^2/2} dt. \tag{2.4}$$

Ignoring the term $\mathcal{O}(\frac{\log \mathrm{L}}{\mathrm{L}})$ (which denotes the remainder terms of order $\frac{\log \mathrm{L}}{\mathrm{L}}$), the maximum achievable rate for Eq. (2.2) can be approximated as

$$R(\mathrm{L}, \epsilon, SNR) \approx C(SNR) - \sqrt{\frac{V(SNR)}{\mathrm{L}}} Q^{-1}(\epsilon). \tag{2.5}$$

Eq. (2.5) is referred to as the normal approximation, based on a Gaussian (normal) distribution. Although Eq. (2.5) has been demonstrated to be accurate for different parameters, its accuracy may be reduced when dealing with extremely short blocklengths (e.g., $\mathrm{L} < 50$) or very small ϵ (e.g., $\epsilon < 10^{-5}$). Therefore, to indicate the accuracy of normal approximation Fig. 2.1 illustrates the achievable coding rate with blocklength for the normal approximation with the upper bound, which represents the converse theorem, and the lower bound represent Shannon's achievability bound [11]. The normal approximation lies between the upper and lower bound and tightens as the blocklength increases, proving the normal approximation's accuracy with the parameters in the derived Eq. (2.5). However, its accuracy may not be valid for a very short blocklength.

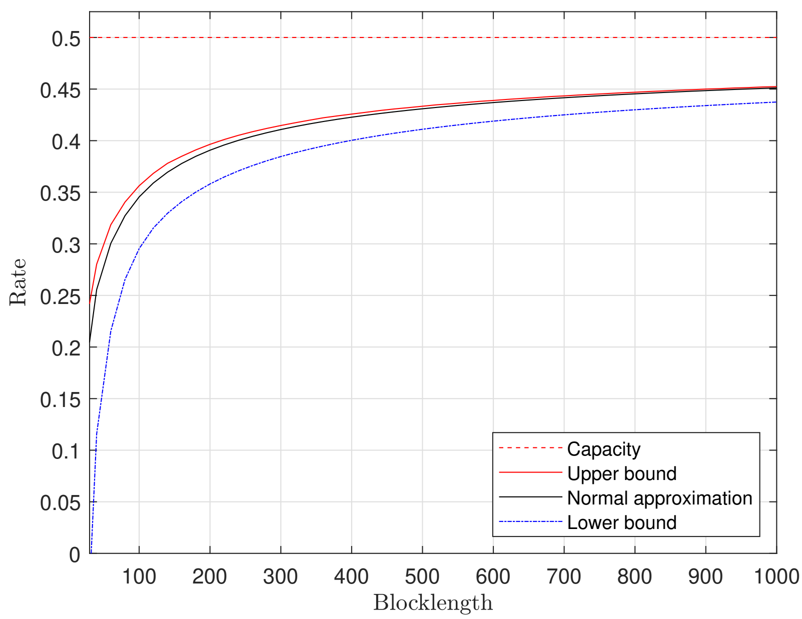


Figure 2.1: Comparison of normal approximation Eq. (2.5) of the achievable coding rate for SNR= 6 dB, $\epsilon = 10^{-3}$ for AWGN channel with upper bound and lower bound. The Figure was generated using the SPECTRE toolbox [1].

2.1.3. Packet Overheads

Packet overhead refers to the additional information included in a transmitted packet alongside the actual data payload. A typical physical (PHY) layer packet structure consists of a header, overhead, payload, and sometimes a trailer. The header contains essential information details such as source, destination addresses, and protocol types. The payload contains the actual transmitted data. The purpose of the packet overhead is to facilitate reliable and efficient data transmission over a network by including necessary control information in the packets for error detection and correction mechanisms, ensuring data integrity during transmission [13]. One crucial component of the packet overhead is the pilot signal, a reference signal embedded in the packet to aid in channel estimation and synchronization at the receiver end. The pilot signal is of great importance as it helps counter the effects of signal distortion and noise during transmission, enabling better decoding of the data and improving overall communication reliability.

Transmitting information over infinite block lengths is demonstrated by the information theory principle, which is considered the foundation of most efficient wireless systems [14]. To ensure the system's efficiency and reliability, a certain number of pilot sequences which contain control information are added to the metadata or overhead. However, in long packets, the size of the metadata is relatively small and not a significant factor compared to the size of the transmitted payload information, as illustrated in Fig. 2.2(a). Hence, the size of the metadata has a negligible impact on the performance of the system.

However, the emerging requirement of using short packets necessitates new considerations. In the case of short packets, the amount of transmitted data is significantly smaller, which can be comparable to the size of the metadata, as depicted in Fig. 2.2(b). Therefore, in the context of short packets, the metadata plays a crucial role. Reducing metadata size may potentially impact the efficiency and reliability of the system. Consequently, the design of short packets with appropriate overhead becomes an open and challenging problem that requires further exploration.

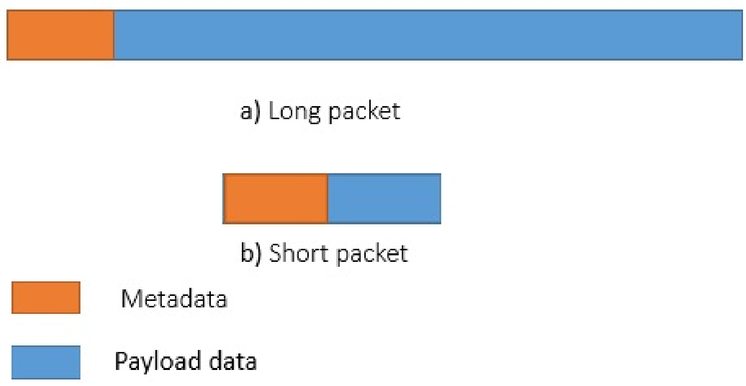


Figure 2.2: Packet structure of long and short packet.

2.2. Machine Learning

Machine learning has emerged as a transformative field within the realm of artificial intelligence, revolutionizing the way we extract knowledge, make predictions, and automate decision-making processes. With its ability to learn from data without explicit programming, ML has found applications in diverse domains, such as computer vision, healthcare, auto-driving and wireless communication [15, 16]. This section serves as an introduction to some fundamental concepts, types, and applications of ML in wireless communication. ML techniques can be broadly classified into three main categories: supervised learning, unsupervised learning, and reinforcement learning; see Fig. 2.3. This section focuses on supervised and unsupervised learning, and we highlight the algorithms used in this thesis: Expectation Maximization (EM), Sparse Coding, Support Vector Machines (SVMs), and K-Nearest Neighbours (KNN).

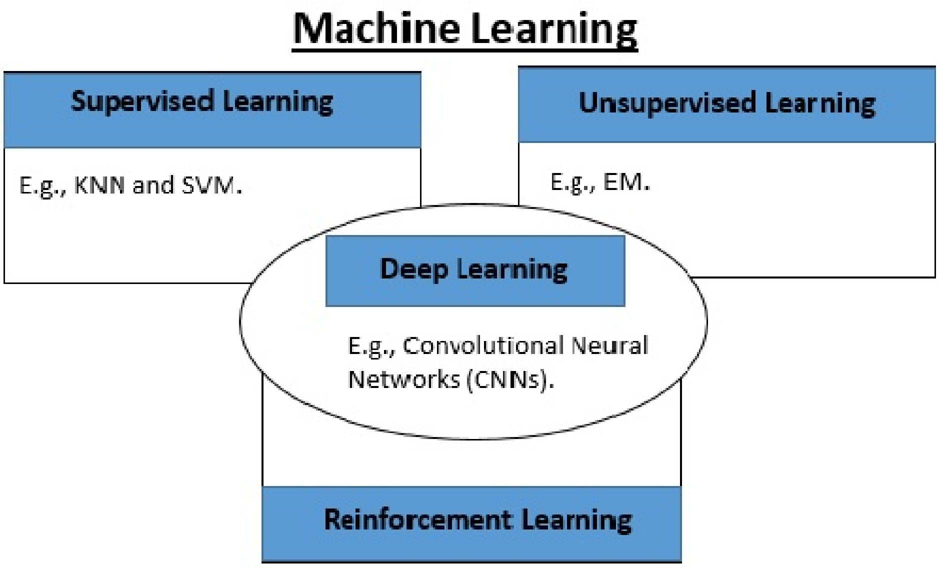


Figure 2.3: Types of machine learning

2.2.1. Supervised Machine Learning

Supervised ML is a fundamental category of ML techniques that plays a key role in solving a wide range of predictive and classification problems [17]. In supervised learning, a model is trained using a labelled dataset, where each label consists of input features and the corresponding known output or target variable [18]. By leveraging this labelled data, supervised learning algorithms aim to learn the underlying patterns and relationships between the input features and the target variable. Examples of popular supervised learning algorithms include linear regression, logistic regression, SVMs, decision trees, random forests, KNN and neural networks. Each algorithm exhibits unique characteristics and is suitable for different types of problems, ranging from simple linear relationships to complex non-linear patterns.

2.2.1.1 Support Vector Machine

The SVM is a powerful learning algorithm widely used for both classification and regression tasks. SVM belongs to the family of supervised learning algorithms and is particularly suitable for tasks involving complex decision boundaries and high-dimensional data. The fundamental concept of SVM is to find the optimal decision boundary that maximally separates the data points to create an effective separation

between different classes. The decision boundary is known as a hyperplane or a set of hyperplanes for high-dimension feature space, and it is selected to maximize the margin, which is defined as the distance between the hyperplane and the nearest data points from different classes. By maximizing the margin, SVM aims to achieve better separation between classes.
The basic formulation of SVM involves mapping the input data into a high dimensional feature space using a kernel function. This transformation allows the algorithm to linearly separate the data points that were inseparable in the original input space. In SVM, the training process involves finding the optimal hyperplane by solving a quadratic optimization problem. The aim is to minimize the classification error while maximizing the margin; see Fig. 2.4. This optimization problem can be formulated as follows [19–21]

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{v}\| + R\sum_{i=1}^{I}\xi_i \\ &\text{s.t.} \quad y_i\left(\mathbf{v}\mathbf{x}_i + b\right) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \tag{2.6}$$

where $\mathbf{x}_i$ and y_i are the training data, $\mathbf{v}$ is the weight vector, and b is the bias term. ξ_i is the slack variable that allows the non-separable data to be misclassified or lie within the margin. The parameter R is a regularization term that controls the trade-off between achieving a large margin and minimizing the classification errors.

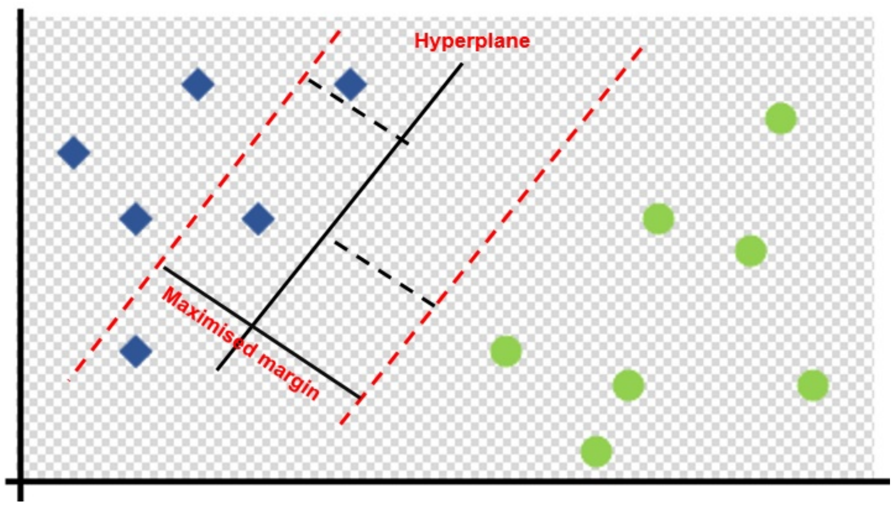


Figure 2.4: SVM algorithm.

2.2.1.2 K-Nearest-Neighbour

The KNN algorithm is a non-parametric learning method used for classification, regression and pattern recognition tasks [22, 23]. It belongs to a family of lazy learning algorithms, as it doesn't make any assumptions about the training data. Instead, it relies on the new data point to predict and classify accordingly.

In the KNN, the "K_n" represent the number of nearest neighbours considered for classification and regression. At its core, the KNN algorithm works on the principle of similarity, assuming that the data points with similar features tend to belong to the same class or have similar output values. Given a new data point, KNN searches the training dataset for the K_n nearest neighbours based on distance metric, typically Euclidean, Manhattan or Minkowski distance [24].

Euclidean distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.7}$$

Manhattan distance

$$d(x,y) = \left|\sum_{i=1}^{n}(x_i - y_i)\right| \tag{2.8}$$

Minkowski distance

$$d(x,y) = \left(\left|\sum_{i=1}^{n}(x_i - y_i)\right|^p\right)^{1/p} \tag{2.9}$$

where x_i, y_i are the training data and test data, n is the number of data points, and p is the order parameter.

The class or value of the new data point is then determined by majority vote or averaging of the labels or values of its K_n nearest neighbours.

The KNN algorithm is advantageous due to its simplicity and ease of implementation [25]. As makes no assumptions about the underlying data distribution, as it is considered a non-parametric method, and this property allows the algorithm to handle complex and non-linear relationships between features and target variables. Additionally, the KNN algorithm can cope with multi-class classification problems and can accommodate mixed-type data. However, its computational complexity increases rapidly with the size of the dataset, as it requires calculating distances between the new data and all the training data [26]. This makes the process computationally expensive for large datasets. Moreover, the algorithm can be sensitive to the choice of the value of K_n and the distance metric, requiring careful selection of both to achieve optimal performance.

2.2.2. Unsupervised Machine Learning

Unsupervised ML is a powerful branch of ML that focuses on extracting meaningful patterns, structures, and relationships from unlabelled data. Unlike supervised learning, unsupervised learning does not require predefined labels or target variables. Instead, it aims to discover hidden insights and structures within the data, enabling valuable discoveries and uncovering previously unknown patterns.

2.2.2.1 Expectation Maximization

EM is a statistical, iterative learning algorithm. EM arose as a means of solving incomplete or missing data problems, where the observed data is partially observed or contains unobserved variables [27]. In such cases, direct estimation of the model parameters becomes challenging or impossible. EM provides a principled approach to estimating these parameters by incorporating the missing or unobserved data through a latent variable framework.

At the core of the EM algorithm is the principle of maximum likelihood estimation (MLE). The goal is to find the set of parameters that maximizes the likelihood function, which measures the probability of observing the given data under the assumed statistical model. However, in the presence of missing or noisy data, the likelihood function cannot be directly solved. EM algorithms offer a systematic way to handle this problem and iteratively estimate the parameter that optimizes the likelihood. This can be achieved in two steps, the E-step and M-step, see Fig. 2.5.

E-step: this makes an initial estimation of the unknown parameters. It involves computing the posterior distribution of the latent variables given the observed data and the current parameter estimates. It describes the process of evaluating the probabilities or as commonly termed in EM algorithm responsibilities of the latent variables associated with each data point, conditioned on the current parameter values.

M-step: in this step, the algorithm updates the model parameters by maximizing the expected complete data log-likelihood obtained from the E-step. This step treats the expected values of the latent variables as if they were observed values, thus simplifying the estimation process. The M-step involves solving an optimization problem to obtain the parameter values that maximize the expected log-likelihood. The mathematical description is provided in detail in Chapter 5.

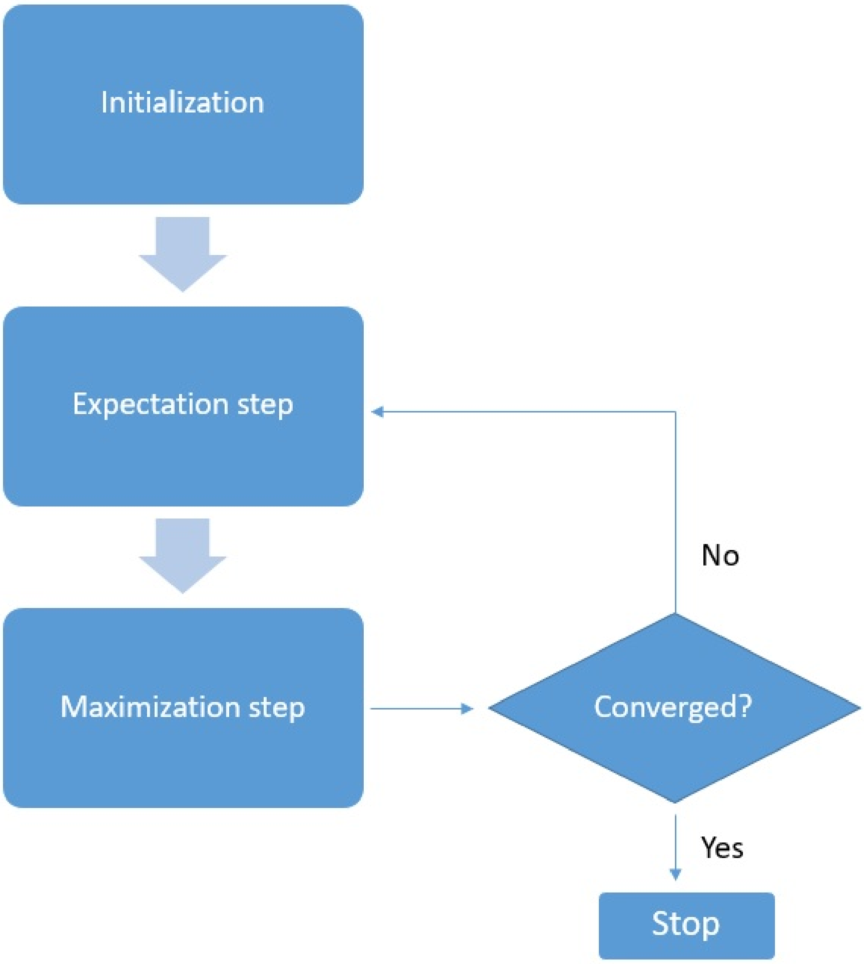


Figure 2.5: EM algorithm process.

2.2.2.2 Sparse Coding

Sparse coding is an unsupervised learning algorithm that discovers an alternative and efficient representation of input data, known as a sparse representation. The fundamental idea behind sparse coding is to express the data using sparse vectors, where most elements are set to zero [28]. By doing so, sparse coding seeks to capture only the essential features and patterns inherent in the data, leading to a more concise and meaningful representation. This approach not only reduces the memory and computational requirements but also enhances the data transmission, allowing for a deeper understanding of its underlying structure. By finding the sparse representation, sparse coding plays an important role in various fields like signal processing, image recognition, and natural language processing, contributing to the advancement of ML and pattern recognition applications [29].

2.2.2.3 Silhouette Analysis

Silhouette Analysis is a method used for unsupervised ML to indicate the best number of clusters. This method does not require any initial representation, such as the clusters' centroids; instead, it depends on the pairwise distance between all data [30]. The silhouette value is represented as [31]

$$s(n) = \frac{b(n) - a(n)}{\max\{a(n), b(n)\}} \tag{2.10}$$

where n is the number of data, $a(n)$ is the distance between the data point and its cluster and b(n) is the distance of the data point to the nearest cluster.
The value of silhouette ranged from 0-1. If the value of $a(n)$ is significantly smaller than $b(n)$, it indicates that the data point is closer to its own cluster compared to other clusters. Thus, if the silhouette value approaches 1, that means the data point is well clustered. Conversely, when the Silhouette value is close to 0, it indicates poor clustering.

2.2.2.4 Applications of Machine Learning in Wireless Communication

ML introduces a new paradigm for the solution of wireless communication problems. For example, the auto-encoders developed to optimize the whole transmitter and receiver system for data detection [32, 33]. Also, channel estimation [34] and path loss [35] can be predicted by regression methods, while beam selection [36] and signal detection [37] can be formulated as classification tasks. Additionally, reinforcement learning can be applied to scenarios with user scheduling and resource allocation [38, 39].

2.3. Long Range LoRa

LoRa is a wireless communication technology that has gained significant attention and been adopted due to its suitability for use with the Internet of Things (IoT)

and low-power, wide-area network (WAN) applications [40]. Fig. 2.6 show the low-power (WAN) compared to other methods of connectivity in wireless communication local area networks (LANs) and cellular networks. Included are the advantages and disadvantages of each method. LoRa is designed to provide long-range communication capabilities with low energy consumption, making it ideal for applications where extended coverage and efficient power management are paramount.
The core of the LoRa technology lies in its unique modulation scheme called the chirp spread spectrum (CSS). By spreading the transmitted signal across a wide bandwidth through linear frequency modulation, LoRa achieves robustness against multipath fading and narrowband interference, ensuring reliable communication over extended distances [41].

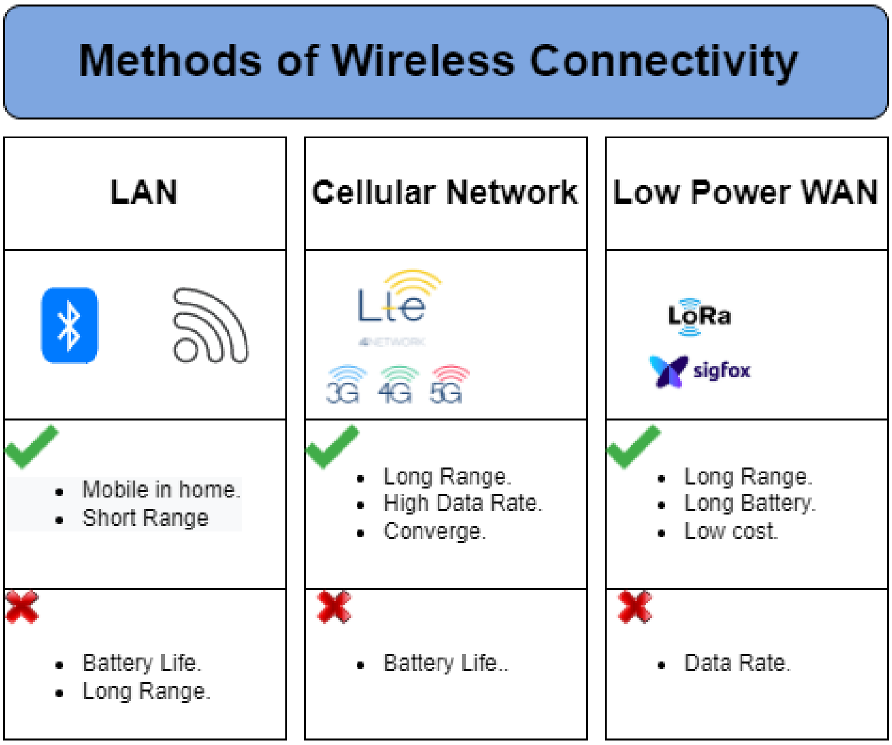


Figure 2.6: Methods of wireless connectivity

2.3.1. Chirp Spreading Spectrum Based Modulation

Chirp spread spectrum (CSS) modulation is a distinct technique employed in communication systems to achieve robust and reliable transmission over a wide range of distances. It involves spreading the transmitted signal across a broad bandwidth by modulating the carrier frequency with a linearly increasing or decreasing waveform known as a chirp [42]. This linear frequency modulation allows for efficient use of the available frequency spectrum. CSS modulation offers several advantages for communication systems. Firstly, it enhances resistance against multipath fading and narrowband interference, as the spread signal exhibits improved resilience to these impairments. Additionally, CSS modulation enables long-range communication by leveraging the wide bandwidth, ensuring extended coverage even in challenging propagation environments. The characteristics of CSS modulation make it well-suited for applications requiring low-power, wide-area connectivity, such as IoT networks. By utilizing the chirp spreading spectrum, communication systems can achieve reliable and energy-efficient transmission, making CSS modulation an attractive choice for various wireless communication scenarios.

2.3.2. LoRa Parameters

LoRa adopted a CSS modulation scheme, details of which can be seen in the datasheets for the Semtech SX1272 and SX1276 low power, long range transceivers [43, 44]. The choice of the LoRa parametric values, spreading factor, bandwidth, code rate and frequency determines the transmission range, resilience to interference and data rate. Following are brief description of the four LoRa parameters:

2.3.2.1 Spreading Factor

The spreading factor (SF) is a fundamental parameter in LoRa modulation which play a significant role in the performance and efficiency of the data transmission. The SF value is the duration of each symbol and directly affects the data rate and receiver sensitivity [45]. A higher spreading factor results in a longer symbol duration, thereby increasing the robustness against interference and enhancing the receiver's ability to capture and demodulate weak signals [46]. However, this comes at the cost of a reduced data rate due to the increased symbol duration. Conversely, a lower spreading factor provides a higher data rate but sacrifices sensitivity to weak signals. Therefore, selecting an appropriate spreading factor is critical in LoRa systems, as it involves a trade-off between data rate and range, allowing system designers to optimize communication performance according to specific application requirements.

2.3.2.2 Bandwidth

The use of chirp spread spectrum modulation in LoRa spreads the transmitted signal across a wide bandwidth, allowing for robust and reliable long-range communication. The choice of bandwidth in LoRa determines the data rate and the number of simultaneous transmissions that can coexist within a given frequency band. LoRa offers different bandwidth options, such as 125 kHz, 250 kHz, 500 kHz, and even wider variations, depending on regional regulations and specific application requirements [47].

2.3.2.3 Code Rate

LoRa utilizes forward error correction (FEC) techniques to enhance the reliability of transmitted data. The code rate determines the amount of redundancy added to the transmitted signal, impacting the error correction capability. A higher code rate, such as 4/5 or 4/6, (e.g., where there four bits of useful information are encoded in five or six transmission bits) provides stronger error correction capabilities by adding more redundant bits. This improves the system's ability to recover from transmission errors, thereby increasing the reliability of data reception. In LoRa, four coding rates are currently used 4/5, 4/6, 4/7, and 4/8 [48].

2.3.2.4 Frequency

LoRa operates within unlicensed frequency bands, which offer the advantage of not requiring any license fees for utilization. These frequency bands are commonly referred to as industrial, scientific, and medical (ISM) bands [49]. Specifically, LoRa operates in the sub-GHz range, with frequencies around 433 MHz, 868 MHz, or 915 MHz, depending on the specific regulations applying in different regions. These unlicensed frequency bands enable LoRa to achieve long-range communication capabilities and are well-suited for a variety of applications. By operating within unlicensed bands, LoRa allows for greater accessibility and flexibility in deploying wireless communication systems without the need for costly licensing procedures.

