KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Deep learning solutions for large scale wireless spectrum monitoring

Sreeraj Rajendran

Supervisors:
Prof. dr. ir. Sofie Pollin
Prof. dr. ir. Liesbet Van der Perre

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2019

Deep learning solutions for large scale wireless spectrum monitoring

Sreeraj RAJENDRAN

Examination committee:
Prof. dr. ir. Pierre Verbaeten, chair
Prof. dr. ir. Sofie Pollin, supervisor
Prof. dr. ir. Liesbet Van der Perre, supervisor
Prof. dr. ir. Dominique Schreurs
Prof. dr. ir. Lieven De Strycker
Dr. ir. Wannes Meert

Dr. ir. Vincent Lenders
(armasuisse)
Prof. dr. ir. Eli De Poorter
(Ghent University)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2019

© 2019 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Sreeraj Rajendran, Kasteelpark Arenberg 10 - box 2444 , B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

Four years and six months, that is what it took me to complete this thesis. For someone like me, who had never thought of pursuing formal studies after my bachelors, this entire PhD was a fun and memorable journey. Needless to say, there are numerous people including some awesome mentors and teachers who helped me to achieve this.

When most of my friends who pursue research in academia ask, how I am doing my PhD stress-free and with this much fun? I always had an impulsive answer, Prof. Sofie Pollin. I joined the Network Systems group when it was young and was fortunate enough to have weekly meetings with Sofie which clearly shaped my view about research. Let it be technical discussions or even personal problems, she was always there to help. I really admire the way she multitasks and gets things done. I am really grateful to Prof. Sofie Pollin for giving me an opportunity to pursue my research, trusting my abilities and giving me enough freedom to chase my research ideas.

Many thanks to Dr. Wannes Meert and Dr. Vincent Lenders for all the inspiring discussions and valuable comments on the papers we have co-authored. Wannes' attention to details and simplified explanations have always amazed me. I could easily recollect how Wannes changed my way of thinking and reasoning in the last three years and without a doubt I could say I have learned a lot from him. Vincent's unique way of identifying worthwhile problems and uncompromising professionalism always seemed close to magic to me. I still cherish some of the discussions that started during Electrosense meetings and vaguely ended by playing table-football, late-night in some bars.

I thank all the people in my jury, Prof. Liesbet Van der Perre, Prof. Dominique Schreurs, Prof. Lieven De Strycker, Dr. Wannes Meert, Dr. Vincent Lenders and Prof. Eli De Poorter for carefully reading my text and for their suggestions. The corrections and suggestions by the jury greatly improved the content of this thesis and made it more informative. I also thank Prof. Pierre Verbaeten for chairing the jury.

Throughout my life, a handful of people have recognized my potential and mentored me in the right direction. I am eternally grateful to Prof. Saravanan Vijayakumaran who recognized me among hundreds during an IIT, Bombay interview, gave me the chance to do my masters thesis with him and showed me the true value of freedom and choice in research. For me, the "push" for doing PhD, in the famous Joker quote "*Madness is like gravity, all it takes is a little push*", came from one of my best friends Deepak (Dr. Deepak!!). He is the one who pointed me to the vacancy in Prof. Sofie's group. A lot of my research ideas got refined as a result of the discussions with Deepak. Cheers, Deepak.

I am thankful to my old and new colleagues in the Network Systems group who made this research period fun-filled. I met Brecht on my first day at KU Leuven while waiting in front of Sofie's office and we continue our research fun since then. Beer drinking in the name of startup planning, making card games mandatory during lunch hours and whatnot, I sincerely blame him for all these mischiefs. Thanks to Tom for all those board game sessions, especially the cycling party, Bertold for showing me how good an engineer can be, Yuri for all those Spanish bad words, Alex for all those food descriptions and Ning and Cheng-Ming for all those Taiwan vs China fights. Sincere thanks to Hazem and Franco for bearing me. I thank Natalie for making my life easy with all those administrative work. I thank the system admins, people from CDE and secretaries for all their help.

I would like to extend my thanks to the Electrosense organization for creating a cool project and Markus and Roberto for making the Electrosense meetings exciting. I thank GNU Radio and the open source community for all the superb projects which lie in the very basis of this thesis work.

My stay in Leuven is supported by my extended family, a fairly big Indian community. Special thanks to Saanthettan (Dr. Sreeprasanth!!) and Krishna for all the guidance and for being there at difficult personal times. Thanks to my dear friends Freddy, Neetha, Shruti, Bismi, Dileep, Manisha, Arun and Archana for all the fun. Thanks to Sebin, Eldho, Nidhin, Govind, Mahadevan, Gain, Sujo, Sajith and Emil for all the support and being there for trips whenever I came back to Kerala.

This preface will be incomplete without mentioning my family and relatives. I am grateful to my dear parents for all the love and freedom that they gave me from the early years of my life. Thanks to brother Sanju for infinite fun (fights) throughout this life: we learned a lot of engineering together. Many thanks to my cousin brother Sujesh for believing in me at times of failure. Thanks to Hari uncle for showing me the wonderful world of wireless engineering in my childhood. Thanking Radhu (Radhika!!) will be like thanking myself. I often think about how dry my life will be without your love and randomness. This thesis is fully dedicated to you and our little one, Sriaan.

Abstract

The ability to represent complex thoughts into a structured symbolic set, a language, and communicate these thoughts over large distances is one of the greatest achievements of the human race. By the end of the 19th century, we had enhanced the speed and range of communication by playing with one of the fundamental forces of nature, the electromagnetic force. Since then there were countless applications that made use of radio spectrum for communication and control. An increase in the demand for this limited radio spectrum resulted in administrative licensing of the spectrum from the 1930s. Thus one of our precious natural resources, the radio spectrum got divided into various licensed and unlicensed frequency bands. The resulting diverse spectrum regulations across countries, the complex electromagnetic space and further chaotic wireless spectrum usages resulted in minimal real-time knowledge of the wireless spectrum usage across frequency, time and space.

Wireless spectrum usage knowledge is critical for various applications ranging from spectrum enforcement to dynamic spectrum sharing. This dissertation discusses the need for wireless spectrum monitoring and relevant usecases. New frameworks for obtaining wireless spectrum data at large scale and further extracting information from this collected data are detailed in this thesis.

First, a new crowdsourced collaborative spectrum monitoring initiative, Electrosense, is introduced, which improves the accessibility of spectrum data for the general public. Electrosense is a non-profit organization based in Switzerland, established by researchers from IMDEA Networks, Spain, Sero systems, Germany, armasuisse, Switzerland and Networked Systems, KU Leuven, which aims at improving the way how the radio frequency spectrum is used. The monitoring network consists of low-cost spectrum sensing nodes designed to monitor wireless spectrum on a large scale. The designed framework scales easily for large scale deployments, data collection, data storage and further data processing. Further, an open spectrum data as a service model is introduced to allow users to build diverse applications using the collected spectrum data.

Second, this thesis explores different techniques for extracting relevant information from the collected spectrum data. Detecting anomalous behaviour in the wireless spectrum is one of the important techniques for various usecases. It is a demanding task as wireless spectrum anomalies can take a wide range of forms from the presence of an unwanted signal in a licensed band to the absence of an expected signal. Spectrum Anomaly Detector with Interpretable FEatures (SAIFE), an Adversarial Autoencoder (AAE) based anomaly detector is proposed which can detect anomalies by formulating the anomaly detection problem as an unsupervised outlier detection problem using Power Spectral Density (PSD) data. The model achieves good anomaly detection and localization capabilities along with lossy PSD data compression up to 120X.

Third, after initial anomaly detection from the PSD data or spectrograms, a better understanding about the anomalous signal is achieved by classifying the signal modulation type. A new data-driven model for Automatic Modulation Classification (AMC) based on long short term memory (LSTM) is proposed which yields an average classification accuracy of close to 90% at varying SNR conditions ranging from 0dB to 20dB. To reduce the data communication overhead from distributed sensors, the feasibility of classification using averaged magnitude spectrum data and online classification on the low-cost spectrum sensors are studied. Furthermore, quantized realizations of the proposed models are analyzed for deployment on sensors with low processing power.

Fourth, the main issues with unsupervised anomaly detection are discussed in detail and a new framework is proposed for semi-supervised anomaly detection on the wireless spectrum data for crowdsourced sensors. The importance of user feedback in the anomaly detection process is investigated, to improve the performance of unsupervised anomaly detection. Furthermore, schemes for generalizing user feedback across sensors are also developed to close the anomaly detection loop.

Finally, the benefits of wireless spectrum awareness specific to a spectrum sharing scenario in cognitive radio are analyzed by organizing various open challenges. State-of-the-Art (SoA) solutions are evaluated in terms of achieving full spectrum awareness and network performance improvements with the provided situational awareness. The presented results emphasize the need for open spectrum sensing frameworks such as Electrosense that can provide online spectrum awareness for enhancing wireless network performance.

Beknopte samenvatting

Het vermogen om complexe gedachten te representeren in een gestructureerde symbolische set, taal, en deze te communiceren over grote afstanden is een van de grootste prestaties van de mensheid. Tegen het einde van de 19e eeuw verbeterde de snelheid en het bereik van communicatie door gebruik te maken van een van de fundamentele krachten van de natuur: de elektromagnetische kracht. Sindsdien ontstonden er talloze toepassingen die gebruik maakten van het draadloze spectrum voor communicatie en controle. De toename van de vraag naar deze beperkte natuurlijke bron zorgde voor de introductie van administratieve vergunningen voor het gebruik van spectrum in de jaren dertig. Een van onze kostbare natuurlijke rijkdommen, het radiospectrum, werd dus opgedeeld in gelicentieerde en niet-gelicentieerde frequentiebanden. De resulterende diverse spectrumregelgeving tussen landen, de complexe elektromagnetische ruimte en verder chaotisch draadloos spectrumgebruik resulteerde in een minimale real-time kennis van het draadloze spectrumgebruik over frequentie, tijd en ruimte. Kennis over het gebruik van draadloos spectrum is van cruciaal belang voor verschillende toepassingen van spectrumbeheer tot dynamisch spectrumgebruik. Dit proefschrift bespreekt de behoefte aan draadloos toezicht en relevante gebruiksscenarios. Nieuwe raamwerken voor het verkrijgen van draadloze spectrumgegevens op grote schaal en het verder extraheren van informatie uit deze verzamelde gegevens worden in dit proefschrift beschreven.

Ten eerste wordt er een nieuw collectief crowdsourcing-initiatief voorgesteld voor toezicht van het spectrum, Electrosense. Dit platfom verbetert de toegankelijkheid van spectrumgegevens voor het grote publiek. Electrosense is een non-profit organisatie gevestigd in Zwitserland, bestaande uit vier onderzoeksgroepen: IMDEA Networks, Spanje, Sero systemen, Duitsland, armsuisse, Zwitserland en Networked Systems, KU Leuven, met als doel de manier waarop het radiofrequentiespectrum gebruikt wordt te verbeteren. Het toezichtsnetwerk bestaat uit laaggeprijsde spectrumsensoren om draadloos spectrum op grote schaal te kunnen surveilleren. Het ontworpen kader schaalt eenvoudig tot grootschalige installaties, gegevensverzameling, gegevensopslag

en verdere gegevensverwerking. Verder wordt open spectrumdata als een servicemodel geïntroduceerd om gebruikers toe te staan verschillende applicaties te bouwen met de verzamelde spectrumgegevens.

Ten tweede onderzoekt dit proefschrift verschillende technieken voor het extraheren van relevante informatie uit de verzamelde spectrumgegevens. De detectie van afwijkend gedrag in het draadloze spectrum is een van de belangrijke technieken voor verschillende gebruiksscenarios. Het is een veeleisende taak omdat afwijkingen in het draadloze spectrum kunnen optreden in een breed scala aan vormen van de aanwezigheid van een ongewenst signaal in een gelicentieerde band tot de afwezigheid van een verwacht signaal. *Spectrum Anomaly Detector with Interpretabele FEatures* (SAIFE), gebaseerd op een *Adversarial Autoencoder* (AAE) afwijkingsdetector, wordt voorgesteld om afwijkingen te detecteren door het afwijkingsdetectie probleem te formuleren als een unsupervised outlier detectieprobleem gebruik makende van *Power Spectral Density* (PSD)-gegevens. Het model behaalt goede afwijkingsdetectie-en lokalisatiemogelijkheden samen met verliesgevende PSD-gegevenscompressie tot 120X.

Ten derde, na initiële afwijkingsdetectie van de PSD-gegevens of spectrogrammen, wordt er een beter begrip van het afwijkende signaal bereikt door de classificatie van de signaalmodulatie. Een nieuw datagedreven model voor Automatische ModulatieClassificatie (AMC) op basis van *long short term memory* (LSTM) wordt voorgesteld wat een gemiddelde nauwkeurigheid van classificatie oplevert tot bijna 90% in verschillende scenarios waar de signaal-tot-ruisvermogenverhouding varieert van 0dB tot 20dB. Om de extra datacommunicatiekosten van gedistribueerde sensoren te verminderen wordt de haalbaarheid van classificatie op basis van gemiddelde gegevens van het magnitudespectrum en online classificatie op de goedkope spectrumsensoren zelf bestudeerd. Bovendien worden de gekwantiseerde realisaties van de voorgestelde modellen geanalyseerd voor inzet op sensoren met een lage verwerkingskracht.

Ten vierde worden de belangrijkste problemen met *unsupervised* afwijkingsdetectie in detail besproken en een nieuw raamwerk wordt voorgesteld voor *semi-supervised* afwijkingsdetectie op de draadloze spectrumgegevens voor crowdsourcing-sensoren. Het belang van gebruikersfeedback in het afwijkingsdetectieproces wordt onderzocht om de prestaties van de detectie zonder toezicht te verbeteren. Ook zijn er ontwerpen voor het generaliseren van de gebruikersfeedback over sensoren ontwikkeld om de detectielus te sluiten.

Ten slotte worden de voordelen van een draadloze spectrumbesef, eigen aan het spectrumdelen in cognitieve radio, geanalyseerd door verschillende open uitdagingen te organiseren. State-of-the-Art (SoA)-oplossingen worden geëvalueerd hetzij in termen van het verkrijgen van volledig spectrumbesef, hetzij op basis van de verbetering van netwerkprestaties met de aangeboden

situatie-informatie. De gepresenteerde resultaten benadrukken de behoefte aan toegankelijke spectrummeetraamwerken zoals Electrosense die online spectrumbesef kunnen leveren om draadloze netwerkprestaties te verbeteren.

Abbreviations

3GPP 3rd Generation Partnership Program

AAE Adversarial Autoencoder

AMC Automatic Modulation Classification

API Application Programming Interface

ARI Adjusted Rand Index

AUC Area Under Curve

AWGN Additive White Gaussian Noise

CNN Convolutional Neural Network

COBRAS Constraint-based Repeated Aggregation and Splitting

DIY do-it-yourself

DSA Dynamic Spectrum Access

EDA Electronic Design Automation

FBMC Filter Bank Multicarrier

FFT Fast Fourier Transform

FPGA Field Programmable Gate Array

GNSS Global Navigation Satellite System

GPS Global Positioning System

GPU Graphics Processing Unit

HMM Hidden Markov Models

IEEE Institute of Electrical and Electronics Engineers

IFO Isolation Forest

IoT Internet of Things

ISM industrial, scientific and medical

IQ In-phase and Quadrature phase

LODA Lightweight on-line detector of anomalies

LOF Local Outlier Factor

LOS line of sight

LSTM Long Short Term Memory

OFDM Orthogonal Frequency Division Multiplexing

OSVM One class Support Vector Machine

PHY physical layer

PU Primary User

PSD Power Spectral Density

QoS Quality of Service

RAT Radio Access Technology

RCOV Robust Covariance

RF Radio Frequency

RFIC Radio Frequency Integrated Circuit

RFNoC RF Network on Chip

RNN Recurrent Neural Networks

ROC Receiver operating characteristic

SAIFE Spectrum Anomaly Detector with Interpretable FEatures

SDR Software Defined Radio

SNR signal-to-noise ratio

SoA State-of-the-Art

sps samples/symbol

SSDO Semi-Supervised Detection of Outliers

SU Secondary User

SVM Support Vector Machines

TDOA Time Difference of Arrival

TOA Time of Arrival

USRP Universal Software Radio Peripheral

VAE Variational Autoencoder

WSN Wireless Sensor Networks

Contents

Abstract iii

Beknopte samenvatting v

Abbreviations ix

Contents xiii

List of Figures xxi

List of Tables xxv

1 Introduction 1

1.1 Spectrum awareness: Need and use cases 2

1.1.1 Regulation and enforcement 2

1.1.2 Private spectrum owners 3

1.1.3 Wireless application stakeholders 4

1.1.4 End users 4

1.2 Traditional spectrum monitoring and analysis 4

1.3 SDR advances 6

1.4 Data driven models 6

1.5 Towards fully automated spectrum awareness 8
1.5.1 Data collection 8
1.5.2 Data analysis 8
1.6 Scope of the thesis 10
1.7 Methodology and contributions 11
1.8 Outline of the thesis 13
1.8.1 Chapter 2: Spectrum sensing network 13
1.8.2 Chapter 3: Unsupervised anomaly detection 13
1.8.3 Chapter 4: Signal classification 15
1.8.4 Chapter 5: Crowdsourced anomaly detection 15
1.8.5 Chapter 6: Situational awareness for spectrum sharing . 15
1.8.6 Chapter 7: Conclusion and future work 16

2 Spectrum sensing network 17

2.1 Introduction 18
2.2 Spectrum Data as a Service 21
2.2.1 Service Model 21
2.2.2 Application Programming Interface (API) 22
2.2.3 Security and Privacy 23
2.3 System Architecture 23
2.3.1 Design Goals and Electrosense Vision 23
2.3.2 Sensor 24
2.3.3 Controller 25
2.3.4 Backend 26
2.4 Spectrum Data Processing and Analysis 29
2.4.1 Live and Historical Spectrum Visualization 29
2.4.2 Dynamic Spectrum Access 30

2.4.3 Spectrum Cop 31
2.4.4 Localization 31
2.4.5 Spectrum decoding 32
2.5 Conclusion and Future work 33

3 Unsupervised anomaly detection 35
3.1 Introduction 36
3.2 Problem Definition 38
3.3 Models 39
3.3.1 Autoencoder and Variational Autoencoder (VAE) 40
3.3.2 Adversarial autoencoder (AAE) 40
3.3.3 SAIFE description 41
3.3.4 Implementation details 41
3.4 Datasets and model training 43
3.4.1 Model training 44
3.5 Anomaly detection 46
3.5.1 Detection scores 46
3.5.2 Performance comparisons 47
3.5.3 Anomaly localization 50
3.5.4 Anomaly detection in the wild 51
3.6 Signal compression and classification 55
3.6.1 Traditional Spectrum Representation 55
3.6.2 Non-linear data compression 56
3.6.3 Wireless signal classification 58
3.7 Conclusion and Future work 59

4 Signal classification 61
4.1 Introduction 62

4.2 Problem Statement 65
4.3 Modulation Datasets 67
4.3.1 RadioML dataset 67
4.3.2 Modified RadioML dataset 67
4.4 Model Description 67
4.4.1 LSTM primer 68
4.4.2 Model for complex signals 69
4.4.3 Baseline In-phase and Quadrature phase (IQ) model . . 70
4.4.4 Model training and testing 70
4.4.5 Implementation details 71
4.5 Results and discussion 71
4.5.1 Classification accuracy on RadioML dataset 71
4.5.2 Classification accuracy on modified RadioML dataset . 78
4.5.3 Learned representations 79
4.5.4 Effect of cell size and layer depth 79
4.6 Semi-supervised Models 81
4.7 Resource Friendly Models 83
4.7.1 Received averaged magnitude Fast Fourier Transform (FFT) signal 83
4.7.2 Electrosense dataset 84
4.7.3 Averaged Magnitude FFT model 85
4.7.4 Classification results 85
4.7.5 Quantized models 88
4.8 Conclusion and Future work 91

5 Crowdsourced anomaly detection 93

5.1 Introduction 94
5.1.1 Challenge 1: Finding an interpretable feature space . . . 95

5.1.2 Challenge 2: Detecting meaningful anomalies 95
5.1.3 Challenge 3: Anomaly knowledge sharing across sensors 97
5.1.4 Challenge 4: Incorporating user feedback 97
5.2 Related work 98
5.2.1 Unsupervised anomaly detection 98
5.2.2 Semi-supervised anomaly detection 99
5.3 Anomaly detection framework 99
5.3.1 Module-1: Data feature generator and unsupervised anomaly detection 99
5.3.2 Module-2: Anomaly feature generator 101
5.3.3 Module-3: Active clustering module 101
5.3.4 Module-4: Active anomaly detector 103
5.3.5 Users and expert feedback 104
5.4 Datasets 105
5.4.1 Feature extraction model training 106
5.4.2 Model retraining 107
5.5 Results and discussion 107
5.5.1 **Q1:** Feature extraction module performance 108
5.5.2 **Q2:** Clustering performance 108
5.5.3 **Q3:** Anomaly detection performance 113
5.6 Conclusion 114

6 Situational awareness for spectrum sharing 117

6.1 Introduction 118
6.2 Spectrum challenge-1 120
6.2.1 Challenge implementation 120
6.2.2 Primary user setup 122
6.2.3 Database and feedback 122

6.2.4 Challenge phases 123
6.2.5 Challenge metric 124
6.2.6 Secondary user algorithms 124
6.2.7 Challenge results and discussion 125
6.2.8 Winning solutions 125
6.3 Spectrum challenge-2 127
6.3.1 Challenge problem definition 128
6.3.2 Challenge scenarios 129
6.3.3 Primary User Setup 130
6.3.4 Database and Feedback 131
6.3.5 Challenge Phases 131
6.3.6 Challenge metric 132
6.3.7 Winning situational awareness approach 132
6.3.8 Winning agile SU approach 133
6.3.9 Challenge results and discussion 134
6.4 Future benchmarking 134
6.5 Conclusions 136

7 Conclusion and Future work 139

7.1 Research review 139
7.1.1 How to monitor the complex and dense electromagnetic space in a cost-effective manner? 139
7.1.2 How to detect and classify anomalous behaviors in the wireless spectrum efficiently? 140
7.1.3 How to incorporate user feedback to the entire monitoring framework to improve spectrum awareness? 141
7.1.4 How good is SoA research in utilizing the achieved spectrum awareness for improving wireless communication performance? 141

7.2 Future work 142
7.2.1 Machine learning 142
7.2.2 Anomaly geo-localization 143
7.2.3 Crowd engagement and sustainability 143
7.2.4 Sensor data quality 143

Curriculum Vitae **145**

List of publications **147**

Bibliography **151**

List of Figures

1.1 Methodology 12
1.2 Thesis outline 14

2.1 Spectrum sensing challenges 17
2.2 High-level overview of the Electrosense network 19
2.3 Electrosense sensors 24
2.4 Sensor architecture 26
2.5 Electrosense backend 27
2.6 TV band occupancy in Madrid and Trieste. 30
2.7 RSSI measured using a LoRa and Electrosense sensor. 32

3.1 Unsupervised anomaly detection 35
3.2 Autoencoder models 39
3.3 Model architecture for anomaly detection. 42
3.4 Sample signals 44
3.5 Error probability density functions 45
3.6 Anomaly detection accuracies for different anomalies 47
3.7 Anomaly detection accuracies for *wpulse* anomaly 48
3.8 Anomaly detection accuracies for *scont* anomaly 49

3.9 Anomaly detection accuracies for *randpulses* anomaly 50
3.10 Anomaly detection accuracies for different algorithms 51
3.11 ROC curves for different detection algorithms 52
3.12 ROC curves for SDR dataset 53
3.13 Localized anomalies for three different synthetic anomalies . . . 53
3.14 Detected anomalies from one of the Electrosense sensors 54
3.15 Learned bandwidth and position features in a semi-supervised fashion 55
3.16 Averaged accuracy Vs number of features 56
3.17 Receiver operating characteristic (ROC) curves with varying number of continuous features 57
3.18 Confusion matrix for the synthetic dataset 59

4.1 Fast signal classification 61
4.2 LSTM cell used in the hidden layers of the model. 68
4.3 Two layer Long Short Term Memory (LSTM) model for classification 69
4.4 Classification accuracy comparison of hyper-parameter optimized 2-layer amplitude-phase LSTM model with others on RadioML dataset. 72
4.5 Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at 18dB SNR 73
4.6 Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at 0dB SNR 73
4.7 Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at -8dB SNR 74
4.8 Classification accuracy for different input sample lengths 74
4.9 Classification accuracy of amplitude-phase LSTM model on modified RadioML dataset for 4 samples/symbol 75

4.10 Classification accuracy of amplitude-phase LSTM model for non-trained data lengths on modified RadioML dataset with 4 samples/symbol. 75

4.11 LSTM temporal activations for a QAM64 input vector 76

4.12 Gate saturation plots for LSTM for the same a QAM64 input vector 77

4.13 Classification accuracy of single layer amplitude-phase LSTM model for different cell size on RadioML dataset. 78

4.14 Classification accuracy of two layer amplitude-phase LSTM model for different cell sizes on RadioML dataset. 80

4.15 Classification accuracy of three layer amplitude-phase LSTM model for different cell sizes on RadioML dataset. 80

4.16 Generator and discriminator details. 81

4.17 Semi-supervised classification results 82

4.18 Confusion matrix for technology classification on Electrosense dataset. 86

4.19 t-SNE plot for magnitude FFT output on radioml dataset. . . . 87

4.20 Classification accuracy of two layer quantized models on RadioML dataset. 90

5.1 Anomaly preferences 93

5.2 Anomaly detection and clustering framework 96

5.3 Active clustering 102

5.4 SSDO anomaly detection example 104

5.5 Averaged anomaly detection accuracies for n-sigma cutoff $n = 2$ 109

5.6 Averaged anomaly detection accuracies for n-sigma cutoff $n = 4$ 109

5.7 Averaged anomaly detection accuracies for constant false alarm rate 110

5.8 Average clustering ARI with queries answered from all four sensors. 110

5.9 Average clustering ARI with queries answered from only one sensor for different SNRs. 111

5.10 Average clustering ARI with queries answered for different sensor counts. 112

5.11 Average clustering ARI with queries answered with different incorrect answering probabilities (P_i). 112

5.12 Average anomaly detection accuracy with correct labeling . . . 113

5.13 Average anomaly detection accuracy with incorrect cluster and anomaly labeling probability $P_i = 0.05$. 114

6.1 Perfect spectrum awareness 117

6.2 Spectrum challenge setup 121

6.3 Primary user setup 123

6.4 Primary and Secondary user performance 127

6.5 Primary user setup 129

6.6 Final scores 135

List of Tables

1.1 Software Defined Radio (SDR) comparison. 7

3.1 Synthetic signal dataset parameters. 43

3.2 Synthetic anomaly dataset parameters. 43

3.3 SDR and Electrosense dataset frequency bands. 44

3.4 Band classification accuracy and reconstruction errors on the test data of different datasets. 58

4.1 RadioML2016.10a dataset parameters. 66

4.2 Modified complex RadioML dataset parameters. 66

4.3 Averaged magnitude FFT dataset parameters. 85

4.4 Classification accuracy on Electrosense dataset for varying layer depths and cell numbers. 86

4.5 Memory size and number of multiplications for the discussed models (excluding nonlinear activations). 89

4.6 Inference engine performance in number of classification per second on different platforms. 89

5.1 Data and anomaly feature module layers. 100

5.2 Electrosense dataset frequency bands. 107

6.1 Secondary user features 126

6.2 Challenge scenarios: Different Primary User (PU) scenarios for the challenge. 129

Chapter 1

Introduction

> The first step in solving a problem is recognizing there is one.
>
> – Aaron Sorkin, *The newsroom*

Radio spectrum is one of our most precious and widely used natural resources. With the advent of new wireless communication technologies, spectrum usage has become very complex resulting in airwave congestion and other interference issues [1]. Diverse spectrum regulations across countries have also contributed to this chaotic spectrum usage when non-standardized wireless devices cross country borders. The fifth generation wireless technologies are envisioned to be more diverse and heterogeneous. Densified small cells are becoming fundamental for these high throughput and low latency networks to support the new high bandwidth applications such as live data streaming, video calls and vehicle-to-vehicle communications [2]. Such a dense and heterogeneous deployment makes the enforcement and management of the wireless spectrum usage difficult. Considering the current spectrum usage scenario, it is quite natural for anyone to ask the following questions:

- Who is going to monitor this complex and dense electromagnetic space on a large scale and how to do it cost-efficiently?
- Even if we know how to sample the Radio Frequency (RF) spectrum cost-effectively, how can we analyze the samples and translate them to spectrum knowledge?
- Are there enough applications that demand spectrum awareness?

- Can the State-of-the-Art (SoA) radios currently make use of this wireless spectrum awareness for improving the network performance?

These questions can be translated to three major processes: obtain spectrum data, extract information and use information. Before elaborating on how we can achieve spectrum awareness, that is obtaining data and extracting information, let us first analyze whether there are enough applications and use cases that demand and can exploit extracted spectrum information.

1.1 Spectrum awareness: Need and use cases

Wireless spectrum awareness in general refers to the wide range of wireless information ranging from radio signal power levels, time-frequency occupancy, specific signal features, wireless channel/environment information or the device details connected to a particular network. Such spectrum awareness is important for various stakeholders, for instance spectrum regulatory agencies, private spectrum owners, defense applications, spectrum sharing companies, wireless system integrators, radio amateurs and even end-users.

1.1.1 Regulation and enforcement

Most countries consider wireless spectrum as an exclusive property of the state which has resulted in specialized regulations and dedicated government supported enforcement agencies. Currently these agencies monitor the spectrum mostly manually and perform inspection based on user complaints. Easily available illegal wireless jammers or low-cost Software Defined Radio (SDR) devices, which are capable of generating custom wireless signals, with easily available do-it-yourself (DIY) instructions, are making this enforcement and inspection difficult. Furthermore, illegal repeaters, which are used to boost mobile coverage, can adversely affect the cell planning of mobile operators, resulting in poor coverage and dropouts [3]. Unintentional and intentional jamming of localization services such as Global Navigation Satellite System (GNSS), which is used for a wide range of applications such as automated vehicle navigation, airplane landing procedures and maritime vessel tracking, is also increasing with the availability of easy jamming devices [4,5]. In addition, wireless bands are getting closer in the frequency domain making problems such as adjacent channel interference difficult to detect and understand.

For instance, as per Belgian Institute for Postal services and Telecommunications (BIPT) 2017 annual report [6], 440 interference cases have been reported only

in Belgium. Most of these complaints that were logged by the operators are caused by short range devices. These devices include products from the United States that work with different regulations. Further, 497 monitoring visits, to inspect spectrum regulations, were done by BIPT based on user complaints. Also inspections have resulted in seizure of numerous equipment: 10370 devices out of 15007 seized at border control. All these results are currently with suboptimal manual scanning with a few stationary and on request mobile monitoring stations.

Furthermore, future spectrum sharing usecases envisioned in the next generation wireless networks (5G and beyond) require complex dynamic regulations which is impossible to be enforced with manual efforts. Automated monitoring, real-time analysis and detection of anomalous behavior in the spectrum is becoming more crucial for spectrum enforcement than ever before.

1.1.2 Private spectrum owners

Many private organizations, such as telecom operators, service providers like Flemish Radio and Television Broadcasting Organisation (VRT) and airports, own licensed spectrum for daily operations. Most of these companies have bought spectrum in different bands with prices that easily exceed 500,000 euros per MHz per year. For these operators, interference detection is critical to provide specified Quality of Service (QoS). Intentional and unintentional interference ranging from private signal booster deployments to jammers affects the cell planning of almost all telecom operators, resulting in coverage and data outage. The operators lose 15 billion dollars yearly to network outages worldwide [7] resulting in a failure to fulfill agreed service layer agreements (SLAs) in terms of data throughput or QoS. Airline operators see costs easily go up when a plane cannot land due to occasional jamming of the Global Positioning System (GPS) or other radar bands [8]. Such jamming also creates issues for services relying on GPS for adjusting oscillators for time keeping, such as mobile base stations and stock markets. These jamming events occur quite often as per the STRIKE3 report [5]. The compensation given to travelers, flight fuel costs, infrastructure costs and inconvenience caused to the people itself result in loss of millions of euros across different airports. In addition, these kind of interference and intentional jamming are life threatening at the first place demanding real-time spectrum awareness for threat mitigation.

1.1.3 Wireless application stakeholders

This group includes companies relying heavily on wireless communication in unlicensed bands, e.g. companies using industrial, scientific and medical (ISM) bands or companies working on cognitive radio architecture with shared spectrum (e.g. Federated Wireless, Dense Air). Companies making use of ISM bands are widespread because of the growing interest in long range communication in the 868 MHz band, especially in Europe [9]. Expected industry value was estimated to reach 28 billion euros [10]. Monitoring interference in these bands to allow agile spectrum use can be critical for such companies [11]. For example, a parking coordination firm with vehicle detection sensors working on LoRa might be interested in ISM band statistics to enable the best communication bands for their sensors in a particular region. In addition, spectrum knowledge is useful for detecting cause of sensor communication malfunctions.

1.1.4 End users

General public is mainly interested in having a network for radiation monitoring, optimizing their indoor WiFi network, detecting fake base stations and localization of jamming, for instance GPS/GNSS jamming used for navigation. Amateur radio users also show keen interest in expanding and monitoring their frequency bands in an automated and remote fashion.

1.2 Traditional spectrum monitoring and analysis

There have been a few spectrum monitoring initiatives in the past both from academia and industries. A few academic initiatives are funded by the National Science Foundation (NSF) projects in the USA [12–14]. A metro scale spectrum observatory [15], an initiative from University of Washington is also sponsored by NSF. Radiohound [16] is another spectrum monitoring initiative concentrating on mobile environment monitoring using low-cost SDR. Some other examples include, Microsoft Spectrum Observatory[1] that allows to sense the spectrum using expensive sensors, Google spectrum[2] for measurements on TV white-spaces and the IBM Horizon[3] project that proposed a generic decentralized architecture to share Internet of Things (IoT) data. While Google Spectrum and

[1] http://observatory.microsoftspectrum.com/
[2] https://www.google.com/get/spectrumdatabase/
[3] https://bluehorizon.network/documentation/sdr-radio-spectrum-analysis

Blue Horizon fail to cover a large part of the spectrum as they are application specific deployments, Microsoft observatory on the other hand fails to enable a large scale sensor deployment mainly due to the cost of the sensing stations. All these solutions failed in deploying a sustainable large scale spectrum monitoring framework mainly due to the high cost of the solutions and lack of clear application benefits.

On the algorithm side, for wireless spectrum representation, in spite of the popularity of various lossy and lossless compression algorithms in image and video processing communities, only a few fine tuned compression algorithms are available in literature. For example, in [17] the authors presented a compression algorithm based on Chebyshev polynomials. The authors in [18] presented a method to separate spectrum noise and other relevant signals specific to L-band satellite signals and then did separate compression to achieve better results when compared to JPEG standards. All these methods are very specific and lack the compression flexibility when the input data is in multiple formats such as Power Spectral Density (PSD) or In-phase and Quadrature phase (IQ).

The tools used for traditional wireless spectrum analysis can be brought down to four main categories: wireless signal detection, signal classification, anomaly detection and signal decoding. For initial signal detection, traditional systems made use of various techniques such as energy detection, matched filter based detection, cyclostationary feature based detection and even other co-operative techniques [19,20]. For signal classification, conventional approaches also relied on expert features like cyclic moments [21]. Spectral correlation functions of various analog and digital modulation schemes covered in [22] and [23] respectively are the popularly used features for classification. On the other hand wireless anomaly detection to some extent has been addressed in wireless sensor networks in the past [24–26]. These techniques again make use of derived expert features from very low rate sensor data such as temperature and pressure instead of high volume radio physical layer data as is our interest. An anomaly detector for Dynamic Spectrum Access (DSA) is presented in [27], where distributed power measurements via cooperative sensing are used for anomaly detection. Similarly [28] makes use of Hidden Markov Models (HMM) on spectral amplitude probabilities that can detect interference on the channel of interest again in the DSA domain. Once the signal parameters are classified, for signal decoding, standard modulation specific demodulators are employed.

All the aforementioned algorithms follow a common design principle: specialized algorithms based on expert features leveraging the knowledge about the signal and noise characteristics. These expert features are derived based on specialized knowledge from a small subset of signals (small data). Many of these expert feature models have their own shortcomings, for example, cyclostationary based signal detection and classification suffers from long classification time and high

computational complexity. Thus, a few sparse spectrum monitoring stations, from various initiatives, relying on traditional statistical feature based techniques proved inadequate in developing a self-sustainable spectrum monitoring solution.

1.3 SDR advances

Innovations in the semiconductor industry, specifically advancement in Radio Frequency Integrated Circuit (RFIC), Field Programmable Gate Array (FPGA) and Electronic Design Automation (EDA) tools, and advancement in the software technologies in the last 30 years, have driven SDR technologies to become a dominant industry standard. Most of consumer radios, such as 4G and LTE handsets moved to SDR architectures, as a result of low power and high performance solutions offered by companies like Ceva, Tensilica and Qualcomm. Wide range of general purpose SDRs[4] in low, medium and high cost and configuration ranges are available nowadays with extensive software support and tutorials. A detailed feature comparison of popular general purpose SDRs are included in Table 1.1. Various government and private organizations publicly supported development of these SDRs, for example, support from DARPA and European Space Agency (ESA) for HackRF and LimeSDR devices respectively.

The availability of high performance SDRs which can scan in the frequency range of 0 to 6 GHz dramatically reduced the efforts involved in data collection and wireless spectrum analysis. Low-cost SDRs when combined with big data solutions has a potential to achieve large scale spectrum monitoring resulting in a wireless spectrum data deluge.

1.4 Data driven models

Low-cost data storage, high speed processors and Graphics Processing Unit (GPU), advanced neural network layers and training algorithms and other high speed data storage and retrieval techniques have made it possible to develop problem solvers from the data itself. Recently, deep learning, making use of these data driven techniques, has been shown to be effective in various tasks such as image classification, machine translation, automatic speech recognition [29] and network optimization [30]. Thanks to multiple hidden layers with non-linear logistic functions which enable learning higher-level information hidden in the data.

[4] https://en.wikipedia.org/wiki/List_of_software-defined_radios

	HackRF One	Ettus B200	Ettus B210	BladeRF x40	RTL-SDR	LimeSDR	LimeSDR Mini
Frequency Range (MHz)	1 - 6000	70 - 6000	70 - 6000	300 - 3800	22 - 2200	0.1 - 3800	10 - 3500
RF Bandwidth (MHz)	20	61.44	61.44	40	3.2	61.44	30.72
Sample Depth (bits)	8	12	12	12	8	12	12
Sample Rate (MSPS)	20	61.44	61.44	40	3.2	61.44	30.72
TX Channels	1	1	2	1	0	2	1
RX Channels	1	1	2	1	1	2	1
Duplex	Half	Full	Full	Full	N/A	Full	Full
Interface	USB 2.0	USB 3.0	USB 3.0	USB 3.0	USB 2.0	USB 3.0	USB 3.0
Programmable Logic Gates	64 macrocell CPLD	75k	100k	40k (115k avail)	NA	40k	16k
Open Source	Full	Schematic, Firmware	Schematic, Firmware	Schematic, Firmware	No	Full	Full
Oscillator Precision	+/-20 ppm	+/-2 ppm	+/- 2 ppm	+/- 1 ppm	-	+/-4 ppm	+/- 4 ppm
Maximum Transmit Power	15 dBm	10 dBm	10 dBm	6 dBm	N/A	10 dBm	10 dBm
Price ($)	299	686	1119	420	10	299	159

Table 1.1: SDR comparison.

The wireless field is also in general moving beyond information theory based communication models to data driven research. Data driven models are getting developed actively for understanding the signals better, for instance very deep models for wireless signal classification [31] and end-to-end communication systems [32]. Apart from man made signals, future wireless signals will be machine made, for achieving higher spectral efficiency, making them much more complex to analyze and interpret.

1.5 Towards fully automated spectrum awareness

All the current and future use cases point to one single direction: *achieving full automated wireless spectrum awareness*. As we clearly identified multiple solid use cases in the previous sections, we can further proceed with discussions on how to obtain spectrum data and what information should be extracted for supporting these identified use cases.

1.5.1 Data collection

Collecting spectrum data over large geographical areas across time and frequency dimensions is a challenging task. First, wireless channel propagation itself is complex and varies across time, frequency and space dimensions. Second, the sensor density requirements varies based on locations: urban or rural and applications such as spectrum occupancy or localization. Third, large scale deployments, sensor control and spectrum data management itself is quite tedious. For a typical sensor, scanning the spectrum range from 20 MHz to 6 GHz, at a low sampling rate of 2.4 MS/s, can easily generate upto 50 Mb/s. This data rate can be further reduced to the order of 50-100 Kb/s by only sending averaged PSD data with proper compression factors. Deploying even these low-end sensors itself will result in a data deluge demanding big data solutions for data management. In addition to these main challenges, there are many engineering difficulties in realizing such a framework. Addressing all these challenges are critical to achieve the large scale spectrum data collection dream.

