UNIVERSITÉ PIERRE ET MARIE CURIE
INSTITUTE MINES-TELECOM/
TÉLÉCOM SUDPARIS

Discipline: Réseaux et Services de Télćommunication

Doctoral School: EDITE de Paris

Presented by

Seif Eddine Hammami

Subject:

Dynamic Network Resources Optimization based on Machine Learning and Cellular Data Mining

Doctoral Committee:

Prof. Hacene Fouchal	Reviewer	URCA (France)
Dr. Mathieu Bouet	Reviewer	Thales (France)
Prof. Houda Labiod	Examiner	Telecom ParisTech
Dr. Yvon Gourhant	Examiner	Orange Labs
Dr. Hassine Moungla	Examiner	University of Paris Descartes
Prof. Housam Afifi	Supervisor	Telecom SudParis

September 20th, 2018

Thesis n° 2018$TELE$0015

Abstract

Mobile phone datasets are the central keystone of the thesis, where we propose new approaches for studying networking problems. We are using, in this thesis, those real dynamic data rather than the old conventional approaches based on simulations and random inputs. Most of these datasets consist of Call Data Records (CDR) metadata, i.e. a time-stamped dataset of all interactions between the subscribers of a mobile operator and the network infrastructure during a given period. Given their large size and the fact that these are real-world datasets, information extracted from these datasets have intensively been used in our work to develop new algorithms that aim to revolutionize the infrastructure management mechanisms and optimize the usage of resource. CDR metadata contains also, in addition to temporal information, other information about the geographic scale of subscribers network usage. Combining the temporal and geographical information certainly helps to infer the spatio-temporal dynamics of subscribers use of the network resource as well as the dynamic patterns of the base-station throughout the day.

Therefore, the goal of this thesis is to provide a general process for analyzing cellular network CDR datasets and provide tools and frameworks based on these datasets to optimize and enhance networks performances.

First, we start by analyzing real CDRs dataset and infer the most relevant profiles of bandwidth consumption. Then we propose an automatic tool that provide an on-line and daily classification of the access network base stations. We exploit then this classification to estimate the amount users are moving from a location to another in order to study the bandwidth mobility across the cellular access network. We continue then mining the network traces and we analyze the periodicity features of base stations load. Based on that, we propose an on-line tool to predict the future load of each cell. Building upon the proposed tools, we develop a global framework which combine them and exploit their results in order to optimize the planning of wireless networks. The framework is then validated on a wireless mesh topology.

Cellular networks may suffer from some unexpected bandwidth demands that may drastically drop their performance. They may even suffer from unusual decrease of bandwidth consumption that need to be explained. In this context, we propose a framework that detect proactively network anomalies. Then, we propose another framework that aim manage and optimize instantly drone-cells based networks. The purpose of these drone-cells is to support the existent macro-cells at rush hours or during mass event when unexpected and massive bandwidth demands occur.

RÉSUMÉ

Les traces réelles des réseaux cellulaire est la clé de voute de ma thèse de doctorat. En effet, je propose dans cette thèse des nouvelles approches dans l'étude et l'analyse des problématiques des réseaux de télécommunications en utilisant ces traces réelles contrairement aux approches classiques basées sur des jeux de données simulés ou générées par des processus aléatoires. Ces traces cellulaires sont présentes sous la forme de jeux de données de CDR (Call Detail Records ou statistiques d'appels) représentés par des information horodatées sur chaque interaction de l'abonné avec l'infrastructure des réseaux mobile, quelques soient les appels reçus/émis, des SMS ou des sessions d'internet. Vu leur richesse et le fait qu'ils reflètent des cas d'usages réels, les informations massives qui peuvent tre extraites et analysées de ces jeux donnés, ont été exploités intensivement dans mes travaux de thèse pour développer de nouveaux algorithmes qui ont pour but de changer litéralement les mécanismes de gestion et d'optimisation dans le cadre de l'usage des ressources réseaux. Outre les informations temporelles, les CDRs contiennent aussi les informations géographiques qui projettent l'emplacement instantané de labonnées durant ses interactions. En combinant les échelles temporelles et géographiques, nous pouvons déduire les dynamicités spatio-temporelle de l'usage réseaux de chaque abonnée ainsi que les modèles dynamiques de l'utilisation de la bande passante sur les stations de bases. Les jeux de données des CDR sont généralement des données brutes et qui nécessitent des outils avancés d'analyse de données et d'intelligence artificielle afin d'extraire les informations les plus importantes. Dans ce contexte, on propose dans cette thèse une étude structurée pour analyser des traces réelles de CDRs réels comme les traces du D4D challenge contenant les données du réseau cellulaire dOrange Sénégal et les traces du Big Data challenge fournis par l'opérateur Telecom Italia. Notre méthode consiste, en premier lieu, à regrouper intelligemment les séries temporelles journalières de charge sur les stations de bases dans des classes pertinentes. Nous proposons pour ça d'utiliser un algorithme modifié de K-means basé sur la distance DTW (Dynamic Time Warping) qui a été montré plus performante que la distance euclidienne classique.

Cet algorithme, nous a permis, de classer les series temporelles de charge pour chaque station de base dans trois classes principales. Une première classe pour les profils de Pic de charge matinale , une classe pour les profils de Charge constante et une dernière classe pour les Pic de charge nocturne . Cette première classification, nous permet de proposer notre algorithme de classification automatique et massive des profiles journalières des stations de bases basé sur la machine dapprentissage SVM (Support Vector Machine). Cette classification automatique est importante pour les opérateurs de réseaux et peut leur servir à adapter l'allocation de ressource radio selon ces profiles.

Afin de garantir la continuité du service pour les abonnées, il est important d'estimer avec précision la dynamicité de la bande passante sa migration instantanée entre les différents endroits dans le futur. Ceci revient à étudier les déplacements des abonnées, qui reflètent aussi un potentiel déplacement de demande de bande passante, entre les zones classifiées précédemment. On propose pour cet objectif, une nouvelle forme de matrice Origine-Destination basée sur les résultats de classification, qui nous permet d'estimer les futurs taux de déplacement de la demande de bande passante entre les classes de zones. En d'autres termes, elle projette la mobilité de bande passante durant la journée.

Le deuxième chapitre de cette thèse répond à une question importante : Est-t-il possible dexploiter les traces de CDRs pour implémenter des algorithmes capables de prédire avec précision les futurs taux de charge sur chaque station de base? Dans la continuité du premier chapitre, nous abordons cette problématique en proposant une étude pour les caractéristiques des séries temporelles de charge journalière et en implémentant un modèle de prédiction basé sur l'algorithme d'apprentissage SVR (Support Vector Regression). Nous fournissons une comparaison des performances avec d'autres algorithmes de prédictions connus qui montrent l'efficacité de notre modèle. Nous intégrons par la suite les modèles que nous avons proposé dans un outil flexible qui permet l'optimisation dynamique des ressource réseaux basé sur les traces réelles. Nous évaluons notre solution en l'appliquant sur une architecture basée sur un réseau sans fil mesh proposé dans le projet national LCI4D. l'optimisation de ce réseau est faite par un algorithme qui exploite les résultats des modules danalyse de données.

Une deuxième évaluation pour notre outil est proposée et qui consiste à l'appliquer sur une topologie dynamique basé sur des cellules-drones (des drones embarquant des femto-cells). Nous proposons pour ça un algorithme d'apprentissage renforcé multi-agent qui exploite aussi les résultats des modules danalyse de données pour optimiser dynamiquement et en temps réel cette topologie.

Dans la continuité du contexte d'analyse des traces réelles de CDRs, nous proposons dans un dernier chapitre, un deuxième outil qui sera capable de détecter proactivement les anomalies dans les réseaux cellulaire qui peuvent se produire suite à un pic de consommation brusque ou une chute due à des problèmes techniques. Cet outil est basé sur les algorithmes OCSVM (One-class SVM) et SVR qui permettent de distinguer en temps réel les profile de charge anormale. L'outil est testé en utilisant les traces du D4D challenge et Big challenge et en le comparant à dautre technique de détection d'anomalies et les résultats montrent qu'il est plus efficace. Nous validons aussi le modèle pour analyser l'impact des données proliférantes issues des nouvelles applications comme celle de le-santé. Notre modèle est capable de détecter les anomalies due à linjection de ces nouvelles sources de données et qui impactent évidement l'usage normal des réseaux cellulaire.

Thesis Publications

International Journals

- **HAMMAMI, Seif eddine** and AFIFI, Hossam. Fault-tolerant dynamic planning for wireless mesh networks based on real load profiles. Computer Networks, 2017, vol. 128, p. 94-107.
- **HAMMAMI, Seif eddine** et AFIFI, Hossam and MAROT, Michel. Mining Call Detail Records for Bandwidth Classification and Prediction, submitted to Performance Evaluation journal.
- Aziza Ben-Mosbah, **Seif Eddine Hammami**, Hassine Moungla, Hossam Afifi, Richard Rouil, and Ahmed E. Kamal, Enhancing Device-to-Device Direct Discovery Based on Predicted User Density Patterns, submitted to Computer Networks, The International Journal of Computer and Telecommunications Networking.

International Conferences

- **HAMMAMI, Seif Eddine**, AFIFI, Hossam, MAROT, Michel, et al. Network planning tool based on network classification and load prediction. In Wireless Communications and Networking Conference (WCNC), 2016 IEEE. IEEE, 2016. p. 1-6.
- **HAMMAMI, Seif Eddine**, MOUNGLA, Hassine and AFIFI, Hossam. Proactive Anomaly Detection Model for eHealth-enabled Data in Next Generation Cellular Networks. In International Conference on Communication (ICC), IEEE, 2018.
- ZHANG, Jun, LABIOD, Houda, **HAMMAMI, Seif Eddine**, et al. Scalable energy efficient routing in multi-layer femtocell networks. In Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International. IEEE, 2017.p.1540-1545.

Acknowledgement

This dissertation would not have been possible without the help of so many people in so many ways.

