LEARNING AND PREDICTION USING RADIO AND NON-RADIO ATTRIBUTES IN WIRELESS SYSTEMS

By

TEJASHRI KUBER

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Narayan B. Mandayam

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2021

ABSTRACT OF THE DISSERTATION

Learning and Prediction using Radio and Non-Radio Attributes in Wireless Systems

by Tejashri Kuber

Dissertation Director:

Narayan B. Mandayam

Wireless technology and connectivity are spreading rapidly around the globe. The advancement of machine learning tools plays a major role in pushing the limits of wireless communication applications. With the volume, variety and velocity of rich datasets available, we are able to accurately model and predict different aspects of wireless systems which can help with efficient utilization of available resources.

We aim to harness radio (such as channel capacity and received signal constellation) and non-radio attributes (such as weather, busy period data from an open-source API) by applying machine learning algorithms to improve the overall wireless system.

In this dissertation, we start by working on a small scale network. We present the problem of jointly predicting modulation and the number of transmit antennas for a non-cooperative system or a dynamically operating cognitive system. We use a discrete-wavelet transform on the received complex samples to separate the different modulation classes. We then use the k-nearest neighbor approach coupled with k-means clustering to further classify the signals, utilizing the symmetry and relative distances of the constellation points. The performance of the classifiers at different stages of the algorithm demonstrates a high accuracy where the packets can be decoded. This is followed by predicting channel transition and the state of the environment to improve the existing transmit beamforming procedure by changing sounding times. The

experimental results using Software Defined Radios (SDRs) on the ORBIT testbed show more than 96% classification success of channel transition and a significant difference in error vector values for different types of channel variations.

We then progress to working on a larger scale, by employing deep learning techniques on a cellular network in a dense urban area using statistically significant non-radio features. We focus on using non-radio attributes and their impact in predicting cellular traffic. Prediction of user traffic in cellular networks is one of the promising ways to improve resource utilization among base stations. We consider traffic from neighboring cells and other non-cellular-traffic related attributes such as weather, busy period data from open-source API as features to augment the cellular traffic data and improve prediction. Specifically, we augment cellular traffic data from the City of Milan and its surroundings and we perform two types of analyses: (i) a one-step prediction or a point-by-point forecast of traffic and (ii) a trend analysis which is the forecast of traffic over an extended period. We compare the results with existing statistical methods such as auto-regression integrated moving averages (ARIMA) and exponential smoothing and observe gains in the trend analysis by providing the augmented data, whereas the one-step prediction is not much impacted.

Acknowledgements

There are a lot of people I would like to acknowledge for making my PhD dream come true.

Firstly, I would like to thank Prof. Narayan Mandayam, my advisor, for spending time and effort in shaping this thesis. His insightful advice on research ideas and presentation skills have been crucial to my work. In my future professional career, I hope to always live up to the high standards that he has set.

A big thank you to Ivan Seskar, for being my co-advisor. His open door policy and readiness to discuss research problems have helped me tremendously. Ivan's breadth and depth of knowledge and passion for all things wireless will always continue to astound, inspire and motivate me.

I am grateful to Prof. Dipankar Raychaudhuri, Prof. Zoran Gajic and Prof. Dola Saha for participating in my dissertation proposal and defense committees. Their thought-provoking questions and comments have been extremely valuable.

I would like to specially thank Prof. Dola Saha for helping me with the early part of my thesis. I am very grateful to her for all the Skype conversations, conference meetings and the encouragement to attend research-related events. Collaborating with her has been one of the highlights of my thesis.

During my time in grad school, I was fortunate enough to work with Prof. David Daut. He kindled my interest in digital communications and coached me for my qualifying examinations. He is very much missed.

I would like to acknowledge the help and support received from my colleagues at WINLAB. I wish to thank Prasanthi Maddala in particular, whose help for all the hardware-related experiments has been immense. I am also thankful to my friends in India and the US for always cheering me on.

Finally, I would like to thank my family. I wouldn't have been able to do this without their unwavering belief in me. This thesis is dedicated to them. My husband Anoop Krishnamurthy has been the source of my strength and support. I am indebted to my brother Karthik Kuber for his advice whenever I needed it. My parents K.S. Krishnamurthy and R. Indira have constantly provided care and encouragement. I am thankful to Shobha Krishnamurthy, Divya Dharwar and Riya Kuber for all their love.

Dedication

to the Kubers and the Krishnamurthys

Table of Contents

Abstract ii

Acknowledgements iv

Dedication vi

List of Figures xi

1. Introduction 1

1.1. Overall Outline 2

1.1.1. Scale of the system 3

Size 3

Time 4

1.2. Summary of Specific Contributions of the Dissertation 4

2. Blind Classification of MIMO Wireless Signals 6

2.1. Introduction 6

2.1.1. Motivation 6

2.1.2. Related Work 7

2.1.3. Contributions of this research 8

2.2. Signal and System Model 8

2.3. Interclass Modulation Recognition 9

2.4. Intraclass Modulation Recognition and Antenna Enumeration 10

2.4.1. Clustering 10

Challenges 12

2.4.2. k Nearest Neighbors 13

Same number of clusters 13

Number of Transmit Antennas in M-QAM 17
2.5. Results 17
Interclass Classification 18
Intraclass Classification: M-PSK family 18
Intraclass Classification: Same number of clusters 18
Intraclass Classification: M-QAM family 18
2.6. Chapter Summary 18

3. Predicting Channel Transition for MU-MIMO Beamforming 20
3.1. Introduction 20
3.2. Related Work 21
3.3. Motivation: Reducing Time Overhead 22
3.4. Proposed Protocol 23
3.4.1. Classifying the environment: CART Analysis 24
Capacity 24
Aggregate of mean-square first differences 25
3.4.2. Quantifying the different environments: Precoding and EVM . . 25
3.5. Methodology 26
Classification Tree 27
EVM Calculation 29
3.6. Hardware Setup 31
3.7. Implementation 33
3.7.1. Overall system 33
3.7.2. Modifications 33
3.8. MU-MIMO Experiments using Software Defined Radios 34
3.8.1. Physical Setup 37
3.8.2. Hardware Setup 39
3.8.3. Collecting the data 40
Transmission and Reception 40

Synchronization . . . 40
3.8.4. Transmitted Data . . . 41
3.8.5. Processing the Received Data . . . 41
3.8.6. Summary of Testbed Setup . . . 41
3.9. Chapter Summary . . . 42
3.9.1. Applications . . . 42

4. Traffic Prediction by Augmenting Cellular Data with Non-Cellular Attributes . . . 43
4.1. Introduction . . . 43
4.1.1. Recurrent Neural Networks . . . 45
4.2. Available Data . . . 46
4.3. Overall Model . . . 48
4.3.1. Sequential module . . . 49
4.3.2. Spatial features . . . 50
4.3.3. Auxiliary features: Non-cellular attributes . . . 51
4.4. Feature Selection . . . 52
4.4.1. Precipitation . . . 52
4.4.2. Google API - Busy Hours . . . 55
4.4.3. Neighboring Cells . . . 55
4.5. Point-by-point Analysis . . . 58
4.6. Trend Analysis . . . 59
4.7. Chapter Summary . . . 62

5. Conclusion and Future Work . . . 64
5.1. Impact of radio attributes on prediction . . . 64
5.1.1. Prediction of modulation and number of transmit antennas . . . 64
5.1.2. Prediction of channel transition and state of environment . . . 64
5.2. Impact of non-radio attributes on prediction . . . 65
5.2.1. Prediction of cellular traffic . . . 65

5.3. Future Work 65

References 66

List of Figures

1.1. Structure of the thesis 3

1.2. Local and global systems 3

2.1. Proposed Algorithm 10

2.2. Clustering 11

2.3. Interclass Classification 13

2.4. Classification of M-PSK with 2 and 4 antennas 14

2.5. Calculating number of antennas when modulation is known 14

2.6. Classification using kNN (Same number of clusters) 15

2.7. M-QAM: 1 and 2 antennas 15

2.8. M-QAM: 2 and 4 antennas 16

2.9. Predictor Importance for Classification between 8PSK-1 and BPSK-4 . 16

3.1. Forward and Reverse Channels 22

3.2. Sounding messages and BF reports in 802.11ac transmit beamforming protocol 23

3.3. Static Channels 23

3.4. Classification Tree 27

3.5. Capacities in bits/OFDM symbol 28

3.6. Aggregate of Mean-Square Differences 30

3.7. OctoClock-G 32

3.8. EVM in dB 35

3.9. 2x4 MIMO 37

3.10. Two tightly synchronized data files 37

3.11. Transmission and Reception with an Oracle 38

3.12. Transmitter details 39

3.13. Receiver details 39
4.1. Traffic demands 44
4.2. Recurrent Neural Networks 45
4.3. LSTM Networks 46
4.4. City of Milan and its surroundings 47
4.5. Internet Activity in Milan 48
4.6. Proposed Framework 48
4.7. Internet activity data for one week 49
4.8. Leveraging spatial features 50
4.9. Internet activity in neighboring cells for one week 50
4.10. Points of Interest 51
4.11. Feature: Precipitation 53
4.12. Weather impact on Cellular Traffic: Cell 4470 54
4.13. Weather impact on Cellular Traffic: Cell 5283 56
4.14. Google API 57
4.15. Spatial Correlation 58
4.16. Point-by-point Analysis 59
4.17. Trend Analysis: ARIMA 60
4.18. Trend Analysis: Only temporal data 61
4.19. Trend Analysis: Temporal data with additional features 61

Chapter 1

Introduction

Speed, consistency and security are the three factors influencing the improvement of wireless communication systems. The design and implementation of traditional communication systems are developed based on assumptions, theories and models. As the systems and networks grow more complex, existing communication protocols exhibit room for improvement.

With the growth of wireless systems, the collection and accessibility of wireless data have flourished as well. In addition to this, mining of data from extraneous sources has increased significantly. This extrinsic data can potentially be used in conjunction with wireless data and these large datasets bring with it opportunities to improve the present wireless systems and protocols.

Using these rich datasets, Machine Learning (ML) tools and their rapid advancement in recent times, have the capability to bridge the gap between the current and future wireless technologies. Therefore, research on applications of ML in wireless communications is experiencing significant enhancement [1, 2].

ML algorithms can be classified into supervised and unsupervised learning, based on the availability of labeled data. Both these techniques lend themselves very well to wireless systems applications [3–5].

Supervised learning is a machine learning task that aims to learn a mapping function from the input to the output, given a labeled dataset. The two important techniques in supervised learning are regression and classification. These include but are not limited to logistic regression, k-nearest neighbors, trees, neural networks and support vector machines. [6–9] represent some of the work on supervised learning algorithms in wireless systems.

Unsupervised learning algorithms only receive training data examples and have no labeled outputs, which are advantageous in scenarios when datasets are produced without targets. A major benefit of unsupervised learning is its accuracy at identifying subtle structure in datasets. Examples of unsupervised learning tasks include probability density estimation, denoising, and clustering. [10–13] focus on unsupervised learning solutions in the wireless communications domain.

Deep learning (DL) techniques fall under the umbrella of ML and the learning can be supervised, unsupervised or semi-supervised. DL is especially useful when the amount of training data is large. Research on DL applications in wireless communications has been massively growing in the last few years [14–19].

In this thesis, we focus on prediction and learning in wireless systems using both radio and non-radio attributes by applying ML algorithms to appropriate datasets. We begin by using supervised and unsupervised learning and move onto deep learning on wireless systems.

1.1 Overall Outline

We present three separate cases of applying machine learning to wireless communications systems to improve performance, as seen in figure 1.1. We focus on the impact of radio attributes such as channel capacity and received waveform characteristics on small-scale systems as well as the non-radio features such as call records, geographical locations and busy periods of popular locations in a dense urban area and their role in forecasting cellular traffic in a large wireless system. Augmenting existing records with non-radio attributes impacts the prediction positively as well as opens exciting areas of research of including and utilizing such non-traditional resources into the wireless system.

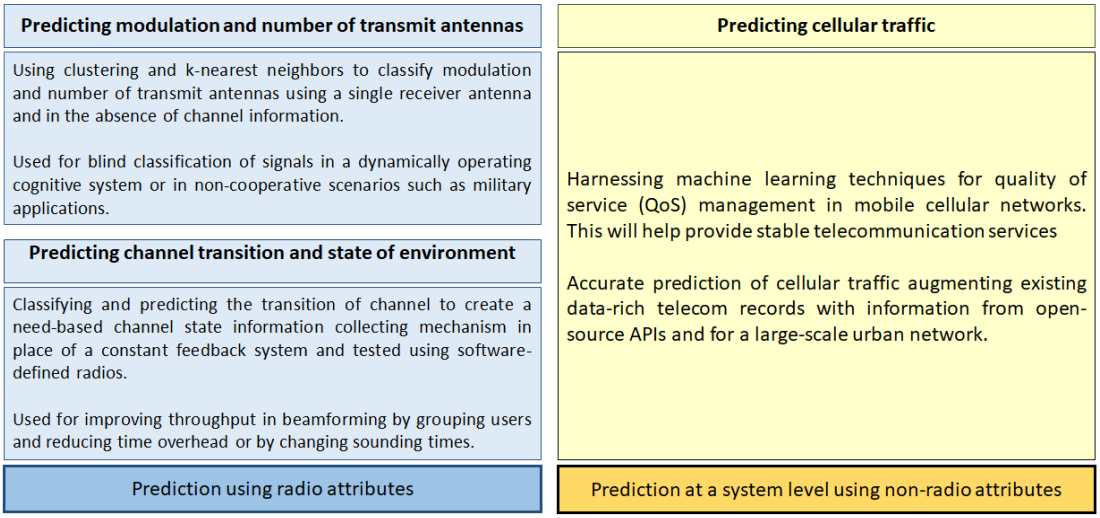


Figure 1.1: Structure of the thesis

1.1.1 Scale of the system

Size

The dissertation presents research on prediction at a local level, using individual radios. These are traditional wireless communications research problems benefiting from the use of ML to improve performance. We also present research on a non-traditional global problem of predicting cellular traffic using non-cellular attributes. This operates at a much larger scale, with cellular information of multiple transceivers at the base stations. This is summarized in figure 1.2.

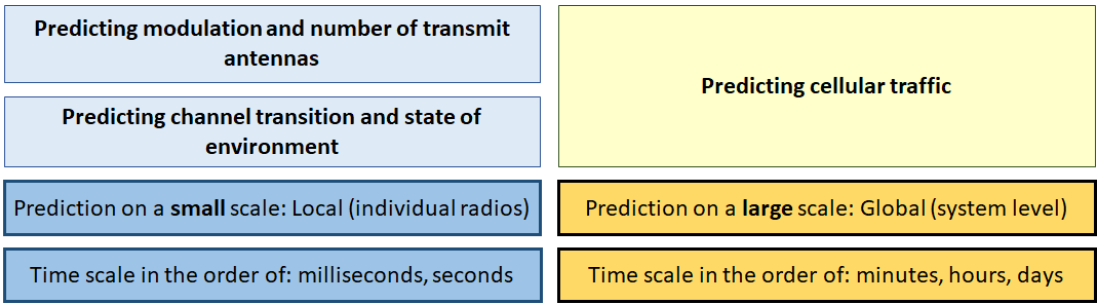


Figure 1.2: Local and global systems

Time

The different sections of the dissertation also vary in the time scale at which they operate as seen in figure 1.2. At the local radio level, it is imperative to perform the prediction within milliseconds or seconds. In the case of modulation and antenna enumeration, the time scale is in the order of the size of a few OFDM packets. Channel transition prediction should occur well within the coherence time of the channel.

The data available for the cellular traffic prediction is for longer periods of time. Specifically, Call Detail Records are available for a period of two months at ten-minute intervals. This gives us two types of forecasts, one in the order of minutes and a longer forecast period in the order of hours.

1.2 Summary of Specific Contributions of the Dissertation

Predicting modulation and number of transmit antennas [20]

Classification of MIMO wireless signals with no prior information of the number of transmitter antenna elements and channel(s) is of tremendous importance in both military and civilian applications. In this research, we present a novel generalized algorithm for classification of digital modulation combined with the recognition of the number of transmit antennas. Our algorithm is developed to rely on only one receiver equipped with a single antenna and does not require any prior channel knowledge. We have introduced a two-phase classifier system. In the first phase, we use a discrete-wavelet transform on the received complex samples to separate the different modulation classes. Following that, we use the k-nearest neighbor approach coupled with k-means clustering to further classify the signals, utilizing the symmetry and relative distances of the constellation points. The performance of the classifiers at different stages of the algorithm demonstrates a high accuracy in an SNR regime where the packets can be decoded.

Predicting channel transition and state of environment [21, 22]

Legacy beamforming procedure for multiuser MIMO (MU-MIMO) requires periodic explicit feedback mechanism for channel estimation resulting in high control overhead

with increasing number of users. Often, these procedures are redundant, especially if the channel remains static between two successive measurements. Projected increase in number of users in emerging 5G network, necessitates us to rethink the channel sounding technique to reduce overhead as well as increase aggregate throughput of the network. In this work, we propose to use changes in channel to trigger the channel assessment procedure at Access Point (AP). This technique is generic enough to be used either in the AP of IEEE 802.11ac or in the eNodeB of LTE systems. The experimental results using Software Defined Radios (SDRs) on the ORBIT testbed show more than 96% classification success of channel transition and a significant difference in error vector values for different types of channel variations. This leads to a need-based CSI collection, which will improve aggregate throughput of the network.

Traffic prediction by augmenting cellular data with non-cellular attributes [23]

Prediction of user traffic in cellular networks is one of the promising ways to improve resource utilization among base stations. In this study, we employ deep learning techniques, specifically a long-short-term memory module to forecast cellular traffic. We consider traffic from neighboring cells and other non-cellular traffic-related attributes such as weather, busy period data from open-source API as features to augment the cellular traffic data and improve prediction. Specifically, we augment cellular traffic data from the City of Milan and its surroundings and we perform two types of analyses: (i) a one-step prediction or a point-by-point forecast of traffic and (ii) a trend analysis which is the forecast of traffic over an extended period. We compare the results with existing statistical methods such as auto-regression integrated moving averages (ARIMA) and exponential smoothing and observe gains in the trend analysis by providing the augmented data, whereas the one-step prediction is not much impacted.

Chapter 2

Blind Classification of MIMO Wireless Signals

2.1 Introduction

2.1.1 Motivation

Blind classification is a technique that automatically classifies the received signal into different categories for signal detection and demodulation, with no or little prior knowledge. Intelligent receivers have a classification step before processing the signal. Such a receiver can improve transmission efficiency by reducing the overhead. Counting the number of antennas, which are transmitting simultaneously, or antenna enumeration, can especially be of use in Software Defined Radios (SDRs) where the secondary users can use the knowledge of the primary users' transmit antennas to set their transmit power and minimize interference in beamforming [24]. There are two main scenarios in which a classifier would be of paramount importance.

- Non-cooperative system: A receiver in military applications performing electronic surveillance will be required to locate and identify a signal with no knowledge of the transmitted signal and the channel.

- Signals operating in multiple frequency bands: To improve spectrum utilization, cognitive radios are programmed and operated dynamically, giving little time at the receiver for complete signal knowledge before demodulation. Blind classifiers are also useful for the SDRs, which cope with a variety of communication systems.

There is a large body of work which deals with Single-Input Single-Output (SISO)

modulation classification (MC) algorithms. These are mainly classified into likelihood-based and feature-based (FB) and are explored in [25]. FB algorithms have the advantage of being simple to implement and deploy and have near-optimal performance. Many of the FB algorithms rely on higher order statistics and cyclostationary moments. Signal classification using higher-order moments as baseline features has been extensively studied in [26]. Blind classification of Multiple-Input Multiple-Output (MIMO) using Machine Learning (ML) to complement the existing FB techniques is a recent development. Intelligent receivers for MIMO require [27] the following: the receiver needs to blindly estimate the number of transmit antennas, the space-time coding, the channel and the modulation. As MC, detection of number of transmit antennas, and classification of STBC algorithms are all inter-related, [28] makes a key observation that joint classification is one of the important steps to be taken in this field.

2.1.2 Related Work

Most of the prior literature pertaining to this subject can be divided into: MC, antenna enumeration and joint classification. MIMO MC is considered especially challenging due to the interference between received signals and the multiplicity of unknown channels [29,30]. Wavelets are also used as FB MC, [31] has used the discrete wavelet transform for classification with cross-correlation with a modest success rate. ML based wavelet classifiers are also created in [32], but require a training phase before testing. Clustering using the constellation shape for MC has been studied in [33–35], but these works have mainly relied on matching the received signal to a library of templates. Many of the MIMO classifiers require more than one receive antenna, such as in [36]. Most of the research available in the area of antenna enumeration requires prior knowledge of the modulation [37], and in many cases, pilot patterns and preamble sequences [24, 38]. In the case of joint classifiers, the important first step is to estimate the channel, [39] which can be expensive. Many classifiers also tackle the issue of space-time-block coding (STBC), instead of MC and antenna computation, which is out of scope of this work.

2.1.3 Contributions of this research

- Joint detection of number of transmit antennas and type of data modulation in the absence of channel information and without added computation time and complexity of channel estimation
- Only one receive antenna is used
- No training required for incoming data
- Easily scalable simple two-step approach, suitable for non-cooperative and/or real-time scenarios

2.2 Signal and System Model

In this work, we consider a MIMO communication system with N_t transmit antennas and 1 receive antenna. At the transmitter, the data is modulated and sent through the STBC encoder. Code rate of the STBC encoder is assumed to be 1 for the $N_t \leq 2$ case, and to be 0.5 in the $N_t = 4$ context. The channel is assumed to be block-fading, where $H \in C^{N_t X 1}$ is assumed constant in an observation block of $N = 1000$ symbols and corrupted by AWGN noise. The system is modeled such that H can vary for every observation block, and is unknown.

We have chosen BPSK, QPSK, 8QAM and 16QAM as the different possible modulations, with two or four transmit antennas, N_t = 2,4. For an example of the notation, the signal as BPSK-4 corresponds to BPSK modulation with $N_t = 4$. Let the modulation order be represented as Q. For example, $Q = 2$ represents BPSK, $Q = 4$ represents QPSK and so on. Therefore, the received signal can be classified into one in this set: M = {BPSK-2, BPSK-4, QPSK-2, QPSK-4, 8QAM-2, 8QAM-4, 16QAM-2, 16QAM-4}. While the set presented is not exhaustive, the proposed algorithm can be scaled to include other entries. We also deal with a few specific scenarios, not included in M.