2.3.3. LoRa Physical Layer

LoRa chirp modulation is patented with no theoretical description provided [50]. Knowing the chirp modulation employ the frequency shift scheme, the best description of the LoRa modulation is Frequency Shift Chirp Modulation (FSCM) [50]. Recently, the mathematical description of the LoRa physical (PHY) layer has been extensively studied [51,52]. In LoRa, the symbols are transmitted as frequency-shifted chirps with bandwidth $\mathrm{B} = \frac{1}{\mathrm{T}}$, where T is the symbol transmission duration. The basic chirp signal at time $(n\mathrm{T})$ where the frequency increase linearly with bandwidth is described as [52]

$$x_0(n\mathrm{T}) = \sqrt{\frac{1}{2^{\mathrm{SF}}}} \exp\left[j2\pi(n\mathrm{B}/2^{\mathrm{SF}})n\mathrm{T}\right] \tag{2.11}$$

where $n = 0, 1, \ldots, 2^{\mathrm{SF}} - 1$ denoted the the sample index at T, $\mathrm{SF} \in \{7, 8, ..12\}$. Once it reaches the maximum bandwidth, the chirp signal wraps around back to zero frequency, creating a cyclic pattern as shown in Fig. 2.7. This wrapping behaviour ensures that the chirp signal remains within the specified bandwidth and does not exceed it, while utilizing the available frequency spectrum effectively. The transmitted LoRa symbol is expressed as [52]

$$x_i(n\mathrm{T}) = \exp\left[j2\pi(i+n) \mod 2^{\mathrm{SF}} \frac{n}{2^{\mathrm{SF}}}\right] \tag{2.12}$$

where $i \in \left\{0, 1, ..., 2^{\mathrm{SF}} - 1\right\}$ and mod is the Modulo operation.
The demodulation in LoRa is determined by the orthogonality of basis LoRa signals, where the cross-correlation is given as [50]

$$C_{l,i} = \sum_{n=0}^{2^{\mathrm{SF}}-1} x_i(n\mathrm{T}).x_l^*(n\mathrm{T}) = \begin{cases} 1 & l = i \\ 0 & l \neq i \end{cases} \tag{2.13}$$

where $C_{l,i}$ denotes the cross-correlation of LoRa basis signal $x_l(n\mathrm{T}).x_i(n\mathrm{T})\forall l$, and $x_l^*(n\mathrm{T})$ is the complex conjugate of the basic chirp.

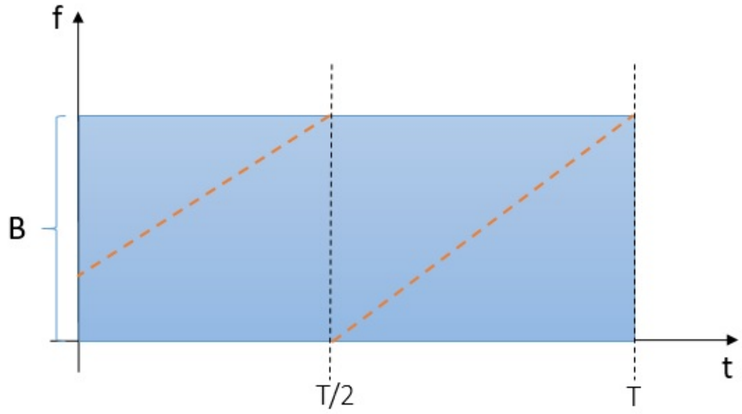


Figure 2.7: Chirp signal

2.3.4. Interference Models for LoRa

This subsection provides an overview of the various interference models applicable to LoRa technology. LoRa systems may encounter different types of interference, which can significantly impact their performance. Two primary forms of interference are discussed in this section: cross-technology interference and same-technology interference. A graphical representation of the interference types in LoRa is shown in Fig. 2.8.

2.3.4.1 Cross-Technology Interference

Cross-technology interference is observed when diverse technologies coexist within the same Industrial, Scientific, and Medical band as, for instance, with SigFox. The influence of other technologies on the performance of LoRa has been extensively investigated and reported in the literature [53, 54].

2.3.4.2 Same-Technology Interference

Same-technology interference in LoRa refers to instances where a LoRa signal experiences interference from other LoRa signals. This type of interference can be further divided into two categories based on the spreading factors employed by the interfering signals.
The first category includes interference from other signals that employing the same spreading factor as the affected LoRa signal. The second category involves interference from signals that use different spreading factors. A study by Bor et

al. [55], investigated the scalability of LoRa networks in the presence of same-spreading-factor interference. The study utilizes a simplified interference model, assuming that the transmitted packet is affected by interference packets with the same SF, and relay on receive signal strength (RSS) method at the receiver.

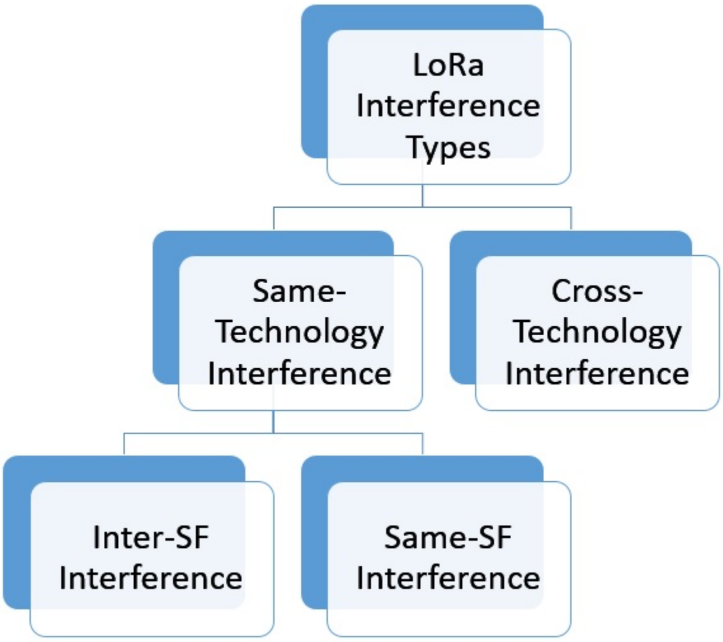


Figure 2.8: Interference types in LoRa

2.4. Chapter Summary

In conclusion, this background theory chapter has provided an overview of the fundamental concepts and principles related to SPC, in contrast to long packet communication. The key differences between these two approaches are discussed, highlighting the unique challenges and considerations associated with short packets, such as the impact of metadata size and the need for efficient overhead design. Furthermore, different types of ML algorithms are explored to provide a basic understanding of the algorithms applied in this thesis. Finally, an overview of the LoRa system is provided, covering the basics of LoRa modulation, the unique parameters, and the different types of interference that occur in LoRa.

Chapter 3

Accurate Evaluation of Packet Error Probability for Short Packet Communications

This chapter presents a new mathematical analysis of the performance of short packet communications in different wireless communication scenarios. This results in new accurate expressions for the average packet error probability in a system with K-interfering packets. The results are investigated by throughput performance.
The rest of this chapter is organized as follows: Section 3.1 introduces the chapter and states the main contribution made by this research. Related work is presented in Section 3.2. Section 3.3 describes the system model with interference expression and the distribution of SINR is provided in Section 3.4. Section 3.5 introduces the analysis of the packet error rate, and the throughput analysis is provided in Section 3.6. The results are presented in Section 3.7. Finally, Section 3.8 concludes the chapter.

3.1. Introduction

Information theory played a pivotal role in developing communication theory, including various applications such as statistics and coding. However, its impact on communication systems has been somewhat limited, as pointed out in [7]. This limitation is primarily due to the asymptotic nature of information theory.

Shannon's formulation of channel capacity necessitates infinitely large blocklength to ensure very low error probabilities. Consequently, classical information measures cannot effectively address scenarios where blocklengths are finite. The concept of finite blocklength (in this chapter, finite blocklength and short packet are used as synonyms) is linked with the emerging requirements of ultra-reliable low latency (URLL) [56]. Thus, the finite blocklength challenge is a fundamental aspect that must be tackled to fully understand its performance in practice.
The contributions of this chapter can be summarized as:

- A novel analysis is presented for evaluating average packet error probability with a random number of interfering packets for short packet transmission.
- The performance of two real scenarios with different distributions is introduced.
- The effect of the coding rate (r), packet length (L), and throughput with different numbers of users (M) have been studied.

3.2. Related Work

Several studies have been conducted to characterize the performance of communication systems with short packet transmission and provide designs that suit the new requirements of SPC. In the pioneering work by Polyanskiy et al. [11], information-theoretic limits on the achievable rate were introduced for a given block length and error rate. Since then, this work has been extended to different scenarios to further investigate the performance of SPC. The study in [57], investigated the performance of SPC in a single-hop transmission system under the influence of additive white Gaussian noise (AWGN). This research aimed to assess the efficiency of SPC in the presence of AWGN and determine the achievable rate under such conditions. Durisi et al. examined the maximum achievable rate in a multi-antenna system in a Rayleigh fading environment [58]. This study aimed to evaluate the trade-off between throughput, latency, and reliability using finite blocklength. By considering the impact of fading and employing multiple antennas, the authors aimed to improve the performance of SPC in realistic wireless communication scenarios. The authors in [59] evaluated packet error rate performance for half duplex and full duplex. This paper concluded that the full duplex performance is superior to the half duplex in the SPC environment.

Many studies have focused on the SPC performance within the context of non-orthogonal multiple access (NOMA).Cooperative NOMA was investigated by Lai et al. [60], while downlink transmission NOMA was examined by [61,62]. Additionally, the performance of SPC has been extensively studied with an intelligent reflecting surface (IRS) [63, 64]. Cognitive radio systems have also been a subject of investigation for SPC performance, as explored in [65, 66]. Furthermore, the reliability of SPC in Unmanned Aerial Vehicle (UAV) communication systems has been addressed by [67].

However, despite the extensive research conducted on SPC, there is a notable lack of studies focusing on the performance of such systems under interference scenarios. Recent research efforts have aimed to address this gap. For instance, Kumar et al. studied the performance in the presence of interference from two users [68]. Similarly, Vu et al., derived a closed-form expression for the block error rate and throughput considering co-channel interference and imperfect successive interference cancellation with two Non-Orthogonal Multiple Access (NOMA) users [69].

These studies highlight the importance of considering interference in short packet transmission and provide valuable insights into the system's performance under realistic interference conditions. It is worth noting that many theoretical studies in the past have made assumptions regarding a limited number of interferences and the availability of data for transmission by the users. In contrast, our work aims to provide a more accurate analysis that considers random interference scenarios, which will further enhance the understanding of SPC system performance in realistic settings.

3.3. System Model

Consider a short packet transmission system model in a slow Rayleigh fading environment. The received signal is affected by K interfering packets. Let a reference packet (packet 0) be subjected to other interference packets $(K-1)$ at the receiver, as shown in Fig. 3.1. The signal at the front end of the receiver can be written as

$$y_0(t) = \sqrt{2p_0}h_0x_0(t) + \sqrt{2p_k}\sum_{k=1}^{K-1} h_kx_k(t-\tau_k) + w(t) \tag{3.1}$$

where $w(t)$ is white Gaussian noise with two-sided power density N_0 and time offset τ_k. $x_k(t)$ is the kth packet $1, 2, \ldots, K-1$ represented as

$$x_k(t) = \sum_{i=0}^{\mathrm{L}-1} c_i^{(k)} \psi(t - iT) \tag{3.2}$$

p_k is the average power of the received packet (which depends on the transmitted power and path loss of the kth packet). h_k is the Rayleigh fading complex gain where the fading is assumed to be constant during the packet's transmission time. The channel gains are complex Gaussian with $\mathbb{E}[h_i * h_j] = 1$ when $i = j$ and 0 otherwise. The $c_i^{(k)}$ is the ith symbol in the packet. In the case of QPSK it equal to $c_i^{(k)} = a_i^{(k)} - jb_i^{(k)}$ where $a_i^{(k)}$ and $b_i^{(k)}$ are the binary inphase and quadrature-phase symbols with values in $\{-1, 1\}$. $\psi(t)$ is a time-limited waveform shape with interval [0,T) and normalized to $\frac{1}{T}\int_0^T |\psi(t)|^2 dt = 1$, T represent a symbol duration, and L is the packet length.

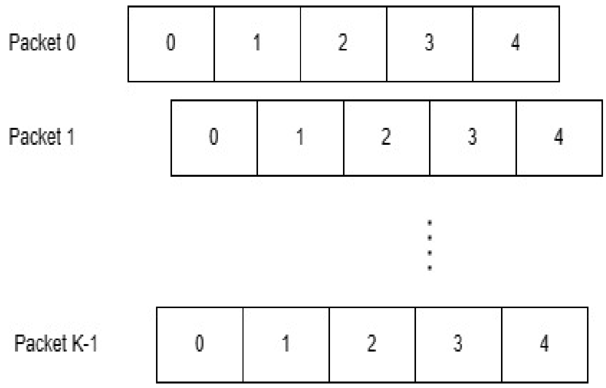


Figure 3.1: Example of interfering packets. In Packet 1, symbol 1 interferes with symbols 1 and 2 of packet 0.

Accordingly, the decision variable of the inphase ith symbol of the reference packet is

$$S_i = \mathrm{Re}\left\{ a_i^{(0)} \sqrt{2p_0}|h_0| + \sum_{k=1}^{K-1} h_k \sqrt{2p_k} I_{k,i} + \eta_i \right\} \tag{3.3}$$

where the η_i is the complex Gaussian variable with variance $\frac{N_0}{2T}$ and $I_{k,i} = \frac{1}{T}\int_{(i-1)T}^{iT} \psi^*(t) x_k(t - \tau_k) dt$, represents the interference component resulting from the kth interference signal, given as

$$I_{k,i} = \left[a_{i-1}^{(k)} R(\tau_k) + a_i^{(k)} \hat{R}(\tau_k)\right] - j\left[b_{i-1}^{(k)} R(\tau_k) + b_i^{(k)} \hat{R}(\tau_k)\right] \tag{3.4}$$

where $R(\tau)$ and $\hat{R}(\tau)$ denote the continuous partial correlation functions of $\psi(t)$ shaping waveform expressed as

$$\begin{aligned} R(\tau) &= \frac{1}{T}\int_0^T \psi(t+T-\tau)\,\psi^*(t)\,dt \\ \hat{R}(\tau) &= \frac{1}{T}\int_0^T \psi(t-\tau)\,\psi^*(t)\,dt. \end{aligned}$$

Consider the case of a rectangular pulse shape, $R(\tau) = \tau$ and $\hat{R}(\tau) = 1-\tau$. Therefore, Eq. (3.4) reduces to

$$I_{k,i} = \left[a_{i-1}^{(k)} \tau_k + a_i^{(k)}(1-\tau_k)\right] - j\left[b_{i-1}^{(k)} \tau_k + b_i^{(k)}(1-\tau_k)\right]. \tag{3.5}$$

It is important to note that when we fix the random variables $p_0, h_0, \left\{a_i^{(k)}, a_{i-1}^{(k)}, b_i^{(k)}, b_{i-1}^{(k)}, \tau_k, p_k\right\}$, the decision variable in Eq. (3.3) becomes Gaussian with a conditional mean

$$\mathbb{E}\left[S_i \mid p_0, h_0\right] = \pm\sqrt{2p_0}\,|h_0| \tag{3.6}$$

and conditional variance

$$\mathrm{Var}\left(S_i | a_i^{(k)}, a_{i-1}^{(k)}, b_i^{(k)}, b_{i-1}^{(k)}, \tau_k, p_k\right) = \sum_{k=1}^{K-1} p_k v_{k,i} + \frac{N_0}{T} \tag{3.7}$$

where $v_{k,i}$ is obtained from Eq. (3.4)

$$\begin{aligned} v_{k,i} &= \mathrm{Var}\left(I_{k,i} | a_i^{(k)}, a_{i-1}^{(k)}, b_i^{(k)}, b_{i-1}^{(k)}, \tau_k\right) \\ &= \left[a_{i-1}^{(k)} R(\tau_k) + a_i^{(k)} \hat{R}(\tau_k)\right]^2 + \left[b_{i-1}^{(k)} R(\tau_k) + b_i^{(k)} \hat{R}(\tau_k)\right]^2. \end{aligned} \tag{3.8}$$

Consequently, the average bit error probability can be derived exactly by computing the means of the Gaussian error function

$$p_b = \frac{1}{2}\mathbb{E}\left[\mathrm{erfc}\sqrt{\frac{p_0|h_0|^2}{\sum_{k=1}^{K-1} p_k v_{k,i} + \frac{N_0}{T}}}\right]. \tag{3.9}$$

Observe that the expectation is to be computed with respect to the three random variables: τ, the channel gains $h_0, h_1, \ldots, h_{K-1}$ and powers of the received packets $p_0, p_1, \ldots, p_{K-1}$ in addition to the variances $v_{k,i},$. Earlier studies focused on obtaining the distribution of $\mathrm{Var}(S)$ to solve for the average of Eq. (3.9) with infinite block length [70]. In this chapter, we compute the statics of the instantaneous signal-to-interference plus noise ratio (γ) instead of using $\mathrm{Var}(S)$ for finite block length.

$$\gamma = \frac{1}{2}\frac{\mathbb{E}^2[S_i]}{\mathrm{Var}(S_i)} = \frac{p_0|h_0|^2}{\sum_{k=1}^{K-1} p_k v_{k,i} + \frac{N_0}{T}}.$$

Accordingly, to find a closed-form expression, the problem of computing the average bit error rate is changed to evaluating the average of ($\mathrm{erfc}\sqrt{(\gamma)}$) with respect to the random variable γ. This method was introduced in [71] to find the accurate bit error probability in an infinite blocklength system.

3.4. The Distribution of the SINR

Condition on the random variables $\{\tilde{v}_k, p_k\}_{k=1}^{K-1}, K$. Recall that $|h_0|$ is a complex Gaussian, thus in Rayleigh fading, $|h_0|^2$ becomes an exponentially distributed random variable. Therefore, we obtain the conditional distribution function of the equivalent SINR (γ) at the ith symbol

$$\Pr(\gamma > z \mid \{\tilde{v}_k, p_k\}_{k=1}^{K-1}, K) = e^{-z\frac{N_0}{Tp_0}} e^{-z\frac{p_k}{p_0}\sum_{k=1}^{K-1}\tilde{v}_k} \tag{3.10}$$

Condition on K and p_0, and assume the K packets are independent

$$\Pr(\gamma > z|K, p_0) = e^{-\frac{z}{\mathrm{SNR}}} V^{K-1}(z) \tag{3.11}$$

where

$$V(z) = \mathbb{E}[e^{-zp_k v_{k,i}}] \tag{3.12}$$

Note that $v_{k,i}$ depends on independent random variables $a_i^{(k)}, a_{i-1}^{(k)}, b_i^{(k)}, b_{i-1}^{(k)}$ that take values of $\{1, -1\}$, and τ_k which is uniform in (0,T). To find $V(z)$ we condition on τ_k and p_K. Using in case of rectangular pulse shapes (3.8) we obtain

$$\begin{aligned} V(z|\tau_k, p_k) &= \mathbb{E}[e^{-zp_k v_k}|\tau_k, p_k] \\ &= \left[\frac{1}{2}e^{-zp_k} + \frac{1}{2}e^{-zp_k(1-2\tau_k)^2}\right]^2. \end{aligned} \tag{3.13}$$

When we average out τ_k which is uniform (0,T], we obtain

$$V(z|p_k) = \frac{1}{4}e^{-2zp_k} + \frac{1}{4}\sqrt{\frac{\pi}{zp_k}}e^{-zp_k}\mathrm{erfc}\left(\sqrt{zp_k}\right) + \frac{1}{16}\sqrt{\frac{2\pi}{zp_k}}\mathrm{erfc}\left(\sqrt{2zp_k}\right). \tag{3.14}$$

Finally, we have the complementary cumulative distribution function (CCDF) for the SINR. In the next section, we will find the average packet error probability.

3.5. Packet Error Rate Analysis

This section presents an exact analysis of the packet error probability in the context of short-packet transmission. The Shannon theorem was basically established for infinite packets, which is not valid with the assumption of short packets. Polyanskiy et al., in [11] introduced a unified approach to obtain tight bounds on the maximal coding rate for SPC

$$\mathrm{r}^*(\mathrm{L}, \varepsilon, \gamma) = C(\gamma) - \sqrt{\frac{V_d(\gamma)}{\mathrm{L}}}Q^{-1}(p_e) + O(\frac{\log \mathrm{L}}{\mathrm{L}}) \tag{3.15}$$

where L is packet length, $C(\gamma)$ is the capacity and $V_d(\gamma)$ is the channel dispersion [11]

$$\begin{aligned} C(\gamma) &= \log_2(1+\gamma) \\ V_d(\gamma) &= \frac{\gamma(\gamma+2)}{2(\gamma+1)^2}\log_2^2 e. \end{aligned}$$

From Eq. (3.15), the conditional packet error probability can be expressed as

$$p_e(\gamma) = Q\left(\frac{\mathrm{L}\log_2(1+\gamma) + \frac{1}{2}\log_2 \mathrm{L} - \mathrm{rL}}{\sqrt{\mathrm{L}}\log_2 e\sqrt{1 - \frac{1}{(1+\gamma)^2}}}\right) \tag{3.16}$$

which can be simplified to

$$\begin{aligned} p_e(\gamma) &= Q\left((1+\gamma)\frac{\mathrm{L}\log_2(1+\gamma)+\frac{1}{2}\log_2\mathrm{L}-\mathrm{rL}}{(\log_2 e)\sqrt{\mathrm{L}}\sqrt{(1+\gamma)^2-1}}\right) \\ &= Q\left(\sqrt{\mathrm{L}}\frac{(1+\gamma)}{\sqrt{\gamma^2+2\gamma}}\left(\frac{\log_2(1+\gamma)}{\log_2 e}+\frac{1}{2}\frac{\log_2\mathrm{L}}{\mathrm{L}\log_2 e}-\frac{\mathrm{r}}{\log_2 e}\right)\right) \end{aligned} \tag{3.17}$$

where $\frac{\log_2(1+\gamma)}{\log_2 e} = \ln(1+\gamma)$. By adding the term $\frac{\ln \mathrm{L}}{\mathrm{L}}$, the packet error probability is formulated as

$$p_e(\gamma) = Q\left(\sqrt{\mathrm{L}}\frac{(1+\gamma)}{\sqrt{\gamma^2+2\gamma}}\left(\ln(1+\gamma)+\frac{\ln\mathrm{L}}{2\mathrm{L}}-\mathrm{r}\ln 2\right)\right). \tag{3.18}$$

Using the relation of $Q(z) = \frac{1}{2}\operatorname{erfc}\left(\frac{z}{\sqrt{2}}\right)$ we obtain

$$p_e(\gamma) = \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{\mathrm{L}}{2}}\frac{(1+\gamma)}{\sqrt{\gamma^2+2\gamma}}\left(\ln(1+\gamma)+\frac{1}{2}\frac{\ln\mathrm{L}}{\mathrm{L}}-\mathrm{r}\ln 2\right)\right). \tag{3.19}$$

Therefore, the packet success probability can be obtained from

$$P_s(\gamma) = 1-\frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{\mathrm{L}}{2}}\frac{(1+\gamma)}{\sqrt{\gamma^2+2\gamma}}\left(\ln(1+\gamma)+\frac{1}{2}\frac{\ln\mathrm{L}}{\mathrm{L}}-\mathrm{r}\ln 2\right)\right). \tag{3.20}$$

Taking the average of Eq. (3.20) to find the average packet success probability with respect to the coding rate and the interfering packets

$$\begin{aligned} P_s &= 1-\frac{1}{2}\mathbb{E}\left[\operatorname{erfc}\left(\sqrt{\frac{\mathrm{L}}{2}}\frac{\gamma+1}{\sqrt{\gamma^2+2\gamma}}\left(\ln(\gamma+1)+\frac{1}{2}\frac{\ln\mathrm{L}}{\mathrm{L}}-\mathrm{r}\ln 2\right)\right)\right] \\ &= 1-\frac{1}{2}\int_0^\infty \operatorname{erfc}\left(\sqrt{\frac{\mathrm{L}}{2}}\frac{z+1}{\sqrt{z^2+2z}}\left(\ln(z+1)+\frac{1}{2}\frac{\ln\mathrm{L}}{\mathrm{L}}\mathrm{r}\ln 2\right)\right) f_\gamma(z)\,dz. \end{aligned} \tag{3.21}$$

The expectation in Eq. (3.21) is taken with respect to the SINR, $f_\gamma(z)$. The pdf of $f(\gamma)$ is unknown and cannot be found in closed form. However, in the special case of Rayleigh fading, the expression for the CCDF $\Pr(\gamma > z)$ has already been derived in Section 3.14. Using integration by parts, the expression can be rewritten as

$$P_s = 1 - \frac{1}{2}\left(2 + \int_0^\infty \frac{\partial}{\partial z}\operatorname{erfc}\left(\sqrt{\frac{\mathrm{L}}{2}}\frac{z+1}{\sqrt{z^2+2z}}\left(\ln(z+1) + \frac{1}{2}\frac{\ln \mathrm{L}}{\mathrm{L}} - \mathrm{r}\ln 2\right)\right)\right)$$
$$\Pr\left(\gamma > z\right)dz \quad (3.22)$$

where $\Pr(\gamma > z)$ is given in Eq. (3.11), and we use the fact that

$$\lim_{z\to 0}\operatorname{erfc}\left(\sqrt{\mathrm{L}}\frac{z+1}{\sqrt{z^2+2z}}\left(\ln(z+1) - \mathrm{r}\ln 2\right)\right) = \lim_{x\to-\infty}\operatorname{erfc}\left(x\right) = 2. \tag{3.23}$$

The derivative of the erfc function results in the following

$$\frac{\partial}{\partial z}\operatorname{erfc}\left(\frac{\sqrt{\mathrm{L}}(z+1)}{\sqrt{z^2+2z}\log_2 e}\left(\log_2(z+1) - \mathrm{r}\right)\right) = -2\frac{\sqrt{\mathrm{L}}}{\sqrt{2\pi}}\mathcal{A}\left(z,\mathrm{r}\right)\exp\left(-\mathrm{L}\mathcal{D}\left(z,\mathrm{r}\right)\right). \tag{3.24}$$

The notation $\mathcal{D}\left(z,\mathrm{r}\right)$ and $\mathcal{A}\left(z,\mathrm{r}\right)$ is used for the simplicity of the expression, with

$$\mathcal{D}\left(z,\mathrm{r}\right) = \frac{(z+1)^2}{2(z^2+2z)}\left(\ln(z+1) + \frac{\ln \mathrm{L}}{2\mathrm{L}} - \mathrm{r}\ln 2\right)^2$$

and

$$\mathcal{A}\left(z,\mathrm{r}\right) = \frac{1}{\sqrt{z^2+2z}}\left(1 - \frac{\ln\left(z+1\right) + \frac{\ln \mathrm{L}}{2\mathrm{L}} - \mathrm{r}\ln 2}{z^2+2z}\right).$$

Thus Eq. (3.22) reduces to

$$\begin{aligned} P_s &= \sqrt{\frac{\mathrm{L}}{2\pi}}\int_0^\infty \mathcal{A}\left(z,\mathrm{r}\right)e^{-\mathrm{L}\mathcal{D}(z,\mathrm{r})}\Pr\left(\gamma > z\right)dz \\ &= \sqrt{\frac{\mathrm{L}}{2\pi}}\int_0^\infty \mathcal{A}\left(z,\mathrm{r}\right)e^{-\mathrm{L}\mathcal{D}(z,\mathrm{r}) - \frac{z}{\mathrm{SNR}}}V^{K-1}\left(z\right)dz. \end{aligned} \tag{3.25}$$

Eq. (3.25) is a new expression for the average error probability in SPC. It is used in the next section for the system throughput.