I would like to express my gratitude towards my thesis director Prof. Hossam Afifi for his valuable guidance and supervision. I am deeply grateful as well to Prof. Michel Marot and Dr. Vincent Gauthier for all their help and support during my PhD.

I would like to thank the members of the dissertation jury for accepting my invitation and honoring me with their presence. I particularly thank Prof. Hacene Fouchal and Prof. Mathier Bouet for reviewing this dissertation and providing me with their insightful feedbacks. I also thank this works examiners Prof. Houda Labiod, Dr. Yvon Gourhant and Dr. Hassine Moungla for their time and flexibility. Many thanks go to the staff of Telecom SudParis and EDITE de Paris, particularly Mrs. Ydalia Garcia, Mrs. Sandra Gchweinder and Mrs. Veronique Guy for their assistance and patience concerning the administrative procedures.

I am also indebted to all my friends and my colleagues in the Networks and Telecommunication Services department and SAMOVAR lab for their unlimited support and encouragement.

Last but not least, I would like to thank my beloved parents and my dearest brother and sister for being always there for me and for their unlimited understanding and unconditioned love. I would like also to thank my extended family and friends for their support.

Contents

List of Figures

List of Tables

Acronyms

3GPP Third Generation Partnership Project.

AODV Ad-hoc On Demand Distance Vector.

ARIMA Auto-regressive and Integrated Moving Average.

ARMA Auto-Regressive and Moving Average.

BS Base Station.

CDR Call Detail Record.

CR Cognitive Radio.

D4D Data For Development.

DSA Dynamic Spectrum Access.

DTW Dynamic Time Warping.

ED Euclidean Distance.

ERM Empirical Risk Minimization.

FCC Federal Communications Commission.

FFT Fast Fourier Transform.

GSM Global System for Mobile Communications.

ICT Information Communication Technology.

IoT Internet of Things.

ITU International Telecommunication Union.

KNN K-Nearest Neighbor.

KPI Key Performance Indicator.

LTE Long Term Evolution.

MARL Multi-Agent Reinforcement Learning.

MILP Multi-Integer Linear Programming.

MIP Mixed Integer Programming.

MSE Mean squared Error.

MVNO Mobile Virtual Network Operators.

OCSVM One-Class Support Vector Machine.

ODM Origin-Destination Matrix.

OEM original equipment manufacturers.

OLSR Optimized Link State Routing protocol.

OPEX Operational Expenses.

OSS Operations Support Systems.

PCA Principal Component Analysis.

PCC Pearson Correlation Coefficient.

PDA Personal Digital Assistants.

PSD Power Spectral Density.

PU Primary User.

QoE Quality of Experience.

QoS Quality of Service.

RB Resource Block.

RBF Radial Basis Function.

SAS Spectrum Access System.

SINR Signal to Interference plus Noise Ratio.

SMS Short Messages exchanges.

SRM Structural Risk Minimization.

SU Secondary User.

SVM Support Vector Machine.

SVR Support Vector Regression.

TOI Time of Interest.

TVWS TV White Space.

UAV Unmanned Aerial Vehicles.

UE User equipement.

UMTS Universal Mobile Telecommunications System.

WMN Wireless Mesh Networks.

Chapter 1

Introduction

1.1 Network optimization challenges

1.1.1 Mobile users' behavior

Mobile internet usage has become in the forefront of the daily life habits of smartphone and tablet users. It allows subscribers to access a infinite source of information and share with their networks. Some studies suggests that the global mobile traffic will increase by seven-folds by 2021 with respect to 2016. The world mobile population counts 3.7 billion unique users at the beginning of 2018 according to Statista' statistics [1]. Other studies reveal that mobile subscribers spend on average 69% of their media time on smartphones. Another study [2] estimates that 68% of the global Internet usage is driven by mobile devices and it predicts that the average will reach 80% by 2018.

These statistics represents a direct impact of the non-stopping development of mobile devices on the rapid development of network architectures that boost the subscribers penetration averages. Moreover, the Internet consumption pattern of subscribers is continuously changing to follow the expansion of networks and mobile devices. For example, nowadays subscribers can use the smartphone everywhere and at any time, while standing in a bus, walking in the street or laying in a bed.

As the development of mobile technology affects the on users behavior, it also has an impact on network resource consumption. In fact, user mobility and various data consumption behaviors induce several load patterns and network fluctuation. Therefore, network operators should stand up for these challenges to deal with user consumption dynamics. They need to abandon the classical techniques such as statically sharing the licensed spectrum, which cannot efficiently manage instantaneous users demand dynamics. New techniques and radio resource allocation mechanisms must be designed to dynamically manage user demands and provide better Quality of Service (QoS). Information about user consumption behaviors and network load fluctuation trends can be extracted from mobile network datasets like Call Detail Record (CDR). Thus, the first step toward enhancing network performance is to analyze the massive information included within CDR datasets

1.1.2 network planning and resource allocation

Another challenge in today's network optimization is the adaptation of the wireless traffic rates in new mobile networks, so that such a way an efficient resource allocation is guaranteed. The growth of data demands and the increase in wireless traffic rates in new mobile networks needs intelligent and dynamic technologies for telecommunication management. Recent studies predicts that the new generation cellular standards (like 5G) will rely much more heavily on a dense and less power consuming networks to serve dynamically requested data rates to the user [3].

In this thesis, we evaluate the network planning and resource allocation via the following types of wireless networks topology:

Wireless mesh networks

Wireless Mesh Networks (WMN) [4] are an easy and low cost alternative for network operators to provide a high speed connectivity access to some areas where installing a cellular access network is difficult. Thus, WMN can extend cellular network femtocell-based coverage areas with a backhaul internet dorsal and then satisfy more users anywhere. The state-of-the-art in WMN architectures focuses on enhancing the performance of wireless mesh networks either by reducing the interference between mesh nodes [5, 6, 7] or by addressing routing issues [8, 9]. However, it is difficult to find WMN contributions explaining how such systems can be efficiently integrated into a real cellular architecture (4G/5G). In this thesis, one part of our contribution is to propose an innovative architecture that combines WMN with cellular femto-cells (3GPP based) to deploy cellular services for small Mobile Virtual Network Operators (MVNO) in developing countries or for special periodic mass events.

Drone-cells networks

Small cells mounted on Unmanned Aerial Vehicles (UAV) [10] or drones (we call them drone-cells hereafter) are proposed, as an alternative to fixed femto-cells, to support existing macro-cell infrastructure. The deployment of these mobile small cells consists on move these small cells toward a target positions (usually within the range of the macro-cell that support) based on the decision made by a mobile network operator. The drone-cells movement toward adequate positions must be correlated to the amount of data requested, i.e. drone-cells should support overloaded cells. Hence, an intelligent entity may be added to the network in order to monitor instantly the network state and find the optimal decision to control drone-cells.

1.2 Thesis motivations

1.2.1 CDR: A mine of data

A Global Digital Forensics report [11] has predicted that smartphones, tablets and social media will deeply impact human beings business and personal network usage in the near future. It is a key objective to make us updated with the worldwide news, keep interaction with others, procure valuable knowledge and valuable information. With the fast growth of the network contents usage, we are able to analyze tremendous amounts of data and predict customer favourites and future demands.

Moreover, network planning is crucial for network operators to enhance services that are cost-effective and QoS-dependent. Tools allowing the inference of instantaneous bandwidth demand, permit network operators to dynamically manage the bandwidth and to implement innovative techniques for radio resource sharing between BSs. In this thesis, we aim to exploit CDR and data analysis techniques in networking context. More precisely, the goal of this research is to apply data mining on real CDR datasets to assist operators in optimizing the network resource consumption. This prevents both under-provisioning and over-provisioning. We use data from the Data For Development (D4D) challenge [12] provided by Orange Senegal, restricted to the set of BS in the Dakar city.

Before rethinking cellular network deployment, operators must begin by analyzing the existing architecture and optimizing the huge number of data flows. CDR data-sets include massive knowledge and information that can be very useful for this step. We claim that current operators policies for radio resource allocation and management can be improved. As a matter of fact, classical resource allocation techniques are founded based on a static sharing of the licensed spectrum, so that the access network cannot efficiently manage instantaneous dynamics of user demands, i.e. when the number of users quickly increases, access network starts to reject some of the users and thus, the quality of service decreases. Therefore, we can make the hypothesis that the classical allocation techniques have some waste of the radio resource as some BS may not use all the resources because of the low number of its attached users, while at the same time, other BSs are over populated. In such scenario, extra resources are needed to satisfy all the users without affecting their Quality of Service (QoS).

1.2.2 Network profiling and classification

In this thesis, we start with characterizing the daily bandwidth consumption for each BS and we propose an on-the-fly algorithm for BS load classification based on Support Vector Machine (SVM). First, in order to have reference classes for our SVM classification, we infer typical BS load classes from a modified K-means algorithm. The obtained profile classes are then used to run an SVM-based algorithm which allows assigning on-the-fly BSs to different classes with high accuracy.

Although, studies such as [13, 14, 15] have analyzed the daily cell load patterns and classified them according to users activities or land-use. Most of them have applied K-means tools with Euclidean Distance (ED) as similarity measurement to cluster load time-series. However, study [16] showed that Dynamic Time Warping (DTW) is more efficient than ED for measuring time-series similarities due to its insensitivity to time distortion. Therefore, in this thesis we propose a modified K-means technique based on DTW distance to detect BS load profile. Besides, clustering load profiles is not sufficient for operator networks, since a BS class may change throughout a period of days. In such scenario, an automated classification technique is needed to adjust the on-the-fly classification by taking into consideration the most recent history. To address the affirmed issues, in this thesis we propose a classification method based on SVM to classify the large scale dataset of BS load profile.