Let $S = s_1, ..., s_k$ represent the set of all possible transmitted symbol vectors, where k represents the number of clusters or frequency-domain constellation locations. For $N_t \leq 2$, the STBC code rate is 1 and therefore, $k = Q^{N_t}$. For $N_t > 2$, the $k < Q^{N_t}$

when viewed on a 2-D plane and the values of k corresponding to set M are given in the subsequent sections.

A complete list of all the cases considered is presented in Table 2.1.

Modulation Type	N_t	k
BPSK	2	4
BPSK	4	8
QPSK	1	4
QPSK	2	16
QPSK	4	32
8PSK	1	8
8QAM	2	64
8QAM	4	256
16QAM	2	256
16QAM	4	512

Table 2.1: Modulations, Number of Antennas and Cluster sizes

Let $x(n) = [x_1(n), ..., x_p(n), ..., x_{N_t}(n)]^T$ be the nth transmitted signal vector, where $x_p(n)$ represents the nth transmitted vector at the pth transmit antenna and $x(n) \in S$. Let $y(n)$ represent the received signal vector. Then, $y(n) = Hx(n) + w(n)$, $n = 1, ...N$, where $w(n)$ is the AWGN vector, where each element in $w(n)$ is an identically and independently distributed (i.i.d) complex Gaussian random variable with zero mean and variance σ^2. The received signal $y = [y(1), ...y(N)]$ is processed to produce a candidate in set M.

2.3 Interclass Modulation Recognition

The modulations chosen for the experiment are from two families, M-PSK and M-QAM. In this algorithm, there are two phases in the classification, as shown in Figure 2.1. In this first level of classification, the received signal is first transformed using the Haar wavelet, chosen for its simplicity and ease of computation [40], and then classified as either belonging to the M-PSK family (BPSK, QPSK) or the M-QAM family (8QAM and 16QAM). The different M-PSK signals cannot be classified using wavelets directly, for low SNRs, as shown by studies in [41]. A particular benefit of

using wavelet transforms for the first stage of classification is because DWTs classify based on amplitude of the received signal, the M-QAM signals are separated into 8-QAM and 16-QAM in the very first step itself. This is then followed by the intraclass classification, exploiting constellation shapes of the received signals.

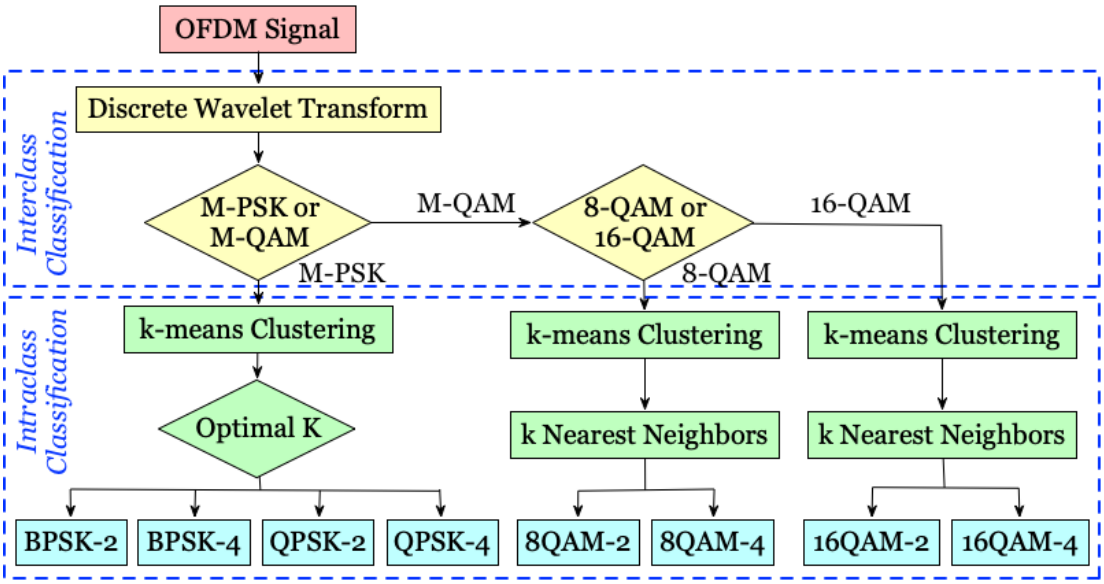


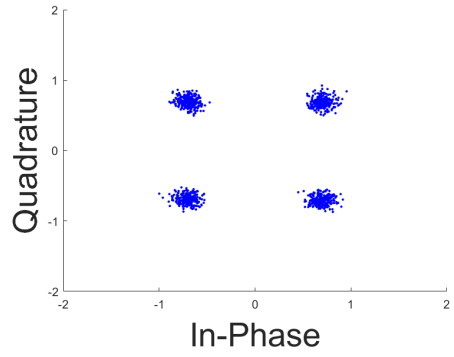
Figure 2.1: Proposed Algorithm

While it is tempting to eliminate the wavelet transform step altogether and choose only clustering for detection of both the unknowns, the large number of cluster points for the M-QAM signals (even with a single transmit antenna), doesn't allow for modulation recognition at low SNRs, as shown in [42].

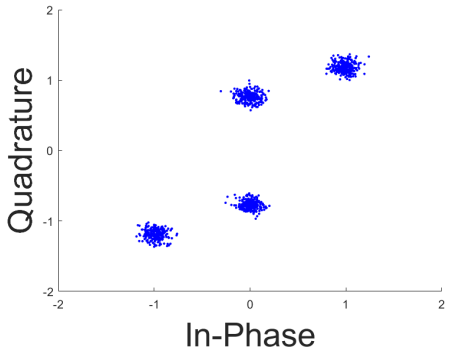
2.4 Intraclass Modulation Recognition and Antenna Enumeration

2.4.1 Clustering

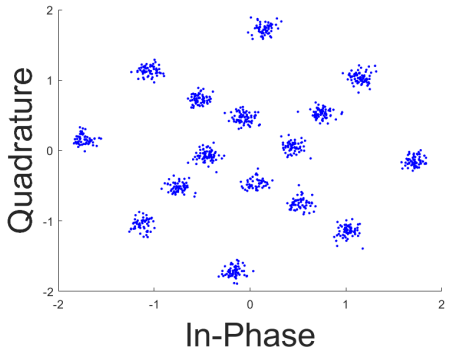
The proposed algorithm uses the same clustering information used for intraclass MC, for the detection of N_t, thereby reducing an additional step. The algorithm to calculate N_t presented here doesn't assume apriori modulation knowledge. The number of transmit antennas is detected jointly with the type of modulation for the different M-PSK constellations. However, since the Haar transform can differentiate between the M-QAM modulations, i.e., 8-QAM and 16-QAM, this knowledge is used to detect



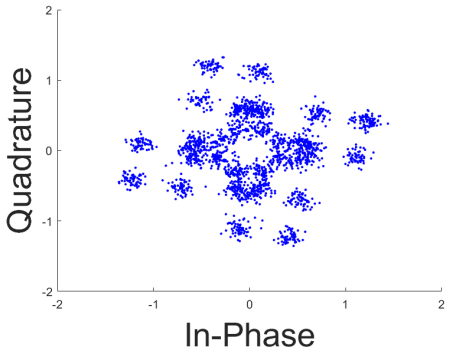
(a) QPSK-1



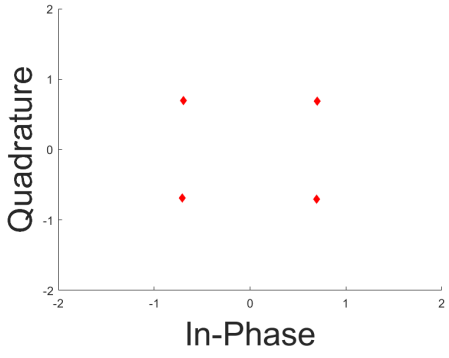
(b) BPSK-2



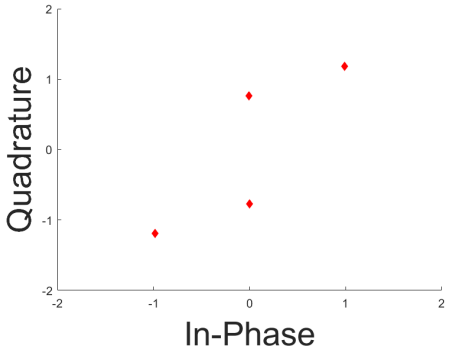
(c) QPSK-2



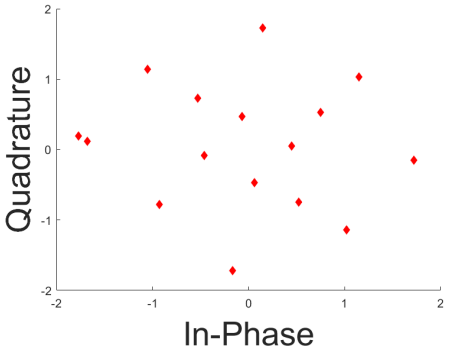
(d) QPSK-4



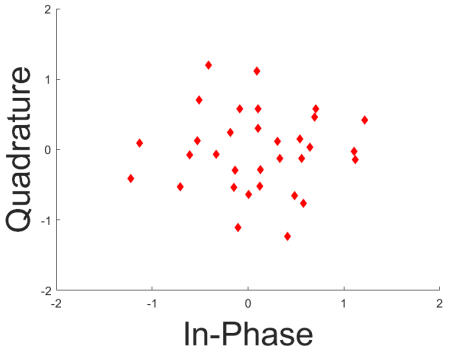
(e) QPSK-1 cluster locations



(f) BPSK-2 cluster locations



(g) QPSK-2 cluster locations



(h) QPSK-4 cluster locations

Figure 2.2: Clustering

the number of transmit antennas for the M-QAM family of signals. The number of constellation points are considered to be the ideal number of clustering locations and these two terms are used interchangeably in the rest of the chapter.

Challenges

There are four main challenges involved for the clustering technique:

- As N_t is unknown, there are N_t number of different channels. The number of constellation points increase exponentially with increase in modulation order Q and N_t.
- Training data cannot be given the exact locations of the received constellation points, as the channels are varying for different data entries.
- The number of clustering points can be similar for a few cases.
- Calculating the number of possible clusters is time-consuming and often erroneous for higher order modulation in the QAM family for the short observation window chosen.

Each modulation type has a distinct, symmetrical constellation shape. We exploit these features to classify the received signal. Instead of training the data using pre-existing templates, by clustering the incoming signal into optimal number of clusters based on the Calinski-Harabasz evaluation criterion [43], we can predict the modulation and the number of transmit antennas.

Given a set of observations $[y(1), ...y(N)]$, where each observation is a 2-dimensional vector, with I and Q samples, k-means clustering partitions the N observations into $k \leq N$ sets, $S = \{s_1, ..., s_k\}$. $k \in \{4, 8, 16, 32\}$ for the M-PSK class, corresponding to the constellation points for $M1 = \{$BPSK-2, BPSK-4, QPSK-2, QPSK-4$\}$ respectively. k is chosen such that the within-cluster sum of squares is minimized, i.e.,

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{y \in S_i} ||y - \mu_i||^2 = \arg\min_{\mathbf{S}} \sum_{i-1}^{k} |S_i| Var S_i \tag{2.1}$$

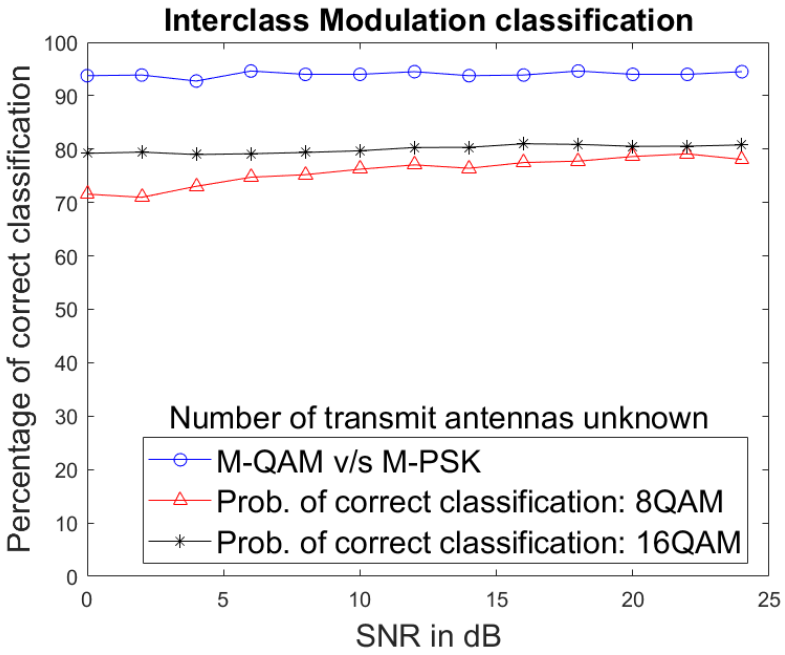


Figure 2.3: Interclass Classification

where μ_i is the mean of points in S_i [44].

An example with QPSK is shown in Figure 2.2. A SISO output of QPSK data modulation, at 20 dB SNR would yield Figure 2.2a, with only 4 clusters. When $N_t = 2$, output is as in Figure 2.2c and as Q for QPSK is 4, the number of possible clusters rapidly increases to $4^2 = 16$. While these clusters are clearly visible for a high SNR, (SNR = 20dB), as seen in Figure 2.2g, large number of clusters are difficult to separate at lower SNRs. This is even seen for the case of $N_t = 4$, at SNR = 20dB, as seen in Figure 2.2d, and the number of optimal clusters is 32 (Figure 2.2h). Using unsupervised learning, the k-means algorithm produces the best-fit number of groups.

2.4.2 k Nearest Neighbors

Same number of clusters

This section addresses the situation wherein the same number of cluster points can occur. While this situation isn't present in the set of candidates chosen here, using the clustering algorithm presented here can run into some road-blocks for a combination of different modulations and number of transmit antennas. To address this issue, we have

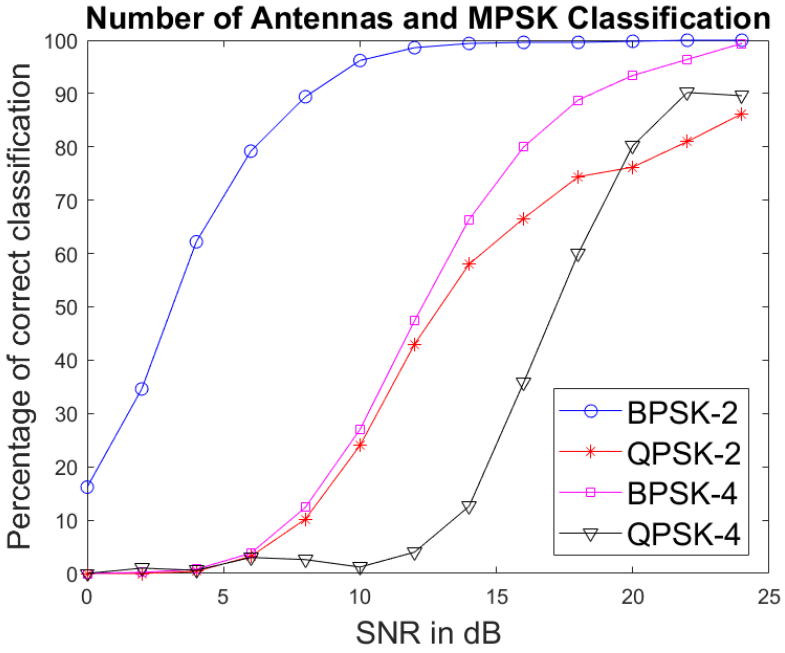


Figure 2.4: Classification of M-PSK with 2 and 4 antennas

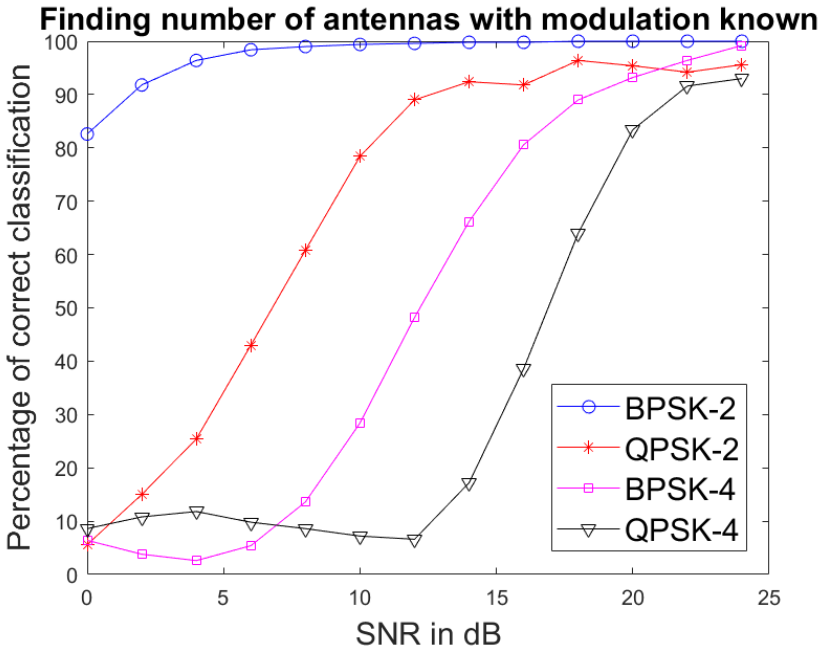


Figure 2.5: Calculating number of antennas when modulation is known

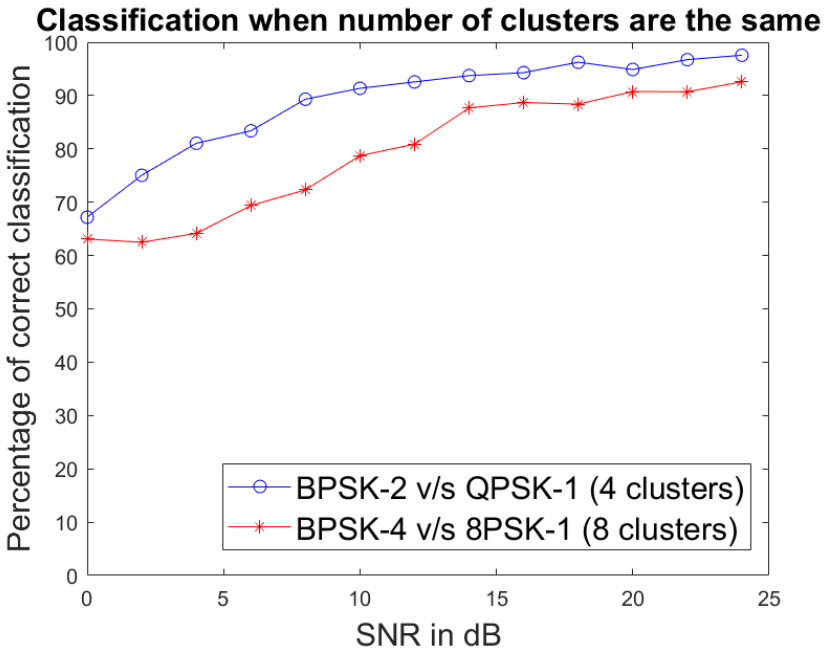


Figure 2.6: Classification using kNN (Same number of clusters)

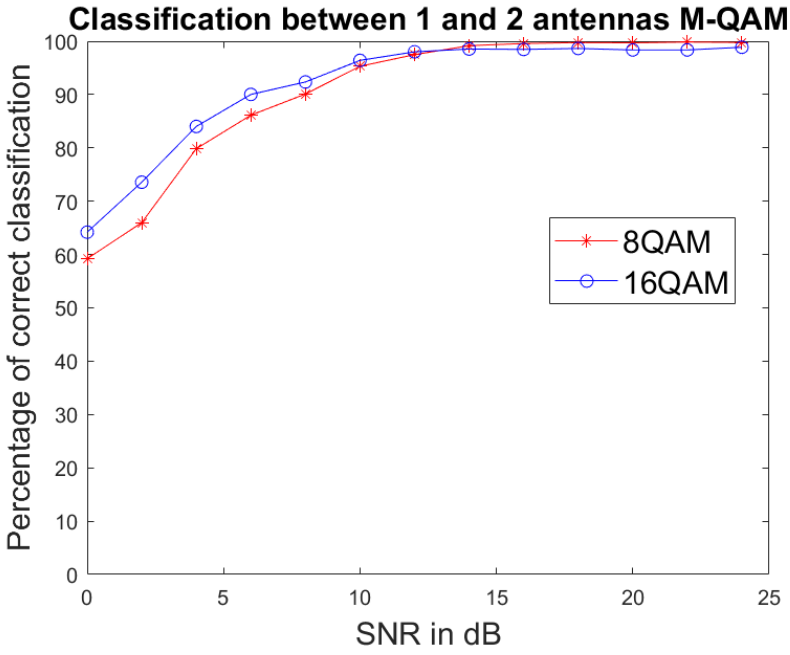


Figure 2.7: M-QAM: 1 and 2 antennas

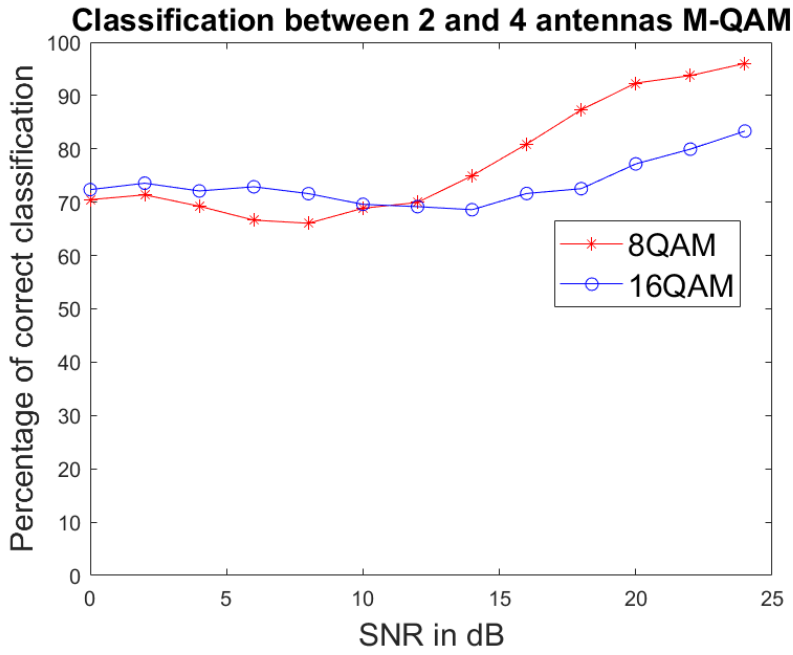


Figure 2.8: M-QAM: 2 and 4 antennas

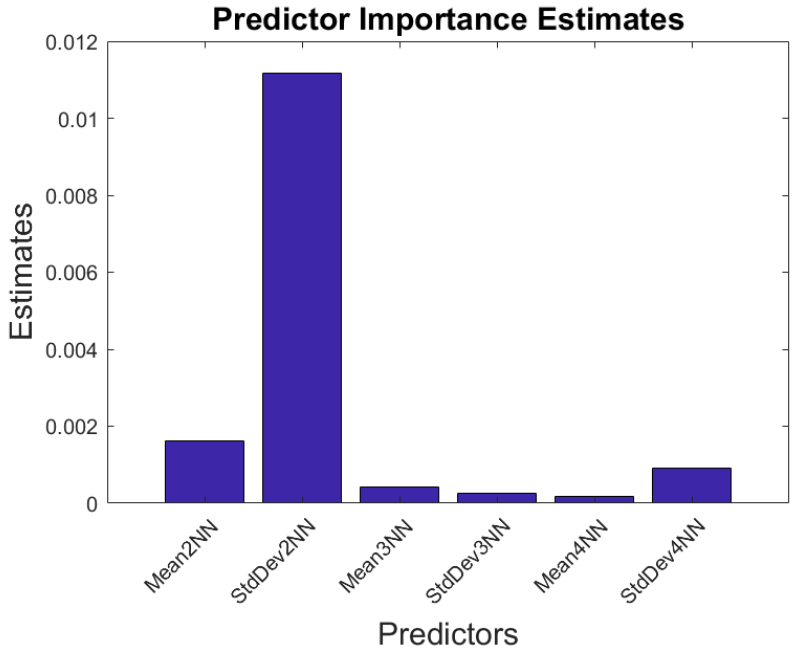


Figure 2.9: Predictor Importance for Classification between 8PSK-1 and BPSK-4

used the example of QPSK modulation, 1 antenna (QPSK-1), Figure 2.2 a, and BPSK modulation, 2 antennas (BPSK-2), Figure 2.2b. Both of these have 4 data clusters, as shown in Figure 2.2e and Figure 2.2f. To differentiate between the two, we exploit the symmetry of the clustering points in QPSK-1.