1.5.2 Data analysis

Once we obtain spectrum data, the next obvious step is analyzing the collected data. Similar to all other fields like speech and image processing, traditional systems in the field of wireless communication, as mentioned in the previous subsections, relied on expert models: models based on human understanding of

the wireless system and data. Unlike other fields like image processing, wireless communication theory has strong roots in information theory [33] and statistics [34]. This helps us to develop tractable mathematical models and formulate optimal bounds for a large variety of problems under linear, time-invariant and Gaussian conditions. However, the practical wireless communication system includes many non-linear hardware components, channel conditions and operations which makes the optimization process difficult. Non-linear machine learning models especially data driven deep-learning models might help to solve these problems more optimistically. In addition, end-to-end systems might further improve the performance by discarding current modular wireless communication structure such as source coding, modulation, channel estimation and equalization.

One of the challenges is addressing the diversity of spectrum knowledge for users and applications. For instance, even for detecting and classifying anomalies in the wireless spectrum different users might have different preferences. Anomalies for a normal user might not be anomalous for a mobile operator, for example new transmitter deployments [35]. So some of the major questions are: Can we use traditional methods to extract all relevant information from the spectrum data? Are there huge benefits in using machine learning models, especially using deep learning models? Specifically, can machine learning, based on low-cost crowdsourced spectrum data that is not labeled, be used for spectrum anomaly detection and modulation classification? Do these deep learning based methods outperform statistical based approaches based on expert knowledge?

Another important challenge related to big data is that most of the best performing models in literature are fully-supervised. By supervision we mean that the exact expected output needed for the data used to train the model is known, for example, in a classification problem all the training data class labels are available. Since data labeling over huge datasets is a nightmare and impossible [36], semi-supervised data driven models should be explored that can work with lower number of labeled samples. This exploration also raises various questions: Are fully unsupervised models good enough for extracting spectrum information? Can semi-supervision and user engagement aid the whole spectrum awareness generation? If yes, how much expert labeling and feedback is needed to achieve meaningful anomaly detection and classification results when compared to fully supervised models?

In the spectrum awareness context, machine learning models can act as analysis tools for explaining the spectrum better. Furthermore, when compared to classic expert feature based models, we believe these models can perform fast analysis, for instance classification of a signal from a few modulated wireless symbols. Energy efficient hardware accelerators for improving machine learning

performance such as GPUs and dedicated neural network accelerators[5] are advancing at a faster rate when compared to the specialized digital signal processing chips. In addition, big data storage solutions have also advanced a lot in the last decade [37], making a strong case for data driven models. After evaluating all these recent advancements, we believe that machine learning has a crucial role in enabling full wireless spectrum awareness, leading to the following thesis hypothesis

Hypothesis of this thesis:
Deep learning based anomaly detection and modulation classification using crowdsourced spectrum data from low-cost sensors can outperform expert based statistical methods even when relying on unsupervised or partially supervised learning methods.

1.6 Scope of the thesis

This doctoral work provides a significant contribution towards the development of a large scale wireless spectrum monitoring and analysis framework leveraging the recent advancements in machine learning, especially some of the deep-learning models. The main contributions of this doctoral work are to design various techniques to analyze the wireless spectrum, in turn enhancing spectrum awareness. We achieve the spectrum monitoring goal by answering the following four major research questions.

- How to monitor the complex and dense electromagnetic space on a large scale in a cost-effective manner?
- How to detect and classify anomalous behaviors in the wireless spectrum efficiently with out supervision?
- How to incorporate user feedback to the entire monitoring framework for improving spectrum awareness?
- How good is SoA research in utilizing the achieved spectrum awareness for improving wireless communication performance?

The proposed techniques take into account various practical design constraints such as computation power limitations of low-cost sensors, data transfer costs and limited sampling rate of the SDRs.

[5] https://software.intel.com/en-us/neural-compute-stick

1.7 Methodology and contributions

Based on the identified use cases and hypothesis, a systematic experimentation flow is followed in this work as shown in Figure 1.1. To substantiate and validate the proposed hypothesis, it is important to have a scalable framework for large scale sensor deployment. The framework is used for collecting large scale spectrum data for further spectrum awareness validation. Further, for supporting multiple use cases and in turn improving spectrum knowledge, detecting anomalous events in the wireless spectrum is identified as an important step. Research in the direction of anomaly detection is advanced both in unsupervised and semi-supervised settings. Once the anomalous events are detected, it is important to understand more about the anomalous signal for taking further steps, leading to the wireless signal classification research. Validation of newly proposed anomaly detection and signal classification modules are completed using synthetically generated theoretical datasets as well as the collected data from the sensor network. In addition, the benefits of spectrum awareness in a spectrum sharing scenario is practically validated in this thesis.

The four research questions mentioned in the previous subsection are answered, by pursuing the research methodology mentioned above, with the following four key contributions of this doctoral work.

Contribution 1: *Framework design for crowdsourced spectrum monitoring and data collection.*
A large scale wireless spectrum monitoring framework has been developed [38] in collaboration with three other research groups IMDEA Networks, Spain, Sero systems, Germany and armasuisse, Switzerland. Worldwide spectrum monitoring is made plausible by combining crowdsourcing paradigm with low-cost sensors. Various sensor, data transfer, data storage and sensor control challenges are tackled systematically resulting in the deployment of a working sensor network which is actively growing.

Contribution 2: *Unsupervised deep learning model for anomaly detection.*
Detecting anomalous behaviors in the wireless spectrum from the sensed wireless spectrum data is challenging due to the sheer complexity of wireless spectrum usage and the characteristics of the electromagnetic space itself. Wireless spectrum anomalies can take a wide range of forms from the presence of an unwanted signal in a licensed band to the absence of an expected signal, which makes manual labeling of anomalies difficult and suboptimal. A data driven unsupervised deep learning model [39] is developed to detect anomalous behaviour from the wireless spectrum data. In addition to detection, the model enhanced anomaly localization in the frequency spectrum and data compression capabilities for deployment.

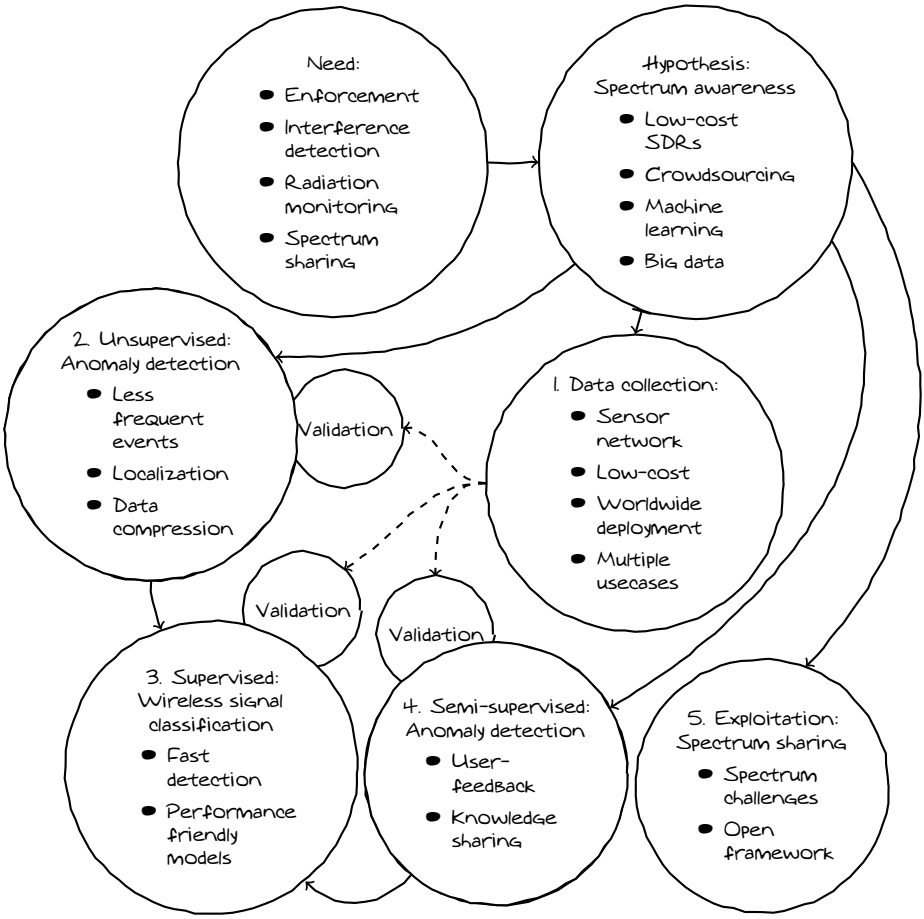


Figure 1.1: Methodology: Based on the hypothesis, new data collection, unsupervised and semi-supervised anomaly detection and signal classification techniques are developed and validated.

Contribution 3: *New models for wireless signal classification.*
A new Long Short Term Memory (LSTM) based data driven wireless signal classification model [40] was developed for detecting the type of modulation used by a particular signal, with detection accuracy higher than SoA models, that only use fewer samples. The model's generalization capabilities along with inference time on various hardware platforms are evaluated and profiled in detail for deployment.

Contribution 4: *Crowdsourced anomaly detection.*
A new anomaly detection framework leveraging user feedback is proposed to improve the detection performance. The model generalizes anomalies across different frequency bands and sensors enabling knowledge sharing across users.

Contribution 5: *Evaluation of the benefits of spectrum awareness for a spectrum sharing scenario.*
Two wireless spectrum challenges [41,42] were organized in co-operation with Institute of Electrical and Electronics Engineers (IEEE) during the DySPAN conference to understand the capabilities of SoA radios in achieving and exploiting situational awareness. The algorithms are also evaluated in terms of performance improvements of network throughput while there is perfect situational awareness for spectrum sharing. Along with these detailed analysis the complete challenge framework and the solutions are made public to the research community for future benchmarks.

1.8 Outline of the thesis

The rest of this thesis is organized into six additional chapters mirroring the contributions mentioned in the previous subsection as shown in Figure 1.2. All chapters contents are adapted from IEEE published journals or journals under review which are indicated at the beginning of respective chapters.

1.8.1 Chapter 2: Spectrum sensing network

Chapter 2 explains how the large scale spectrum monitoring problem is solved leveraging crowdsourcing. Electrosense, the spectrum monitoring framework is presented in this chapter along with relevant applications. A wide variety of problems ranging from spectrum sensing hardware to user privacy issues are systematically analyzed and solved in this chapter. Electrosense deployment showed that volunteer based large scale spectrum monitoring is feasible and can scale easily.

1.8.2 Chapter 3: Unsupervised anomaly detection

Detailed literature and challenges in wireless spectrum anomaly detection is presented in Chapter 4. Spectrum Anomaly Detector with Interpretable FEatures (SAIFE), an unsupervised anomaly detection algorithm with interpretable features, that can detect and localize anomalies, is proposed

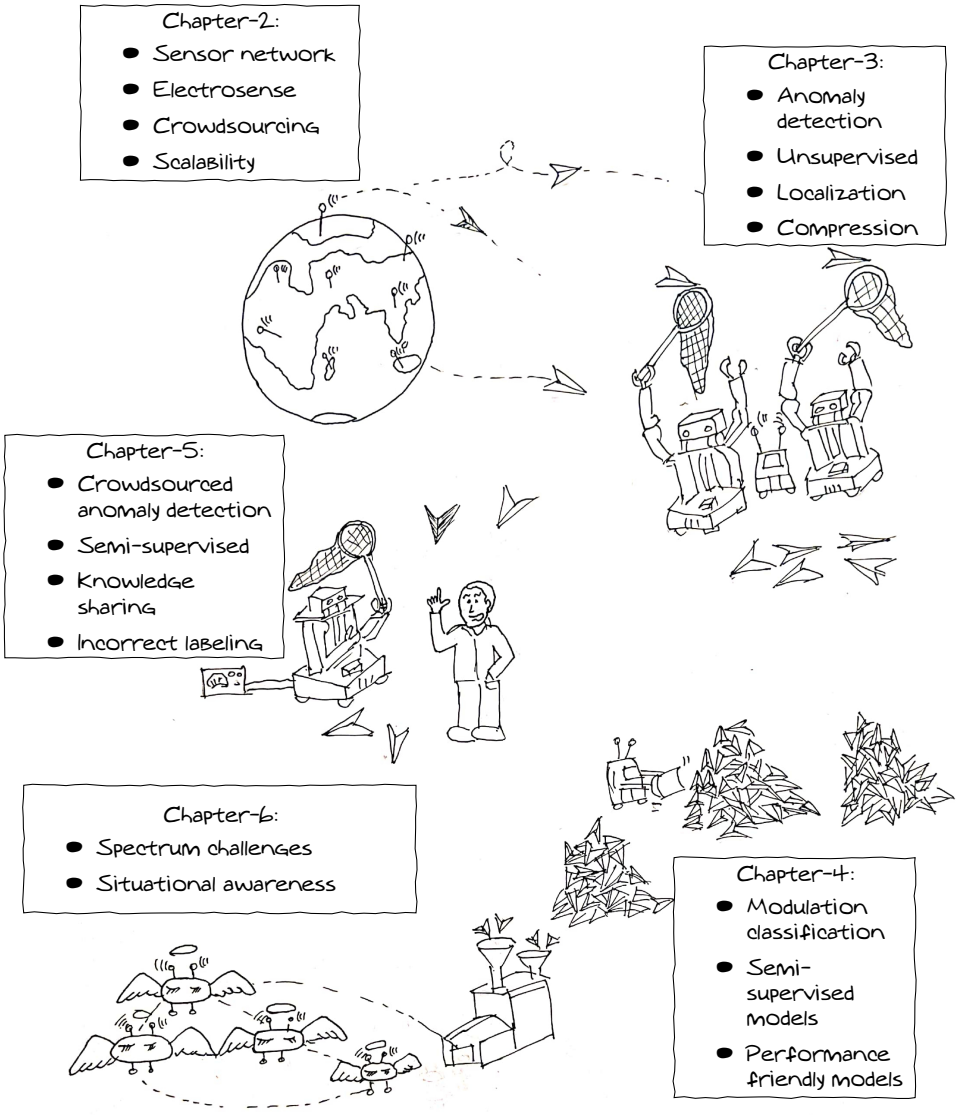


Figure 1.2: Thesis outline

in this chapter. Anomaly detection is done on PSD data enabling detection even with low-cost spectrum sensors at lower data transfer costs. The main problems with unsupervised anomaly detection are discussed which are further solved in Chapter 5.

1.8.3 Chapter 4: Signal classification

Once anomalies are detected, it is quite important to understand more about the anomalous signals. To contribute in this research direction, new models for wireless modulation classification are proposed in Chapter 4. Signal classification on both PSD and full IQ data are presented. Chapter 4 also discusses detailed implementation issues and how the models can be fine tuned for low-cost sensor deployments.

1.8.4 Chapter 5: Crowdsourced anomaly detection

Various challenges associated with unsupervised anomaly detection are addressed in Chapter 5 by formulating the problem as a semi-supervised detection problem. A new anomaly detection framework for wireless spectrum data is proposed which can make use of expert user feedback across frequency bands and sensors. This chapter also discusses how to generalize expert anomaly knowledge across sensors from different locations. Incorrect labeling, which is one of the issues of crowdsourced platforms is also addressed in detail in this chapter.

1.8.5 Chapter 6: Situational awareness for spectrum sharing

This chapter details the design of two wireless spectrum challenges organized to study and quantify the benefits of spectrum awareness for a spectrum sharing scenario. The spectrum challenge setup, the problem statement and the algorithm performance are discussed in detail in this chapter. The main challenges in realizing wireless spectrum awareness in general wireless radios are further analyzed with practical experiments. In addition, the capabilities of SoA algorithms in leveraging recent advancements in machine learning and spectrally efficient waveforms are evaluated.

1.8.6 Chapter 7: Conclusion and future work

The thesis is concluded in this chapter with a final review of the research questions and contributions. A few future research directions are also proposed for improving the current state of spectrum monitoring.

Chapter 2

Spectrum sensing network

While the radio spectrum allocation is well regulated, there is little knowledge about its actual utilization over time and space. This limitation hinders taking

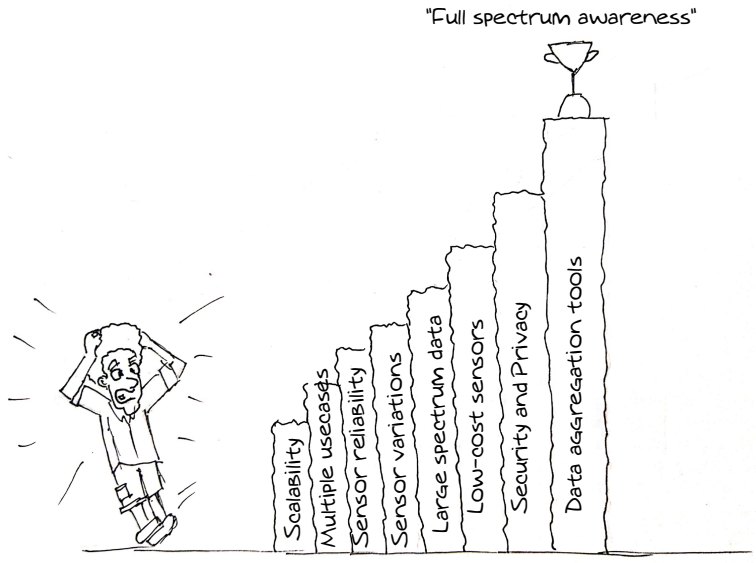


Figure 2.1: Spectrum sensing challenges to achieve wireless spectrum awareness at a large scale. Scalability and sensor challenges are dealt jointly by the Electrosense team.

effective actions in various applications including cognitive radios, electrosmog monitoring, and law enforcement. This chapter introduces Electrosense, an initiative that seeks a more efficient, safe and reliable monitoring of the electromagnetic space by improving the accessibility of spectrum data for the general public. A collaborative spectrum monitoring network is designed that monitors the spectrum at large scale with low-cost spectrum sensing nodes. The large set of data is stored and processed in a big data architecture and provided back to the community with an open spectrum data as a service *model, that allows users to build diverse and novel applications with different requirements. Different useful usage scenarios of the Electrosense data are also illustrated in this chapter.*

This chapter is adapted from: Rajendran, Sreeraj; Calvo-Palomino, Roberto; Fuchs, Markus; Van den Bergh, Bertold; Cordobes, Hector; Giustiniano, Domenico; Pollin, Sofie; Lenders, Vincent; *Electrosense: Open and Big Spectrum Data.* IEEE Communications Magazine; 2018; Vol. 56; iss. 1; pp. 210 - 217.

The sensor scalability, sensor control, large spectrum data and security issues solved in this chapter is the result of a collaborative effort from the entire Electrosense team. Unsupervised and semi-supervised data analysis/aggregation tools presented in the subsequent chapters are personal contributions of the author.

2.1 Introduction

Over the past years, we have seen a tremendous increase in mobile data usage. To meet the data demands, more radio frequency (RF) bands are being used, cells become smaller, data rates increase and more users accommodated. In addition, novel technologies are being proposed that can coexist with legacy technology and modulation types, and short and long range Internet-of-Things (IoT) communications increase the variety of physical and medium access protocols connected to the Internet. As a result, RF spectrum use is fragmented, bursty and very diverse. The monitoring of this spectrum is a complex task, as it requires a very dense sampling in time, frequency and space, resulting in a radio spectrum data deluge. Nevertheless, with the increasing spectrum usage complexity, knowledge and understanding of the spectrum usage patterns is becoming more and more critical to ensure continued effective use of this scarce resource.

Spectrum resource monitoring is important for end users, operators, spectrum regulatory bodies and also military applications. Each use case has its own specific needs and challenges. The grand challenge is how to design a cost-

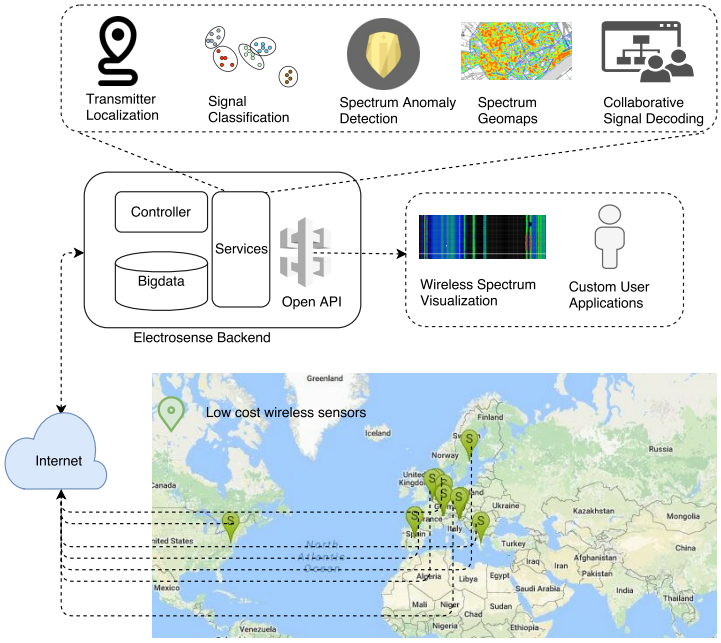


Figure 2.2: High-level overview of the Electrosense network: Low-cost sensors collect spectrum information which are sent to the Electrosense backend. Different algorithms are run on the collected information in the backend and the results of these algorithms are provided to the users as a service through an open API. Users can develop their own applications from the spectrum information retrieved using the API.

effective solution that meets the requirements of all potential end users. Users might be interested in electrosmog or optimization of their indoor WiFi network. Regulatory bodies might be keen on enforcing spectrum regulation. Operators might be concerned about coverage maps over time for optimizing their cell networks or refarming of their frequency bands. Military applications might be the most challenging, requiring the detection and positioning of any signal hidden on purpose. Novel operators might be interested in IoT cases, such as cooperative detection of signals transmitted by low-cost low-power transceivers. This chapter looks at one key aspect: is it possible to design a spectrum sensing network that meets all the aforementioned requirements and can be deployed quickly by synergistic cooperation between all stakeholders?

Even though a majority of wireless researchers, industries and spectral regulators are keen to develop a worldwide spectrum monitoring infrastructure, albeit several attempts, the research community has not succeeded in deploying one. The multidisciplinary nature of the spectrum monitoring solution is one of the

main challenges that prevents the realization of such a system, which in turn requires proper integration of new disruptive technologies. The infrastructure should flexibly address the variability and cost of the used sensors, large spectrum data management, sensor reliability, security and privacy concerns, which can also target a wide variety of the use cases as mentioned before.

Wireless spectrum monitoring is an interesting field of research which has attracted both academia and industries and there had been many initiatives in the past. A few initiatives are funded by the NSF projects in the USA [12–14]. A metro scale spectrum observatory [15], an initiative from University of Washington is also sponsored by NSF. Radiohound [16] is another spectrum monitoring initiative concentrating on mobile environment monitoring using low-cost SDR. Hawkeye360[1] is a recent organization planning to build low earth orbit satellites to monitor global radio frequency activity across air, land and sea and assist with emergencies.

A few spectrum monitoring solutions are proposed in the literature. Some examples include, Microsoft Spectrum Observatory[2] that allows to sense the spectrum using expensive sensors, Google spectrum[3] for measurements on TV white-spaces and the IBM Horizon[4] project that proposed a generic decentralized architecture to share *IoT* data. While Google Spectrum and Blue Horizon fail to cover a large part of the spectrum as they are application specific deployments, Microsoft observatory on the other hand fails to enable a large scale sensor deployment mainly due to the cost of the sensing stations.

We introduce and design Electrosense[5]. Electrosense follows the crowdsourcing paradigm to collect and analyze spectrum data using low-cost sensors as main device for sensing the spectrum. The main goal of this initiative, as shown in Figure 2.2, is to sense the whole spectrum in different regions of the world and provide the processed spectrum data to any user interested to acquire a deeper knowledge of the spectrum usage, enabling applications as the ones mentioned above [43,44]. Worldwide deployments are plausible when the crowdsourcing paradigm is combined with low cost sensors. Electrosense sensors are designed using inexpensive and easily accessible software-defined radio (SDR) front-ends and embedded platforms like Raspberry Pi, to reduce the cost that spectrum data contributors have to bear. A low-cost down-converter design[6] is also provided to extend the range of the low-end SDR platforms from 1.7 GHz to 6 GHz. Even though the aim of the Electrosense project is to use low-cost

[1] http://www.he360.com
[2] http://observatory.microsoftspectrum.com/
[3] https://www.google.com/get/spectrumdatabase/
[4] https://bluehorizon.network/documentation/sdr-radio-spectrum-analysis
[5] https://electrosense.org/
[6] https://github.com/electrosense/hardware

hardware, high-end SDR devices can also be part of the network. Open source software modules for low- and high-end sensors are provided by Electrosense for easy setup[7].

Crowdsourcing initiatives have been successfully applied in other sensing contexts, for example, distributed sensor networks collecting information about temperature, air quality, pollution or air traffic are now in widespread use today. Yet, the radio spectrum data collection poses novel challenges because of the sheer volume of information in the spectrum, which is several orders of magnitude higher than the data collected in typical sensing contexts as mentioned above. The amount of data produced by a single sensor typically varies between 50 Kb/s and 50 Mb/s depending on the sensor platform and configuration. Storing spectrum data received from each sensor can swiftly get any server machine out of storage space, for instance data from 60 sensors for a single month needs a storage space of 1 terabyte in 50 Kb/s mode. In order to handle this large volume of data in the backend and extract meaningful information over the entire spectrum, a flexible big data architecture is designed. This spectrum management architecture is responsible for the sensors' control, data storage, and algorithm deployment in the back-end for further processing.

Electrosense also includes a complete framework for identifying, locating and deploying sensors securely through a consistent registration process and remote control framework of the sensors. The sensed spectrum data from different sensors can be retrieved from the electrosense backend using an open API[8] over the Internet. Data privacy concerns are addressed by enabling public, private and restricted data access permissions which restrict user data access with suitable time resolutions. Readily available spectrum aggregation tools in the back-end help the network users, to do easy spectrum data analysis as an additional incentive. The data access permissions and other applications are detailed in Section 2.2 and Section 2.4 of this chapter.

2.2 Spectrum Data as a Service

2.2.1 Service Model

In order to make the most out of the data, we introduce an Open Spectrum Data as a Service (OSDaaS) model in which the spectrum data can be used by several applications, each of them with unique requirements. This approach differs from a classical "Infrastructure as a Service" model used in other contexts where

[7] https://github.com/electrosense
[8] https://electrosense.org/open-api-spec.html

different applications cannot run at the same time due to conflicting requirements. The spectrum data inherently changes over time and thus allocating the node to a specific application would reduce the information and the interest of several users interested in the same spectrum data. Therefore, the current framework prevents the users from launching their own measurement campaigns as they only need to access gathered data from one of the two pipelines of the sensor: PSD (Power Spectral Density) and IQ (in-phase and quadrature components of raw signals). In contrast, the flexibility of the OSDaaS model is an incentive to join and use the network and an opportunity for a wide engagement and participation of citizens. In addition, the spectrum data gathered in the past can be re-processed with a new application in mind or new spectrum data can be compared with older spectrum data. Users can join the Electrosense network easily by adding their own sensor through the web interface[9]. Specifics about the sensor such as antenna details and sensor location can be specified during the registration process. The sensor status and location are readily visible in a global sensor map once it is registered.

2.2.2 Application Programming Interface (API)

Alongside with data processing, it is important to provide the community with easy access to the data. An open API serves this purpose for bulk or streaming data retrieval. The API also allows access to the algorithms' output running in the backend, for example results of the modulation classification algorithms or the anomaly detector. Electrosense users can retrieve magnitude data along with the sensor details using the API. The API allows two data retrieval modes: aggregated and raw Fast Fourier Transform (FFT) data. Aggregated query type allows the user to request for a bulk of data with specified frequency and time resolutions, after applying a predefined aggregation function such as averaging or max-value. Raw requests permit to retrieve magnitude FFT data acquired by a sensor as such without any modifications. Raw FFT data from a sensor can be accessed only by its owner. In addition, as shown in Figure 2.4, there is an IQ data pipeline which is enabled on demand for data testing or for algorithms that work on IQ data in the backend. The IQ data from a sensor is only directly available to its owner through the API, but conclusions from algorithms related to IQ data will be made available to all users. Furthermore, real-time spectrum monitoring is possible through a streaming API in the backend. A few examples for data retrieval can be found in the repository[10].

[9] https://electrosense.org/join.html
[10] https://github.com/electrosense/api-examples

2.2.3 Security and Privacy

Wideband spectrum monitoring on a large scale typically raises critical security and privacy concerns. Security concerns include secure data transmissions from the sensor and proper sensor identification. Data privacy concerns in terms of data access restrictions should also be addressed. In addition, capture and storage of IQ data, which can be decoded to reveal content, is not advisable especially in the military bands. Electrosense's design addresses sensor identification via a proper registration process. The data from Electrosense sensors are sent over secure TLS channels which guarantees data privacy and integrity and prevents the data getting modified by an attacker. The IQ pipeline is only accessible by the Electrosense backend and the data is deleted once the backend algorithms are done processing it. In addition, IQ data is only made available to the users for their own sensors through the API. Furthermore, the highest time and frequency resolution that users can get from other sensors over the API are limited to 60 seconds and 100 kHz respectively. Electrosense also allows to obfuscate sensor's publicly displayed location within a range of a few kilometers in the sensor map, thus protecting users' locational privacy.

2.3 System Architecture

Electrosense was designed with a clear vision in mind that translates into a list of design goals as described below. We present the actual architecture chosen to meet these requirements and give a more detailed explanation of its components.

2.3.1 Design Goals and Electrosense Vision

Electrosense is a crowdsourcing approach towards spectrum monitoring where volunteers play a central role. To keep the entry barrier as low as possible, large-scale deployment can only be achieved with *low-cost sensors*. The backend should be flexible, to enable *wide-band monitoring* from sweeping low-end sensors, to *real-time scanning* with more capable SDR platforms or spectrum analyzers. In addition, the backend should be horizontally *scalable*, enabling to grow the network continuously. Scalability introduces even more complexity into the system, which makes it prone to failures. As such, the architecture must also be *fault-tolerant*. Finally, the backend should be able to process a large amount of data from any sensor with *low latency*.

With its main goal to serve researchers as a platform for spectrum analysis, the system should have a *central component to control* the sensors for specific

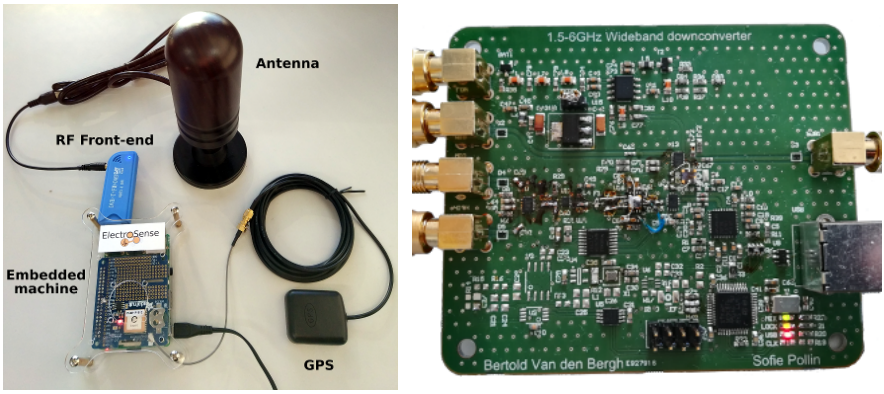


Figure 2.3: [Left] Sensor based on a single-board computer (RPi-2) and RF front-end (RTL-SDR) device responsible for signal acquisition. [Right] First prototype of custom RF converter to enable spectrum scanning in 0-6 GHz range.

measurement campaigns. Some applications might need to give immediate response while others generate insights by inspecting the data set as a whole. Following these requirements, the system should support *low-latency stream processing* and *large-scale batch analyses* at the same time.

In order to address these goals and requirements, the Electrosense architecture consists of three main components: (i) the sensors deployed by users collecting spectrum measurements, (ii) a centralized controller infrastructure to interact with the sensors and to administrate measurement campaigns and (iii) the backend which is responsible for collecting data from all the sensors and provide insights by applying algorithms. The following subsections describe these components in detail.

2.3.2 Sensor

The sensing nodes used in the Electrosense network consist of small-sized, low-cost, software-defined embedded computing devices connected to a simplistic RF front-end and a general purpose antenna [45] as shown in Figure 2.3 (left). The sensors can measure the spectrum ranging from 20 MHz up to 6 GHz using an optional down-converter (Figure 2.3 right). The sensors include an optional low-cost GPS device to synchronize the time among them which in turn helps in enabling collaborative spectrum scanning and detection algorithms. For instance, in [43] the feasibility to recombine signals collaboratively from different sensors, using GPS as a reference clock, is analyzed in detail.

Two signal pre-processing pipelines are enabled on the sensor as illustrated in Figure 2.4. Each sensor can be configured in order to work with both PSD or IQ pipeline. When the sensor is configured in the PSD mode, the spectrum sensed by the RF front-end is converted to the frequency domain using a Fast Fourier Transform (FFT). In this mode only the averaged squared magnitude FFTs are sent to the backend. Depending on the averaging and compression factors, the produced data is in the order of 50-100 Kb/s. This pipeline saves bandwidth and storage over the IQ pipeline and thus contributes to the cost-efficiency of the whole network. In addition, the on-board GPU on the Raspberry Pi is used for computing FFTs, reducing the processing time. The PSD mode aids spectrum data retrieval using a restricted network bandwidth which enables applications where phase information is not crucial to analyze data in terms of signal power (Section 2.4.2). In addition, wide-band scanning techniques are applied that prioritize frequency bands of bursty activities over bands which are stationary [45] to overcome the hardware-limitations of low-cost radios, for instance the limited sampling bandwidth of the low-end ADCs (e.g 2.4 MHz for RTL-SDR).

For applications which require phase and non-averaged information, the sensor implements an IQ pipeline. The IQ mode retrieves raw measurements (IQ samples) from the RF front-end that are then compressed and sent to the backend. This pipeline easily produces up to 50 Mb/s and hence the data is only stored temporarily in the backend. Besides spectrum measurements in both pipelines, the sensor also provides timing information together with RF-specific values such as antenna gain. The data structure used to transport results over the Internet is flexible and can be extended in the future.

The software that runs in the electrosense nodes is released as open source[11]. In addition to this, a GNU Radio based software module, gr-electrosense[12], is provided to allow high-end SDR users to be a part of the Electrosense network.

2.3.3 Controller

Apart from continuous wide-band monitoring, it is interesting to have a more detailed look at particular parts of the frequency spectrum. This is accomplished with the help of a command-and-control layer as depicted in Figure 2.5. In order to start and control such in-depth measurements, the controller communicates with sensors directly and influences their scanning strategy, some examples include, the frequency range the sensor should scan, the sensor frequency hopping strategy or the sensor sampling rate.

[11] https://github.com/electrosense/es-sensor
[12] https://github.com/electrosense/gr-electrosensexamplese

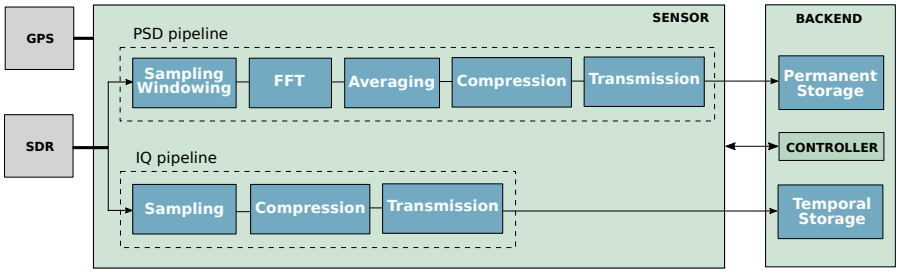


Figure 2.4: Sensor architecture: Sensor software contains two different pipelines for retrieving spectrum information.

The core of the controller infrastructure is a Message Queue Telemetry Transport (MQTT) broker system which consists of a set of several independent machines forming a cluster. This ensures scalability and fault-tolerance in case of a hardware failure. MQTT is a publish-subscribe-based messaging protocol and has a wide adoption in IoT applications [46]. The brokers have a permanent connection to the sensors over secure TLS channels as well as to a machine called *master controller*. The master offers an interface where administrators can start and stop *measurement campaigns*. It is responsible for executing these commands on different sensors using the MQTT connection. Spectrum measurements collected by the sensors are passed to the backend over secure TLS channels as shown in Figure 2.5. The control layer is a part of the Electrosense backend and works in a closed loop with the data processing path for easy response validation.

2.3.4 Backend

All data collected by the sensors is sent to the *backend*. In order to fulfill the design goals, in particular scalability, the system is built upon a Lambda architecture [47] with exclusively horizontally scalable system components. The Lambda architecture is a three-layered architecture consisting of *batch*, *speed* and *serving* layer. An overview of the whole backend architecture can be found in Figure 2.5. The batch layer is responsible for executing bulk analysis tasks on large data sets. These tasks, mostly involve complex long-running algorithms, whose results might take several minutes to hours to complete. To overcome this gap and, furthermore, provide the capability for real-time monitoring with sub-second delay, there is the need for a speed layer. It is possible to run different or identical applications on each of these layers simultaneously.

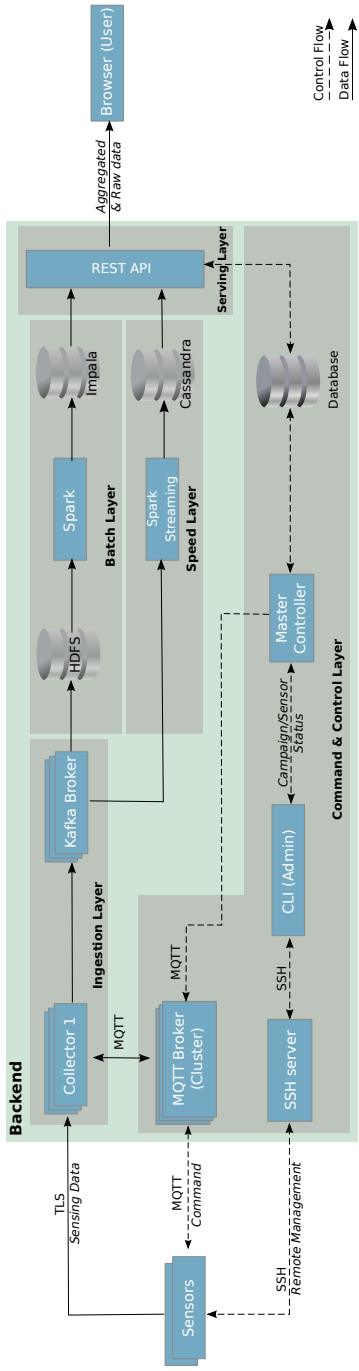


Figure 2.5: Sensors are managed by the control layer which sets the scanning parameters for a specific campaign. The configured sensors send the spectrum information to the backend, where the data is processed and distributed over the API to the users.

In addition, the Electrosense backend has an *ingestion* layer to support high availability. It consists of a distributed message queue that stores multiple replicas of each incoming measurement. Implementation and functionality of each layer is detailed in the subsequent sections.

Ingestion Layer

Sensor measurements eventually reach the *collector* as shown in Figure 2.5. The collector inserts received data as messages into a distributed queuing system. Using a distributed queuing system, we decouple data ingestion from processing in the batch and speed layer, allowing an asynchronous mode of operation. Furthermore, the ingestion layer serves as a intermediate buffer for incoming data, allowing maintenance of batch and speed layers without data loss. Apache Kafka [48], a distributed messaging system which features replication, partitioning and data retention is used as the queuing system. Following the ingestion layer, spectrum data is processed on two different paths: in batch and speed layer simultaneously.

Batch Layer

The batch layer is responsible for storing all incoming raw data as an immutable master dataset and executing long-running workloads. The batch layer facilitates processing of huge amounts of historical data, ensuring no loss of information in case of algorithmic errors. The master dataset is stored in a binary format on a distributed file system. The Hadoop Distributed File System (HDFS) [49] is employed allowing data to be stored in files which are distributed and replicated over a cluster of servers. Apache Spark [50], which allows parallel computation by evenly distributing workload over several computing nodes, is employed for data processing. For accessing data stored on HDFS, a fast and scalable query engine Cloudera Impala [51] which understands a subset of the SQL language is deployed. The latency of the batch layer as a whole is in the order of minutes or even hours.