3.6. Throughput

This section will analyze the throughput of two real wireless communication scenarios considering binomial and Poisson distributions.

3.6.1. Binomial Distribution

In practice, users are not transmitting all the time. Thus the transmission occurs only for the active user. Let's consider a random number of active users K distributed uniformly, where K is represented as a binomial random variable with probability [72]

$$\Pr(K = i) = \binom{M}{i} \rho^i (1 - \rho)^{M-i} \tag{3.26}$$

where M is the total number of users, and ρ is the probability that an arbitrary user is active. The three parameters r, M, K can be tuned to analyze the performance. The throughput

$$S(\mathrm{r}, \rho, M) = r\mathbb{E}\left[K P_s(\mathrm{r}|K)\right]. \tag{3.27}$$

It is known that $\mathbb{E}[K] = \rho M$, and the probability generating function (PGF) of the binomial random variable is given as [73]

$$\mathbb{E}\left[V^K\right] = (1 - \rho + \rho V)^M . \tag{3.28}$$

Then,

$$\begin{aligned} \mathbb{E}\left[K V^{K-1}\right] &= \frac{\partial}{\partial x}\mathbb{E}\left[V^K\right] \\ &= \rho\left[1 - \rho + \rho V\right]^{M-1}. \end{aligned} \tag{3.29}$$

Accordingly,

$$\mathbb{E}\left[K V^{K-1}(z)\right] = \rho\left[1 - \rho + \rho V(z)\right]^{M-1}. \tag{3.30}$$

Substituting Eq. (3.30) into Eq. (3.27) gives the throughput as

$$S(\mathrm{r}, \rho, M) = \mathrm{r}\rho M \sqrt{\frac{\mathrm{L}}{2\pi}} \int_0^\infty \mathcal{A}(z, \mathrm{r})\, e^{-\mathrm{L}\mathcal{D}(z,\mathrm{r}) - \frac{z}{\mathrm{SNR}}} \left[1 - \rho + \rho V(z)\right]^{M-1} dz. \tag{3.31}$$

Eq. (3.31) is the throughput expression when K is binomial.

3.6.2. Poisson Distribution

In a realistic wireless communication system, specifically in massive machine type communication (mMTC) scenario, it is reasonable to consider a large number of users with random interference. Assuming a Poisson distribution where the number of users tends to infinite with K considered as a Poisson random variable. In the limit with an infinite number of users $M \to \infty$ and $\rho \to 0$, then in the limit the success rate would be $M\rho \to \lambda$, which is a parameter of Poisson distribution. The PGF of the Poisson random variable is defined as [73]

$$\mathbb{E}\left[V^K\right] = \sum_{K=0}^{\infty} V^K \frac{\lambda^K}{K!} e^{-\lambda} = e^{-\lambda(1-V)}. \tag{3.32}$$

Accordingly, Eq. (3.11) can be written as

$$\Pr(\gamma > z) = e^{-\frac{z}{\mathrm{SNR}}} e^{-\lambda(1-V(z))}. \tag{3.33}$$

The packet success probability, as in Eq. (3.25), when expressed as Poisson distribution, is

$$\begin{aligned} P_s &= \sqrt{\frac{\mathrm{L}}{2\pi}} \int_0^{\infty} \mathcal{A}\left(z, \mathrm{r}\right) e^{-\mathrm{L}\mathcal{D}(z,\mathrm{r})} \Pr\left(\gamma > z\right) dz \\ &= \sqrt{\frac{\mathrm{L}}{2\pi}} \int_0^{\infty} \mathcal{A}\left(z, \mathrm{r}\right) e^{-\mathrm{L}\mathcal{D}(z,\mathrm{r}) - \frac{z}{\mathrm{SNR}}} e^{-\lambda(1-V(z))} \left(z\right) dz. \end{aligned} \tag{3.34}$$

Then, the throughput is given as

$$S = \mathrm{r}\lambda P_s. \tag{3.35}$$

Eq. (3.35) is the throughput expression when K is Poisson.

3.7. Results and Discussion

In this section, analytical results of packet success probability Eq. (3.25), binomial throughput Eq. (3.31) and Poisson throughput Eq. (3.35) are presented. Let's consider an equal power level where $p_0 = p_1 = \ldots = p_{K-1} = p$. First, we validate the packet success probability by comparing the analytical with a Monte-Carlo simulation. As shown in table 3.7, the analytical results closely match the simulation results.

Figure 3.2 illustrates the relationship between the packet success probability and different coding rate values ($\mathrm{r} = 0.2, 0.5, 0.8$); note r ranges from 0 to 1. The graph indicates that a lower coding rate leads to a higher packet success probability, confirming our results as expressed in Eq. (3.25). This finding suggests that reducing the transmission rate improves the likelihood of successful packet transmission in an interference environment. Additionally, the results demonstrate the negative impact of an increasing number of interferers on the probability packet success.

In our analysis of the performance of short packets, examining the influence of packet length on packet delivery in the presence of interference is crucial. Fig. 3.3 and Fig. 3.4 provide insights into this effect, focusing on the relationship between packet length and packet success probability. Remarkably, both figures reveal consistent findings, wherein the packet success probability is better when the packet length is small L=50 and when L increases, the packet success probability remains unchanged.

For binomial distribution we first examine the performance of the throughput as a function of M and ρ. In Fig. 3.5, the throughput is plotted against the number of users for different coding rate values. The results demonstrate a distinct pattern with a high coding rate, such as 0.8; the throughput experiences a sharp increase initially with a low number of users. However, as the number of users grows, the throughput declines significantly. Conversely, a low coding rate of 0.2 yields a lower initial throughput, but it gradually improves as the number of users increases, peaking between 15 to 20 users. Fig. 3.6, presents the impact of the packet length on the system throughput. Notably, the results indicate that a lower packet length yields better throughput performance, thereby validating the effectiveness of our mathematical analysis in the context of SPC. Additionally, Fig 3.7 show the relationship between the throughput and the probability of active users for r $= 0.2, 0.3, 0.5$. The plots all starts with a high value, rise to a maximum and decrease dramatically. This indicates that when $\rho = 1$, the number of interference is high, so the throughput is decreased. Similarly, Fig. 3.8 illustrates the performance of the throughput with three values of L, 50, 100, 150. This exhibits a similar pattern with different packet lengths. However, the performance of the smallest length L=50 is superior to the larger length L=100 and L=150.

Next, we investigate the performance of the throughput against the active users λ in the case of Poisson distribution. Fig. 3.9, depict the performance of the throughput with different r. The results illustrate a similar pattern where the throughput initially shows a steep increase until it reaches a peak at $\lambda = 3$, $\lambda = 5$, and $\lambda = 9$, for corresponding values of r $= 0.5$, r $= 0.3$, and r $= 0.2$ respectively.

Subsequently, the throughput experiences a dramatic decline. This behaviour is attributed to the relationship between the parameter λ and the interference. As λ increases, which indicates that the number of active users increases, the number of interference increases, adversely impacting the throughput. When the value of λ is small, indicating a low number of active users, means there is low interference, with no significant need for extensive coding. However, higher coding rate is needed as the number of active users increases. Similarly, Fig. 3.10 illustrates the performance of the throughput for three values of L. The plot shows a similar pattern to that in Fig. 3.9, with different packet lengths. It is seen that the performance of the smallest length L=50 outperforms the larger lengths.

Table 3.1: Packet success probability comparison with different K values between analytical and simulation with L=50, SNR=50 dB and r=0.3.

K	***Analytical***	***Simulation***
1	0.97	0.98
2	0.76	0.74
3	0.59	0.52
4	0.47	0.46
5	0.38	0.42

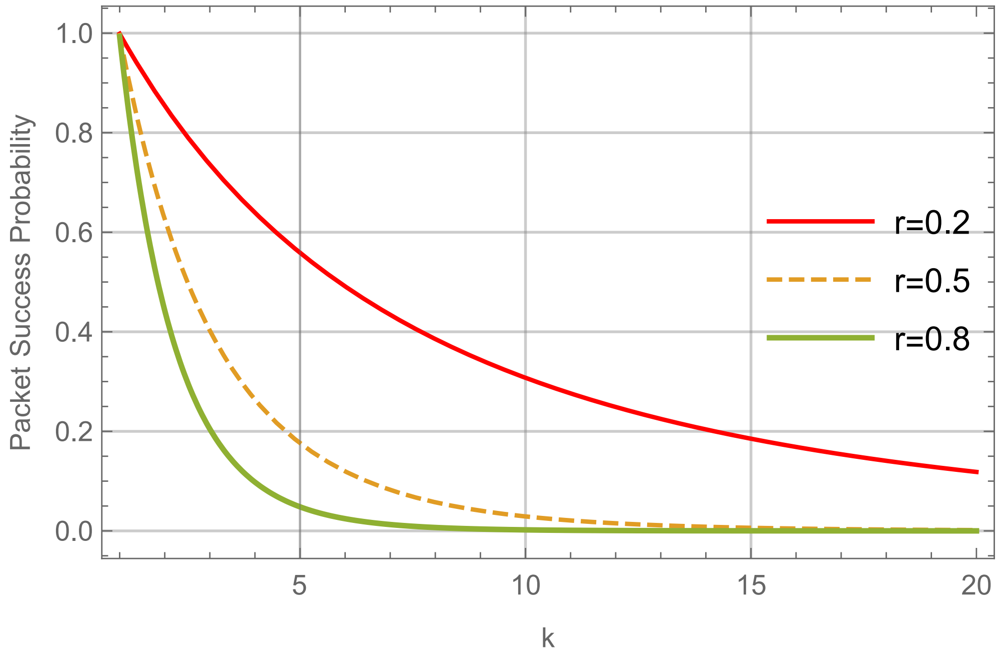


Figure 3.2: Packet success probability versus the number of interfering packets with different coding rates r, L=50, and SNR=100 dB

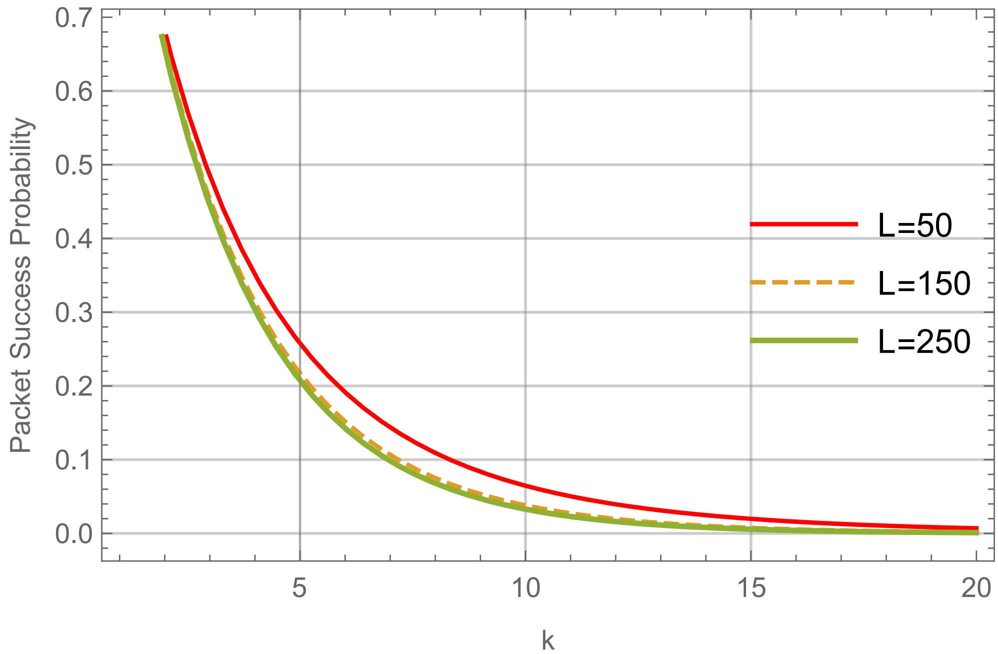


Figure 3.3: Packet success probability versus the number of interfering packets for three values of L, with r=0.5 and SNR=100 dB.

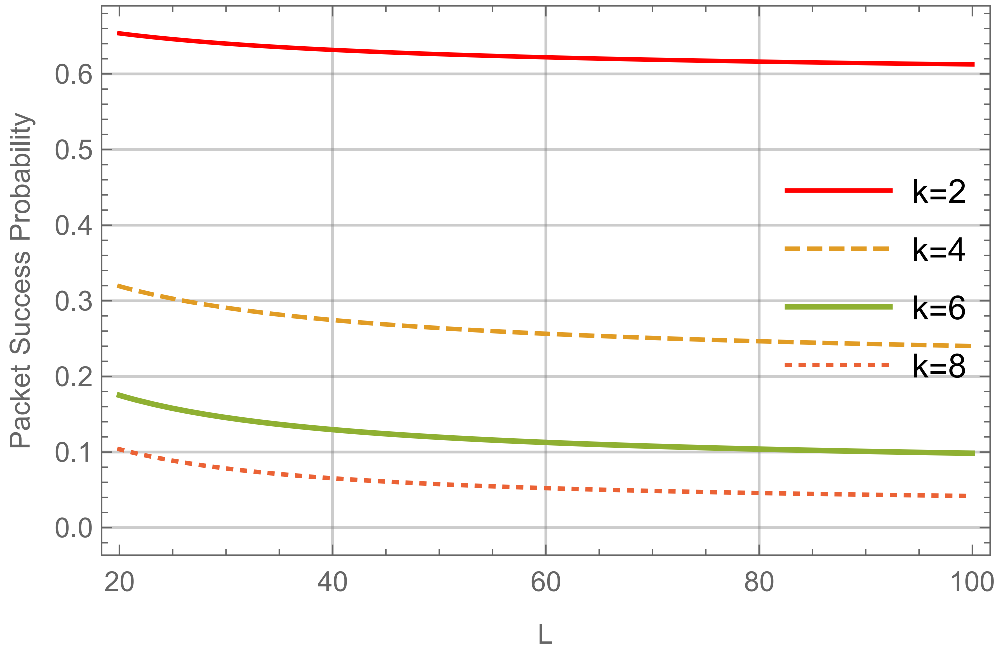


Figure 3.4: Packet success probability versus the packet length with different interfering packet values, r=0.5 and SNR=100 dB.

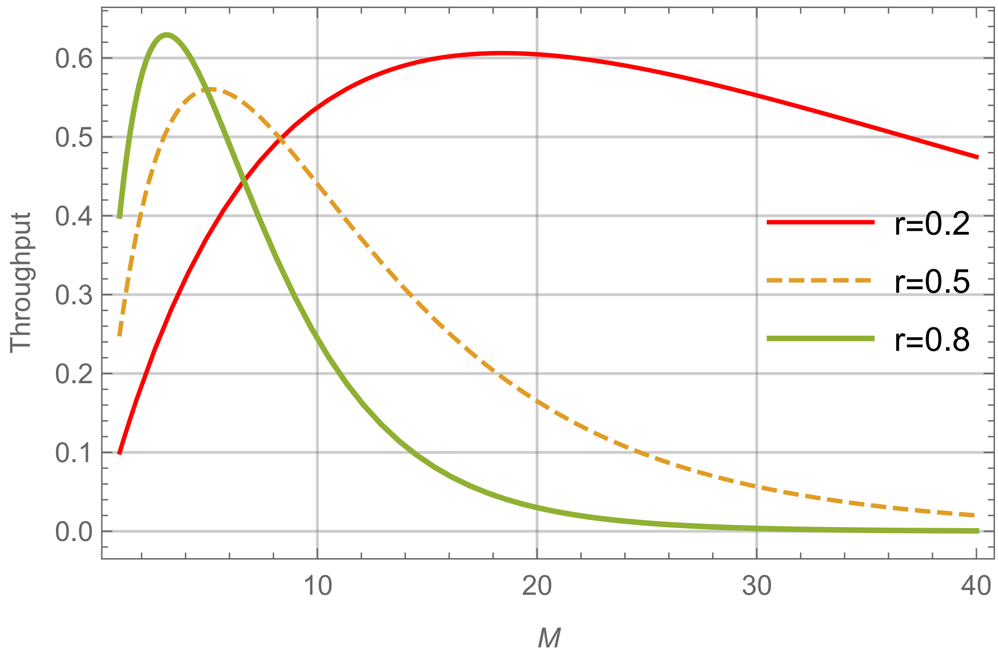


Figure 3.5: Throughput versus the number of users with different r values, L=50, $\rho = 0.5$, and SNR=100 dB in the case of the binomial distribution.

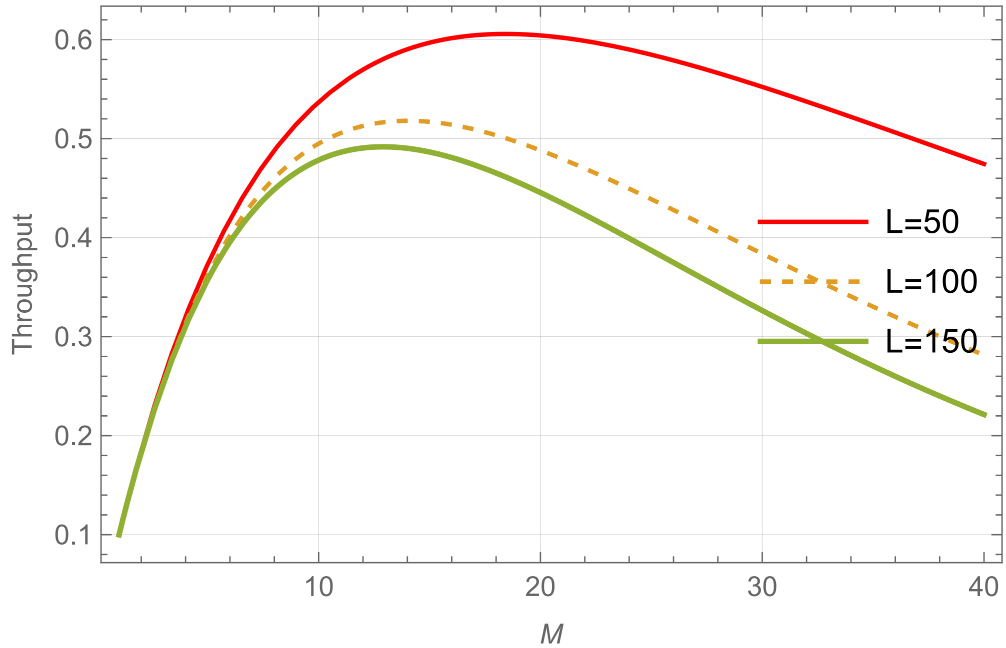


Figure 3.6: Throughput versus the number of users with different L values, r=0.2, $\rho = 0.5$ and SNR=100 dB in the case of the binomial distribution.

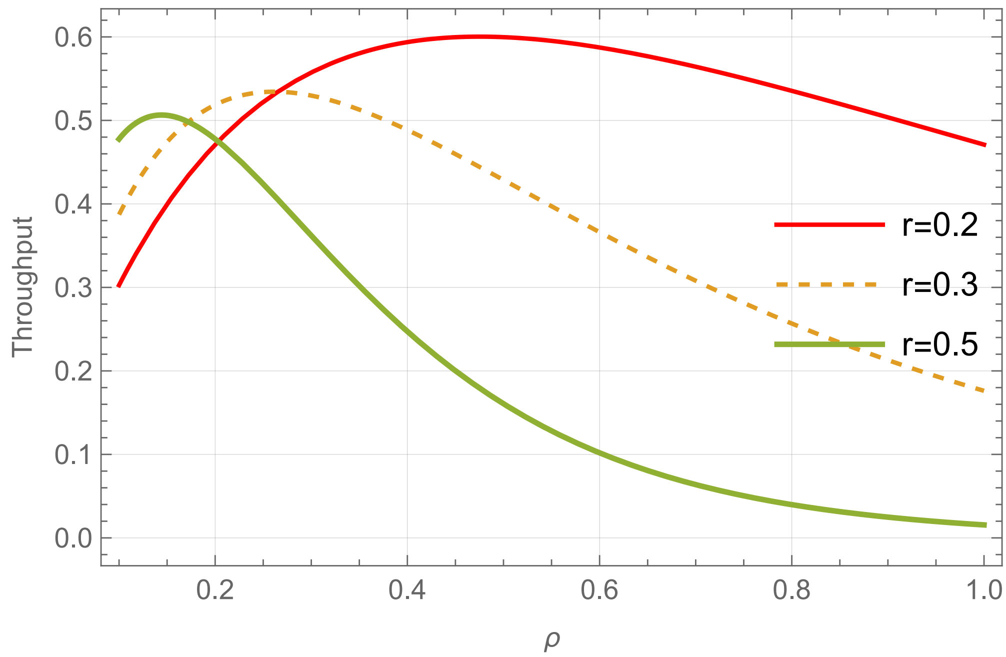


Figure 3.7: Throughput versus ρ with different r values, L=50, M=20 and SNR=100 dB in the case of the binomial distribution.

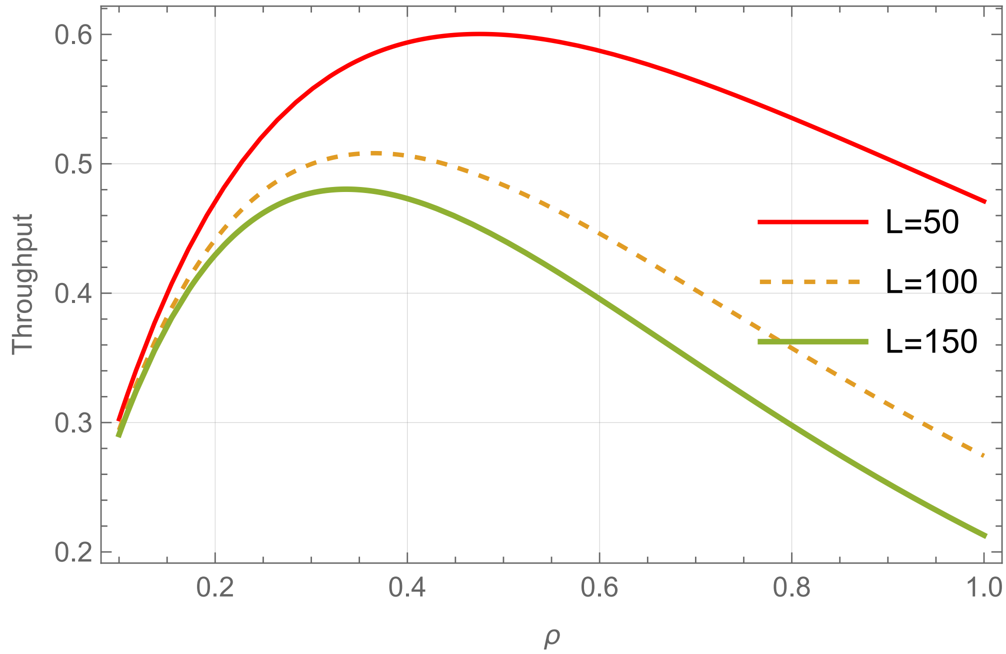


Figure 3.8: Throughput versus ρ with different L values, r=0.2, $M = 20$ and SNR=100 dB in the case of the binomial distribution.

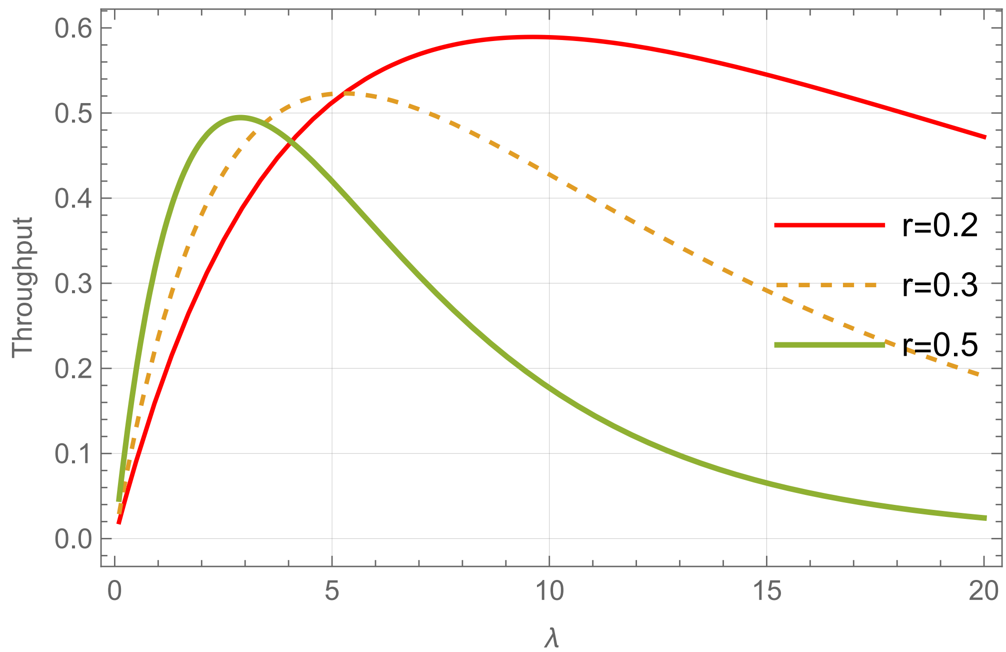


Figure 3.9: Throughput versus λ with different r values, L=50 and SNR=100 dB in the case of the Poisson distribution.