The k-nearest neighbor (kNN) search produces the k closest points in a data set to set of points in another set. Given the set $S = \{s_1, ..., s_k\}$, Euclidean distances between all the pairs (s_i, s_j), for $i = 1..k, i \neq j$ are calculated and arranged in ascending order. For this example, distances of 2 of the nearest neighbors are calculated. The mean and standard deviation of these distances is then used to create a regression model to classify the two systems. The key feature is the standard deviation, the value of which is lower for the symmetrical QPSK-1 system. For other such systems, with a higher number of clusters (such as 8PSK-1 and BPSK-4 each with 8 clusters), distances of more than 2 neighbors can be calculated. However, the standard deviation of the distances of the 2 nearest neighbors is of the most predictor importance.

Number of Transmit Antennas in M-QAM

As the interclass classification presented in 2.3 separates 8-QAM from 16-QAM, we use kNN clustering technique presented in 2.4.2 to calculate N_t. The set $M2 = \{$8QAM-2, 8QAM-4$\}$ and set $M3 = \{$16QAM-2, 16QAM-4$\}$ correspond to $\{64, 256\}$ and $\{256, 512\}$ constellation points respectively. However, as we use only $N = 1000$ symbols, grouping into 256 and 512 locations might not yield accurate results. Therefore the data is grouped into 16, 64 and 256 points for the 16-QAM case, and calculates the relative distances of the nearest neighbors in those cases.

2.5 Results

Each data set contains 1000 digitally modulated symbols collected after a Rayleigh block fading MIMO channel. Monte Carlo trials of 500 different channels for each data set were collected and 80% of the data was used to create the model. The other 20% was used as testing data. The graphs in Figure 2.2 capture only the results in the SNR

range of interest, between 0dB and 25dB.

Interclass Classification

The interclass MC using DWT yields a clear distinction between the M-PSK and the M-QAM families, as well as within M-QAM, as seen in Figure 2.3.

Intraclass Classification: M-PSK family

Using k-means clustering, N_t and the modulation of the different elements in set $M1$ are separated, as can be seen in Figure 2.4. It is to be noted that QPSK-2 and QPSK-4 have 16 and 32 clusters, and cannot be compared to a regular QPSK MC. With more information, in this case prior knowledge of the type of modulation, improves the performance of classification of N_t, as seen in Figure 2.5.

Intraclass Classification: Same number of clusters

When k-means clustering yields the same number of clusters for two different candidates, the kNN technique presented in section 2.4.2 is applied. Figure 2.6 shows the application of the algorithm on two such situations, {QPSK-1, BPSK-2} and {8PSK-1, BPSK-4}. Figure 2.9 highlights the importance of the standard deviation of the relative distances for 2 nearest neighbors.

Intraclass Classification: M-QAM family

kNN in conjuction with the possible groups formed by k-means clustering is used on the M-QAM family. The symmetry of the single-antenna output yields better results than in the case of only multiple antennas, as can be seen in Figure 2.7 and Figure 2.8.

2.6 Chapter Summary

Blind recognition of MIMO signal parameters is important in both the military context as the system is non-cooperative as well as civilian applications, such as in SDRs where secondary users can utilize the knowledge to avoid interference from primary users.

Blind classification can also help in improving transmitters' throughput as overhead is minimized. As modulation recognition and computing the number of transmit antennas are part of the same problem, a joint classifier is designed. In this chapter, we have presented a novel way using a combination of unsupervised clustering and kNN to jointly classify modulation and the number of transmit antennas, on top of a DWT classifier, without knowledge of or predicting the channels. These techniques can be easily scaled for different sets of possible candidates and future work comprises of classification with higher-order modulation schemes and greater than 4 transmit antennas.

Chapter 3

Predicting Channel Transition for MU-MIMO Beamforming

3.1 Introduction

For new 5G technology to evolve from existing protocols and technologies, each aspect of the wireless networking standards must be improved. In this chapter, a protocol is proposed which would supersede the existing transmit beamforming protocol in 802.11ac used by the Multi-User MIMO (MU-MIMO) technology. Transmit beamforming at the Access Point(AP) in 802.11ac requires explicit feedback from its clients with the Channel State Information (CSI) processed at the clients [45]. In this existing protocol, AP transmits a known packet and each of the clients uses this packet to create the channel matrix. This channel matrix is then compressed, called the compressed beamforming (BF) matrix, and sent back to the AP, which uses this information to precode the signal to each spacial stream to achieve maximum aggregate throughput and lowest error rate the channel will support.

The process of gathering CSI has been studied extensively. Research on MIMO technologies, which use CSI can be broadly classified into few categories: a) using instantaneous CSI [46], b) using reciprocal channels [47] c) using partial CSI feedback or d) novel methods of reducing the amount of CSI [48, 49]. The forward channel knowledge, i.e., the channel information available at the clients based on packets from the AP, known completely and instantaneously at the AP is the best-case scenario for MIMO communications. However, this is not realistically possible as there would be delays and errors on the feedback channel even if there is wired reverse channel (channel from the client to the AP) explicitly for feedback. Having such a wired feedback reverse channel would also result in hardware overheads and is not practically feasible. If a

wireless reverse channel is used, the channel information is corrupt and is limited by the coherence time of the forward link.

The next option is to use the channel information from the reciprocal channel. For a linear time-invariant communication system, the frequency response of the channels in both directions, forward and reverse, is considered to be the same [47]. The reciprocity principle for the forward and reverse channel states that the channel from one antenna to another is the is identical to the transpose of the channel in the reverse direction, given that the forward and reverse links operate using the same time, frequency and antennas [50]. Using this principle, in the absence of any feedback channel, the reverse channel link information is used to obtain the forward link information. In practical systems, these rigid conditions of the reciprocity principle cannot be fulfilled and thus, research such as [47], which focuses on identifying mismatches in the RF circuits and [51], which introduces a closed-loop method to compensate for the non-reciprocal interference structure in the forward and reverse links, are geared towards creating a reciprocal channel system.

3.2 Related Work

Overall improvements in MU-MIMO throughputs have been researched by controlling the grouping the clients by using different techniques such as in [52–54], or manipulating antennas, like a tilt in the AP antenna, as suggested by [55]. Attempts to improve transmit beamforming particularly have been studied in the past as well, in research such as [56], where the feedback overhead is reduced by differential quantization, and [57], where throughput based on data accumulation is used to optimize sounding rates. The work presented here tackles the issues with transmit beamforming at a more fundamental level; by reducing the number of control packets and BF reports, as opposed to modifying the structure of the packets, or grouping clients, or antenna reorganization, using only the channel information available.

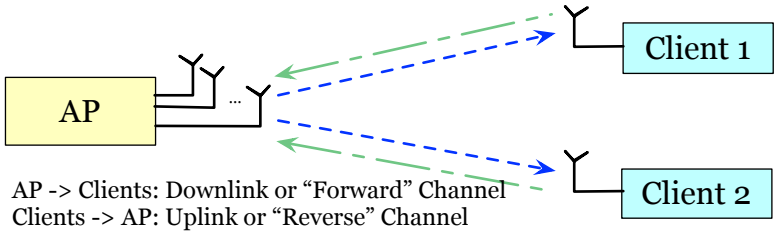


Figure 3.1: Forward and Reverse Channels

3.3 Motivation: Reducing Time Overhead

A typical explicit beamforming system, for a 1x2 SIMO system [58], such as Figure 3.1, is shown in Figure 3.2: The time taken for the transmit beamforming routine to periodically beamform to either of the clients, assuming 2 clients, can be summed up as: size of Null Data Packet (NDP) announcement packet + size of NDP + time taken to calculate the compressed BF or steering matrix by client 1 + size of compressed BF matrix + size of NDP + time taken to calculate the BF matrix by client 2 + size of compressed BF matrix. This calculation of time taken does not include possible collisions and thereby, any exponential back-off times. While the overhead for transmit beamforming can seem negligible for a few clients being handled by the AP, it becomes significant for a large client group. From [57] and [59], it can be seen that in an indoor environment, where there are non-static clients/the environment is not static, the sounding should occur not more than 20ms apart, for CSI accuracy. Taking 20ms as the sounding interval, for a modest simulation of 50 clients, the current 802.11ac transmit beamforming overhead could be a whopping 25% of the total time of transmission.

By monitoring periodic packets from the clients, the AP can decide whether or not to continue sending the sounding packets, depending on the already available (stale) CSI. Channel state can be seen to be unvarying for multiple seconds for static clients, as can been seen in Figure 3.3, which shows that there is no variation in mean-square differences between consecutive symbols, as will be explained in section 3.4.1. Given the state of the environment, if current CSI is required, from say Client 1, the total time can be calculated as: size of sounding packet + time taken to calculate CSI by client1 + size of BF matrix packet. This also eliminates the need of BF reports from

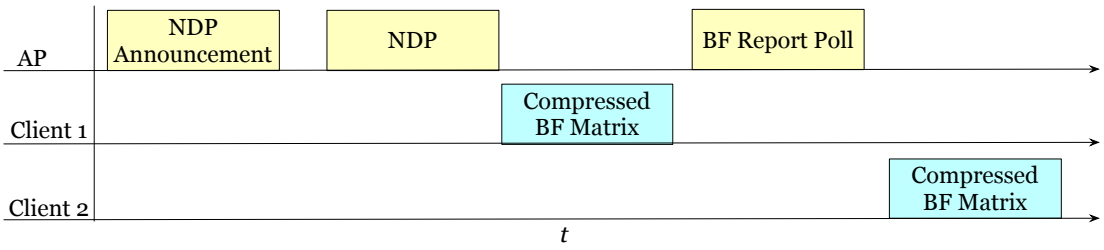


Figure 3.2: Sounding messages and BF reports in 802.11ac transmit beamforming protocol

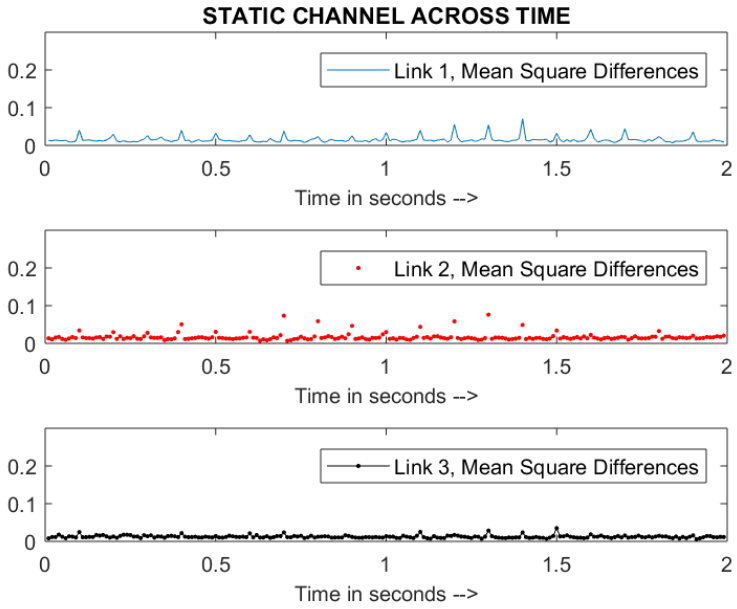


Figure 3.3: Static Channels

other clients, if there is no channel variation observed between them and the AP, a time-saving feature which will have more significant payoff for MIMO systems with many clients.

3.4 Proposed Protocol

The main objective of this protocol is to reduce the need for constant CSI feedback. This can be achieved by observing and classifying the environment and using this knowledge to decrease time overhead. To prove that polling for CSI is unnecessary in situations where the channel doesn't change rapidly, i.e., a static environment, it is necessary to observe channel behavior in both static and dynamic environments. If the channel

between the AP and the client with which it is communicating is classified as dynamic, then the AP will send a sounding packet, to which the only the client in question would respond with the beamforming matrix, which is the current CSI. If the channel is classified as static, then the AP will use previously received data from the client (i.e., stale CSI), if the data is within limits of allowed time. In order to keep off the costs of processing additional data, it would beneficial to be able to classify the environment using only the channel data, in order to trigger the collection of CSI. It would be imperative to study how the channels are affected by factors such as time and direction of data, based on the environment, to justify the use of this classification protocol.

Therefore, achieving this protocol can be broadly divided into two steps:

- Classification of environment
- Collection of CSI

3.4.1 Classifying the environment: CART Analysis

The collection of CSI for the different scenarios mentioned is to be collected and analyzed. The observer can then make use of this data as "training data" for a Classification and Regression Tree (CART) [60]. Instead of feeding raw channel information into the classification tree, the following parameters are chosen as inputs:

Capacity

The capacities of the different channels are calculated using [61]:

$$Capacity = \sum_{n=1}^{N_{sc}} log\left(1 + \frac{P_n|h_n|^2}{N_0}\right)$$

bits/OFDM symbol

Where, $\frac{P_n}{N_o}$ is the SNR at the n^{th} sub-carrier of the OFDM symbol ; h_n is the channel at the n^{th} sub-carrier of the OFDM symbol ; N_{sc} is the total number of occupied sub-carriers.

Capacity is calculated for both the forward channel, i.e., sounding packets from AP to the client and for the reverse channel, i.e., acknowledgement or "ACK" packets from

the client to the AP.

Aggregate of mean-square first differences

This parameter is chosen to preserve the absolute value of the differences between received consecutive packets. The data is carried using OFDM symbols in each of these packets. The Mean-Square (MS) difference can be computed as:

$$MSDiff = \frac{\sum_{n=1}^{N_{sc}} \left(sym_{n,i} - sym_{n,i-1}\right)^2}{N_{sc}}$$

Where, i is the total number of OFDM symbols ; sym_{ni} is the signal strength of the received n^{th} sub-carrier of the i^{th} OFDM symbol ; N_{sc} is the total number of occupied sub-carriers per OFDM symbol.

MS differences are calculated for both the sounding packets and the acknowledgement packets, i.e., for the forward and reverse channel. The above-mentioned parameters for an exhaustive set of scenarios are to be fed to the CART tree with the classification results, which will train the system to identify the unknown environment when the same set of parameters are inputted.

3.4.2 Quantifying the different environments: Precoding and EVM

For the purposes of justifying this protocol, it will be assumed that the knowledge of the forward and reverse channels is available at the AP and all the clients. Simple precoders or beamform steering matrices, will be created using different channel information to compare with respect to the instantaneous CSI available at the AP. Beamformed packets can then be analyzed to check for performance under different conditions.

The different precoders considered are:

- Instantaneous CSI available at the AP (best-case scenario)
- Stale CSI (Information using the first packet received as reference)
- Instantaneous reverse CSI (Information available using packets received by the client)

- Stale reverse CSI (Information using the first packet received by the client)

Quality of modulation of the 802.11 transceivers can be assessed using Error Vector Magnitude (EVM) [62]. The error vector is calculated by measuring how far the magnitude and phase of the received symbol have strayed from the ideal symbol location. EVM is calculated as the ratio between the error vector and the magnitude vector and is depicted in dB using:

$$EVM_{dB} = 20log(EVM_{\%})$$

Therefore, EVM will be a good indicator of how the beamformed packets behave with information gathered under different environments.

3.5 Methodology

A number of different environments would be required to be observed and collected to map channel behavior. Movement of any kind would cause a change in the channels, and the environment would be classified as dynamic. The environment can be broadly classified into the following:

- Static environment: Data collected when there is no movement around the AP or the clients.
- Dynamic environment: Data collected when there is movement around the AP or the clients.

These can be further categorized into 2 scenarios, one where all the space-time streams are affected, that is, when there is movement around the AP, and the other where only one (or more), but not all, of them is affected, that is, there is movement around one (or more) of the clients, but not all of the clients. The latter scenario has elements covering both static and dynamic environments. A study of the scenario where there is movement around only one of the clients would be of most interest, as this would cover all the aforementioned situations at the same time.

This protocol does not address the calculation of compressed beamforming matrix, and the quantization required to send the information to the AP. As the size of the

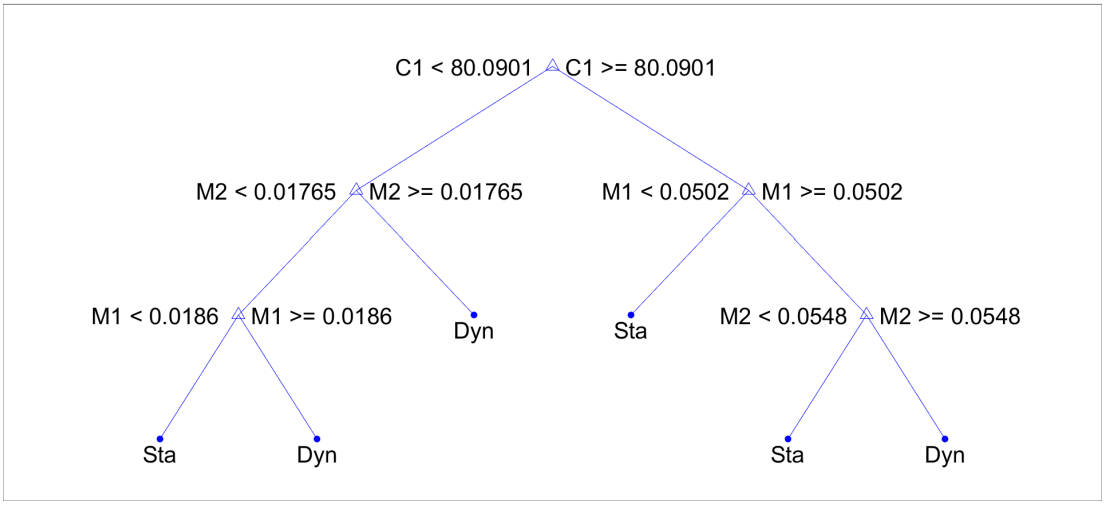
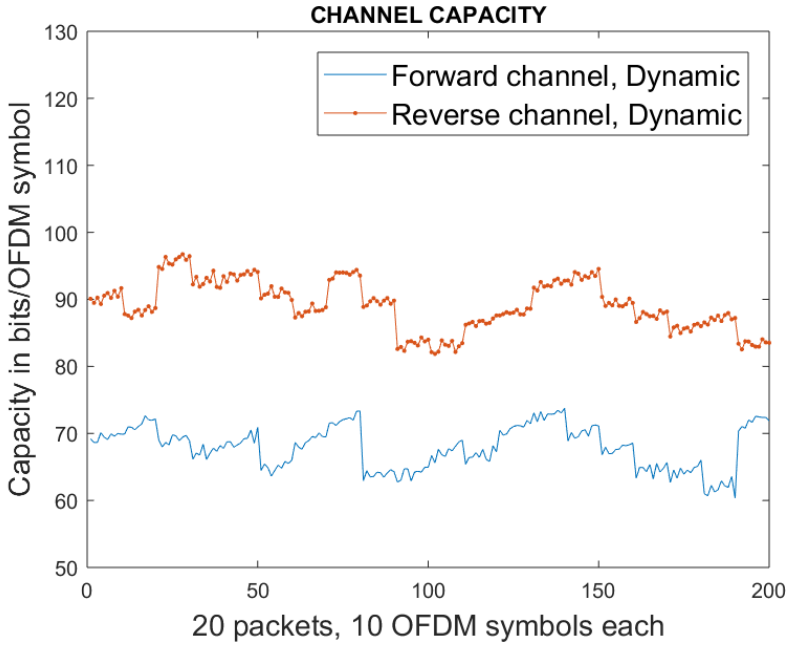


Figure 3.4: Classification Tree

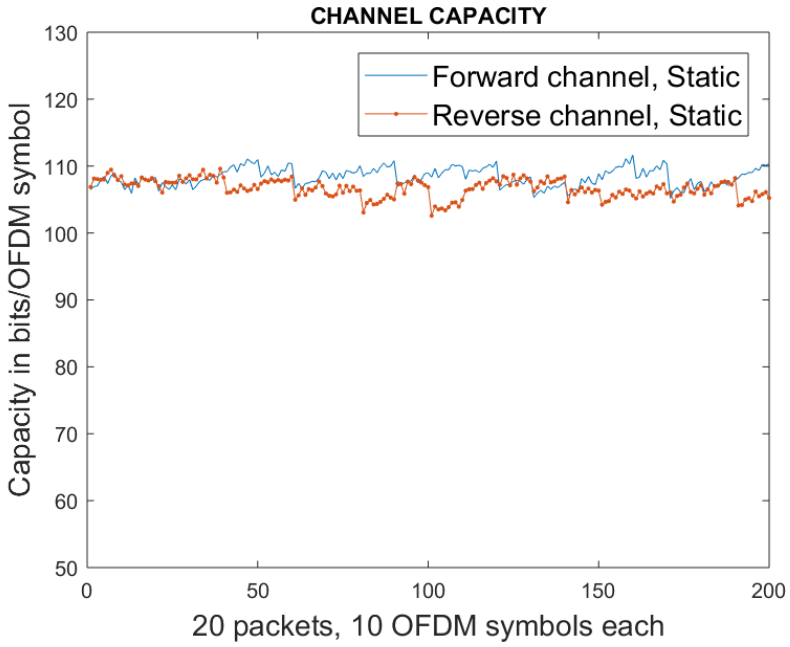
steering matrix does not effect the purpose of this experiment, it is assumed that the AP and both the clients have prior knowledge of the contents of the packets. Therefore, all the 52 occupied subcarriers per OFDM symbol in the 802.11ac data packet can be used as pilots.