The base station profile classification algorithm constitutes an important step towards analyzing the spatio-spatial bandwidth consumption patterns and also dynamically optimizing the overall network resource allocation. By classifying the BS load profile, we are inferring the data request trends and thus, we can even infer the semantics areas, i.e. whether residential, business, night-life areas, etc. The classification also allows to customize their offer plans according to the bandwidth consumption profiles. In fact, for business areas, the network operator can propose some offer plans with more emailing, browsing, chatting services. For residential area customers, the network operator may propose offer plans with higher mobile data for streaming, video gaming services, etc.

1.2.3 Network load prediction and anomaly detection

Network load prediction

We claim that current operators policies for radio resource allocation and management are no more efficient. The classical technique for allocation is based on a static sharing of the licensed spectrum. Therefore, with the help of our field classifications, we can say that this allocation seems to be a waste of the radio resources as some BS may not consume all the allocated resource in case of under-load. In the same time, there can be some BS where the number of users is too high and then it demands extra resources to satisfy all the users without affecting the quality of service. Dynamic Techniques for radio resource allocation must substitute the classic ones, so that we can exploit dynamically the unused resources on a given period of time on another places and allocate them to overloaded BS at that moment.

Many techniques can helps operators and OEM (original equipment manufacturers) to change their resource allocation policies as well as an on-line load prediction to dynamically optimize network resources. By predicting the future capacity, we can anticipate the class for which the BS will stand. By this way, we dynamically decide how much resource to allocate and when. In our study, we propose an algorithm based on Support Vector Regression (SVR) to predict the load of BS according its history.

In this context, we propose also in this work a machine learning technique based on SVR to predict the instant load of each BS. The combination of network classification and load prediction tools allows network operators to monitor their network resources and to provide new mechanisms that take into account the dynamics of user demands and adapt the network resource allocation to enhance the QoS and Quality of Experience (QoE). Also, load prediction provides a proactive mechanism to complete bandwidth sharing between areas and to dynamically allocate bandwidth for new attaching users.

Network anomaly detection

On July 2012, a general system failure occurred in the network of Orange Telecom, the historic national French operator. The breakdown was very severe and most of Orange subscribers (almost 26 million subscribers) were out of service for 9 hours. They were unable to make calls, texts or to use data services. The failure has also affected the Orange MVNOs and the interconnection with other Network operators. Due to the lack of adequate management tools that can rapidly detect this kind of anomaly, the operator was unable to avoid the blackout in its network or even reduce its impact. In addition, it was forced to deploy more resources to fix the failure which added extra cost in addition to its subscribers dissatisfaction. Investigations showed that an earlier update of a software stack was the origin of this blackout and the anomaly has not been notified by any alarm signal (Probably it was identified as a true positive alarm). From this incident and many others, it appears that there is a need to upgrade management and alarm systems with efficient automated techniques that, by the analysis of real-time traces, is able to detect on the fly network anomalies. These tools can also help the operators to monitor their infrastructures and more accurately manage their networks. Strong by their learning capabilities, they avoid the long and fastidious hand work to build evolving traffic profiles.

In fact, network outliers detection techniques aim to automatically identify and detect abnormal and anomalous patterns which differ from a normal behavior or may present a local deviation from the normal data. Next-generation cellular systems and cognitive networks aim to introduce more flexible techniques to better react to these dynamics. On the other hand, a major issue for network operators is to handle and detect sudden and local anomalous behavior within the network, whether it is a sharp peak of users demands (occurred during mass events for example), an abnormal brief decrease or even an non-common daily data consumption patterns. If the first anomaly type needs a fast reaction to guarantee network resilience and service survivability and hence, avoiding users rejections, the second one may be due to some technical malfunction of the network infrastructure that need an instant maintenance. These anomalies are also time-dependent and need not only geographic identification but also temporal detection of the time interval in which they occur with high precision.

In our contribution, we address the problem of detecting outliers within radio access network. In this thesis, we propose a general outliers detection framework for radio access networks and we evaluate our proposal using real datasets CDRs. The framework is based on the combination of One-Class Support Vector Machine (OCSVM) and SVR algorithms. The OCSVM algorithm, is able to detect the cell presenting an anomaly, thus its geographical location. While the SVR prediction algorithm is used to provide more precision to the framework and allow detecting the time-interval in which occurred the network anomaly.

Merging the two algorithms allows to implement more accurate framework for network anomaly detection, since we need to detect outliers in a spatio-temporal context. In fact, the OCSVM is insensitive to the temporal scale, but it allow to optimize the processing by detecting the anomalous BS activity within a large scale dataset. Thus the temporal anomaly detection is only performed on the detected set of anomalous activity instead of the entire large set. Moreover, using the SVR based prediction for the temporal detection avoid employing the OCSVM with a window-based approach [17] for anomaly detection for time-series. So it permits to save the computational resource since the window-based techniques divide the time-series into many sub-sequences to be processed. In addition, the choice of the window size and the manner to split the time-series is complex.

1.3 Contributions

This thesis constitutes a flexible toolbox that consists on a set of frameworks for mining and analyzing mobile network datasets. The main topic of the thesis addresses the analysis and mining of real cellular network CDR datasets. It also tackles the network optimization and the resource allocation in wireless networks.

First, we provide in this thesis a data mining study of a real-world CDRs dataset such as D4D challenge dataset provided by Orange Senegal and the big data challenge dataset provided by Telecom Italia. Our analysis method consists on clustering the BS daily load time-series into relevant classes. We use for that a modified k-means clustering algorithm based on the DTW distance. This clustering results in dividing the BS load time-series, extracted from the D4D challenge dataset, into three relevant classes. Each class belong to a specific BS load profile, such as a day-peak load profile, Constant load profile and Night-peak load profile. This first analysis phase permits to tag each BS with its corresponding profile class. The profiled data are used then to implement an automatic classification machine learning based on SVM. The classification algorithm allow us to infer automatically, in large scale, the daily class of each BS time-series. This information are important for network operators to propose dynamic algorithms for radio resource allocation that follow the instantaneous load fluctuation. In addition, to enhance the continuity of network services, it is important to estimate with high confidence how the bandwidth demand on a BS is shared among all the base stations in the near future. Hence, we exploit the classification of BS profiles to analyze the mobility of the network bandwidth between areas. We use for this objective a novel form of the origin-destination matrix based on the classification. This classified OD matrix provides aggregate information about the mobility of the load usage. In other words, it projects the mobility of the bandwidth between areas.

Second, we address this issue in this thesis and we provide an analysis of the BS load time-series characteristics. Then, we propose a prediction model based on SVR. Our solution is compared to other prediction techniques and the results prove the high efficiency of the SVR-based prediction model.

Third, we combine the network classification and load prediction algorithms into a global framework that propose a dynamic network resource allocation techniques based on real data analysis. We evaluate the framework by applying it on a wireless mesh network topology in order to optimize its planning. In this part of the thesis, we propose a Multi-Integer Linear Programming (MILP) algorithm that provide a dynamic and fault-tolerant planning for a wireless mesh network that takes as input the cell load time-series resulting from the machine learning tools presented previously.

Fourth, in the continuity of the CDRs dataset analysis and the load prediction, we propose in this thesis a second framework that consists on detecting pro-actively the anomalous load patterns of the network that may occur during mass events or network technical malfunctions. The anomaly detection framework is based on One-class SVM (OCSVM) and SVR algorithms. It is tested and validated with D4D challenge CDR and Italia telecom datasets. Comparison results show that our model outperforms other techniques. The framework is used then to analyze the impact of the proliferous e-health data generated by the medical smart-phone applications.

Finally, we propose a dynamic solution based on drone-cells that exploit real traces of demand profiles, output from the framework, and adapt in real time the deployment of drones-cell according these demands. The framework helps network operator to handle unexpected bandwidth consumption in a mass event context and ensure a faster recovery of network services. In this part, we propose to optimize the deployment using the machine learning paradigm instead of classical linear programming models. Our solution is based on a Multi-Agent Reinforcement Learning (MARL) approach.

1.4 Outline

The remainder of this thesis is organized as follows. Chapter 2 highlights future cellular network approaches such as cognitive radio and resource allocation techniques. It also provides an insight about network data mining. Chapter 3 presents the first CDR dataset used in the thesis and provides a analysis of this dataset. We depict in this chapter the clustering of BS profiles into three pertinent classes and we provide a tool based on SVM to classify the large dataset of BS profiles. We exploit the classification results to also study the bandwidth mobility over the network. In Chapter 4, we propose a prediction tool based on SVR to forecast the future BS load. In chapter 5, a combined use of classification and prediction and classification algorithms is proposed to optimize the planning and management wireless network. An application of the framework is applied on a wireless mesh topology to evaluate its performance. In chapter 6, we propose a second tool that aim to detect network anomalies caused during mass events where unexpected data demand occur or even for unusual drop of network usage. While in chapter 7, we propose a multi-agent reinforcement learning based tool that aim to optimize the dynamic deployment of drone-cells networks in a mass event context. Chapter 6 concludes the thesis and discusses further research opportunities.

Chapter 2

State of the Art

2.1 Wireless networks resource optimization

As wireless devices become democratized and lead to develop a pletora of novel services that require higher QoS characteristics and data rate demands, resource allocation and optimization becomes more challenging problem. Otherwise, the "Mobile Internet" revolutionized the traditional mobile services and provided an unprecedented experience for users. Therefore, network operators are facing a critical problem with the ever-growing expansion of data rate requests for new services. Actual network architecture are suffering from resource limitations of wireless networks that places increased stress on the static radio spectrum allocation techniques. Hence, optimized and dynamic allocation techniques become more essential since its is shown [18] that the spectrum is somehow over-planned (or even under-planned) and it is used sporadically over the day, leading to its under-utilization. The spectrum under-utilization can be highlighted by the spectrum holes where no devices are requesting service in a given area, blank periods (or Resource Block (RB)) when no device is transmitting in the channel or poor utilization due to sparse users density. In the other hand, the mobile network spectrum is limited. Otherwise, it need more intelligent management policies or exploiting extra unused and unlicensed spectrum such as Tv-white space [19].