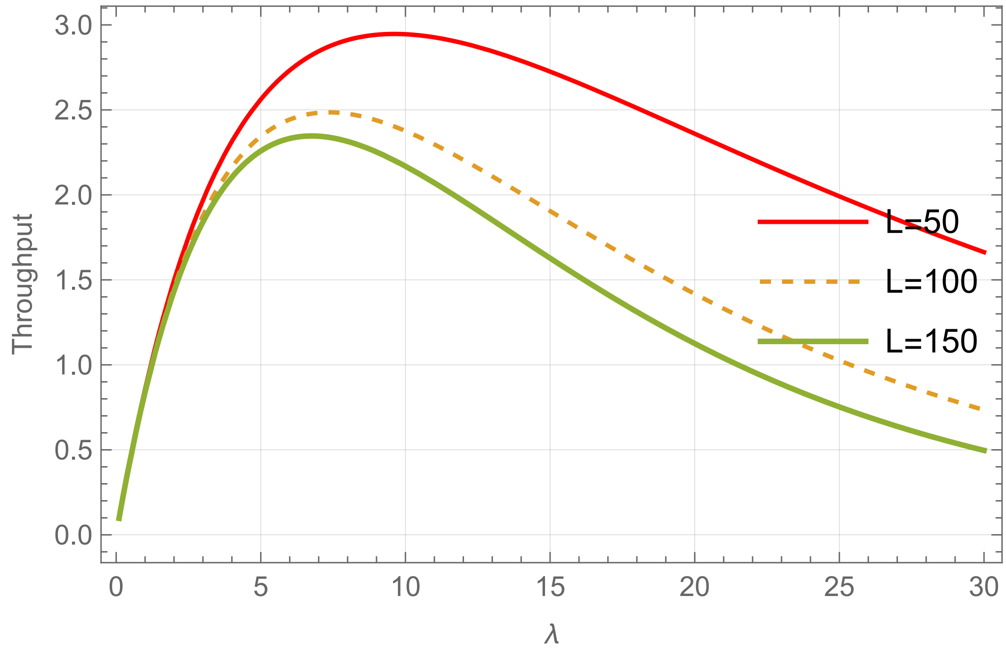


Figure 3.10: Throughput versus λ with different L values, r=0.2 and SNR=100 dB in the case of the Poisson distribution.

3.8. Chapter Summary

This chapter introduces a new analysis technique to accurately evaluate packet error probability in interference environments within SPC systems. By considering finite blocklength assumptions and utilizing an equivalent signal-to-interference plus noise ratio, we derive an exact expression for packet error probability. This approach considers the specific characteristics of SPC systems, such as packet length. It offers a more precise understanding of their performance in the presence of interference. The accuracy of the expression is validated by Monte Carlo simulation. Our results show the impact of different short packet parameters, such as packet length and coding rate, on the packet success probability and system throughput. In the subsequent chapter, the performance of SPC with various sparse recovery algorithms will be investigated.

Chapter 4

Sparse Recovery Algorithms for Short Packet Communications

In this chapter, four recovery algorithms are introduced for sparse vector coding. The aim is to apply different sparse recovery algorithms and study the relative performances of algorithms comprehensively.
The chapter is organized as follows. Section 4.1 describes how the chapter contributes to the present work. Section 4.2 provides the literature review. Section 4.3 introduces compressive sensing in SPCs. Section 4.4 presents the system model. Section 4.5 describes the sparse recovery algorithms. The performance analysis is introduced in Section 4.6. The results are presented in Section 4.7. Finally, Section 4.8 summarizes the chapter.

4.1. Introduction

The fifth and subsequent generations of mobile technology will support a massive number, in the many billions, of connected devices and will play an important role in many key applications of the Internet of Things (IoT) such as the Internet of Health (IoH) and Internet of Vehicles (IoV) [74]. In such applications, the transmitted information will mainly be control instructions and sensor information and be small in size; thus, effective and efficient short packet communication is considered a key requirement. Recently, a new scheme that successfully supports SPC, namely sparse vector coding (SVC) has been developed [75]. The principle

of SVC is to transmit the information in sparse vector form. This step is achieved by converting the information to a sparse vector and sending the sparse vector by random spreading. Then, the SVC formulates the decoding process for the sparse recovery problem using position indices [76]. In the context of SPC, the existing work concentrates on developing sparse coding techniques with different system models but not on the performance of the recovery algorithms. The decoding for SVC schemes is a process of finding the positions of the non-zeros in the transmitted sparse vector by compressed sensing. One potential problem is that the incorrect selection of the non-zero index leads to the failure of packet decoding. Also, in SPC, the packet delivery time is important according to URLL standards [77]. This chapter introduces four different compression-sensing algorithms, also known as sparse recovery algorithms. The four algorithms are comprehensively investigated using different parameters.
The contributions of this chapter are:

1. It introduces two new compressed sensing algorithms: Compressive Sampling Matching Pursuit (CoSaMP) and Stagewise Matching Pursuit algorithms (St-OMP) to SVC for use with SPC. Then, we compare their performances with simplified Multiple Matching Pursuit (MMP) and Orthogonal Matching Pursuit (OMP) algorithms and present a summary of the results (Table 4.1).

2. It shows the simulated recovery time, covariance, block error rate, and recovery error for MMP, OMP, St-MP, and CoSaMP.

3. It presents an analysis of the Symbol Error Rate (SER) by pairwise error probability.

4.2. Related Work

Compressive sensing (CS) has received significant attention over the past few years in many fields, such as radar imaging [78], image processing [79], electromagnetics [80] and, wireless communication [81]. Compressed sensing provides a new paradigm for sparse signals. Sparsity is a low-dimension framework of the signal that contains only a few elements of a coefficient vector [82]. SVC arose from compressed sensing and has been studied in different system scenarios, where it has shown a lower block error rate than conventional channel coding [75, 76, 83].

In [76], the authors showed the SVC framework to be suitable for wireless communications. In this paper, the main guidelines and some tips are provided for applying compressed sensing to wireless communications and the possible methods of its application. Ji et al. [75] proposed applying SVC to short packet transmission for ultra-reliable and low latency communication (URLLC) systems. This work was extended to enhance the performance of SVC by applying M-ary Quadrature Amplitude Modulation (M-QAM, such as 16-QAM), which introduced a higher degree of freedom [84], where the non-zero elements are produced from the same constellation alphabets. Later, Zhang et al., [83] provided a superimposed transmission scheme to further improve the transmission of SVC. The challenge of the pilot overhead with short packets in SVC systems has been widely recognized, and in [85], the possibility of a pilot-less SVC was suggested for short packet transmission.

On the other hand, the decoding of SVC is done by finding the position of non-zero elements using sparse recovery algorithms. However, most of the studies related to SVC in SPC use MMP greedy algorithm support vector recovery (the non-zero elements). The MMP selects multiple indices in a tree search for the best indices which form the support vector [86].

4.3. Compressive Sensing in SPC

The basic concept of compressed sensing is to restore the original signal using a small number of measurements after having converted the signal to a sparse signal. Compressed sensing theory requires fewer measurements than the Nyquist/Shannon sampling theorem and involves three main steps: sparse representation, measurement matrix (encoding), and sparse recovery (decoding) [75]. The driving force behind sparse vector coding was to enable short packet transmission SVC. The authors introduced a new scheme suitable for SPC, which was relayed on a sparse vector. The first step was to map the information into a sparse vector. The second step was to take a small number of measurements from the sparse vector. The third and final step was to recover a sparse vector via the small number of measurements and a randomly generated matrix (the sensing matrix) as discussed in [83].

The sparsity of the signal is described by the number of non-zero elements (Z). For example, the sparsity of matrix $\mathbf{s}$, where $\mathbf{s} = \begin{bmatrix} 0200400 \end{bmatrix}$ is Z=2 and the position of the non-zero elements, called support elements $\Omega = \{s_2, s_5\} = \{2, 5\}$. Then, the sparse signal can be expressed as

$$y = s_2 a_2 + s_5 a_5 \tag{4.1}$$

where a is a column in the sensing matrix $\mathbf{A} = \mathbf{HC}$, where $\mathbf{C}$ is a randomly generated matrix.
Generally, the received signal is given by [76]

$$\mathbf{y} = \mathbf{HCs} + \mathbf{w}. \tag{4.2}$$

Thus, if sparsity, Z, is two, and the length of the sparse vector is N, then $\binom{N}{\mathrm{Z}}$ columns should be checked by the sparse recovery algorithm to determine the correct sparse signal. The greedy algorithm is one form of a sparse recovery algorithm. It is an iterative algorithm that seeks a local optimum in each iteration to obtain the global optimal selection on completion of the algorithmic process.

4.4. Sparse Vector Coding System Model

We consider the sparse vector coding system model as a modulated transmitted vector, $s \in \mathbb{C}^{N \times 1}$, with added white Gaussian noise (AWGN) and a received signal

$$\mathbf{y} = \mathbf{HFs} + \mathbf{w}, \tag{4.3}$$

where $\mathbf{y} \in \mathbb{C}^{m \times 1}$ and $\mathbf{H} \in \mathbb{C}^{m \times m}$ is the diagonal matrix with the elements representing each resource h_{ii}, $\mathbf{F} \in \mathbb{C}^{m \times N}$ is the symbol and resource mapping matrix and $\mathbf{w} \sim (0, \sigma_n^2 I)$ is additive white Gaussian noise.
In the SVC system, after mapping the information to a sparse vector, we transmit a short packet as shown in Fig. 4.1. The encoded information bits depend on choosing the positions of the Z sparse and N symbols, so the number of encoded bits is equal to $\left\lfloor \log_2 \binom{N}{\mathrm{Z}} \right\rfloor$. Then, using a codeword contained in the spreading codebook the non-zero elements are spread into m resources. Thus, the matrix $\mathbf{F}$ is replaced by the codebook matrix $\mathbf{C}$ as in Eq. (4.2). The received signal is given

as [75]

$$
\begin{aligned}
\mathbf{y} &= \mathbf{Hx} + \mathbf{w} \\
&= [\mathbf{Hc}_2 \quad \mathbf{Hc}_5] \begin{bmatrix} s_2 \\ s_5 \end{bmatrix} + \mathbf{w}
\end{aligned}
\tag{4.4}
$$

where $\mathbf{x} = \mathbf{Cs}$.

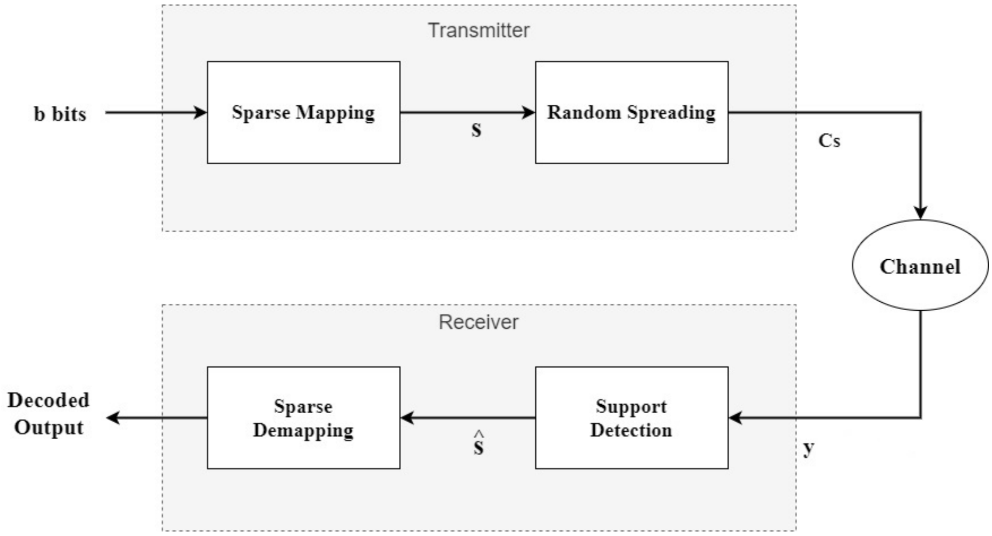


Figure 4.1: SVC system block diagram.

4.5. Sparse Recovery Algorithms

In this work, we focus on the performance of the recovery algorithms. The first basic algorithm is OMP, which uses only one path to find the recovered signal. Thus, if an incorrect index is chosen at the start of the iteration, this mistake will continue until the end, and the algorithm's output will be incorrect. To overcome this drawback, the CoSaMP algorithm selects more than one index in each iteration, and the MMP algorithm uses different paths in a tree-based strategy to find the optimum solution.

4.5.1. Orthogonal Matching Pursuit (OMP)

The main concept of the OMP is to find one support in each iteration. The process starts by checking the correlation between the residual ($r = \mathbf{y} - \mathbf{A}s$) and the matrix $\mathbf{A}$ to find the largest correlation. Specifically, the first step in OMP is solving the optimization problem by finding the index $I^t = \arg\max |\langle r^{t-1}, a_i \rangle|$. Then, merge the selected column with the index matrix. After that, the new signal is estimated by solving the least squares problem, as shown in Algorithm 1. Let us define $\mathbf{A}$ to be the sensing matrix, $\mathbf{y}$ as the observed measurement, S^t is the support set at t-th iteration and Z be the sparsity of the vector.

Algorithm 1 Orthogonal Matching Pursuit

Input: $\mathbf{A}, \mathbf{y}, \mathrm{Z}$
Initialize:
$S^0 \leftarrow \phi$
$t \leftarrow 0$
$r^0 \leftarrow \mathbf{y}$
while $t < \mathrm{Z}$ **do**
    $t = t + 1$
    $I^t = \arg\max |\langle r^{t-1}, a_i \rangle|$ ▷ Identify
    $S^t = S^{t-1} \bigcup \{I^t\}$ ▷ Merge
    $\hat{s}^t = \underset{\tilde{s}:supp(\tilde{s}) \subseteq S^t}{\arg\min} \|\mathbf{y} - \mathbf{A}\tilde{s}\|_2$ ▷ Estimate: least square
    $r^t = \mathbf{y} - \mathbf{A}\hat{s}^t$ ▷ Update
end

4.5.2. Compressive Sampling Matching Pursuit (CoSaMP)

Compressive Sampling Matching Pursuit is an improvement of the OMP algorithm while using a similar overall strategy. CoSaMP works by determining the correlations of the residual vector with the columns of the measurement matrix and solving the least squares problem for the selected columns. Algorithm 2 introduces the general definition of the CoSaMP algorithm [87]. The first step of the algorithm is to correlate the residual r with the columns of the matrix $\mathbf{A}$. Then it selects the best 2Z columns (where Z denotes the vector's sparsity) from the measured matrix $\mathbf{A}$, which is considered the highest correlated value $\mathbf{A}r$, where l_{Z} is the hard threshold. The selected columns are expressed as $\mathbf{A}r$ in the algorithm. All elements are set to zero except the largest 2Z elements, which are added to the support of the estimation of the unknown vector. The most crucial step in CoSaMP is solving the least squares problem to obtain the 3Z sparse estimate, when all other elements are set to zero. This step ensures that the estimated vector is Z-sparse and deletes all the columns that do not correspond to the true signal. The OMP algorithm, on the other hand, takes the wrongly selected column to the end and it cannot be removed.

Algorithm 2 Compressive Sampling Matching Pursuit

Input: $\mathbf{A}, \mathbf{y}, \mathrm{Z}$
Initialize:
$S^0 \leftarrow \phi$
$\hat{s}^0 \leftarrow 0$
$t \leftarrow 1$
while stopping criterion **do**
  $r \leftarrow \mathbf{y} - \mathbf{A}s$ ▷ Residual error vector
  $\tilde{S}^t \leftarrow S^{(t-1)} \bigcup supp(l_{2\mathrm{Z}}(\mathbf{A}^T r))$
  $\tilde{s}^t \leftarrow \arg\min \left\|\mathbf{y} - \mathbf{A}(\mathbf{A}^T r)\right\|_2^2$ ▷ Least square error
  $\hat{s}^t \leftarrow l_{\mathrm{Z}}(\tilde{s}^t)$
  $S^t \leftarrow supp(\hat{s}^t)$
  $t \leftarrow t + 1$
end

4.5.3. Multiple Matching Pursuit (MMP)

MMP is referred to as the Tree-based Orthogonal Matching Pursuit algorithm because it executes a tree search with the help of a greedy strategy [88]. The main task of the MMP is to find the maximum likelihood that best matches the original sparse vector. Previously, maximum likelihood searches required enumerating all possible points with Z, which consumed substantial time. However, the MMP algorithm carries out an efficient tree search to locate the near-maximum likelihood [89]. In each iteration, MMP identifies multiple indices, and after some iterations, the smallest cost function is chosen as provided in [75].

4.5.4. Stagewise Orthogonal Matching Pursuit (St-OMP)

St-OMP is an extended version of the OMP algorithm. St-OMP operates for a specific number of iterations, and the threshold value should be determined. This algorithm is sensitive to the threshold value because different threshold values may result in different outputs [90]. The St-OMP is summarized in the Algorithm 3, where $\mathbf{e}$ is the threshold value, $e \in (0, 1]$ [91].

Algorithm 3 Stagewise Orthogonal Matching Pursuit

Input:$\mathbf{A}, \mathbf{y}, e$
Initialize:
$S^0 \leftarrow \phi$
$t \leftarrow 0$
$r^0 \leftarrow \mathbf{y}$
while $t = t + 1$ **do**
  $c^t = \mathbf{A}^T r^{t-1}$
  $I^t = \{I : |c_t(I)| > e\}$
  $S^t = S^{t-1} \bigcup \{I^t\}$
  $s^t = (\mathbf{A}_{S^t}^T \mathbf{A}_{S^t})^{-1} \mathbf{A}_{S^t}^T \mathbf{y}$
  $r^t = \mathbf{y} - \mathbf{A} s^t$
end

4.6. Performance Analysis

In this section, we present a general error probability expression for the proposed sparse recovery algorithms. The two most common errors in the sparse recovery algorithm are:

- The recovery algorithm wrongly detects support elements (non-zero index).
- The maximum likelihood finds incorrect symbols after correctly finding non-zero identification.

Let $s_{\Omega p}$ be the transmitted symbol and $s_{\widetilde{\Omega p}}$ be the incorrectly detected symbol (assume p and q are non-zero elements). The incorrect detector decision can be expressed as

$$\Pr(s_{\Omega p} \rightarrow s_{\widetilde{\Omega p}}) = \Pr\left(\left|\phi_{\Omega}, r^0\right| \leq max \left|\phi_i, r^0\right|\right). \quad (4.5)$$

Thus, the error probability of identifying the wrong support for a giving channel $\mathbf{h}$ is

$$\Pr(\Omega \mid \mathbf{h}) = \Pr\left(\left|\left\langle\phi_{\Omega}, r^t\right\rangle\right| \leq \underset{i}{max} \left|\left\langle\phi_i, r^t\right\rangle\right|\right) \quad (4.6)$$

where $\langle , \rangle$ is the inner product between two vectors and t is the iteration. For Z=2, assume the first support is $s_{\Omega p} = \Re_{\Omega p} + j\Im_{\Omega p}$ and the second support

$s_{\Omega q} = \Re_{\Omega q} + j\Im_{\Omega q}$. Thus, for the first iteration, we have

$$\Re\left\langle \frac{\phi_{\Omega p}}{\|\phi_{\Omega p}\|_2}, r^0 \right\rangle = \Re\left\langle \frac{\phi_{\Omega p}}{\|\phi_{\Omega p}\|_2}, \phi_{\Omega p} + \phi_{\Omega q} + w \right\rangle. \tag{4.7}$$

When $\mu_{ij} = 1$,

$$\Re\left\langle \frac{\phi_{\Omega p}}{\|\phi_{\Omega p}\|_2}, r^0 \right\rangle = \Re_{\Omega p}\|\mathbf{h}\|_2 + \Re_{\Omega q}\|\mathbf{h}\|_2\,\mu_{i\Omega_q} + \Re\left(\frac{\phi_{\Omega p}^T}{\|\phi_{\Omega p}\|_2}\right)w. \tag{4.8}$$

where μ_{ij} denoted the correlation between two column. The pairwise error probability (PEP) is given by

$$p(s_{\Omega p} \to s_{\widetilde{\Omega p}} \mid \mathbf{h}) = Q\left(\frac{\left\|\mathbf{h}(s_{\Omega p} - s_{\widetilde{\Omega p}})\right\|^2 \mu^*}{\sqrt{2\sigma^2}}\right) \tag{4.9}$$

where $\mu^* = \max\limits_{i\neq j}|\mu_{ij}| = \max|\langle\phi_i, \phi_j\rangle|$ is the maximum correlation between two columns of $\mathbf{A}$ and $Q(.)$ is the Q function.
Using the approximation

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{x^2}{2}\right)du. \tag{4.10}$$

The PEP is

$$p(s_{\Omega p} \to s_{\widetilde{\Omega p}} \mid \mathbf{h}) = \exp\left(\frac{\left\|\mathbf{h}(s_{\Omega p} - s_{\widetilde{\Omega p}})\right\|^2 \mu^*}{2\sigma^2}\right). \tag{4.11}$$

$$p(s_{\Omega} \to s_{\widetilde{\Omega}} \mid \mathbf{h}) = \prod_{i\neq\Omega}^{N}\mathbb{E}_h\left[\exp\left(\frac{\left\|\mathbf{h}(s_{\Omega p} - s_{\widetilde{\Omega p}})\right\|^2 \mu^*}{2\sigma^2}\right) \mid \mathbf{h}\right]. \tag{4.12}$$

Since $\|h\|_2^2$ follows the Chi-Square distribution with 2m degree of freedom [Lemma 2, [75]] we have

$$\begin{aligned}\mathbb{E}_h\left[\exp\left(\frac{\|\mathbf{h}\|_2^2}{2\sigma^2}\right) \mid \mathbf{h}\right] &= \int_0^\infty \exp\left(-\frac{x\mu}{\sigma^2}\right)\frac{x^{m-1}\text{exp(-x)}\mu}{(m-1)!}dx \\ &= \frac{1}{(\frac{1}{\sigma^2}-1)^m}.\end{aligned} \tag{4.13}$$

Similarly, the SER is

$$\mathbb{E}_h\left[\exp\left(\frac{\left\|\mathbf{h}(s_{\Omega p}-s_{\widetilde{\Omega p}})\right\|_2^2\mu^*}{2\sigma^2}\right)\mid\mathbf{h}\right]=\left(1+\frac{(s_{\Omega p}-s_{\widetilde{\Omega p}})\mu^2}{\sigma^2}\right)^{-m}. \tag{4.14}$$

Thus,

$$\prod_{i\neq\Omega}^{N}\mathbb{E}_h\left[\exp\left(\frac{\left\|\mathbf{h}(s_{\Omega p}-s_{\widetilde{\Omega p}})\right\|_2^2\mu^*}{2\sigma^2}\right)\mid\mathbf{h}\right]=\left(\left(1+\frac{(s_{\Omega p}-s_{\widetilde{\Omega p}})\mu^2}{\sigma^2}\right)^{-m}\right)^N \tag{4.15}$$

where μ is the maximum correlation to find the largest magnitude between the columns. It should be noted that the process of finding the largest magnitude is different in each algorithm, as shown in Section 4.5.

4.7. Results and Discussion

In this section, we analyse and compare the performance of the four sparse recovery algorithms. For a fair comparison, we used the same simulation arrangement for all the algorithms using the random binary spreading codebook with Z=2, $m = 42$, and $N = 96$. We compare the performance of the four algorithms, focusing on recovery error, covariance, recovery time and Block Error Rate (BLER).

- The recovery error [92] is defined as the error between the original sparse vector s_0 and the recovered sparse signal s_r

$$\text{Error} = \frac{\|s_0 - s_r\|_1}{\|s_0\|} \tag{4.16}$$

- The covariance is defined as the correlation between the sparse signal and the sensing matrix. Thus, the covariance of original sparse signal s_o, and recovered sparse signal s_r is

$$Cov\,(s_0, s_r) = \mathbb{E}([s_o - \mathbb{E}(s_0)]\,[s_r - \mathbb{E}(s_r)]) \tag{4.17}$$

where $\mathbb{E}$ is the expectation.

- Recovery time is the time needed to recover the sparse vector.

Table 4.1: Summary of the performance of four sparse recovery algorithms

Sparse recovery algorithms	***Column selection***	***Recovery time***	***Recovery error***	***Covariance***
OMP	$\max_j \left\{\mathbf{A}^T r_j\right\}$	Fast	High	Low
CoSaMP	$2\mathrm{Z}\max\left\{\mathbf{A}^T r_j\right\}$	Fast	High	High
MMP	$\max_j \left\{\mathbf{A}^T r_j\right\}$	Slow	Low	High
StOMP	$\{j : \lvert c_t(j)\rvert > e\}$	Fast	High	Low

- BLER is the ratio of incorrect blocks to total transmitted blocks.

Fig. 4.2 compares the recovery error of all four algorithms with respect to the number of measurements. Overall, the recovery error gradually decreases with increasing number of measurements. Interestingly, the MMP algorithm reached almost zero error in only 17 measurements. Fig. 4.3 shows the covariance of all four algorithms for different numbers of measurements. It is noticeable that the MMP algorithm has the highest covariance of all the algorithms up to about 20 measurements, above which all four algorithms asymptotically approach a covariance value of almost 2. Fig. 4.4 indicates that the MMP algorithm consumes significantly more time than the other algorithms, with the OMP algorithm being the fastest. Fig. 4.5 presents the BLER performance of the four algorithms as a function of SNR. Again, we observe the MMP algorithm outperforms the other three algorithms by a large margin.

Table 4.1 summarizes the performance of the four sparse recovery algorithms. From the table, it can be seen that the performance of MMP is better than that of the other three algorithms with minor recovery errors but suffers from a prolonged recovery time. The other three algorithms, OMP, CoSaMP, and St-OMP have shorter recovery times, but at the expense of a high recovery error. To summarize, the MMP algorithm performs well with sparse recovery with less error; however, it consumes more time, which does not meet the URLL requirements.