Speed Layer

The Electrosense speed layer is based on Spark Streaming. It is an extension of Spark which computes its' results continuously on a parallel cluster over a small window of recent data, with the current default window length set to 5 seconds. Spark Streaming meets the design requirements as the spectrum data is high in volume and algorithms do not demand sub-second latency. As Impala is not

suitable for near real-time storage requirements, the speed layer persists its results in another database. We use Apache Cassandra [52], a hybrid between a key-value and column-oriented distributed storage system for this real-time database as it represents well the data structure of spectrum data.

Serving Layer

As seen before, the batch and speed layers store their results differently. The serving layer takes care of data fusion to provide query results to the user while hiding the complexity of the multi-layer architecture. It offers the open API which serves as the endpoint for any user or application to retrieve data. The serving layer handles queries on different views and combines results from batch and speed layer if necessary. In case results are available in both layers, batch layer results are preferred as they provide more accurate data. In Electrosense, the serving layer is a custom component which offers a RESTful web service over HTTP.

2.4 Spectrum Data Processing and Analysis

To substantiate the value of the data generated by the Electrosense framework, a few applications are presented in this section. We envision some of these applications to be integrated in Electrosense's batch and speed layer in the back-end, but the goal of Electrosense is that users can start implementing their own applications by using the open API.

2.4.1 Live and Historical Spectrum Visualization

We have developed a web front-end to enable easy visualization of the spectrum data[13]. The spectrum visualizer allows interactive sensor selection and varying frequency resolution for detailed spectrum viewing. The visualization tool allows to look at historical spectrum data from the sensors, since the sensor registration. For this purpose, the batch layer pre-computes spectrum data at various temporal and frequency aggregation levels and stores this information in query-able tables for low-latency access over the web. For live spectrum updates with 5 second delays, a streaming display is incorporated to the web interface directly from the speed layer.

[13] https://electrosense.org/app.html

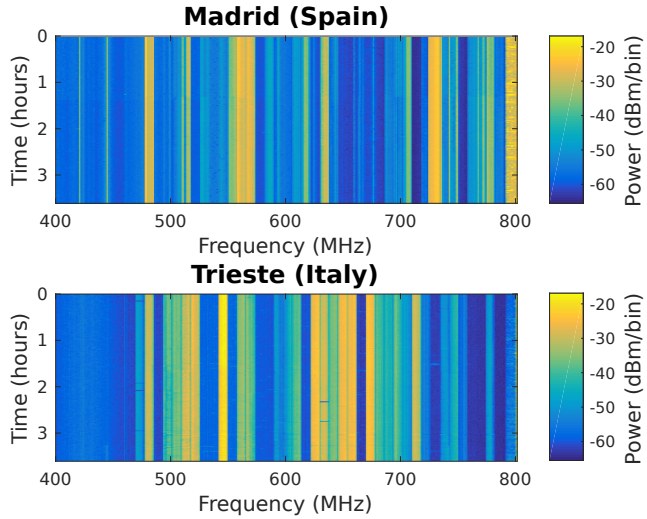


Figure 2.6: TV band occupancy in Madrid and Trieste.

2.4.2 Dynamic Spectrum Access

Dynamic Spectrum Access (DSA) enables a technology model to manage spectrum availability and needs in terms of frequency, time, geography, quality and cost of the service. Cooperative and distributed spectrum sensing approaches for cognitive radios can solve hidden primary user problems and reduce the non-detection probability and false alarms remarkably [53]. Advanced collaborative sensing strategies along with detailed sensor density analysis is required to address the complete cognitive cycle. A white space detection study is done to validate the feasibility of the framework in this direction.

The term *white spaces* usually refers to the unused broadcasting frequencies in the TV band (400-800 MHz). The availability of white spaces is highly region dependent, for instance the white spaces in Europe in the 470-790 MHz band is found to be less than in the USA [54]. Using the Electrosense infrastructure and the sensors configured in the PSD pipeline, it is feasible to detect the white spaces in a certain region. Figure 2.6 shows the spectrum occupancy in the broadcasting TV band (400-800 MHz) of two different cities in Europe with a frequency resolution of 1 MHz and 60 seconds time resolution. Similar analysis can be easily performed by retrieving the aggregated data from the Electrosense backend through the API.

2.4.3 Spectrum Cop

Spectrum enforcement is challenging mainly due to the effort involved in analyzing, detecting and locating anomalies occurring in the spectrum. Even with a large scale deployment of spectrum scanners, the anomaly detection process is demanding. Automated systems to detect pirate FM stations, fake GSM towers, unauthorized transmissions hindering normal functioning of meteorological radars and even transmissions at higher power levels than the permitted levels of operation is fundamental for effective enforcement. A good anomaly detector should first detect whether a particular transmission type is allowed in the frequency of interest. Moreover, it should fingerprint normal spectrum transmissions by analyzing the temporal and spatial features of the transmitters in terms of power levels and spectral occupancy. The former can be achieved to some extent using a modulation classifier and the later by learning spectral occupancy distributions over time. A drastic variation of any of the aforementioned features can be classified as an anomaly.

To validate modulation classification using Electrosense, a time domain deep learning modulation classifier is used in the backend which works effectively with gr-electrosense using the IQ pipeline [40,55]. The deep learning model takes IQ samples as input giving out the probability of the data belonging to a particular modulation class. The model is trained to learn from the modulation schemes' time domain amplitude and phase information, without requiring expert features, such as higher order cyclic moments. Analyses show that the proposed model yields an average classification accuracy of over 90% at varying SNR conditions ranging from 0 dB to 20 dB, independent of channel characteristics. In future a computationally efficient binarized version of the model will be deployed on low-cost sensors reducing the data overhead of sending IQ samples to the backend [40].

2.4.4 Localization

Transmitter localization is a promising application used for generating both automated transmitter maps and transmitter fingerprinting. Research on received signal strength indicator (RSSI)-based indoor and outdoor localization has gained interest in recent years [56]. However, the accuracy of the RSSI information drawn from the Electrosense sensors must be verified to validate localization. Such a sensor is calibrated and then the system gain, consisting of the antenna gain, cable losses and front-end gain, is measured for absolute power measurements. RSSI measurements are made in the 435 MHz ISM band using standard LoRa sensors. A splitter is employed to ensure an equal antenna signal reaches both a LoRa receiver and a low-cost Electrosense sensor. A

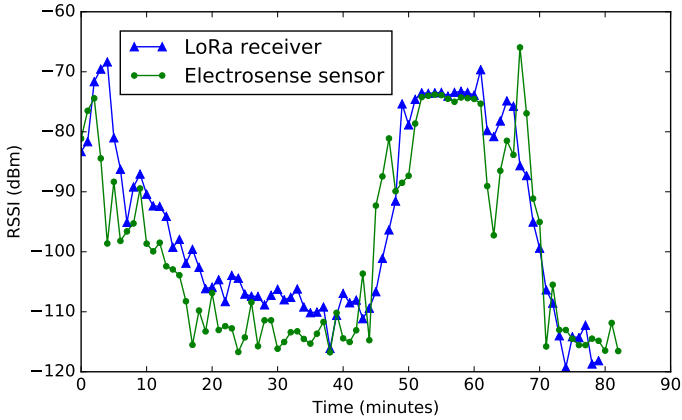


Figure 2.7: RSSI measured using a LoRa and Electrosense sensor.

mobile LoRa transmitter is configured to send packets every three second over a period of 80 minutes. Their GPS logged and time-stamped locations are sent to the receiver. The collected RSS measurements from both the LoRa and Electrosense sensors are depicted in Figure 2.7. This plot shows that accurate RSSI information can be obtained from properly calibrated Electrosense sensors. The RSSI dips in the sensor data are attributed to the lengthy frequency scan of the sensor, which is over 40 seconds, causing it to miss some packets. Existing algorithms presented in the literature [56] can be deployed using the sensed spectrum data retrieved through the open API or implemented directly in the backend.

2.4.5 Spectrum decoding

A new update to Electrosense [57], allow users to remotely decode specific parts of the radio spectrum, extending the centralized architecture of Electrosense for controlling and monitoring the spectrum IoT sensors. It implements a real-time and peer-to-peer communication system for scalable spectrum data decoding. The system architecture allows to decode any public decodable wireless technologies that is within the reception range of the deployed sensors. As new technologies are emerging constantly, active support is enabled for the integration of open-source spectrum decoders developed by the community. The system architecture thus defines open interfaces and APIs to allow easy integration of various types of decoders.

2.5 Conclusion and Future work

We have introduced Electrosense, a global initiative that seeks a more efficient, safe and reliable use of the electromagnetic space by improving the accessibility of spectrum data for the general public. Electrosense as a crowdsourced network provides solutions to the problems associated with large scale spectrum monitoring and the resulting data deluge. The network users receive incentives in terms of free data storage, readily available applications and an open API.

As the first version of the Electrosense network is fully functional, the next step will be in the direction of enabling usable applications in the backend. The feasibility of various applications were already presented in the section 2.4 of this chapter. Research has been continued on the application space, to develop new algorithms for anomaly detection and signal classification which are presented in the subsequent chapters. Electrosense can be the first step to democratize the access to the spectrum data to everyone. The age of spectrum data democratization has arrived and it could help to increase the transparency and the knowledge about spectrum usage.

Chapter 3

Unsupervised anomaly detection

Detecting anomalous behavior in wireless spectrum is a demanding task due to the sheer complexity of the electromagnetic spectrum use. Wireless spectrum anomalies can take a wide range of forms from the presence of an unwanted signal in a licensed band to the absence of an expected signal, which makes manual labeling of anomalies difficult and sub-optimal. In this chapter we present, SAIFE, an Adversarial Autoencoder (AAE) based anomaly detector for wireless spectrum anomaly detection using PSD data which achieves good anomaly detection and localization in an unsupervised setting. In addition, we investigate the model's capabilities to learn interpretable features such as signal bandwidth, class and center frequency in a semi-supervised fashion. Along with anomaly detection the model exhibits promising results for lossy PSD data compression up to 120X and semi-supervised signal classification accuracy close to 100% on three datasets

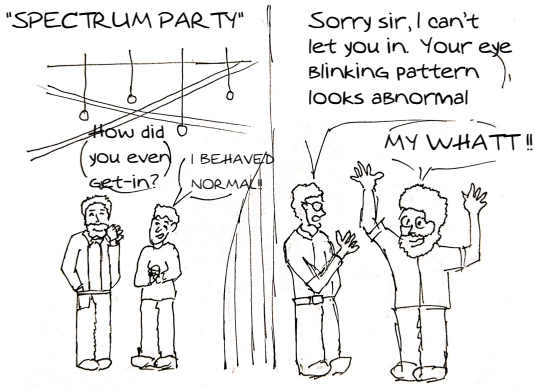


Figure 3.1: Unsupervised anomaly detection is challenging as normal frequent behaviour in one scenario might be anomalous in another.

just using 20% labeled samples. The model proposed in this chapter makes a basic assumption that non-frequent behaviour is anomalous. This assumption is further refined in Chapter 5 with the help of semi-supervised models and user interaction.

This chapter is adapted from:

Rajendran, Sreeraj; Meert, Wannes; Lenders, Vincent; Pollin, Sofie; 2018. SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features. IEEE International Symposium on Dynamic Spectrum Access Networks; 2018 Publisher: IEEE.

Rajendran, Sreeraj; Meert, Wannes; Lenders, Vincent; Pollin, Sofie; 2019. Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features. IEEE Transactions on Cognitive Communications and Networking; *to appear*, 2019.

3.1 Introduction

The new generation of wireless technologies is promising improved throughput, latency and reliability enabling the creation of novel applications. The fifth generation wireless deployments will be very heterogeneous ranging from millimeter wave communications to massive MIMO and LoRa/Sigfox deployments for line of sight (LOS), medium and long range communication systems respectively. Such dense and heterogeneous deployment makes the enforcement and management of the wireless spectrum usage difficult. In addition, manual spectrum management is inefficient and can only deal with a limited number of anomalies and measurement locations. Complex spectrum regulations across frequency bands in various countries along with illegal interference worsen the problem. Automated spectrum monitoring solutions covering frequency, time and space dimensions are becoming more crucial than ever before.

Unlike other sensing contexts such as air quality, temperature or city traffic monitoring, wireless spectrum monitoring on a large scale raises many unique problems ranging from the data costs associated with the sheer volume of sensed spectrum information to sensor quality and data privacy issues. These wide ranging infrastructure problems were systematically analyzed and partially solved by the Electrosense[38] platform. Electrosense is interdisciplinary and combines the power of crowdsourcing with Big data to solve the wireless spectrum monitoring problem. The sensing devices are low cost SDR dongles connected to embedded devices like a Raspberry Pi or high end SDR devices connected

through a personal computer. Through Electrosense, an Open Spectrum Data as a Service (OSDaaS) model was introduced to address the usability of the spectrum data for a wide range of stakeholders including wireless operators, spectrum enforcement agencies, military and generic users.

In addition to the sensor infrastructure problems that were tackled in Electrosense, various algorithmic challenges still need to be addressed to provide advanced spectrum utilization awareness. The central coup to achieve this vision is a wireless spectrum anomaly detector which can continuously monitor the spectrum and detect unexpected behavior. Furthermore, in addition to the detection of anomalies, it is important to understand the cause of an anomaly. This ranges from an unexpected transmission in the analyzed band that can be classified [40], to absence of an expected signal. Wireless anomaly detection to some extent has been addressed in wireless sensor networks in the past [24–26]. These techniques make use of derived expert features from very low rate sensor data such as temperature and pressure instead of high volume radio physical layer data as is our interest. An anomaly detector for DSA is presented in [27], where distributed power measurements via cooperative sensing are used for anomaly detection. The proposed detector is limited to authorized user anomaly detection only, for the specific case of DSA. Similarly [28] makes use of HMM on spectral amplitude probabilities that can detect interference on the channel of interest again in the DSA domain.

Recently in [58], the authors presented a recurrent anomaly detector based on predictive modeling of raw IQ data. The authors used a LSTM model for predicting the next 4 IQ samples from the past 32 samples and an anomaly is detected based on the prediction error. Even though this model works on raw physical layer data which requires no expert feature extraction, it is still not sufficiently automated and generic for practical anomaly detection. First, different copies of the same model need to be trained for different wireless bands such that the model is able to predict anomalies specific to the band of interest. For instance, an LTE signal in the FM broadcast band is definitely an anomaly thus preventing a single model to be trained on both bands. Second, the model does not extract any interpretable features to understand the cause of the anomaly. In [59], the authors extend this prediction idea on spectrograms and test the model on some synthetic anomalies. A reconstruction based anomaly detector based on vanilla deep autoencoders is presented in [60]. This model lacks interpretable feature extraction properties like class labels which implies the need for training multiple copies of the same model on different bands.

In this chapter we argue that, reconstruction based anomaly detection could be superior to prediction based techniques as prediction is a tougher problem than reconstruction in complex time series datasets. For instance, while digitally modulating signals the basic assumption is each constellation point is selected

with equal probability to maximize the information transfer which makes the prediction of the future symbols difficult. On the other hand reconstruction of input data from compressed features is an easier problem if the model can efficiently capture the complex data distributions. In addition, the time-varying random wireless channel makes prediction of future samples difficult outside the channel coherence time.

We propose SAIFE, an AAE based model which fills the shortcomings of these SoA models. First, we show that a *single model can be trained over multiple bands* in an unsupervised fashion avoiding the need for multiple copies of models on various bands. Second, the same model can be *trained in a semi-supervised fashion for extracting interpretable features* such as signal bandwidth and position. Third, the reconstructed signal from the proposed model can be used for *localizing anomalies* in the wireless spectrum. Furthermore we explore various other advantages of the model such as *wireless data compression* and *signal classification* which are significant contributions in contrast to the SoA models [58–60].

The rest of the chapter is organized as follows. The anomaly detection problem is clearly stated in Section 3.2. Section 3.3 explains the AAE model used for anomaly detection along with the implementation details. The dataset and the parameters used for training are presented in Section 3.4. Section 3.5 details the performance results and discusses the advantages of the proposed model. Section 3.6 explores the signal compression and classification features of the model. Conclusions and future work are presented in Section 3.7.

3.2 Problem Definition

Given: Let X_S be the source time-series data, where $\mathbf{x} \in X_S$ could be either a complex IQ vector or a frequency-based PSD vector from any wireless frequency band. The dataset X_S contains wireless signals that are assumed to be normal behavior. Thus the probability of anomalous behavior in this source dataset is assumed to be low. The superset $X_S = X_{S0} \cup X_{S1} ... \cup X_{Sn}$ contains signals from various frequency bands.

Goal: A model that learns the source data distribution $p(X_S)$ and detects when a target vector's distribution deviates from the source data distribution. For each target vector $\mathbf{x} \in X_T$, X_T being the test dataset, the model should infer whether the vector is normal (H_0) or anomalous (H_a), where H_0 and H_a are hypothesis listed below.

- H_0: Sample data comes from $p(X_S)$



(a) Vanilla autoencoder

(b) Variational autoencoder

Figure 3.2: (a) Encoder decoder structure of an unsupervised vanilla autoencoder model, (b) Stochastic variant of the autoencoder where the internal representations are probability distributions in general

- H_a: Sample data does not come from $p(X_S)$

A signal type of $\mathbf{x} \in X_{Sl}$ from frequency band l occurring in a band k where we are expecting X_{Sk} is also an anomalous behavior which demands the model to capture class labels for fine grained anomaly detection.

Assumptions:

1. The probability of anomalous behavior in the source dataset is very low.
2. No explicit anomaly labeling is done on the source and target dataset.
3. No expert feature extraction is performed before feeding data to the model.

3.3 Models

We leverage the recent advances in generative modeling using neural networks which are trained through back-propagation directly from data [61–64]. The key insight of these previous work is to bring the higher dimensional input

data to some lower dimensional latent space ($\mathbf{Z}$), whose prior distributions can be specified. This latent space which captures relevant features or settings can be then used to reconstruct the actual input data, ideally with minimal reconstruction loss. A basic introduction to some of the recent SoA generative models are covered in the following subsections.

3.3.1 Autoencoder and Variational Autoencoder (VAE)

A traditional autoencoder, as shown in Figure 3.2, is a neural network that consists of an encoder (E) and a decoder (D). The encoder and decoder are trained to reduce the reconstruction loss. This entire network basically performs a non-linear dimensionality reduction optimizing the encoder and decoder parameters (θ and ϕ), the neural network weights, to achieve minimum reconstruction loss such as minimum squared error as given below

$$\theta, \phi = \arg\min_{\theta,\phi} ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{3.1}$$

A VAE [61] also makes use of an its encoder-decoder structure. VAEs encode the input data vector to a vector $\mathbf{z}$ in the latent space $\mathbf{Z}$ whose priors can be imposed by using a Kullback-Leibler (KL) divergence penalty. VAE optimizes the network parameters θ and ϕ to minimize the following upper-bound on the negative log-likelihood of $\mathbf{x}$, where p_{data} is the distribution of the data $\mathbf{x}$:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim p_{data}}[-log\ p(\mathbf{x})] < \mathbb{E}_{\mathbf{x}\sim p_{data}}[\mathbb{E}_{\mathbf{z}\sim q_\theta(\mathbf{z}|\mathbf{x})}[-log(p_\phi(\mathbf{x}|\mathbf{z})]] \\ + \mathbb{E}_{\mathbf{x}\sim p_{data}}[KL(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]\end{aligned} \tag{3.2}$$

Thus VAE optimize the reconstruction loss (first term) similar to a standard autoencoder but adds regularization terms (second term: KL divergence or cross-entropy term) which helps it to learn a latent representation that is consistent with the defined prior $p(\mathbf{z})$.

3.3.2 Adversarial autoencoder (AAE)

Adversarial autoencoders [62] make use of the recent advances in generative modeling [63] to replace the KL divergence in VAEs with adversarial training that encourages the decoder to map the imposed prior to the data distribution. Thus AAE provides two major advantages over VAE: (i) the model ensures that the decoder will generate meaningful samples if we sample from any part of the prior space and (ii) as the aggregate posterior matches the prior distribution,

variations of these distributions can be used for detecting unknown data inputs which is very useful for applications such as anomaly detection. In addition, AAE provides a flexible and robust architecture for semi-supervised learning and data visualization.

3.3.3 SAIFE description

We make use of a deep learning model based on AAE to enable all the requirements mentioned in the problem definition as shown in Figure 3.3. An LSTM layer with 512 cells is used as the encoder for extracting interpretable features while a Convolutional Neural Network (CNN) based decoder is employed for reconstructing the input data from the extracted features. The AAE architecture is trained in a semi-supervised fashion for making the features more interpretable while the reconstruction is fully unsupervised. Two layer feed forward networks with 256 cells and relu activations are employed in both discriminators. The LSTM output is fed through a softmax layer for signal classification and a linear layer for extracting the latent features.

The discriminators (D_s) are neural networks that evaluate the probability that the latent code $\mathbf{z}$ is from the prior distribution $p(\mathbf{z})$ that we are trying to impose rather than a sample from the output of the encoder (E) model. The discriminator receives $\mathbf{z}$ from both the encoder and the prior distribution and is trained to distinguish between them. The encoder is trained to confuse the discriminators into believing that the samples it generates are from the prior distribution. Thus the encoder is trained to reach the solution by optimizing both networks by playing a min-max adversarial game which is expressed in [63] as

$$\min_{E} \max_{D_s} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[log(D_s(\mathbf{z}))] + \mathbb{E}_{\mathbf{x}\sim p_{data}}[log\ (1 - D_s(E(\mathbf{x})))] \tag{3.3}$$

Generative models try to model the underlying distributions of the input data, the latent variables, which are further used for data reconstruction. In SAIFE, the input PSD data is assumed to be generated by the latent *Class* variable which comes from a Categorical distribution with number of categories k =*number of frequency bands* and the continuous latent *Features* from a Gaussian distribution of zero mean and unit variance; $p(\mathbf{y}) = Cat(\mathbf{y})$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$.

3.3.4 Implementation details

The model is implemented using TensorFlow [65], a data flow graph based numerical computation library from Google. Python and C++ bindings of

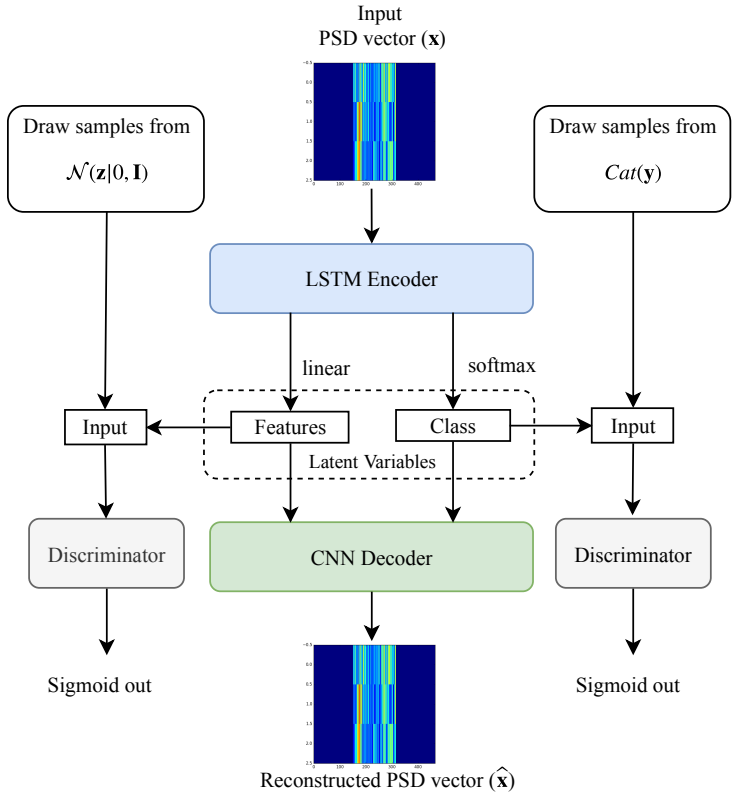


Figure 3.3: Model architecture for anomaly detection.

Tensorflow makes the usage of the final trained model easily portable to host based SDR frameworks like GNU Radio [66]. The trained model can be easily imported as a block in GNU Radio which can be readily used in practice with any supported hardware front-end. These models can be quantized and deployed on FPGAs or embedded CPUs close to the radio frontend for improving processing speeds [40,67]. In addition, the availability of new machine learning accelerators such as Intel neural compute stick combined with low-cost single-board computers such as Raspberry Pi enable wide deployment and realtime inference of deep learning models easier.

Type	single-cont, single-rshort, mult-cont, dethop
Input frame size	6x64
SNR Range	5dB to +20dB
Number of training samples	6000 vectors
Number of test samples	6000 vectors

Table 3.1: Synthetic signal dataset parameters.

Type	scont, randpulses, wpulse, oclass
Input frame size	6x64
SNR Range	-20dB to +20dB
Number of training samples	6000 vectors
Number of test samples	6000 vectors

Table 3.2: Synthetic anomaly dataset parameters.

3.4 Datasets and model training

We use three spectrum datasets along with one synthetic anomaly set to evaluate the performance of the used model. A synthetic spectrum dataset is necessary to understand the performance of the model in a controlled environment. The synthetic dataset consists of four different signal types with signal parameters as reported in Table 3.1. The signals being (i) *single-cont*: single continuous signal with random bandwidth, signal-to-noise ratio (SNR) and center frequency, (ii) *single-rshort*: pulsed signals in time with similar parameters as *single-cont*, (iii) *mult-cont*: multiple continuous signals with possible overlap and (iv) *dethop*: random bandwidth and SNR signals with deterministic shifts/hops in frequency as depicted in Figure 3.4. Similarly, four synthetic signals (i) *scont*: same as single-cont, (ii) *randpulses*: random pulsed transmissions on the given band, (iii) *wpulse*: pulsed wideband signals covering the entire frequency, (iv) *oclass*: signals from other classes in synthetic dataset are used as anomalies.

In addition to the synthetic dataset we validate using two real wireless datasets. The first is a SDR dataset collected using a HackRF SDR from two different cities in Belgium covering frequencies from 10 MHz to 3 GHz. HackRF with its firmware sweep mode can scan the spectrum at up to 8 GHz per second, which allows scanning of 0-6 GHz under a second. Twelve frequency bands are selected from these spectrum scans, continuous in time covering various audio and video broadcast, GSM and LTE bands with a frequency resolution of

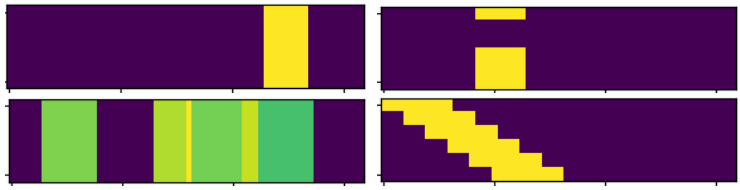


Figure 3.4: Sample signals *single-cont*, *single-rshort*, *mult-cont* and *dethop* from synthetic signal dataset (time on y-axis and frequency on x-axis).

100 KHz whose frequency ranges are listed in Table 3.3. The second dataset consists of PSD sensor data from multiple Electrosense sensors deployed all over Europe retrieved through the open API [38] with 7 selected frequency bands as listed in Table 3.3.

Dataset and bands	Frequencies (MHz)
SDR dataset, 0-11	80-107, 109-115.5, 117-140, 166-172, 196-208, 212.5-217.5, 220-227.5, 388-396, 422-427, 640-660, 790-800, 920-960.
Electrosense dataset, 0-6	86-108, 192-197, 790-801, 801-810, 811-821, 933-935, 955-960.

Table 3.3: SDR and Electrosense dataset frequency bands.

3.4.1 Model training

All the datasets mentioned in the previous section are split into two subsets, a training and a testing subset, with equal number of vectors. A seed is used to generate random mutually exclusive array indices, which are then used to split the data into two ascertaining the training and testing sets are entirely different. The train and test datasets for performance analysis on the SDR and Electrosense datasets are selected from different, non-adjacent, time periods to make sure that there are no identical points in the training and test dataset. The model is trained in an unsupervised fashion to reduce the mean squared error between the input and decoder output and a semi-supervised fashion to learn the the continuous features and class labels. The adversarial networks as well as the autoencoder are trained in three phases: the reconstruction, regularization and semi-supervised phase as mentioned in [62]. The Adam optimizer [68], a first-order gradient based optimizer, with a learning rate of 0.001 is used for training in all the phases. In the semi-supervised phase the model is trained to learn the class, position and bandwidth of the input signal

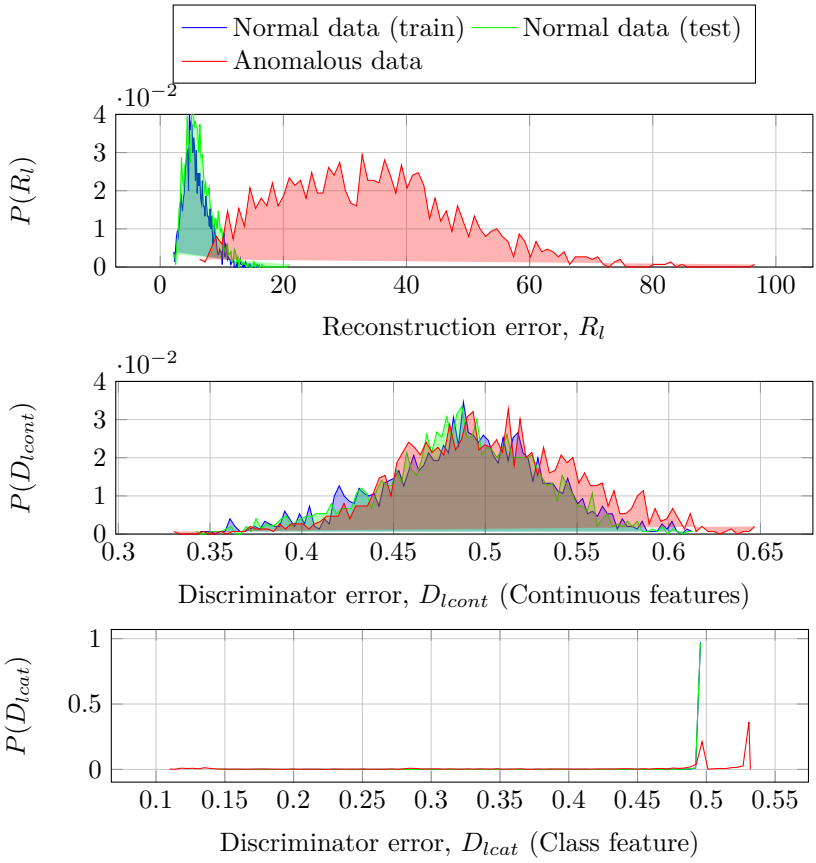


Figure 3.5: Probability density functions of reconstruction error, continuous discriminator error and categorical discriminator error for *dethop* signal and *dethop* signal with *scont* anomaly.

by training it on 20% of the labeled samples from the training set. The model is trained for 500 epochs which takes around one hour of training time on a x86 PC with Nvidia GeForce GTX 980 Ti graphics card. Once trained, the profiled model inference time for 500 input spectrograms is 0.0456 seconds on the same hardware resulting in a processing time of around 0.91 microseconds per input vector.

3.5 Anomaly detection

Once the training process is complete, the model weights are frozen and new input data is fed to the model. As mentioned in the model architecture section, anomalies are detected primarily based on the reconstruction error of the model. In addition to the reconstruction error, the classification error and the discriminator loss are also used for detecting anomalous behaviors.

3.5.1 Detection scores

Three scores are used to detect whether the input data frame is anomalous or not. They are

1. Reconstruction loss: This error measures the similarity between the input data and the reconstructed data defined as $R_l = \sum_{i=0}^{N} |\mathbf{x} - \hat{\mathbf{x}}|$ where $\mathbf{x}$ is the frame input, $\hat{\mathbf{x}} = D(\mathbf{z})$ is the decoder frame output and N is the number of data points in the frame.
2. Discriminator loss: The discriminator in the AAE model is trained to distinguish between the samples from the prior distribution and the samples generated by the encoder. We use the same discrimination loss used during the training process which is defined as $D_l = \sigma(\mathbf{z}, 1)$ where σ is the sigmoid cross entropy. The loss from both continuous (D_{lcont}) and categorical (D_{lcat}) discriminators are used for computing the final anomaly score.
3. Classification error: The class labels predicted by the encoder is cross checked with the original band of interest for detecting the presence of other known but unexpected signals in a selected frequency band.

A simple n-sigma threshold is employed on the reconstruction and discriminator loss based on the mean and standard deviation values from the training data. An input data frame is classified as anomalous if A_{score} is $True$:

$$\begin{aligned} A_{score} = (R_l > (\mu_{R_{lt}} + n * \sigma_{R_{lt}})) \\ \vee ((\mu_{D_{ltcont}} - n * \sigma_{D_{ltcont}}) > D_{lcont} > (\mu_{D_{ltcont}} + n * \sigma_{D_{ltcont}})) \\ \vee ((\mu_{D_{ltcat}} - n * \sigma_{D_{ltcat}}) > D_{lcat} > (\mu_{D_{ltcat}} + n * \sigma_{D_{ltcat}})) \\ \vee (Class_{Encoder} ! = Class_{input}) \end{aligned} \quad (3.4)$$

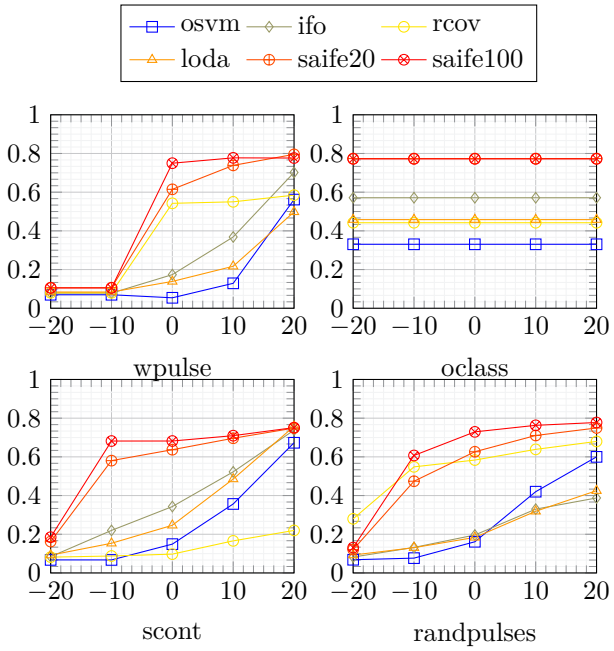


Figure 3.6: Anomaly detection accuracies for different anomalies with a constant false alarm rate of 10% averaged over four different frequency bands. For *oclass* anomaly, anomaly vectors are randomly selected from other classes without specific SNR based evaluation resulting in one detection accuracy value (plotted as a line for uniformity). Anomaly SNR (dB) on the x-axis and detection accuracy on the y-axis

The threshold value n is selected empirically based on the expected true positive rate and false detection rate. From the probability distributions of *dethop* signal and *dethop* signal with *scont* anomaly shown in Figure 3.5, it can be clearly noticed that the reconstruction loss along with class labels plays a major for anomaly detection.

3.5.2 Performance comparisons

To evaluate the performance of SAIFE, the anomaly detection performance is compared against various SoA algorithms such as One class Support Vector Machine (OSVM), Isolation Forest (IFO)[69], Lightweight on-line detector of anomalies (LODA)[70] and Robust Covariance (RCOV)[71]. The average anomaly detection accuracy of these algorithms over different frequency bands are plotted in Figure 3.6. On an average SAIFE performs better than all other algorithms for all anomalies on all synthetic frequency bands. *Oclass*

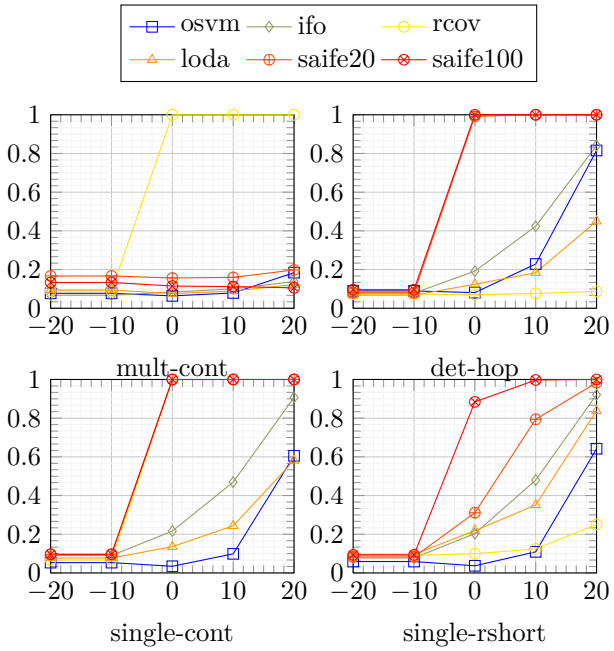


Figure 3.7: Anomaly detection accuracies for *wpulse* anomaly with a constant false alarm rate of 10% on four different frequency bands. Anomaly SNR (dB) on the x-axis and detection accuracy on the y-axis

anomaly performance is quite good when compared to other algorithms as SAIFE performs explicit frequency band classification as one of the features. Detailed detection accuracies for various bands are shown in Figures 3.7, 3.8 and 3.9 for reference. Figure 3.11 shows the Receiver operating characteristic (ROC) curves for all anomalies on the *det-hop* channel for all algorithms. Anomaly signals similar to the original signals are intentionally selected to thoroughly analyze the detection capabilities of the model. For instance, from Figure 3.8, it can be seen that detection of *scont* anomaly is difficult in *mult-cont* band as another continuous signal is not an anomalous behavior in the multiple continuous signal band. Similarly detection of *wpulse* works well only on SNRs above 0 dB as the signal is only visible above the noise floor above 0 dB. Improving the number of features of SAIFE from 20 to 100 can also help to improve the detection accuracy to some extent as shown in Figure 3.6. More detailed analysis on optimal number of features can be found in section 3.6.

These experiments are repeated on the SDR dataset and the results are plotted in Figure 3.12. Only the two best and worst performing frequency bands

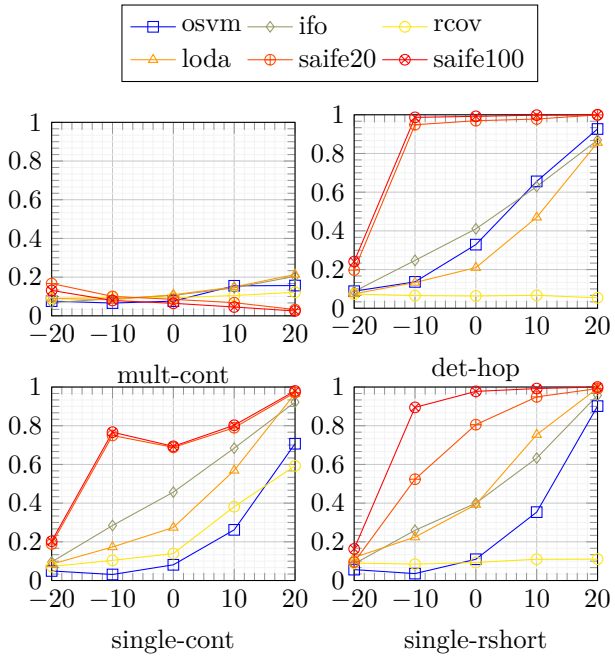


Figure 3.8: Anomaly detection accuracies for *scont* anomaly with a constant false alarm rate of 10% on four different frequency bands. Anomaly SNR (dB) on the x-axis and detection accuracy on the y-axis

for different anomalies based on the Area Under Curve (AUC) for the lowest anomaly SNR of -20dB are shown due to space limitations, as there are 12 frequency bands in the SDR dataset. Results similar to the synthetic dataset can be noticed in the real capture SDR dataset also. Detecting *scont* and *randpulses* anomalies in frequency band 0 (80-107 MHz) is very difficult as the selected band is very wide and it contains strong FM broadcast stations. Similar results can be noticed in the the other worst performing band 11 (920-960 MHz) which contains GSM signal transmissions that includes both continuous and hopping transmissions. This shows the pressing need to split the 40 MHz bandwidth to multiple bands, for instance continuous and random hopping bands, for better detection accuracies. It can be also noticed that the *oclass* detection accuracies are quite good even in the worst performing band 0 showing the robustness of the signal classification module of the encoder.

The anomaly detection accuracies of SAIFE are also compared against other algorithms on the SDR dataset as shown in Figure 3.10. On the high SNR regions OSVM and LODA gives a better or close enough performance as SAIFE.