New techniques are proposed recently that propose to enhance the spectrum usage, such as cognitive radio concept, dynamic spectrum access, etc. Dynamic Spectrum Access (DSA) models [20] consists on optimizing the access competition between devices especially when the channel is overloaded. In this context, several studies proposes techniques such as open sharing and hierarchical access models [21, 22, 23].

2.1.1 Cognitive radio concept

The term of Cognitive Radio (CR) is more generic and it introduces an scalable network architecture. The concept was introduced in 1998 by Joseph Mitola who described it as follows:

> "The point in which wireless Personal Digital Assistants (PDA) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to detect user communications needs as a function of use context, and to provide radio resources and wireless services most appropriate to those needs." [24]

Cognitive radio is an access network radio that can be dynamically programmed and configured to choose the best wireless channels in its vicinity to avoid network interference and congestion. It is also able to detect automatically available wireless channels, then accordingly adapt its communication parameters in a given spectrum band at a given location. This process is a form of dynamic spectrum management.

The cognitive radio concept need a whole chapter to present all its features and models, but we will focus in this section on its most relevant features that are in a direct relationship with our contributions.

Resource allocation and networking

As stated earlier, the fixed resource allocation techniques led to an under-utilization/over-utilization of the spectrum in the temporal and spatial scale. Hence, the concept of DSA has raised. DSA mechanism enable more flexible spectrum sharing allowing secondary users to exploit white spaces (idle frequency) in the licensed spectrum. This needs also a cooperative and smooth management of primary and secondary users to avoid interference issues between both groups of users

Hence, we can classify the dynamic resource allocation in CRs into 4 categories as follows:

- Graph based spectrum allocation [25]: This approach aims to establish a graph coloring model for spectrum allocation according to interference and other constraints.

- Game based spectrum allocation [26, 27] This approach is based on a competitive game between primary and secondary users to improve spectrum usage against the increasing service demands

- Auction based spectrum allocation [28, 29] Its objective is to preserve the QoS of Primary User (PU) by making the Secondary User (SU) paying for its utilization.

- Carrier aggregation based spectrum allocation [30] Aggregate multiple carriers into wider spectrum to improve the spectral efficiency.

Spectrum sharing and sensing

One of the essential functional key in CR networks is how to detect the spectrum holes (or white space: idle available radio resources) within the available spectrum band. The efficiency of CR networks relies especially on the accuracy of spectrum sensing approaches [31]. Moreover, other spectrum sensing approaches based on temporal and spatial correlation [32, 33] are proposed. Other models are based on energy-detection have been developed [34]. Furthermore, QoS-aware spectrum sensing model are proposed to support real time and latency-sensitive applications [31, 35].

We can also distinguish three major categories of spectrum sensing in CR concept:

- Cooperative spectrum sensing [36, 37] : In this approach a SU needs the cooperation of all other SUs and it collects their sensing results to improve the detection reliability.

- Full-Duplex spectrum sensing [38] Full-duplex sensing means that the SU is able to transmit using a spectrum hole and continue sensing at the same time the spectrum. This approach avoid conflicts and collision between SU and PU to happen.

- Database based spectrum sensing [39] A spectrum database is utilized to store the historical spectrum information and to generate a new available spectrum table based on the current spectrum state information. As an example of this category we mention the Spectrum Access System (SAS) proposed by the Federal Communications Commission (FCC) [40].

Energy efficiency in CR

Energy efficiency and saving is not only primordial during the sensing process, but also for the whole communication in CR networks and its one of the major challenges. Green radio (GR) concept is recently introduced and become essential for wireless communications especially with the slow advances in battery innovations. Many studies addressed this issue in CR context. Studies in [41] optimize the average of energy efficiency while authors in [42] aim to maximize it for different scenarios. In [43], authors propose a mathematical optimization model to maximize the energy efficiency under different constraints in order to find a trade-off between energy and spectral efficiency. Moreover, study in [44] investigates the trade-offs between several features such as QoS, fairness, primary user interference, and security.

In addition, power control is essential for spectrum sharing mechanisms in CR networks. Power control aim to minimize as much the interference between primary and secondary users when the channel is divided between them. Many studies addressed the power control subject via artificial intelligence and game theory models [45, 46]. Power adaptation during spectrum sensing phase has been the focus of study [42] taking into consideration the interference constraints.

2.1.2 Tv-white spaces

Due to spectrum limitations, solutions based on unlicensed spectrum has been proposed to extend network coverage and ensure more bandwidth availability. As example of unlicensed spectrum we cite the TV White Space (TVWS) [47] at low UHF frequencies that is able to support long-range communications, hence the ability to deploy large number of access points and controlling at the same time the interference issues. The band allocated for Tv broadcasting services has similar characteristics as the frequency band of mobile communications. In the other hand, the development of TV services affect the usage of this UHF band so that we get more unused frequency in this band [48]. These unused (or idle) frequency called Tv White Space can be allocated for mobile communication use. The exploitation of TVWS allows also to boost the deployment of 5G networks and spread mobile connectivity in rural areas.

Recent studies are exploiting the advantages of TV white space and introduced models to enhance the network QoS. Authors in [19] present a optimization framework for extra TV-white space spectrum allocation in Long Term Evolution (LTE) networks based on femtocells. Authors in [49] address the subject of dynamic spectrum sharing in heterogeneous networks.

2.1.3 Wireless mesh networks

Wireless mesh networks attract the interest of many researchers due to its autonomous deployment and low cost. In terms of real deployment contributions, Bicket et *al.* propose in [50] the design of a mesh network called *Roofnet* where the mesh routers are installed on the top of buildings to provide fast internet access over an urban area of four square kilometers in Massachusetts.

One major issue to consider when deploying a WMN is the number of wireless mesh nodes to be installed in the service area, the placement of these nodes and the interference issues that affect the performance of the network. The first two issues are related to a classic planning problem and in the literature many research papers address it by using mathematical models [51, 52, 53, 54] to infer the optimal number of mesh nodes to be installed to serve a fixed demand.

On the other hand, there are several recent studies that propose to enhance the performance of an already deployed WMN in terms of interference. In [7], the authors propose an efficient channel selection scheme to reduce the interference between co-located wireless mesh networks. In the same context, works like [55, 54, 56, 57] propose a channel assignment technique to alleviate the interference within the

WMN. Other researchers propose to enhance the WMN performance by focusing on link scheduling techniques [58] or multi-hop routing [9, 8, 59].

The solutions of most of the papers proposing WMN planning result in optimal fully connected networks satisfying the demand of clients but without considering fault-tolerance. Only few papers address fault-tolerance in WMN. In [60], the authors study the fault-tolerant planning of WMN in industrial environments. This work only proposes a radio coverage-based solution and neglects the connectivity and QoS concerns. In [61], authors propose a placement algorithm with fault-tolerance for mesh gateways. Usually, the number of gateways in a mesh network is very small compared to mesh routers so the problem is not really complex. We propose hereafter a fault-tolerance planning algorithm for mesh routers including also the above mentioned requirements.

Globally, the WMN planning solutions mentioned above, even with fault-tolerance, are based on static planning algorithms with fixed user throughput requests. This is not sufficient today, because users' demands are variable during the day and can be very high compared to the initial planning assumptions. Data demand fluctuation leads us to investigate issues in reducing the Operational Expenses (OPEX). Thus, when the demand is very low, some mesh nodes can be turned off and then the network energy consumption may be reduced. To study this issue, temporal patterns of the user demand fluctuation must be taken into consideration. In this context, Capone et *al* [62] address the problem of reducing the energy consumption by using a traffic profile. They propose to assign a randomly generated rate for each time interval and the traffic profile they propose is split into 8 intervals of 3 hours each. It is a good start but with such a profile, some traffic demand details can be lost. In our previous contribution [63], we proved that each base station (BS) is characterized by a specific profile class with more refined time intervals. Note that Capone et *al* concentrate on this profile and they did not consider the effect of interference, like most of the WMN planning contributions, and assigned to their WMN links an over-estimated rate which is not so realistic and that will definitely cause waste in resources.

In this thesis, we propose a dynamic planning solution for WMN with fault tolerance in order to save energy consumption and guarantee a more available bandwidth in the network since reducing the number of active mesh routers in each time interval alleviates the bandwidth use and enhance channel availability for the active ones. To achieve the dynamic planning task we use a realistic set of traffic profiles inferred from our study in [63]. Moreover, contrary to the work mentioned above that neglects the interference based on the assumption that multiple antennas avoid this phenomenon, we include this factor in our model. And we believe that as far as our investigations went, there is no contribution combining the objective of WMN planning, fault tolerance, energy efficiency with realistic dynamic traffic demands.

2.1.4 Drone-cells based networks

Smart deployment of drone-cells has been a topic of extensive research; examples of such work include [10], where the authors use binary integer linear programming (BILP) to selectively place and drone-cells networks and integrate them to an existent cellular network infrastructure while temporal increase of user demands occurs. They design the air-to-ground channel for drone-cells as the combination of two component: a component for a non-line-of-sight model (NLoS) and another one for a line-of-sight (LoS) model. Authors in [64] investigate the number and the 3D placement while drone-cells formation. The study aim to optimize not only the geographic coordinates of drone-cells but also their total number while serving users.

In [65], authors propose a framework for drone-cells formation. Thee framework aim to optimize the coverage (by increasing or decreasing the drone altitude) in order to remove dead zones and reach network capacity goals.

While previous mentioned studies investigate the small cells deployment in only drone-cells networks, other studies such as [66] investigate the issue of drone-cells assisting the existent cellular network and take into consideration the design of the macro BS channel. Authors in [66] addressed in their study two types of channel design such as air-to-ground channel for drone-cells transmission (designed as in [10]) and ground-to-ground for classical macro BS channel. The authors consider also the energy issues for drone-cells formation and propose an assisted solar energy for drones battery. Spectrum sharing with cellular networks and link design are also considered in [67] and authors modeled the drone-cells network as a poison point process (PPP) within a limited height.

whilst, authors in [68] consider the problem of congestion and propose a framework for congestion management in drone-cells network. The solution of drone-cells as relay or repeater for existent macrocells are considered in [67, 69]. Studies as, [67, 69, 70] addressed the drone-cells problem in a public safety context.