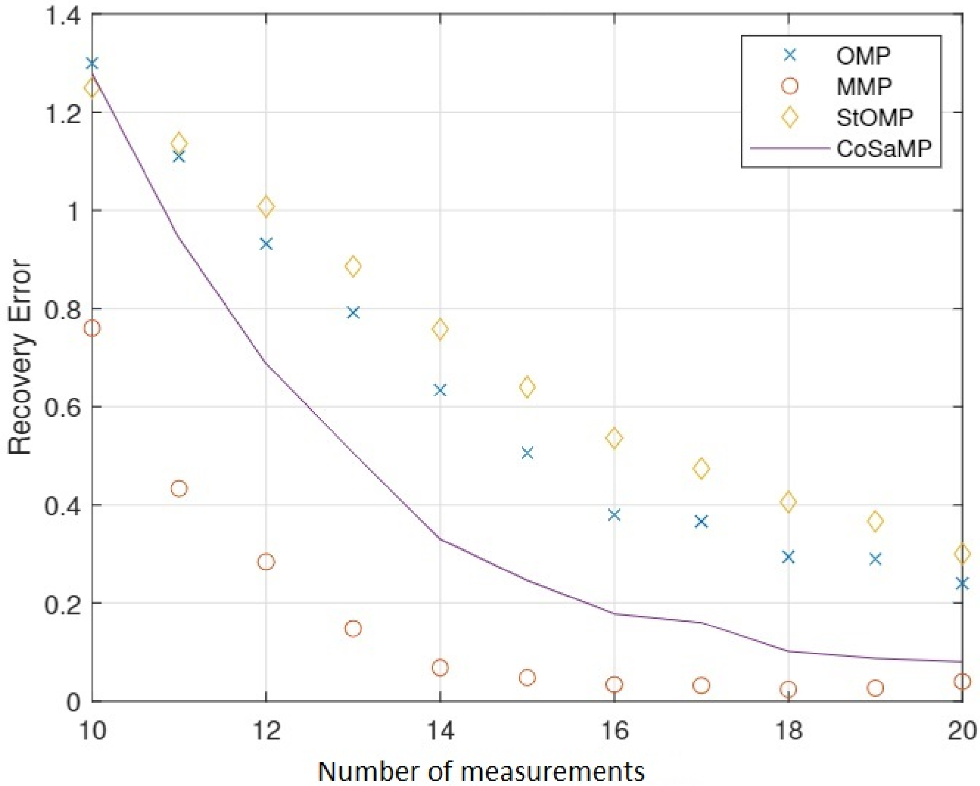


Figure 4.2: Recovery error with respect to the number of measurements.

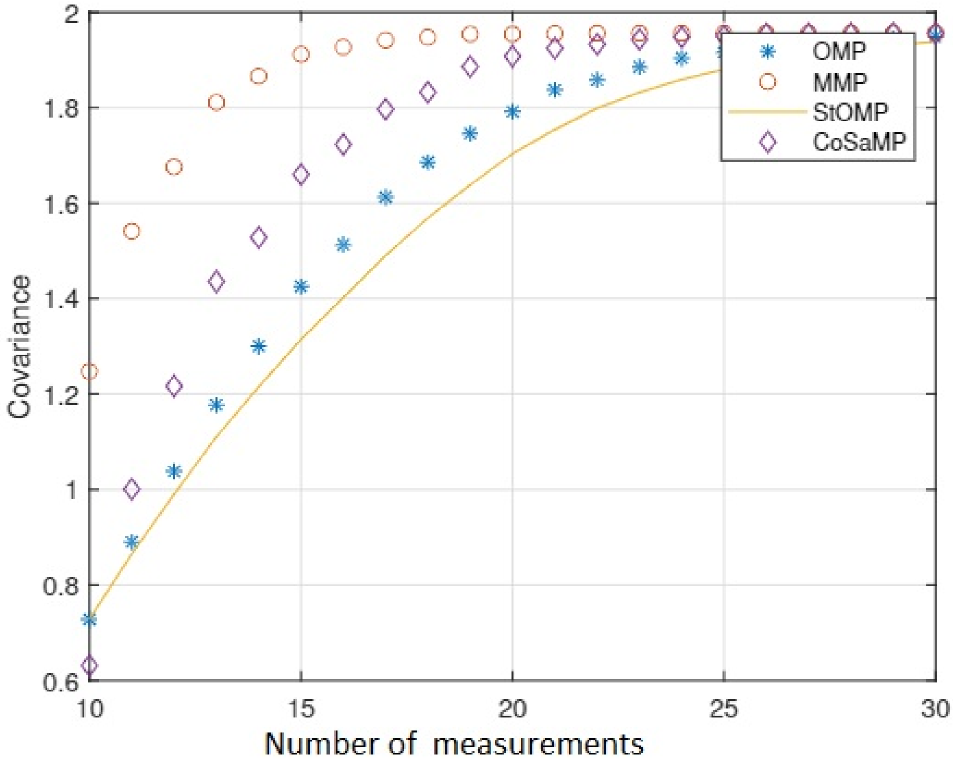


Figure 4.3: Covariance with respect to the number of measurements.

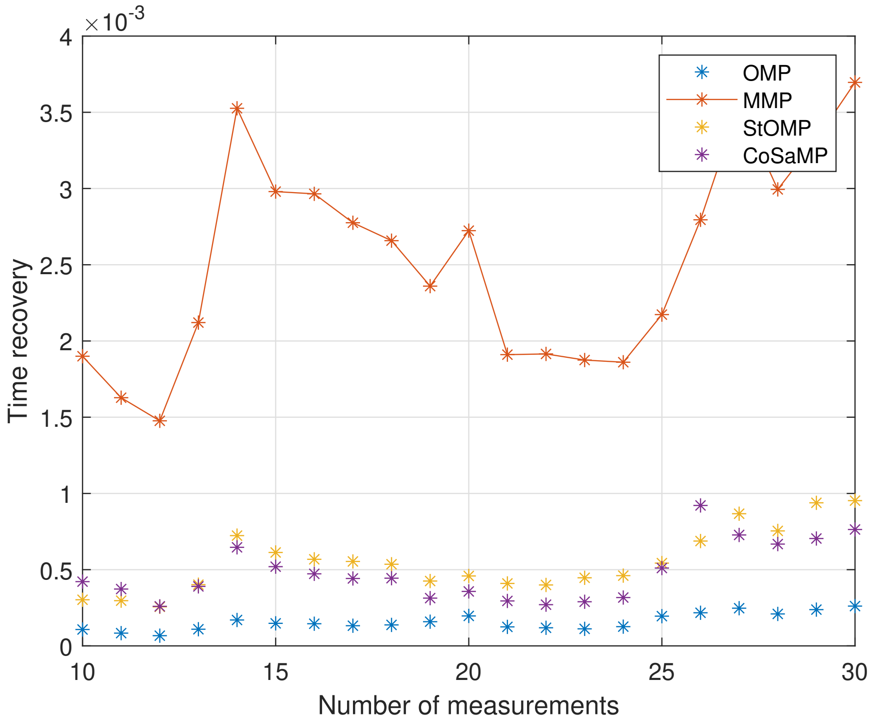


Figure 4.4: Recovery time with respect to the number of measurements.

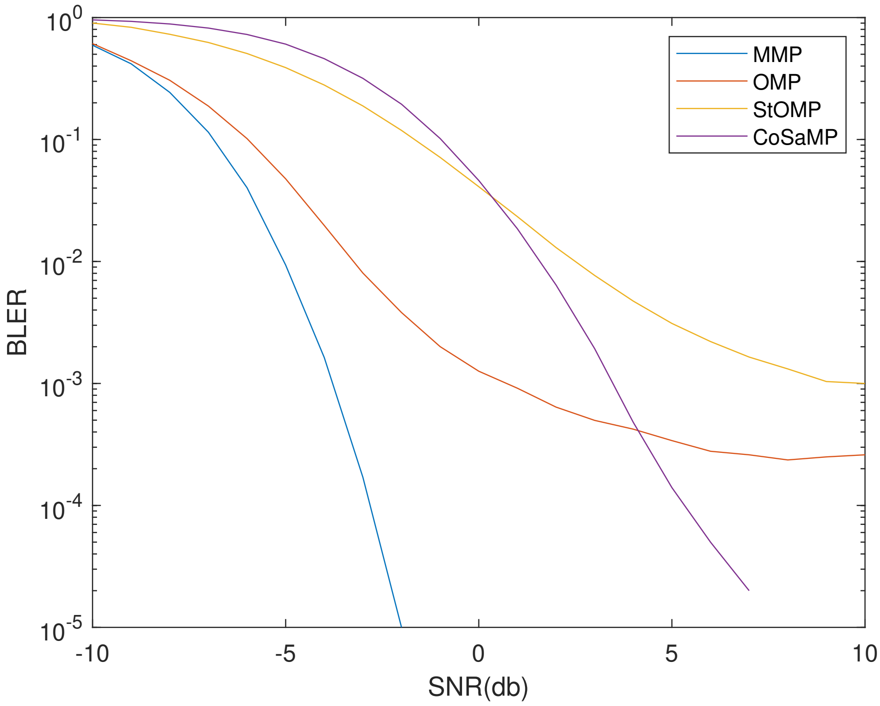


Figure 4.5: BLER with respect to SNR.

4.8. Chapter Summary

In this chapter, Compressive Sampling Matching Pursuit (CoSaMP) and Stagewise Orthogonal Matching Pursuit (St-OPM) algorithms have been applied to SVC and compared their performances with two other sparse recovery algorithms, Multipath Matching Pursuit (MMP) and Orthogonal Matching Pursuit (OMP) using recovery error, recovery time, covariance, and block error rate. Additionally, we provide a performance analysis of SER. The MMP algorithm showed minimum errors when recovering sparse elements, while OMP was the fastest. The MMP algorithm offers better performance than the other three algorithms, but time is a strict requirement with massive machine-to-machine communication and URLL. In the next chapter, supervised and unsupervised ML algorithms will be applied to reduce the packet overhead for SPC.

Chapter 5

Short Packet Communications: A Machine Learning Approach

This chapter introduces a ML approach for signal classification and estimation, which aims to recover the transmitted information using both supervised and unsupervised ML algorithms. Two methods are applied to support and improve this approach, Label Assisted Transmission (LAT) and Silhouette Analysis. These methods are explained in this chapter, providing insights into how they can be utilized to enhance the performance of SPC.
The rest of this chapter is organized as follows: Section 5.1 introduces the chapter and states its main contribution to SPC. A literature review of the application of ML in SPC is presented in Section 5.2. Section 5.3 describes the system model and the LAT method. Section 5.4 introduces the clustering framework with applied algorithms. Section 5.5 presents the simulation results showing the performance of the Gaussian Mixture Model with Expectation Maximization, Silhouette Analysis SVM, and KNN. Finally, the chapter summary is presented in Section 5.6.

5.1. Introduction

Massive machine-type communication (mMTC) devices, like temperature sensors, robots, and drones, possess a unique capability to transmit only a limited amount of command information. This information usually consists of instructions for movement, like right and/or left, or basic control commands, such as start and/or stop [93]. Unlike conditional systems that rely on infinite packet transmission, mMTC requires the use of short packets to convey this type of information effectively and efficiently. A notable distinction of finite block length packets is the non-negligible pilot overhead length. Therefore, it is crucial to explore suitable solutions tailored to the unique requirements of SPC systems. In this context, pilots, labels, and training symbols are synonyms that refer to the transmission of known symbols used for channel estimation.

In signal transmission, transmitted symbols are selected from predetermined constellation points based on the type of modulation employed. Due to the relationship between the transmitted symbols and the defined constellation points, the received signal naturally forms clusters. Utilizing this phenomenon, we propose a ML framework for signal detection. This framework aims to classify the received signal according to its corresponding cluster, which is represented according to the modulation type; however, there is no mapping to the transmitted symbol. Thus, we use LAT method to provide the correspondence between the transmitted symbol and clusters [94, 95]. Moreover, only one pilot, which corresponds to one constellation point, is transmitted to overcome the overhead problem in SPC.

The contributions of this chapter are summarized as follows:

- A bit recovery problem is formulated as a clustering and classification problem using the LAT framework to reduce the packet overhead.
- Two different supervised learning algorithms, namely SVM and KNN, are applied, and a comparison is made with the unsupervised learning algorithm referred to as modulation constraints- Gaussian mixture model (MC-GMM).
- Silhouette Analysis is proposed for determining the optimal clustering solution.
- The performance is evaluated in terms of Bit Error Rate (BER).

5.2. Related Work

The recent academic interest in applying ML to wireless communication has led to its successful application in tackling various challenges, such as channel estimation [96–98], and detection [99].

In the context of SPC, Leblanc et al. proposed a deep learning-based channel

estimator [100], which outperforms traditional methods in terms of both computational complexity and estimation accuracy. Hoang et al. [101] designed a deep multiple-output neural network (DMNN) framework to predict the matrices' reliability and energy efficiency performance in a multi-hop MIMO full-duplex system under different diversity schemes. Similarly, the authors in [66], introduced a deep neural network (DNN) algorithm in wireless-powered cognitive Internet-of-Things (IoT) networks for SPC, considering multiple primary receivers, where the relay selection is converted to a regression problem.
Several studies have been conducted to reduce the pilot overhead in SPC [102–104]. In [102], the authors proposed a receiver design that uses a virtual pilot by exploiting the reliable data symbol for channel estimation. While in [103], an optimization algorithm is introduced to reduce the packet overhead under the constraints of block error probability and block length. In [104], the authors obtained an upper bound on packet error rate and optimized the pilot symbols at a given block length and fixed rate over the MIMO system. In addition, [94] and [95] used an Expectation Maximization clustering algorithm for signal detection, and pilot assisted transmission to minimize the packet overhead.
However, some researchers have adopted alternative techniques to minimize the overhead in SPC. For instance, Ji et al. [86] used sparse vectors to transmit short packets, while Wu et al. [105] proposed a new transmission method without pilot signals using sparse vectors. The proposed schemes used a non-zero position of the sparse vector for data encoding, and employed a DNN to retrieve the non-zero sparse vector.

5.3. System Model

This section presents a system model for SPC over frequency-flat channels. We consider a connected transmitter and receiver pair with one antenna each. The channel response h is assumed to be constant over a packet of N symbols and may vary from packet to packet. Denoting the Q-ary symbol set as $\{\mathbf{S} = S_0, \ldots, S_{Q-1}\}$, and x_n, as the transmitted symbol, where $x_n \in \mathbf{S}$. The received signal,y_n, at time n is given as

$$y_n = \sqrt{P}hx_n + w_n \tag{5.1}$$

where $n = 0, 1, \ldots, N-1$, and P is the transmitted power. Without loss of generality, let $P = 1$ and assume w_n is complex Gaussian noise with zero mean and variance σ^2. Also, in this work, the noise variance is assumed to be the same for all clusters.

5.3.1. Label Assisted Transmission

Channel estimation can be accomplished by transmitting known training symbols, utilizing semi-blind methods, or applying blind techniques. Throughout this thesis, we employ two distinct methods for transmitting a known label: separate pilot transmission and superimposed pilot transmission. The idea behind superimposed labels is that we can transmit training symbols in the same time duration as the data symbols. However, the separate pilot transmission method utilizes a part of the overhead separate from the data.

The clustering algorithms classify the received signals, and then the transmitted label symbols are used to indicate the mapping between each cluster and the corresponding transmitted data. The transmitted labels are superimposed with the data using the EM algorithm but are transmitted separately in the training phase of the SVM and KNN algorithms. To reduce the packet overhead that may be caused by sending labels both ways, we apply the label-assisted transmitting method provided by [95]. The idea behind this technique is to transmit one label that forms one of the constellation points. In this work, we consider Quadrature Phase Shift Keying (QPSK), with $\mathbf{S} = \left[\frac{1+j1}{\sqrt{2}}, \frac{1-j1}{\sqrt{2}}, \frac{-1+j1}{\sqrt{2}}, \frac{-1-j1}{\sqrt{2}}\right]$ and use complex multiplication to reconstruct others by predefined complex values $\alpha = [1, j, -1, -j]$. The received label is expressed as

$$\breve{y} = \breve{x}\breve{h} + \breve{w} \tag{5.2}$$

where $\breve{x} = \frac{1+j1}{\sqrt{2}}$ and $\breve{w}$ is the noise. The other labels are reconstructed using the following

$$\breve{y}_c = \breve{y}\alpha_c \tag{5.3}$$

where c indicates the c-th cluster, and $C = 4$ is the number of groups used in QPSK.

5.4. Clustering Framework

In a wireless communication system, the received signals are divided into C groups at the receiver, depending on the modulation type. Therefore, if the mapping between the transmitted bit tuples and the received groups is known, the detection problem can be considered a form of clustering problem.
Thus, in this work, clustering algorithms are employed, supported by the LAT method, to group the received signals into four clusters based on QPSK modulation. It is worth emphasising that each algorithm follows a different working process. For instance, supervised learning algorithms require training data prior to the clustering process, whereas unsupervised machine learning utilizes labels during the clustering process.

5.4.1. MC-GMM

The Gaussian mixture model (GMM) is an unsupervised clustering method for clustering analysis. In the context of this work, GMM is used to cluster the received signals, and the LAT method is used to recover the transmitted bit. Since the noise is considered to be Gaussian, the received signals can be expressed in the form of the GMM [106]

$$p(y_n; \mathbf{\Phi}) = \sum_{c=0}^{C-1} \pi_c \mathcal{CN}(y_n; \mu_c, \sigma_c^2) \tag{5.4}$$

where $\mathbf{\Phi} = \{\pi_c, \mu_c, \sigma_c^2\}_{c=0}^{C-1}$ is a set of the unknown parameters; π_c the prior probability of a transmitted symbol, μ_c and σ_c^2 are the mean and the variance respectively. $\mathcal{CN}(y_n; \mu_c, \sigma_c^2)$ is the probability density function (PDF) of the complex Gaussian

$$\mathcal{CN}(y_n; \mu_c, \sigma_c^2) = \frac{1}{\pi\sigma_c^2} \exp\left(\frac{-\|y_n - \mu_c\|}{\sigma_c^2}\right) \tag{5.5}$$

In GMM clustering, we need to find the three unknown parameters $\{\pi_c, \mu_c, \sigma_c^2\}$ for each cluster [95]. However, some digital communication characteristics can be used as constraints to reduce the number of parameters. This is called modulation constraint (MC) [95], and can be summarized as following

- The variance is considered to be equal for all clusters. Thus, instead of finding σ_c^2 for each cluster, we need to find only σ^2.

- The symbol is transmitted from the defined constellations. Thus, the transmitted symbol can be assumed to have equal probability $\pi_c = 1/C$. For example, in QPSK, $\pi_c = 1/4$.

- From the LAT method, all symbols in $\mathbf{S}$ can be denoted as $\breve{x}$, such that $S_c = \breve{x}\alpha_c$. Let us define $\breve{\mu} = h\breve{x}$ as the mean of $\breve{x}$ and the mean of c-th cluster is $\mu_c = hS_c$. Thus, the mean of each cluster can be also defined in term of $\breve{\mu}$ as $\mu_c = \breve{\mu}\alpha_c$.

Accordingly, the parameters in $\mathbf{\Phi}$ are reduced to $\mathbf{\Phi} = \{\breve{\mu}, \sigma^2\}$. The Log-likelihood is given by [95] as

$$\max_{\mathbf{\Phi}} LL(\mathbf{\Phi};\mathbf{y}) = \max_{\mathbf{\Phi}} \sum_{n=0}^{N-1} \ln\left(\sum_{c=0}^{C-1} \frac{1}{C}\mathcal{CN}\left(y_n; \breve{\mu}a_c, \sigma^2\right)\right). \tag{5.6}$$

To solve Eq. (5.6), the EM algorithm is used with a defined latent variable. A latent variable used for each received signal is defined as $\mathbf{l_n} = \left[l_{n,0}, \cdots, l_{n,(C-1)}\right]^T$, where

$$l_{n,c} = \begin{cases} 1 & \text{if } x_n = S_c \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

For $n = 0, \ldots, N-1$ and $c = 0, \ldots, C-1$. Thus, the log-likelihood of $\mathbf{\Phi}$ given $\mathbf{y}$ and $\mathbf{L} = [\mathbf{l}_0, \cdots, \mathbf{l}_{N-1}]^T$ becomes

$$LL(\mathbf{\Phi};\mathbf{y},\mathbf{L}) = \sum_{n,c} l_{n,c}\left(\ln\frac{1}{C} + \ln\mathcal{CN}\left(y_n; \breve{\mu}a_c, \sigma^2\right)\right). \tag{5.8}$$

Then, EM is executed in two steps interactively to obtain a solution for $\mathbf{\Phi}$ [106], see Fig. 5.1. Here $\hat{\mathbf{\Phi}} = \left\{\hat{\breve{\mu}}_{old}, \hat{\sigma}^2_{old}\right\}$ and $\hat{\mathbf{\Phi}} = \left\{\hat{\breve{\mu}}_{new}, \hat{\sigma}^2_{new}\right\}$, denoted the value of $\mathbf{\Phi}$ in the E-step and M-step, respectively. The E-step determines the initial values of Φ before the iteration (denoted as old). The expectation of $l_{n,c}$ giving $\mathbf{y}$ and $\mathbf{\Phi}$ can be expressed as [95]

$$\begin{aligned} \tilde{\gamma}_{n,c} &\triangleq \mathbb{E}\left[l_{n,c}|\mathbf{y}; \hat{\mathbf{\Phi}}_{old}\right] \\ &= \frac{\mathcal{CN}\left(y_n; \hat{\breve{\mu}}_{old}\alpha_c, (\hat{\sigma}^2)_{old}\right)}{\sum_{\ell=0}^{C-1}\mathcal{CN}\left(y_n; \hat{\breve{\mu}}_{old}\alpha_\ell, (\hat{\sigma}^2)_{old}\right)}. \end{aligned} \tag{5.9}$$

The M-step: setting the log-likelihood derivatives of Eq. (5.8) with respect to $\mathbf{\Phi}$ to zero, we get the following expression for the estimated mean and variance [95]

$$\hat{\mu}_{new} = \frac{\sum_{n=0}^{N-1}\sum_{c=0}^{C-1}\tilde{\gamma}_{n,c} y_n \alpha_c^H}{\sum_{n=0}^{N-1}\sum_{c=0}^{C-1}\tilde{\gamma}_{n,c}|\alpha_c|^2} \tag{5.10}$$

$$(\hat{\sigma}^2)_{new} = \frac{\sum_{n=0}^{N-1}\sum_{c=0}^{C-1}\tilde{\gamma}_{n,c}|y_n - \hat{\mu}_{new}\alpha_c|^2}{\sum_{n=0}^{N-1}\sum_{c=0}^{C-1}\tilde{\gamma}_{n,c}} \tag{5.11}$$

The output of the clustering algorithm classifies the received signal according to the labels. Thus, symbol-cluster mapping is used to recover the transmitted information using the relations of the reconstructed labels. Therefore, y_n is assigned to cluster $c_n^* = \arg\max_c \tilde{\gamma}_{n,c}$, and the estimated symbol $\hat{x}_n = S_{c_n^*}$, where $S_{c_n^*} = \alpha_{c_n^*}\breve{x}$.

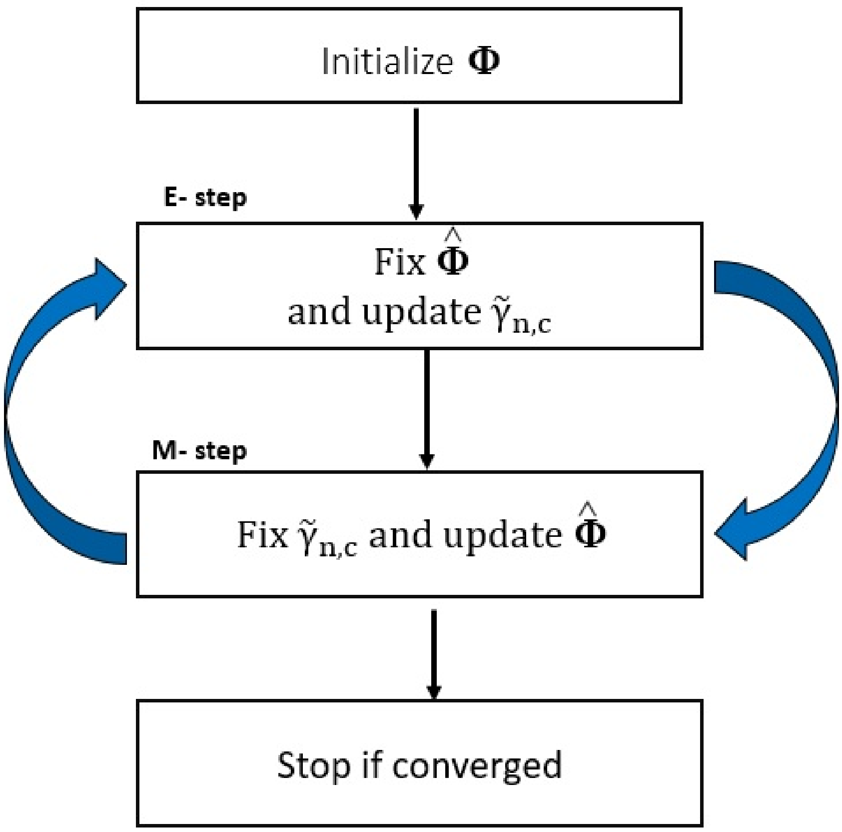


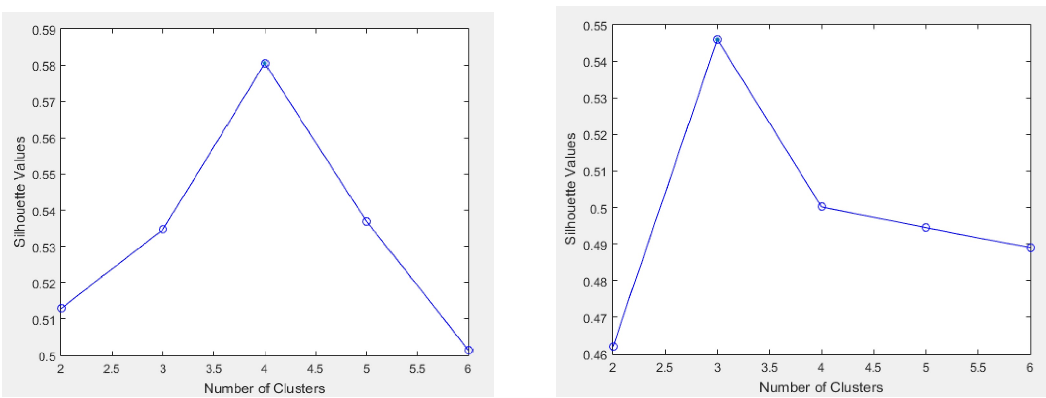
Figure 5.1: EM algorithm for MC-GMM.