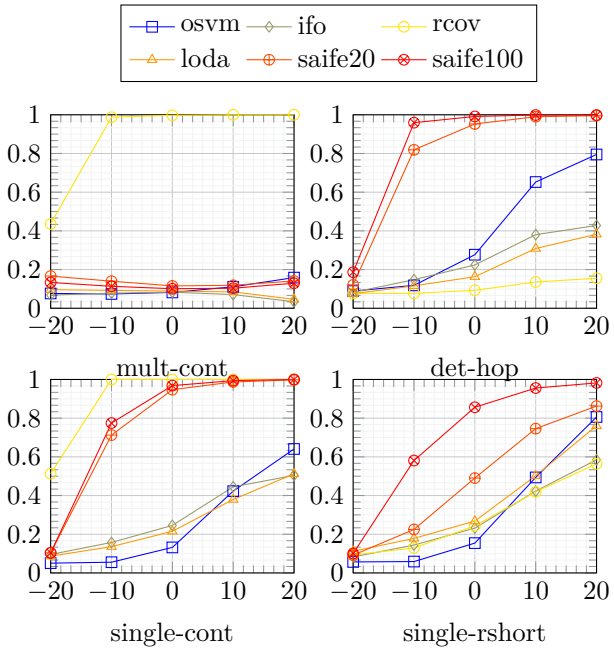


Figure 3.9: Anomaly detection accuracies for *randpulses* anomaly with a constant false alarm rate of 10% on four different frequency bands. Anomaly SNR (dB) on the x-axis and detection accuracy on the y-axis

On the low SNR regions SAIFE performs better. Once again SAIFE adds much more interpretability to the whole anomaly detection process which is a huge advantage.

3.5.3 Anomaly localization

Localizing anomalies in the wireless frequency spectrum is not common in any of the SoA algorithms. SAIFE presents a simple and robust way to localize the anomalous region from the input PSD data which is a significant contribution of this chapter. In addition to detection of anomalies, the reconstruction error along with the semi-supervised features can be used to localize and understand the anomaly better as shown in Figure 3.13. Anomaly localization is achieved by plotting the absolute reconstruction error, that is $|\hat{\mathbf{x}} - \mathbf{x}|$. This method works well unless there is a drastic change in the estimated class label which can be noticed in the third row where an *scont* anomaly occurs in an *srshort* band. The model accurately detects it as an anomaly since there is a variation

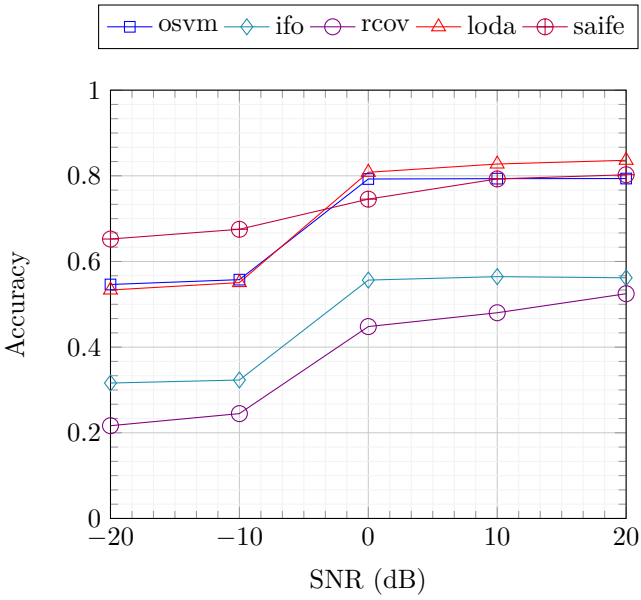


Figure 3.10: Anomaly detection accuracies for different algorithms with a constant false alarm rate of 1% averaged over all frequency bands and anomalies. Anomaly SNR (dB) on the x-axis and detection accuracy on the y-axis

in the estimated class label, but shows the *srshort* signal as the anomalous region instead of *scont*. Figure 3.15 gives some sample plots of the estimated signal position and bandwidth. The current model is only trained for three semi-supervised features including the class label and these interpretable features can be used for analyzing the anomalies better.

3.5.4 Anomaly detection in the wild

To understand the performance on detecting real anomalies, the model is tested on the real-world Electrosense dataset. The model is trained on 7 days of data from one of the Electrosense sensors and tested on the next 500 hours for anomalies with a detection threshold of 3σ ($n = 3$). The number of detected anomalies, based on A_{score}, along with a few sample anomalies for 7 frequency bands are shown in Figure 3.14. The model detects unexpected missing transmissions (top-right and bottom-right), high power transmissions (bottom-left) and some out of band transmissions (top-left). It can be noticed that after 230 hours the 192-197 MHz bands started giving more anomalous detections. Visual inspection of the anomalous PSD patches in this band

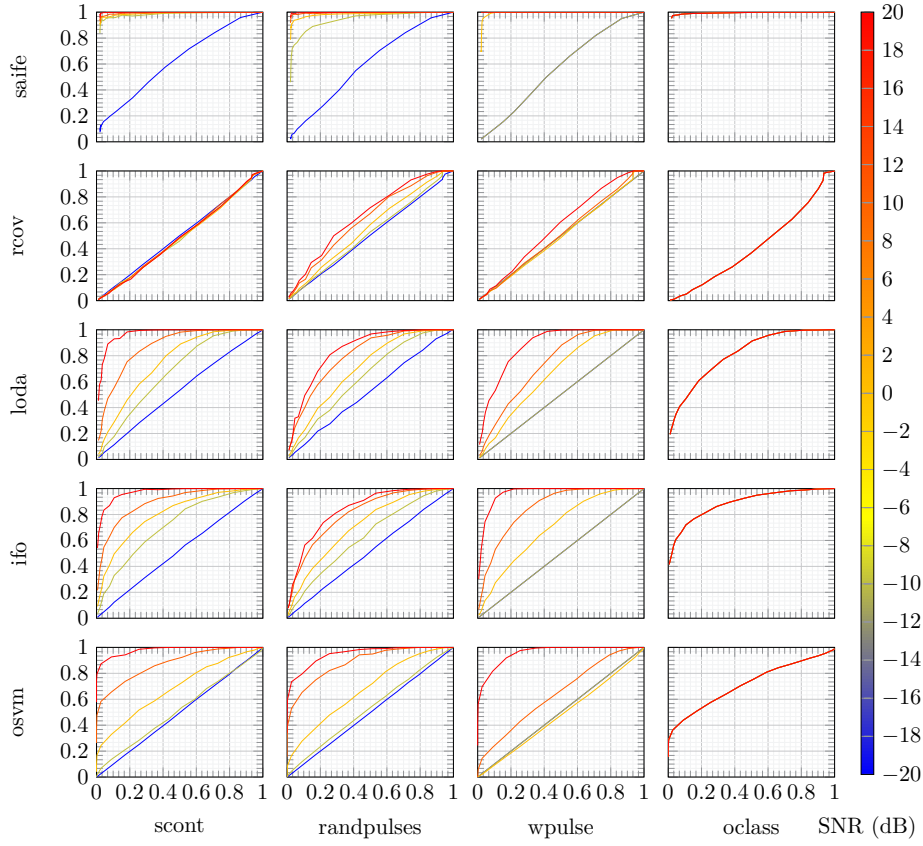


Figure 3.11: ROC curves for different detection algorithms on *det-hop* synthetic band for various anomalies.

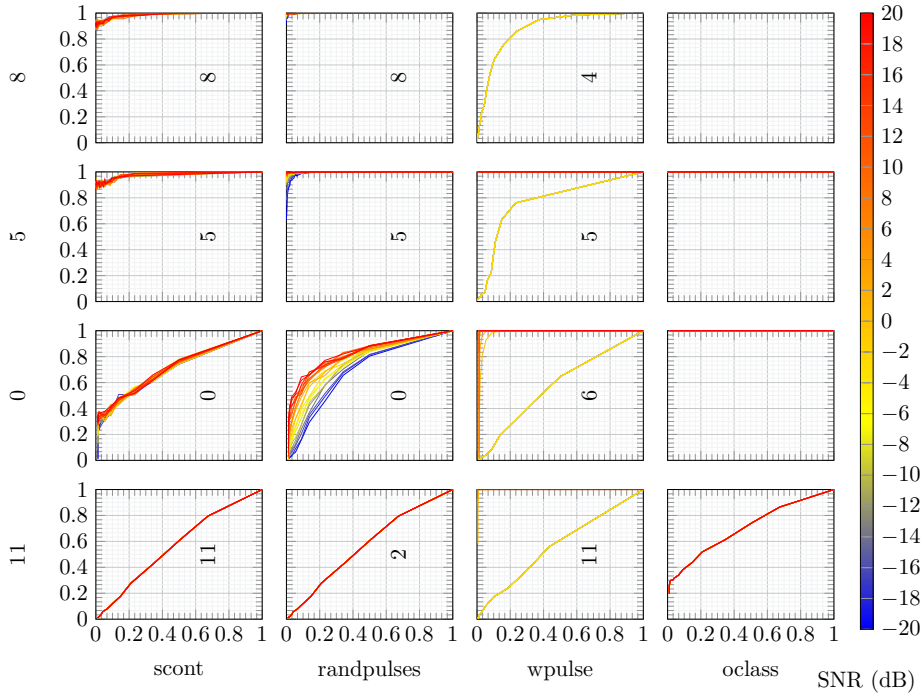


Figure 3.12: ROC curves for the best two (top rows) and worst two (bottom rows) ROC AUC for -20dB SNR in SDR dataset. Synthetic bands with band labels are shown in rows for each synthetic anomaly (columns). On each plot false positive rate is represented on x-axis and true positive rate on y-axis. For *oclass* anomaly, anomaly vectors are randomly selected from other classes and specific SNR based ROC curves are not plotted.

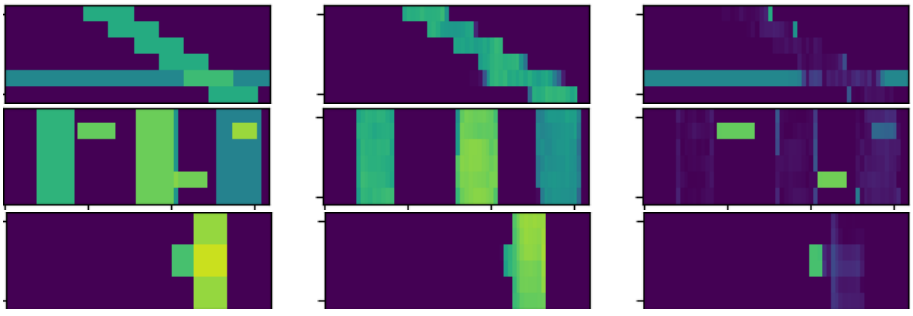


Figure 3.13: Localized anomalies for three different synthetic anomalies. Original input signal, decoder reconstructed signal and the localized anomaly is shown in each row from left to right. First row: *wpulse* anomaly on *dethop* signal, second row: *randpulses* anomaly on *mult-cont* signal, third row: *scont* anomaly on *single-rshort* signal.

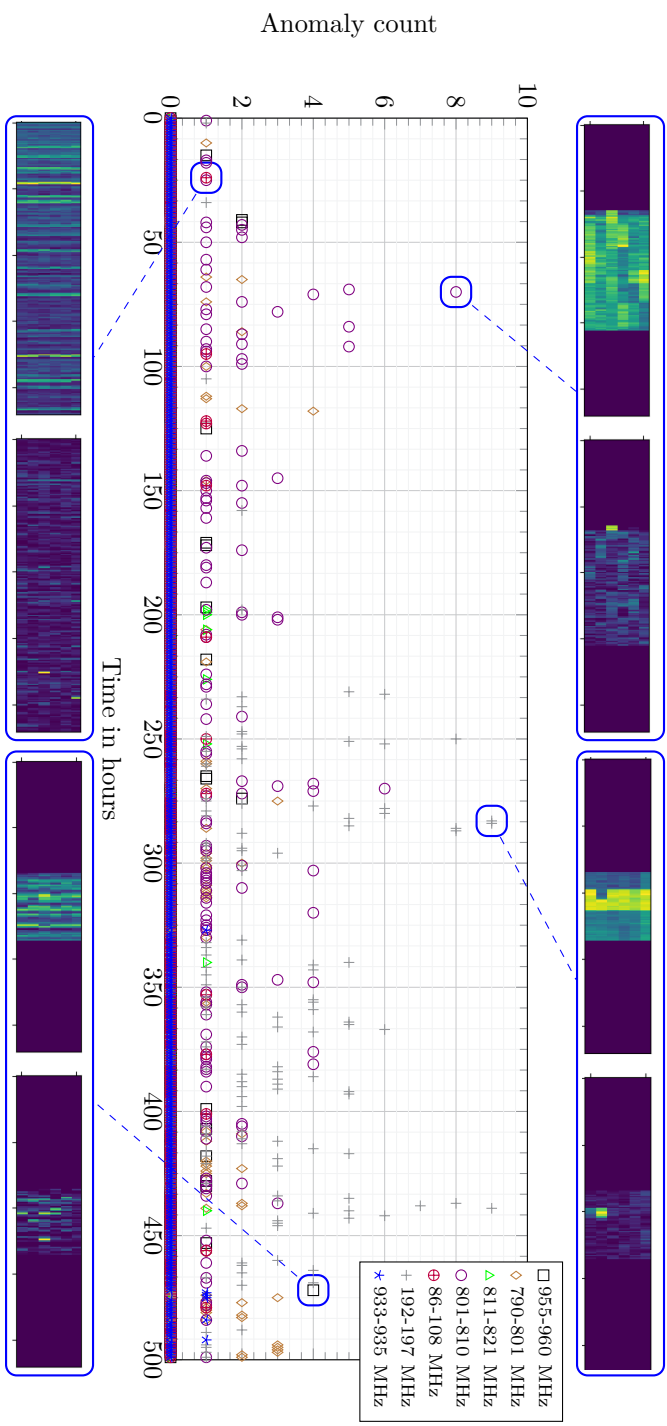


Figure 3.14: Detected anomalies for a duration of 500 hours from one of the Electrosense sensors. Sample input data (left) and the localized anomaly (right) for some sample anomalies are also plotted for some frequency bands.

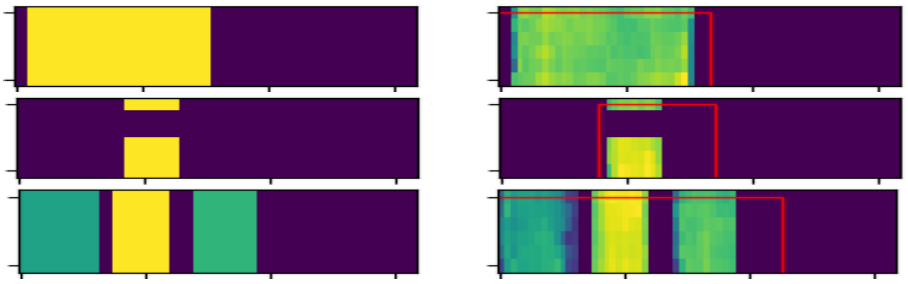


Figure 3.15: Learned bandwidth and position features in a semi-supervised fashion. On each row the left one is the input signal and the right one is the reconstructed signal along with the estimated parameters.

revealed transmission pattern variations. These detected variations could be either because of the transmitter behavior changes or from the position/antenna changes of the sensor. The model also provides the flexibility to add these anomalous detections to the training set, enabling incremental learning, if the user believes that the behavior is normal. The model retraining complexity is quite moderate as detailed in section 3.4.1. Incorporating this user feedback and enabling automated retraining of models on these kind of anomalous behaviors will be addressed in future work.

3.6 Signal compression and classification

To control the data transfer costs associated with the sensing, Electrosense sensors enable three pipelines with very low, medium and high data transfer costs namely: Feature, PSD and IQ pipeline. While the IQ pipeline allows to send raw data to the backend, which can be used to support a broad range of applications, such as advanced modulation classification and signal decoding, the data transfer rate required is in the 30 Mbps to 100 Mbps range based on the sampling rate of the sensor. The PSD pipeline on the other hand brings down this rate to hundreds of Kbps. In this section we analyze the compression and classification capabilities of SAIFE to reduce the associated data transfer costs.

3.6.1 Traditional Spectrum Representation

In spite of the popularity of various lossy and lossless compression algorithms in image and video processing communities, there are only a few compression

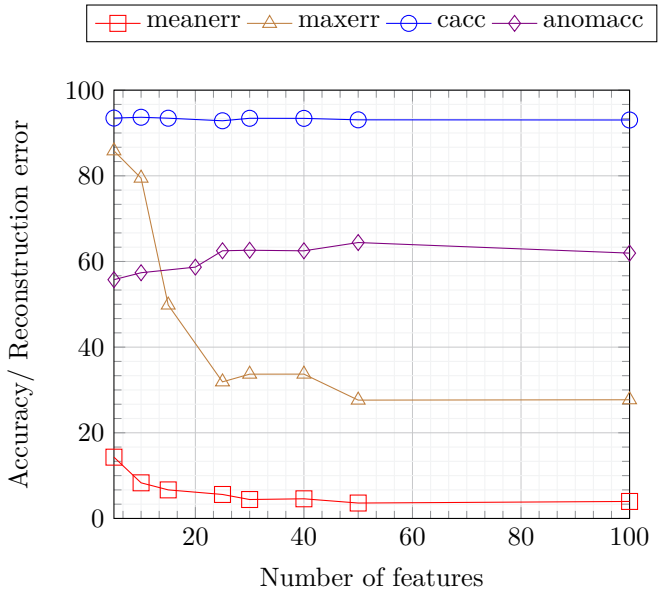


Figure 3.16: Overall averaged anomaly detection accuracy (anomacc), wireless frequency band classification accuracy (cacc), mean absolute reconstruction error and maximum absolute reconstruction error of SAIFE on the synthetic dataset for a false alarm rate of 10%.

algorithms fine tuned for wireless spectrum data. In [17] the authors presented a compression algorithm based on Chebyshev polynomials. The authors in [18] presented a method to separate spectrum noise and other relevant signals specific to L-band satellite signals and then did separate compression to achieve better results when compared to JPEG standards. The aforementioned methods are very specific and lack the compression flexibility when the input data is in multiple formats such as PSD or IQ.

3.6.2 Non-linear data compression

Recently unsupervised deep learning models have shown great improvements in compressing input information. In [72] the authors have achieved 4X to 16X compression ratio on raw sampled IQ data using autoencoder models. In SAIFE, 20 compressed features are used for representing the input PSD frame. This helps to achieve a lossy compression of 19X, 60X and 120X on the Synthetic, Electrosense and SDR datasets respectively, which can considerably reduce the data transfer costs. Mean absolute reconstruction error of SAIFE with 20 features are summarized in Table 3.4. In addition to spectrum reconstruction

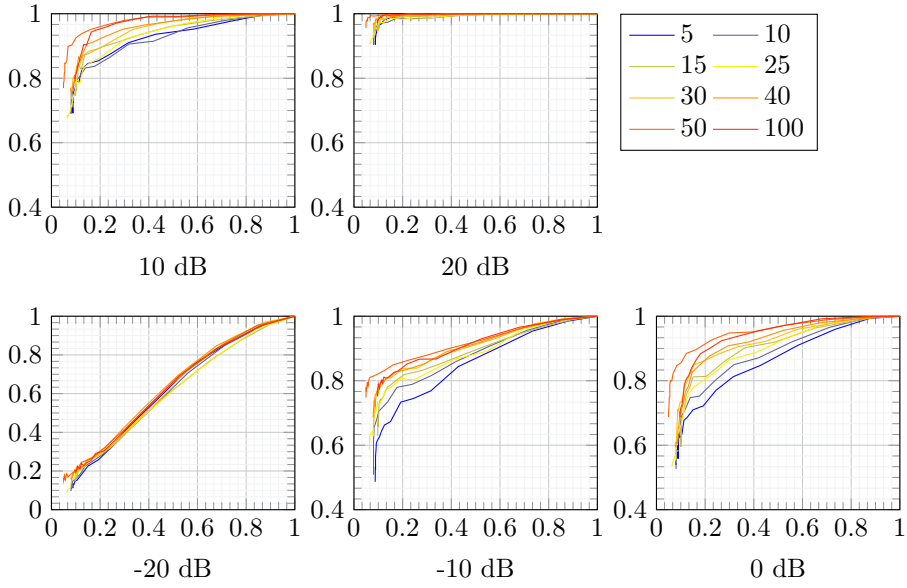


Figure 3.17: ROC curves for *single-rshort* signal with *scont* anomalies at different SNRs with varying number of continuous features.

these features can be used for anomaly detection and signal classification which makes it more attractive. The models can be easily adapted for different data inputs, for instance PSD data in time and frequency or IQ data supporting flexible compression architectures for different sensor data pipelines. The dimension of the compressed feature space along with the model complexity can be adapted to suit the reconstruction loss requirements. For instance, the number of features required to represent the time-frequency PSD patches of static wireless channels like commercial FM bands will be very less when compared to very random hopping channels.

An initial analysis is performed, on the synthetic dataset, to understand the trade-off between level of compression and anomaly detection performance by varying the number of continuous features, thereby the compression ratio of the model, which is presented in Figure 3.17. At very low (-20 dB) and high (20 dB) anomaly SNRs there are not much performance gains by increasing the number of features as expected. Detecting signals at -20 dB SNR is very difficult even with a large number of features whereas at 20 dB SNR with a smaller set of features can easily detect anomalies due to large variations in the reconstruction loss. While at common SNR values (-10dB, 0 and 10dB) the

anomaly detection performance increases with increasing number of features. We would like to emphasize that the number of features required to achieve reasonable detection performance will depend on the input data dimensions, the encoder and decoder capacity and the dataset complexity itself.

Figure 3.16 presents the variations in the overall anomaly detection accuracy along the maximum and mean reconstruction error and the wireless frequency band classification performance with increasing number of intermediate features. It can be noticed that the reconstruction error reduces steadily till one point (50 features) and then saturates afterwards as expected. We strongly believe that lower reconstruction loss can be achieved by using very deep and robust CNN models which are successfully used in high resolution image reconstruction problems. For these future models, even with lower reconstruction error we expect a similar saturation with increasing number of features. The lower anomaly detection accuracies (around 60%) is attributed to the averaging of accuracies over different bands, anomalies and SNRs. The wireless band classification accuracy is not that affected by varying number of features as one of the features is *class* which is trained in a semi-supervised fashion.

Dataset	Classification accuracy (%)	Mean absolute reconstruction error
Synthetic dataset	92.86	7.67 (for 6x64 samples)
SDR dataset	100	84.12 (for 6x400 samples)
Electrosense dataset	100	101.16 (for 6x221 samples)

Table 3.4: Band classification accuracy and reconstruction errors on the test data of different datasets.

3.6.3 Wireless signal classification

In addition to anomaly detection ROC curves, wireless band classification accuracies on the test data of three datasets are summarized on Table 3.4. Since the real wireless bands use different parameters such as signal bandwidths, modulation type, and temporal occupancies at mostly high SNRs, the wireless band classification problem is not very tough as the classical modulation classification problem [40]. On the synthetic dataset the model confuses between *single-cont* and *mult-cont* signal resulting in a classification accuracy of 92.86%. The confusion matrix for the same is also shown in Figure 3.18. The model achieves excellent classification accuracy of 100% on the real SDR and Electrosense dataset. The high classification accuracy stress the fact that a

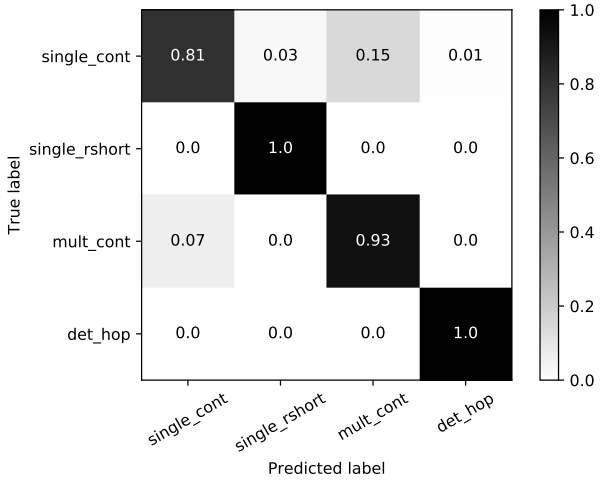


Figure 3.18: Confusion matrix for the synthetic dataset

categorical variable helps the encoding process which in-turn helps the decoder to generate fine variations which are specific to a particular class.

3.7 Conclusion and Future work

Automated monitoring of wireless spectrum over frequency, time and space is still a difficult research problem. In this chapter we have analyzed the use of an AAE model in wireless spectrum data anomaly detection, compression and signal classification. We have shown that the proposed model can achieve good anomaly detection and localization along with interpretable feature extraction. The model also can achieve a wireless band classification accuracy close to 100% by only using 20% labeled samples. Further, the performance of the proposed model is compared against various SoA anomaly detection algorithms in literature showing its robustness. To understand the anomalous signals better, we look into the signal classification problem in detail in the next chapter.

Chapter 4

Signal classification

This chapter looks into the modulation classification problem for a distributed wireless spectrum sensing network. First, a new data-driven model for Automatic Modulation Classification (AMC) based on long short term memory (LSTM) is proposed. The model learns from the time domain amplitude and phase information of the modulation schemes present in the training data without requiring expert features like higher order cyclic moments. Analyses show that the proposed model yields an average classification accuracy of close to 90% at varying SNR conditions ranging from 0dB to 20dB. Further, we explore the utility of this LSTM model for a variable symbol rate scenario. We show that a LSTM based model can learn good representations of variable length time domain sequences, which is useful in classifying modulation signals with different symbol rates. The achieved accuracy of 75% on an input sample length of 64 for which it was not trained, substantiates the representation power of the model. To reduce the data communication overhead from distributed sensors,



Figure 4.1: Fast signal classification from a few modulated symbols can be achieved using SoA deep learning models.

the feasibility of classification using averaged magnitude spectrum data and on-line classification on the low-cost spectrum sensors are studied. Furthermore, quantized realizations of the proposed models are analyzed for deployment on sensors with low processing power.

This chapter is adapted from: Rajendran, Sreeraj; Meert, Wannes; Giustiniano, Domenico; Lenders, Vincent; Pollin, Sofie; *Deep Learning Models for Wireless Signal Classification with Distributed Low-Cost Spectrum Sensors.* IEEE Transactions on Cognitive Communications and Networking; 2018; Vol. 4; iss. 3; pp. 433 - 445.

4.1 Introduction

Wireless spectrum monitoring over frequency, time and space is important for a wide range of applications such as spectrum enforcement for regulatory bodies, generating coverage maps for wireless operators, and applications including wireless signal detection and positioning. Continuous spectrum monitoring over a large geographical area is extremely challenging mainly due to the multidisciplinary nature of the solution. The monitoring infrastructure requires proper integration of new disruptive technologies than can flexibly address the variability and cost of the used sensors, large spectrum data management, sensor reliability, security and privacy concerns, which can also target a wide variety of the use cases. Electrosense was designed to address these challenges and support a diverse set of applications [38]. Electrosense is a crowd-sourced spectrum monitoring solution deployed on a large scale using low cost sensors.

One of the main goals of Electrosense is to accomplish automated wireless spectrum anomaly detection, thus enabling efficient spectrum enforcement. Technology classification or specifically Automatic Modulation Classification (AMC) is an integral part of spectrum enforcement. Such a classifier can help in identifying suspicious transmissions in a particular wireless band. Furthermore, technology classification modules are fundamental for interference detection and wireless environment analysis. Considering the aforementioned large application space this chapter looks into two key aspects: Is efficient wireless technology classification achievable on a large scale with low cost sensor networks and limited uplink communication bandwidth? If possible, which are the key classification models suitable for the same.

The number of publications related to AMC appearing in literature is large [21,73–77] mainly due to the broad range of problems associated with AMC and huge interest in the problem itself for surveillance applications. AMC helps a radio system for environment identification, defining policies and taking actions

for throughput or reliability improvements. It is also used for applications like transmitter identification, anomaly detection and localization of interference [73,74].

Various approaches for modulation classification discussed in literature can be brought down to two categories [75], one being the *decision theoretic* approach and the other the *feature based* approach. In decision-theoretic approaches the modulation classification problem is presented as a multiple hypothesis testing problem [75]. The maximum likelihood criterion is applied to the received signal directly or after some simple transformations such as averaging. Even though decision-theoretic classifiers are optimal in the sense that they minimize the probability of miss-classifications, practical implementations of such systems suffer from computational complexity as they typically require buffering a large number of samples. These methods are also not robust in the presence of unknown channel conditions and other receiver discrepancies like clock frequency offset.

Conventional feature-based approaches for AMC make use of expert features like cyclic moments [21]. Spectral correlation functions of various analog and digital modulation schemes covered in [22] and [23] respectively are the popularly used features for classification. Detailed analysis of various methods using these cyclostationary features for modulation classification are presented in [76]. Various statistical tests for detecting the presence of cycles in the kth-order cyclic cumulants without assuming any specific distribution on the data are presented in [77]. In [78] authors used a multilayer linear perceptron network over spectral correlation functions for classifying some basic modulation types. Another method makes use of the cyclic prefix [79] to distinguish between multi-carrier and single carrier modulation schemes which is used for Orthogonal Frequency Division Multiplexing (OFDM) signal identification.

All these aforementioned model driven approaches exploit knowledge about the structure of different modulation schemes to define the rules for AMC. This manual selection of expert features is tedious which makes it difficult to model all channel discrepancies. For instance, it is quite challenging to develop models which are robust to fading, pathloss, time shift and sample rate variations. In addition, a distributed collection of IQ data over frequency, space and time is expensive in terms of transmission bandwidth and storage. Furthermore, most of these algorithms are processor intensive and could not be easily deployed on low-end distributed sensors.

Recently, deep learning has been shown to be effective in various tasks such as image classification, machine translation, automatic speech recognition [29] and network optimization [30], thanks to multiple hidden layers with non-linear logistic functions which enable learning higher-level information hidden in the

data. A recently proposed deep learning based model for AMC makes use of a CNN based classifier [80]. The CNN model operates on the time domain IQ data and learns different matched filters for various SNR. However, this model may not be efficient on data with unknown sampling rates and pulse shaping filters which the model has never encountered during the training phase. Also being a fixed input length model, the number of modulated symbols the model can process remains limited across various symbol rates. Furthermore, the training and computational complexity of the model increases with increasing input sample length. In [55], the authors extended the analysis on the effect of CNN layer sizes and depths on classification accuracy. They also proposed complex inception modules combining CNN and LSTM modules for improving the classification results. In this chapter we show that simple LSTM models can itself achieve good accuracy, if input data is formatted as amplitude and phase (polar coordinates) instead of IQ samples (rectangular coordinates).

This chapter proposes a LSTM [81] based deep learning classifier solution, which can learn long term temporal representations, to address the aforementioned issues. The proposed variable input length model can capture sample rate variations without explicit feature extraction. We first train the LSTM model to classify 11 typical modulation types, as also used in [82], and show our approach outperforms the SoA. Being a variable input length model we also show that the model enables efficient classification on variable sample rates and sequence lengths. Even though these deep learning models can provide good classification accuracies on lower input sample lengths, their computational power requirements are still high preventing them from low-end sensor deployment as Electrosense.

The wireless sensing nodes deployed in the Electrosense network consist of a low-cost and bandwidth limited SDR interfaced with a small sized embedded platform [38]. PSD and IQ pipelines are enabled in the sensor to support various applications producing data in the order of 50-100 Kbps and 50 Mbps respectively. First, the embedded hardware of the sensors is not powerful enough to handle performance intensive AMC algorithms. Second, transferring IQ samples to the backend for classification by enabling the IQ pipeline is not a scalable solution as it is expensive in terms of data transfer and storage. Finally, the sensors are bandwidth limited which prevents them from acquiring wideband signals.

To enable instantiation of the newly proposed LSTM model for modulation classification in a large distributed network of low cost sensor nodes, we compare various approaches to decrease the implementation cost of the classifier. In the first approach we study the advantages and limitations of classification models for modulation classification on a deployed distributed sensor network with limited bandwidth sensors based on averaged magnitude Fast Fourier

Transform (FFT) data which decreases the communication cost by a factor 1000. Moreover, quantized versions of the proposed models are studied in detail for sensor deployment. These quantized versions can be run on a low cost sensor and do not require the instantiation of the classifier in the cloud. As a result, the sensor should only communicate the decision variable, which further decreases the communication cost. The *code and datasets* for all the deep learning models are *made public* for future research[1]. The models are also available for use through Electrosense.

The contribution of this chapter is thus threefold. First, we develop a new LSTM based deep learning solution using time domain amplitude and phase samples which provides SoA results for high SNRs on a standard dataset. Second, we explore the use of deep learning models for technology classification task in a distributed sensor network only using averaged magnitude FFT data. Finally, we explore the model performance by quantizing the deep neural networks for sensor deployment.

The rest of the chapter is organized as follows. The classification problem is clearly stated in Section 4.2. A brief overview of the modulation dataset and the channel models used are presented in Section 4.3. Section 4.4 explains the LSTM model used for classification and the parameters used for training along with other implementation details. Section 4.5 details the classification results and discusses the advantages of the proposed model. Low-implementation cost models are discussed in Section 4.7. Conclusions and future work are presented in Section 4.8.

4.2 Problem Statement

Technology or modulation recognition can be framed as a N-class classification problem in general. A general representation for the received signal is given by

$$r(t) = s(t) * c(t) + n(t), \tag{4.1}$$

where $s(t)$ is the noise free complex baseband envelope of the received signal, $n(t)$ is Additive White Gaussian Noise (AWGN) with zero mean and variance σ_n^2 and $c(t)$ is the time varying impulse response of the transmitted wireless channel. The basic aim of any modulation classifier is to give out $P(s(t) \in N_i|r(t))$ with $r(t)$ as the only signal for reference and N_i represents the ith class. The received signal $r(t)$ is commonly represented in IQ format due to its flexibility and simplicity for mathematical operations and hardware design. The in-phase and

[1] `https://github.com/zeroXzero/modulation_classif`

Modulations	8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, WBFM
Samples per symbol	4
Sample length	128
SNR Range	-20dB to +20dB
Number of training samples	82500 vectors
Number of test samples	82500 vectors

Table 4.1: RadioML2016.10a dataset parameters.

Modulations	8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, WBFM
Samples per symbol	4 and 8 sps
Sample length	128 to 512
SNR Range	-20dB to +20dB
Number of training samples	165000 vectors
Number of test samples	165000 vectors

Table 4.2: Modified complex RadioML dataset parameters.

quadrature components are expressed as $I = Acos(\phi)$ and $Q = Asin(\phi)$, where A and ϕ are the instantaneous amplitude and phase of the received signal $r(t)$.

The RadioML and modified RadioML datasets used for testing the proposed model, presented in the next section of this chapter, follow the signal representation as given in equation 4.1. These datasets make a practical assumption that the sensor's sampling rate is high enough to receive the full-bandwidth signal of interest at the receiver end as $r(t)$. The datasets also take into account complex receiver imperfections which are explained in detail in Section 4.3. The *samples per symbol* parameter used in the tables 4.1 and 4.2 specify the number of samples representing each modulated symbol which is a modulation characteristic. Similarly *sample length* parameter specifies the number of received signal samples used for classification.

4.3 Modulation Datasets

A publicly available dataset used for evaluating the performance of the proposed model is detailed in this section. The standard dataset is also extended to evaluate the sample rate dependence of the proposed model.

4.3.1 RadioML dataset

A standard modulation dataset presented in [82] is used as the baseline for training and evaluating the performance of the proposed classifier. The used RadioML2016.10a dataset is a synthetically generated dataset using GNU Radio [66] with commercially used modulation parameters. This dataset also includes a number of realistic channel imperfections such as channel frequency offset, sample rate offset, additive white gaussian noise along with multipath fading. It contains modulated signals with 4 samples/symbol (sps) and a sample length of 128 samples. Used modulations along with the complete parameter list can be found in Table 4.1. Detailed specifications and generation details of the dataset can be found in [82].

4.3.2 Modified RadioML dataset

The standard radioML dataset is extended using the generation code[2] by varying the *samples per symbol* and *sample length* parameters for evaluating the sample rate dependencies of the LSTM model. The extended parameters of the used dataset are listed in the Table 4.2. The extended dataset contains signals with 4 and 8 samples per symbol. This dataset is generated to evaluate the robustness of the model in varying symbol rate scenarios.

4.4 Model Description

The proposed LSTM model, that works on the time domain amplitude and phase signal, is introduced in the following subsection. In addition, the baseline CNN model used for comparisons is also detailed.

[2]https://github.com/radioML/dataset

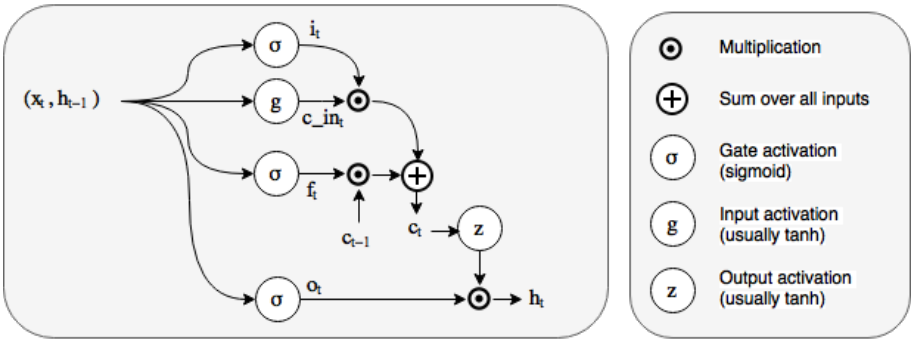


Figure 4.2: LSTM cell used in the hidden layers of the model.

4.4.1 LSTM primer

Recurrent Neural Networks (RNN) are heavily used for learning persistent features from time series data. LSTM [81] is a special type of RNN which is efficient in learning long-term dependencies. The block diagram of a basic version of a LSTM cell is presented in Figure 4.2 along with the corresponding equations (2-7).

- Gates

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4.2}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4.3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{4.4}$$

- Input transform

$$c_in_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c_in}) \tag{4.5}$$

- State update

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c_in_t \tag{4.6}$$

$$h_t = o_t \cdot tanh(c_t) \tag{4.7}$$

LSTM cells have an internal state or memory (c_t) along with three gates namely input date (i_t), forget gate (f_t) and output gate (f_t). Based on the previous state and the input data the cells can learn the gate weights for the specified problem. This gating mechanism helps LSTM cells to store information for longer duration thereby enabling persistent feature learning.

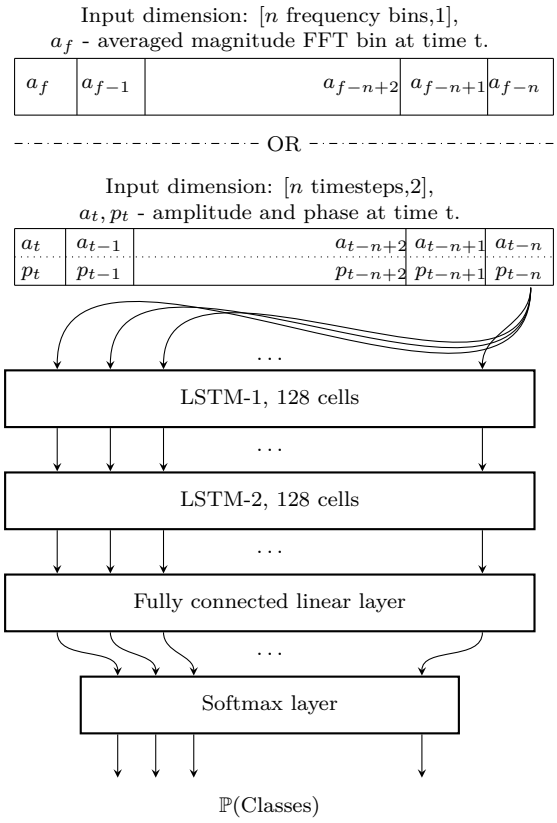


Figure 4.3: Two layer LSTM model for classification. The model is trained and deployed for modulation classification using either amplitude-phase signal or the averaged magnitude-FFT signal as input.

4.4.2 Model for complex signals

A LSTM network with different layers is used for complex data classification as shown in Figure 4.3. The amplitude and phase of the time domain modulated signal are fed to all cells of the LSTM model as a two dimensional vector, at each time step for classification. The amplitude vector is L2 normalized and the phase, which is in radians is normalized between -1 and 1. The first two layers are comprised of 128 LSTM cells each. The final output from the second LSTM layer, a vector of dimension 128, after all time steps, is fed to the last dense layer of the model. The final layer is a dense softmax layer which maps the classified features to one of the 11 output classes representing the modulation

schemes. The two layer model is selected after detailed analysis varying the cell size and layer depths which are detailed in Section 4.5.4.