Contrary to previous studies that aim to optimize the placement of drone-cells while deployment, [71] investigates path optimization issues during drone-cells formation.

We aim in our thesis to propose a framework for dynamic and proactive deployment of drone-cells to assist existent cellular networks during mass event such as sport events. Our solution uses a predicted time-series of users data demand based on a real traces of cellular network demands that was proposed in the third chapter. Our proposed contribution draws on advances not only in the issues of optimal network deployment for drone-cells, but also in how dynamically manage this deployment in order to enhance the existent infrastructure using a multi-agent reinforcement learning approach.

Most of these innovative techniques are suffering from data availability and some study that are proved to be efficient theoretically, fail to be applied in realistic networking contexts. We are addressing in this thesis network resource management and optimization from a realistic perspective by using real network dataset traces. Mining this plethora of traces helps to understand more the real behavior of users data consumption and how the network reacts in return. Therefore, these network analysis results help us to propose more efficient network resource optimization techniques that are more adequate to the real-world bandwidth consumption. Note also that cognitive radios approaches differentiate primary and secondary users and aim at most of times to promote primary users. Whereas, in our work we aim to classify users based on their data consumption behaviors.

2.2 Network traces analysis: Data mining approach

2.2.1 Network classification

Before rethinking cellular network deployment, operators must begin by analyzing the existing architecture and optimize the huge number of data flows. CDRs can be very useful for this step due to the rich quantity of information that they offer. Operators can propose new algorithms for the optimization of the actual resource allocation. In addition to that, an online prediction algorithm made to forecast the load or the number of users attached to each antenna can be very useful as well to build a dynamic set of resource distribution techniques.

CDR and mobile network datasets provide rich quantity of information. This recently attracted many researchers focusing on mining datasets. With the help of several techniques such as classical statistics or artificial intelligence algorithms, we can extract useful knowledge for characterizing and modeling mobile networks. These results are essential to understand users' behavior and their needs in terms of transportation systems [72, 73], urban planning [14], telecommunication, etc.

Mobile traces analysis are important for city planning projects and may help to categorize cities and understand their structures [74, 72, 75]. In [76], authors exploit mobile traces to extract the most visited sites. This is used to develop tourism applications [77] and to semantically annotate those places [78].

Other studies focused on analyzing the behavior of users' consumption of network resources. M. Fiore et *al.* [79] have analyzed a CDR dataset in order to classify BS call profiles based on traffic volume snapshots defined within a time interval T. After identifying the class of each snapshot within T, they classify the remaining time interval snapshots using k-means. The method results on classifying BS' snapshots according to traffic level and time. The drawback of this work is that it does not provide a straightforward classification of the BS itself and you need to analyze all the BS resulting snapshot classes in order to infer the nature of users' activity within it. In [15], authors propose a classification method for geographic zones based on location signatures analysis. The signature is measured on (aggregated) compressed data (week) and then clustered using k-means. However, the compression level may impact the classification because the profile may change from a day to another in some places and compression may hide some outliers or special events.

Authors in [80] analyze the relationship between the service applications used by customers and the correspondent type of land-use and they inferred three land-use profile as home, work and mixed areas. Their analysis is only based on the types of application and not on the quantity of data consumed per subscriber number which may be more accurate. J. Toole *et al.* [14] propose a classification of cell profiles using a CDR dataset and zoning regulation dataset containing administrative information about land-use from which they inferred the different classes to train a supervised machine learning. Actually, it is very useful to correlate CDR dataset with zoning regulation information to enhance the classification outlines but the latter dataset is not easy to access in some countries. It is hence preferable to restrict the study on CDR dataset to infer profiles and classify the BSs.

Authors in [13] analyze users' behavior by investigating their activities dynamics and use k-means to cluster similar patterns and identify the structure of the cities. These two contributions ([13, 14]) result in clustering network patterns into several classes. The problem is that some of these classes have very similar shapes. These results in numerous similar users' activity patterns. As we will explain later, it is important to group similar patterns into a small number of classes rather than to have a multitude of patterns that are correlated. Moreover, most of these studies use k-means, with the classical euclidean distance, to classify network patterns.

2.2.2 Network data prediction

In addition to network classification, CDR can be exploited to investigate the seasonality and periodicity characteristics of users' network activities. Literature shows that the users' data consumption pattern can be divided into two components. The first one is a seasonal component and the second one is stochastic. Study in [81] proved that the second component is unpredictable which leads to the difficulty of predicting mobile traffic time-series with a simple identification of the deterministic trends or using analytic tools as in [73]. Therefore, it is necessary to use more efficient tools like machine learning which preserve the required stochastic generality such as SVR. Moreover, authors in [82] proved that SVR is more accurate than other machine learning tools for network traffic prediction.

Authors in [83] propose a Markov chain model to predict Internet traffic. They used a one-week aggregated dataset to analyze network periodicity. But, the dataset size allows only to capture daily cycles which can not be efficient for long-term prediction. In our study, we capture all network cycles (daily, weekly, monthly ..) which is more accurate. In addition, they propose two sub-models to predict traffic for weekdays and weekends. Our prediction model is more general and performs prediction for all day types. It even captures the change of BS profile throughout the week.

In our work, we use SVM rather than classical K-means or Markovian algorithms. It helps to consider all details of the dataset without the need of additional semantic information and to treat on-line data. Moreover, the DTW is proposed for time series similarity comparison instead of the ED distance. The choice of three daily classes is proposed and it is shown to be optimal with respect to DTW distances. The designed SVM algorithm will automatically classify all BSs profiles into the three traffic classes. We finally prove that the BS load is predictable for short and long term cycles.

2.2.3 Network data mining for anomaly detection

Anomaly detection covers a wide variety of domains and its definition depends on the idea that normal behavior can be distinguished from abnormal (i.e anomalous) behavior. A general definition of anomaly or outliers detection is given by Chandola *et al.* in [84], and it consists on signaling an anomalous observation if it presents a deviation from known normal data. Outliers detection can be divided into two main domains. The first one looks for individual objects from the data and these kind of applications may refer for genetics domain, medicine, image processing etc. The second one focuses on extracting anomaly from a large set of ordered sequence data or time-series and this domain covers a large set of applications like economy, financial analysis, energy, e-commerce , etc. We focus on the latter category since we address in our contribution the issue of detecting anomalous sequence (or sub-sequence) within network activities time-series. A deeper analysis reveals that different techniques actually address different problem formulations of outliers detection. Most of the existing research focuses on one of the following problem formulations:

Table 2.1: Sequence Anomaly detection related works

Sequence-Based	Subsequence-Based	Pattern frequency	Contextual
Unsupervised & Semi-supervised techniques [85, 86, 87]	window scoring techniques [88]	Subsequence occurrence based [89]	Window based [90]
Window-based [17, 91]	Segmentation techniques [92]	Non-contiguous subsequence occurrence[93]	Proximity-based[94]
Markovian & HMM models [95, 96, 86, 97]	Anomaly dictionary [98, 99]	subsequence permutation based [100]	Prediction based[101]

- ***Sequence-Based approach:*** consists on Detecting anomalous sequences from a dataset of test sequences.
- ***Subsequence-Based approach:*** aims to detect anomalous contiguous subsequences within a large sequence.
- ***Pattern-based approach:*** Detects patterns in a test sequence with anomalous frequency of occurrence.
- ***Contextual anomaly detection approach:*** Detects a group of points or periods of time that are anomalous regarding its normal behavior.

Table 2.1 summarizes these approaches and highlights contributions of each solution.

Moreover, anomaly detection in networking and telecommunication fields is recent. Researches on cellular access networks anomaly detection are limited due to the difficulty of accessing network data, but since operators networks are publishing some of these data, it becomes possible to explore and model these information in order to analyze networks issues. Some studies about networks anomaly detection have applied their proposed method on simulated data or limited set of network traces datasets which may decrease the generalization ability of some algorithms and affect their performance. Authors in [102] propose a KPI-based faults detection method to monitor radio measurements and comparing them to normal behavior. They extracted a dataset of Key Performance Indicator (KPI) generated from a developed network simulator. A Bayesian method is proposed in [103] for anomaly detection in GSM and UMTS networks but they only consider the call drop rate to detect anomalies which cannot be efficient to monitor the network. In [104], authors proposed a framework for automatic fault detection and diagnosis in cellular network based on a set of Operations Support Systems (OSS) KPIs. The issue with this contribution is that the solution does not allow an on-line detection of faults which guarantee an proactive monitoring of the network, since they use an unsupervised clustering techniques for detection. Moreover, these contributions just address one type of network anomaly; fault detection, although many types of outliers features may be captured within network data and we need a general method able to detect them. A K-Nearest Neighbor (KNN) based algorithm was proposed in [87] to detect sleeping cell in LTE networks. Obviously, K-NN is a robust and fast unsupervised machine learning but it suffers from high-dimensional input data. On-line semi-supervised anomaly detection techniques are used for other types of networking applications. One-class SVM-based algorithm is proposed in [105] for malware detection. In [106], authors use random forest for network intrusion detection. Random forest is considered to be one of the most accurate methods for anomaly detection and is proved to avoid over fitting issues with less sensitivity toward noisy data [107, 108]. In [109], an Isolation Forest algorithm, which is an enhanced variant of random forest, is proposed for video streaming outliers detection.