5.4.1.1 Optimal Number of Clusters Using Silhouette Analysis Method

In unsupervised ML algorithms, the number of clusters must be predefined. However, due to the presence of noise, the number of received clusters detected may be affected. Thus, Silhouette Analysis is introduced is applied at the receiver to find the actual number of received clusters.

Silhouette Analysis is used to find the optimal number of clusters. The Silhouette value ranges between 0 and 1, with 1 being the ideal number of clusters. For QPSK, the expected number of clusters is four according to the constellation points. Therefore, if the received clusters are less or more than four, this indicates that the whole transmitted packet has been affected by noise, in which case it is hard or even impossible to recover the information.

Fig. 5.2 shows the performance of Silhouette Analysis for QPSK modulation clusters with two different noise levels. Fig. 5.2(a) with SNR=4 dB, it is seen that the highest Silhouette value is when the number of clusters is four, which is the expected cluster number with low noise. Fig. 5.2(b) shows the optimal number of received clusters is three due to the high level of noise. In future, Silhouette Analysis can be added at the receiver to improve the system performance by assessing the correct number of clusters according to the modulation type. Precisely, erase the packet with the wrong number of received clusters before decoding it because it has been corrupted by noise.



(a) Silhouette with SNR=4 dB

(b) Silhouette with SNR=-100 dB

Figure 5.2: Silhouette Analysis

5.4.2. SVM

The SVM algorithm was primarily developed as a two-class classifier to separate data into two distinct classes. However, in order to tackle the challenge of multiple-class problems, SVM has been extended to solve multi-class scenarios by utilizing multi-class classifiers. This extension enables SVM to effectively handle situations involving multiple classes. Additionally, SVM has demonstrated its capability in channel estimation and data detection [107].
In a QPSK model, four constellation points are employed, with each point corresponding to a unique class. Therefore, we utilized a multi-class SVM classifier to satisfy the earlier mentioned condition of equal probability for QPSK, where $\pi = 1/4$, see Fig. 5.3. The overall process is carried out in two stages: the first stage involves channel estimation, while the second stage focuses on data symbol detection. During the training process, labelled symbols are initially indicated based on the LAT method mentioned in Section 5.3.1. Subsequently, the remaining labels are reconstructed using Eq. (5.3). The SVM training data are then represented as

$$\mathbf{y}_p = \operatorname{sign}\left(\mathbf{h}_p \mathbf{x}_p + \mathbf{w}_p\right) \tag{5.12}$$

where $\mathbf{w}_p$ can acts as the bias and sign is the sign function. The received signal can be expressed as

$$\mathbf{y} = \operatorname{sign}\left(\mathbf{h}\mathbf{x} + \mathbf{w}\right). \tag{5.13}$$

In SVM multi-class problems, we apply the rule of one versus all [108], in which the received label (indicating the constellation point) belongs to a cluster if the other clusters reject it. Specifically, each point is tested for all clusters but only assigned to the correct cluster. So, to find the best classification, SVM solves the optimization problem

$$\begin{aligned} &\min_{\mathbf{h}_p} \frac{1}{2} \left\|\mathbf{h}_p\right\|^2 + R \sum_{p=1}^{P} \xi_p \\ &\text{s.t.} \quad \mathbf{y}_p \left(\mathbf{h}_p\right)^T \mathbf{x}_p \geq 1 - \xi_p, \\ &\xi_p \geq 0, \ p = 1, \ldots, P \end{aligned} \tag{5.14}$$

where the factor 1/2 is added for the convenience of the derivative [106], R is the regularization parameter which controls the error between the margin and the slack

variable ξ_i. Let's define $\hat{\mathbf{h}}_p$ as a solution of Eq. (5.14).
We assume the labels and the data are experiencing the same fading channel. Then, the channel in Eq. (5.12) and the data channel are the same, where $\hat{\mathbf{h}}_p = \mathbf{h}$. As discussed in Section 5.4.1, the transmitted symbol can be expressed from the defined constellation set, and $\mathbf{h}$ is defined by Eq. (5.14). Then, the second stage provides a solution for $\mathbf{x}$ by solving the optimization problem

$$\begin{aligned} &\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x}\|^2 + R \sum_{n=1}^{N} \xi_n \\ &\text{s.t.} \quad \mathbf{y}(\mathbf{x})^T \mathbf{h} \geq 1 - \xi_n \\ &\xi_n \geq 0 \end{aligned} \tag{5.15}$$

where $\hat{\mathbf{x}}$ is denoted a solution of Eq. (5.15).

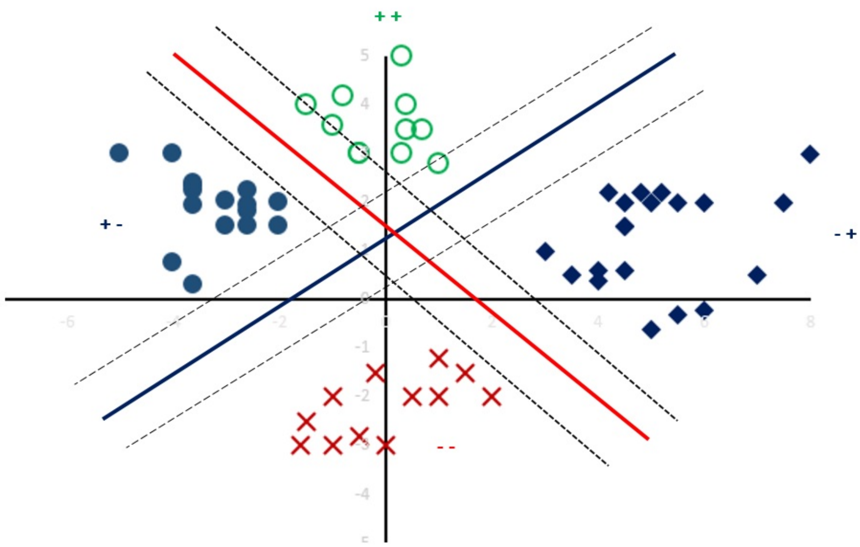


Figure 5.3: Example of four classes SVM algorithm to model QPSK.

5.4.3. KNN

The process begins by sending a single label and subsequently reconstructing the remaining labels using Eq. (5.3). Once all four labels have been reconstructed, the Euclidean distance is computed between each received symbol and the labelled data points. In the KNN algorithm, it is essential to specify the number of neighbours, K_n, initially. These neighbours are then used as votes, and the symbol is assigned to the cluster from which it receives the highest number of votes from its nearest neighbours. Determining an optimal value for the number of neighbours (K_n) is crucial for the performance of the KNN algorithm. This can be achieved through techniques such as cross-validation or grid search, which evaluate the algorithm's accuracy for different values of K_n, and select the value that yields the best results. However, it can also be determined according to the system model. For instance, since we have four labels in this scenario, we can select K_n to be equal to four. The Euclidean distance is given as

$$d(\breve{x}, y_n) = \sqrt{\sum_{n=1}^{N} (\breve{x}_c - y_n)^2}. \tag{5.16}$$

After assigning the received signal to its corresponding cluster, we can utilize the information provided by Eq. (5.3), indicating that the clusters' order corresponds to the reconstructed labels. By considering this, we can obtain the estimated symbol from

$$\hat{x} = \breve{x}\alpha_c \tag{5.17}$$

5.5. Results and Discussion

This section presents the bit error rate (BER) with signal-to-noise ratio (SNR) results to show the effectiveness of the applied ML to SPC using Monte Carlo simulations. Each Monte Carlo simulation indicates one packet transmission. The transmission symbols employ the QPSK modulation scheme. We consider a flat fading channel with AWGN and packet length: $N = 50$ and C=4.

Fig. 5.4 illustrates the BER performance of the MC-GMM in comparison to two alternative detection schemes: the maximum likelihood detector (MLD) with perfect channel state information (CSI), as discussed in Sklar's work [109], and MLD with estimated CSI as detailed in Zhang's investigation [95]. The results for MC-GMM demonstrate superior performance compared to the estimated CSI approach. In alignment with these results, Fig. 5.5 shows the iterative steps employed to achieve optimal clustering output for EM algorithm.

Fig. 5.6 presents the BER versus SNR performance of KNN and SVM compared with MC-GMM algorithm. It is shown that supervised learning (KNN and SVM) outperforms unsupervised learning (MC-GMM). It is observed that the KNN

exhibited marginally superior performance compared to the MC-GMM algorithm. The outcome obtained using KNN is for a supervised learning algorithm where the receiver is trained by predefined labels prior to receiving the data.
However, it is important to mention that KNN is a soft classification method, which implies that some data points of a certain cluster may overlap with other clusters, as shown in Fig. 5.7(a). Similar to the observation made for KNN, it is noted that the SVM demonstrates a slightly superior performance compared to the MC-GMM algorithm. This advantage is again attributed to the supervised nature of SVM. However, we also note it has a slightly better performance than KNN. It is necessary to note that SVM is a hard classification method, which entails that some data points situated at the boundaries of clusters may be erroneously assigned to the incorrect cluster due to strict classification boundaries, as illustrated in Fig. 5.7(b).

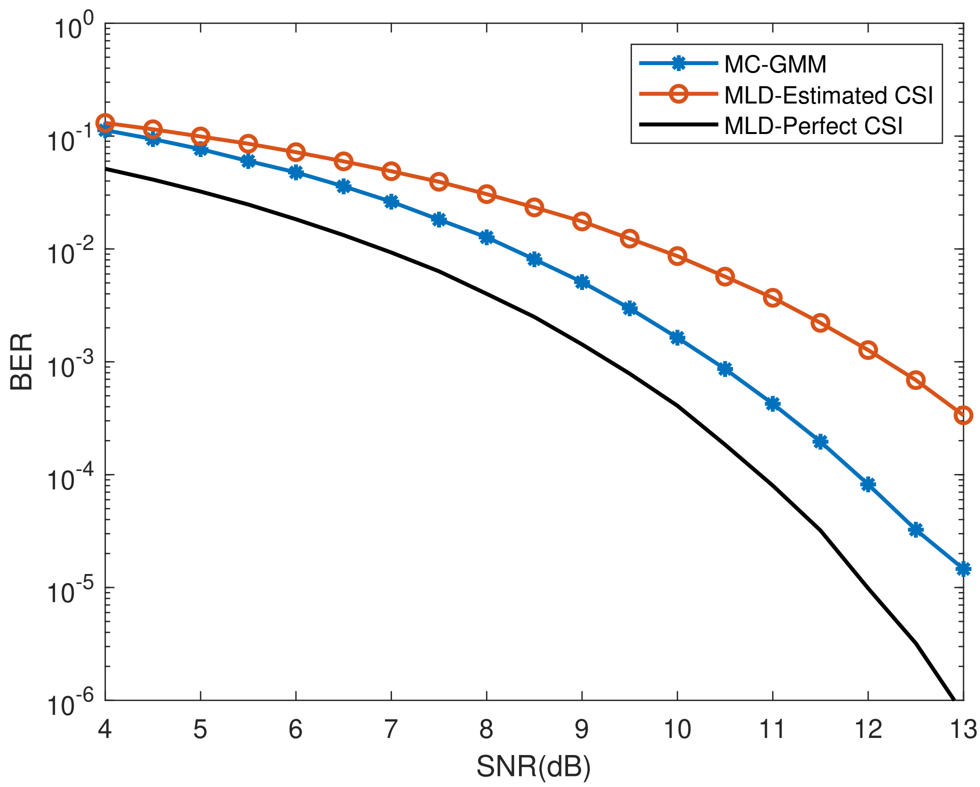


Figure 5.4: BER performance of MC-GMM, MLD-estimated CSI, and MLD-perfect CSI.

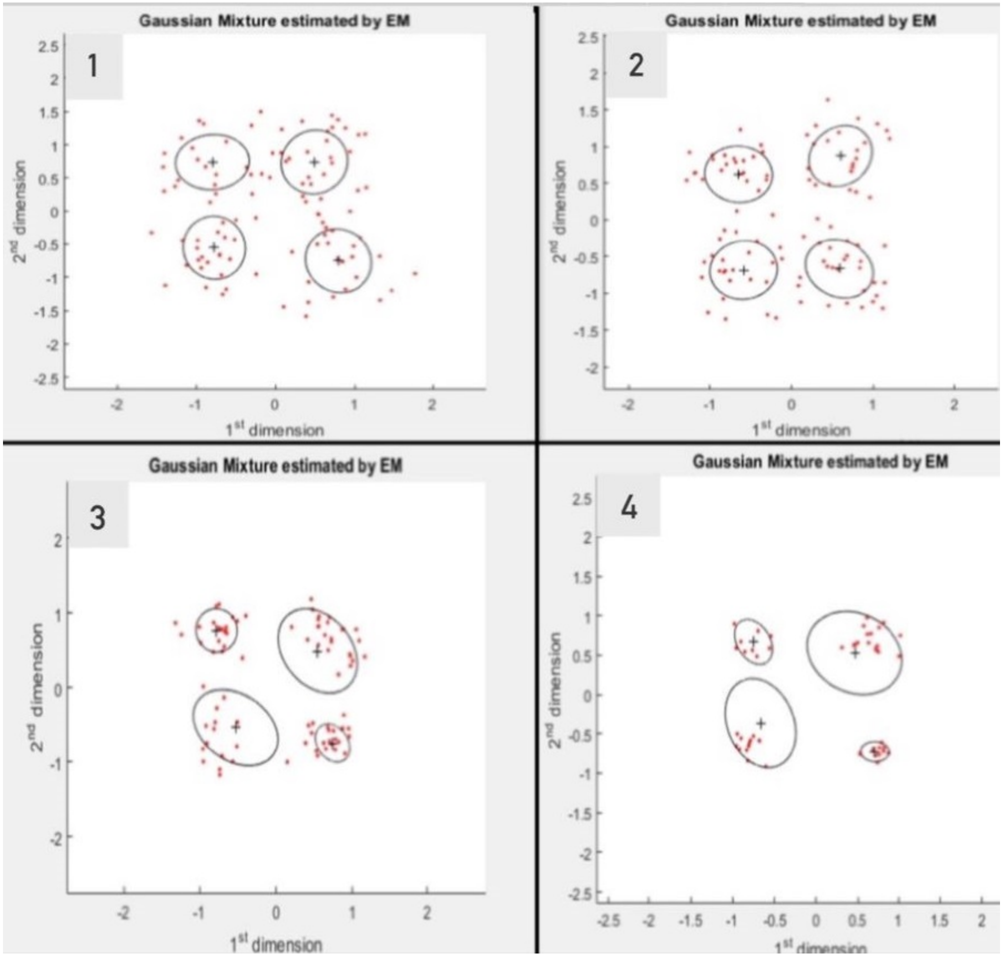


Figure 5.5: EM clustering process until convergence.

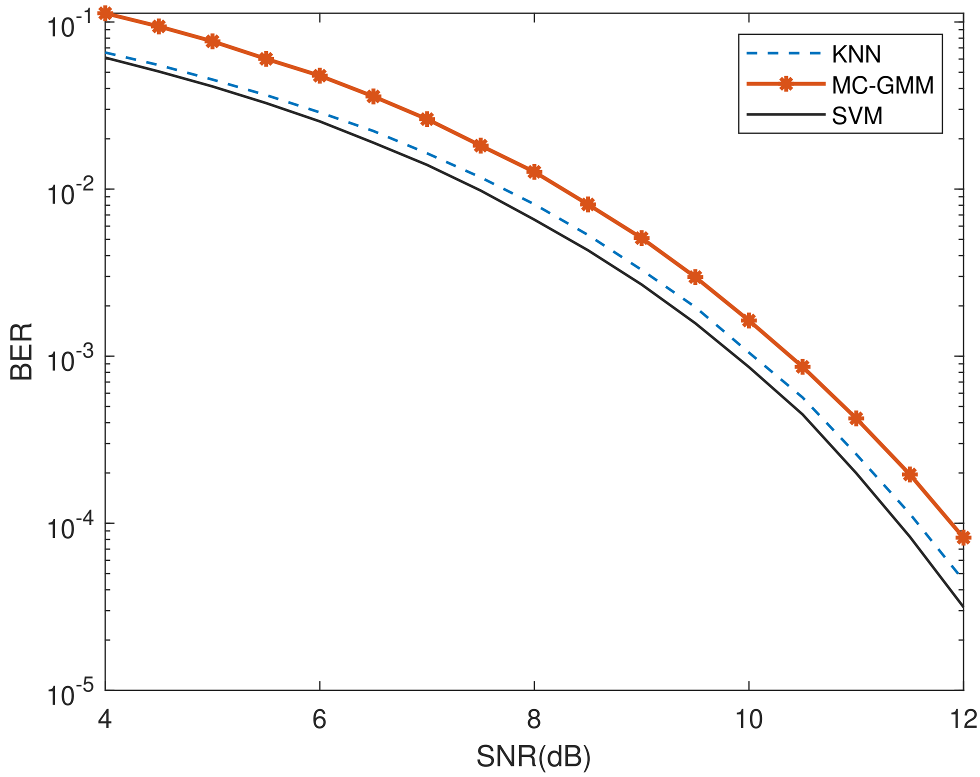
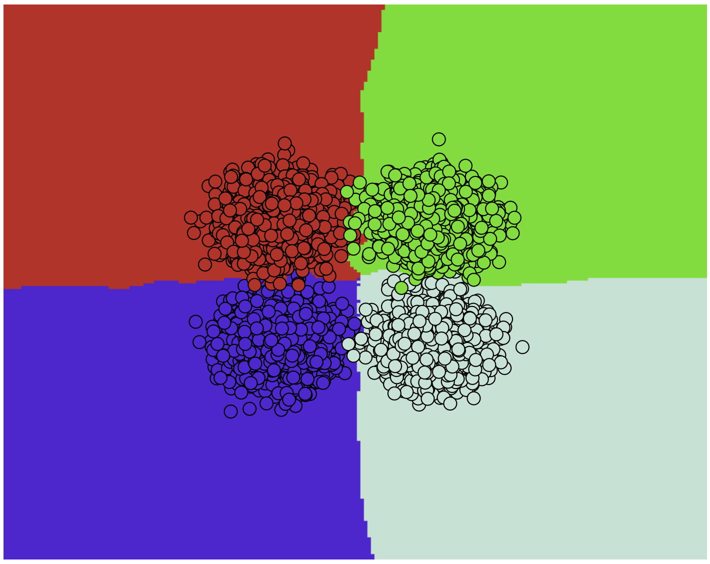
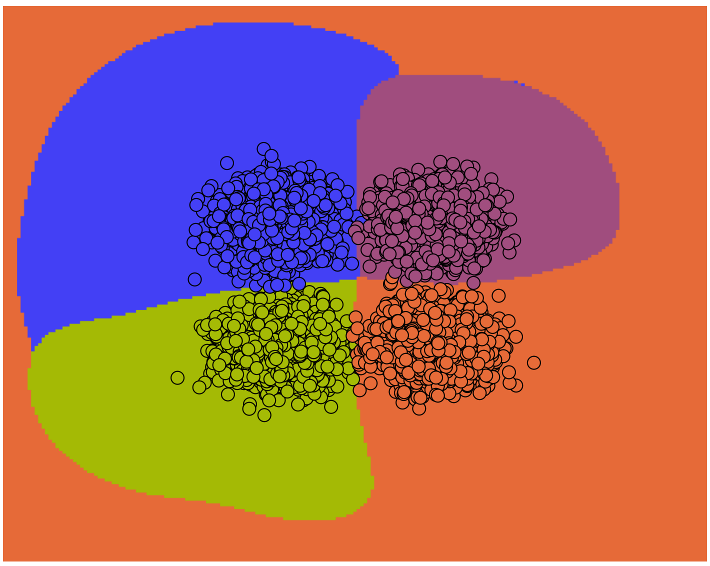


Figure 5.6: BER performance of KNN, MC-GMM and SVM.



(a) KNN classification output for QPSK



(b) SVM classification output for QPSK

Figure 5.7: Supervised Learning Classification output

5.6. Chapter Summary

This chapter has presented various machine-learning approaches for symbol detection in SPC systems, namely the MC-GMM, KNN, and SVM. The ML algorithm demonstrates better performance compared to the conventional system. KNN and SVM perform slightly better than the MC-GMM algorithm, mainly due to their supervised nature. The LAT method has been applied to address the overhead in short packets, which helps to reduce the number of transmitted labels utilizing signal properties such as equal probability among received clusters. Moreover, the introduction of Silhouette Analysis offers insights into optimizing the clustering performance under various noise conditions.

Chapter 6

Application of Machine Learning in LoRa

In the LoRa system, interference is highly correlated with the spreading factor (SF) values of the end users. In this chapter, we propose applying a new machine learning framework to the LoRa communications system to recover the SF. This framework converts the SF recovery in the LoRa network to a multiclass problem using ML algorithms. We first propose and then evaluate the performance of two classification algorithms: a support vector machine (SVM) and k- nearest neighbours (KNN). The performance of the proposed algorithms is investigated by Monte Carlo simulation, which shows promising results with both SVM and KNN.

The rest of this chapter is organized as follows: Section 6.1 introduces the chapter and states the main contribution made by this research for signal detection. A literature review of the application of ML and SF assignment in LoRa is presented in section 6.2. Section 6.3 describes the problem and formulates the system model. Section 6.4 describes the classification framework, and Section 6.5 presents the simulation results showing the performance of SVM and KNN. Finally, a summary of this chapter is presented in Section 6.6.

6.1. Introduction

Low-Power Wide-Area Networks (LPWANs) are one component of technologies concerned with transmitting and receiving small amounts of data, requiring the use of short packets to maximize bandwidth efficiency [110]. LPWANs have emerged as a promising solution for low-power, long-range communication with LoRa being one of the most widely used LPWAN technologies [111]. The success of LoRa transmissions depends on three main parameters, bandwidth (BW), transmission power (TP), and spreading factor (SF). The SF parameter plays an important role in the LoRa transmission performance. Increasing the SF value yields a lower data rate but enhances signal resilience to noise, which improves the transmission range. However, it also increases transmission duration and so increases power consumption. Considering the significant impact of SF on the success of the transmission and interference resistance, it is considered one of LoRa's most critical parameters.

The majority of reported studies have been concerned with SF assignments at the transmitter; however, in practice, the transmitted signal is affected by many natural factors, such as fading, which may lead to misdetection of a signal due to a large number of nodes transmitting with different SFs. Hence, SF recovery at the receiver helps improve the system's performance, overcoming the interference due to other networks.

The motivation behind this work is to enhance the performance of LoRa by applying a new supervised ML-based SF classification assignment scheme using KNN and SVM algorithms to find the optimal SF, i.e., we introduce a new supervised ML method to solve the SF recovery problem. Utilizing the power of classification algorithms, we formulate the SF recovery as a multiclass classification problem, with each value of SF assumed to be a different class. Thus, in the LoRa system, we have six classes with label values in the range of [7,.., 12]. The motivation behind this work is to enhance the performance of LoRa by applying a supervised ML-based SF classification assignment scheme using KNN and SVM algorithms to find the optimal SF. Our main contributions can be summarized as follows:

- A new supervised machine learning algorithm is developed to recover a target Spreading Factor (SF) in the LoRa system, with SF recovery considered a multiclass problem using the SVM and KNN algorithms.

- A confusion matrix is presented to display the prediction accuracy of the proposed algorithms.

6.2. Related Work

In LoRa, there are two main research directions concerned with SF: SF allocation assignment and SF recovery or identification. Assigning the appropriate SF is crucial for good long-distance performance, with many possible approaches for SF allocation. In [112], the authors introduce two SF allocation techniques. The first technique considers the sensitivity and the signal strength of the end devices at the initial deployment stage. The second technique is that the SF recovery algorithm depends on the status of the channel, with the authors claiming the technique achieved a higher packet transmission compared to other SF assignment techniques. In [46], the authors propose a new scheme whereby the SF is determined by considerations of transmission distance and power. The smallest value of SF is assigned to the nearest signal, e.g., SF=7 and a higher SF value is assigned to the weaker signal. In [113], the authors considered compressive sensing to detect the bandwidth and the SF. This work used embedded preambles in the packets to obtain the bandwidth and the duration of the symbol for the signal and then the SF.

Due to the success of ML algorithms in wireless communications [114], some researchers have attempted to apply ML techniques to LoRa SF assignments. The authors in [46] apply the decision tree (DT) technique for allocating SF. In [115], the authors provided a comparison of supervised classification algorithms, Naive Bayes and KNN for SF allocation in LoRa, assuming that the area classified to circles around the receiver, e.g., the closest circle is for SF=7. The authors in [116] exploited a new architecture for selecting the optimal SF using deep learning (DL) with both a convolutional neural network (CNN) and a fully connected neural network (FCNN) to find the optimal SF. In [117], the authors proposed a deep learning method, specifically a CNN, to detect the LoRa signal and the interference from other LLPWAN techniques such as IEEE 802.15.4g and SigFox.

On the other hand, few researchers have studied SF recovery. Recently, the authors in [118] considered the classical SF detection methods that mainly depend on optimizing the packet overhead. In contrast, the authors in [119] use a phase-locked loop (PLL) method to detect the SF at the receiver.

6.3. System Model

We consider typical LoRa wireless networks [50], where chirp spreading modulation (CSM) is adopted as a modulation technique. In this system, a CSM device transmits a symbol, denoted as s, with a duration of $T_s = 2^{\alpha}.T$, where α is the SF value and T is the duration of bits transmission. The symbol corresponds to 2^{α} bits transmitted per frame and can take values from the set $\{0, 1, 2, ..., 2^{\alpha-1}\}$. The channel h represents a quasi-static Gaussian independent and identically distributed (i.i.d.) slow fading channel. The transmitted vector is represented as [120]

$$\mathbf{x} = \left[\frac{1}{\sqrt{2^{\alpha}}} \exp\left(j2\pi \left[(s+i)_{mod2^{\alpha}} \right] \frac{i}{2^{\alpha}} \right) \right]^T \tag{6.1}$$

where $i = 0, \cdots, 2^{\alpha-1}$. The received signal is

$$\mathbf{y} = \mathbf{x}h + \mathbf{w} \tag{6.2}$$

where $[.]^T$ represent vector transpose, $\mathbf{w}$ is assumed to AWGN noise with zero mean and variance σ^2, i.e., $\mathbf{w} \sim CN(0, \sigma^2)$. Assume $|h| = 1$ without loss of generality, and the $SNR = 1/\sigma^2$.