The intuition to use a LSTM model for classification is based on the fact that different modulation schemes exhibit different amplitude and phase characteristics and the model can learn these temporal dependencies effectively. Even though fading and other real world effects may slightly hamper the characteristics of the signal, we expect the model to classify signals efficiently by learning good fading resistant representations. Since the proposed model can work on variable length input time domain samples, we expect the model to learn useful symbol rate independent representations for classification. In addition, the importance of the number of LSTM cells and layer depth are further investigated by varying these parameters. Model classification accuracies are analyzed with varying layer depth from 1 to 3 and number of cells from 16 to 256. We further analyze these aspects in detail in Section 4.5.

4.4.3 Baseline IQ model

The two layer CNN 8 tap model presented in [55] is used as the baseline model for further comparisons. The baseline model uses 256 and 80 filters in the first two convolutional layers respectively. A publicly available training model is used for generating the baseline performance graph [80].

4.4.4 Model training and testing

Each of the datasets mentioned in Tables 4.1 and 4.2 are split into two, one training set and the other testing set. A seed is used to generate random mutually exclusive array indices, which are then used to split the data into two ascertaining the training and testing sets are entirely different. The number of the training and testing vectors are listed in the corresponding tables. A softmax cross entropy with logits[3], that measures the probability error in discrete classification tasks in which the classes are mutually exclusive, is used as the loss function. Stochastic gradient descent with a minibatch size of 400 vectors is used to avoid local optima. We use the Adam optimizer [68], a first-order gradient based optimizer with a learning rate of 0.001. The complex two layer LSTM network is trained for 70 epochs which takes around an hour of training time on a x86 PC with Nvidia GeForce GTX 980 Ti graphics card. We use a basic LSTM cell with starting training forget bias set to one. While initializing

[3]https://www.tensorflow.org/api_docs/python/tf/nn/softmax_cross_entropy_with_logits

the network, it is helpful to keep the scale of the input variance constant, so that it does not explode or diminish by reaching the final layer. To achieve this LSTM weights are initialized with a default uniform unit scaling initializer which generates weights with a uniform variance. All the models use the same training parameters unless specified explicitly.

4.4.5 Implementation details

The neural network is implemented using TensorFlow [65], a data flow graph based numerical computation library from Google. Python and C++ bindings of Tensorflow makes the usage of the final trained model easily portable to host based SDR frameworks like GNU Radio [66]. The trained model can be easily imported as a block in GNU Radio which can be readily used in practice with any supported hardware front-end.

4.5 Results and discussion

The classification accuracies of the model for the aforementioned datasets along with the learned representations are discussed in the following subsections.

4.5.1 Classification accuracy on RadioML dataset

The two layer amplitude-phase LSTM model, shown in Figure 4.3, is trained on SNR ranges from -10dB to 20dB. Training vectors with SNR ranges below -10dB were not used as the model was converging slowly when those vectors were used. Alternate models with varying LSTM layer depths are also trained to understand the performance improvements provided by the different layer depths. The classification accuracy of all the four models are presented in Figure 4.4. The two layer LSTM model gave an average accuracy of 90% in SNR ranges from 0dB to 20dB. It can be noticed that the single layer LSTM also reaches a high accuracy at high SNRs, 6% less than the two layer model. It was also noticed that the classification accuracy saturates for layer depths of two. Hence, layer depth of two is selected for the final model and its parameters are fine tuned (dropout = 0.8 and learning rate = 0.001) to achieve the best test performance as shown in Figure 4.4. Rigorous fine tuning was not performed on layer depths other than two accounting for a slightly lower accuracy levels, for instance the accuracy level for layer depth 3 is slightly lower than layer depth two. The performance of the baseline CNN model was shown to be much better on the low SNR regions in [55]. We were not able to reproduce the reported

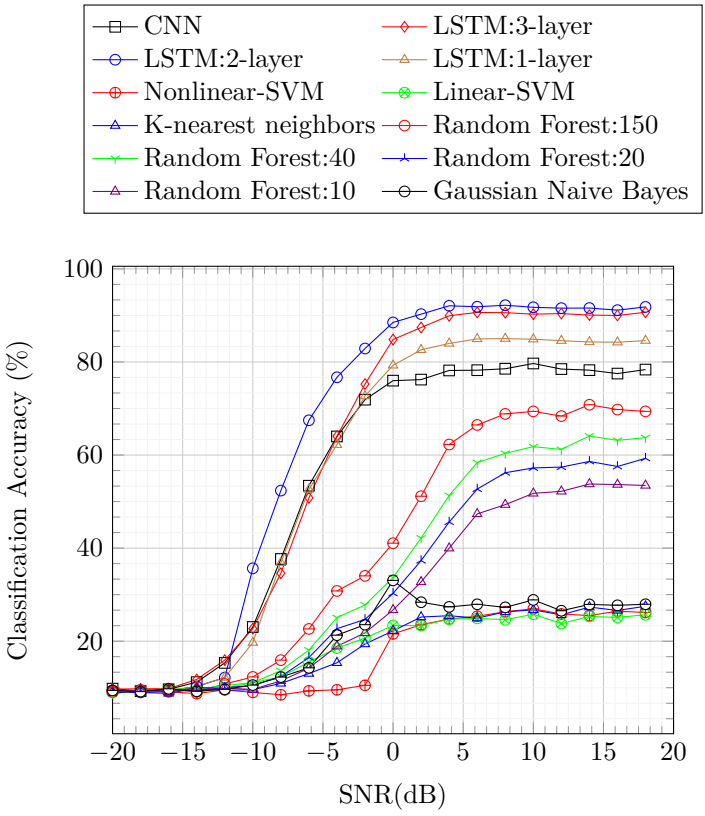


Figure 4.4: Classification accuracy comparison of hyper-parameter optimized 2-layer amplitude-phase LSTM model with others on RadioML dataset.

results on the low SNR regions after various attempts, which may be because of the difference in hyper-parameter tuning. Though, the high SNR results of the baseline model matches with that of the reported ones in the chapter. Detailed discussions on the effect of layer depth and number of LSTM cells are presented in Section 4.5.4.

Classification performance of other standard machine learning models such as Support Vector Machines (SVM), random forest, k-nearest neighbors and Gaussian Naive Bayes are also summarized in Figure 4.4. All models are fed with the same amplitude-phase training and test data for this comparison. Random forest with 150 decision trees is able to provide close to 70% of accuracy at very high SNR conditions while others could reach only around 26%. It could

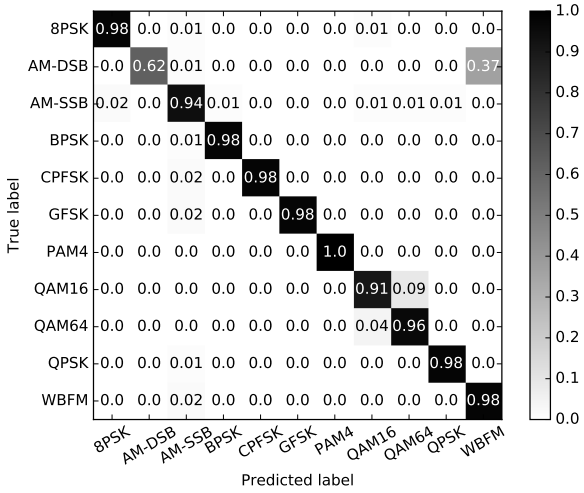


Figure 4.5: Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at 18dB SNR

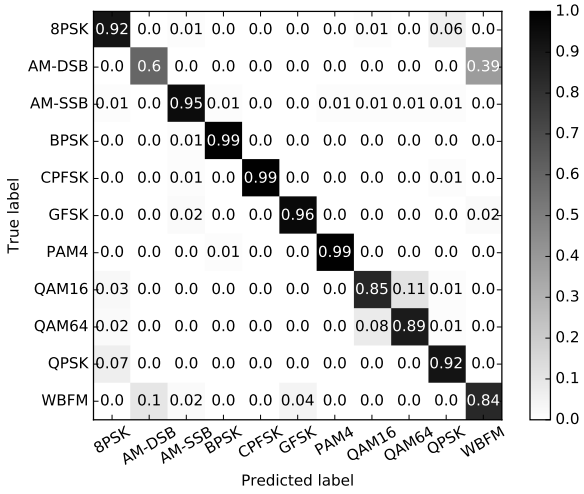


Figure 4.6: Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at 0dB SNR

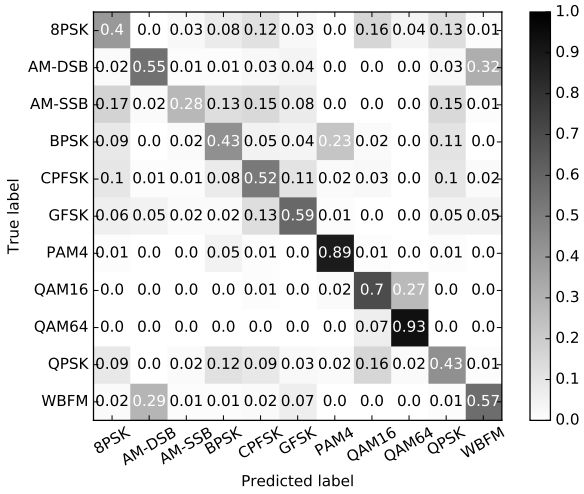


Figure 4.7: Confusion matrix for 2-layer amplitude-phase LSTM model on RadioML dataset at -8dB SNR

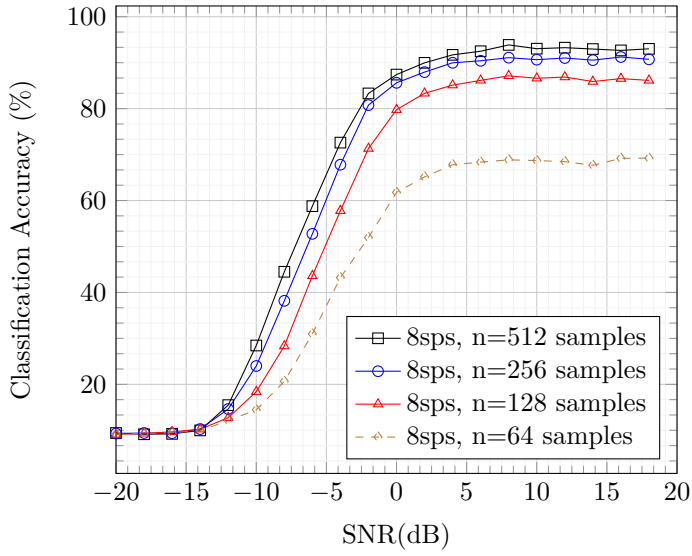


Figure 4.8: Classification accuracy of amplitude-phase LSTM model on modified RadioML dataset for 8 samples/symbol. The model is trained only on input sample lengths (n) from 128 to 512. The model gives close to 70% accuracy on 64 input samples (dashed line) which it is not trained for.

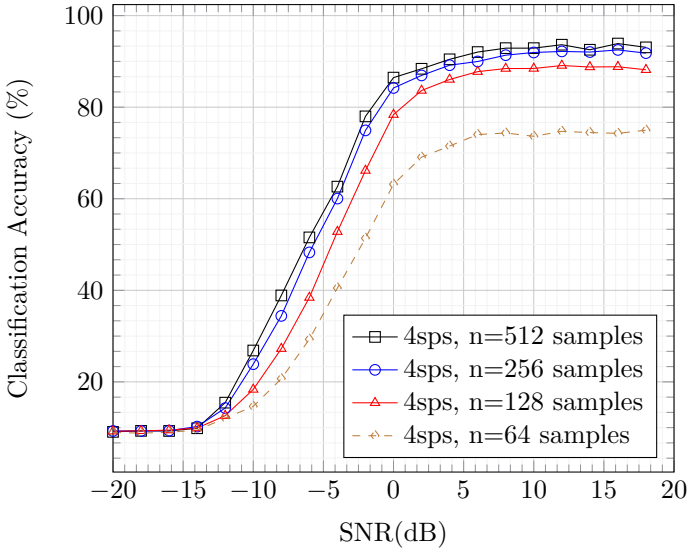


Figure 4.9: Classification accuracy of amplitude-phase LSTM model on modified RadioML dataset for 4 samples/symbol. The model is trained only on input sample lengths (n) from 128 to 512. The model gives close to 75% accuracy on 64 input samples (dashed line) which it is not trained for.

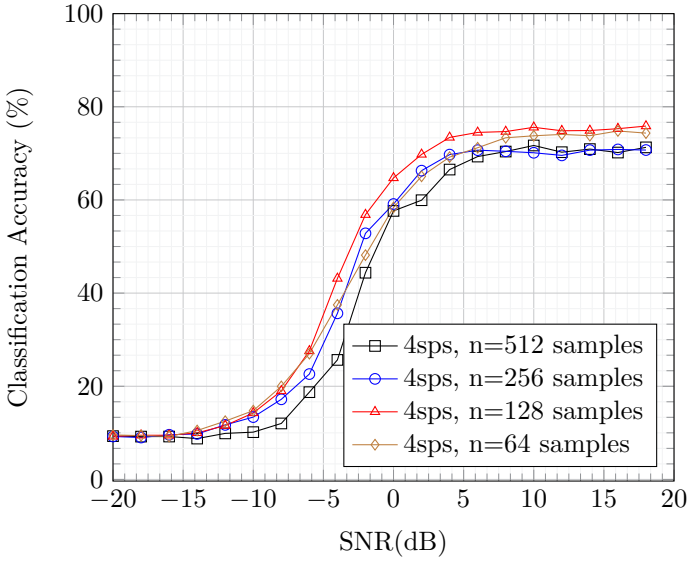


Figure 4.10: Classification accuracy of amplitude-phase LSTM model for non-trained data lengths on modified RadioML dataset with 4 samples/symbol.

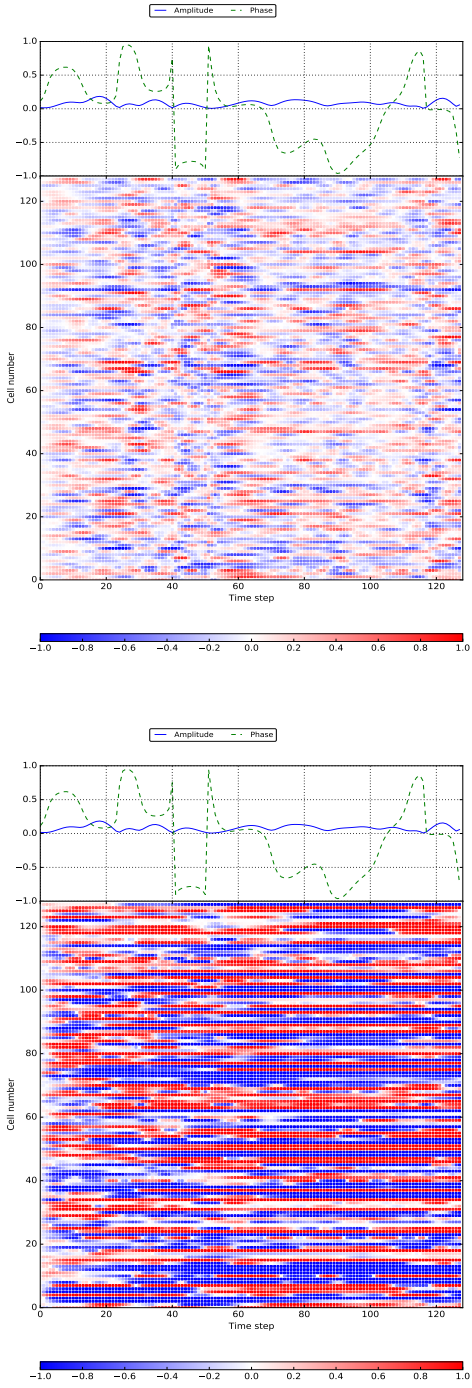


Figure 4.11: Layer-1 (top) and layer-2 (bottom) LSTM temporal activations for a QAM64 input vector. On the top the amplitude and phase of the input signal (y-axis) is plotted at each time step (x-axis). Below the amplitude and phase of the signal, temporal activations for all cells in the LSTM model for each time step are shown. Blue denotes *tanh(c)* activations of value -1 and red denotes a value of +1.

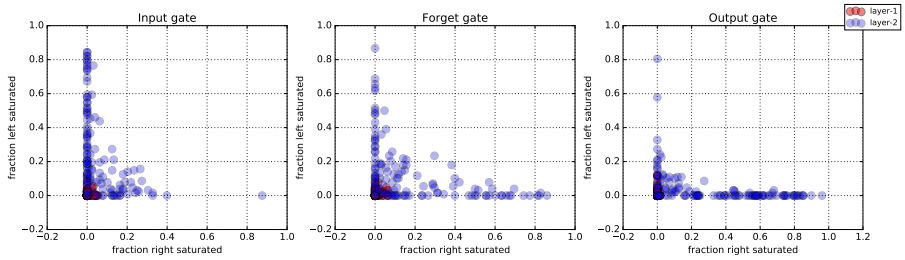


Figure 4.12: Gate saturation plots for LSTM for the same a QAM64 input vector. A circle represents a gate in a particular LSTM cell.

be clearly noticed that the deep learning models perform superior to the other standard techniques when fed with the raw sensed data. The deep learning models can classify signals very efficiently with a very low number of symbols, usually with hundreds of samples (tens of modulated symbols) when compared to the classical cyclostationary based expert feature models which requires samples in thousands range (hundreds of modulated symbols) for averaging. Similarly extracting expert cyclostationary features using tens of symbols is very sub-optimal, which substantiate the use of deep learning models.

To understand the results better confusion matrices for the two layer LSTM model for various SNRs are also included. It can be seen in Figure 4.5 that at a high SNR of 18dB the diagonal is much more sharp even though there are difficulties in separating AM-DSB and WBFM signals. This is mainly due to the silence periods of audio as the modulated signals are generated from real audio streams. Similarly in Figure 4.6, at 0dB SNR it is noticed that there is some level of confusion regarding QAM16 and QAM64 as the former is a subset of the the latter. The confusion increases further at low SNRs as shown in Figure 4.7. From these basic analysis it is clear that deep complex structures as mentioned in [55] are not required to achieved good SoA classification accuracy at high SNRs. However, use of convolutional layers might turn useful at low SNRs as reported in [55]. In our experiments we also noticed that simply providing IQ samples to the LSTM model yielded poor results while normalized amplitude and phase interpretation provided good results. The models even failed to reduce the training loss when fed with time domain IQ samples, giving a constant accuracy of 9% on the radioML dataset, as the LSTMs were not able to extract any meaningful representations. Similarly feeding amplitude-phase information to the CNN model did not provide any accuracy improvements over the IQ-CNN model. The classification accuracy improvement is achieved from the combined benefits of using amplitude-phase information along with 2-layer LSTM model.

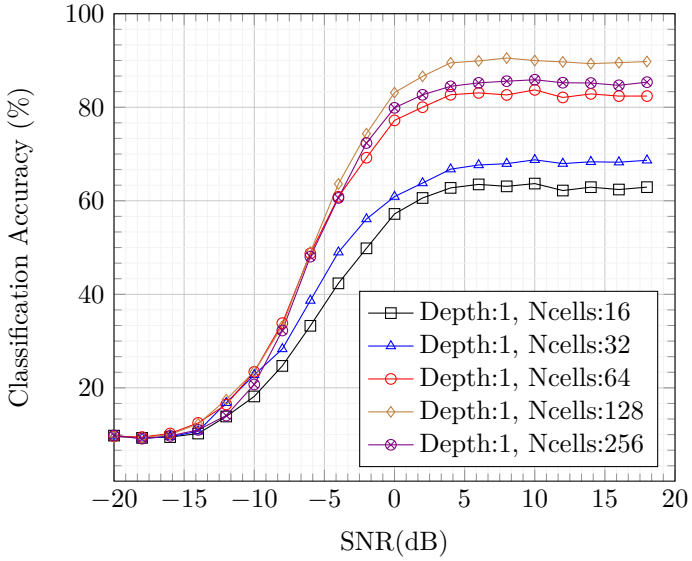


Figure 4.13: Classification accuracy of single layer amplitude-phase LSTM model for different cell size on RadioML dataset.

4.5.2 Classification accuracy on modified RadioML dataset

The same two layer LSTM model is trained on SNRs ranging from -20dB to 20dB and input sample lengths from 128 to 512 samples. The accuracy of the model is tested on the full range of SNRs and also on input sample length that is smaller than the training set (e.g, 64). It is evident from the results in Figures 4.8 and 4.9 that the classification accuracy improves as the model sees more modulated symbols. Even though the model is trained on varying data lengths from 128 to 512 samples, it gives an average accuracy of 75% with 64 samples and 4 samples per symbol scenario for which it was not trained, which confirms the model's generalization capabilities. To further analyze the generalization capabilities of the model on unseen sample lengths, four balanced folds of data each containing sequences with sample lengths of 64, 128, 256 and 512 are created. The model is then trained only on three folds, and the left-out fold is used to test generalization to the unseen length. This process is repeated for all four sample lengths and the results are presented in Figure 4.10. The model consistently gives an average accuracy above 70% for high SNR conditions.

4.5.3 Learned representations

The inherent non-linearity and deep structures makes understanding the representations learned by LSTMs difficult. In order to obtain some good insights we use visualization techniques similar to the ones presented in [83]. These visualizations can help to understand how LSTM cells behave for an input signal, for instance which cells gets activated at each time step and how long each gate remains open. Figures 4.11 and 4.12 presents the gate activation and saturation of the trained two layer LSTM model for a QAM64 input signal with 18dB SNR. As explained in Section 4.4.1 the gates of LSTM cells have sigmoid activation functions, giving an output value between 0 and 1. A gate is said to be left saturated if its activation is less than 0.1 and right saturated if the activation is greater than 0.9. The fraction of time for which the gate is in left or right saturated mode in the entire 128 samples time is plotted in Figure 4.12. On the first layer, it can be noticed that all the three gates are confined close to the origin showing that they are not highly left or right saturated. The absence of right saturation in the first layer forget gates, confirms that the cells do not store information for long term. There are no cells in the first layer that function in purely feed-forward fashion, since their forget gates would show up as consistently left-saturated. The output gate plots in the first layer also show that there are no cells that are revealed or blocked to the hidden state. This is also visible in the activation plots of the first layer in Figure 4.11. The activations are short when compared to the second layer and it can be noticed in Figure 4.11 that many cell activations follow the input amplitude and phase changes in the input waveform. The second layer stores much long term dependencies from the fine grained representations generated from the first layer.

4.5.4 Effect of cell size and layer depth

A comprehensive study is also performed to understand the effect of the number cells and layer depth on the model performance. The number of LSTM cells and layer depth are varied from 16 to 256 and 1 to 3 respectively. The models are trained on RadioML dataset on all SNRs. The accuracy levels for various layer depths are presented in Figures 4.13, 4.14 and 4.15. An initial analysis clearly shows that the model accuracy increases with increasing layer depth for mostly all cell sizes. It can be also noted that as the depth of the model increases, increasing the number of cells doesn't give much performance improvements. For instance, at depths 2 and 3 increasing the cell numbers from 128 to 256 doesn't provide any performance improvements.

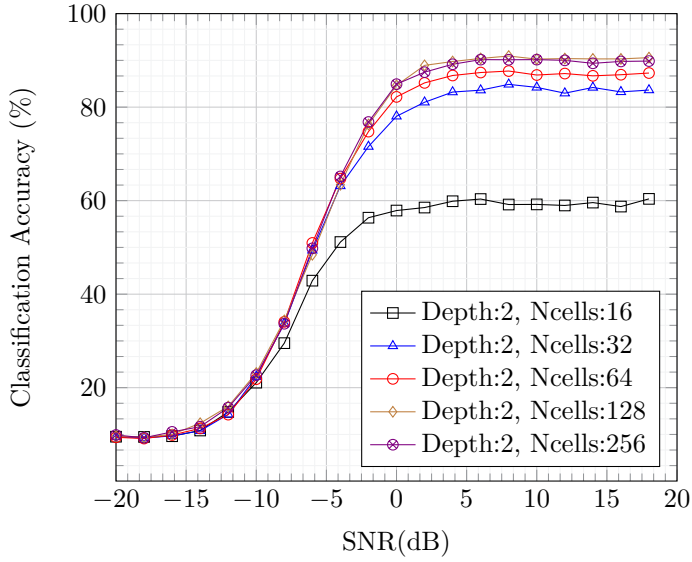


Figure 4.14: Classification accuracy of two layer amplitude-phase LSTM model for different cell sizes on RadioML dataset.

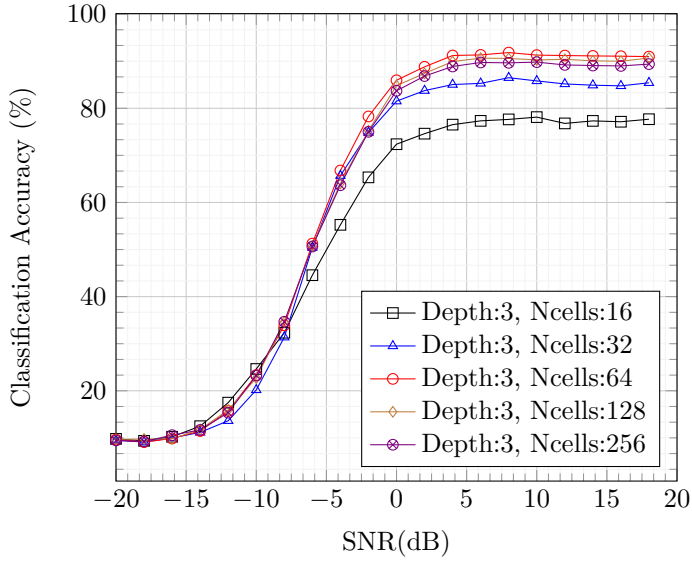


Figure 4.15: Classification accuracy of three layer amplitude-phase LSTM model for different cell sizes on RadioML dataset.

4.6 Semi-supervised Models

A few papers in the literature have looked into wireless signal classification in a semi-supervised manner. In [84], the authors perform a supervised bootstrapping of sparse representation, that is training a deep learning model in a supervised fashion with known labels, and then proceed with the idea that by discarding the last softmax-layer of the fully supervised model, the features generated have the ability to separate additional classes. The model is shown to be able to extract abstract features which can be used to separate unseen modulations with simple clustering techniques. However, the model makes use of non-labeled data only for clustering and not for training. This prevents the model from providing guaranteed clustering for all unseen modulations with drastic parameter variations.

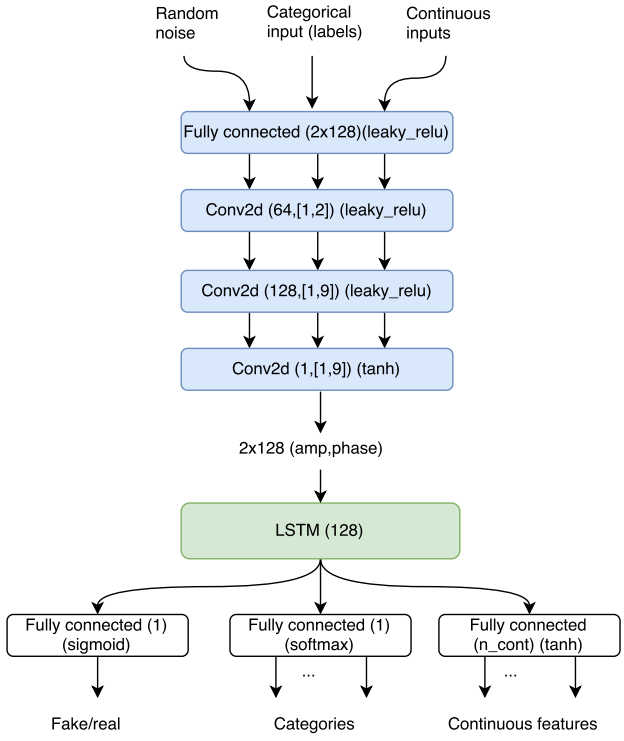


Figure 4.16: Generator and discriminator details.

Another work [85] makes use of sparse coding and dictionary learning to perform blind signal classification. Sparse coding is an unsupervised method to learn

a dictionary of over-complete basis vectors that can represent data efficiently. Each basis vector in the dictionary is also known as an atom. During the training stage a few dictionary atoms are generated in a semi-supervised fashion by using well-known pulse shaping filters and similarity matching [86] is used for finding the best suited modulation. In a realistic scenario selecting parameters for generating supervised dictionary is difficult as filter shape is not the only parameter that can vary. Also the model is tested only on a AWGN channel and proper analysis is not done with fading and other hardware effects on the dictionary generation. To address the shortcomings, very generic models which can use both labeled and unlabeled samples and can learn relevant features are required for proper classification. In addition, similar to the fully supervised deep learning models, they should do automated feature extraction avoiding the manual expert feature extraction requirements.

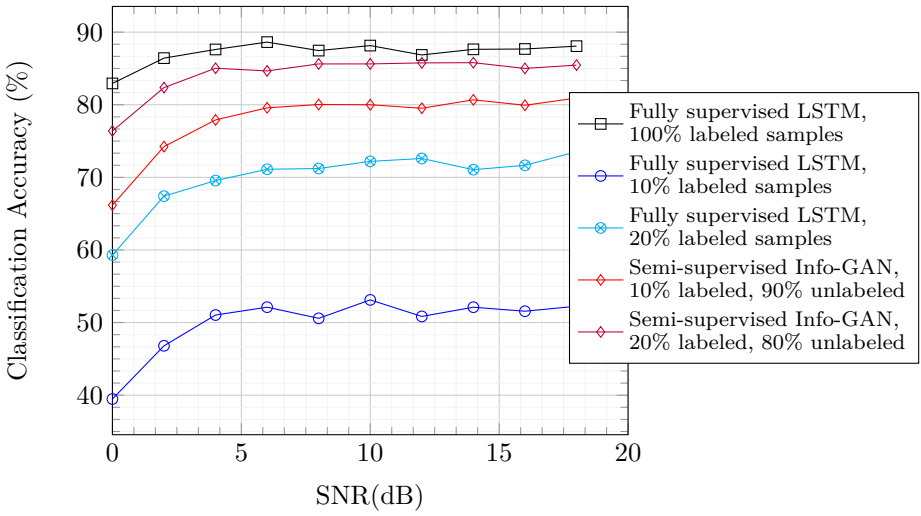


Figure 4.17: Classification results for a fading channel with receiver effects for ['BPSK', 'QPSK', 'QAM16', 'QAM64', 'PAM4', '8PSK', 'CPFSK', 'GFSK'] from RadioML dataset.

To enable the specified requirements, a mutual information maximizing generative adversarial network with layers specific to wireless domain is selected [87]. This model is an advanced version of the Information Maximizing Generative Adversarial Networks (InfoGAN) [64] model with a semi-supervised setting. Used layers in the generator and discriminator are detailed in Figure 4.16, where convolutional layers are used in the generator and LSTM in the discriminator. Being a generative model, the generator makes signals which resemble the actual input data while the discriminator tries to segregate them. In addition, a softmax layer is added to the discriminator to enable signal

classification along with segregation. A discriminator which is trained on such generated and real inputs is seen to be much robust as the amount of signal variations the model sees is much larger. In-depth mathematical details about the model can be found in [87] and are omitted here in order to avoid repetition. As shown in Figure 4.17, the trained discriminator on instantaneous amplitude and phase of the received signal is able to achieve an average accuracy of 80% only by using 10% labeled samples, which is only 10% lower than the accuracy of the fully supervised discriminator. On the other hand, a fully supervised model with the same LSTM layers could achieve only 50% of accuracy using 10% of labeled samples. The discriminator also reaches accuracy close to the fully supervised model by only using 20% labeled samples.

4.7 Resource Friendly Models

As mentioned earlier in the introduction, it is quite difficult to deploy SoA AMC algorithms on low-end distributed sensors such as the ones in Electrosense. We extend our study in two directions to reduce the resource requirements in terms of data transfer rate to the cloud, data storage and computational power. First, a study is conducted to understand to what extent technology classification based on AMC can be done using averaged magnitude FFT data, as the PSD pipeline being the default enabled one in the Electrosense sensors with medium data transfer costs. As the sensors are sequentially scanning the spectrum, they are capable of generating magnitude spectrum information for wideband signals which is an added advantage. Second, the performance of quantized versions of the proposed deep learning models are analyzed which can reduce the computational cost of the models enabling deployment of these models on the sensors itself. The averaged magnitude FFT signal sent by the sensor, the selected dataset for testing the model, averaged magnitude FFT classification model, other quantized models and the classification results are detailed in the following subsections.

4.7.1 Received averaged magnitude FFT signal

Electrosense sensors scan the wireless spectrum sequentially. The sensor samples the wireless spectrum at a fixed sampling rate $N = 2.4\ MS/s$ tuned to a particular centre frequency f_x. As the sensor's sampling rate is limited, a wideband signal's magnitude spectrum can be received only by sequential scans

to cover the entire bandwidth as given in equation 4.8.

$$
\begin{aligned}
R(f) = \frac{1}{M}\sum_{m=0}^{M} |FFT_m(e^{-j2\pi f_1(t_0+mD_t)}s(t_0+mD_t)* \\
c(t_0+mD_t)+n(t_0+mD_t))| \\
||\frac{1}{M}\sum_{m=0}^{M} |FFT_m(e^{-j2\pi f_2(t_1+mD_t)}s(t_1+mD_t)* \\
c(t_1+mD_t)+n(t_1+mD_t))| \\
||\dots
\end{aligned}
\tag{4.8}
$$

In equation 4.8 $||$ represents the concatenation operation where the full bandwidth of the signal of interest is captured by a sequentially scanning sensor sampling at a lower sampling rate, similar to the Electrosense dataset mentioned in the following subsection. The averaged magnitude FFT signal at centre frequencies f_i, where $f_i \in (50\,MHz, 6\,GHz)$ based on the sensor sampling rate and frequency range, are sent to the cloud where they are concatenated together. In equation 4.8, M represents the magnitude-FFT averaging factor and t_x the sequential sampling time. For instance, $t_n = t_{n-1} + T$, where $T = MD_t$ is the amount of time spent at a particular centre frequency and D_t being the time for collecting *fft_size* samples for a single FFT input.

4.7.2 Electrosense dataset

Six commercially deployed technologies are selected to validate the classification accuracy using averaged magnitude FFT data as given in Table 4.3. Over-the-air data from multiple Electrosense sensors are retrieved through the Electrosense API[4] with a spectral resolution of 10 kHz and time resolution of 60 seconds. The data is collected from sensors with omni-directional antennas which are deployed indoors. The sensors follow sequential scanning of the spectrum with an FFT size set to 256 giving a frequency resolution close to 10 kHz. With a FFT size of 256 and sensor ADC bit-width of 8, we get an effective bitwidth of 12 resulting in a theoretical dynamic range of 74dB. Practical dynamic range depends on the ADC frontend stages and the noise level, which may vary between 60 to 65dB. Five FFT vectors are averaged for reducing the thermal noise of the receiver. Some of the selected technologies such as LTE and DVB

[4]https://electrosense.org/open-api-spec.html

Technology	WFM, TETRA, DVB, RADAR, LTE, GSM
Time resolution	60s
Frequency resolution	10 kHz
Sensor sampling rate	2.4 MS/s
Scanning strategy	Sequential
FFT averaging factor	5
Sensor location and antenna type	Indoor, Omni-directional
Number of training samples	3100 vectors
Number of test samples	3100 vectors

Table 4.3: Averaged magnitude FFT dataset parameters.

have an effective bandwidth which is higher than the sampling bandwidth of the of the low-end SDR. As the sensor is sequentially scanning, full spectrum shapes of these wideband signals are obtained by combining FFT outputs of these sequential scans. The entire data is split into two, one half for training and the other half for testing the model.

4.7.3 Averaged Magnitude FFT model

Sequentially sensed frequency spectrum data from the sensors contain signals of different bandwidth. The model should be able to process this variable length data and classify them to proper groups. We use the same LSTM model used for classifying complex input data as shown in Figure 4.3. The averaged magnitude FFT signal is fed to the model as a sequence as presented in Figure 4.3. The same LSTM model is chosen as it can handle variable length input signals and is also good at learning long term dependencies. The final output of the LSTM model after feeding n frequency bins is given as input to the softmax layer through a fully connected linear layer. The softmax layer outputs the probability $P(y = l|a; \theta)$ for $l \in \{0, 1, .., 5\}$ where a denotes the input averaged FFT bins, θ the model parameters and l the enumerated label for one of the six technologies as listed in Table 4.3.

4.7.4 Classification results

An initial study is conducted to understand the technology classification accuracy of the averaged magnitude FFT model when compared to full IQ information. On the Electrosense dataset (Section 4.7.2) the proposed model achieves a

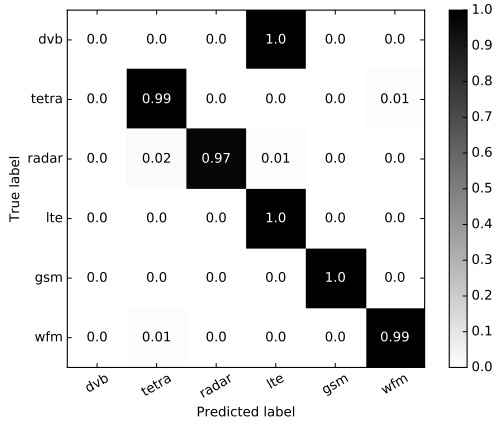


Figure 4.18: Confusion matrix for technology classification on Electrosense dataset.

Layer-depth	Number of cells			
	16	32	64	128
1	70.02%	74.39%	79.38%	81.05%
2	72.69%	75.93%	80.92%	81.68%

Table 4.4: Classification accuracy on Electrosense dataset for varying layer depths and cell numbers.

classification accuracy of 80%. The confusion matrix for the same is shown in Figure 4.18. From the confusion matrix it is clear that there is a large confusion between LTE and DVB. This is expected as the power spectra of both DVB and LTE looks very similar as both of them are based on OFDM. As multiple technologies might share the same modulation types, the assumption that a modulation classifier can be used for technology classification is not always valid with the current deployed technologies. We also investigated the effect of number of LSTM cells and layer depth on this dataset whose results are summarized in Table 4.4. Increasing the layer depth did not contribute significantly to the classification accuracy as there might be no more low level abstract features to be learned from the magnitude spectrum information. Furthermore, there are a large number of modulation schemes which exhibit the same power spectral density making averaged spectrum a sub-optimal feature for classification. For instance, the power spectral densities of different modulations schemes such as 8PSK, QAM16 and QPSK are identical once passed through a pulse shaping filter with a fixed roll-off factor. This can be theoretically shown and easily

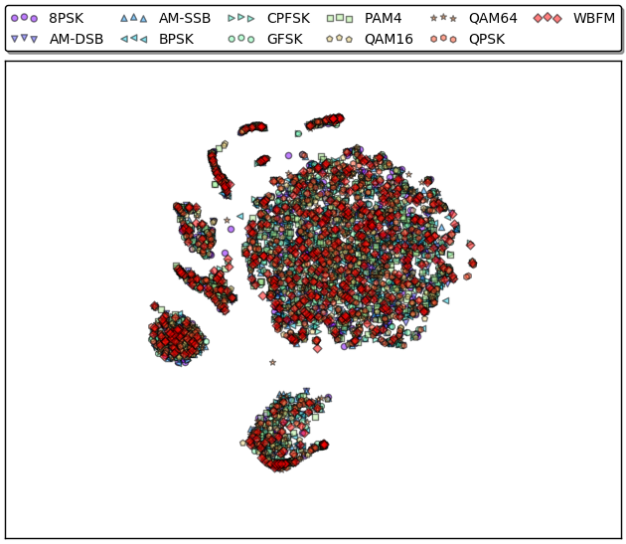


Figure 4.19: t-SNE plot for magnitude FFT output on radioml dataset.

verified with manual inspection [88].

To further validate the argument, magnitude-FFT is calculated on the same RadioML dataset which was used for testing the performance of the amplitude-phase model. As the RadioML dataset consists of modulations with same bandwidths passed through the same pulse shaping filter, their magnitude-FFTs looks identical giving very low classification accuracy of only 19% for all 11 modulations even at high SNRs. To get a better understanding of the generated magnitude-FFT dataset, a visualization of a subset of the data in two dimensions is provided. For reducing the dimensionality of the data to 2 and for the ease of plotting, the t-SNE technique [89] is used. A small subset of the radioML dataset of 5000 vectors, containing 128 point magnitude FFT of all generated modulation schemes with varying SNR ranges from +20dB to +20dB are fed to the t-SNE algorithm. t-SNE is useful for a preliminary analysis to check whether classes are separable in some linear or nonlinear representation. The representation generated by t-SNE on the data subset is presented in Figure 4.19. It can be seen that the representation overlap is very high and t-SNE could not generate any meaningful clustering as the phase information is completely lost when computing magnitude-FFT, leaving identical magnitude spectrum for many modulation schemes. The obvious solution is to switch to IQ pipeline and deploy optimized versions of complex input signal models on the sensors itself, thus reducing the uplink data transfer rate. This is further

investigated in the following subsection.