Classification Tree

For this experiment, data packets were collected in a number of situations, and the system was set up to include the aggregate of mean-square differences and the capacities and was assigned the classification variable, Dynamic or Static, depending on the scenario. The capacities of the forward and reverse channel have been plotted in Figure 3.5 for the two possible environments, using the channel between AP to Client 1 to represent a dynamic environment and the channel between AP and Client 2 to represent a static environment. The difference in capacities between the forward and the reverse channel is of importance as they clearly distinguish between the type of environment. 20 packets are collected, separated by 100ms, each with 10 OFDM symbols. The plots in Figure 3.6 contain the mean square differences between 200 OFDM symbols. Each of the plots contains both the forward and reverse channel mean square differences, for comparison. The figures shown are for one of the test cases, where there is movement around one of the client's antennas.



(a) Channel Capacity: Static



(b) Channel Capacity: Dynamic

Figure 3.5: Capacities in bits/OFDM symbol

Using many such data points, varying in time of day, location of interference, SNR and frequency of packet transmissions, a CART analysis is performed. The overall set contains 494 data points, of which a random set of 370 points were allocated as a training set and the rest were used at the test set. An accuracy of 96.15% was achieved, with a precision value of 96.4%, a recall of 95.4%, and an F1 score of 95.8%, averaged over 30 trials with a different sample of 370 training points and test set chosen randomly in each run. For one such trial, the decision tree is shown in Figure 3.4, where the forward and reverse channel have attributes $M1$ and $M2$ (the MS differences of consecutive OFDM symbols) and C1 and C2 (the capacities) respectively.

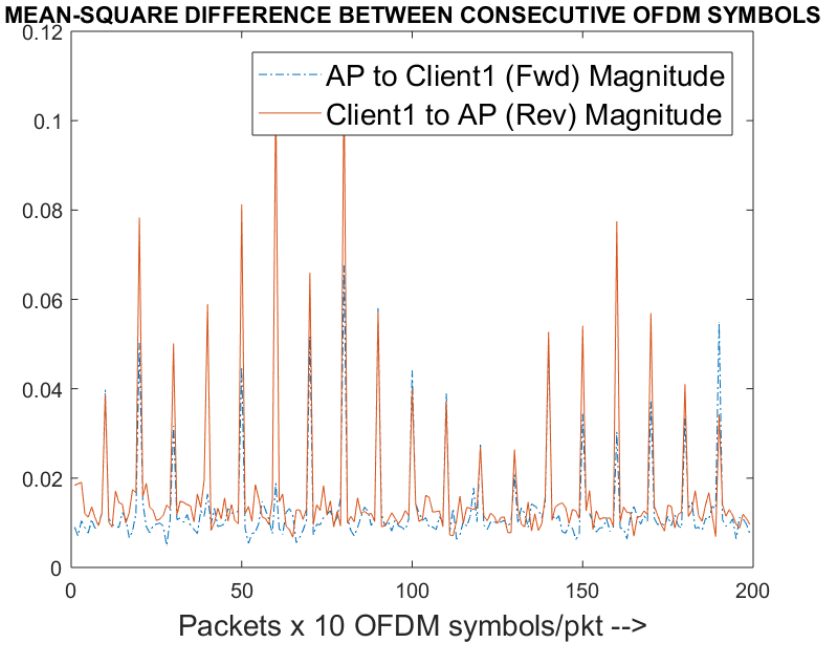
EVM Calculation

The precoders are created by isolating the channel matrix from the received packet. A new input sequence is created by multiplying this precoder with the original input matrix and this new packet is sent via a simulated noisy channel to observe the results.

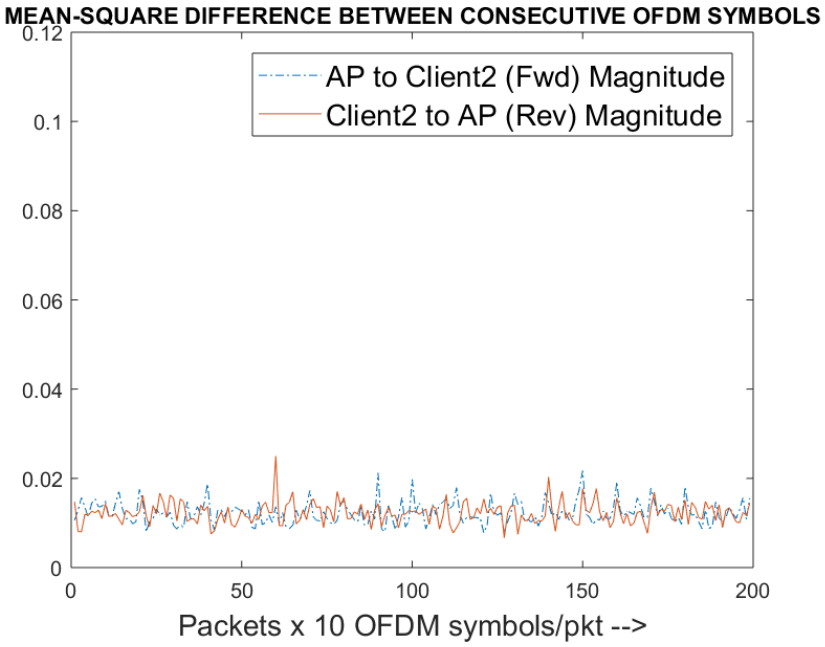
In this experiment, QPSK modulation with $\frac{3}{4}$ code-rate is used, but the EVM calculations can be extrapolated for other modulation schemes. Therefore, for example, even if the EVM is still acceptable while using stale CSI, or reverse CSI, from Figure 3.8, it might not be acceptable if using a higher modulation scheme. The EVM limits for different modulations are presented in [63].

These EVM values from the figures highlight the importance of the classification system, as stale CSI or reverse CSI can be used with lesser impact on throughput than when there is movement around the system setup. The difference in the EVM using reverse CSI and the current CSI is significant in the case of the dynamic environment, especially if higher modulation schemes are used.

If the system wrongly recognizes a static environment as dynamic, the penalty is an overhead, as the current CSI will be collected via sounding packets and calculation of the BF matrix. However, if a dynamic environment is wrongly classified as static, there would be an increase in the EVM value. However, that does not determine immediate packet loss as coding will recover some of the error, which is not considered here. If the packet received is erroneous, it is retransmitted by MAC layer and can also trigger the



(a) MS Difference: Static



(b) MS Difference: Dynamic

Figure 3.6: Aggregate of Mean-Square Differences

channel sounding procedure.

3.6 Hardware Setup

This experiment was performed in the ORBIT (Open-Access Research Testbed for Next-Generation Wireless Networks) Testbed in WINLAB (Wireless Information Network Laboratory), Rutgers University, New Jersey. The AP and the client nodes are part of the fixed ORBIT testbed [64], which consists of 400 radio nodes placed in a 20×20 grid with 1m spacing between the adjacent nodes. These large testbed nodes are equipped with a variety of radio platforms including 802.11 a/b/g, Bluetooth, Zigbee, and various versions of SDRs such as USRP platforms, WARP platforms, and the CRKit cognitive radios that were developed at WINLAB. Each of the AP and the client nodes chosen has an B210 USRP mounted on it and are fixed in location. Each client is at a distance of 15m from the AP and is at a distance of $15\sqrt{2}$ m from each other.

Synchronization between the AP and the clients for the transmission of sounding packets and the ACK packets occurs by using PPS input from the OctoClock-G to time the start of transmission as will be described in section 3.7. The OctoClock-G uses an internal clock source to consistently generate a 10 MHz reference signal as well as 1 PPS signal [65]. The OctoClock-G can also behave as a distributor of the 10 MHz clock signal and the 1 PPS signal, when the PPS input switch in the USRP is turned on to external. Each of the USRPs is connected to the OctoClock-G via matching-length co-axial cables. As this OctoClock-G has the capability to support up to 8 systems, it can be used for multiple clients.

The OctoClock-G system in the ORBIT grid has a Master Clock whose output is fed to various floor units and ceiling units, as seen in Figure 3.7, which in turn can be connected to the USRPs, as mentioned in [21]. A tutorial detailing this experiment can be found on the ORBIT website, including the library for the CART analysis.

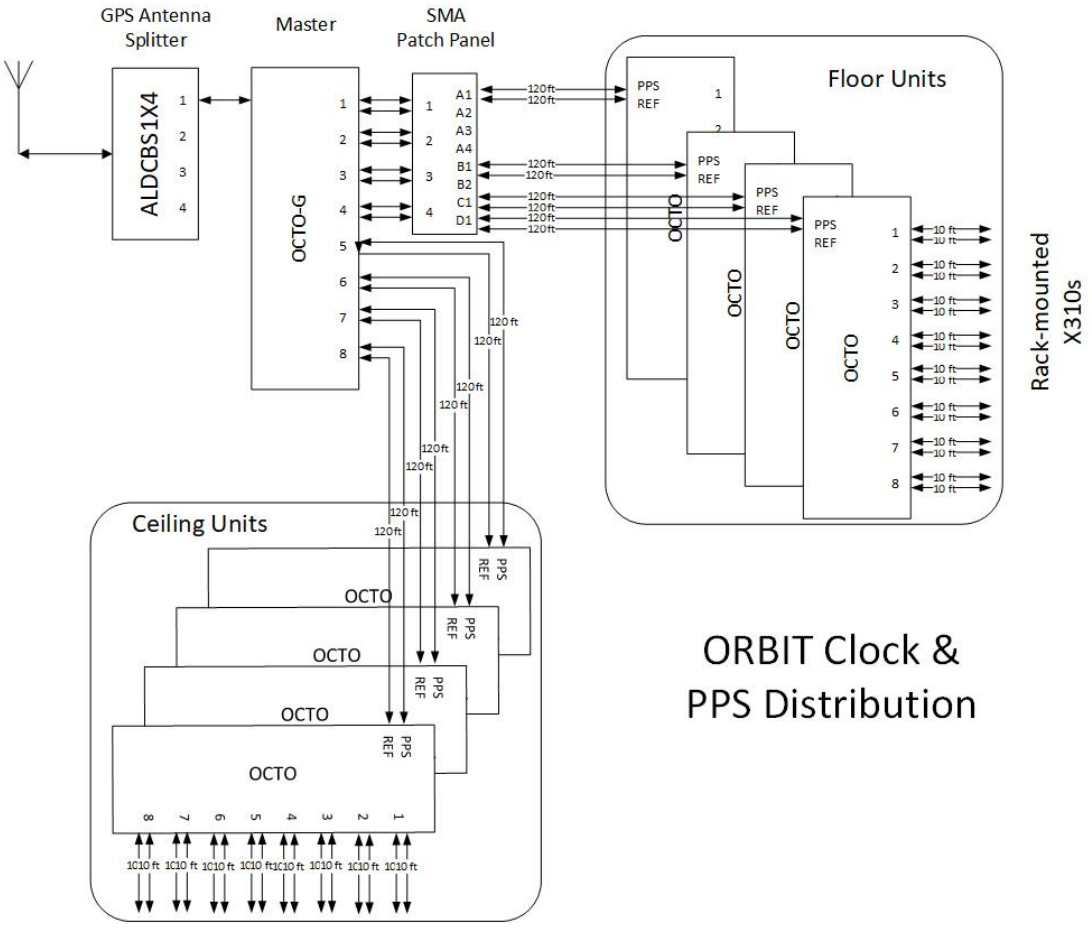


Figure 3.7: OctoClock-G

3.7 Implementation

3.7.1 Overall system

Three synchronized USRPs (two B210 USRPs for the clients and one B210 USRP for the AP) are setup for the observation of each channel. To achieve a system to compare channels at the same time, frequency and antennas, a switching mechanism is used, wherein the TX/RX antenna of one of the daughterboards of each of the USRPs is fixed for both the transmission and receiver operation. The switching between transmission and reception occurs every 100ms, therefore, the two channels are only 100ms apart, in a low-fading environment, well within the coherence time. The same operating frequency is maintained in both directions, chosen to be 5.4GHz based on the type of antenna. An important aspect be noted here is that while the experiment uses two clients, the system is scalable and can support up to 8 MU-MIMO clients.

3.7.2 Modifications

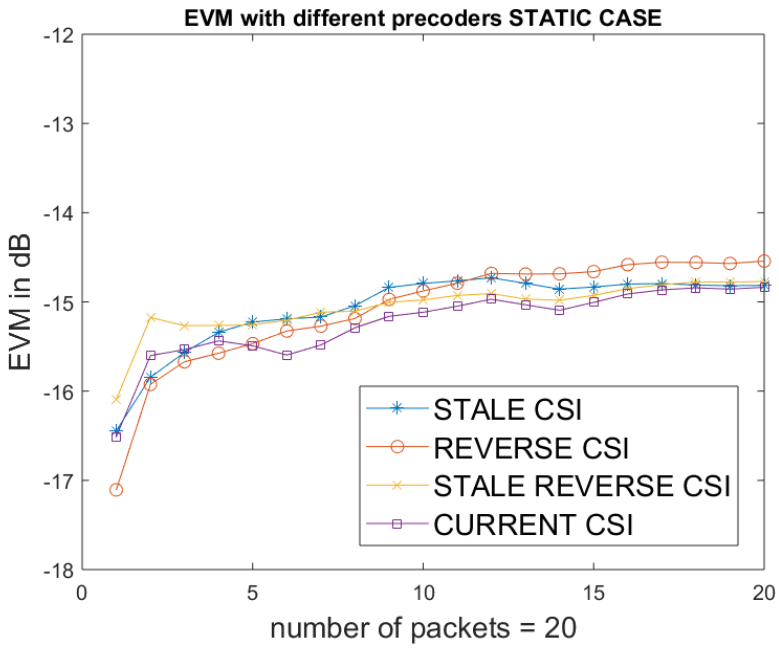
This experiment intends to study the behavior of the forward channel with respect to the reverse channel using one AP and two clients, in order to draw conclusions about the rate at which CSI is required at the AP for beamforming. In a traditional system as shown in Figure 3.1, the forward channel is created by the messages sent by the AP to the clients. These clients scan the spectrum continuously to spot the incoming packets. The clients then respond to the incoming packets by sending acknowledgement packets in return, back to the AP, thus creating the reverse or up-link channel. To replicate this exact scenario in SDRs, the following issues would be faced:

- Continuous polling of the spectrum by multiple clients creates overhead.
- The response time of the clients in sending the ACK packets should be lower than the coherence time of the channel.
- The ACK packets from multiple clients might be received at the same time, which could require further processing at the AP to disentangle the ACK packets to clearly distinguish the information from the 2 clients.

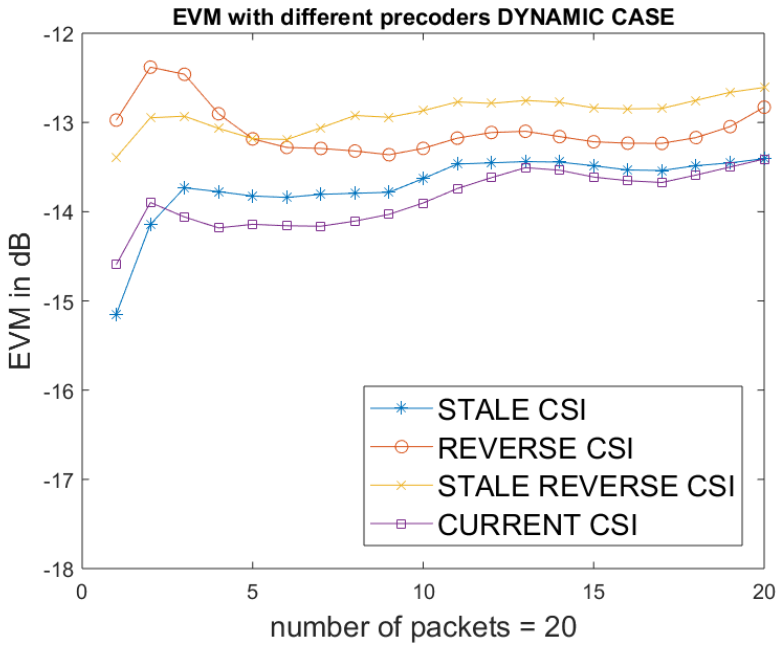
This experiment addresses the above issues by using external synchronization. An OctoClock-G is used as a synchronizer for the AP and the 2 clients and its PPS output is used to start transmissions at these nodes. That is, at the input from the PPS pulse, the AP continuously transmits packets at time t_i, $i = 1, 2, 3...$ Instead of scanning the spectrum for the AP's sounding packets, the Client 1 uses the PPS output of the OctoClock-G and transmits its ACK packet at time $t_i + \delta_1$. Similarly, Client 2 transmits its "acknowledgement" packet at time $t_i + \delta_2$, such that δ_1 + length of acknowledgement packet $> \delta_2$, so that there is no collision of the ACK packets, and δ_2 + length of acknowledgement packet $< t_i + 1$ so that there is no collision between the ACK packet and the messages from the AP. Here, transmissions from each of the systems are set to occur at 100ms, and δ_1 and δ_2 are set as 30ms and 60ms respectively. To maintain the same antenna location and hardware components in the RF chain, each of the USRPs uses only one TX/RX antenna from one of the daughter-boards, switching rapidly between transmission and reception functions, using the PPS signal as the trigger to switch between the functions. The system is set up such that each of the USRPs is constantly receiving, switching to the transmission mode only at t_i, $t_i + \delta_1$, $t_i + \delta_2$, where $i = 1, 2, 3..$, for the AP, Client 1 and Client 2 respectively.

3.8 MU-MIMO Experiments using Software Defined Radios

IEEE 802.11ac has Multi-User MIMO capability, which is the key improvement over its predecessor, 802.11n. MU-MIMO techniques are also adopted by wireless standards such as LTE [66] and WiMAX [67]. A general MU-MIMO system has one transmitter that can support data streams to multiple receive antennas simultaneously. These receive antennas can be on multiple receivers. A great deal of theoretical work has been done on the capacity of MU-MIMO systems, algorithms to choose clients, improve beamforming gains, and some practical work has been done on deployment of MU-MIMO in a controlled environment and for field trials [68], [69]. Another separate line of work that requires multiple transmitters and receivers are phased arrays [53, 70–72]. Given the broad applications of multi-antenna transceivers, a testbed is created here that allows for experiments involving multi-antennas to be easily run. The proposed



(a) EVM in a static environment



(b) EVM in a dynamic environment

Figure 3.8: EVM in dB

set up is open-sourced and the system can be easily replicated and modified to change parameters such as distance between the transmitter and receiver antennas, transmitter or receiver gain, operating frequency etc., to suit the conditions of the experiments. The motivation behind such a testbed is to allow the user to explore the various capabilities and boundaries of using MU-MIMO without having to create the testbed from scratch. This testbed can be easily expanded to the scope of massive MIMO experiments, distributed antenna systems and other experiments which require a large number of software-defined radios for transmission and reception.

MU-MIMO technology is a single transmitter communicating with multiple receivers in the wireless domain. Channel estimation becomes a critical aspect for any MU-MIMO system. The experimental model for such a system can be summarized as follows. The AP broadcasts sounding messages, independent signals from each of the transmitter antennas, and the client receives these frames and computes the individual channels. This information is sent back to the AP in the form of a channel matrix. The AP uses this channel information to create the beamforming matrix and choose the clients, such that multiple clients can be serviced simultaneously.

Replicating this model exactly in Software Defined Radios(SDRs) is difficult, as the processing time for this system would be high. With the above-mentioned model as a reference, the setup is simplified to include more offline processing, freeing up the hardware at the transmitter and receivers. Consider the setup as shown in Figure 3.9, where there is 1 transmitter, with 2 transmitting antennas and 2 receivers, each with 2 receiving antennas. The sounding messages at the transmitter are created such that the switching mechanism between the two transmit antennas is mimicked. Instead of switching rapidly between the two streams of data as in a traditional MU-MIMO, the system transmits 2 streams of data simultaneously, on both the transmit antennas. To receive the data in the MU-MIMO data structure, the 2 transmit data files are created such that there are zeros in the place where the other transmit file has data and vice-versa. Figure 3.10 shows how the 2 data files are constructed. Thus, this technique creates a simple system to perform channel estimation in MU-MIMO.