6.3.1. SF Labels Method

In CSM systems, the possible waveforms are orthogonal, so the CSM devices adopt appropriate SFs for transmission to ensure orthogonality. Each transmitted signal in LoRa has a specific value of SF, indicating a symbol of 2^{α} chips in each chirp. Thus, each transmitted signal has a different pattern depending on the SF value, e.g., $2^7 = 128$ chips/ symbol, while $2^{12} = 4096$ chips/symbol, see Fig. 6.1. SF identification can be considered as a classification problem since SFs can have specific values in the range [7,..,12]. Because the transmitted blocks in the CSS system are modulated, i.e., chosen from the predefined SFs available, though possibly unknown, the received signals can be classified according to the SF value. The majority of studies are concerned with SF assignments at the transmitter [115]; however, in practice, the signal is affected by many natural factors, such as fading, which may lead to miss-signal detection due to the large number of nodes transmitting with different SFs. Hence, SF recovery at the receiver helps improve the system's performance despite the presence of interference from other networks.

The unique feature of the LoRa system motivates us to design a new receiver using classification ML algorithms. It consists of sending six labels as a training phase for the applied supervised algorithms and then using the test phase to predict the transmitted SF. The classification algorithms only classify the received signals into a number of classes.

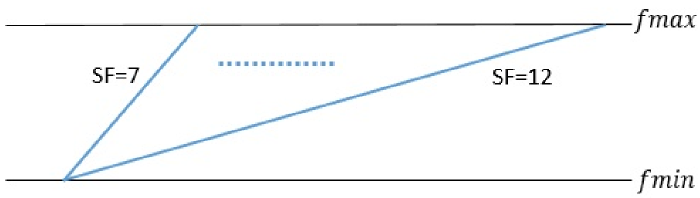


Figure 6.1: LoRa chirps with SF $\in [7, \ldots, 12]$

6.4. Classification Framework

The capability of the ML clustering algorithms has been demonstrated to be suitable for demodulation and detection of signals [120]. KNN and SVM are sub-fields of supervised ML which process the data for classification or regression. In typical clustering algorithms, the number of clusters and the centroid of the clusters that may be considered as a label are crucial and must be defined in the initial stages. In the context of this work, we define each cluster as a class that indicates a specific SF value. KNN and SVM are split into the training phase and the test phase; see Fig. 6.2.

6.4.1. SVM

In this work, the SVM algorithm is used in two stages. The training phase is the learning stage that is performed only once with labelled data, and the test stage contains the received signals. In the learning stage, the algorithm trains the machine to create a specific decision function to enable it to differentiate between the different types of transmitted SFs. In the testing stage, any newly received data is then classified according to the decision rules set in the training phase. In the training stage, the transmitter sends a sequence of labelled symbols, including a specific SF (e.g., 7) to the receiver. This enables the SVM algorithm at the receiver to determine those input features contained in the received signal that

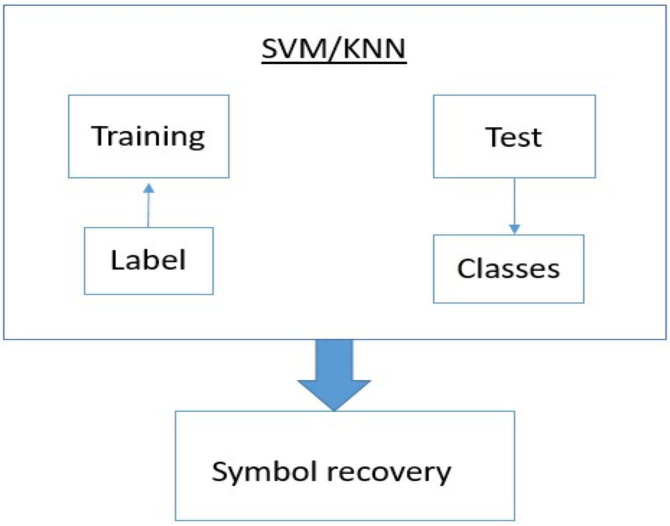


Figure 6.2: Classification framework

determines the specific SF. This process is carried out for each SF. The criterion or criteria obtained from the training data can then, subsequently, be applied to the test signals so that in the testing stage, any newly received data is classified according to the decision rules set in the training phase.
Using SVM, the receiver is trained to generate a hyperplane from the training data. According to the hyperplane, any subsequent received signal can be categorized into one of two groups, e.g., 7 or not 7; see Fig. 6.3. After completing the training phase, the corresponding transmitted symbol with a specific SF can be identified by making a hard decision based on the training undertaken; Algorithm 4 shows a summary of the SVM process.

6.4.1.1 Feature extraction

The training phase is the input of the SVM receiver. Consider the received training data with n number chirp symbols labelled with different SF. The ith training data is represented as $\mathbf{x}_i = 2^{\mathrm{SF}}$ samples/chips, which is modulated with defined SF in the range SF $\in [7, \ldots, 12]$. Each data stream is labelled with a binary message $l_i = \{-1, +1\}$, and the training data set represented as: $S = \{(\mathbf{x}_{i,1}, l_{i,1}), \ldots, (\mathbf{x}_{i,n}, l_{i,n})\}$.

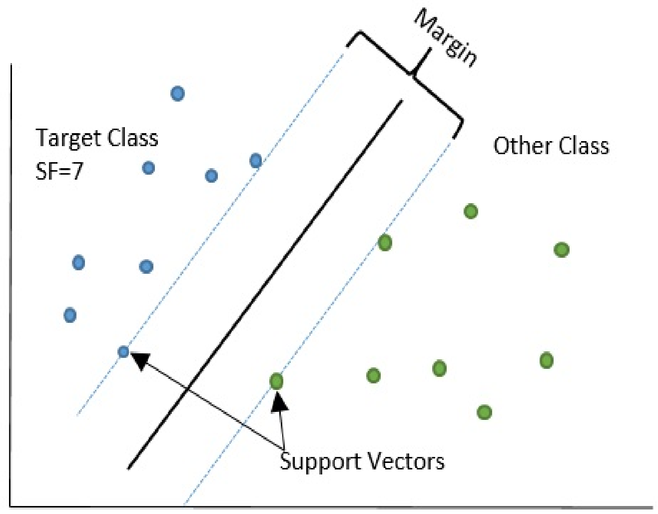


Figure 6.3: An example of a classification using SVM

6.4.1.2 The Training Phase

SVM was originally a two-class classifier that uses the hyperplanes with maximum margin to separate the input features into two classes. The largest margin passes between the two hyperplanes, which are used to separate the classes of the training data, see Fig. 6.3. SVM maximises the width of the margin which is determined by the nearest data points, known as support vectors. In LoRa, we consider the positive class as a true class with the transmitted SF and the negative class as a false class with any other SF. The two hyperplanes are expressed as

$$\begin{cases} \mathbf{v}^\top \mathbf{x}_i + \varrho \geq +1, & l_i = +1 \\ \mathbf{v}^\top \mathbf{x}_i + \varrho \leq -1, & l_i = -1 \end{cases} \tag{6.3}$$

where $\mathbf{x}_i$ is the input, $\mathbf{v}^\top$ is a normal non-zero vector that indicates the direction of the hyperplane, and ϱ is a scalar that indicates the distance between the hyperplane and the original point.
The optimal plane passes along the middle between the two hyperplanes and can be expressed as

$$\left\{\mathbf{x}_i \in S : \mathbf{v}^\top \mathbf{x}_i + \varrho = 0\right\} \tag{6.4}$$

where S is the training set. The distance between the hyperplane and the point in the training set $\mathbf{x}_i$ is given by

$$d_i = \frac{\left|\mathbf{v}^\top \mathbf{x}_i + \varrho\right|}{\|\mathbf{v}\|} \tag{6.5}$$

Thus, the maximum margin is defined as the closest distance of the point in the training set to the largest hyperplane

$$\rho = \min_{(\mathbf{x}_i, l_i)\in S} \frac{\left|\mathbf{v}^\top \mathbf{x}_i + \varrho\right|}{\|\mathbf{v}\|} \tag{6.6}$$

In the training phase, the decision function for the SVM that classifies the SF classes is constructed via

$$f(\mathbf{x}) = \sum_{n=1}^{n} l_i \alpha_i K(\mathbf{x}', \mathbf{x}_i) + \varrho \tag{6.7}$$

where $\alpha_i(\alpha_i \geq 0)$ is a Lagrangian constant, $K(\mathbf{x}', \mathbf{x}_i) = \phi^T(\mathbf{x}_i).\phi(\mathbf{x})$ is defined as a kernel function, where $\phi(\mathbf{x})$ maps the input training data into the higher dimensional feature space. The training stage is completed by solving the optimization problem for the penalty constant R which determines the trade-off between the training error and the largest possible margin of the decision function [121]

$$\begin{aligned}
&\min \frac{1}{2}\mathbf{v}^T\mathbf{v} + R\sum_{n=1}^{n} \xi_i \\
&\text{s.t.} \quad l_i \left(\mathbf{v}^\top \phi(\mathbf{x}_i) + \varrho\right) \geq 1 - \xi_i \\
&\xi_i \geq 0
\end{aligned} \tag{6.8}$$

where ξ_i is a slack variable applied for any non-linearly separable training data. At the end of the training phase, the training data is classified into two classes: $\alpha_i = 0$ and $\alpha_i \neq 0$. The training data with non-zero α_i are the support vectors that are used as a final decision variable indicating the exact SF class.

6.4.1.3 The Testing Phase

After the training phase, the task of the SVM receiver becomes a pattern classification problem. The receiver is ready to classify and estimate any newly received data by applying a hard decision based on Eq. (6.7). The newly received data are classified to determine the SF, and the decision is expressed as

$$\hat{\alpha} = \text{sign}(f(\mathbf{x})) = \text{sign}\left(\sum_{n=1}^{l} l_n \alpha_i K(\mathbf{y}, \mathbf{x}_i) + \varrho\right). \tag{6.9}$$

The target class using the voting strategy of (SVM) is determined by selecting the class with the highest classification function. To illustrate, if the class corresponding to SF=7 receives the most votes among all classes, it is identified as the target class.

Algorithm 4 SVM LoRa algorithm

1: **Input**: the training set defined SF $(\mathbf{x}_i, l_i)^n$.
2: Apply linear kernel function to obtain initial classification $K(\mathbf{x}', \mathbf{x}_i) = \phi^T(\mathbf{x}_i).\phi(\mathbf{x}')$.
3: Assign the decision function using Eq. (6.7)
4: Complete the training phase by solving the optimization problem Eq. (6.8).
5: Obtain new kernel for testing data.
6: **Output**: classifier as Eq. (6.7) decision function for the target SF class.

6.4.2. KNN

The KNN algorithm framework is used to identify, classify, and predict patterns for any type of data. In this work, the main goal of using KNN is to assign the right class of a given SF to the target point by identifying the nearest neighbours of the input data. The KNN predicts the correct class for the received data by finding the distance between each newly received data point and the training data. The process of the KNN algorithm starts by defining the number of nearest neighbours K_n. It then computes the Euclidean distance between the received data points in the test phase and the data points from the training phase. After that, the class labels of the maximum number of K_n entries are assigned as the prediction result of the testing phase, see Algorithm 5. Fig. 6.4 shows an example of a classification problem with three classes. Each class indicates a defined SF [7, 8, and 9] labelled as (C1, C2, C3), and $K_n = 3$. To make the correct decision regarding the class of the received data, KNN uses a voting method. Specifically, the received data will be assigned to the class with the maximum votes, which means the cluster with the most neighbours. In this example, $\mathbf{x}$ is assigned to cluster C1 because it has two neighbours there, but only one in cluster C2.

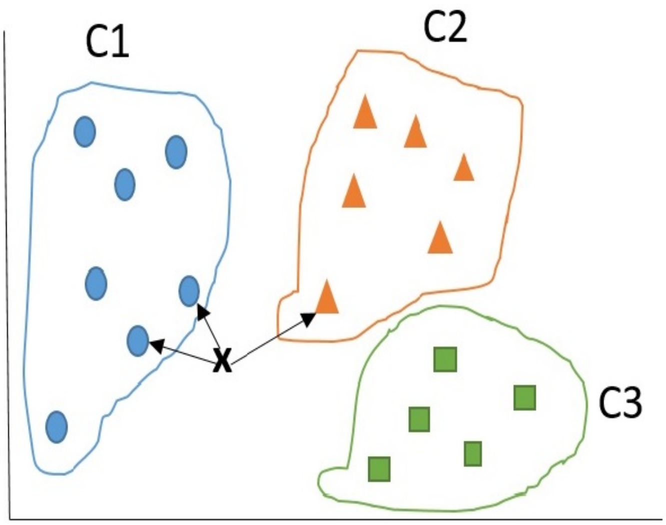


Figure 6.4: An example of KNN classification of 3 classes

Algorithm 5 KNN LoRa algorithm

Initialize training data (symbols with defined labels), K_n=4.
Input: Symbols.
for each input symbol **do**
Find the Euclidean distance between each of the received symbols and the labelled class from each training point.

$$D(\mathbf{x}_i, \mathbf{y}) = \sqrt{\sum_{n=1}^{n} (\mathbf{x}_i - \mathbf{y})^2}.$$

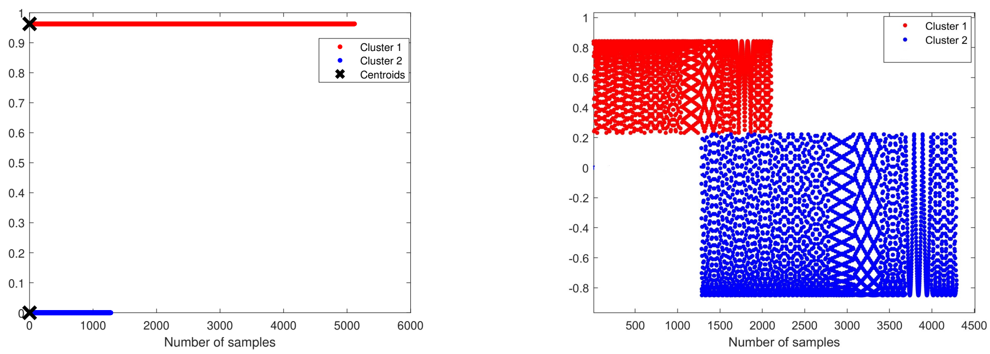
Assign the training data to the nearest class.
The most voted class is the target class

$$(\hat{\alpha}, \mathbf{x}_i) = \arg\min D(\mathbf{x}_i, \mathbf{y}).$$

end
Output: Target label.

6.4.3. Classification Output

In this section, we demonstrate the ability of clustering algorithms to classify the received data. As explained above, the classification algorithm divides the received data into groups with similar features. Fig. 6.5 illustrates the output clusters received at the receiver when two signals are transmitted, one with SF=7 and the other with SF=9. We see the receiver classifies the data into two clusters of different sizes. The smaller cluster refers to the transmitted data with SF=7, and the larger cluster is related to the transmitted data with SF=9.



(a) Without noise (b) With AWGN

Figure 6.5: Clustering outputs

6.5. Simulation Results

This section presents the relative performance results of KNN and SVM in a LoRa network. A Monte Carlo simulation is used to generate random data for both algorithms' training and test phases. The data was generated under different values of spreading factors [7, 8, 9 and 10]. In the SVM algorithm, there are three types of kernel functions [122]:

- Linear kernels $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x})^T.(\mathbf{x}')$.
- Polynomial kernels $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T.\mathbf{x}')$.
- Gaussian kernels $K(\mathbf{x}, \mathbf{x}') = \exp(-\left\|\mathbf{x}' - \mathbf{x}\right\|)^2/2\sigma^2$

For simplicity, the SVM kernel function is chosen to be linear [122].
Fig. 6.6 shows the classification error rate performance of a KNN and a SVM receiver using SF=7. Evidently, the SVM algorithm performs significantly better than KNN, making it the preferred choice for LoRa SF classification. When classifying, the SVM algorithm maximises the margin with the optimal plane, and this helps to alleviate the effects of noise and overfitting. In contrast, KNN assigns

labels and classifies the data based on the most voted K_n neighbours, making it sensitive to noisy data.
The KNN algorithm is highly dependent on the number of K_n neighbours. Choosing a large value of K_n will necessitate significant computing power, while a small K_n will affect the outliers of the clusters and may cause misclassification. In this work, based on the number of classes, we use the value K_n=4. Fig. 6.7 presents the result for the classification error rate for the KNN algorithm with different values of SF. The figure indicates better performance with a lower value of the SF. When we increase the SF, the performance of the classification is reduced due to the increasing amount of data transmitted per symbol with an increase in the value of SF.
Fig. 6.8 presents the confusion matrices showing SVM and KNN performance accuracy for SF values 7, 8, and 9. It demonstrates the classification accuracy between the predicted and the actual classes, whether the transmitted SF matched the received SF. The figure indicates the achieved accuracy is 83.87 % with SVM, while KNN achieved 80.77 %. In both algorithms, the best classification was achieved when the SF value was lowest at SF=7, and it was worse when the SF increased to SF=9. An SVM receiver can correctly predict 90% of the class with SF=7 but with SF=9, 20% of the data are misclassified. The corresponding figures for the KNN receiver are 85.7% and 77.8%.
SVM and KNN classification algorithms exhibit contrasting behaviour to the classical error rate, which aligns with the expectation. These classification algorithms work well with small data despite the presence of noise. However, as the data set expands with an increasing SF value and the complexity of the classification process grows, errors increase sharply with increase in SNR.

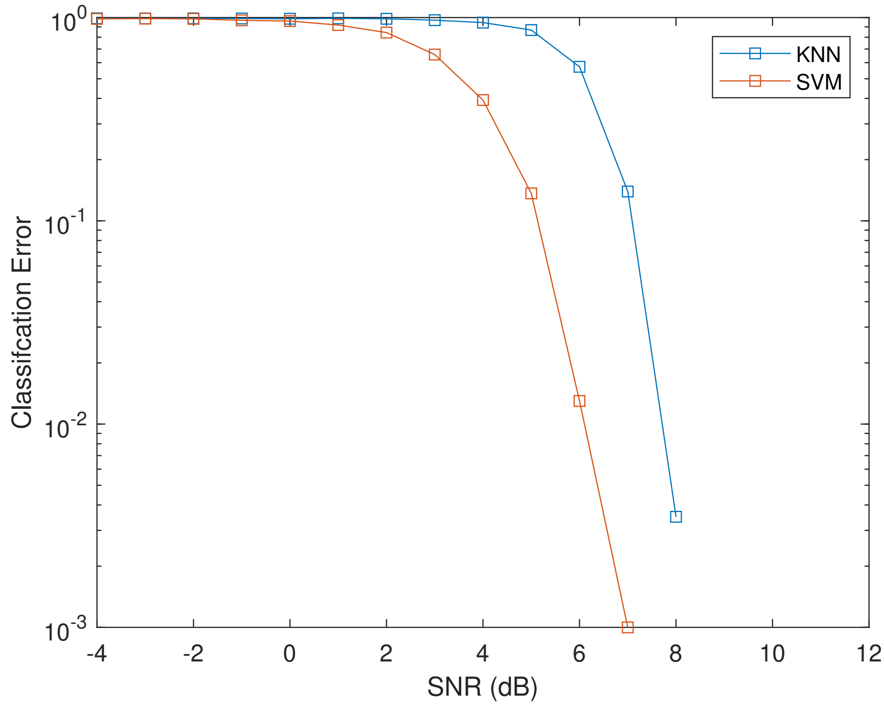


Figure 6.6: Classification error vs SNR for SVM and KNN (SF=7).

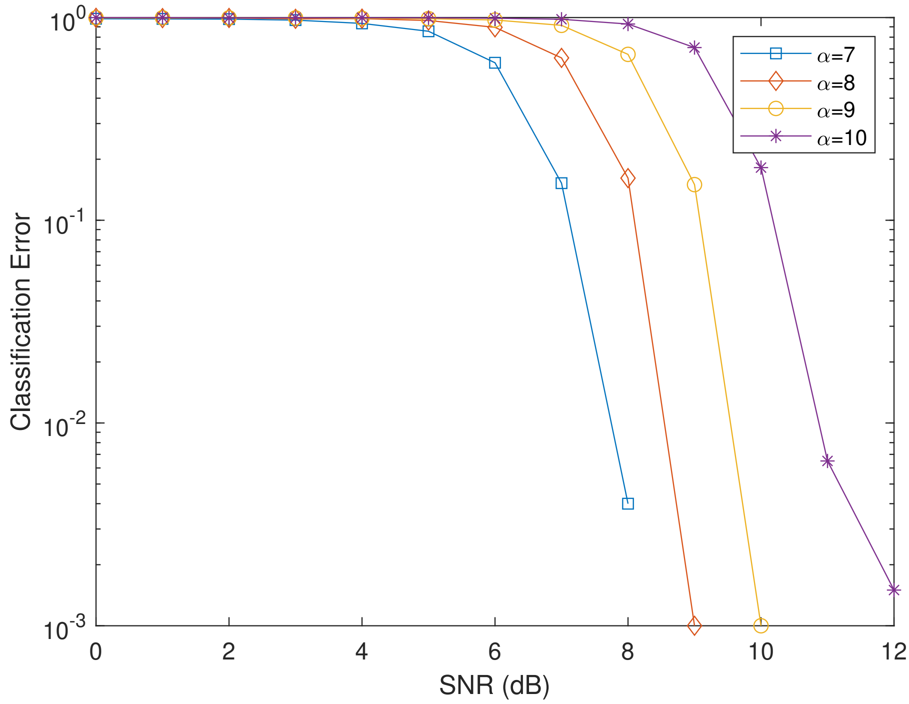
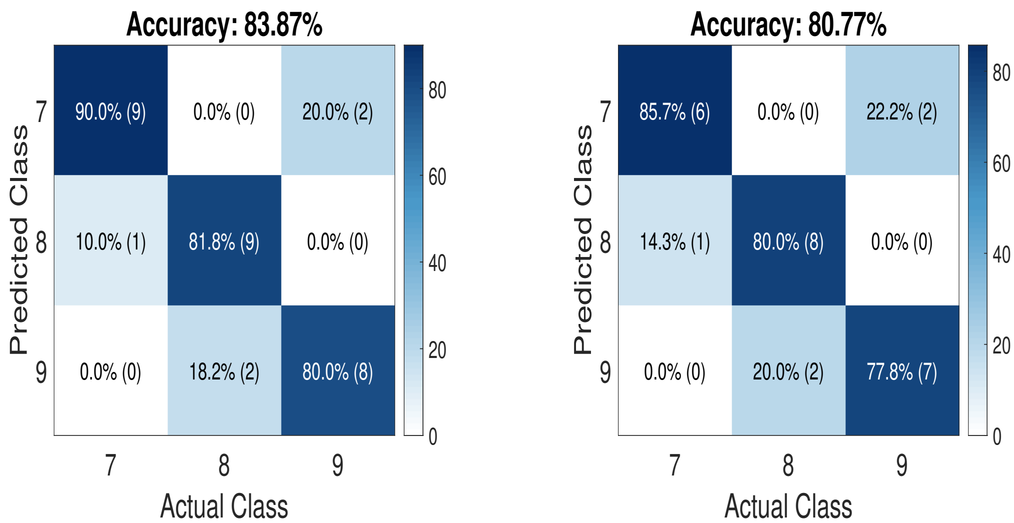


Figure 6.7: Classification error vs SNR for KNN with values different SF.



(a) SVM

(b) KNN

Figure 6.8: Confusion matrix performance accuracy of SVM and KNN.

6.6. Chapter Summary

In LoRa, each transmitted signal must have a defined SF value within the range [7, ..., 12], with the spreading factor playing a significant role in determining, e.g., the interference. This chapter introduced KNN and SVM supervised learning algorithms for SF classification and SF recovery. Each value of SF is considered as a class known to the receiver; thus, the transmitted signal can be directly assigned to the proper class. Both algorithms show good performance for the classification error rate. However, the performance of SVM was found to be better than KNN for all values of SF. Confusion matrices showed the relative effectiveness of KNN and SVM, indicating the relative accuracy of the algorithms at the receiver. It is again pointed out that SVM has an overall accuracy of 83.87 %, while KNN has an overall accuracy of 80.77 %. Both KNN and SVM showed their greatest accuracy for the lowest value of SF, 90.0%, and 85.7%, respectively.

Chapter 7

Conclusions and Future Work

This chapter presents the conclusions and the main results of this research. Following that are some suggestions for future research work.

7.1. Conclusions

The work in this thesis focused on improving the performance of the short packet communication system using machine learning techniques. This required a through investigation to provide an accurate evaluation error probability of SPC in the presence of interference, the application of different sparse recovery algorithms and ML applications to SPC and LoRa.
Chapter 2 comprehensively outlines the fundamental concepts and principles of SPC, clearly differentiating it from long-packet communication. The unique challenges and factors specific to short packets, including the influence of metadata size and the importance of optimizing overhead design, were emphasized. In addition, a foundational understanding of the Support Vector Machines (SVC), K-nearest Neighbour (KNN) and Expectation Maximization (EM) algorithms employed in this thesis was provided through a discussion of various ML algorithms. The LoRa system was also presented, including the basics of LoRa modulation, the distinctive parameters involved, and a discussion of different types of interference encountered in LoRa.
In Chapter 3, a new closed form to accurately evaluate packet error probability within SPC systems in an interference environment was derived. This evaluation

took into account the unique characteristics of SPC systems, such as packet length, and employed an equivalent signal-to-interference plus noise ratio to derive an exact expression for the packet error probability. Firstly, the accuracy of the expression was validated by Monte Carlo simulation. Then, the results illustrated to show the impact of blocklength, coding rate and the number of interference. Finally, the system throughput was investigated under two different scenarios: the binomial and Poisson distribution.

In Chapter 4, the performance of different sparse recovery algorithms was explored. We introduced two new algorithms, Compressive Sampling Matching Pursuit (CoSaMP) and Stagewise Orthogonal Matching Pursuit (St-OMP), to recover the non-zero elements in a sparse vector coding scheme. The performance of these algorithms was compared with Multipath Matching Pursuit (MMP) and Orthogonal Matching Pursuit (OMP). To assess their effectiveness, we evaluated recovery error, recovery time, covariance, and block error rate. Furthermore, we introduced a performance analysis of Symbol Error Rate (SER) using pairwise error probability. Of the four algorithms, MMP algorithm demonstrated the lowest errors in recovering sparse elements among the four algorithms. On the other hand, OMP proved to be the fastest algorithm in terms of recovery time. While the MMP algorithm exhibited superior performance compared to the other three algorithms, it is crucial to consider time constraints, especially in scenarios involving extensive machine-to-machine communication and the standards for Ultra-reliable and Low Latency Communication systems.