4.7.5 Quantized models

Deep learning models are processor intensive which makes them quite difficult to be deployed on low-end sensor networks. As mentioned in the introduction, transferring pure IQ data to the backend server is not a scalable solution. In addition, our results indicate that some signals require IQ information for correct classification. To enable low-end sensor deployment, a feasibility study is conducted by quantizing the weights and activations of the newly proposed as well as baseline neural network models. Binarized networks can exceptionally reduce the required memory size for holding intermediate results and replace most of the arithmetic operations with bitwise operations [67]. For instance, when compared to a full precision network with 32 bits weights and activation, a binarized network only needs 32 times smaller memory resulting in a reduced required memory size and memory access cost. In [67] the authors had already noticed that binarizing LSTMs results in very poor accuracy. We confirm the same observation with our LSTM models too. However, models with binarized CNNs have been reported to provide accuracy close to their full precision variants. To validate this, the performance of the binarized baseline CNN model is also investigated on the radioML modulation dataset. Furthermore, by allowing more quantization levels on the LSTM models a higher accuracy can be achieved while still reducing the computational cost. Two quantized LSTM model variants are tested, one with ternary weights (-1, 0, +1) and full precision activation (TW_FA) and the other with ternary weights and four bits activation (TW_4BA). The accuracy results of these models are summarized in Figure 4.20. Results show that LSTM models with ternary weights and 4bit activation can provide close to 80% accuracy reducing the very high computational power required for full precision models. Binary CNN models also provided an accuracy level 10% below the full precision variants. We believe the classification accuracy can be further improved by proper hyper-parameter tuning and longer training.

The theoretical memory requirements for the trained weights along with number of multiplications required for the entire model, excluding activations, are summarized in Table 4.5. A binarized neural network can drastically reduce the processing power requirements of the model. For instance, in a binarized network all weights and activation are either -1 or +1, replacing all multiply operations by XNORs. The multiply-accumulate, which is the core operation in neural networks, can be replaced by 1-bit XNOR-count operation [67]. Convolutions also comprises of multiply and accumulate operations which can also be replaced by its binarized variants. Thus the baseline CNN model can provide very good performance improvements on the general purpose ARM based Electrosense

Model	Weights count	MUL/ XNOR count	Memory
CNN: full precision	5451096	5451096	174.44 MB
CNN: binary	5451096	5451096	5.45 MB
LSTM: full precision	200075	262144	6.4 MB
LSTM: TW_4BA	200075	262144	400.15 KB
LSTM: TW_FA	200075	262144	400.15 KB

Table 4.5: Memory size and number of multiplications for the discussed models (excluding nonlinear activations).

Platform	Model			
	CNN-FP	CNN-8b	LSTM-FP	LSTM-8b
Nvidia GeForce GTX 1060	15239	85	12874	76
Nvidia Tegra TX2	2342	205	2785	127
Intel i7-3770	1039	85	1035	81
Jetson board (ARMv8)	264	217	275	164
Raspberry pi-2 (ARMv7)	36	30	24	21

Table 4.6: Inference engine performance in number of classification per second on different platforms.

sensors. For the CNN models the convolutional layer output numbers are high, as we are not using any pooling layers, which accounts for the larger memory size in the succeeding dense layers. We would like to emphasize the fact that the given memory sizes are for the entire model and the weights that should be hold in the memory might vary based on practical implementations.

As binarized LSTM models did not provide good accuracy, we are forced to use 4-bit quantized variants of the same. Even though the performance improvements are not that extreme similar to binarized models, quantized LSTMs can also reduce the resource consumption. First of all, as no large dynamic range is required all the 4-bit multiply-accumulate operations can be implemented in fixed point arithmetic, which is much more faster in ARM CPUs when compared to their floating point versions. Secondly, routines can also be implemented to reduce the space requirements to hold intermediate results and the activations can be implemented as look-up tables. We would also like to emphasize the fact that on a special purpose hardware, such as FPGAs, quantized models can obviously reduce the space used and power-consumption as the multiply-accumulate units have smaller bit-widths.

Most of the machine learning frameworks such as Tensorflow have started

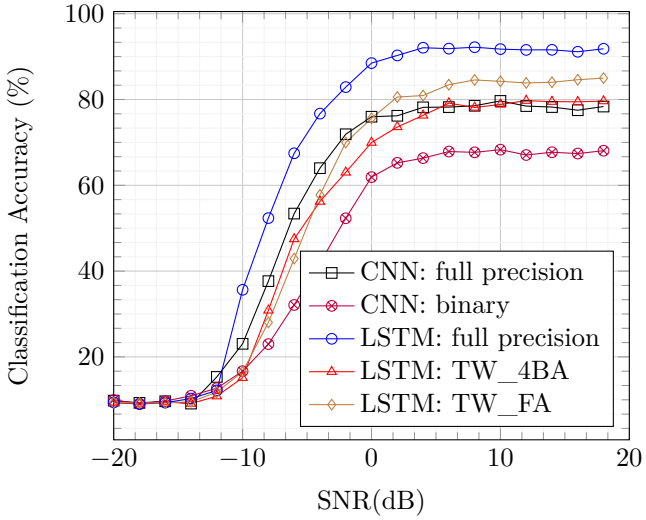


Figure 4.20: Classification accuracy of two layer quantized models on RadioML dataset.

supporting quantized models for low-end processor deployment. The quantized kernels are under active development at the time of writing this chapter which currently only supports a minimum quantization of 8 bit. Quantized kernels for all operations in all platforms are not available in these libraries resulting in low performance than expected. The full IQ information model classification performances on various platforms such as Nvidia GPUs, Intel and ARM processors for full precision and 8bit precision models are summarized in Table 4.6. To avoid implementation mismatches across various models and platforms all the comparisons are done using quantization tools provided by Google's Tensorflow which is currently under active development. These tools allow to freeze and compress a trained model to a single file and then test it on various platforms easily. These values in the table are performance indicators in number of classifications per second for 128 sample length vectors. It can be noticed that the quantized models perform very bad on GPUs and Intel PCs due to lack of support. The quantized models currently provide performance very close to floating point variants on ARM processors. Quantized kernel support is improving for ARM processors due to increased demand for deploying these models on mobile devices. The aforementioned advantages of quantized models is expected to be available in the near future through these standard libraries.

4.8 Conclusion and Future work

In this chapter we proposed a new LSTM based model which can classify signals with time domain amplitude and phase as input. SoA results on high SNRs (0 to 20dB) is achieved without using complex CNN-LSTM models as mentioned in [55]. Being a recurrent model we showed that the model can handle variable length inputs thus can capture sample rate variations effectively.

Though neural networks are good at function approximation, our experiments emphasize the fact that *data preprocessing* and *proper representation* are equally important. This claim is substantiated from our experiments with the LSTM model where the model gave poor results when fed with time domain IQ samples while it gave accuracies close to 90% for high SNRs when provided with time domain amplitude and phase information. As shown by various SoA models in speech and image domains, performance improvements are seen with increasing layer depth which saturates after a few levels. In addition, we showed that basic technology classification is achievable by only using averaged magnitude FFT information over a distributed set of sensors complying with the uplink bandwidth resource constraints. Furthermore, experiments showed that quantized LSTM models can achieve good classification results thus reducing the processing power requirements at the cost of 10% accuracy loss. This allows the deployment of these models on low cost sensors networks such as Electrosense enabling a wide area deployment. It is also remarkable that these deep learning models can classify signals with a fewer number of samples when compared to the expert feature variants, such as cyclic frequency estimators, enabling faster classification. Furthermore, deep learning allows for incremental learning, thus it would not be required to retrain the entire network from scratch for the new wireless non-idealities like antenna patterns and sensitivity. In addition, dedicated hardware is gaining popularity to reduce the deep learning model's energy and memory footprints which demands quantized versions of the models.

Although the LSTM models perform very well at high SNR conditions, CNN models seems to provide an additional 5-10% accuracy on the low SNR conditions (SNRs below -2dB) as shown in [55]. Even though we are not able to replicate the results in [55] (because of hyperparameter tuning), it is reasonable to conclude that the learned filters in CNN for a fixed sample rate might give performance improvements for low SNR values. Furthermore, all the implemented code for the proposed models are made publicly available for reproducing and verifying the results presented in this chapter and for supporting future research.

Low SNR performance of these SoA deep learning models could be further improved with the help of efficient blind denoising models. Models which can perform automated channel equalization and compensate receiver imperfections

such as frequency offset can further improve the classification performance. The current radio deep learning models make use of layers which basically applies non-linearity after simple multiply-accumulate-add operations while it is well established in the research community that cyclic cumulants, which are generated by time-shifted multiplication and averaging of the input itself, performs well in the expert feature space. Deep learning models which can extract features similar to cyclic cumulants might improve the performance metrics.

The analysis would not be complete without emphasizing the limitations of the SoA deep learning models. First, the current complex models are tested on a dataset with normalized bandwidth parameters. Real life transmitted signals generally have varying symbol rates and bandwidths. Even though the variable length LSTM model is shown to be capable of adapting to these scenarios, further analysis is required to validate the claim. In future, models that can handle all possible spread spectrum modulations should be also tested. Second, the generalization capabilities of these models should be further investigated, in terms of performance of these models in unknown channel conditions and modulation parameters. Finally, most of the successful SoA models are supervised models which requires labeled training data. Labeling is a very tedious task which projects the importance of semi-supervised models for classification tasks. Semi-supervised model proposed in this chapter and other published studies [90] on semi-supervised machine learning models for cellular network resource management validates the need for more semi-supervised models, which is also an active direction for future research. The benefits of crowdsourced semi-supervised anomaly detection is further analyzed in the next chapter. We believe deep learning models adapted to radio domain can help in understanding, analyzing and decision making in future complex radio environments.

Chapter 5

Crowdsourced anomaly detection

Automated wireless spectrum monitoring across frequency, time and space will be essential for many future applications. Manual and fine-grained spectrum analysis is becoming impossible because of the large number of measurement locations and complexity of the spectrum use landscape. Detecting unexpected behaviors in the wireless spectrum from the collected data is a crucial part of this automated monitoring, and the control of detected anomalies is a key functionality to enable interaction between the automated system and the end user. In this chapter we look into the wireless spectrum anomaly detection problem for crowdsourced sensors. We first analyze in detail the nature of these anomalies and design effective algorithms to bring the higher dimensional input data to a common feature space across sensors. Anomalies can then be detected as outliers in this feature space. In addition, we investigate the importance of user feedback in the anomaly detection process to improve the performance of unsupervised anomaly detection. Furthermore, schemes for generalizing user feedback across sensors are also developed to close the anomaly detection loop.

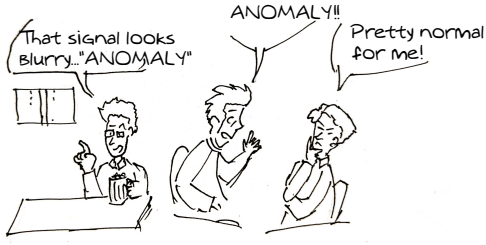


Figure 5.1: Resolving user anomaly preferences and combating incorrect anomaly labeling is challenging in a semi-supervised anomaly detection setup.

This chapter is adapted from the journal under review: Rajendran, Sreeraj; Lenders, Vincent; Meert, Wannes; Pollin, Sofie; 2019. *Crowdsourced wireless spectrum anomaly detection.* IEEE Transactions on Cognitive Communications and Networking; 2019.

5.1 Introduction

Radio spectrum is one of our most precious and widely used natural resources. With the advent of new wireless communication technologies, spectrum usage has become very complex resulting in airwave congestion and other interference issues [1,91]. Diverse spectrum regulations across countries have also contributed to this chaotic spectrum usage when non-standardized wireless devices cross country borders. In addition, easily available illegal wireless jammers or low-cost SDR devices which are capable of generating custom wireless signals are making the problem worse. Unintentional and intentional jamming of localization services such as GNSS, which is used for a wide range of applications such as automated vehicle navigation, airplane landing procedures and maritime vessel tracking, is increasing with the availability of easy jamming devices [4,5,8]. Furthermore, illegal repeaters, which are used to boost mobile coverage, can adversely affect the cell planning of mobile operators resulting in poor coverage and dropouts [3]. On the other hand, densified small cells are becoming fundamental for the new high throughput and low latency networks [2]. Automated analysis and detection of anomalous behaviors in the spectrum is becoming crucial to promote the growth of new generation wireless systems.

The wide range of wireless spectrum anomalies makes it infeasible to collect and label all types of anomalies and formulate anomaly detection as a supervised learning problem. A majority of the algorithms in literature [27,28,39,58–60] formulate wireless spectrum anomaly detection as an unsupervised learning problem where the models try to learn the normal spectrum data distributions and detect the uncommon patterns as anomalous. These models make the basic assumption that non-frequent behaviour is anomalous which is not always true. For example, transmissions in ISM bands exist that are very sparse but are not anomalies. Similarly, pirate transmitters or transmission duty cycle limit violations in a unlicensed band are anomalous even when they are very frequent. In this chapter we solve wireless spectrum anomaly detection by formulating it as a crowdsourced active learning problem. More specifically we provide solutions to the following questions addressing various challenges posed by them:

1. How to optimize anomaly detection based on user feedback from different users in a crowdsourced network?
2. How to improve the anomaly detection of a particular user by using expert feedback from other users which may not have fully overlapping interests?
3. How to generalize user feedback across different sensors?

Some of the major challenges that need to be addressed to answer the aforementioned questions are given below.

5.1.1 Challenge 1: Finding an interpretable feature space

Contrary to other sensing contexts such as temperature and air pollution monitoring, the amount of data generated by wireless spectrum sensors is quite huge. In addition to the huge amount of data, most of the interesting techniques such as classification, prediction and anomaly detection suffer from the curse of dimensionality when addressed in the higher dimensional input data space. Furthermore, the data costs related to the transfer and storage of the spectrum sensing sensor data is huge. For instance, the crowdsourced spectrum sensing framework, Electrosense [38], enables three pipelines with very low, medium and high data transfer costs namely: Feature, PSD and IQ pipeline respectively. For fulfilling the long term spectrum monitoring dream, it is essential to extract critical features and employ the feature pipeline whenever possible. First, these features should also be compressed and interpretable for further analyzing the cause for the anomaly. Second, the compressed selected feature space should promote outlier detection. Finally, the selected feature space should be the same across sensors for unifying the anomaly detection process for different users.

Our contribution: We introduce novel data driven feature extraction modules for both normal and anomalous wireless spectrum input data. The proposed training process helps to generate a common feature space across sensors without compromising the model's anomaly detection capabilities of individual sensors.

5.1.2 Challenge 2: Detecting meaningful anomalies

Anomalies can be outliers in the normal feature space or process variations. Anomalies can also be some frequent events hidden inside the feature space, such as transmitter duty cycle variations, which prevents us from putting a simple outlier threshold in the feature space. It is also challenging to separate anomalies

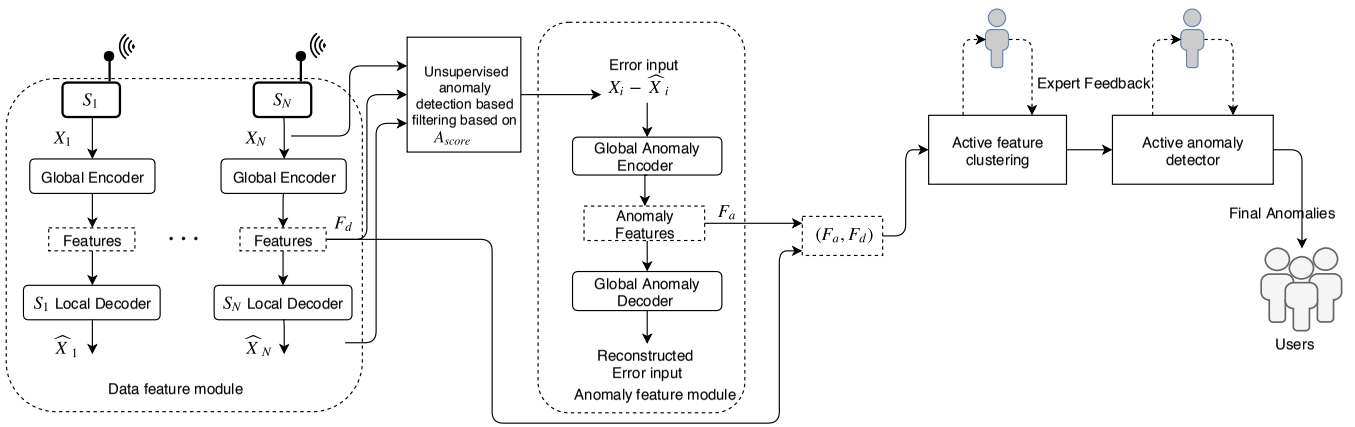


Figure 5.2: Anomaly detection and clustering framework: The framework consists of an unsupervised data feature module and anomaly feature module for optimized feature extraction from the data collected from multiple sensors S_1 to S_N. These features are further used for active anomaly clustering and detection with user interaction.

in an unsupervised feature space across a large number of crowdsourced sensors. The developed framework should also detect a change in the process generating the input spectrum data and retrain the unsupervised model in case of large input distribution variations.

Our contribution: An unsupervised anomaly detection phase followed by a semi-supervised phase is introduced to detect anomalies from frequent as well as less frequent features. Additionally, the proposed framework incorporates provisions for automated model retraining and threshold adjustments of the unsupervised detection phase enabling improved anomaly detection across sensors.

5.1.3 Challenge 3: Anomaly knowledge sharing across sensors

Large variations of sensed wireless spectrum from sensors at different physical locations and regions makes anomaly knowledge sharing across sensors challenging. Knowledge about common anomaly types should be shared across all sensors and frequency bands for fast anomaly detection and generalization of the cause of anomalies in different locations.

Our contribution: We introduce an active anomaly clustering process, as shown in Figure 5.2, to group similar anomalies across sensors from different frequency bands with varying SNRs. The specialized active clustering helps to learn relevant local level distances to separate even closely located clusters if they show different behaviour. We show that knowledge acquired through pairwise queries on a single sensor can be effectively used to group similar anomalies from other sensors.

5.1.4 Challenge 4: Incorporating user feedback

In a crowdsourced network anomalies vary based on user or application perspective. One user's signal might be another's anomaly. So it is important to curate and present anomalies to the users based on their preferences across multiple sensors. In addition, the user anomaly feedback can aid the global anomaly detection process by semi-supervised learning. Thus the main challenge is to device a scheme that can make use of expert labeling from minimal interaction with the users. The labeling process should neither be labour intensive for the users nor prone to much accuracy variations with some level of expected incorrect labeling.

Our contribution: An active anomaly detection after interactive clustering is proposed, which makes use of label propagation on the clusters, for anomaly detection. Anomaly detection over clusters helps the framework to learn

anomalies from a few labeled instances (as anomalous or not) in each cluster and even easily store anomalies based on user preferences. Furthermore, the proposed anomaly detection model parameters can be adjusted to combat incorrect answering both in the active clustering and anomaly detection phases.

The rest of the chapter is organized as follows. Relevant anomaly detectors proposed in the literature and presented in Section 5.2. Section 5.3 explains the anomaly detection framework along with the training and detection algorithms. The dataset used for analyzing the performance of the proposed framework is presented in Section 5.4. The performance results and the advantages of the proposed model are discussed in Section 5.5. The chapter is concluded in Section 5.6.

5.2 Related work

The anomaly detectors proposed in literature can be mainly brought down to two major classes: unsupervised and semi-supervised anomaly detection.

5.2.1 Unsupervised anomaly detection

Various general anomaly detectors are proposed in literature such as OSVM, IFO [69] and Local Outlier Factor (LOF) [92] which makes the basic assumption that outliers represent potential anomalies. These models normally perform poor with increasing input data dimensionality. An anomaly detector for DSA is presented in [27], where distributed power measurements via cooperative sensing are used for anomaly detection. The proposed detector is limited to authorized user anomaly detection only, for the specific case of DSA. Similarly [28] makes use of HMM on spectral amplitude probabilities that can detect interference on the channel of interest again in the DSA domain. Recently in [58], the authors presented a RNN anomaly detector based on predictive modeling of raw IQ data. The authors used a LSTM model for predicting the next 4 IQ samples from the past 32 samples and an anomaly is detected based on the prediction error. Even though this model works on raw physical layer data which requires no expert feature extraction, it is still not sufficiently automated and generic for practical anomaly detection.

In [59], the authors extend this prediction idea on spectrograms and test the model on some synthetic anomalies. A reconstruction based anomaly detector based on vanilla deep autoencoders is presented in [60]. In SAIFE [39] we proposed an AAE [62] based anomaly detector which fills the shortcomings of

all the aforementioned methods and can be trained in a semi-supervised fashion for extracting interpretable features, localize anomalies in the spectrum along with data compression and signal classification abilities. An adapted version of SAIFE is used for the initial unsupervised anomaly detection part of the proposed framework in this chapter.

5.2.2 Semi-supervised anomaly detection

A few semi-supervised anomaly detection algorithms exist in literature such as LODA [70,93]. In [70], the authors make use of an unsupervised ensemble-based algorithm which is further enhanced with an active learning phase in [93]. A semi-supervised accuracy-at-the-top loss function is used to update weights of LODA based on user feedback. The base anomaly detection algorithm is not limited to LODA and the authors also show that the active anomaly detection phase can be incorporated with tree based algorithms too [94]. On the other hand in [95], the authors formulate anomaly detection as an optimization problem and propose a generalization using both labeled and unlabeled instances. Recently, in [96] the authors proposed Semi-Supervised Detection of Outliers (SSDO), a semi-supervised anomaly scoring algorithm which relies on clustering and user labeling.

5.3 Anomaly detection framework

We address all the challenges mentioned in the introduction by developing an unsupervised anomaly detection module followed by active anomaly clustering and detection modules as shown in the framework Figure 5.2. The framework consists of four major components: a data feature module, an anomaly feature module, an active clustering module and an active anomaly detector module which are explained in detail in the following subsections.

5.3.1 Module-1: Data feature generator and unsupervised anomaly detection

The data feature module extracts optimized compressed features from the sensed spectrogram data generated by each sensor. We adapt SAIFE [39], an AAE based unsupervised anomaly detector, with a different training procedure to address *Challenge 1* mentioned in the introduction. The layers used for the encoder and decoder of both data and anomaly feature modules are summarized

Section	Layer	Type	Kernel	No. of filters
Encoder	1,2,3	Conv+maxpool	(3,3)	32, 32, 32
	4_1	Fully connected (softmax)	8 (nlabels)	-
	4_2	Fully connected (linear)	50 (nfeatures)	-
Decoder	1,2,3	Conv+upsample	(3,3)	32, 64, 64
	4	Conv	(3,3)	1
	5	Fully connected (linear)	6*300	-

Table 5.1: Data and anomaly feature module layers.

in Table 5.1. It has been already shown that SAIFE achieves very good anomaly detection accuracy when the anomaly is defined as a less frequent event. In addition, SAIFE can be trained in a semi-supervised fashion to extract interpretable results such as the signal position and bandwidth. The model can also achieve anomaly localization in the frequency space.

For this specific crowdsourced problem and further to help addressing *Challenge 3*, the SAIFE encoder and decoder modules are adapted to address the data variability across the sensors. We make use of a global data feature encoder which helps to obtain a *uniform feature space* across sensors for further clustering. In addition, a global encoder also allows to *share knowledge* across sensors and *generalize* the feature space better, as a single sensor might not witness all wireless signals. Along with a common generalized data feature space, the model should also achieve good initial anomaly detection capabilities. To achieve this, we make use of a sensor specific decoder module which prevents the model from averaging out sensor specific anomalies. For example, if we consider FM radio stations at various locations their centre frequencies might be different. If the decoder is trained globally, the model will perceive that an FM transmission is present at all centre frequencies preventing it from detecting an anomalous pirate transmitter at a particular location.

An initial anomaly detection is performed after generating the data features using the standard anomaly detection score [39] based on the reconstruction loss (R_l), discriminator losses (D_{lcont} and D_{lcat}) and classification error as

summarized below:

$$A_{score} = (R_l > (\mu_{R_{lt}} + n * \sigma_{R_{lt}}))$$
$$\vee ((\mu_{D_{ltcont}} - n * \sigma_{D_{ltcont}}) > D_{lcont} > (\mu_{D_{ltcont}} + n * \sigma_{D_{ltcont}}))$$
$$\vee ((\mu_{D_{ltcat}} - n * \sigma_{D_{ltcat}}) > D_{lcat} > (\mu_{D_{ltcat}} + n * \sigma_{D_{ltcat}}))$$
$$\vee (Class_{Encoder}! = Class_{input}) \quad (5.1)$$

A simple n-sigma threshold is employed on the reconstruction and discriminator loss based on the mean (μ) and standard deviation (σ) values from the training data. An input data frame is classified as anomalous if A_{score} is $True$. The data feature module filters out a large number of normal input spectrum regions enabling the anomaly feature module to learn much more anomaly specific features as shown in Figure 5.2.

5.3.2 Module-2: Anomaly feature generator

This module captures the anomalous spectrum features that the data feature module is trained not to capture (as the data feature module extracts only frequent data features). The difference between the input and the reconstructed signal ($X - \hat{X}$) of all the anomalies detected by the data feature module is fed to the anomaly feature extractor module. We make use of the difference signal as these would represent only anomalies without normal data. This module is global to all the sensors which creates a common anomaly feature space across sensors for future clustering. An AAE similar to the one used in the data feature module is also employed in this module. The whole training process of the data and anomaly feature modules are summarized in Algorithm 1. Both the data and anomaly feature use the same encoder and decoder layers as summarized in Table 5.1. $Tanh$ activation is used for all convolutional layers and maxpooling and upsampling uses a factor of 2.

5.3.3 Module-3: Active clustering module

The extracted anomaly and data features are then fed to an interactive clustering module for further processing. The aim of the clustering module is to group similar anomalies and normal behaviour which are present in different frequency bands across different sensors with minimal user interaction. We make use of the data features along with anomaly features (F_a, F_d), to enable users to cluster even anomalies from the normal feature space. Constraint-based

Algorithm 1 Data and anomaly features module training

Input: **X**: a set of spectrograms from different sensors
Output: a model that can generate data and anomaly features, (F_a, F_d)

1: Train data feature encoder and decoder to reduce reconstruction loss of **X** along with other losses
2: Global encoder: Freeze encoder weights
3: Sensor i specific decoder D_i: Train only the decoder part of the model specific to each sensor i using the data X_i from corresponding sensors.
4: Freeze weights of the data feature module
5: Determine detection threshold n, for the n-sigma cutoff of SAIFE
6: Initial unsupervised anomaly detection with SAIFE
7: Train anomaly feature module on the error signal $X_i - \hat{X}_i$ from all sensors i
8: Freeze weights of the anomaly feature module

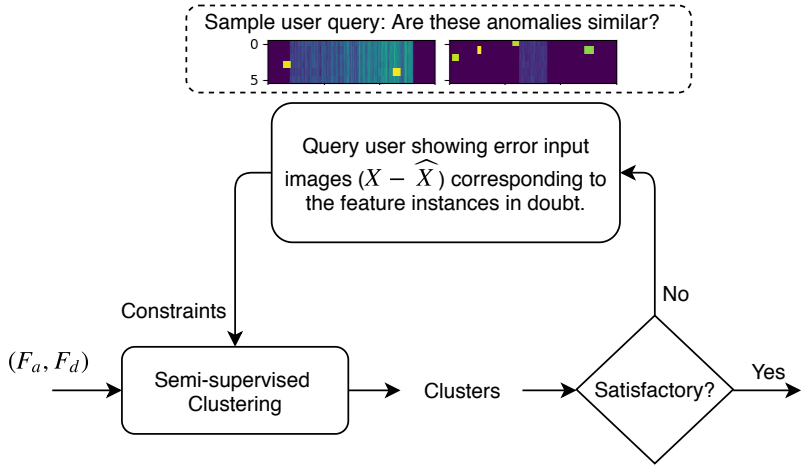


Figure 5.3: COBRAS clustering system relying on interaction in the form of pairwise queries.

Repeated Aggregation and Splitting (COBRAS) [97], a query and time efficient interactive clustering algorithm is used for achieving clustering.

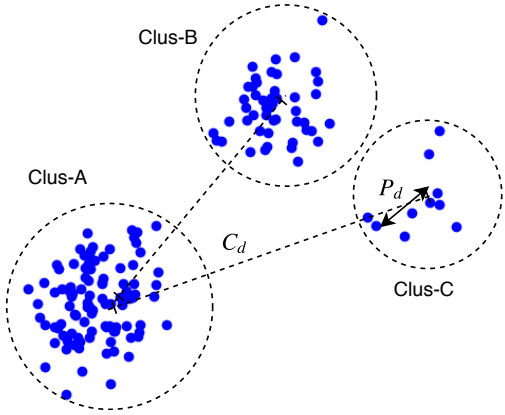
COBRAS makes use of the concept of super-instances: sets of instances that are assumed to belong to the same cluster in the unknown target clustering. COBRAS starts with a super-instance that contains all instances and gradually refines the clustering based on pairwise queries asking whether the shown anomalies belong to the same cluster or not. In each iteration COBRAS performs two steps. First, it picks the largest super-instance from the cluster and splits it into new super-instances. A constraint based procedure is used

for splitting the largest super-instance. The algorithm splits the largest super-instance into two new temporary super-instances based on 2-means and then queries the user, the relation between their medoids [98]. Medoids are similar in concept to means, but are restricted to the data points in the dataset or cluster whose average dissimilarity with all the cluster points are minimal. In our case, the error input images $(X_i - \hat{X}_i)$ for the corresponding medoid features are shown to the user, asking whether they belong to the same cluster or not, as shown in Figure 5.3. In this chapter the term *query* when mentioned related to clustering refers to the process of asking the user whether a pair belongs to the same cluster or not. If they can be linked, COBRAS considers the super-instance as pure, otherwise the clustering process is repeated on other super-instances. In the second merging step the relation between newly created clusters are determined and clusters are merged if they belong to the same group based on constraints. If the relation between two clusters is not known, they are queried again for user response. Detailed algorithm steps can be found in the original paper.

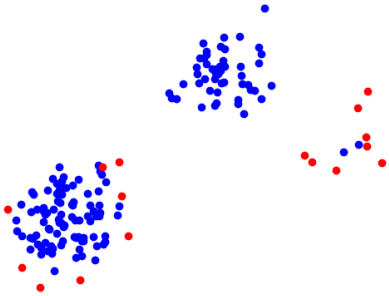
5.3.4 Module-4: Active anomaly detector

The active anomaly detector is an integral module of the entire framework which solves the *Challenge 3* and *Challenge 4* mentioned in the introduction. This module helps to label the clusters generated by COBRAS as anomalous or not. As mentioned in the introduction less frequent events detected by SAIFE could be non anomalous. In addition, the active anomaly detection also helps to store user specific anomaly preferences when anomaly labeling is done by a single user, still giving the flexibility to share these anomaly label knowledge across users. For instance, in a crowdsourced network the clusters which are labeled as anomaly by a majority of users can be considered as a global anomaly.

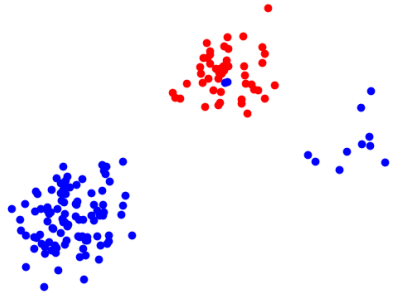
An anomaly score is generated on the clustered instances using SSDO [96], which can work in a semi-supervised setting. SSDO performs semi-supervised anomaly detection in two major steps after the clustering step as shown in Figure 5.4. First, it does an unsupervised anomaly scoring of instances based on three measures: (i) the point deviation of an instance from its cluster centre, (ii) the cluster deviation of a selected cluster from other clusters and (iii) the cluster size. In addition, SSDO supports score updating based on user anomaly labels with two controllable parameters α and k. α controls the impact of the user provided labels versus the labels from the clustering process and k determines the depth of label propagation for a single labeled instance. These parameters are very useful in controlling the speed of the entire process and helps in dealing with noisy labels from users. In this chapter the term *query* when mentioned related to anomaly detection refers to the process of asking the



(a) Initial clustering: Intuitive point deviation (P_d) and cluster deviation (C_d)



(b) Unsupervised anomaly detection: detects outliers in clusters.



(c) Anomalies after label propagation: allows to refine anomaly labels for clusters.

Figure 5.4: SSDO anomaly detection example: After achieving initial clustering using COBRAS, the algorithm performs unsupervised anomaly scoring and anomaly label propagation. In this example labels are provided saying clusters A and C are normal and Clus-B is anomalous (anomalies in red).

user whether an instance is anomalous or not. The whole anomaly detection process is summarized in Algorithm 2. The reader is encouraged to read the original paper [96] for detailed equations of these measures.

5.3.5 Users and expert feedback

As shown in the framework Figure 5.2, users also play an important role in the clustering and anomaly detection process. Electrosense, already leverages crowdsourced sensors for large scale spectrum monitoring. In the newly proposed

Algorithm 2 Active Anomaly detection

Input: (F_a, F_d)
Output: Anomalous spectrograms

1: Active clustering on (F_a, F_d) using COBRAS
2: Active anomaly score generation using SSDO
3: Present final anomalies to users

Algorithm 3 Model retraining (Section 5.4.2)

1: R_{la} = Reconstruction loss of the $X_i - \hat{X}_i$ from the anomaly feature module
2: **if** $R_{la} >$ threshold(2σ) **then**
3: Add input to the anomaly training set
4: **while** True **do**
5: Retrain anomaly feature module
6: Adjust data feature module detection threshold based on new labeled anomalies not detected by the data feature SAIFE and other anomaly clusters
7: Retrain data and anomaly feature module in case of low anomaly scores for majority of clusters
8: Sleep(threshold_days)

framework, expert user feedback is utilized both in the clustering and anomaly detection phase. By expert feedback, we refer to feedback from the wireless domain experts: either who develop and train the model or reputed experts among users. As the clustering process helps to share knowledge across sensors, we propose to involve experts in the clustering process. We also analyze the effect of incorrect labeling or feedback in detail in Section 5.5. The active anomaly detection after clustering could be made open to all users as even with labeling errors we can store them as user specific anomaly preferences and use it globally across sensors based on majority voting. We would like to eventually enhance the spectrum monitoring experience for users with more user engagement in the anomaly detection process, by making this framework a part of Electrosense.

5.4 Datasets

Spectrogram data from the PSD pipeline of four electrosense sensors over a period of 56 days (starting from 8 Nov, 2018) are collected for the analysis of

the proposed framework. The data is collected through the Electrosense API[1] with a spectral resolution of 10 kHz and time resolution of 60 seconds. The selected sensors are located in Spain, Denmark, Switzerland and Slovenia and data from the same frequency bands are collected from all these sensors as given in Table 5.2. These sensors are low-cost RTLSDRs configured at a sampling rate of 2.4 MS/s with omni-directional antennas which are deployed indoors. The sensors follow sequential scanning of the spectrum with an FFT size set to 256 giving a frequency resolution close to 10 kHz. With a FFT size of 256 and sensor ADC bit-width of 8, we get an effective bitwidth of 12 resulting in a theoretical dynamic range of 74dB. Practical dynamic range depends on the ADC frontend stages and the noise level, which may vary between 60 to 65dB. Five FFT vectors are averaged for reducing the thermal noise of the receiver.

Three synthetic signals (i) *scont*: single continuous signal with random bandwidth, (ii) *randpulses*: random pulsed transmissions on the given band and (iii) *wpulse*: pulsed wideband signals covering the entire frequency with SNR from -20 to +20 dB are used as anomalies. These synthetic anomalies are injected to the original collected data which help us to have ground truth clusters (three clusters) and labels of the anomalies for further analysis.

5.4.1 Feature extraction model training

The dataset mentioned in the previous section is split into two subsets, a training and a testing subset, with equal number of vectors. A seed is used to generate random mutually exclusive array indices, which are then used to split the data into two ascertaining the training and testing sets are entirely different. The train and test datasets for performance analysis on the Electrosense dataset are selected from different, non-adjacent, time periods (7 days for training and the other 49 days for testing) to make sure that there are no identical points in the training and test dataset. Spectrogram inputs each with a vector size 6x300 (time-frequency bins) are used as input to the model. The model is trained in an unsupervised fashion to reduce the mean squared error between the input and decoder output and a semi-supervised fashion to learn the the continuous features and class labels. The adversarial networks as well as the autoencoder are trained in three phases: the reconstruction, regularization and semi-supervised phase as mentioned in [62]. The Adam optimizer [68], a first-order gradient based optimizer, with a learning rate of 0.001 is used for training in all the phases. In the semi-supervised phase the model is trained to learn the class, position and bandwidth of the input signal by training it on 20% of the labeled samples from the training set. The model is trained for 500

[1] https://electrosense.org/open-api-spec.html

Count	Frequencies (MHz)	Expected technology/modulations
1	86-108	FM
2	192-197	Broadcasting (DAB/Microphones)
3	198-228	Broadcasting (DAB/Microphones)
4	790-801	Tactical links/TV Broadcasting
5	801-810	Tactical links/TV Broadcasting
6	811-821	Tactical links/TV Broadcasting
7	933-935	Mobile (GSM-900)
8	955-960	Mobile (GSM-900)

Table 5.2: Electrosense dataset frequency bands.

epochs which takes around one hour of training time on a x86 PC with Nvidia GeForce GTX 980 Ti graphics card.

5.4.2 Model retraining

The anomaly and data feature module should be incrementally trained in the following scenarios. An increase in the anomaly detector reconstruction loss (R_{la}) indicates that new unseen anomalies are getting fed to the anomaly feature module. This demands retraining of the anomaly feature module. Similarly, a large number of non-anomalous clusters after expert labeling process might either indicate there are large process variations in the sensor output or the selected low frequency events are not anomalous to the users. This demands retraining of the feature extractor modules along with the initial unsupervised threshold update. The framework can also enable users to label portions of the spectrum that are anomalous but not detected by the initial unsupervised detection phase, resulting in a possible threshold reduction for incorporating more normal behaviours to the anomaly space. Summary of the training phase can be found in Algorithms 1, 2 and 3.

5.5 Results and discussion

The framework is evaluated based on the unsupervised anomaly detection performance, clustering performance and finally based on the anomaly detection accuracy. We answer the following questions based on the analysis:

Q1: How good is the initial unsupervised anomaly detection performance compared to SoA models with the new interpretable common feature space and training process (Challenge 1)?

Q2: Does the common feature space along with active clustering help to share knowledge of anomalies across sensors (Challenge 3)?

Q3: Can the model effectively detect meaningful anomalies even in the feature space with active learning (Challenge 2)? Can the model achieve this with a few labels and work reasonably well even with incorrect feedback (Challenge 4)?

5.5.1 Q1: Feature extraction module performance

The trained data feature module's performance is compared against various SoA algorithms such as OSVM, IFO [69] and LODA [70]. The average anomaly detection accuracy and false positive rates of these algorithms over 8 frequency bands, 3 anomalies and four sensors for a cutoff $n = 2$ are plotted in Figure 5.5. It can be noticed that SAIFE performs better than all algorithms with good detection accuracy and low false positive rates. The false positive rate of other algorithms can be reduced by increasing the n value of the n-sigma cutoff threshold as shown in Figure 5.6 at the cost of detection accuracy. These results show that unsupervised threshold selection, even with full anomaly ground truth knowledge, is difficult highlighting the need for a further semi-supervised phase. To systematically compare different algorithms, the detection accuracy for a constant false alarm rate of 5% and 10% are plotted in Figure 5.7. It can be noticed that SAIFE performs superior to all algorithms. Even with the new training process the performance of the model is consistent with the original model mentioned in [39] with anomalies like *scont* which simulates new random continuous transmissions in different bands and sensor locations. This shows that the proposed training process is able to detect new pirate transmitters as discussed in Section 5.3.2. In addition, SAIFE provides an interpretable feature space when compared to other algorithms.

5.5.2 Q2: Clustering performance

The average clustering performance of COBRAS for three anomalies (three ground truth clusters) on 8 different frequency bands and four sensors for different SNRs is plotted in Figure 5.8. The clustering performance measure plotted is Adjusted Rand Index (ARI) [99] which is a symmetric similarity measure between two clusterings, in this case between the produced anomaly clustering and ground truth. As mentioned in Section 5.4, we make use of

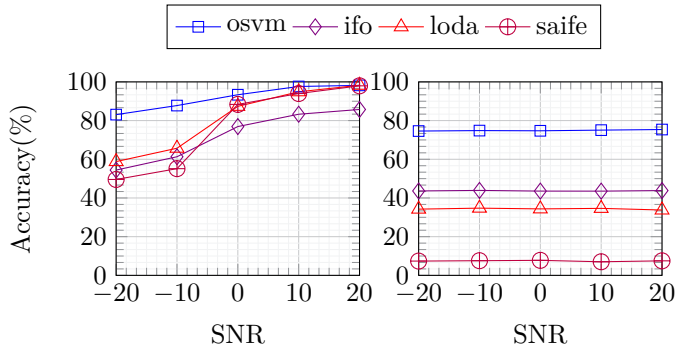


Figure 5.5: Averaged true anomaly detection accuracies (left) and false positive rates (right) for different algorithms on 8 frequency bands from four Electrosense sensors with n-sigma cutoff $n = 2$.