In addition to the spatial anomaly, our framework is able also to detect with precision the network outliers in a temporal scale. An SVR [110] based prediction algorithm is used for this task. In [111], SVR algorithm is used to predict the daily BS time-series and results show high accuracy of the algorithm. Moreover, authors in [82] compared the SVR prediction with other algorithms and they proved that SVR provides better results. Furthermore, in [91], authors are proposing a tool for network anomaly detection based on arbitrarily defined thresholds. They label time slots as outliers if they are strongly deviated from the expected normal values. These kind of approaches need a deep analyzes for all BS and time slot in order to fix the so-called normal threshold for each BS time-slot. This constitutes a heavy duty and is not efficient at all since thresholds are base-station and time dependent. Hence, the SVR based prediction approach avoid this issues by predicting the daily normal behavior for each BS (which is defined then as a threshold to be compared to the testing time-series) with high precision.

We propose in this thesis a general framework for network anomaly detection that allows to detect any type of network spatio-temporal anomalies, either fault detection or sudden peak of occupancy (data consumption) or abnormal behavior within a cell. To the best of our knowledge, no previous contributions propose a general spatio-temporal anomaly detection framework based on OCSVM and SVR for cellular radio access network and using real-world CDRs dataset. Besides, our model can be easily adapted to other types of network and wireless architectures with little modification.

Chapter 3

Base station profiles classification

3.1 Introduction

High user mobility and various data consumption behaviors induce several load patterns and network fluctuation. Therefore, network operators must face these challenges to deal with user consumption dynamics. They need to abandon the classical techniques such as statically sharing the licensed spectrum, which cannot efficiently manage instantaneous users demand dynamics. Hence, new techniques and radio resource allocation mechanisms must be designed to dynamically manage user demands and provide better quality of service (QoS) [19].

Moreover, rich information about user consumption behaviors and network load fluctuation trends can be extracted from mobile network datasets like call detail records (CDR). The first step toward enhancing network performance is to mine and analyze the relevant information hidden into the huge metadata of CDR datasets.

CDR datasets have been exploited in many fields such as studying human mobility [112, 113, 114, 115, 116], analyzing urban planning [72, 75, 117] and studying the structures of cities. Several studies focused on human mobility from an urban perspective (city structure, tourism, etc, ...) [118, 72, 119, 120]. They mainly address problems in spatial scale, such as inferring places of interest (POI). Whereas, analyzing these POI as well as their evolution in time is important for cellular network planners.

Network planning is crucial for network operators to enhance services that are cost-effective and QoS-dependent. Tools allowing the inference of instantaneous bandwidth demand permit network operators to dynamically manage the bandwidth and to implement innovative techniques for radio resource sharing between BSs. In this chapter, we aim to exploit CDR and data analysis techniques in networking context. More precisely, the goal of this work is to apply data mining on real datasets of CDR to assist operators in optimizing the network resource consumption. This prevents both under-provisioning and over-provisioning. Hence, network operators can innovate their network by integrating intelligent solutions to optimize dynamically their cellular access network to guarantee a better quality of service.

In order to enhance the continuity of network services, it is important to estimate with high confidence how the bandwidth demand on a base station (BS) at a given time t is shared among all the BSs in the following instants. In this chapter, we propose a method to derive a Temporal Origin-Destination Matrix (TODM) describing how bandwidth request at a BS at a given time is spread over the other BSs in the next time instant. TODM is a temporal matrix because it depends obviously on the time of the day and the day of the week. Note also that it is very different from classical traffic [121] origin-destination matrices : it includes network load measurement and expresses the part of the load on BS i that moves at t towards BS j. Actually, when a user moves from a BS i to a BS j, network resources must be moved accordingly and the TODM is a tool intended for that. But, besides the continuity of service, bandwidth dimensioning on the BS requires to know the evolution of the bandwidth demand on this BS, taking into account also newly arrived users in the network and other ones which just left it. TODM helps to estimate the amount of bandwidth that moves from one place to another. However, it does not provide information about the total instant load locally (in each BS).

But before deriving the Temporal Origin-Destination Matrix (ODM), we start by characterizing the daily bandwidth consumption for each BS and we propose an on-the-fly algorithm for BS load classification based on Support Vector Machine (SVM). First, in order to have reference classes for our SVM classification, we infer typical BS load classes from a modified K-means algorithm. The obtained profile classes are then used to run an SVM-based algorithm which allows assigning BSs on-the-fly to different classes with high accuracy and efficiency. We propose in this chapter a modified K-means technique based on DTW distance, instead of classical Euclidean distance, to detect BS load profile. Besides, clustering load profiles is not sufficient for operator networks, since a BS class may change throughout a period of days. An automated classification technique is needed to adjust the on-the-fly classification by taking into consideration the most recent history. That is why we propose a classification method based on SVM to classify the large scale dataset of BS load profile. We then exploit the classification results to estimate the bandwidth mobility.

This chapter is organized as follows. Section 2 presents the data processing scheme and the CDR dataset used. Section 3 presents the clustering model of BS load profile. Section 4 presents the classification model and shows its results evaluation. In Section 5 we present the temporal network flows evolution and the TODM. After a discussion on section 6, we finally conclude in section 7.

3.2 Base station clustering and classification

3.2.1 Framework description

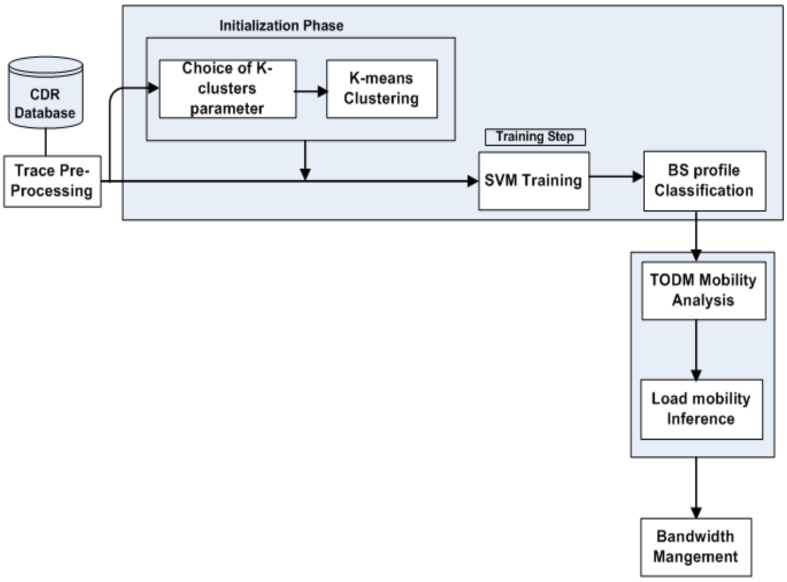


Figure 3.1: Framework scheme and methodological process

The framework in figure 3.1 describes the full processing sequence for bandwidth management. It is composed of an initialization step and a training step. The initialization step consists of two phases: choice of the pertinent number of clusters and data clustering based on K-means. The initialization phase helps to label a subset of BS data. The resulting labeled data is used to train the SVM module. The resulting classes are also used as input for the network bandwidth mobility analysis in order to infer the amount of bandwidth that moves between different areas. The pre-processed traces are finally used as input for the regression (prediction) module which consists on predicted the local BS load and complete the bandwidth mobility information in order to provide a complete bandwidth management framework.

Figure 3.1 also depicts the methodological approach of the section: Section 3 bloc incorporates the BS clustering and classification process, Section 4 addresses the network load mobility and Section 5 bloc develops the BS load prediction part. Mobility analysis based on TODM and SVR load prediction are described in subsequent sections, while in this section we present section 3 bloc.

3.2.2 CDR dataset description

For our study, we use a real dataset of CDR provided by the network operator Orange Senegal in the context of a challenge for development named D4D-Senegal [12]. The D4D dataset contains call detail records of phone calls and short messages (SMS) exchanges of about 9 millions users for the year 2013. The data-set is divided into 3 sets: one set contains the antenna-to-antenna traffic for 1666 antennas on an hourly basis, another contains one year of coarse-grained mobility data at district level, and a last one contains fine-grained mobility data on a rolling 2-week basis for a year for about 300,000 randomly sampled users. Table 3.1 presents an example of raw data contained in the latter set. In this study, We exploit the last set to analyze the traffic load and its distribution between BSs over the country. We extract important information such as user temporal location according to BS attachment, so we can track users and infer their most visited places. We also obtain information about instant traffic load and BS capacity for the whole day. This allows us to identify the daily load profile of each BS.

Table 3.1: Example of raw data

User ID	Time	BS ID
1	2013-03-18 21:10:00	275
1	2013-03-18 21:20:00	275
1	2013-03-18 21:40:00	280
2	2013-03-21 08:30:00	763
2	2013-03-21 08:40:00	761

Fine-grained mobility dataset contains raw data which can not be exploited directly. Therefore, for our needs, we have to extract more relevant knowledge about BS activities. We start by applying a statistical analysis on the dataset to extract time-series of the instant numbers of attached users to each BS and within each time interval of 10 min (granularity assessed by the dataset) along the day. Figure 3.2 represents a graphic visualization of BSs' load evolution throughout the day in three different districts of Dakar (each horizontal line represents one BS). We notice from Figure 3.2 that the load pattern differs from one BS to another and we have different BS profiles which is analyzed in the next parts.

The extracted data is strongly representative and proportional to the network load and this is one of the goals of our work. Figure 3.3 (top) shows the load of a BS in a typical week. The middle figure corresponds to a real daily user average bandwidth consumption on this BS. The bottom figure is the product of the two traces and gives the exact data consumption in Gb/s.