The programming model is depicted in Figure 3.11. Once the receiver obtains the

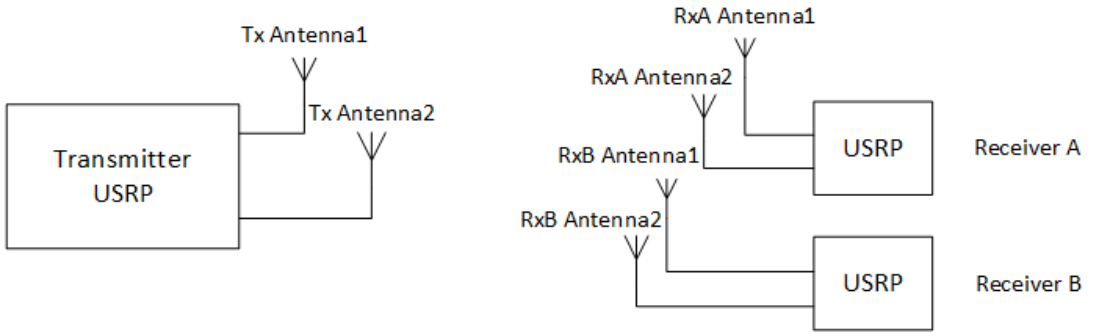


Figure 3.9: 2x4 MIMO

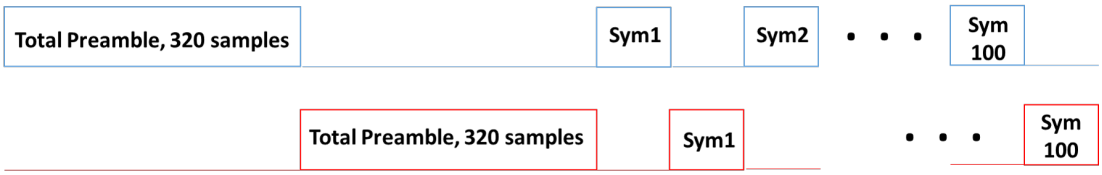


Figure 3.10: Two tightly synchronized data files

sounding messages from the transmitter, it stores this data. These sounding messages are assumed to be fully known at the receiver. The file is then processed in MATLAB, outside the real-time system. Once the channel matrix is obtained, a beamforming matrix can be calculated, and can be used to select multiple users. The presence of an Oracle is assumed, an all-knowing system, which can carry CSI from the receiver to the transmitter instantaneously, without loss or degrading of information, and without the use of the conventional channel. This Oracle is also required for checking the experimental results against the base case scenario. Future work includes the use of this beamforming information by the transmitter to service multiple users in real-time. This limits the amount of time that can be spent outside the real-time system, to process the data, as it has to be completed within the coherence time of the CSI.

3.8.1 Physical Setup

This experiment was run in the ORBIT Testbed in WINLAB, Rutgers University, New Jersey. The comprehensive tutorial section in the ORBIT website can guide the user through the various steps to set up a basic communication path between USRPs. This

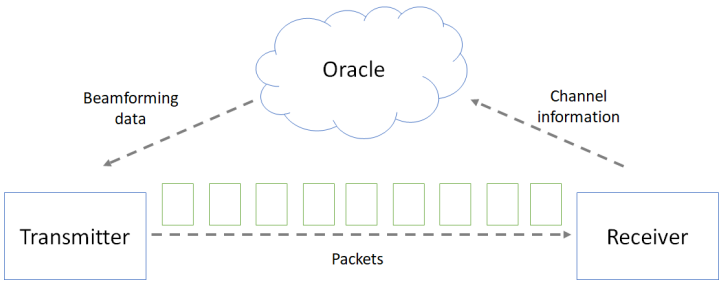


Figure 3.11: Transmission and Reception with an Oracle

experiment can be reproduced and expanded by loading the transmitter and receiver programs available in the tutorial section onto the images on the appropriate radio nodes on the ORBIT grid or on any testbed similar to the one described. The experiment is setup for 2 transmit antennas and 4 receive antennas, i.e., a 2×2 MU-MIMO system, with 1 transmitter and 2 receivers. The transmitter node is part of the fixed ORBIT testbed. There are an additional 8 nodes with X310 USRPs on them placed together in the corner of the ORBIT laboratory, referred to as the MIMO rack. The transmitter node chosen has an X310 USRP and has 2 transmit antennas. The transmitter is fixed in location.

	Transmitter	**Receivers**
USRP type	X310	B210
Location	MIMO rack, Node23-1	Mobile (Outdoor) Nodes 2-7, 2-8
Type of Antenna	Abracon APAMSJ-137	Apex II TG.35 4G Wideband

Table 3.1: Details of the USRPs

Two of the mobile nodes from the outdoor domain function as the receivers and have B210 USRPs mounted on them. These nodes are portable and could be moved anywhere within the ORBIT room. The details of the nodes are given in Table 3.1. The distance between the mobile receiver nodes and the transmitter node is arbitrary and was chosen such that the system enjoyed a high SNR.

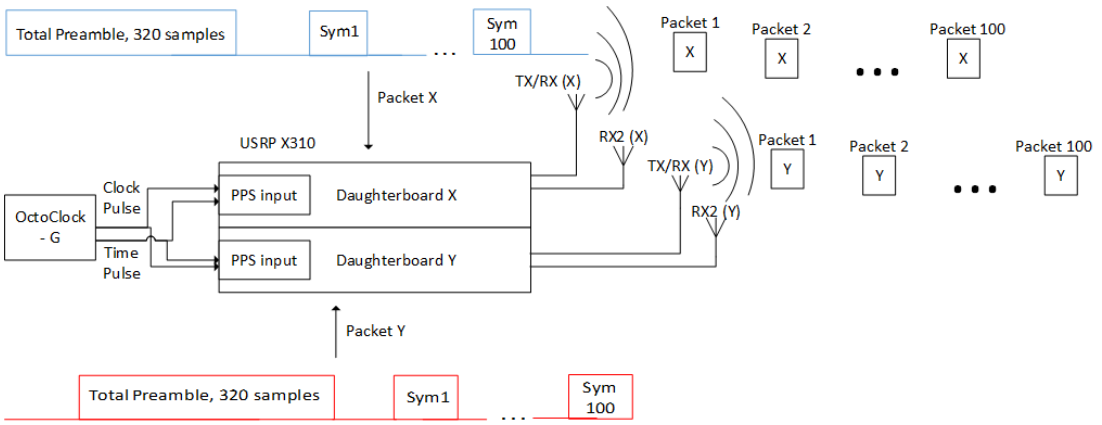


Figure 3.12: Transmitter details

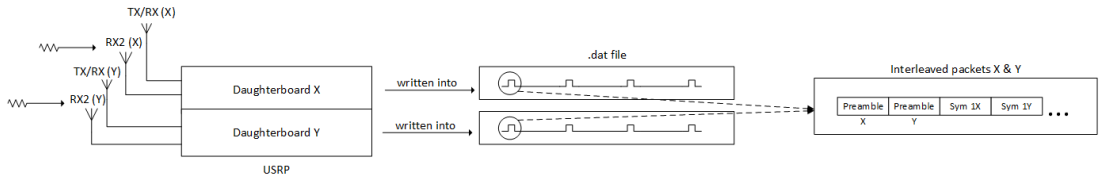


Figure 3.13: Receiver details

3.8.2 Hardware Setup

The transmitter X310 has 2 UBX daughterboards, which support an operating frequency range of 10 MHz to 6 GHz. The receiver B210s can operate from 70 MHz 6 GHz, [73]. Given the different antennas on the devices and their tunable frequencies, 5.4 GHz was chosen to be the operating frequency. The USRPs have 2 daughterboards and/or 2 separate RF chains and support full-duplex MIMO applications. The 4 antennas mounted on each of these USRPs is of relevance here. There are 2 TX/RX antennas which can be configured to be transmitters or receivers, and 2 RX2 antennas which are used as receivers by default. At the transmitter, the TX/RX antenna on each of the daughterboards is used for transmission. The two RX2 antennas are used at the receiver.

The 2 RF chains can be tuned to different frequencies. However, for the purposes of this experiment, they are all tuned to the 5.4 GHz frequency, for the simultaneous switching transmission described earlier.

3.8.3 Collecting the data

Transmission and Reception

Two of the APIs, tx_samples_from_file and rx_samples_to_file, which can be found in the USRPs UHD example folder have been chosen to be modified for the purposes of this experiment. The original examples are setup for a Single-Input Single-Output(SISO) system. These APIs are modified such that the transmitter can now transmit on 2 antennas and the receiver is able to receive on 2 antennas as well. This can be seen in Figure 3.12.

Synchronization

The 2 transmit streams on the transmitter were synchronized using the OctoClock-G. To tightly synchronize the 2 daughterboards such that these 2 data files are transmitted at exactly the same time and clock tick, an external PPS input of the OctoClock-G is used. This technique of transmitting files, including using the same operating frequency for both the transmit stream ensures that the files received at the receivers are interleaved packets from the 2 transmit data streams, and can be seen in Figure 3.13. These sounding messages are sent repeatedly, with a delay of 100ms between each transmission, at the rate of 2 MHz. As there are 2 receivers and 2 RF receiving streams on each of the receivers, there are 4 received data files to be processed. If the observation window is 1s, there are 20 data packets on each of the files to process.

Synchronization in the transmitter occurs by using PPS input from the OctoClock-G to time the 2 transmit files tightly.

The OctoClock-G system in the ORBIT grid has a Master Clock whose output is fed to various floor units and ceiling units, which in turn can be connected to the USRPs. The transmitter in the experiment is a rack-mounted X310 and gets its PPS input from the floor unit. This synchronization setup is particularly useful as it would enable to scale synchronization when there are multiple nodes, as in the case of massive MIMO or a distributed antenna system.

3.8.4 Transmitted Data

The structure of the data packet can be observed in Figure 3.10. The gaps in the data correspond to the length of the symbols of the other transmit stream. Each packet has the preamble followed by 100 OFDM symbols. Each of these OFDM symbols has 52 occupied subcarriers. The preamble for the 802.11ac is divided into 2 parts, the short and the long preamble, as described in [74]. The Short Preamble which is totally 160 samples long consists of a complex sequence which is repeated 10 times. The Short Preamble is generally used for Carrier Frequency Offset (CFO) correction. The Long Preamble which follows the Short Preamble, is also 160 samples long and has a complex sequence that is repeated 2.5 times. This is used mainly for synchronization purposes.

3.8.5 Processing the Received Data

An OEDL script is used to turn on the transmitter and the receiver simultaneously and switch off the receiver at the end of the time window required to record the packets. The raw data samples are then put through the OFDM receiver chain in MATLAB till the channel information is obtained. The MATLAB code uses the 802.11ac preamble to detect the start of each of the packets collected on the file in the time interval the receiver is switched on. The start of the packet is used as a guide to verify the synchronization at the transmitter. That is, the start of the packet from the second transmit antenna must be observed exactly at a distance of the length of the preamble (long+short) of the packet from the first transmit antenna for perfect synchronization. The data, which follows the preamble in the packet, is used to correct for CFO.

3.8.6 Summary of Testbed Setup

This section gives a comprehensive look at steps required for the set up and execution of a generic MU-MIMO experiment on the ORBIT testbed. The testbed is also set up such that it can be expanded and the scope is not limited to MU-MIMO. The set-up is a subset of the ORBIT grid and is available for experimentation. A combination of 4 antenna USRPs, X310s and B210s is used to create the multi-antenna multi-user scenario.

The mobile nodes provide location flexibility for the clients in the experiment. Precise synchronization using OctoClock-G is used at the transmitter to mimic rapid switching between antennas. This synchronization technique in addition to the structure of the preambles in the packets is also useful to verify the start of the data symbols in the packets. An Oracle is assumed which can send the CSI to the transmitter without loss of time of channel capacity. The received data is processed using the OFDM receiver chain and the channel is extracted. Once the channel state information is processed from the received data it can be used for various applications.

3.9 Chapter Summary

In this chapter, a protocol to replace the control packet-heavy 802.11ac transmit beamforming is proposed. This protocol uses the channel transition as the trigger for collecting CSI. The CART classification tree uses a library of previous results to predict the environment with over a 96% accuracy, and using this information the AP can decide whether or not to transmit a sounding packet to the clients. Differences in capacities and error vectors for the different environments are discussed. This protocol is shown to yield significant payoffs in reducing the time-overhead of the 802.11ac explicit feedback beamforming. A testbed for performing MU-MIMO experiments is also discussed.

The important next step is to test this system setup with more than two clients, and prove the benefits of using the protocol at full capacity of eight clients, and thereby improve the training data in the CART analysis library.

3.9.1 Applications

A system that cuts down the number of control packets and time overhead and uses the state of the environment to trigger a sounding procedure to collect CSI will be useful in situations which see both static and dynamic environments in the same setup, such as terminals at an airport, depending on the presence of passengers in that location. Broadly, using stale CSI without any new CSI can be useful in slow-fading channels.

Chapter 4

Traffic Prediction by Augmenting Cellular Data with Non-Cellular Attributes

4.1 Introduction

We consider harnessing principles from machine learning and deep learning for quality of service (QoS) management in mobile cellular networks. We propose to accomplish this through accurate prediction of mobile cellular traffic using past traffic data as well as other cellular and non-cellular attributes. The prediction may be in different quantities of interest − for instance the number of calls that will be made in the next n minutes, the number of messages that will be sent/received in the next n minutes, or any other quantity of interest.

Accurate prediction of mobile cellular traffic is essential for both service and infrastructure providers to provide a stable internet service and meet QoS guarantees. For instance, accurate predictions facilitate a schedule-based balancing of traffic load between cellular base stations, as depicted in figure 4.1, i.e., the network operators can distribute traffic demands from busy stations to free ones.

Existing literature models traffic prediction at a base station as a general time series forecasting problem and leverages methods from statistics such as ARIMA and support vector regression to compute predictions [75, 76]. These methods, however, do not capture several features of cellular mobile traffic that are predominant in real environments. Firstly, cellular traffic data is highly dependent on user mobility and is often highly bursty. Existing methods of modeling rely on mean values of historical data and hence do not capture such rapid variations. Secondly, existing methods do not consider the spatial and temporal correlations in mobile traffic data. Traffic data collected from real world has been shown to be multi-level periodical. The temporal characteristics

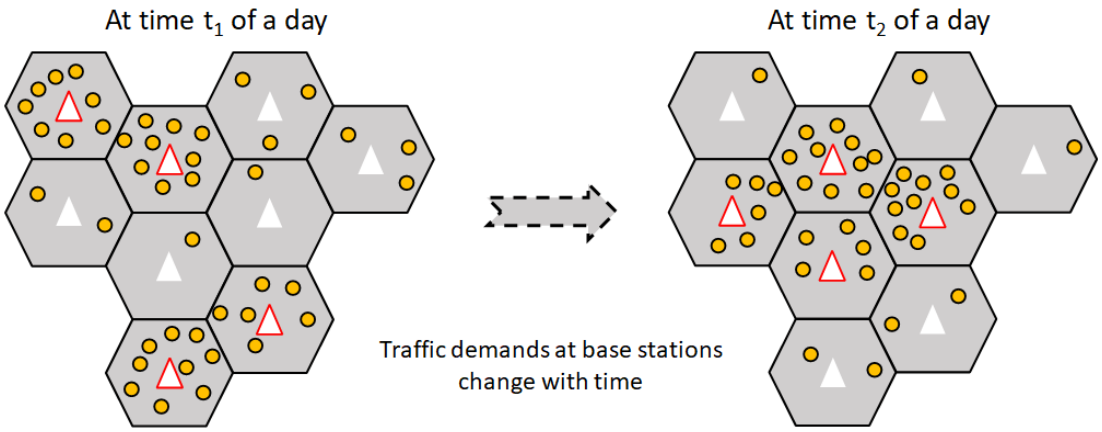


Figure 4.1: Traffic demands

of the traffic data collected from a cellular base station on a workday are similar to that collected from the same base station on a different workday. Furthermore, there is periodicity in traffic data within a day as well.

Similarly, cellular traffic data between neighboring base stations have been found to be correlated, thereby demonstrating spatial dependencies. Furthermore, traffic data at a base station is highly influenced by other external factors such as point of interest (PoI) distribution, that is, the presence of local attractions, subway stations, transportation hubs in the vicinity of a base station where users typically congregate. The traffic is also dependent on the day of the week and the time of the day.

To incorporate all of the above-mentioned factors, we aim to augment existing cellular traffic data with data from neighboring cells [77] as well as non-cellular data such as weather and data gleaned from an open-source API. We perform a one-step prediction as well as a trend analysis to observe gains by using augmented data.

We leverage techniques from deep learning using the cellular data available for the City of Milan, part of Telecom Italia Big Data Challenge, as recorded in [78]. Specifically, we use recurrent neural networks (RNNs) that can capture long temporal relationships in data and can model non-linear sequence relationships. Theoretically, RNNs can capture infinite temporal relationships in data. They are widely used in speech recognition, natural language processing, and time-series forecasting. We use long short-term memory, or LSTM, as described in [79] which is an artificial RNN architecture, for the

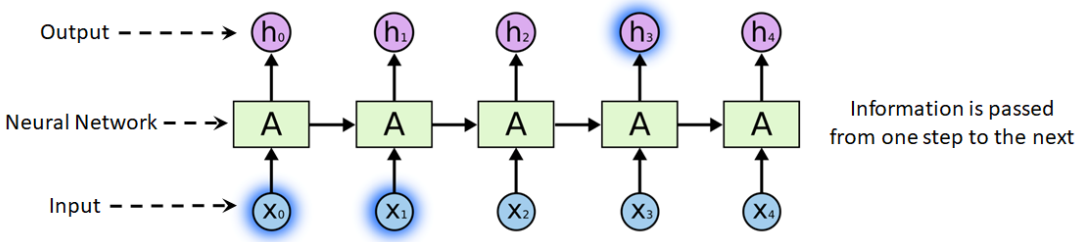


Figure 4.2: Recurrent Neural Networks

task at hand [80].

4.1.1 Recurrent Neural Networks

Since traffic trends observed at a base station are typically bursty and they have long term relationships or dependencies, simple feed-forward neural networks cannot be employed to extract the temporal correlation features from the observed traffic data. To this end, we use RNNs. RNNs are networks with loops in them thereby allowing information to persist.

Figure 4.2 describes a representative architecture of RNNs. The block A is a neural network. The x values denote the input and the h values denote the output. An RNN can be thought of as multiple copies of the same network, each passing a message to a successor. This chain-like nature makes RNNs closely related to sequences and lists. One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame.

LSTMs are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior. All RNNs have the form of a chain of repeating modules of neural network, as discussed previously. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a special way.

The key to LSTMs is the cell state, the horizontal line running through the top of

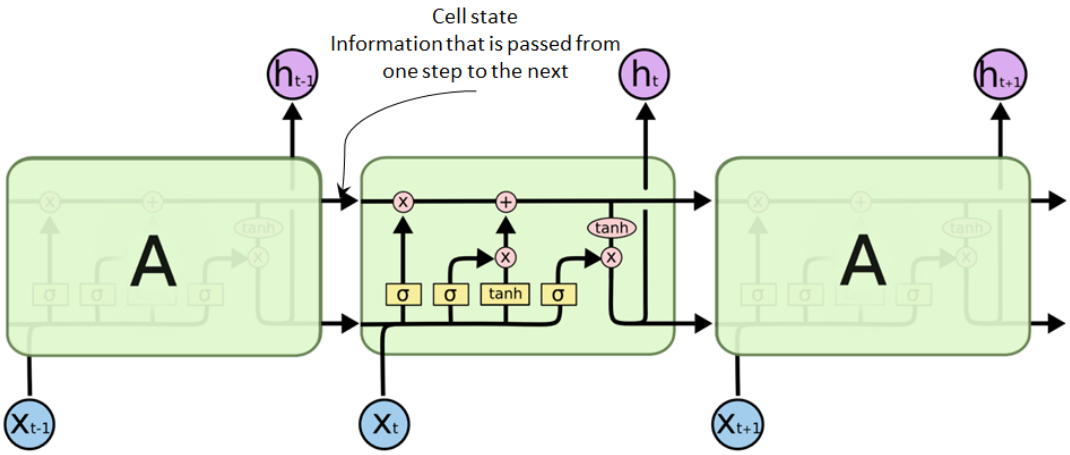


Figure 4.3: LSTM Networks

the figure 4.3. The cell state is like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates denoted by yellow boxes in the figure.

4.2 Available Data

City of Milan and surroundings are composed of a grid overlay of 10,000 squares, as can be seen in Figure 4.4. The area covered by the data is divided into a grid of 100 x 100 squares. Each square is of size 235m x 235m.

The shared datasets (Call Detail Records or CDRs) were created with a temporal aggregation of time slots of ten minutes, for a total of two months (November and December 2013).