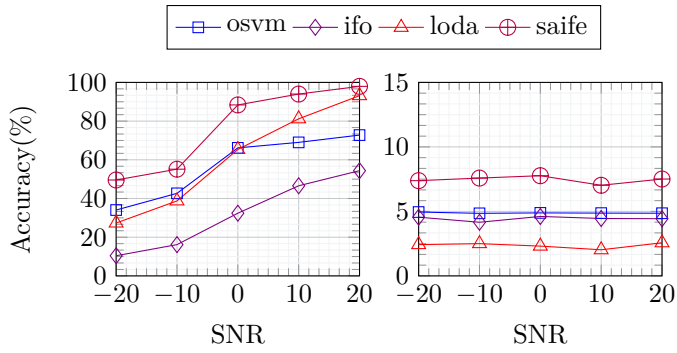


Figure 5.6: Averaged true anomaly detection accuracies (left) and false positive rates (right) for different algorithms on 8 frequency bands from four Electrosense sensors. n-sigma cutoff $n = 4$ for IFO and LODA, $n = 6$ for OSVM and SAIFE with $n = 2$ for comparison.

samples with three added synthetic anomalies, providing three clusters as our ground truth. The results shown are averaged scores from 10 runs where COBRAS is allowed to query from data from all sensors. The clustering accuracy of high SNR anomalies (20 and 10 dB) reach above 80% with tens of queries while the low SNR anomalies require queries in hundreds to reach the same.

In order to test the generalization capability across different sensors the clustering performance is plotted again with only answering queries of anomalies from a single sensor. The results shown in Figure 5.9 confirms that a few more queries are required even at high SNR anomaly conditions to reach the 80% mark when compared to answering queries from all sensors as expected. This validates that the global anomaly feature space across sensors helps to

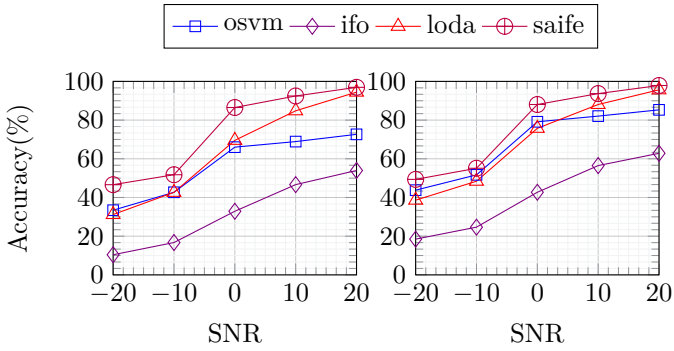


Figure 5.7: Averaged true anomaly detection accuracies for different algorithms on 8 frequency bands from four Electrosense sensors for a constant false alarm rate of 5% (left) and 10%(right).

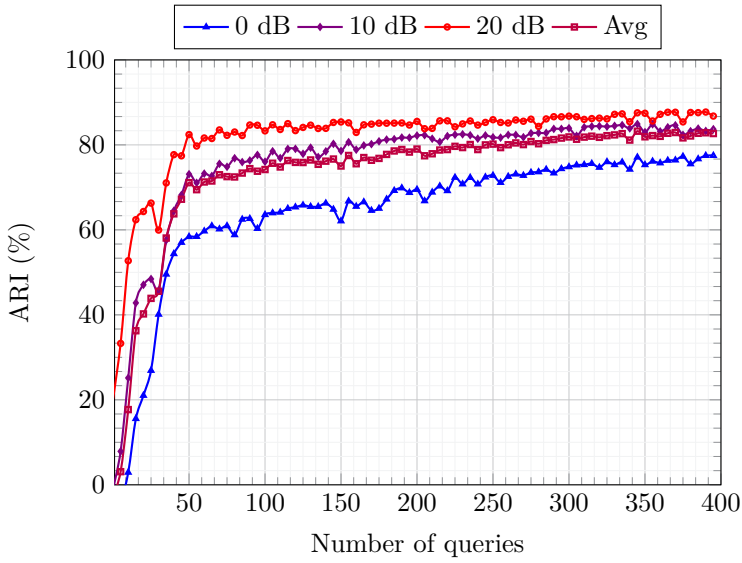


Figure 5.8: Average clustering ARI with queries answered from all four sensors.

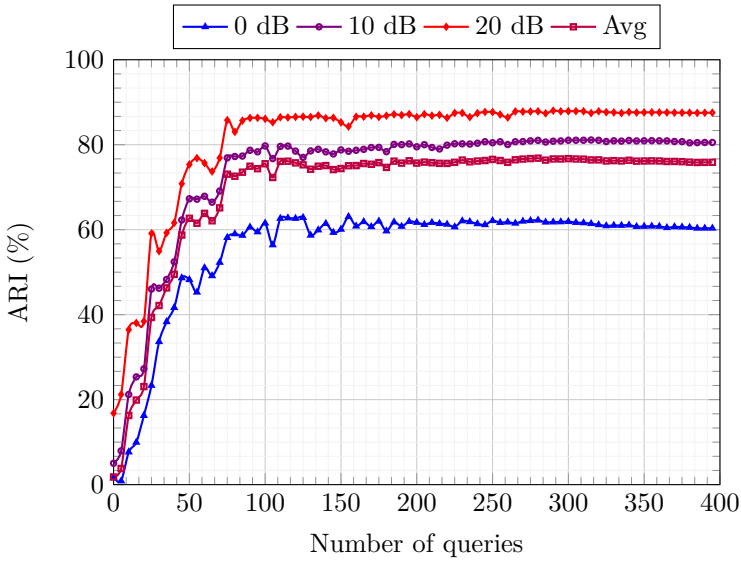


Figure 5.9: Average clustering ARI with queries answered from only one sensor for different SNRs.

generalize similar anomalies across sensors. After a few queries, the clustering performance saturates and it doesn't keep on improving as in the previous multi-sensor querying scenario. This indeed indicates the need to enable interactive querying from a few different sensors to improve the clustering performance. To further validate this, we include the average ARI across all SNRs by enabling querying from multiple sensors. The results plotted in Figure 5.10 shows that the clustering accuracy gets better with a fewer queries with more sensors and it keeps on increasing at a faster rate when compared to lower sensor counts.

Further, the clustering performance is evaluated for understanding the effect of wrong query answers which might happen especially in a crowdsourced expert-feedback paradigm. The average ARI for various probabilities of incorrect answering (P_i) are shown in Figure 5.11. It can be seen that the clustering performance is significantly affected by incorrect answering. Above $P_i = 0.2$ the clustering performance is largely affected. At very high incorrect answering rates it is better to switch to an unsupervised clustering algorithm or adjust the α and k parameters of the anomaly detection algorithm (SSDO) to give low priority to the clustering process.

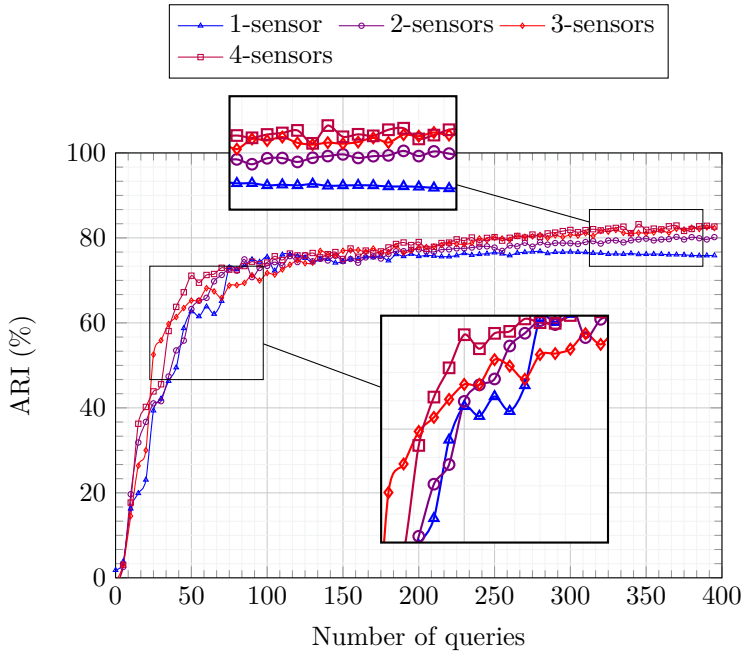


Figure 5.10: Average clustering ARI with queries answered for different sensor counts.

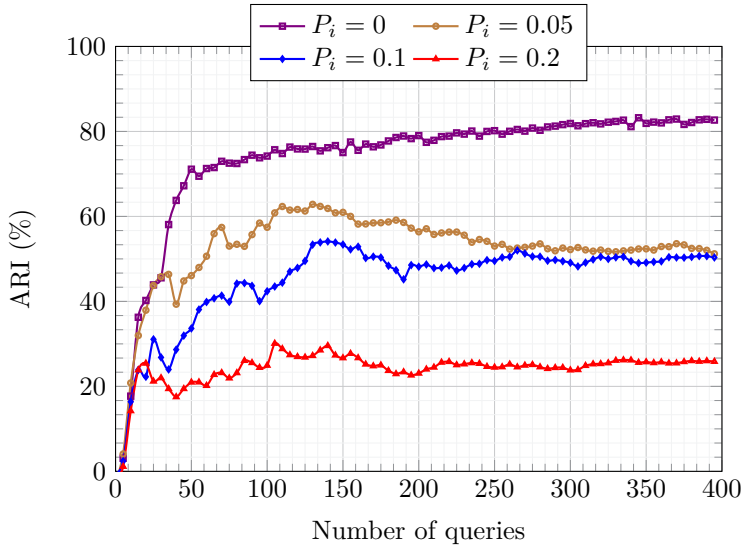


Figure 5.11: Average clustering ARI with queries answered with different incorrect answering probabilities (P_i).

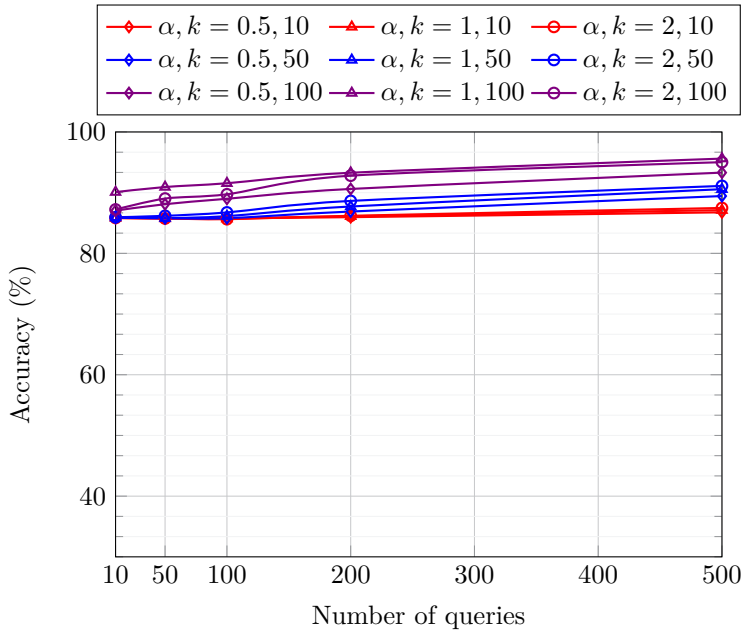


Figure 5.12: Average anomaly detection accuracy with correct labeling. α controls the impact of the user provided labels versus the labels from the clustering process and k determines the depth of label propagation for a single labeled instance.

5.5.3 Q3: Anomaly detection performance

The clustered features are fed to the SSDO algorithm for further anomaly labeling as explained in Section 5.3.4. As the signals fed to the detection phase are initially detected anomalies from the data feature module, we randomly select one of the three synthetic anomalies as non-anomalous and then continue the detection procedure to investigate if we can learn that it is not an anomaly. A random instance is chosen from each cluster and shown to the user for confirming as anomalous or not and the process is continued in a round *robin-fashion* for all clusters. From Figure 5.12, showing the averaged anomaly detection accuracy across all sensors and SNRs, it is clear that with a few labels or queries itself (each query provides single anomaly label), the model can achieve a good detection accuracy above 90%. The speed of convergence is also affected by the values of α and k, mentioned in the Section 5.3.4, as expected. The anomaly detection performance is also analyzed for various parameter values with a incorrect labeling probability of $P_i = 0.05$ both in the clustering and anomaly detection phase. It can be seen that even with a low starting clustering accuracy

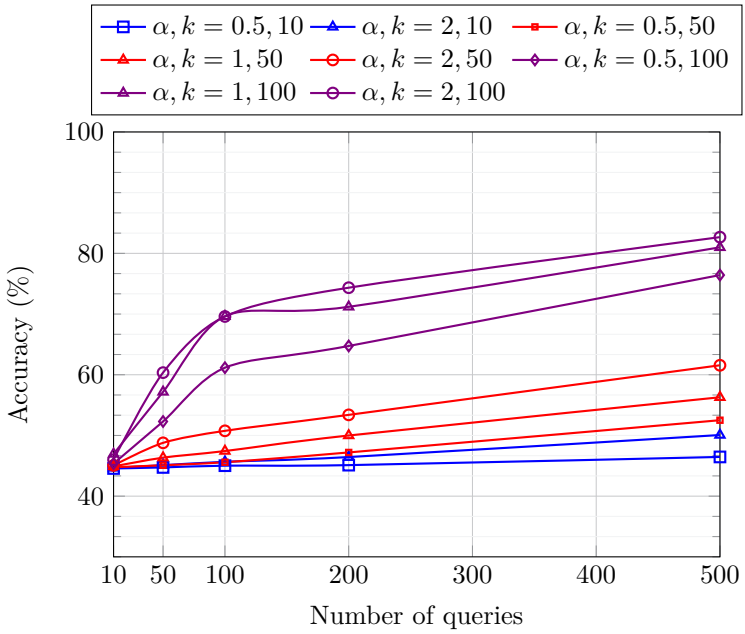


Figure 5.13: Average anomaly detection accuracy with incorrect cluster and anomaly labeling probability $P_i = 0.05$.

of 40% at $P_i = 0.05$ the anomaly detection process is able to reach close to 80% accuracy with 500 queries. With wrong labeling the convergence is slower which is also an expected behaviour. More advanced strategies such as *selective repeated-labeling* [100] can be deployed for improving the detection accuracy under noisy conditions.

5.6 Conclusion

Detecting wireless spectrum anomalies is a challenging problem due to the complexity of wireless spectrum usage and the electromagnetic environment itself. In this chapter we have proposed an active framework for wireless spectrum anomaly detection with distributed crowdsourced sensors. The model performs an unsupervised initial anomaly detection and outperforms SoA algorithms. This unsupervised anomaly detection is followed by a semi-supervised active anomaly detection phase where it provides above 95% detection accuracy on the selected anomalies. The active anomaly detection phase is also resilient to noisy labeling and it is shown that the model achieves good accuracy close to

80% with only hundreds of queries. The model also provides good anomaly clustering performance which will be used in future for informing users about relevant anomalies based on user preferences.

Chapter 6

Situational awareness for spectrum sharing

The benefits of wireless spectrum awareness specific to a spectrum sharing scenario in cognitive radio is analyzed in this chapter. An open challenge framework has been created and radio experts from all over the world were invited to showcase their spectrum sharing solutions. SoA solutions were evaluated in terms of achieving full spectrum awareness and network performance improvements with the provided situational awareness. The presented results emphasize the need for open spectrum sensing frameworks such as Electrosense that can provide online spectrum awareness for enhancing wireless network performance.



Figure 6.1: Perfect wireless situational awareness and transmission prediction are key for DSA.

This chapter contains results from:

Wunsch, Felix; Paisana, Francisco; Rajendran, Sreeraj; Selim, Ahmed; Alvarez, Pedro; Muller, Sebastian; Koslowski, Sebastian; Van den Bergh, Bertold; Pollin, Sofie; *DySPAN Spectrum Challenge:*

Situational Awareness and Opportunistic Spectrum Access Benchmarked. IEEE Transactions on Cognitive Communications and Networking; 2017; Vol. 3; iss. 3; pp. 550 - 562.

Rajendran, Sreeraj; Van den Bergh, Bertold; Vermeulen, Tom; Pollin, Sofie; *IEEE 5G Spectrum Sharing Challenge: A Practical Evaluation of Learning and Feedback.* IEEE Communications Magazine; 2016; Vol. 54; iss. 11; pp. 210 - 216.

6.1 Introduction

It has been more than two decades since Joseph Mitola coined the term cognitive radio in 1998. A cognitive radio is an intelligent and autonomous system that can adapt its transmission and reception parameters based on the environment. Since then, wireless technology has become a lot more sophisticated. Capacity is boosted by adding more configurable algorithms and by utilizing various spectrum bands. Still it remains an open question how much learning is needed and how relevant it is to tune those knobs and bands dynamically in function of the environment. Learning the environment requires feedback about the environment. This feedback can be obtained by online spectrum sensing or from a database.

Wireless data usage has increased tremendously over the past few years. New spectral resources are allocated to meet this ever increasing demand. These new allocations are limited as the wireless spectrum is scarce. To enable efficient utilization of this scarce resource, it is becoming crucial that wireless radios share the available spectrum. Efficient sharing can be enabled only when all the participating radios are well aware of their operating wireless environment. Thus wireless situation awareness is unavoidable in modern radios for enabling efficient spectrum sharing and reliable performance especially in highly interfering scenarios. For instance, multiple self-organized wireless networks can share the spectrum and work independently without explicit coordination. In essence, the main questions related to the design of dynamic spectrum access radios, or cognitive radios as Mitola framed them, are (a) how much benefits can be achieved by learning and adapting, compared to other solutions at physical or medium access layer, and (b) what feedback information should be available to facilitate learning and adaptation.

To understand the current state of these spectrum sharing algorithms, various competitions were held in the past. The first DARPA spectrum challenge[1],

[1] www.darpa.mil/program/spectrum-challenge

a prominent one among them, put SoA cognitive algorithms to the test by designing competitive and cooperative competitions. This challenge gave insight to the adaptability of the SoA radio algorithms in using the spectrum aggressively or sharing it among other users based on the environment. A second version of the challenge is hosted by DARPA[2], which is still going on at the time of writing this thesis, is pushing the research in dynamic spectrum sharing further. Emulating electromagnetic waves for testing algorithms in the physical world is no small task. DARPA designed a complex SDR testbed for electromagnetic space, named Colosseum[3], with 128 two-antenna SDR units. The framework used and the solutions created during both challenges were not made public to the research community.

We organized two spectrum challenges collaborating with IEEE, to comprehend the benefits of learning and adaptation of wireless parameters and the importance of feedback information to facilitate this learning and adaptation, in 2015 [41] and 2017 respectively [42]. The main difference with the DARPA challenge is that the DARPA challenge focused mainly on cooperation using a given radio and physical layer challenges and not so much on the learning or feedback information requirements of a cognitive radio network. Whereas the IEEE spectrum sharing challenge focused on learning and feedback information requirements of a cognitive radio network using database aided spectrum sharing. The participants were given complete freedom in their radio and physical layer design, within the spectrum constraints. Another spectrum sharing challenge is the SPECTRUM-SHARC Student Cognitive Radio Contest[4], where students are given access to the CORNET cognitive radio testbed. Here the hardware was fixed but teams were allowed to make waveform changes. The goals of the challenge are similar to the IEEE spectrum sharing challenge in that the Secondary User (SU) needs to maximize its throughput while minimizing its interference to the Primary User (PU). However, realtime feedback of the PU and SU throughput and physical design freedom made IEEE spectrum sharing challenge different from SPECTRUM-SHARC challenge. The complete challenge framework and proposed solutions of the IEEE spectrum challenges were also made public for the research community.

[2]https://www.spectrumcollaborationchallenge.com/
[3]https://www.darpa.mil/news-events/2017-04-21
[4]http://radiocontest.wireless.vt.edu/index.html

6.2 Spectrum challenge-1

IEEE DySPAN 2015 has organized a spectrum challenge[5], which was designed in such a way that any wireless research group could participate. It was up to the teams to decide how much effort they would spend on antenna design, novel hardware, physical layer solutions or medium access protocols. Each team could individually decide how much adaptation and learning to include. As a novel enhancement to the learning process, a database was implemented with real-time feedback about the throughput of the incumbent link. This would give each team instantaneous feedback about interference caused to the primary receiver, enabling detailed learning of the optimal transmission parameters.

While receiver feedback is currently not present in any spectrum sharing database, it is not unrealistic to explore with new growing spectrum monitoring databases like the one from Electrosense. First, it would give an upper bound on what learning and feedback would bring. Second, we see a trend towards more and more receiver based regulation, giving the receiver a larger role in the spectrum optimization problem[6]. Third, a lot of wireless systems already implement some kind of acknowledgment, which is receiver feedback that can be used to optimize transmitter parameters.

A high level overview of the IEEE DySPAN 2015 challenge setup is given in Figure 6.2. The database provides packets and performance metrics in real-time to the transmitter and collects statistics about the system performance. Both the PU and the SU radio were connected to the database and could at most use two antennas. The PU system was based on IEEE 802.15.4, which was chosen as this is one of the cheapest wireless system one can find. Therefore, even on a very limited budget, each team would be able to setup the PU system in their lab. For the challenge, the PU system was implemented using a SDR IEEE 802.15.4 implementation running on two Universal Software Radio Peripheral (USRP)s. This implementation follows the standard but has different RF properties compared to the low-cost dongle. The main idea being that while some properties are given, some remain to be learned and calibrated by using the feedback in the database or spectrum sensing.

6.2.1 Challenge implementation

The challenge was designed to meet a range of criteria: enable breadth in the solutions, enable teams with no hardware or expertise to participate and enable

[5] http://dyspan2015.ieee-dyspan.org/content/5g-spectrum-sharing-challenge

[6] www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_spectrum_report_final_july_20_2012.pdf

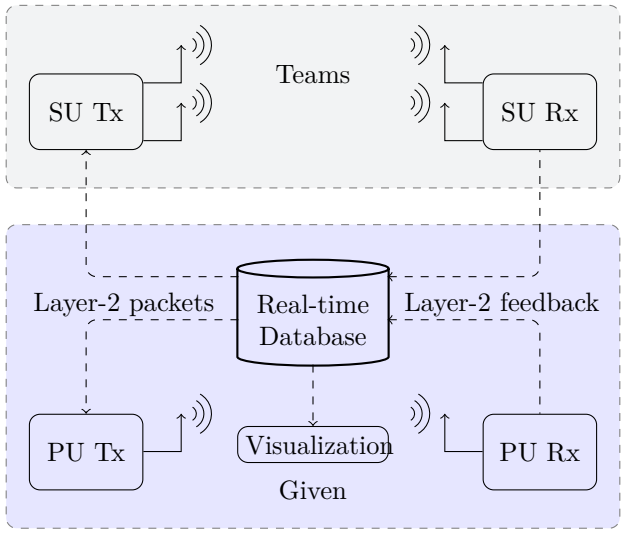


Figure 6.2: Spectrum challenge setup: A real-time feedback from the database is enabled along with score logging. PU and SU throughput and the spectrum usage is displayed by the visualization module.

learning with real-time feedback about the primary user throughput statistics. Given those high-level criteria, a standard IEEE 802.15.4 stack for a widely available wireless dongle was provided to ensure the participation of teams with only MAC experts who want to avoid physical layer implementation hassles. The heart of the solution was a database, that could in real-time talk to the dongle's IEEE 802.15.4 standard stack, as well as GNU Radio and LabVIEW software defined radio systems. As a result, PU and SU could be any system from a low-cost dongle with an IEEE 802.15.4 stack to custom implementations on SDRs.

To make the competition more challenging, the PU simultaneously transmitted on four predefined frequency bands with channel spacing of 5 MHz. The secondary user had to maximize its throughput over the same 20 MHz which the PU was using. The center frequency of the band is 2.3 GHz, which was a dedicated interference-free band used for the challenge. As a result, teams had to learn only the room characteristics (fading and pathloss parameters), PU traffic patterns and the exact PU RF properties of the USRP-X310s used.

6.2.2 Primary user setup

The PU radio had a four channel GNU Radio based IEEE 802.15.4 stack [101] with an O-QPSK physical layer connected to a USRP front-end via Ethernet interface as shown in Figure 6.5. The PU used four independent streams which were configured to transmit packets independently on the four channels using a fixed packet length of 127 bytes. The packet generator block controlled these four streams by requesting data from the database, re-sizing it and pushing the packets to the independent 802.15.4 MAC modules of the streams. The timing of these packets was controlled by the packet generator and was based on four instantiations of the same random distribution. The packet generator switched to a new distribution every 3 minutes to mimic the non-stationary nature of channel occupancy.

The random distributions used in the challenge consisted of

- a uniform distribution with minimum and maximum inter-packet duration of 8ms and 150ms respectively;
- two poisson distributions with means 20ms and 150ms;
- a back-to-back transmission scheme with minimal inter-packet period of 5ms.

All the parameters of these distributions were configured before the start of the challenge. A common starting seed was used for all distributions which helped in keeping the randomness fixed across the teams during both phases of the challenge. The implemented code can be easily adapted to follow different distributions to test more complex channel occupancy scenarios in future. In reality, there can be many different spectrum occupancy scenarios, so a fairly large set of possible random distributions were chosen for the challenge.

6.2.3 Database and feedback

Giving feedback of the PU performance to the SU radio should allow for much faster and more fine grained adaptation of the SU settings. This feedback mechanism was implemented using a central database server. The server delivers random packets to both the PU and SU. Received packets are delivered back to the database server where they were verified and statistics were updated. These statistics were accessible from both PU and SU.

To generate packets with strong randomness at high throughput AES-128 was used in counter mode with a random key. It is important that frames are

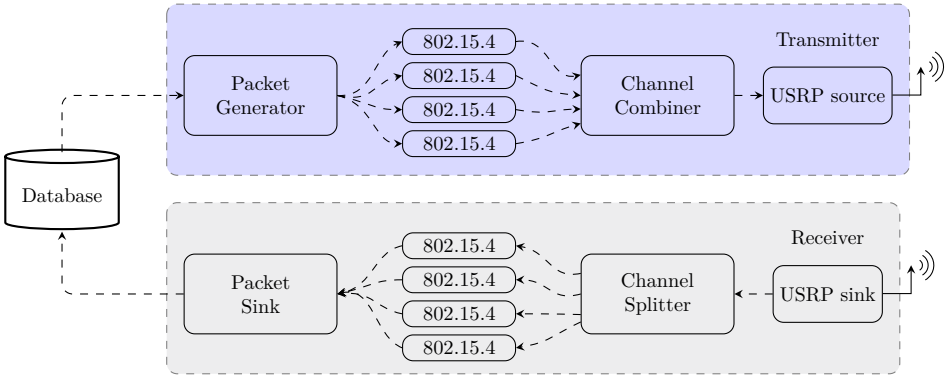


Figure 6.3: Primary user setup

sufficiently random since otherwise they could be compressed before transmission. A secure message authentication code (MAC) is calculated over the frame to detect if it has been corrupted. A sequence number is added to prevent packets from being counted multiple times.

Since we feel that this database platform coupled with client libraries for both GNU Radio and LabVIEW may be useful to others it is made publicly available[7][8]. The PU implementation used in the challenge is also made available in this package.

6.2.4 Challenge phases

The challenge consisted of two phases: a learning phase and a test phase as given below.

- Learning Phase (10 min)
  - SU can learn PU statistics;
  - PU feedback is provided to optimize the SU parameters;
  - No scoring in this phase.
- Test Phase (10 min)
  - Actual scoring phase;

[7] http://claws.be/spectrum-challenge/
[8] https://github.com/networkedsystems/dyspanchallenge_2015

- PU transmission statistics same as during the learning phase;
- SU penalized for interference.

Each phase had a duration of 10 minutes. The SU radio could learn about the environment, the PU transmission statistics and the exact PU transmitter and receiver RF properties impacting the interference sensitivity. This acquired knowledge could then be used to calibrate the SU parameters and algorithms to improve its performance. During the learning phase, the SU could also test its algorithms and validate them efficiently by making use of the real-time feedback from the database. The final scores were only calculated during the test phase of the challenge.

6.2.5 Challenge metric

Two winners were selected. One was selected based on a single metric combining both SU and PU throughput, which could be measured unambiguously from the database. The final challenge score was represented by the product of SU throughput (T_{SU}) and PU satisfaction. The PU satisfaction (S_{PU}) was calculated from the offered PU throughput ($\widehat{T}_{\mathrm{PU}}$) and the delivered PU throughput (T_{PU}) as given in Equation 6.1. A maximum throughput loss tolerance of 10% was allowed. More than 10% PU throughput loss would be counted as zero PU satisfaction.

$$\begin{aligned} Score &= T_{\mathrm{SU}} \times S_{\mathrm{PU}} \\ S_{\mathrm{PU}} &= max\Big(0, \frac{10}{9} T_{\mathrm{PU}} - \widehat{T}_{\mathrm{PU}}\Big). \end{aligned} \tag{6.1}$$

In addition to this single performance metric, which can be measured but does not capture the novelty of the solution, a second winner that was selected by a jury was announced. The jury evaluated mainly the maturity of the solution, and most importantly the breadth, i.e., how much cognitive aspects were considered in the final solution. This would exclude non-adaptive teams, or teams failing to take advantage one way or another from the PU or SU feedback enabled in the system.

6.2.6 Secondary user algorithms

The challenge was designed to enable a large variety of teams to participate in the event. Eventually, 5 teams were selected to compete, using techniques

ranging from special waveforms to advanced statistical PU profiling techniques. A brief overview of each team's implementation is given below. The summary of the used physical layer parameters and co-existence techniques can be found in Table 6.1. Each team published the details of their solution as an IEEE DySPAN 2015 challenge paper [102–106].

6.2.7 Challenge results and discussion

Two runs (20 min for each team per run) of the challenge were conducted and the winner was selected based on the best scores out of two runs. The number of runs were fixed to two in order to give the participants a second opportunity to fine tune their algorithms. The availability of dedicated spectrum and the reproducible PU spectral occupancy pattern made the performance results repeatable. Figure 6.6a and Figure 6.4b details the SU and PU performances respectively. Figure 6.6a summarizes the bytes transferred by each team during both runs of the challenge. Even though most of the teams were able to achieve high SU throughput, only two teams managed to keep the PU throughput loss below 10% (Figure 6.4b). Some teams tried to overcome this using short length packets and fine transmit power control while others used directional systems to keep the interference level low.

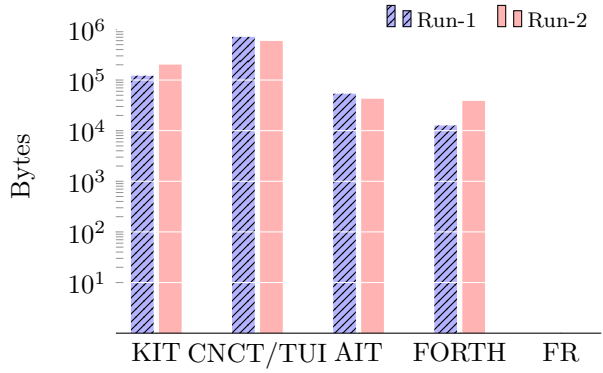
6.2.8 Winning solutions

From the final scores computed from both runs, team CNCT/TUI was selected as the winner based on the metric. CNCT/TUI's high throughput PHY-MAC combination along with beamforming techniques helped in achieving the winning score. Team KIT was selected by the jury as the winner based on the breadth and consistency of the solution. Team KIT performed consistently well with their spectrally efficient PHY and short length packets. Even though team AIT employed many features including adaptive power control based on the PU feedback, they failed in suppressing the carrier leakage from their radio front-end which interfered continuously with the PU. Team FORTH also had difficulties in controlling the interference with the PU as their algorithm assumed only a single channel usage by PU at a time. The non-interfering signal design from team FR also performed badly as the implemented adaptive power control mechanism failed in reducing the interference.

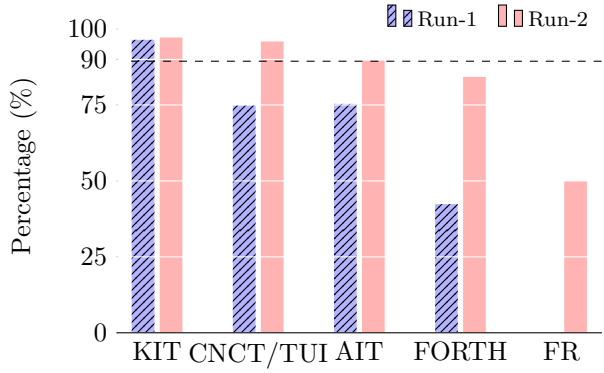
Referring to the feature list we could conclude that a radio with (a) local sensing based learning, (b) spectrally efficient physical layer and (c) directional antennas or equivalent adaptive power control for reducing interference are promising building blocks for dynamic and cognitive spectrum access. Even though the

Team	Physical Layer	Coexistence Techniques
KIT	Fully parameterizable Filter Bank Multicarrier (FBMC) PHY 256 Subcarriers over 4 channels with 5 MHz each, 2 b/s/Hz Low adjacent channel interference High spectral efficiency Short packets for low probability of interference	Adaptive noise floor estimation, no calibration needed Energy detection gives reliable information about PU beyond database Past knowledge from SU transmitter sensing used to improve detection probability Locally sensed busy channels are temporarily blacklisted Omnidirectional antennas ensure consistent performance in dynamic scenarios
CNCT/TUI	OFDM PHY Receive on 4x5MHz channels at Rx Directional Antennas	Efficient multi-threaded solution for simultaneous multi-channel transmission Noise floor estimation on each channel which improves PU detection performance Best modulation and coding scheme selected during the learning phase PU channel occupancy modeled as a Markov chain SU's transmission channel selected based on this Markov model
AIT	Tx-Rx directional Yagi-Uda planar parasitic antennas OFDM Modulation with QPSK or 16QAM	Modulation selected based on PU and the SU throughput feedback SU transmits between two adjacent PU channels to avoid interference (e.g 2.295 - 2.3 GHz) Dynamic gain control to minimize PU interference and boost SU's throughput
FORTH-ICS	4x IEEE 802.11a/g @ 5 MHz (SU Tx/Rx) 4x IEEE 802.15.4 (PU detector) Abstraction in OS - everything appears as typical interface	Virtualization in time and frequency domain PU frame, inter-frame and channel transition durations are decoded Decoded parameters used to select packet length and modulation parameters
FR	Custom physical layer waveform FSK modulated tones used as the base waveforms	SU waveform exhibits negligible projection on the PU waveforms No mutual interaction between PU and SU waveforms Transmit gain control based on the feedback

Table 6.1: Secondary user features



(a) SU Bytes transfered



(b) PU packet success percentage

Figure 6.4: Primary and Secondary user performance

participating radios were not fully context aware, the challenge showed that promising technological building blocks to design DSA solutions for challenging PU systems exist.

6.3 Spectrum challenge-2

During the first challenge it was concluded that the main challenges in realizing such situational awareness include limitations in observing the channel in a half duplex radio receiver, modeling limitations of the non-stationary PU transmissions and the inherent non-determinism of channel occupancy.

Furthermore, benchmarking the importance of situational awareness was challenging mainly due to the versatility of the solutions.

For the IEEE DySPAN spectrum challenge 2017[9], a generic platform was developed to evaluate the SoA wireless algorithm performance in terms of radio situation awareness and dynamic spectrum usage. First, a few fixed scenarios are designed to benchmark the detection performance of the SoA algorithms. Secondly, the algorithms' dynamic spectrum usage performance is tested and analyzed in these scenarios. The complete challenge framework along with the solutions is made public to the research community to enable future benchmarks [10].

The second challenge also followed the same high level structure as given in Figure 6.2. The setup consists of two PU and two SU radios which are connected to a central database server. The rules of the challenge allowed each radio to use at most two antennas. Real-time packets and performance metrics were provided by the database. In addition, the database also logged statistics about the system performance which is forwarded to the visualization module for easy analysis during the course of the challenge.

6.3.1 Challenge problem definition

Some of the major criteria that were considered while designing the challenge included enabling breadth in the solutions in terms of hardware and software solutions and real-time feedback about the PU and SU throughput statistics. To comply with the criteria, an open GNU Radio [66] based OFDM stack was used in the physical layer and made available to the participating teams for easy testing. Even though the PU design was made public, the challenge scenario parameters were randomized during the actual challenge. Challenge winning metrics were selected to answer the following questions

- Situation awareness: How good is SoA research in detecting wireless scenarios?
- Spectrum sharing: How well can the secondary users exploit the spectrum, once they detect the scenarios?
- Receiver feedback: What is the upper bound in performance that can be achieved if there is perfect PU and SU receiver feedback?

[9] http://dyspan2017.ieee-dyspan.org/spectrum-challenge
[10] https://github.com/networkedsystems/dyspanchallenge_2017

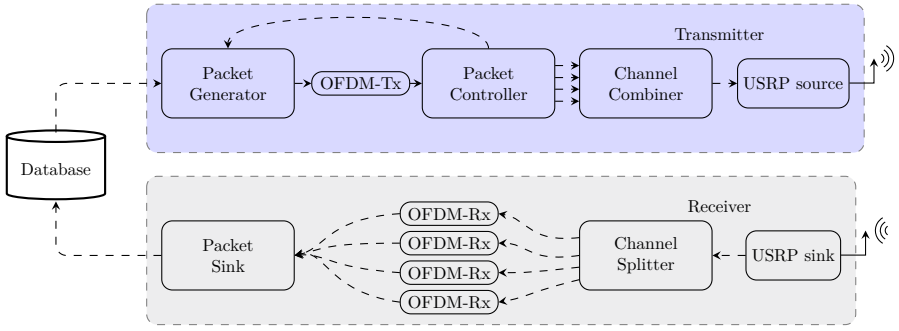


Figure 6.5: Primary user setup

Scenario	Description	Inter-packet delay
0	Single random channel	5 ms
1	Single random channel	10 ms
2	Two random channels, hopping	5 ms
3	Four random channels, hopping	10 ms
4	Four channels, synchronous	5 ms
5	Two random channels, synchronous	5 ms
6	Four channels, synchronous	2 ms
7	Four channels, Poisson-distributed delays	20 ms (mean)
8	Four channels, Poisson-distributed delays	10 ms (mean)
9	Four channels, Poisson-distributed delays	5 ms (mean)

Table 6.2: Challenge scenarios: Different PU scenarios for the challenge.

The PU simultaneously transmitted on four predefined frequency bands with a channel bandwidth of 2.5 MHz. The secondary user had to transmit over the same 10 MHz band which the PU was using. The spectrum usage was monitored in real-time and all out-of-band transmissions were heavily penalized. In order to avoid interferences from other wireless devices, a dedicated band with a center frequency of 3195 MHz was used during the challenge. More about the challenge setup, phases and scores is detailed in the following subsections.

6.3.2 Challenge scenarios

Selecting realistic wireless spectrum occupancy scenarios was important to benchmark the current capabilities of SoA research. Detailed models could be found in literature capturing the very nature of spectrum occupancies, in

terms of probability distributions, that can model the busy and idle periods of real systems [107–110]. Various experimental studies assuming a constant packet length and Poisson distributed packet arrival times were conducted in [107]. Wireless traffic analysis based on more realistic sources were done in [108]. In most of these analyses the time spent on data transmission and acknowledgement states were found to be deterministic, while the idle periods were fitted to generalized Pareto distribution or a mixture of uniform distributions. In [110] authors analyzed the busy and idle periods with multiple wireless devices operating in 2.4GHz ISM band. They concluded that hyper-exponential distributions fits the busy and idle periods very well while generalized Pareto distributions can provide good approximations.

During the DySPAN spectrum challenge 2015 [41], it was noticed that participants found it difficult to adapt their algorithms to highly random scenarios. Furthermore, simple distributions were preferred over complex ones to understand the baseline performance of SoA algorithms as well as reduce the parameter estimation overhead for participants. Acknowledging all these factors, 10 scenarios were selected with varying levels of difficulty. The scenarios ranged from the PU occupying a single random channel with deterministic delay to independent transmissions on all channels with random delays drawn from a distribution as listed in Table 6.2. The deterministic single channel scenarios modeled realistic pilot transmissions in real world wireless transmissions. For difficult scenarios, the inter-packet delays were sampled from Poisson distributions under the assumption that with a quite large number of users the packet arrival event is uncorrelated holding the memoryless property. A packet payload length of 64 bytes was selected and the inter-packet period for each scenario is also listed in the table. Including the header and CRC, the entire packet duration was about 200 microseconds. The scenario switching time, the time period in which the PU stays in a particular scenario, is selected randomly from a uniform distribution between 15 and 30 seconds.