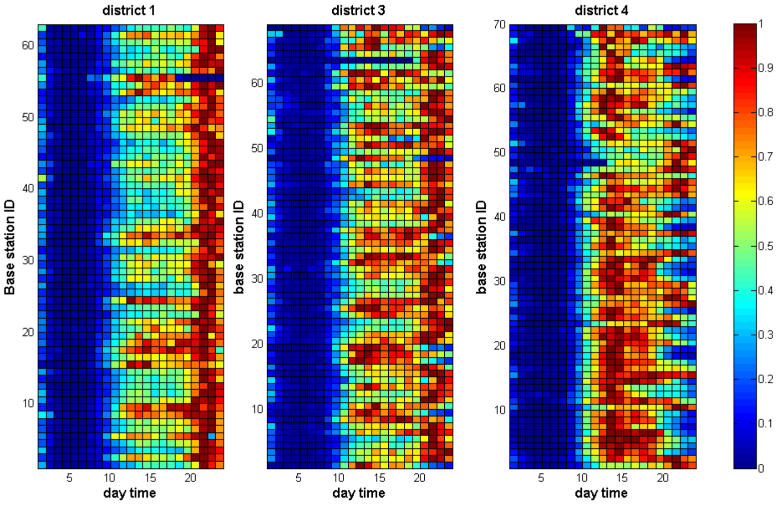


Figure 3.2: Data visualization of BS time-series: each plot correspond to a district and each plot line correspond to a BS time-series

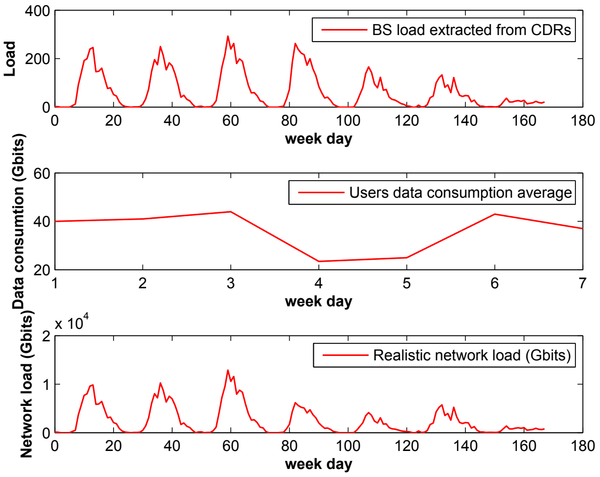


Figure 3.3: Realistic network load extracted from CDR data combined to extra network information

3.3 Base station profile categorization

The set of BS' time-series S, extracted from the dataset, is defined as $S = \{(BS_i, d)\}$ with:

$$(BS_i, d) = \{N(t), t \in T\}, i \in B$$

Where BS_i corresponds to the i^{th} BS ID from the set B of BSs and d stands for the day date. $N(t)$ corresponds to the number of attached user at time interval t (with ten minutes duration) and T is the set of time interval of a day.

We notice from figure 3.2 that the dynamics of BS load along the day can be represented by different profiles. As these profiles can not be recognizable directly from the dataset, a non-supervised clustering is applied at first to the set of BS time-series. The objective of data clustering algorithms is to partition data into different groups while guaranteeing a maximum intra-group similarity, so that elements in the same cluster are very similar, while the similarity between groups is minimal. Usually, clustering tools use the ED as a similarity measurement between data features, but this measurement is not suitable for time-series as it is very sensitive to time distortion. Therefore, we propose to use dynamic time warping (DTW) as a similarity measurement for BS time-series.

3.3.1 Dynamic time warping

We present in this section the dynamic time warping (DTW) technique that aims to measure the similarity between BSs' time-series.

Studies such as [13, 14, 15] have analyzed the daily cell load patterns and classified them according to users activities or land-use. Most of them have applied K-means tools with euclidean distance (ED) as similarity measurement to cluster load time-series. However, in [16] it was proved that dynamic time warping (DTW) is more efficient than ED for measuring time-series similarities due to its insensitivity to time distortion.

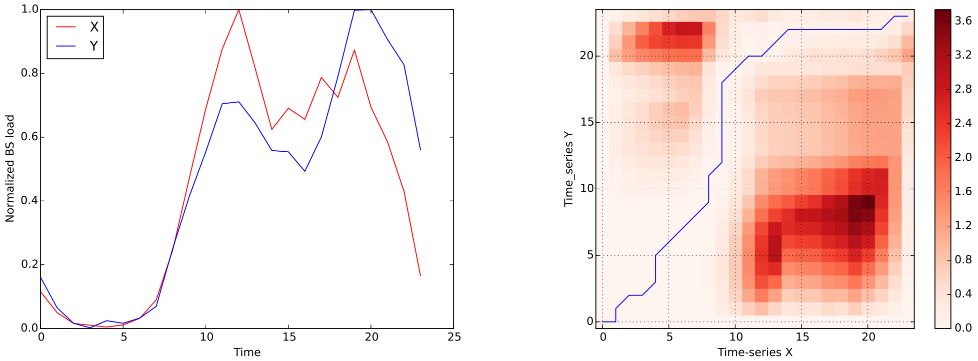
DTW [16] was introduced to replace Euclidean distance for time series processing. Given the following two time-series:

$$X : x_1, x_2, x_3, ...x_n \tag{3.1}$$

$$Y : y_1, y_2, y_3, ...y_m \tag{3.2}$$

To align these two sequences using DTW, we first construct an $n \times m$ matrix where the (i^{th}, j^{th}) element of the matrix corresponds the squared distance:

$$d(x_i, y_j) = (x_i - y_j)^2 \tag{3.3}$$



(a) Time-series with different load profiles

(b) DTW path illustration

Figure 3.4: Example of DTW path for different load time-series

which is the alignment between points x_i and y_j.

The challenge of the DTW method is to discover the optimal *warping path* that minimizes the total cumulative distance between time-series. The optimal path is the path that minimizes the warping cost across the matrix:

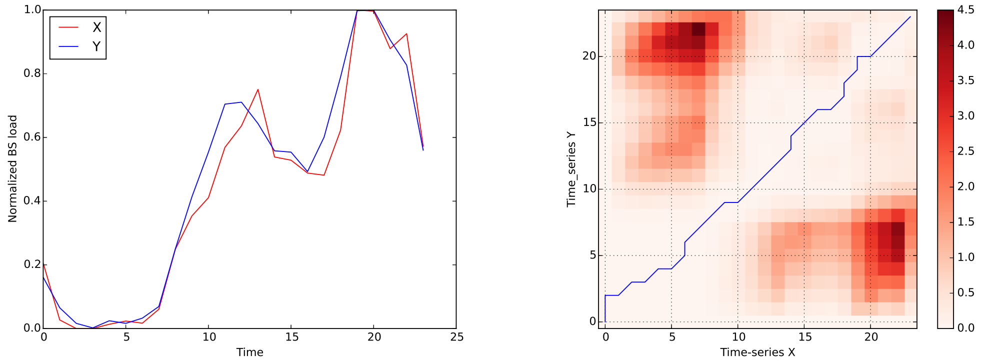
$$D_{\mathrm{DTW}} = \sum_{w_{\mathrm{opt}}} D(x_{\mathrm{i}}, y_{\mathrm{j}}) = \min\left(\sum_{k=1}^{K} w_{\mathrm{k}}\right). \tag{3.4}$$

where w_k is the matrix element $(i, j)^k$ that also belongs to k^{th} element of a warping path W, a contiguous set of matrix elements that represent a mapping between X and Y. This warping path can be found using dynamic programming to evaluate the following recurrence:

$$\mathrm{D}(\mathrm{i},\mathrm{j}) = \mathrm{d}(\mathrm{i},\mathrm{j}) + \mathrm{min}\{\mathrm{D}(\mathrm{i}-1,\mathrm{j}-\mathrm{l}), \mathrm{D}(\mathrm{i}-1,\mathrm{j}),\ \mathrm{D}(\mathrm{i},\ \mathrm{j}-1)\}$$

where $d(i, j)$ is the distance found in the current cell and $D(i, j)$ is the cumulative distance of $d(i, j)$ and the minimum cumulative distances from the three adjacent cells.

Figures 3.4 and 3.5 show two examples for measuring and searching the shortest warping path between two time-series of load belonging to two different BSs extracted from the D4D dataset (that should be classified into different classes). Figure 3.4(a) shows load time-series of two different profiles (representing time-series X and Y of the model described earlier) and figure 3.4(b) shows the shortest warping path between both time-series. We can notice that the path does not



(a) Time-series with similar load profiles

(b) DTW path illustration

Figure 3.5: Example of DTW path for similar load time-series

pass through the diagonal of the matrix. The DTW distance for this example is equal to 1.96 while the euclidean distance is equal to 1.28. Hence, the DTW distance is more precise for measuring the similarity between these two different profiles.

Figure 3.5 shows an example of two very similar load profiles, where the shortest warping path cross the matrix diagonal. The DTW distance for this example is equal to 0.74 while the euclidean distance is equal to 0.82. These results confirm that DTW is more accurate and suitable for time-series similarity measurement than the euclidean distance.

3.3.2 Similarity measurement comparison and choice of K-clusters parameter

Figure 3.6 presents a comparison between DTW-based clustering and Pearson Correlation Coefficient (PCC) based clustering. The comparison is based on silhouette measure [122]. It measures how close each point in one cluster is compared to points in the neighboring clusters. The silhouette ranges from -1 to $+1$, where positive values indicate that the object is well matched to its own cluster and poorly matched to neighboring clusters. We used this metric because it does not need prior knowledge of the real BS profiles (An information that we want to extract from clustering). In this analysis, since we need to classify the BS profile day by day, we applied correlation coefficient to normalized daily signatures (Same signature used for DTW clustering) to more efficiently manage the bandwidth. We notice from figure 3.6 that DTW-based clustering is always positive for different K values while the Pearson coefficient-based clustering has negative values. This means bad matching within each cluster. Therefore, DTW-based clustering is more efficient than the Pearson' method.