Telecom Italia has recorded the following activities and a CDR is generated for each of these:

- Received SMS: a user receives an SMS
- Sent SMS: a user sends an SMS
- Incoming Call: a user receives a call
- Outgoing Call: a user issues a call

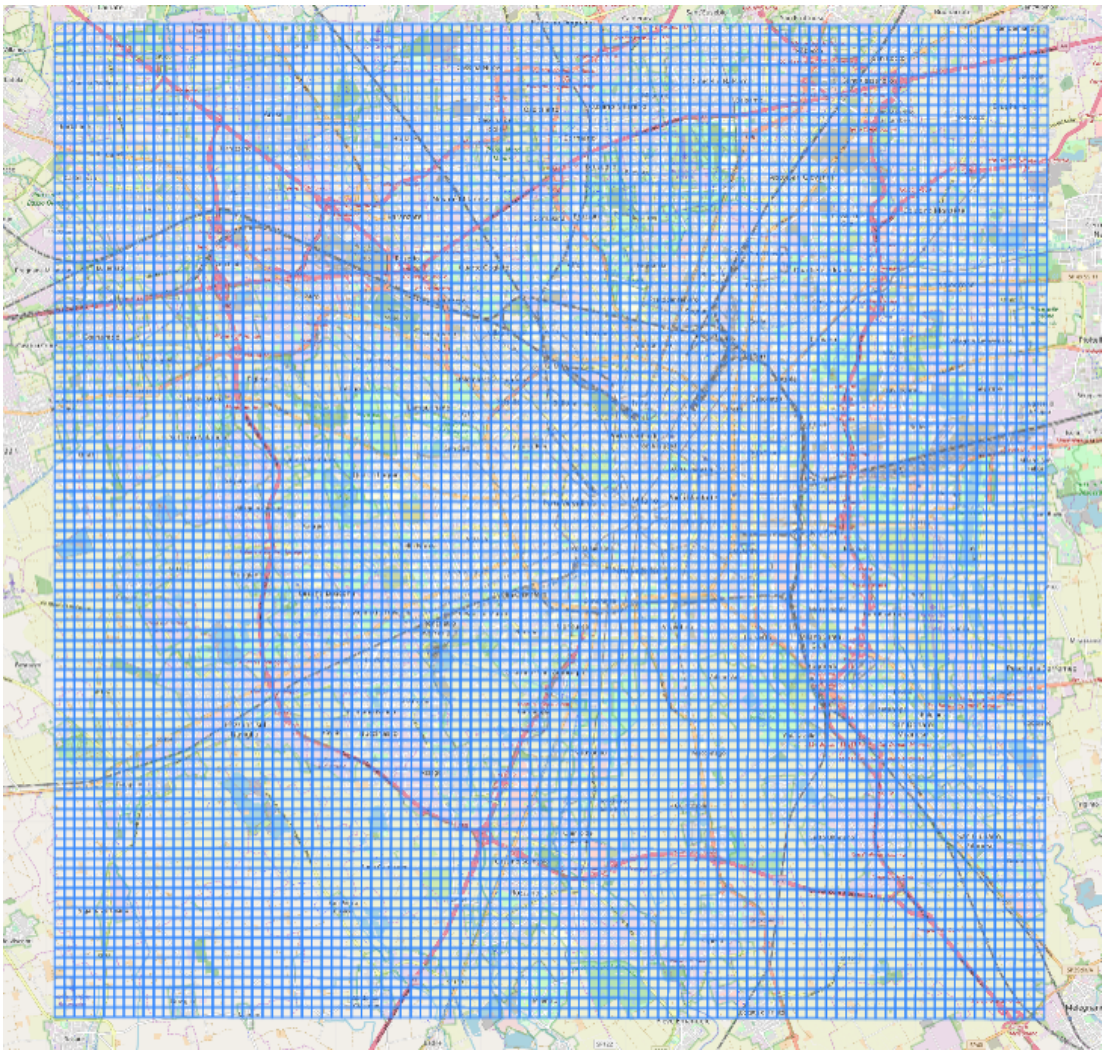


Figure 4.4: City of Milan and its surroundings

- Internet: a user starts an Internet connection or ends an Internet connection. During the same connection a CDR is generated if the connection lasts for more than 15 min or the user transferred more than 5 MB.

Figure 4.5 is a pictorial representation of the internet activity for the City of Milan and its surroundings as given by Telecom Italia. The x and y axis are geographical co-ordinates and the picture is made up of 10,000 cells. The number of records $S_i'(t)$ in the datasets is: $S_i'(t) = S_i(t)k$, where $S_i(t)$ is the actual number of records and k is a constant defined by Telecom Italia, which hides the true number of calls, SMS and connections. Here, the number of internet connections overall have been scaled to have values between 0 and 1, where 1 represents the maximum number of connections,

though the exact number is unknown.

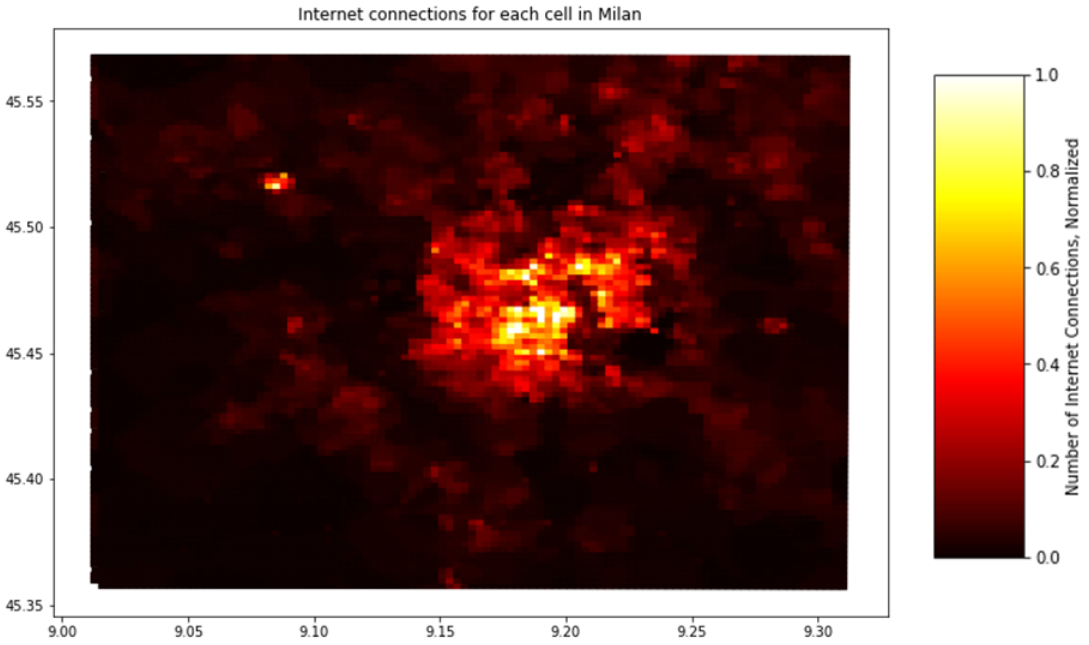


Figure 4.5: Internet Activity in Milan

4.3 Overall Model

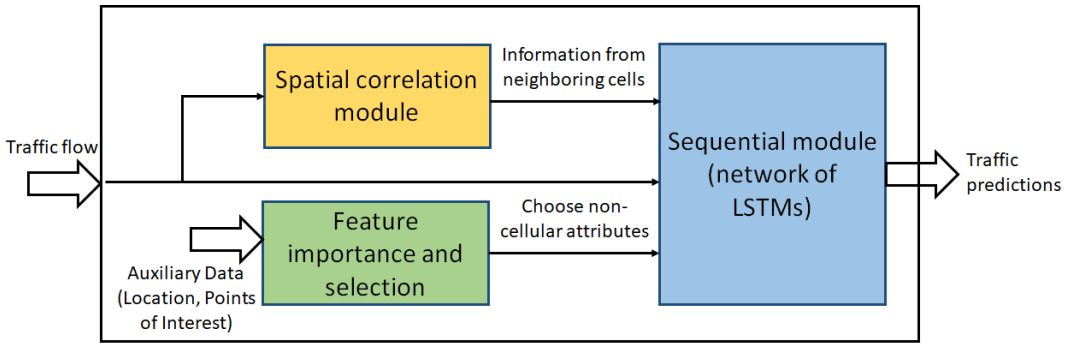


Figure 4.6: Proposed Framework

Figure 4.6 describes the proposed framework for cellular traffic prediction. Generally, the predicted quantity at a base station may be the number of calls made in the next n minutes, or the number of messages sent in the next n minutes, or any other quantity of interest. We break down the problem specifically to capture the complexity of cellular traffic patterns in a city and focus on three modules:

- A sequential module to do the temporal analysis, i.e., to model the time series of the system
- A spatial correlation module, which uses the proximity of neighboring cells and uses their time series as input features
- Auxiliary data such as Points of Interest, which model the behavior of special cases (airport, banks, restaurants etc)

4.3.1 Sequential module

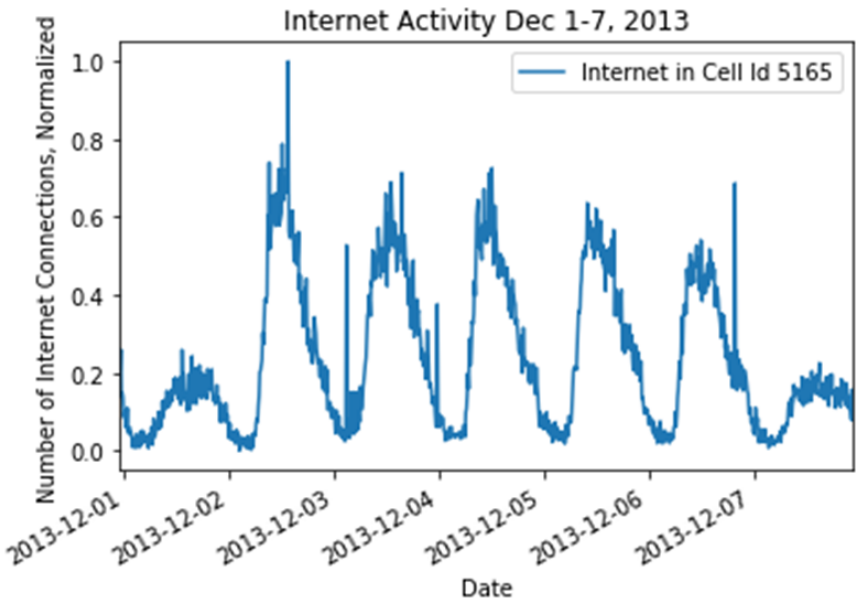


Figure 4.7: Internet activity data for one week

The traffic trends observed in a base station are found to be temporally correlated. For instance, consider the internet activity in Figure 4.7 that describes the traffic demands recorded in a base station over a period of 1 week. There is an apparent periodicity in the trends. It is also obvious that there is a pattern in the observed data. Extracting this pattern and the underlying temporal correlation characteristics can help in accurate prediction of traffic at the base station.

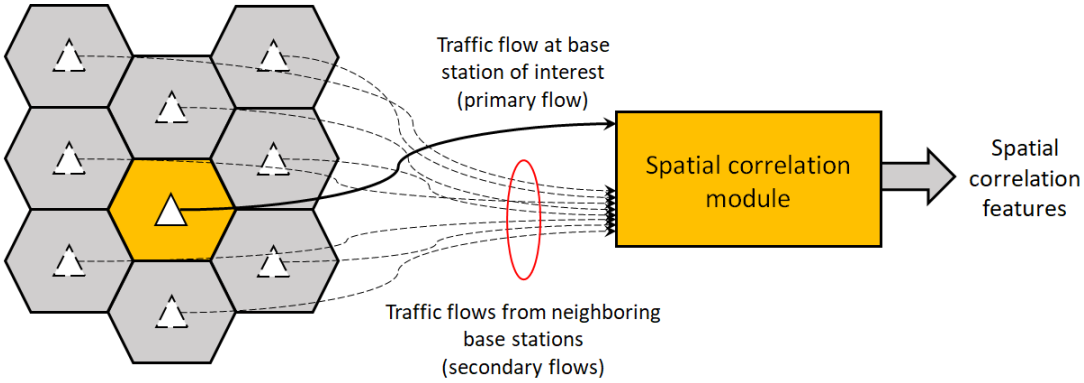


Figure 4.8: Leveraging spatial features

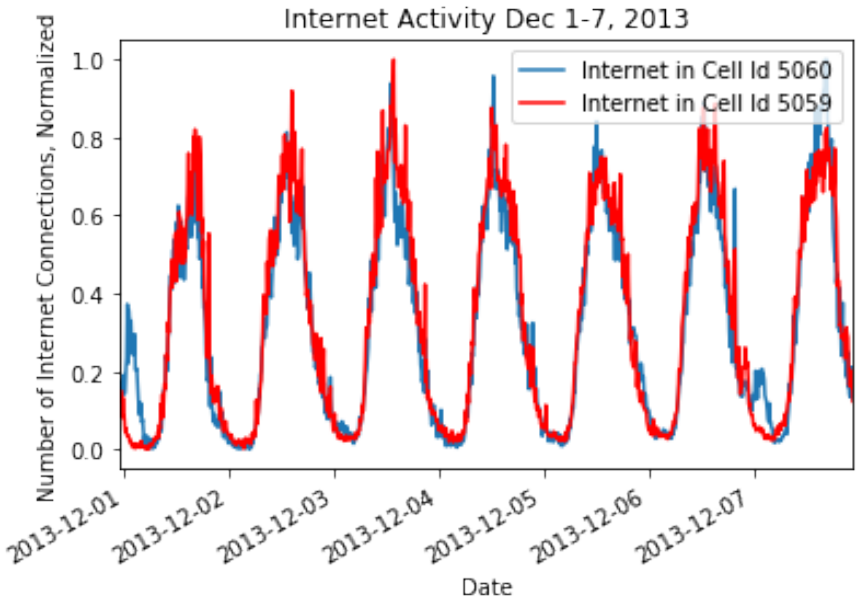


Figure 4.9: Internet activity in neighboring cells for one week

4.3.2 Spatial features

We propose the use of data from nearby cells to obtain a more complete picture of the data patterns of the cell in question. The highlighted cell in Figure 4.8 represents the base station for which we would like to make predictions. Figure 4.9 shows how similar the data of the primary cell (or highlighted cell) is compared to the data of the neighboring cell (or secondary flow). This suggests that the cell adjacent to the primary cell has similar traffic flow patterns. We intend to harness the similarity of traffic flows between the cells. Correlation would be a good indicator of how related the traffic flows

in surrounding cells are to the primary cell, and is further explored in section 4.4.3.

4.3.3 Auxiliary features: Non-cellular attributes

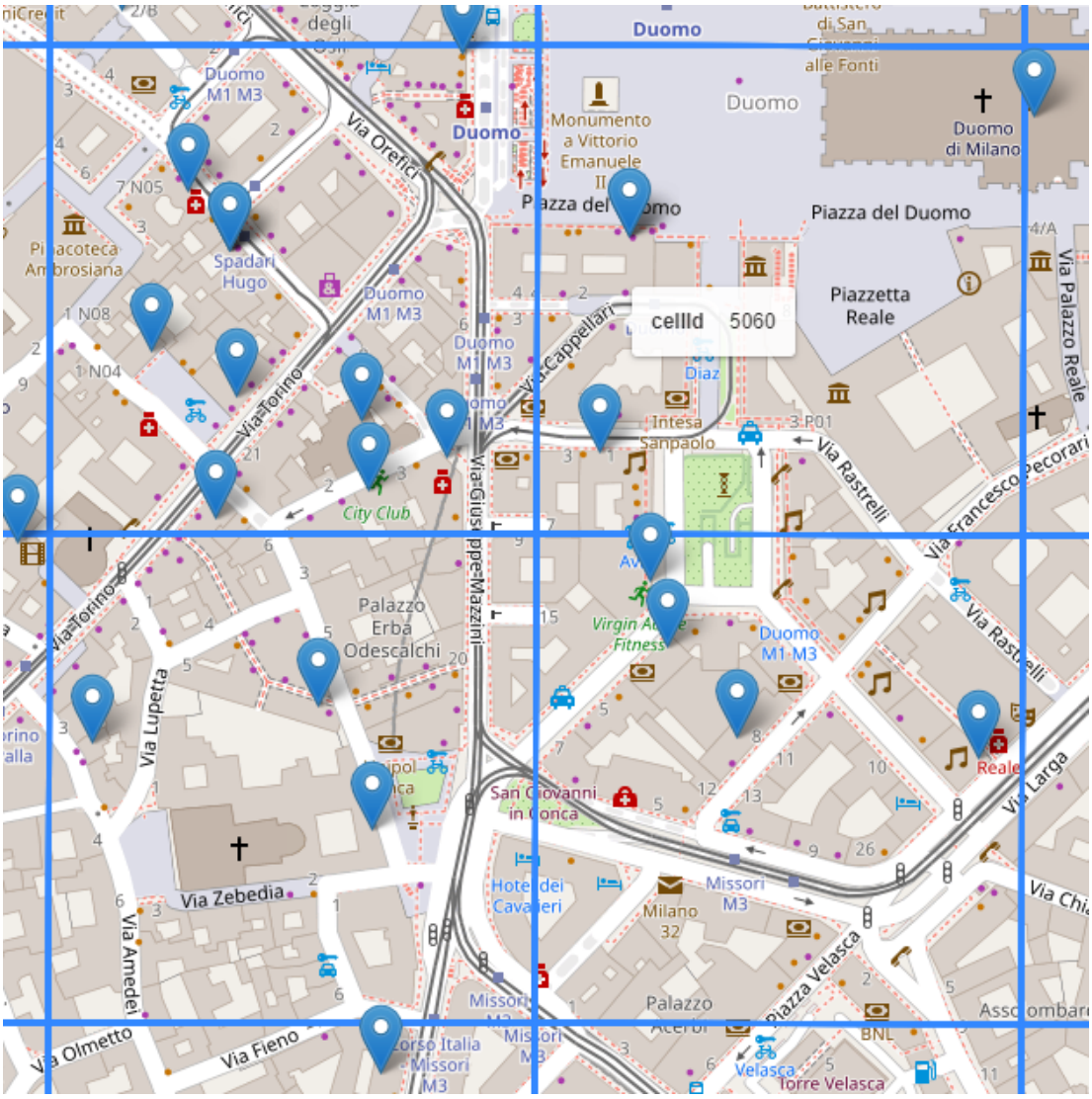


Figure 4.10: Points of Interest

We have identified the Places of Interest using a Google API as seen in Figure 4.10. The places seen here with a Google Maps Pin are cafes, restaurants, airport etc. The blue lines breaking the map into square cells correspond to a part zoomed out of the 100 x 100 grid presented in Figure 4.5.

4.4 Feature Selection

Two modules feed information into the network of LSTMs. We aim to identify the features outside of the temporal data of internet activity directly available which influence the prediction of cellular data.

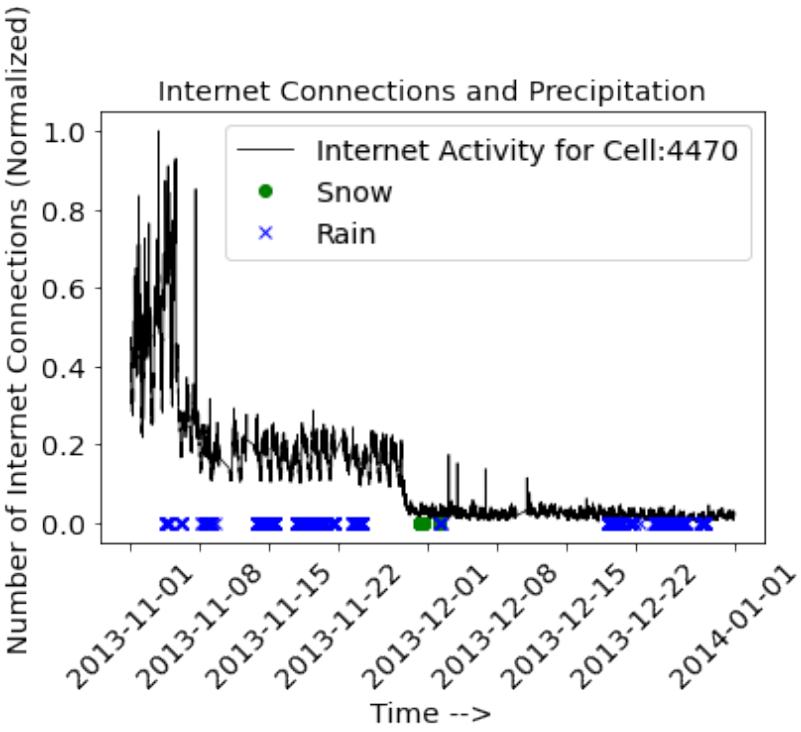
4.4.1 Precipitation

We explore the impact of weather, especially precipitation data, on cellular traffic patterns. Precipitation data for the City of Milan for November and December 2013 is provided in [78]. The weather data available provides the time of precipitation, whether it was rain or snow, and classified the intensity of precipitation as a measure of severity with a range of 1-5. Figures 4.11a and 4.11b mark the times of rain/snow in cells 4470 and 5283 and also plot the internet activity in these 2 cells.

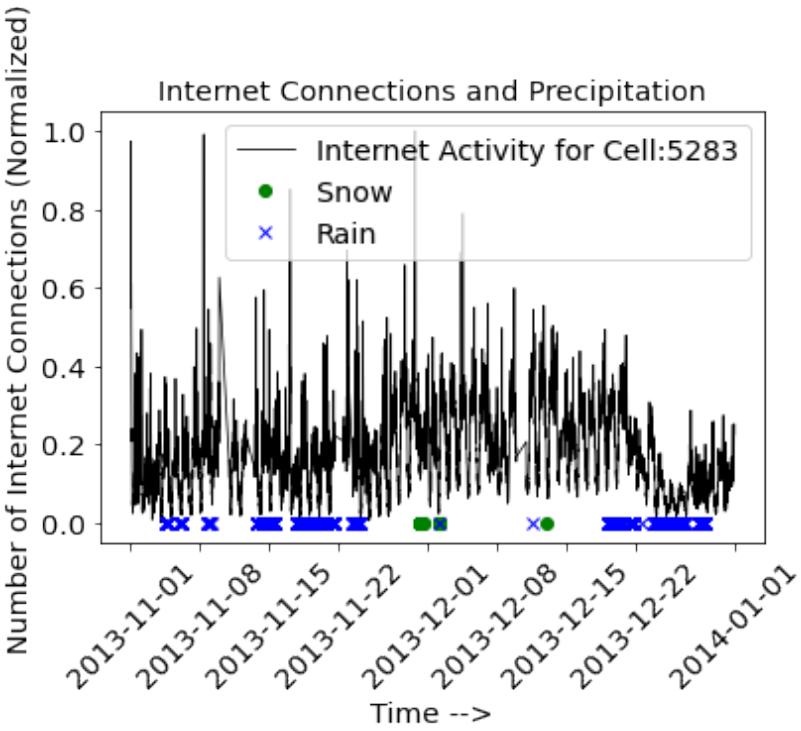
The point of interest identified by Google Maps in cell 4470 seen in Figure 4.12a is called "Acquarium Point", an aquarium and a pet shop. For our study, it is important to note that the main business identified is an enclosed area. We visualize the important features enabling the prediction of cellular traffic using a random forest, a simple ML technique. We can see in Figure 4.12b that the temporal features are far more likely to influence the prediction than the type of precipitation and the intensity of the precipitation.

To further understand the lack of impact of precipitation in Cell 4470, we study the weather patterns in Cell 5283, which is an open golf course, seen in Figure 4.13a. By observing the random forest output for feature importance in Figure 4.13b, we see that the weather has little to no impact on cellular traffic patterns.