In Chapter 5, several ML approaches were introduced for symbol classification and detection in SPC systems. These approaches included MC-GMM, KNN, and SVM. Compared to the conventional system, implementing ML algorithms exhibited superior performance. Among the three approaches, KNN and SVM showed slightly better performance than the MC-GMM algorithm, primarily because they are supervised learning methods. The Label Assisted Transmission method was employed to address the overhead associated with short packets. This method leverages signal properties, such as equal probability among received clusters, to reduce the number of transmitted labels. Additionally, the introduction of Silhouette Analysis provided valuable insights into optimizing the optimal number of clusters in different noise conditions. This analysis aided in improving the accuracy and effectiveness of the clustering process.

Chapter 6 of this thesis highlights the important role of the spreading factor (SF) in the LoRa system. In LoRa, each transmitted signal requires a defined SF value within the range of 7 to 12. This chapter introduced supervised clustering algorithms, namely KNN and SVM, for SF classification and SF recovery. In this approach, each SF value is treated as a distinct class known to the receiver. Based on the results, both KNN and SVM demonstrate favourable performance in terms of classification error rate. However, it is worth noting that SVM outperforms KNN in this regard, as evidenced by the comparative confusion matrices. These matrices depict the relative effectiveness of these algorithms at the receiver. Specifically, SVM achieved an overall accuracy of 83.87%, while KNN attained an overall accuracy of 80.77%. Notably, both algorithms exhibited their highest accuracy for the lowest SF value.

7.2. Future Work

This thesis has extensively investigated applications that could significantly enhance the overall performance of SPC. Among these, the most promising approach was the implementation of ML algorithms, specifically in the context of SPC. However, there are many potential areas for future research on this topic, such as those presented below.

7.2.1. Interference Cancellation using Clustering Algorithm

Non-orthogonal Multiple Access (NOMA) allows multiple users to share the same frequency bands for improved spectral efficiency, but it leads to the problem of multi-interference issues [123]. We focused on the challenge of cancelling interference from multiple sources at the receiver of a secondary user in a NOMA system. In [124], the authors adopted an iterative receiver approach to cancel the interference from primary user transmitters and recover the secondary user signal using the K-means algorithm considering infinite blocklength. However, this algorithm cannot accurately estimate the superposition of interference signals from different primary user transmitters. Thus, for unknown channel state information (CSI) between the primary user transmitters and the secondary user receiver link, the provision of proper initializations for any clustering module based on K-means or Gaussian

Mixture Model-Expectation Maximization (GMM-EM) is difficult.

To overcome this issue, we recommend investigating the replacement of the Kmeans/GMM-EM algorithms with the Affinity Propagation (AP) algorithm, which offers a notable advantage over K-means and GMM-EM algorithms by not relying so heavily on the initialization of cluster centroids [125]. Instead, it assigns data points as exemplars, which serve as representatives of clusters. The algorithm calculates similarities between data points and updates the availability and responsibility matrices of the AP algorithm iteratively. As a result, the AP algorithm can accurately divide the interference signals into clusters, enhancing the overall performance of interference cancellation. Fig. 7.1, illustrate an example of the use of the AP algorithm.

Additionally, this method could be extended to cancel different types of interference in the LoRa system.

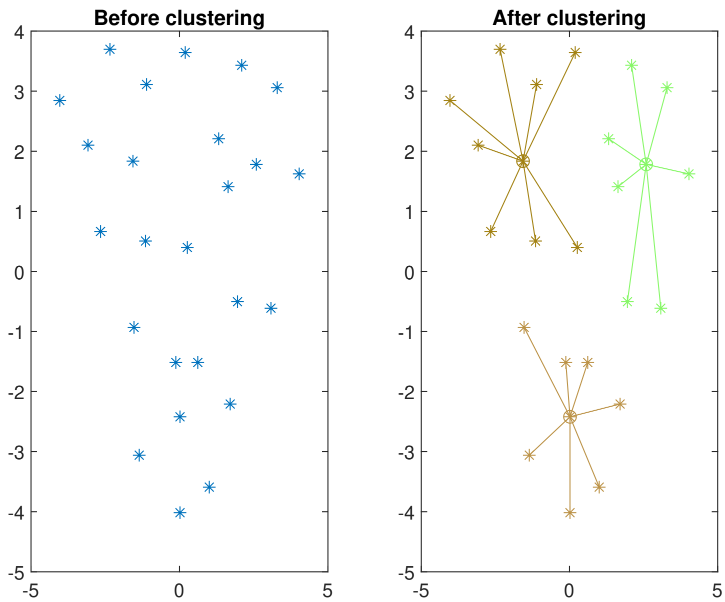


Figure 7.1: Example of Affinity Propagation algorithm output.

7.2.2. Sparse Recovery Using Machine Learning

The algorithms outlined in Chapter 4 are iterative in nature, which takes more computing time and can be computationally expensive. To address this, a machine learning-based approach such as a classification and deep learning algorithms could help enhance sparse signal recovery performance while reducing computational complexity. This solution is particularly beneficial for meeting the demanding requirements of URLLC. The decoding process (SVC) could be redefined as a multi-label classification task, where the non-zero positions of the sparse vector represent the labels from the SVC encoded vector to be classified. Also, the work can be extended to further examine the performance of SVC when the non-zero elements are more than two.

7.2.3. Short Packet Communication with UAV

Recently, Unmanned Aerial Vehicle (UAV or drone) based communication systems have attracted substantial attention due to their wide-ranging applications such as URLLC and IoT applications [126–129]. In the context of SPCs with UAVs, there are several promising avenues for future research and development. Firstly, exploring advanced coding schemes specifically tailored for short packet transmissions in UAV networks would be valuable. This could involve investigating innovative techniques for error correction and reducing overheads techniques, efficient channel coding schemes, and adaptive modulation and coding strategies to enhance the reliability and throughput of data transmission.

In addition, optimizing resource allocation algorithms to maximize the utilization of limited UAV resources, such as bandwidth and energy, while ensuring low-latency and reliable communication, could be a focus. Moreover, employing machine learning techniques to enable intelligent decision-making in UAV networks, such as dynamic packet routing, interference management, and adaptive transmission schemes, hold great potential for improving overall system performance. Finally, investigating the impact of various environmental factors, such as fading and interference, on short packet communication in UAV networks and developing robust techniques to mitigate their effects would be an important area of future research.

7.2.4. Short Packet Communication with IRS

Recently, there has been increasing interest and attention on studying the intelligent reflecting surface (IRS) [130–132]. Exploring the potential of using SPC with the IRS to achieve URLLC needs further investigation.

7.2.5. Other Extensions

Some further recommended extensions based directly on this thesis could be:

- The work in Chapter 3 assumed equal power and QPSK modulation. This work could be extended to unequal power and other types of modulation.
- The analysis provided in Chapter 3 could be applied to the LoRa system to improve system performance in an interference environment.
- In Chapter 5, the LAT method works for QPSK. It would be interesting if higher modulation was investigated regarding how to send one label and reconstruct the other labels without increasing the packet overhead.
- The work in Chapter 6 could be extended to examine the interference of LoRa signals due to signal types, such as Sigfox.

Bibliography

[1] Y. Polyanskiy, "Spectre: Short packet communication toolbox." *available at github. com/yp-mit/spectre*, 2016.

[2] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel *et al.*, "Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[3] D. 3GPP, "Study on new radio access technology physical layer aspects," *Technical Report (TR) 38.802, V14. 2.0*, 2017.

[4] A. Aijaz and M. Sooriyabandara, "The tactile internet for industries: A review," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 414–435, 2018.

[5] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick *et al.*, "6g white paper on machine learning in wireless communication networks," *arXiv preprint arXiv:2004.13875*, 2020.

[6] K. I. Pedersen, G. Berardinelli, F. Frederiksen, P. Mogensen, and A. Szufarska, "A flexible 5g frame structure design for frequency-division duplex cases," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 53–59, 2016.

[7] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.

[8] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[9] A. Goldsmith, *Wireless communications.* Cambridge university press, 2005.

[10] L. H. Ozarow, S. Shamai, and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE transactions on Vehicular Technology*, vol. 43, no. 2, pp. 359–378, 1994.

[11] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[12] C. Li, "Short-packet communications: Transmission strategies and power control policies design," Ph.D. dissertation, The Australian National University (Australia), 2021.

[13] M. Abo-Zahhad, M. Farrag, and A. Ali, "Optimization of transmitted power and modulation level for minimizing energy consumption in wireless sensor networks," *Wireless Personal Communications*, vol. 96, pp. 4047–4062, 2017.

[14] J. R. Pierce, *An introduction to information theory: symbols, signals and noise.* Courier Corporation, 2012.

[15] G. Wang, J. C. Ye, and B. De Man, "Deep learning for tomographic image reconstruction," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 737–748, 2020.

[16] A. Bandi and S. Yalamarthi, "Towards artificial intelligence empowered security and privacy issues in 6g communications," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS).* IEEE, 2022, pp. 372–378.

[17] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.

[18] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification," in *2018 Second international conference on intelligent computing and control systems (ICICCS).* IEEE, 2018, pp. 945–949.

[19] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.

[20] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya, "Optimization problems for machine learning: A survey," *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828, 2021.

[21] N.-E. Ayat, M. Cheriet, and C. Y. Suen, "Automatic model selection for the optimization of svm kernels," *Pattern Recognition*, vol. 38, no. 10, pp. 1733–1745, 2005.

[22] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 international conference on intelligent computing and control systems (ICCS)*. IEEE, 2019, pp. 1255–1260.

[23] A. Pandey and A. Jain, "Comparative analysis of knn algorithm using various normalization techniques," *International Journal of Computer Network and Information Security*, vol. 11, no. 11, p. 36, 2017.

[24] É. O. Rodrigues, "Combining minkowski and chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier," *Pattern Recognition Letters*, vol. 110, pp. 66–71, 2018.

[25] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[26] H. Saadatfar, S. Khosravi, J. H. Joloudari, A. Mosavi, and S. Shamshirband, "A new k-nearest neighbors classifier for big data based on efficient data pruning," *Mathematics*, vol. 8, no. 2, p. 286, 2020.

[27] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE access*, vol. 7, pp. 65 579–65 615, 2019.

[28] J. Mairal, "Sparse coding for machine learning, image processing and computer vision," Ph.D. dissertation, École normale supérieure de Cachan-ENS Cachan, 2010.

[29] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith, "Deep learning, sparse coding, and svm for melanoma recognition in dermoscopy images," in *International workshop on machine learning in medical imaging.* Springer, 2015, pp. 118–126.

[30] F. Wang, H.-H. Franco-Penya, J. D. Kelleher, J. Pugh, and R. Ross, "An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity," in *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13.* Springer, 2017, pp. 291–305.

[31] A. Dudek, "Silhouette index as clustering evaluation tool," in *Classification and Data Analysis: Theory and Applications 28.* Springer, 2020, pp. 19–33.

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

[33] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[34] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5g mimo data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA).* IEEE, 2018, pp. 1–9.

[35] S. Aldossari and K.-C. Chen, "Predicting the path loss of wireless channel models using machine learning techniques in mmwave urban communications," in *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC).* IEEE, 2019, pp. 1–6.

[36] C.-H. Lin, W.-C. Kao, S.-Q. Zhan, and T.-S. Lee, "Bsnet: A deep learning-based beam selection method for mmwave communications," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall).* IEEE, 2019, pp. 1–6.

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

[38] H. Ye and G. Y. Li, “Deep reinforcement learning for resource allocation in v2v communications,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[39] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[40] S. Devalal and A. Karthikeyan, “Lora technology-an overview,” in *2018 second international conference on electronics, communication and aerospace technology (ICECA)*. IEEE, 2018, pp. 284–290.

[41] M. J. Faber, K. M. van der Zwaag, W. G. V. dos Santos, H. R. d. O. Rocha, M. E. Segatto, and J. A. Silva, “A theoretical and experimental evaluation on the performance of lora technology,” *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9480–9489, 2020.

[42] T. Elshabrawy and J. Robert, “Interleaved chirp spreading lora-based modulation,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3855–3863, 2019.

[43] Semtech, “Semtech SX1272 , https://www.semtech.com/products/wireless-rf/lora-connect/sx1272,” accessed: 30- 06-2023.

[44] Semtech76, “Semtech SX1276, https://www.fr.semtech.com/products/wireless-rf/lora-connect/sx1276,” accessed: 30- 06-2023.

[45] S. Kim, H. Lee, and S. Jeon, “An adaptive spreading factor selection scheme for a single channel lora modem,” *Sensors*, vol. 20, no. 4, p. 1008, 2020.

[46] T. Yatagan and S. Oktug, “Smart spreading factor assignment for lorawans,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–7.

[47] M. Bor and U. Roedig, "Lora transmission parameter selection," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2017, pp. 27–34.

[48] P. Kulkarni, Q. O. A. Hakim, and A. Lakas, "Experimental evaluation of a campus-deployed iot network using lora," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2803–2811, 2019.

[49] K. C. Wiklundh, "Understanding the iot technology lora and its interference vulnerability," in *2019 International Symposium on Electromagnetic Compatibility-EMC EUROPE*. IEEE, 2019, pp. 533–538.

[50] L. Vangelista, "Frequency shift chirp modulation: The lora modulation," *IEEE signal processing letters*, vol. 24, no. 12, pp. 1818–1821, 2017.

[51] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[52] T. Elshabrawy and J. Robert, "Closed-form approximation of lora modulation ber performance," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1778–1781, 2018.

[53] J. Shi, D. Mu, and M. Sha, "Lorabee: Cross-technology communication from lora to zigbee via payload encoding," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–11.

[54] ——, "Enabling cross-technology communication from lora to zigbee via payload encoding in sub-1 ghz bands," *ACM Transactions on Sensor Networks (TOSN)*, vol. 18, no. 1, pp. 1–26, 2021.

[55] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" pp. 59–67, 2016.

[56] F. Rancy, "Imt for 2020 and beyond," *5G Outlook-Innovations and Applications*, p. 69, 2016.

[57] J.-H. Park and D.-J. Park, "A new power allocation method for parallel awgn channels in the finite block length regime," *IEEE communications letters*, vol. 16, no. 9, pp. 1392–1395, 2012.

[58] G. Durisi, T. Koch, J. Östman, Y. Polyanskiy, and W. Yang, "Short-packet communications over multiple-antenna rayleigh-fading channels," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 618–629, 2015.

[59] Y. Gu, H. Chen, Y. Li, and B. Vucetic, "Ultra-reliable short-packet communications: Half-duplex or full-duplex relaying?" *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 348–351, 2017.

[60] X. Lai, Q. Zhang, and J. Qin, "Cooperative noma short-packet communications in flat rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6182–6186, 2019.

[61] J. Zheng, Q. Zhang, and J. Qin, "Average block error rate of downlink noma short-packet communication systems in nakagami-m fading channels," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1712–1716, 2019.

[62] X. Huang and N. Yang, "On the block error performance of short-packet non-orthogonal multiple access systems," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[63] T.-H. Vu, T.-V. Nguyen, D. B. da Costa, and S. Kim, "Intelligent reflecting surface-aided short-packet non-orthogonal multiple access systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4500–4505, 2022.

[64] J. Xu, L. Yuan, N. Yang, and Y. Guo, "Performance analysis of star-irs aided noma short-packet communications with statistical csi," *IEEE Transactions on Vehicular Technology*, 2023.

[65] Y. Chen, T. Zhang, Y. Zhang, B. Yu, and Y. Cai, "Relay-assisted secure short-packet communications in cognitive internet of things," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[66] C. D. Ho, T.-V. Nguyen, T. Huynh-The, T.-T. Nguyen, D. B. da Costa, and B. An, "Short-packet communications in wireless-powered cognitive iot networks: Performance analysis and deep learning evaluation," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2894–2899, 2021.

[67] P. Raut, K. Singh, W.-J. Huang, C.-P. Li, and M.-S. Alouini, "Reliability analysis of fd-enabled multi-uav systems with short-packet communication," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 191–12 196, 2021.

[68] J. D. Kumar, P. Mohapatra, and N. Pappas, "Short packet communication over a two-user z-interference channel with rayleigh fading," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4746–4751.

[69] T.-H. Vu, T.-V. Nguyen, T.-T. Nguyen, V. N. Q. Bao, and S. Kim, "Short-packet communications in noma-cdrt iot networks with cochannel interference and imperfect sic," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5552–5557, 2022.

[70] R. K. Morrow and J. S. Lehnert, "Packet throughput in slotted aloha ds/ssma radio systems with random signature sequences," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1223–1230, 1992.

[71] K. A. Hamdi, L. Pap, and E. Alsusa, "Accurate evaluation of packet error probabilities considering bit-to-bit error dependence," in *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, vol. 5. IEEE, 2005, pp. 5–pp.

[72] K. A. Hamdi, "Packet-error probability analysis for fh-cdma unslotted packet networks," *IEEE transactions on communications*, vol. 51, no. 2, pp. 151–154, 2003.

[73] K. A. Hamdi and L. Pap, "A unified framework for interference analysis of noncoherent mfsk wireless communications," *IEEE transactions on communications*, vol. 58, no. 8, pp. 2333–2344, 2010.

[74] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6g frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[75] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5g downlink: Physical layer aspects," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, 2018.

[76] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1527–1550, 2017.

[77] A. Hoglund, X. Lin, O. Liberg, A. Behravan, E. A. Yavuz, M. Van Der Zee, Y. Sui, T. Tirronen, A. Ratilainen, and D. Eriksson, "Overview of 3gpp release 14 enhanced nb-iot," *IEEE network*, vol. 31, no. 6, pp. 16–22, 2017.

[78] R. Baraniuk and P. Steeghs, "Compressive radar imaging," in *2007 IEEE radar conference*. IEEE, 2007, pp. 128–133.

[79] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[80] A. Massa, P. Rocca, and G. Oliveri, "Compressive sensing in electromagnetics-a review," *IEEE Antennas and Propagation Magazine*, vol. 57, no. 1, pp. 224–238, 2015.

[81] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 164–174, 2010.

[82] T. Wimalajeewa and P. K. Varshney, "Compressive sensing based signal processing in wireless sensor networks: A survey," *arXiv preprint arXiv:1709.10401*, 2017.

[83] X. Zhang, G. Han, D. Zhang, B. Shim, and D. Zhang, "Sparse vector coding-based superimposed transmission for short packet urllc," in *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2021, pp. 1–6.

[84] W. Kim, S. K. Bandari, and B. Shim, "Enhanced sparse vector coding for ultra-reliable and low latency communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5698–5702, 2020.

[85] H. Ji, W. Kim, and B. Shim, "Pilot-less sparse vector coding for short packet transmission," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1036–1039, 2019.

[86] H. Ji, S. Park, and B. Shim, "Sparse vector coding for ultra reliable and low latency communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6693–6706, 2018.

[87] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and computational harmonic analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[88] E. C. Marques, N. Maciel, L. Naviner, H. Cai, and J. Yang, "A review of sparse recovery algorithms," *IEEE access*, vol. 7, pp. 1300–1322, 2018.

[89] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2986–3001, 2014.

[90] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[91] Y. Zhang and G. Sun, "Stagewise arithmetic orthogonal matching pursuit," *International Journal of Wireless Information Networks*, vol. 25, pp. 221–228, 2018.

[92] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "Compressive sensing: Performance comparison of sparse recovery algorithms," in *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*. IEEE, 2017, pp. 1–7.

[93] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, "Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive mimo," *IEEE transactions on signal processing*, vol. 68, pp. 764–779, 2020.

[94] Y.-D. Huang, P. P. Liang, Q. Zhang, and Y.-C. Liang, "A machine learning approach to mimo communications," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[95] Q. Zhang, P. P. Liang, Y.-D. Huang, Y. Pei, and Y.-C. Liang, "Label-assisted transmission for short packet communications: A machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8846–8859, 2018.

[96] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint channel estimation and data detection in mimo-ofdm systems: A sparse bayesian learning approach," *IEEE Transactions on signal processing*, vol. 63, no. 20, pp. 5369–5382, 2015.

[97] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive mimo systems," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1228–1231, 2019.

[98] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, "Online extreme learning machine-based channel estimation and equalization for ofdm systems," *IEEE Communications Letters*, vol. 23, no. 7, pp. 1276–1279, 2019.

[99] H.-W. Liang, W.-H. Chung, and S.-Y. Kuo, "Coding-aided k-means clustering blind transceiver for space shift keying mimo systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 103–115, 2015.

[100] S. Leblanc and M. Kaneko, "Deep learning-based sub-6ghz/mmwave partitioning for short-packet communications," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[101] T. Ngo Hoang and K. Lee, "Short-packet urllcs for multihop mimo full-duplex relay networks: Analytical and deep-learning-based real-time evaluation," 2023.

[102] B. Lee, S. Park, D. J. Love, H. Ji, and B. Shim, "Packet structure and receiver design for low latency wireless communications with ultra-short packets," *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 796–807, 2017.

[103] M. Mousaei and B. Smida, "Optimizing pilot overhead for ultra-reliable short-packet transmission," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–5.

[104] G. C. Ferrante, J. Ostman, G. Durisi, and K. Kittichokechai, "Pilot-assisted short-packet transmission over multiantenna fading channels: A 5g case study," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–6.

[105] J. Wu, W. Kim, and B. Shim, "Pilot-less one-shot sparse coding for short packet-based machine-type communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9117–9120, 2020.

[106] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[107] L. V. Nguyen, D. H. Nguyen, and A. L. Swindlehurs, "Svm-based channel estimation and data detection for massive mimo systems with one-bit adcs," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[108] M. A. Kumar and M. Gopal, "Reduced one-against-all method for multiclass svm classification," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14 238–14 248, 2011.

[109] B. Sklar, "Digital communication system performance," in *Mobile Communications Handbook*. CRC Press Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300 . . . , 2017, pp. 333–354.

[110] K. Saied, "Quasi-cyclic short packet (qcsp) transmission for lot," Ph.D. dissertation, Université de Bretagne Sud, 2022.

[111] K. O. Adefemi Alimi, K. Ouahada, A. M. Abu-Mahfouz, and S. Rimer, "A survey on the security of low power wide area networks: Threats, challenges, and potential solutions," *Sensors*, vol. 20, no. 20, p. 5800, 2020.

[112] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "Resource allocation to massive internet of things in lorawans," *Sensors*, vol. 20, no. 9, p. 2645, 2020.

[113] L. Angrisani, M. D'Arco, C. Dassi, and A. Liccardo, "Lora signals classification through a cs-based method," in *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*. IEEE, 2018, pp. 1–5.

[114] C.-X. Wang, M. Di Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5g and beyond: Recent advances and future challenges," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 16–23, 2020.

[115] C. Bouras, A. Gkamas, S. A. Katsampiris Salgado, and N. Papachristos, "Spreading factor analysis for lora networks: A supervised learning approach," in *Trends and Applications in Information Systems and Technologies: Volume 1 9*. Springer, 2021, pp. 344–353.

[116] W. U. Khan, S. I. A. Elkarim, M. Elsherbini, O. Mohammed, O. Waqar, and B. M. Elhalawany, "Deep learning based joint collision detection and spreading factor allocation in lorawan," 2022.

[117] A. Shahid, J. Fontaine, M. Camelo, J. Haxhibeqiri, M. Saelens, Z. Khan, I. Moerman, and E. De Poorter, "A convolutional neural network approach for classification of lpwan technologies: Sigfox, lora and ieee 802.15. 4g," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–8.

[118] X. Tang, Y. Zhang, Y. Wang, D. Zhao, H. Li, and X. Zhao, "Performance analysis of preamble detection of lora system," in *2019 International Conference on Electronic Engineering and Informatics (EEI)*. IEEE, 2019, pp. 175–180.

[119] M. Potéreau, Y. Veyrac, and G. Ferre, "Leveraging lora spreading factor detection to enhance transmission efficiency," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[120] R. Hamdi and M. Qaraqe, "A novel index modulation based chirp spreading modulation scheme for wireless communications systems," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 01–05.

[121] L. Wang, *Support vector machines: theory and applications*. Springer Science & Business Media, 2005, vol. 177.

[122] Y. Hu, P. Wang, Z. Lin, M. Ding, and Y.-C. Liang, "Machine learning based signal detection for ambient backscatter communications," in *ICC 2019-2019*

IEEE International Conference on Communications (ICC). IEEE, 2019, pp. 1–6.

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

[124] Y. Liu, Z. He, X. Kuai, and X. Yuan, "Learning based interference cancellation for cognitive radio," in *2018 IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC)*. IEEE, 2018, pp. 1–5.

[125] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.

[126] Z. Chu, W. Hao, P. Xiao, and J. Shi, "Uav assisted spectrum sharing ultra-reliable and low-latency communications," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[127] N.-N. Dao, Q.-V. Pham, N. H. Tu, T. T. Thanh, V. N. Q. Bao, D. S. Lakew, and S. Cho, "Survey on aerial radio access networks: Toward a comprehensive 6g access infrastructure," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1193–1225, 2021.

[128] Y. Han, L. Liu, L. Duan, and R. Zhang, "Towards reliable uav swarm communication in d2d-enhanced cellular networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1567–1581, 2020.

[129] T. Bouzid, N. Chaib, M. L. Bensaad, and O. S. Oubbati, "5g network slicing with unmanned aerial vehicles: Taxonomy, survey, and future directions," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 3, p. e4721, 2023.

[130] A. M. Salhab and M. H. Samuh, "Accurate performance analysis of reconfigurable intelligent surfaces over rician fading channels," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1051–1055, 2021.

[131] D. Selimis, K. P. Peppas, G. C. Alexandropoulos, and F. I. Lazarakis, "On the performance analysis of ris-empowered communications over nakagami-m fading," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2191–2195, 2021.

[132] Z. Ding, L. Lv, F. Fang, O. A. Dobre, G. K. Karagiannidis, N. Al-Dhahir, R. Schober, and H. V. Poor, "A state-of-the-art survey on reconfigurable intelligent surface-assisted non-orthogonal multiple access networks," *Proceedings of the IEEE*, vol. 110, no. 9, pp. 1358–1379, 2022.