6.3.3 Primary User Setup

The PU radio used a four channel GNU Radio based OFDM stack which was connected to a USRP X310 front-end via Ethernet interface as shown in Figure 6.5. The packet controller block was responsible for controlling the inter-packet timing in the four channels for different scenarios. The packet controller selects a particular scenario from the ones shown in Table 6.2, instructs the packet generator to request required packets from the database and sends the packets to the OFDM block for modulation. The modulated packets are then split into four independent channels which are passed to the channel combiner for further transmission over the air using the USRP transmitter. The packet

controller and channel combiner together achieved the exact timing parameters for each scenario by following a sample count based design. On the receiver side the channel splitter splits the four channels and distributes them to four independent OFDM receiver blocks for demodulation. The demodulated packets are then sent to the database server for validation.

6.3.4 Database and Feedback

A central database server is implemented to provide PU and SU performance feedback. The server delivers random packets in requested number and size to the PU and SU transmitters. Packets received at the receiver end are sent back to the database where they are verified for correctness. The database also logs the PU and SU throughput along with the situation awareness status reported by the SU.

AES-128 is used in counter mode with a random key to generate packets with strong randomness at high throughput. The frames are sufficiently random thereby restricting further compression before transmission. A secure message authentication code (MAC) is calculated over the frame to detect if it has been corrupted. A sequence number is added to prevent packets from being counted multiple times. AES along with a secure MAC were used to prevent participating teams from predicting future packets or generating their own packets at the receiver end without transmitting over the air.

6.3.5 Challenge Phases

The challenge consisted of two phases: a situation awareness phase and an agile spectrum usage phase as given below.

- Situation awareness
  - SU should detect each scenario and report it to the database
  - PU feedback is provided to optimize the SU spectrum usage and radio parameters. SU is not penalized for interference which allows testing of spectrum sharing algorithms along with scenario detection.
  - Situation awareness score calculated in this phase
- Agile spectrum usage
  - SU should share the spectrum for data transmission
  - SU is penalized for interference

– Spectrum usage metric calculated in this phase

Each phase had a duration of 10 minutes. The SU radio could learn about the environment, the PU transmission statistics and the exact PU transmitter and receiver RF properties impacting the interference sensitivity. This acquired knowledge could then be used to calibrate the SU parameters and algorithms to improve its performance.

6.3.6 Challenge metric

From the competing teams, two winners are selected, one metric scores the teams in function of the throughput and the other in function of their spectrum situation awareness. The spectrum agility metric was represented by the product of SU throughput (T_{SU}) and PU satisfaction. The PU satisfaction (S_{PU}) was calculated from the offered PU throughput ($\widehat{T}_{\mathrm{PU}}$) and the delivered PU throughput (T_{PU}) as given in Equation 6.2.

$$\begin{aligned} Score &= T_{\mathrm{SU}} \times S_{\mathrm{PU}} \\ S_{\mathrm{PU}} &= e^{-10\left(\frac{\widehat{T}_{\mathrm{PU}} - T_{\mathrm{PU}}}{T_{\mathrm{PU}}}\right)}. \end{aligned} \tag{6.2}$$

The situation awareness score is computed as the percentage of time the SU had perfect situation awareness. The score is calculated as given in Equation 6.3 where $\widehat{D}$ is the number of 1 millisecond time slots in which the SU radio has correct situation awareness. The total number of slots $D = 10 \times 60 \times 10^3$.

$$Score = \frac{\widehat{D}}{D} \tag{6.3}$$

6.3.7 Winning situational awareness approach

The solution provided by Trinity College, Dublin (TCD) [111] comprises the following components:

- frequency-hopping system capable of overlay operation
- transmitter and receiver with OFDM waveform
- deep learning chain for PU's scenario classification

- energy detection for PU packet detection and positioning in frequency and time
- continuous adaptation of SU's frequency hopping pattern based on sensing results
- transmit power control for underlay operation, which relies on the database feedback

This solution is implemented in C++, using standalone libraries such as liquid-dsp for signal processing, and caffe for deep learning. The solution source code[11] and the samples dataset[12] used for training and testing the deep learning chain along with the trained models are made public for future research. The solution uses off-the-shelf USRP N210 hardware, where the reception and sensing is done using omnidirectional antennas, while the transmitter uses a directional waveguide antenna made of an open-ended metal can, for increased SNR and directionality. This solution was built on top of TCD's DySPAN challenge 2015 objective winner solution [112].

6.3.8 Winning agile SU approach

The team from KIT presented a cognitive overlay system based on a Filter Bank Multicarrier (FBMC) physical layer (PHY) [113]. The solution is based on the system that was first demonstrated in Stockholm at DySPAN 2015 [114]. In order to demonstrate a competitive system for the challenge, however, the PHY as well as the cognitive component were improved in several respects. FBMC is a suitable choice for overlay systems due to its high selectivity and high spectral efficiency. With respect to the previous challenge, the synchronization and equalization algorithms is adapted to reflect current state-of-the-art research. More reliable synchronization and equalization also allows the utilization of higher order modulation schemes and therefore a higher achievable throughput.

To enable a cognitive SU behavior, a classification of the PU scenario based on the extraction of multiple features of the PU's behavior is performed. This knowledge is then used to predict transmission opportunities and thereby completely avoids interference with the PU. Additionally, the current scenario estimation is continuously validated in order to react to unexpected changes in the environment in a timely manner. For the implementation, the GNU Radio SDR framework [115] was chosen due to its good performance in real-time applications. The employed hardware comprised two USRP B210 SDRs and two commercial off-the-shelf notebook computers.

[11] `https://github.com/alvasMan/dyspan_radio_2017`
[12] `http://dx.doi.org/10.7910/DVN/EBLENC`

6.3.9 Challenge results and discussion

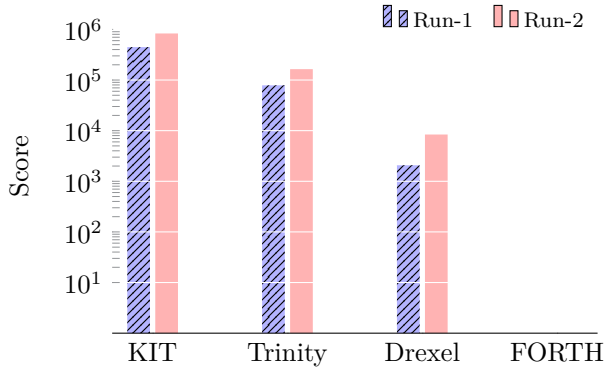
Two runs of the challenge, each of 20 minutes with two phases, were carried out to give the teams a second opportunity to fine tune their algorithms. The winner for each of the two phases was selected based on the best score out of the two runs. The availability of the dedicated spectrum and reproducible PU spectral occupancy pattern made the performance results repeatable.

The final agile spectrum usage score is summarized in Figure 6.6a. Team KIT achieved the highest spectrum usage score during both runs with their advanced FBMC physical layer and cognitive allocator. As shown in Figure 6.6b, team Trinity achieved the highest situational awareness score of 99.31% using the non-linear deep learning model which had rich internal state representations. Team FORTH also achieved a good score of 99.04 during their second run. Team KIT's expert feature based classifier performed quite well with an overall accuracy of 98.14% which turned out to be one of the best solutions for both phases.

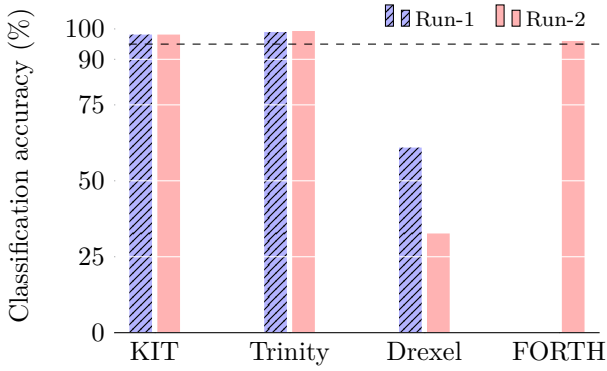
6.4 Future benchmarking

This second IEEE DySPAN spectrum challenge was an evolution of the first challenge organized at IEEE DySPAN 2015 [41]. To strengthen the benefits of learning and situational awareness, it was decided to predefine specific patterns for the PU. The PU in 2015 was too irregular and impossible to predict. In addition, the PU PHY was replaced by an OFDM PHY, instead of the IEEE 802.15.4 PHY used in the first challenge. A part from these changes, the core logic and setup of the challenge (implemented by a central database that controls SU and PU packet flows) was reused. We made the source code of the challenge core database, the metric and PU scenarios of the IEEE DySPAN challenge 2017, and the two winning solutions as described in this chapter public through the github repository mentioned in the introduction. Below, we describe two example scenarios for future experiments building on the current results and code.

First, it is possible to rerun the challenge exactly as it was defined, by testing both winning solutions in a specific and new environment. While rerunning the winning solutions, it is also possible to test a novel improved SU design, that can then then benchmarked against the current best solutions. If a better score is obtained, a new record can then be established. If the source code of the new winner is made available, future solutions can then again be benchmarked against the new winner as well. As the source code for the challenge database,



(a) Agile spectrum usage score



(b) Situation awareness score

Figure 6.6: Final scores

and the two winners as described in this chapter are shared in a joint github repository, this avenue is already possible.

Alternatively, it is possible to start from the challenge core database, and change the rules, metrics or PU characteristics. By doing so, a new challenge is defined, emphasizing novel research problems or aiming to steer research effort towards a well defined problem. This requires somebody to take the lead in reshaping the problem statement, and altering the metric or database. By doing so, a new series of winners can be established, targeting a slightly different version of the challenge. We can compare this to the Guiness book of records, where there are numerous winners, for numerous types of world records.

While sharing the challenge source code, and the code for the winner on GitHub,

is a good starting point for organizing such benchmarking series, we should realise that the performance of each run is expected to differ a lot. Depending on the size or the room, the level of interference, or the exact hardware used to run the code, a different performance is expected. To minimize the impact of the environment or hardware on the performance, it would be better if there would be a test facility where competing solutions could be tested, continuously using the same environment and hardware. While the establishment of such facility is at the moment future work, it is for sure something the DySPAN community should consider.

6.5 Conclusions

The DySPAN spectrum challenges brought together researchers to demonstrate and compare their solutions for cognitive SU systems in a scenario with a highly dynamic PU. The challenge offered an opportunity for the exchange of experience, ideas, and visions for the development of practical DSA systems.

The first phase of the spectrum challenge-2 showed how recent advances in the research on machine learning can be leveraged and applied to a cognitive radio context. The team from Trinity College, Dublin, achieved the highest situation awareness score with over 99% classification accuracy by successfully employing deep learning techniques to classify PU behavioral patterns. Their classification approach was both highly accurate and fast, which resulted in this near-perfect score.

In the second phase of the challenge-2, SU throughput and PU interference were rated. The winning solution for this task was presented by the team from the Karlsruhe Institute of Technology. Their solution combined a spectrally efficient FBMC waveform with the prediction of transmission opportunities through continuous classification of the PU, therefore completely avoiding interference. By the effective use of the available time-frequency resources, a peak throughput of about 12 Mbit/s was achieved.

In order to create a lasting impact and to enable reproducibility, the code for the PU as well as the winning SU solutions were published online. This way, new solutions can be benchmarked and compared with the existing systems, possibly establishing a series of records and continually improving the system's design and algorithms. In order to allow better control over the environment, future spectrum challenges could also be held in a test facility, providing exactly identical conditions for all participants. Alternatively, the freely available code for the PU and database server could be used to adapt the challenge to focus on different problems encountered in the research on spectrum sharing.

Finally, we can conclude that the spectrum challenges showed that the practical implementation of DSA systems is still a challenging task. Even though designing such systems are challenging, the spectrum challenges scores clearly showed that spectrum situational awareness and transmission prediction are key to the success of cognitive radios in practice. These experiment results are inline with research directions presented in the previous chapters and back the need for spectrum monitoring and information extraction frameworks such as Electrosense.

Chapter 7

Conclusion and Future work

> I would rather have questions that can't be answered than answers that can't be questioned.
>
> – Richard Feynman

This chapter concludes the thesis with a concise review of the original research questions and contributions of this work along with some suggestions for future research.

7.1 Research review

7.1.1 How to monitor the complex and dense electromagnetic space in a cost-effective manner?

Monitoring wireless spectrum over time, frequency and space is a hard target to achieve. Electrosense, a crowdsourced spectrum monitoring platform with low-cost sensors that can support a wide range of usecases ranging from spectrum decoding to anomalous event detection for spectrum enforcement, is designed in collaboration with other partners in the Electrosense team. In this thesis, I proposed various data extraction, compression and aggregation techniques for big data analysis, using unsupervised and semi-supervised machine learning techniques. In addition, the sensor control and data backend are developed keeping scalability as the main design parameter. Electrosense has proved that

large scale spectrum monitoring can be achieved in a cost-effective manner providing benefits both to participating volunteers and other spectrum owning organizations.

In a crowdsourced network, quality of the produced data vary from user to user based on various factors such as sensor location and the used antenna type. Evaluating sensor quality and ranking sensors based on uptime and data quality is found to be challenging. Further, user incentives that can motivate more volunteers to join crowdsourced spectrum sensing networks, is another active discussion that is pursued. Various spectrum analysis tools and decoders are getting added to Electrosense network for enhancing user engagements. The cost of SDRs and single board computers are decreasing every year, which I believe will also help to improve the current sensor hardware configurations with additional features such as machine learning acceleration and signal localization capabilities.

7.1.2 How to detect and classify anomalous behaviors in the wireless spectrum efficiently?

I introduced SAIFE, a new unsupervised anomaly detection model which can directly detect anomalies from spectrogram inputs without any extra feature extraction. The model's compression, band classification, anomaly localization and semi-supervised feature extraction capabilities are other advantages which promotes the use of this model. The model also showed very good anomaly detection capabilities compared to SoA, when anomalies are less frequent events in the training dataset. Thus the proposed model supported the fact that high quality anomaly detection is possible by only using unsupervised models, with some shortcomings if the anomalies are highly frequent in the training dataset.

Further, new supervised deep learning architectures for signal classification were also proposed. It was shown that deep learning models can perform fast signal classification using IQ samples and perform better than traditional methods. It was also concluded that even a coarse level of technology classification can be achieved by only using PSD data. However, PSD based signal classification fails when the signals have same bandwidth and transmission parameters such as the same pulse shaping filter.

7.1.3 How to incorporate user feedback to the entire monitoring framework to improve spectrum awareness?

Anomalies can be outliers in the data or very frequent events which prevents modeling of complete anomaly detection as an unsupervised learning problem. In Chapter 5, I introduced a novel anomaly detection framework which works in a semi-supervised fashion with expert feedback. The model exhibited very good anomaly detection accuracy compared to SoA, in various frequency bands across sensors. The framework also combats incorrect feedback from users which is also studied in detail. The framework also includes facilities to share expert knowledge across sensors. Thus fully featured anomaly detection in wireless spectrum space is achieved by closing the loop with user feedback. It was also shown that semi-supervised deep learning models can help in reducing the labeling challenges associated with big data, especially specific to the signal classification problem. Thus expert semi-supervision can be leveraged in data-driven models, which requires only 10% labeled samples for signal classification and tens of labels for anomaly detection on the entire training set which is promising for future research.

7.1.4 How good is SoA research in utilizing the achieved spectrum awareness for improving wireless communication performance?

Two organized spectrum challenges gave a lot of insights to the current state of cognitive radio systems. All the winning teams showed that wireless spectrum awareness can greatly help the communication system to improve throughput of both the individual systems and the network as a whole. We have also seen great improvements in the radio systems in the second challenge when compared to the first one. These improvements are mainly from the teams leveraging recent advancements in physical layer modulations, SDR architectures especially combing recent FPGA based SDR implementations and machine learning. Machine learning and game theoretic models have reached a mature state to tackle complex wireless spectrum sharing problems which can be also seen in the latest DARPA spectrum challenges.

To conclude, it is quite clear that deep learning and future machine learning models will change the way in which we design modern wireless communication systems. The proposed methods clearly show the high value of deep learning models specifically for the spectrum monitoring problem. Semi-supervision, especially from wireless domain experts, can be efficiently leveraged to solve the data labeling problems in data driven spectrum monitoring solutions. The age of

AI-radios will be arriving soon and the future of seamless wireless communication looks bright.

7.2 Future work

Low-cost SDR hardware and advancements in big data have helped in developing large scale spectrum monitoring frameworks. New machine learning models are getting developed actively for better understanding the signals for instance, very deep models for wireless signal classification [31]. Gradually, deep learning models are becoming an ingrained part of future wireless technologies especially for sensing and end-to-end communication systems [32]. The following are some open problems related to wireless spectrum monitoring where new solutions can create a huge impact.

7.2.1 Machine learning

Interpretable feature extraction: In Chapter 3, with SAIFE, I have only looked into one discrete interpretable feature namely category of the signal. For a wireless signal there are a wide range of other continuous features such a the bandwidth, symbol rate, channel occupancy duty cycle and so on. The semi-supervised InfoGAN model presented in Chapter 4 also extracts some continuous features which can also be trained in a semi-supervised manner. The expressive power of alternative models based on variational autoencoders in terms of extracting disentangled features from the wireless data is also an interesting research direction that can help the automated spectrum interpretation.

Expert feature extraction models: Deep learning models for wireless signal classification presently use layers which basically apply non-linearity after simple multiply-accumulate-add operations while it is well established in the research community that cyclic cumulants, which are generated by time-shifted multiplication and averaging of the input itself, performs well in the expert feature space. Models that can take advantage of these features in the internal layers might be useful in improving the performance of current models. These models might also turn useful for multi-label classification setup which can further aid interference analysis.

Models for sub-sampled wireless signals: Recently there is a surge in interest for sub-sampled wireless classification models which can enable low-energy spectrum sensing [116]. It should be also noted that, in Electrosense, the deployed low-cost spectrum sensors' SDR has many limitations including low

sampling bandwidth (max of 3.2 MS/s) which prevents it from receiving high bandwidth signals like Wi-Fi or LTE. Efforts to develop deep learning models that can work efficiently on sub-sampled signals are worth pursuing.

7.2.2 Anomaly geo-localization

Localizing detected anomalies geographically is very important to mitigate imminent threats in high priority wireless communication scenarios such as airports. Anomaly geo-localization is also important to mitigate intentional or unintentional jamming events. When compared to conventional localization, one of the major challenges in localizing these signals is that they are not cooperating. It is difficult to localize unknown interfering signals with complex frequency hopping patterns. This itself makes the problem an interesting one to tackle in future.

7.2.3 Crowd engagement and sustainability

An important challenge that we see for large scale deployments is to improve the engagement of new users. What are the best incentives that can motivate new users to join the network and share spectrum data? How to keep the crowd-sourced model cost-effective in terms of the backend processing and storage costs? These are all worthwhile problems to be tackled for a self-sustaining spectrum monitoring framework.

7.2.4 Sensor data quality

There are serious concerns about the data quality in any crowdsourced network. Tackling sensor data quality challenges with proper data quality based sensor ranking and fake data detection algorithms are important problems to be tackled in the near future.

Curriculum Vitae

Sreeraj Rajendran received his Masters degree in communication and signal processing from the Indian Institute of Technology, Bombay, in 2013. He is currently pursuing the PhD degree in the Department of Electrical Engineering, KU Leuven, Belgium. Before joining KU Leuven, he worked as a senior design engineer in the baseband team of Cadence and as an ASIC verification engineer in Wipro Technologies. His main research interests include machine learning algorithms for wireless and low power wireless sensor networks.

List of publications

Articles in International Journals (6)

- **Rajendran, Sreeraj**; Meert, Wannes; Lenders, Vincent; Pollin, Sofie; 2019. *Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features.* IEEE Transactions on Cognitive Communications and Networking; *to appear*, 2019
- **Rajendran, Sreeraj**; Meert, Wannes; Giustiniano, Domenico; Lenders, Vincent; Pollin, Sofie; *Deep Learning Models for Wireless Signal Classification with Distributed Low-Cost Spectrum Sensors.* IEEE Transactions on Cognitive Communications and Networking; 2018; Vol. 4; iss. 3; pp. 433 - 445.
- **Rajendran, Sreeraj**; Calvo-Palomino, Roberto; Fuchs, Markus; Van den Bergh, Bertold; Cordobés, Héctor; Giustiniano, Domenico; Pollin, Sofie; Lenders, Vincent; *Electrosense: Open and Big Spectrum Data.* IEEE Communications Magazine; 2018; Vol. 56; iss. 1; pp. 210 - 217.
- Azari, Mohammad Mahdi; Sallouha, Hazem; Chiumento, Alessandro; **Rajendran, Sreeraj**; Vinogradov, Zhenya; Pollin, Sofie; *Key Technologies and System Trade-Offs for Detection and Localization of Amateur Drones.* IEEE Communications Magazine; 2018; Vol. 56; iss. 1; pp. 51 - 57.
- Wunsch, Felix; Paisana, Francisco; **Rajendran, Sreeraj**; Selim, Ahmed; Alvarez, Pedro; Muller, Sebastian; Koslowski, Sebastian; Van den Bergh, Bertold; Pollin, Sofie; *DySPAN Spectrum Challenge: Situational Awareness and Opportunistic Spectrum Access Benchmarked.* IEEE Transactions on Cognitive Communications and Networking; 2017; Vol. 3; iss. 3; pp. 550 - 562.
- **Rajendran, Sreeraj**; Van den Bergh, Bertold; Vermeulen, Tom; Pollin, Sofie; *IEEE 5G Spectrum Sharing Challenge: A Practical Evaluation of*

Learning and Feedback. IEEE Communications Magazine; 2016; Vol. 54; iss. 11; pp. 210 - 216.

Journals submitted and under review (3)

- **Rajendran, Sreeraj**; Lenders, Vincent; Meert, Wannes; Pollin, Sofie; 2019. *Crowdsourced wireless spectrum anomaly detection.* IEEE Transactions on Cognitive Communications and Networking; 2019 – *Submitted.*
- Sallouha, Hazem; Chiumento, Alessandro; **Rajendran, Sreeraj**; Pollin, Sofie; *Localization in Ultra Narrow Band IoT Networks: Design Guidelines and Trade-Offs.* IEEE Internet of Things Journal, 2019 –*Submitted.*
- Calvo-Palomino, Roberto; Cordobés, Héctor; Engel, Markus; Fuchs, Markus; Jain, Pratiksha; Liechti, Marc; **Rajendran, Sreeraj**; Schäfer, Matthias; Bergh, Bertold Van den; Pollin, Sofie; "*Electrosense+: Empowering People to Decode the Radio Spectrum*, arXiv preprint arXiv:1811.12265, 2018 –*Submitted.*

Book chapter under review (1)

- **Rajendran, Sreeraj** and Pollin, Sofie; *Large scale wireless spectrum monitoring: Challenges and Solutions*, Wiley book on Spectrum Sharing, 2019.

Articles in Conferences (6)

- Reynders, Brecht; Iyare, Rachel; **Rajendran, Sreeraj**; Volskiy, Vladimir; Vandenbosch, Guy AE; Pollin, Sofie; 2018. *Using Cheap RTL-SDRs for Measuring Electrosmog.* 2018 IEEE Symposium on Communications and Vehicular Technology (SCVT) (SCVT'18); 2018 Publisher: IEEE
- **Rajendran, Sreeraj**; Meert, Wannes; Lenders, Vincent; Pollin, Sofie; 2018. *SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features.* IEEE International Symposium on Dynamic Spectrum Access Networks; 2018 Publisher: IEEE
- Minucci, Franco; Van den Bergh, Bertold; **Rajendran, Sreeraj**; Giustiniano, Domenico; Cordobés, Héctor; Fuchs, Markus; Calvo-Palomino, Roberto; Lenders, Vincent; Pollin, Sofie; Demo: *Electrosense - Spectrum Sensing with Increased Frequency Range.* EWSN 2018; 2018 Publisher: ACM;

- Van den Bergh, Bertold; Giustiniano, Domenico; Cordobés, Héctor; Fuchs, Markus; Calvo-Palomino, Roberto; Pollin, Sofie; **Rajendran, Sreeraj**; Lenders, Vincent; 2017. *Electrosense: Crowdsourcing Spectrum Monitoring.* 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (IEEE DYSPAN); 2017; pp. - Publisher: IEEE

- Thoen, Bart; Ottoy, Geoffrey; Rosas, Fernando; Lauwereins, Steven; **Rajendran, Sreeraj**; De Strycker, Lieven; Pollin, Sofie; Verhelst, Marian; *Saving Energy in WSNs for Acoustic Surveillance Applications while Maintaining QoS.* 2017 IEEE Sensors Applications Symposium (SAS); 2017; pp. - Publisher: IEEE

- **Rajendran, Sreeraj**; Pollin, Sofie; *Adaptive Channel Selection and Sensing based on Reinforcement Learning.* WIC/IEEE SP Symposium on Information Theory and Signal Processing; 2015.

Bibliography

[1] P. Denisowski, "Recognizing and resolving lte/catv interference issues," *White Paper, Rohde and Schwarz*, 2011.

[2] M. Matinmikko, M. Latva-Aho, P. Ahokangas, S. Yrjölä, and T. Koivumäki, "Micro operators to boost local service delivery in 5G," *Wireless Personal Communications*, vol. 95, no. 1, pp. 69–82, 2017.

[3] "Telstra." `https://exchange.telstra.com.au/illegal-mobile-repeaters/`. Accessed: 2018-07-01.

[4] S. Thombre, M. Z. H. Bhuiyan, P. Eliardsson, B. Gabrielsson, M. Pattinson, M. Dumville, D. Fryganiotis, S. Hill, V. Manikundalam, M. Pölöskey, *et al.*, "GNSS threat monitoring and reporting: Past, present, and a proposed future," *The Journal of Navigation*, vol. 71, no. 3, pp. 513–529, 2018.

[5] "STRIKE3, Initial Findings from the STRIKE3 GNSS Interference Monitoring Network." `https://www.gps.gov/governance/advisory/meetings/2018-05/dumville.pdf`. Accessed: 2018-07-01.

[6] BIPT, "Belgisch Instituut voor Postdiensten en Telecommunicatie: Jaarverslag 2017." `http://www.bipt.be/en/operators/bipt/publications/annual-report-2017`. Accessed: 2018-07-01.

[7] P. Donegan, "Mobile network outages & service degradations: A heavy reading survey analysis," 2013.

[8] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic, "On perception and reality in wireless air traffic communication security," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1338–1357, June 2017.

[9] M. Lauridsen, B. Vejlgaard, I. Z. Kovacs, H. Nguyen, and P. Mogensen, "Interference measurements in the European 868 MHz ISM band with focus on LoRa and SigFox," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2017.

[10] S. Smith, "The LPWAN market is expected to grow US$ 28746.5 Mn by 2025 from US$ 986.0 Mn in 2017 ." `https://www.prnewswire.com`. Accessed: 2018-05-01.

[11] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovács, P. Mogensen, and M. Sorensen, "Interference impact on coverage and capacity for low power wide area IoT networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2017.

[12] A. Chakraborty and S. R. Das, "Measurement-augmented spectrum databases for white space spectrum," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 67–74, ACM, 2014.

[13] S. Kasera, N. Patwari, and J. Phillips, "NSF project #1564287: Detecting and localizing spectrum offenders using crowdsourcing," August 2016.

[14] S. Das, H. Gupta, P. Milder, and P. Djuric, "NSF project #1642965: Specsense: Bringing spectrum sensing to the masses," October 2016.

[15] S. Roy, K. Shin, A. Ashok, M. McHenry, G. Vigil, S. Kannam, and D. Aragon, "Cityscape: A metro-area spectrum observatory," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, July 2017.

[16] N. Kleber, A. Termos, G. Martinez, J. Merritt, B. Hochwald, J. Chisum, A. Striegel, and J. N. Laneman, "Radiohound: A pervasive sensing platform for sub-6 ghz dynamic spectrum monitoring," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, March 2017.

[17] S. E. Hawkins Iii, E. H. Darlington, A. F. Cheng, and J. R. Hayes, "A new compression algorithm for spectral and time-series data," *Acta Astronautica*, vol. 52, no. 2, pp. 487–492, 2003.

[18] Y. Li, Z. Gao, L. Huang, Z. Tang, and X. Du, "A wideband spectrum data segment compression algorithm in cognitive radio networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, March 2017.

[19] E. Hossain and V. K. Bhargava, *Cognitive wireless communication networks*. Springer Science & Business Media, 2007.

[20] L. Doyle, *Essentials of cognitive radio*. Cambridge University Press, 2009.

[21] W. A. Gardner, "Signal interception: a unifying theoretical framework for feature detection," *IEEE Transactions on Communications*, vol. 36, pp. 897–906, Aug 1988.

[22] W. Gardner, "Spectral correlation of modulated signals: Part i - analog modulation," *IEEE Transactions on Communications*, vol. 35, pp. 584–594, June 1987.

[23] W. Gardner, W. Brown, and C.-K. Chen, "Spectral correlation of modulated signals: Part ii - digital modulation," *IEEE Transactions on Communications*, vol. 35, pp. 595–601, June 1987.

[24] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, pp. 34–40, Aug 2008.

[25] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302 – 1325, 2011. Advanced Topics in Cloud Computing.

[26] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448 – 3470, 2007.

[27] S. Liu, Y. Chen, W. Trappe, and L. J. Greenstein, "ALDO: An anomaly detection framework for dynamic spectrum access networks," in *IEEE INFOCOM 2009*, pp. 675–683, April 2009.

[28] W. Honghao, J. Yunfeng, and W. Lei, "Spectrum anomalies autonomous detection in cognitive radio using hidden markov models," in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 388–392, Dec 2015.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. Insight.

[30] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, 2015.

[31] T. J. O'Shea, 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, pp. 168–179, Feb 2018.

[32] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 563–575, Dec 2017.

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

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

[35] T. Pimentel, M. Monteiro, J. Viana, A. Veloso, and N. Ziviani, "A generalized active learning approach for unsupervised anomaly detection," *arXiv preprint arXiv:1805.09411*, 2018.

[36] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *arXiv preprint arXiv:1811.03402*, 2018.

[37] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big data technologies: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[38] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. V. den Bergh, H. Cordobes, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and big spectrum data," *IEEE Communications Magazine*, vol. 56, pp. 210–217, Jan 2018.

[39] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "SAIFE: Unsupervised wireless spectrum anomaly detection with interpretable features," *IEEE International Symposium on Dynamic Spectrum Access Networks*, 2018.

[40] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, pp. 433–445, Sep. 2018.

[41] S. Rajendran, B. V. den Bergh, T. Vermeulen, and S. Pollin, "IEEE 5G Spectrum Sharing Challenge: A Practical Evaluation of Learning and Feedback," *IEEE Communications Magazine*, vol. 54, pp. 210–216, November 2016.

[42] F. Wunsch, F. Paisana, S. Rajendran, A. Selim, P. Alvarez, S. Müller, S. Koslowski, B. Van den Bergh, and S. Pollin, "DySPAN spectrum challenge: Situational awareness and opportunistic spectrum access benchmarked," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 550–562, 2017.

[43] R. Calvo-Palomino, D. Giustiniano, V. Lenders, and A. Fakhreddine, "Crowdsourcing spectrum data decoding," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–4, May 2017.

[44] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring.," in *SenSys*, pp. 96–108, 2016.

[45] D. Pfammatter, D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, (New York, NY, USA), pp. 71–82, ACM, 2015.

[46] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[47] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Manning Publications Co., April 2015.

[48] J. Kreps, N. Narkhede, J. Rao, *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, pp. 1–7, 2011.

[49] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pp. 1–10, IEEE, 2010.

[50] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2–2, USENIX Association, 2012.

[51] M. Kornacker, A. Behm, V. Bittorf, and et al., "Impala: A Modern, Open-Source SQL Engine for Hadoop," in *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research*, 2015.

[52] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

[53] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE communications surveys & tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[54] J. van de Beek, J. Riihijarvi, A. Achtzehn, and P. Mahonen, "TV white space in Europe," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 178–188, Feb 2012.

[55] N. E. West and T. J. O'Shea, "Deep architectures for modulation recognition," *CoRR*, vol. abs/1703.09197, 2017.

[56] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2662–2686, Fourthquarter 2016.

[57] R. Calvo-Palomino, H. Cordobés, M. Engel, M. Fuchs, P. Jain, M. Liechti, S. Rajendran, M. Schäfer, B. V. d. Bergh, S. Pollin, *et al.*, "Electrosense+: Empowering people to decode the radio spectrum," *arXiv preprint arXiv:1811.12265*, 2018.

[58] T. J. O'Shea, T. C. Clancy, and R. W. McGwier, "Recurrent neural radio anomaly detection," *arXiv preprint arXiv:1611.00301*, 2016.

[59] N. Tandiya, A. Jauhar, V. Marojevic, and J. H. Reed, "Deep predictive coding neural network for RF anomaly detection in wireless networks," *arXiv preprint arXiv:1803.06054*, 2018.

[60] Q. Feng, Y. Zhang, C. Li, Z. Dou, and J. Wang, "Anomaly detection of spectrum in wireless communication via deep auto-encoders," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3161–3178, 2017.

[61] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *ArXiv e-prints*, Dec. 2013.

[62] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[64] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," *ArXiv e-prints*, June 2016.

[65] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[66] "GNU Radio website." `http://www.gnuradio.org`. Accessed: 2018-05-01.

[67] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," *ArXiv e-prints*, Sept. 2016.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[69] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.

[70] T. Pevnỳ, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.

[71] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

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

[73] M. Zheleva, R. Chandra, A. Chowdhery, A. Kapoor, and P. Garnett, "TxMiner: Identifying transmitters in real-world spectrum measurements," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 94–105, Sept 2015.

[74] S. S. Hong and S. R. Katti, "DOF: A local wireless information plane," in *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, (New York, NY, USA), pp. 230–241, ACM, 2011.

[75] 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, pp. 137–156, April 2007.

[76] Z. Yu, *Automatic modulation classification of communication signals*. PhD thesis, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, 2006.

[77] A. V. Dandawate and G. B. Giannakis, "Statistical tests for presence of cyclostationarity," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2355–2369, Sep 1994.

[78] A. Fehske, J. Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, pp. 144–150, Nov 2005.

[79] T. Yucek and H. Arslan, "OFDM signal identification and transmission parameter estimation for cognitive radio applications," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, pp. 4056–4060, Nov 2007.

[80] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*, pp. 213–226, Springer, 2016.

[81] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[82] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proceedings of the GNU Radio Conference*, vol. 1, 2016.

[83] A. Karpathy, J. Johnson, and F. Li, "Visualizing and understanding recurrent networks," *CoRR*, vol. abs/1506.02078, 2015.

[84] T. J. O'Shea, N. West, M. Vondal, and T. C. Clancy, "Semi-supervised radio signal identification," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 33–38, Feb 2017.

[85] Y. Gwon, S. Dastangoo, H. T. Kung, and C. Fossa, "Blind signal classification via sparse coding," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2016.

[86] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[87] A. Spurr, E. Aksan, and O. Hilliges, "Guiding InfoGAN with Semi-Supervision," *ArXiv e-prints*, July 2017.

[88] D. Divsalar and M. Simon, "The power spectral density of digital modulations transmitted over nonlinear channels," *IEEE Transactions on Communications*, vol. 30, pp. 142–151, January 1982.

[89] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[90] A. Testolin, M. Zanforlin, M. D. F. D. Grazia, D. Munaretto, A. Zanella, M. Zorzi, and M. Zorzi, "A machine learning approach to qoe-based video admission control and resource allocation in wireless systems," in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pp. 31–38, June 2014.

[91] "Viavi." `https://www.rcrwireless.com/20161101/network-infrastructure/interference-hunting-fundamental-quality-service`. Accessed: 2018-07-01.

[92] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, pp. 93–104, ACM, 2000.

[93] S. Das, W. K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 853–858, Dec 2016.

[94] S. Das, W.-K. Wong, A. Fern, T. G. Dietterich, and M. A. Siddiqui, "Incorporating feedback into tree-based anomaly detection," *arXiv preprint arXiv:1708.09441*, 2017.

[95] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *Journal of Artificial Intelligence Research*, vol. 46, pp. 235–262, 2013.

[96] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Bäumer, and J. Davis, "Semi-supervised anomaly detection with an application to water analytics," in *Proceedings/IEEE International Conference on Data Mining.*, IEEE, 2018.

[97] T. Van Craenendonck, S. Dumančić, E. Van Wolputte, and H. Blockeel, "Cobras: Interactive clustering with pairwise queries," in *International Symposium on Intelligent Data Analysis*, pp. 353–366, Springer, 2018.

[98] A. Struyf, M. Hubert, P. Rousseeuw, *et al.*, "Clustering in an object-oriented environment," *Journal of Statistical Software*, vol. 1, pp. 1–30, 1997.

[99] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[100] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, ACM, 2008.

[101] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer, "A GNU radio-based IEEE 802.15.4 testbed," in *Proceedings of 12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)*, pp. 37–40, 2013.

[102] A. Kaushik, F. Wunsch, A. Sagainov, N. Cuervo, J. Demel, S. Koslowski, H. Jäkel, and F. Jondral, "Spectrum sharing for 5G wireless systems (spectrum sharing challenge)," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, IEEE, 2015.

[103] J. Tallon, C. Blumm, A. Puschmann, F. Paisana, J. van de Belt, P. Di Francesco, N. Kaminski, and H. Ahmadi, "Coexistence through adaptive sensing and markov chains," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 7–8, IEEE, 2015.

[104] D. Ntaikos, K. Ntougias, B. Gizas, F. Verdou, and C. B. P. Athens, "Low-complexity air-interface-agnostic cooperative parasitic multi-antenna spectrum sharing system," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 5–6, IEEE, 2015.

[105] S. Papadakis, M. Surligas, A. Makrogiannakis, K. Triantafyllakis, G. Vardakis, A. Tzougarakis, and G. Floros, "Robust spectrum sharing through virtualization," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 9–10, IEEE, 2015.

[106] F. Harris, R. Bell, and V. K. Adsumilli, "Spectrum sharing between a ZigBee frequency hopper and an FSK modem," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 3–4, IEEE, 2015.

[107] S. Geirhofer, L. Tong, and B. M. Sadler, "A measurement-based model for dynamic spectrum access in WLAN channels," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, pp. 1–7, Oct 2006.

[108] S. Geirhofer, L. Tong, and B. M. Sadler, "Cognitive radios for dynamic spectrum access - dynamic spectrum access in the time domain: Modeling and exploiting white space," *IEEE Communications Magazine*, vol. 45, pp. 66–72, May 2007.

[109] M. López-Benítez and F. Casadevall, "Spectrum usage models for the analysis, design and simulation of cognitive radio networks," in *Cognitive radio and its application for next generation cellular and wireless networks*, pp. 27–73, Springer, 2012.

[110] L. Stabellini, "Quantifying and modeling spectrum opportunities in a real wireless environment," in *2010 IEEE Wireless Communication and Networking Conference*, pp. 1–6, April 2010.

[111] F. Paisana, A. Selim, M. Kist, P. Alvarez, J. Tallon, C. Bluemm, A. Puschmann, and L. DaSilva, "Context-aware cognitive radio using deep learning," in *DySPAN*, 2017.

[112] J. Tallon, C. Blümm, A. Puschmann, F. Paisana, J. van de Belt, P. Di Francesco, N. J. Kaminski, and H. Ahmadi, "Coexistence through adaptive sensing and markov chains.," in *DySPAN*, pp. 7–8, 2015.

[113] F. Wunsch, S. Koslowski, S. Müller, N. Cuervo, H. Jäkel, and F. K. Jondral, "A Cognitive Overlay System Based on FBMC," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 171–172, Mar. 2017.

[114] A. Kaushik, F. Wunsch, A. Sagainov, N. Cuervo, J. Demel, S. Koslowski, H. Jäkel, and F. Jondral, "Spectrum sharing for 5G wireless systems (Spectrum sharing challenge)," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, Sept 2015.

[115] T. W. Rondeau, "On the GNU Radio Ecosystem," in *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, Wiley, 2015.

[116] C. M. Spooner, A. N. Mody, J. Chuang, and M. P. Anthony, "Tunnelized cyclostationary signal processing: A novel approach to low-energy spectrum sensing," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pp. 811–816, Nov 2013.



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
TELEMIC
Kasteelpark Arenberg 10 - box 2444
B-3001 Leuven
sreeraj.rajendran@esat.kuleuven.be
https://www.esat.kuleuven.be/telemic