It would seem that even in an open area location like a golf course, precipitation did not have a significant impact on the machine-learning problem. This could be because the data spans only the months of November and December, when golf might not be heavily played. This could also be because the number of instances of precipitation is very low; and did not affect the prediction. Therefore, before discarding the notion of including a non-cellular attribute such as weather, to a cellular traffic prediction, it

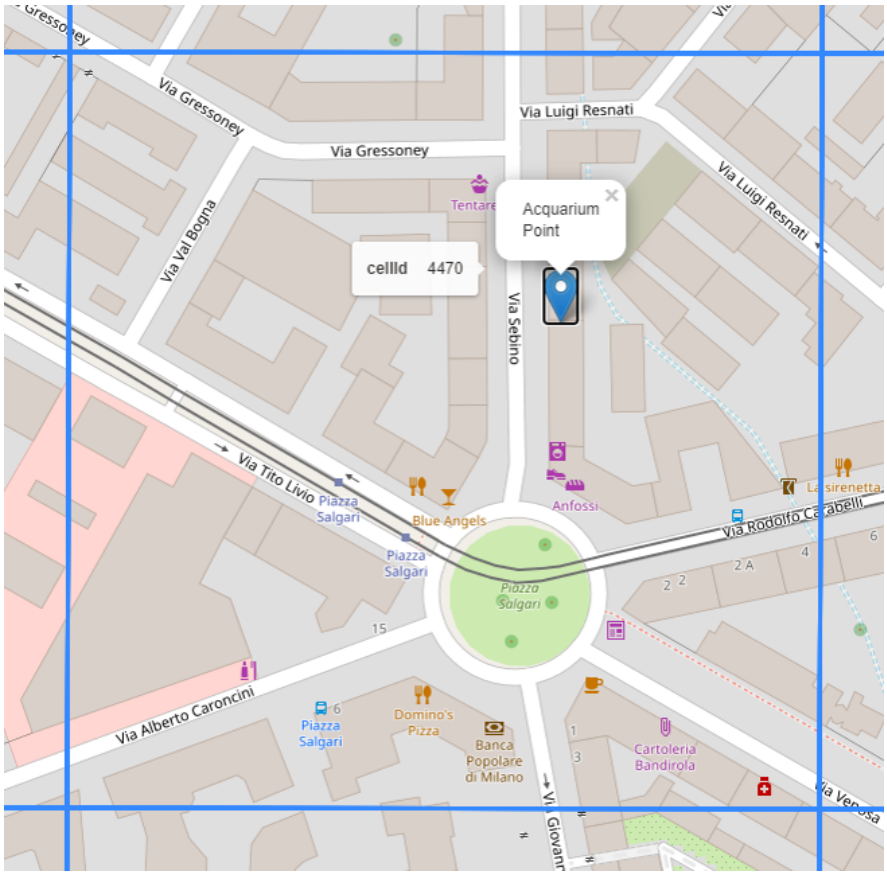


(a) 4470 precipitation

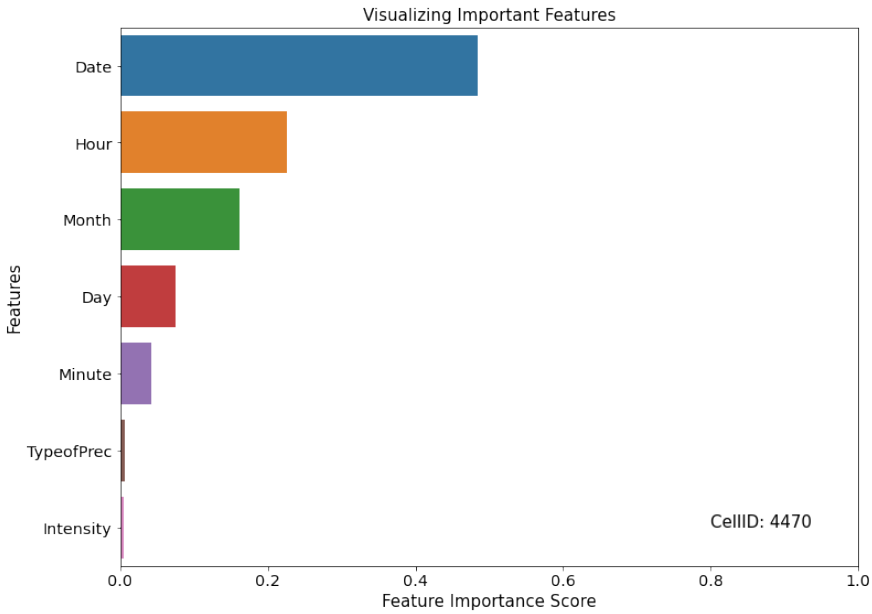


(b) 5283 precipitation

Figure 4.11: Feature: Precipitation



(a) Location of Cell 4470



(b) Feature Importance graph for Cell 4470, with weather data

Figure 4.12: Weather impact on Cellular Traffic: Cell 4470

would be interesting to observe the dataset for a longer period of time, preferably over many months.

4.4.2 Google API - Busy Hours

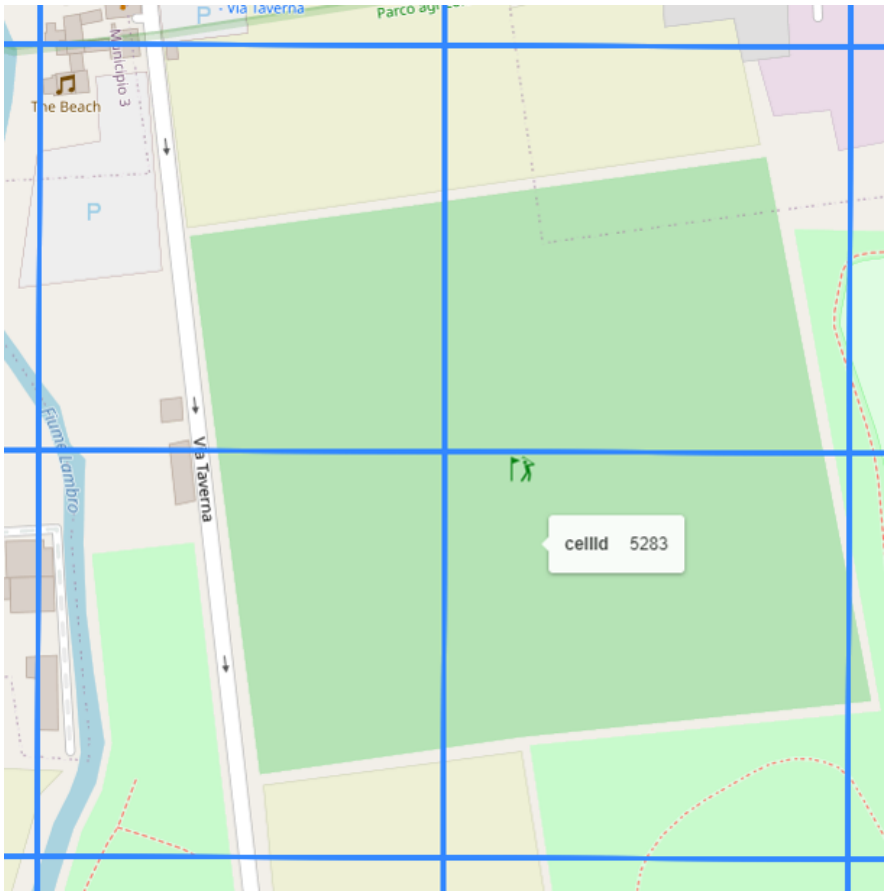
The Google API helps us locate the various points of interest in Milan. The API also provides information of the foot traffic of the locations, in the form of a feature called "Busy Hours". Figure 4.14a shows the Internet Data overlapped with the normalized busy hours data from the Google API. This data (for the points of interest) provided by Google is done by crowd-sourcing. Data is collected from users who use the locations on feature. The API data is more recent, averaged over data collected in 2019-2020. The data could also be effected by businesses being closed. The Google API data seems to have a moderate amount of weight in the prediction analysis, as seen in Figure 4.14b.

Independent data from Google API

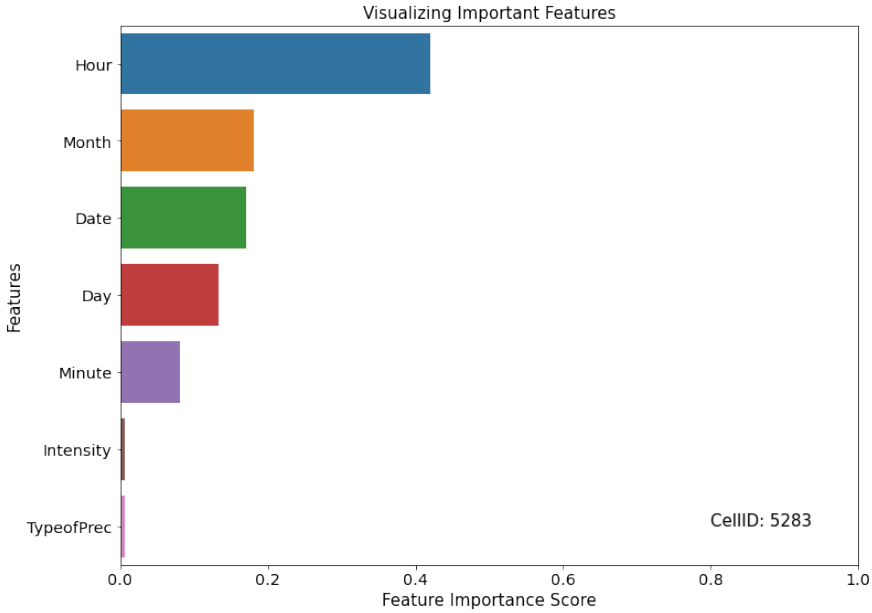
Data collection by the API is not dependent of type of carrier, therefore, in this case, not limited to Telecom Italia. The data obtained from Google is an independent feature as information from user generated tests show that the "locations on" feature consume about less than 2MB of data use for 20 minutes, or <1MB for 10 minutes, whereas, internet activity is updated only for 5MB or more for every 10 minutes by Telecom Italia. Another important reason for the independence of the feature is that the measurements from Google and Telecom Italia are not concurrent.

4.4.3 Neighboring Cells

Figure 4.15 represents the correlation of the internet activity of a cell with the internet activity of the cells surrounding it. As expected, the information available from the secondary flows is highly correlated with the primary data flow. We choose the data from one of the correlated cells and use this as an additional feature to help predict internet activity of the focused cell. It is to be noted that the correlation study has been performed at a dense urban location which is of interest in traffic prediction and

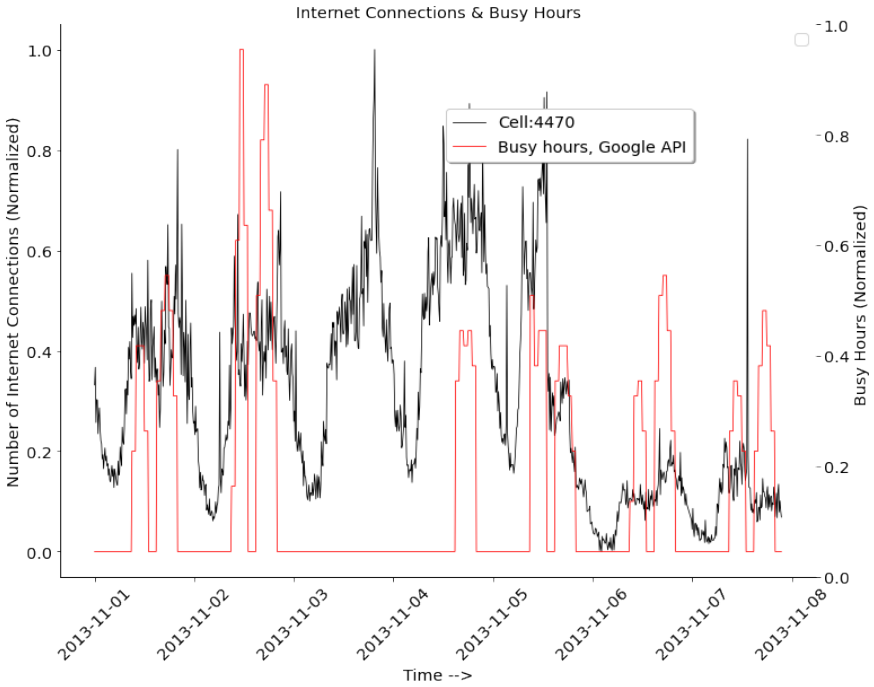


(a) Location of Cell 5283

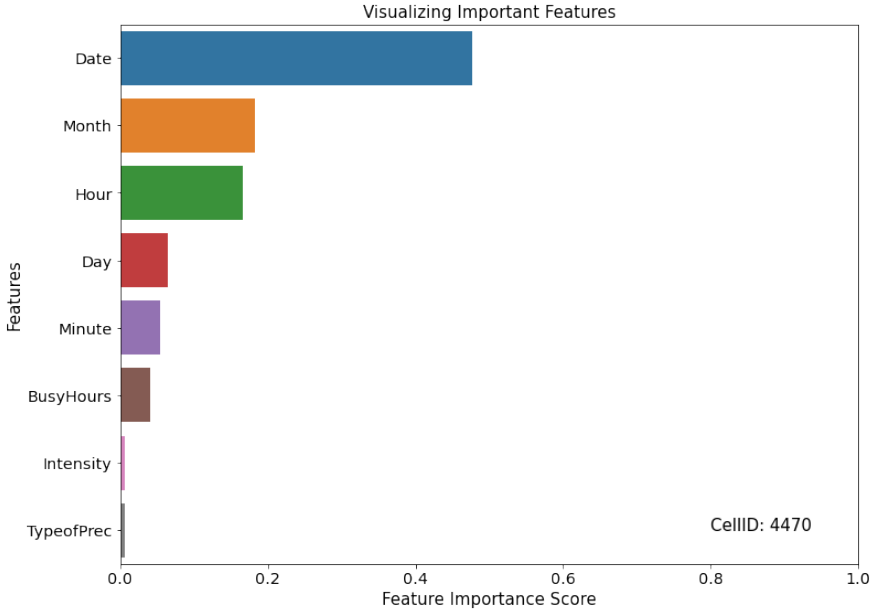


(b) Feature Importance graph for Cell 5283, with weather data

Figure 4.13: Weather impact on Cellular Traffic: Cell 5283



(a) Busy Hours from Google API



(b) Feature Importance

Figure 4.14: Google API

modeling. Correlation would exist in such spots, when multiple cells cover a similar location, but it might not be the case in a sparsely populated suburban area.

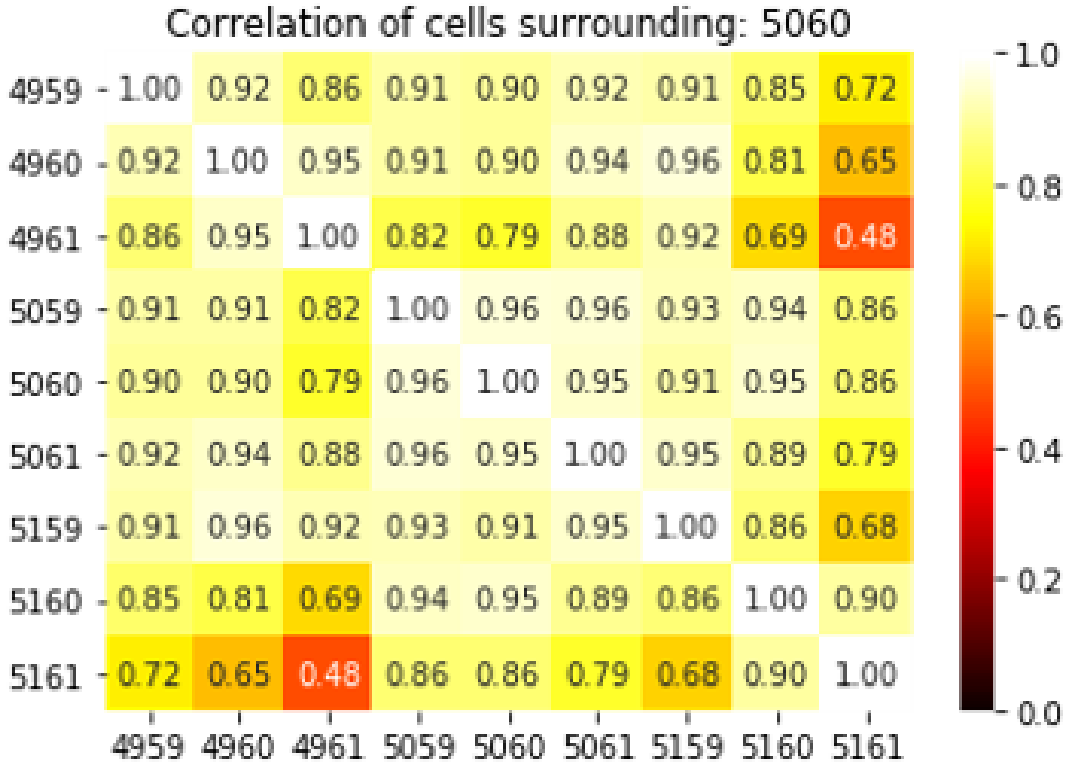


Figure 4.15: Spatial Correlation

4.5 Point-by-point Analysis

Each point to be forecasted represents internet activity for 10 minutes of time in the time-series, as noted in Section 4.2. The LSTM system has a day's worth of data (or 144 data points) to predict the next data instance. This predicted data point is compared to the corresponding test data point, and the difference is noted. To predict the next point in the sequence, the sliding window moves one data point to the right. The previously predicted (and corrected) data point is now part of the test data. This is an excessive amount of data used to predict a very short period in the time-series data. Figure 4.16 shows the predicted data points with and without features against the test data, as well as predicted values using ARIMA and exponential smoothing. The test data covers the last 20% of the dataset. To quantify the prediction, we

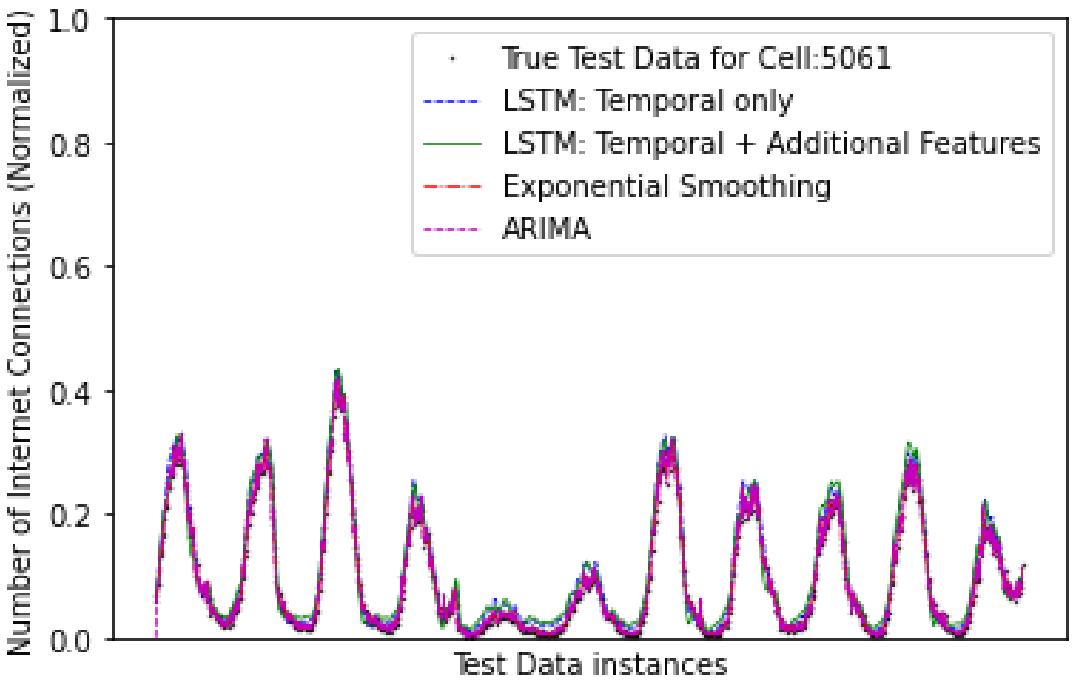


Figure 4.16: Point-by-point Analysis

use the Root Mean Square Error (RMSE). RMSE of predicted values $\hat{y}_t$ for times t of a dependent variable y_t, with variables observed over T times, is computed for T different predictions as the square root of the mean of the squares of the deviations and is given by $RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$. Another metric used is Mean Absolute Error (MAE), defined as $MAE = \frac{\sum_{t=1}^{T}|(\hat{y}_t - y_t)|}{T}$. Table 4.1 lists the different statistical and ML methods performed and the corresponding RMSEs and MAEs for one trial on data of Cell 5061. As the data is normalized, RMSE indicates the predicted value is within 2-5% of the test value. LSTM performs similar to the other one-step prediction statistical methods presented. As can be observed, the difference in RMSE for the prediction with and without features is negligible. The experiment is repeated over a few randomly selected data cells with similar results.

4.6 Trend Analysis

In real-world cellular traffic prediction, there is no application for the prediction of such a trivial amount of time-series data presented in Section 4.5. It is better to predict data

Methods	RMSE	MAE
LSTM(temporal data only)	0.02319	0.0156
LSTM(temporal+ additional features)	0.02434	0.0173
ARIMA	0.0128	0.0091
Exponential Smoothing optimized α	0.02471	0.0164

Table 4.1: RMSE and MAE values for point-by-point analysis

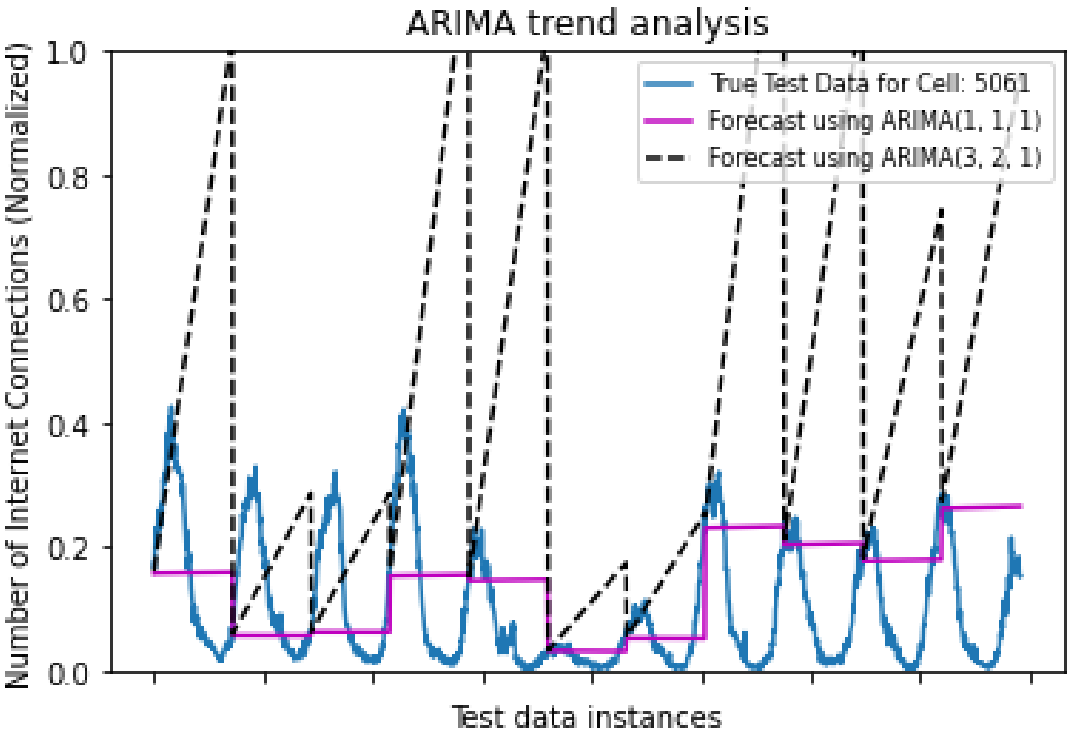


Figure 4.17: Trend Analysis: ARIMA

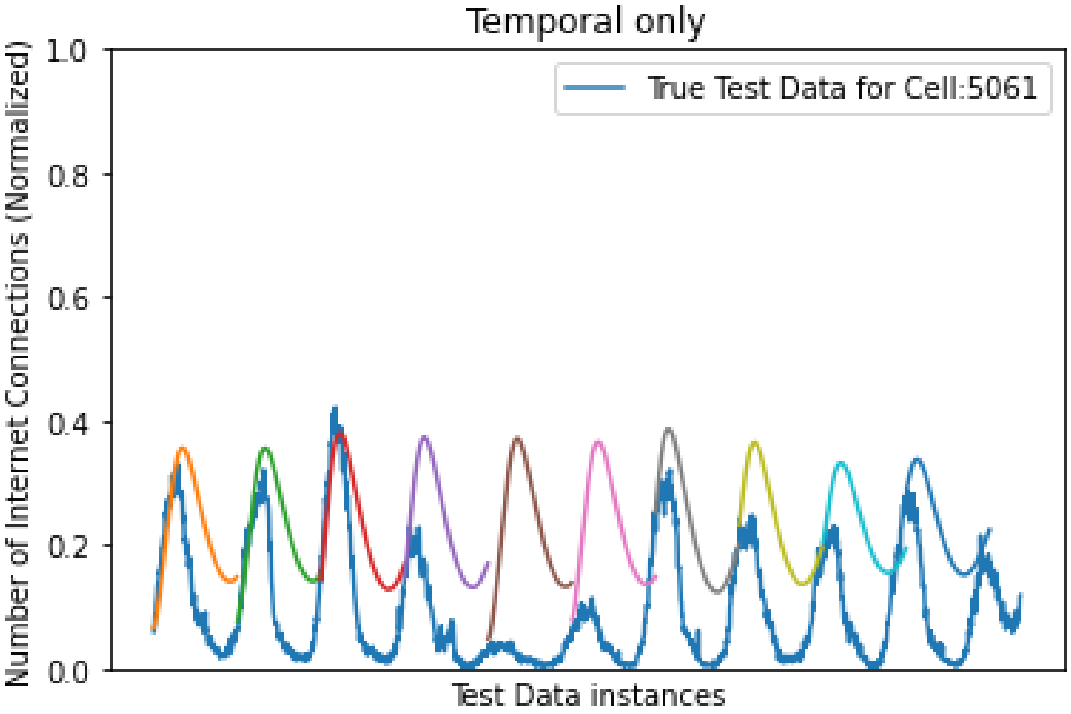


Figure 4.18: Trend Analysis: Only temporal data

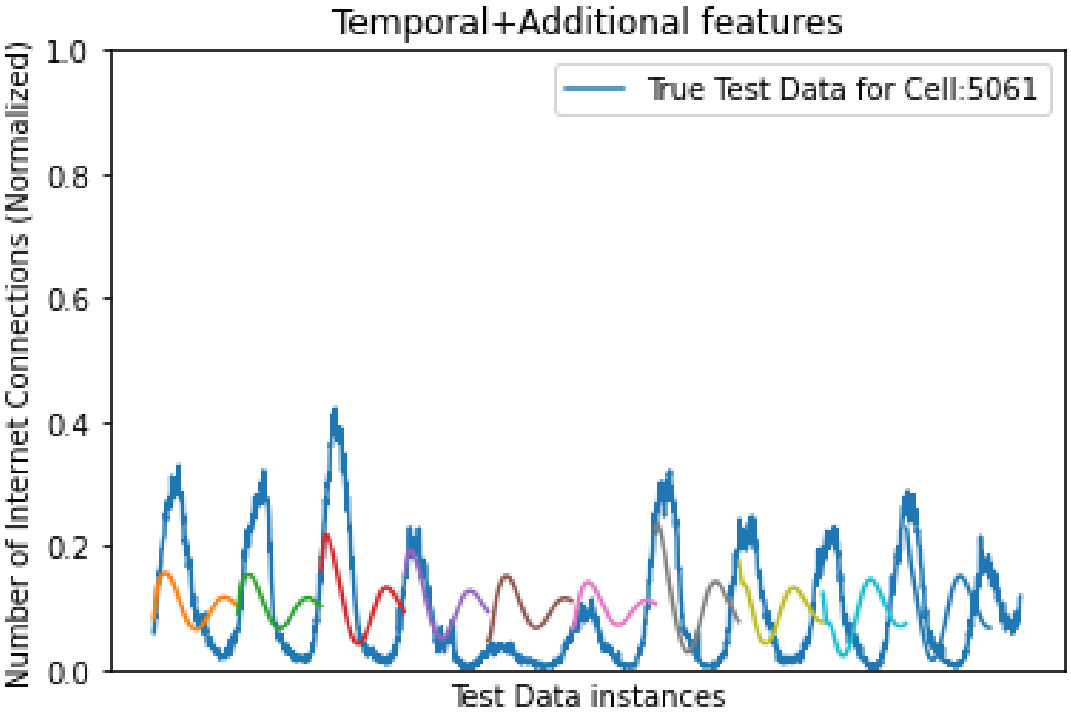


Figure 4.19: Trend Analysis: Temporal data with additional features

hours or days in advance. Figure 4.17 shows the univariate trend analysis using ARIMA. Figures 4.18 and 4.19 show the trend analysis using LSTM when the data (with and without additional features respectively) for 1 day is predicted in advance. The colored lines show the prediction for data points corresponding to 24 hours at a time. Figures 4.18 and 4.19 reveal the pattern in traffic with respect to the time of day, which is not evident in the ARIMA analysis. The RMSE and MAE details for one such trial are in Table 4.2. The data in the table indicates that augmenting the data with other cellular and non-cellular feature improves the RMSE and the MAE values, highlighting the importance of these features.

Methods	RMSE	MAE
LSTM(temporal data only)	0.1579	0.1365
LSTM(temporal+ additional features)	0.0919	0.0779
ARIMA(3,2,1)	0.4998	0.3744
ARIMA(1,1,1)	0.1238	0.09845

Table 4.2: RMSE and MAE values for trend analysis

4.7 Chapter Summary

This chapter described the effect of augmenting cellular traffic data with additional cellular and non-cellular attributes in traffic prediction. We augmented the existing traffic data with information from neighboring cells and non-cellular attributes such as weather and data from Google API and determined the importance of these features to forecast cellular traffic. We used LSTM, a deep learning architecture, to perform prediction on cellular traffic data (with and without additional features) available from the City of Milan and its surroundings. Two types of analyses were performed and compared with existing statistical methods: (i) a point-by-point or one-step prediction analysis; and (ii) a trend analysis. We observed that in the former case, the prediction was not impacted by the additional features where as the augmentation with additional features improved the trend analysis.

A deeper understanding of cellular traffic patterns and the importance of the chosen additional features could be obtained by studying datasets spanning over multiple years.

It would also be interesting to select other non-cellular attributes from non-traditional avenues and study their impact. Future work also includes predicting the traffic load of a cell by using its non-cellular features and data from neighboring cells only, eliminating the dependency on temporal data, which could be helpful at a failed or compromised base station.

Chapter 5

Conclusion and Future Work

In this dissertation, we have presented ML algorithms applied to different aspects of a wireless system. We have used radio attributes (such as channel capacity and received signal constellation) as well as non-radio attributes (such as busy period data from an open-source API) to improve the overall wireless system.

5.1 Impact of radio attributes on prediction

We first explored the impact of radio features on two traditional small-scale wireless system research problems.

5.1.1 Prediction of modulation and number of transmit antennas

Here, we presented an algorithm for jointly classifying the digital modulation and predicting the number of transmit antennas. We utilized only one receiver equipped with a single antenna which did not require any prior channel knowledge. We introduced a two-phase classifier system. In the initial phase, we used a discrete-wavelet transform on the received complex samples in order to separate the different modulation classes. To further classify the signals, we then used k-means clustering and k-nearest neighbor algorithms by exploiting the symmetry and relative distances of the constellation points. The performance of the classifiers at different stages of the algorithm showed a high accuracy at SNRs where the packets could be decoded.

5.1.2 Prediction of channel transition and state of environment

In this work, we proposed to use changes in the channel to trigger the channel assessment procedure at the Access Point (AP). This would lead to need-based CSI collection, to

improve aggregate throughput of the network. Using Software Defined Radios (SDRs) on the ORBIT testbed, the experiment showed high classification success of channel transition and a significant difference in error vector values for different types of channel variations.

5.2 Impact of non-radio attributes on prediction

We then studied the impact of non-radio attributes on a large-scale cellular network. The approach of using non-cellular features to augment the existing cellular system data-set proved to benefit forecasting over an extended period of time.

5.2.1 Prediction of cellular traffic

We considered traffic from neighboring cells and other non-cellular-traffic related attributes such as weather, busy period data from open-source API as features to augment the cellular traffic data and improve prediction. We augmented cellular traffic data and performed two types of analyses: (i) a point-by-point forecast of traffic and (ii) a forecast of traffic over an extended period and compared the prediction against existing statistical methods and found improvement in the long-term forecasting.

5.3 Future Work

Machine learning can be used in many ways to improve the existing wireless systems. On an individual radio level, it would be interesting to apply ML algorithms in other aspects of the wireless system such as channel estimation and channel encoding and decoding.

Based on our analysis, the impact of non-traditional attributes on prediction in wireless systems is significant. It would be beneficial to completely model cellular traffic at a base station (BS) using only non-cellular features and surrounding BS cellular data by studying feature importance of these attributes, which can help in resource allocation if the BS is compromised or fails. Planning in advance for such eventualities would ensure robustness of the mobile cellular network.

References

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[2] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, 2007.

[3] T. Erpek, T. J. OShea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, *Deep learning for wireless communications*, ser. Development and Analysis of Deep Learning Architectures. Springer, 2020, pp. 223–266.

[4] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: A comprehensive survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.

[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2016.

[6] J. Liu, R. Deng, S. Zhou, and Z. Niu, "Seeing the unobservable: Channel learning for wireless communication networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[7] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2016.

[8] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive mimo fingerprint-based positioning," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.

[9] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8440–8450, 2018.

[10] S. Xue, Y. Ma, N. Yi, and R. Tafazolli, "Unsupervised deep learning for mu-simo joint transmitter and noncoherent receiver design," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 177–180, 2018.

[11] A. Li, Y. Ma, S. Xue, N. Yi, R. Tafazolli, and T. E. Dodgson, "Unsupervised deep learning for blind multiuser frequency synchronization in ofdma uplink," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[12] J. Sun and Z.-M. Liu, "K-means clustering algorithm for full duplex communication," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2016, pp. 1555–1560.

[13] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*. IEEE, 2016, pp. 1–5.

[14] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[15] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, 2018.

[16] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2326–2330.

[17] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2017, pp. 1–6.

[18] S. Drner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[19] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[20] T. Kuber, D. Saha, and I. Seskar, "Blind classification of mimo wireless signals," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–5.

[21] T. Kuber, G. Sridharan, D. Saha, and I. Seskar, "Practical mu-mimo experiments using sdrs," in *Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conference on*. IEEE, 2017, pp. 517–522.

[22] T. Kuber, D. Saha, and I. Seskar, "Predicting channel transition for mu-mimo beamforming," in *2018 IEEE 5G World Forum (5GWF)*. IEEE, 2018, pp. 83–88.

[23] T. Kuber, I. Seskar, and N. B. Mandayam, "Traffic prediction by augmenting cellular data with non-cellular attributes," in *2021 IEEE WCNC (in progress)*.

[24] M.-R. Oularbi, S. Gazor, A. Aissa-El-Bey, and S. Houcke, "Enumeration of base station antennas in a cognitive receiver by exploiting pilot patterns," *IEEE Communications Letters*, vol. 17, no. 1, pp. 8–11, 2013.

[25] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[26] T. J. OShea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[27] V. Choqueuse, S. Azou, K. C. Yao, L. Collin, and G. Burel, "Modulation recognition for mimo communications," *Military Technical Academy Review*, vol. 19, no. 2, pp. 183–196, 2009.

[28] Y. A. Eldemerdash, O. A. Dobre, and M. ner, "Signal identification for multiple-antenna wireless systems: Achievements and challenges," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 1524–1551, 2016.

[29] Y. Liu, A. M. Haimovich, W. Su, J. Dabin, and E. Kanterakis, "Modulation classification of mimo-ofdm signals by independent component analysis and support vector machines," 2013, iD: arxiv1307.4430.

[30] M. R. Bahloul, M. Z. Yusoff, A. Abdel-Aty, M. Saad, and M. Al-Jemeli, "Modulation classification for mimo systems: State of the art and research directions," *Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena; Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena*, vol. 89, pp. 497–505, 2016.

[31] T. Kuber, *Automatic modulation recognition using the discrete wavelet transform*, 2013.

[32] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau, "Automatic modulation recognition using wavelet transform and neural networks in wireless systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 42, 2010.

[33] N. Ahmadi, "Using fuzzy clustering and ttsas algorithm for modulation classification based on constellation diagram," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 357–370, 2010.

[34] Z. Zhu, A. K. Nandi, and M. W. Aslam, "Approximate centroid estimation with constellation grid segmentation for blind m-qam classification," in *MILCOM 2013-2013 IEEE Military Communications Conference*. IEEE, 2013, pp. 46–51.

[35] G. Jajoo, Y. Kumar, S. K. Yadav, B. Adhikari, and A. Kumar, "Blind signal modulation recognition through clustering analysis of constellation signature," *Expert Systems with Applications*, vol. 90, pp. 13–22, 2017.

[36] J. Tian, Y. Pei, Y.-D. Huang, and Y.-C. Liang, "A machine learning approach to blind modulation classification for mimo systems," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[37] M. Mohammadkarimi, E. Karami, O. A. Dobre, and M. Z. Win, "Number of transmit antennas detection using time-diversity of the fading channel," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4031–4046, 2017.

[38] M.-R. Oularbi, S. Gazor, A. Aissa-El-Bey, and S. Houcke, "Exploiting the pilot pattern orthogonality of ofdma signals for the estimation of base stations number

of antennas," in *2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)*. IEEE, 2013, pp. 465–470.

[39] G. Qian, Z. Ruan, and J. Lu, "Joint modulation classification and user number detection for multiuser mimo-stbc systems," *Information*, vol. 7, no. 4, p. 70, 2016.

[40] K. Ho, W. Prokopiw, and Y. Chan, "Modulation identification by the wavelet transform," in *Proceedings of MILCOM'95*, vol. 2. IEEE, 1995, pp. 886–890.

[41] K. Maliatsos, S. Vassaki, and P. Constantinou, "Interclass and intraclass modulation recognition using the wavelet transform," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE, 2007, pp. 1–5.

[42] C. Zhendong, J. Weining, X. Changbo, and L. Min, "Modulation recognition based on constellation diagram for m-qam signals," in *Electronic Measurement & Instruments (ICEMI), 2013 IEEE 11th International Conference on*, vol. 1. IEEE, 2013, pp. 70–74.

[43] T. Caliski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[44] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1. Oakland, CA, USA, 1967, pp. 281–297.

[45] E. Perahia and M. X. Gong, "Gigabit wireless lans: an overview of ieee 802.11 ac and 802.11 ad," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 15, no. 3, pp. 23–33, 2011.

[46] A. Tulino, A. Lozano, and S. Verdu, "Mimo capacity with channel state information at the transmitter," in *Spread Spectrum Techniques and Applications, 2004 IEEE Eighth International Symposium on*. IEEE, 2004, pp. 22–26.

[47] J.-C. Guey and L. D. Larsson, "Modeling and evaluation of mimo systems exploiting channel reciprocity in tdd mode," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 6. IEEE, 2004, pp. 4265–4269.

[48] M. Rui, C. Ruo, W. Chaowei, and W. Weidong, "Downlink transmission scheme for massive mu-mimo system with reduced csi," in *Network Infrastructure and Digital Content (IC-NIDC), 2014 4th IEEE International Conference on*. IEEE, 2014, pp. 322–327.

[49] M. S. Sim, J. Park, C.-B. Chae, and R. W. Heath, "Compressed channel feedback for correlated massive mimo systems," *Journal of Communications and Networks*, vol. 18, no. 1, pp. 95–104, 2016.

[50] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO wireless communications*. Cambridge university press, 2007.

[51] A. Tolli and M. Codreanu, "Compensation of interference non-reciprocity in adaptive tdd mimo-ofdm systems," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 2. IEEE, 2004, pp. 859–863.

[52] T. Ji, C. Zhou, S. Zhou, and Y. Yao, "Low complex user selection strategies for multi-user mimo downlink scenario," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE.* IEEE, 2007, pp. 1532–1537.

[53] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Practical mu-mimo user selection on 802.11 ac commodity networks," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking.* ACM, 2016, pp. 122–134.

[54] T. Yoo, N. Jindal, and A. Goldsmith, "Multi-antenna downlink channels with limited feedback and user selection," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 7, 2007.

[55] N. Seifi, M. Coldrey, and T. Svensson, "Throughput optimization in mu-mimo systems via exploiting bs antenna tilt," in *Globecom Workshops (GC Wkshps), 2012 IEEE.* IEEE, 2012, pp. 653–657.

[56] P. Xia, M. Ghosh, H. Lou, and R. Olesen, "Improved transmit beamforming for wlan systems," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE.* IEEE, 2013, pp. 3500–3505.

[57] X. Ma, J. Wang, V. Marojevic, J. H. Reed, and Q. Gao, "Dynamic sounding for multi-user mimo in wireless lans," pp. 135–144, 2017.

[58] H. Yu and T. Kim, "Beamforming transmission in ieee 802.11ac under time-varying channels," *TheScientificWorldJournal*, vol. 2014, p. 920937, 2014, lR: 20140825; JID: 101131163; OID: NLM: PMC4135171; 2014/05/21 [received]; 2014/07/13 [accepted]; ppublish.

[59] R. Kudo, K. Ishihara, and Y. Takatori, "Measured channel variation and coherence time in ntt lab," *IEEE 802.11-10/0087r0*, 2010.

[60] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[61] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge university press, 2005.

[62] C. Zhao and R. J. Baxley, "Error vector magnitude analysis for ofdm systems," in *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on.* IEEE, 2006, pp. 1830–1834.

[63] "Wifi on steroids: 802.11ac and 802.11ad," *IEEE Wireless Communications, Wireless Communications, IEEE, IEEE Wireless Commun.*, no. 6, p. 30, 2013.

[64] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3. IEEE, 2005, pp. 1664–1669.

[65] E. K. Base, "Octoclock cda-2990 — ettus knowledge base,," 2017, [Online; accessed 18-January-2017]. [Online]. Available: https://kb.ettus.com/index.php?title=OctoClock_CDA-2990&oldid=3330

[66] D. Gesbert, F. Tosato, C. van Rensburg, and F. Kaltenberger, "Umts long term evolution: From theory to practice," 2009.

[67] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: understanding broadband wireless networking*. Pearson Education, 2007.

[68] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and experimental evaluation of multi-user beamforming in wireless lans," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 197–208.

[69] X. Xie, E. Chai, X. Zhang, K. Sundaresan, A. Khojastepour, and S. Rangarajan, "Hekaton: Efficient and practical large-scale mimo," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 304–316.

[70] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[71] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: single access point-based indoor localization," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 315–328.

[72] J. Xiong, K. Sundaresan, K. Jamieson, M. A. Khojastepour, and S. Rangarajan, "Midas: Empowering 802.11 ac networks with multiple-input distributed antenna systems," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 2014, pp. 29–40.

[73] M. Ettus and M. Braun, "The universal software radio peripheral (usrp) family of low-cost sdrs," *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, pp. 3–23, 2015.

[74] O. Bejarano, E. W. Knightly, and M. Park, "Ieee 802.11 ac: from channelization to multi-user mimo." *IEEE Communications Magazine*, vol. 51, no. 10, pp. 84–90, 2013.

[75] K. Sultan, H. Ali, H. Anwaar, K. P. Nkabiti, A. Ahamd, and Z. Zhang, "Understanding and partitioning mobile traffic using internet activity records data-a spatiotemporal approach," in *2019 28th Wireless and Optical Communications Conference (WOCC)*. IEEE, 2019, pp. 1–5.

[76] D. Corts-Polo, L. I. J. Gil, J. Calle-Cancho, and J.-L. Gonzlez-Snchez, "A novel methodology based on orthogonal projections for a mobile network data set analysis," *IEEE Access*, vol. 7, pp. 158 007–158 015, 2019.

[77] S. Zhang, S. Zhao, M. Yuan, J. Zeng, J. Yao, M. R. Lyu, and I. King, "Traffic prediction based power saving in cellular networks: A machine learning method," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 1–10.

[78] G. Barlacchi, M. D. Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific data*, vol. 2, p. 150055, 2015.

[79] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[80] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "Cellular traffic prediction and classification: a comparative evaluation of lstm and arima," in *International Conference on Discovery Science*. Springer, 2019, pp. 129–144.