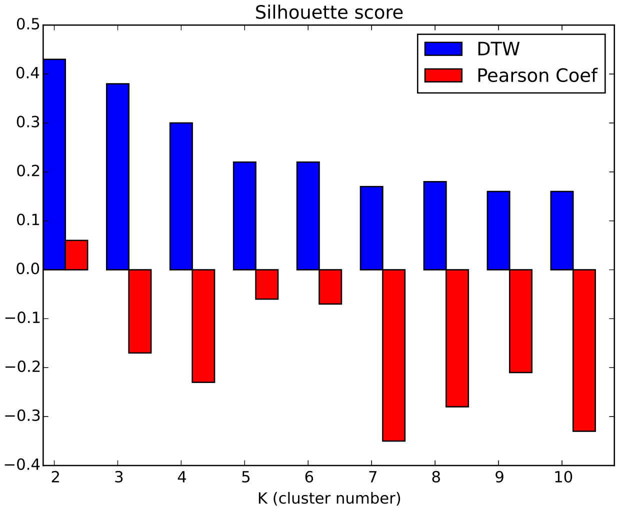


Figure 3.6: Silhouette coefficient comparison between DTW and Pearson coefficient clustering

The K-means can render so many classes. However, we place the study in a wide-scale national network context with a network that has to be managed in terms of bandwidth. We think that a typical operator or a researcher that is willing to analyze the massive coarse grain variations of such massive data has very sufficient and pertinent information with $k = 3$. To find this value, we measured the variance of the cluster number K. We applied the Elbow method to determine the optimal clusters number K. The Elbow method looks at the percentage of variance as a function of K. The number of clusters is chosen in manner that adding another cluster does not give much additional information. Figure 3.7 shows the evolution of the Elbow score (Variance measure) in function of K-clusters. This evaluation shows that we can stop at $k = 4$.

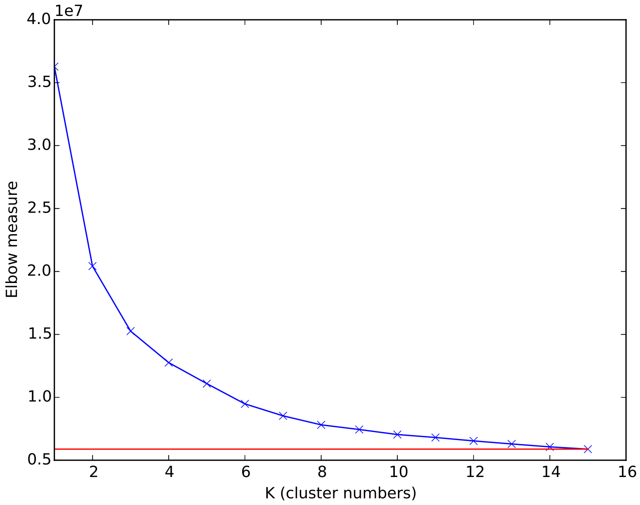


Figure 3.7: K choice: Elbow method

3.3.3 BS time-series clustering

To detect patterns from BS load set, we use then K-means [123] algorithm but with DTW distance instead of classical ED and PCC.

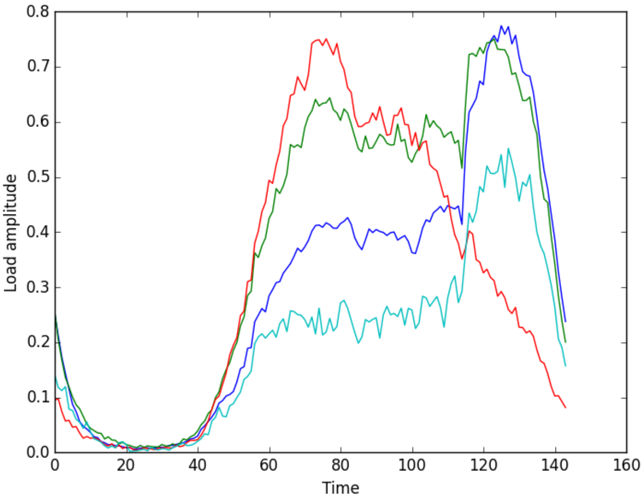


Figure 3.8: Clusters centroid for k=4

Figure 3.8 depicts the centroid of clusters in the case of $k = 4$, i.e. four classes of data traffic. We notice that we have 3 different clusters with different profiles: one cluster shape with an almost constant activity during the day (will be referred hereafter as the 'Almost-Constant' class or class 1), a cluster shape with peak activity at the morning (will be referred hereafter as the "Morning-Peak" class or class 2), and another one with peak activity during the night (will be referred hereafter as the "Night-Peak" class or class 3). We also obtained two similar demand profiles (blue and sky blue curves) which correspond to data demand behavior related to the night peak profile, but with different amplitudes. As our goal, is to categorize BS profile which deal better with our main goal of dynamic resource allocation, we logically choose, for the rest of the work, to classify the BS profiles into three classes only (and not four) that are proportional to users data consumption behaviors (i.e. morning peak profile, constant profile and night peak profile).

3.4 Classification algorithm for base station profile

3.4.1 Support Vector Machine

Let x denote the variable of the input space $\mathcal{X}$, y denote the variable of the output space $\mathcal{Y}$, and $f : \mathcal{X} \rightarrow \mathcal{Y}$ denote the learning function between the input space and output space. Note that $\mathcal{Y}$ can be arbitrary; thus, it is called the structured output. For example, in a binary classification problem, $\mathcal{Y} \in \{-1, +1\}$; and in tracking, $\mathcal{Y}$, represent all possible transformations of the target.

The objective of structured SVM is to learn a discriminant function (or hyperplane) $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over inputoutput pairs (x, y). Then the classification function f for a given input x can be derived by maximizing F

$$f(x) = \arg\max_{y \in \mathcal{Y}} F(x, y). \tag{3.5}$$

Assume that F has the following form:

$$F(x, y) = \langle w, \mathbf{\Phi}(x, y) \rangle \tag{3.6}$$

where $\mathbf{\Phi}(x, y)$ is the combined feature mapping of the inputoutput pair (x, y), and w is the parameter. Thus, F is linear in $\mathbf{\Phi}$.

Given a set of inputoutput example pairs $(x, y)(i = 1, \ldots, N; y \in \mathcal{Y})$, F can be learned by solving the following objective function (primal formulation):

$$\begin{aligned}
&\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\
&\text{s.t. } \forall i \quad \forall \mathbf{y} \neq \mathbf{y}_i : \langle \mathbf{w}, \delta\mathbf{\Phi}_i(\mathbf{y})\rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \quad \xi_i \geq 0
\end{aligned} \tag{3.7}$$

where C is the trade-off parameter and $\delta\mathbf{\Phi}_i(\mathbf{y}) = \mathbf{\Phi}_i(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{\Phi}_i(\mathbf{x}_i, \mathbf{y})$. $\Delta(\mathbf{y}_i, \mathbf{y})$ is the loss function, which can be used to evaluate the importance of the sample $(\mathbf{x}_i, \mathbf{y})$

To solve the quadratic problem of the primal formulation, the previous equation can be converted into its dual form by Lagrangian duality and we obtain the following dual formulation:

$$\begin{aligned}
&\max_{\alpha} \sum_{i, \mathbf{y} \neq \mathbf{y}_i} \Delta(\mathbf{y}_i, \mathbf{y})\alpha_i^{\mathbf{y}} - \frac{1}{2} \sum_{\substack{i, \mathbf{y} \neq \mathbf{y}_i \\ j, \bar{\mathbf{y}} \neq \mathbf{y}_j}} \alpha_i^{\mathbf{y}} \alpha_j^{\bar{\mathbf{y}}} \langle \delta\mathbf{\Phi}_i(\mathbf{y}), \delta\mathbf{\Phi}_j(\bar{\mathbf{y}})\rangle \\
&\text{s.t. } \forall i \quad \forall \mathbf{y} \neq \mathbf{y}_i : \alpha_i^{\mathbf{y}} \geq 0, \quad \sum_{\mathbf{y} \neq \mathbf{y}_i} \alpha_i^{\mathbf{y}} \leq C
\end{aligned} \tag{3.8}$$

where $\alpha_i^{\mathbf{y}}$ represents the Lagrangian multiplier, $\bar{\mathbf{y}} \in \mathcal{Y}$ denotes the auxiliary output variable, which has the same meaning as $\mathbf{y}$, and $\delta\mathbf{\Phi}_j(\bar{\mathbf{y}}) = \mathbf{\Phi}_j(\mathbf{x}_j, \bar{\mathbf{y}}_j) - \mathbf{\Phi}_j(\mathbf{x}_j, \bar{\mathbf{y}})$

3.4.2 Kernel functions

Some pre-defined kernel functions are used for solving the dual problem formulation of the SVM model. Table 3.2 depicts some examples of kernel functions that are used in the literature.

Table 3.2: Examples of SVR kernel functions

Kernel function	**equation**
Linear	$K(x_i, x_j) =< x_i.x_j > +b$
Polynomial	$K(x_i, x_j) =< x_i.x_j >^d +b$
Sigmoid	$K(x_i, x_j) = tan(< x_i.x_j >) + b$
RBF	$K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|)^2$

For our model, we used the Radial Basis Function (RBF) due to its adaptability to handling non-linear and multidimensional time-series as the datasets that we use on all the thesis works.

3.4.3 Multi-class SVM

SVM is essentially a binary method and data is divided into two classes of $+1$ and -1. Most of rotating machinery fault diagnosis require more than two classes. Two multi-class SVM methods will be mentioned below:

One-against-all approach

In this approach, SVM classifiers are trained as a single classifier, i.e for an individual class, which is labeled by "$+1$" for a specific data class, where all the rest of data classes are considered as a second entire class and are labeled by "1". Then, a SVM multi-class network will be created by combining developed SVM series. This algorithm is presented in fig 3.9. Here, four classes are considered including healthy state, and three unbalance conditions associated with faults in the first, second and third discs. To obtain a four class classifier, it is required to consider a set of binary classifiers SVMl, SVM2 and SVM3. At each step, one class separate from the others, then by combining the classifiers, multi-class classification is performing according to the maximum output before sign function applying. In fig 3.9, the flow chart for OAA procedure is presented. As it is shown, the i^{th} SVM classifier is trained by using all of the dataset in the i^{th} class. This class has positive label and all other classes with have negative labels. In the classification, the classifier with the maximal output is the estimated class label of the current input vector.

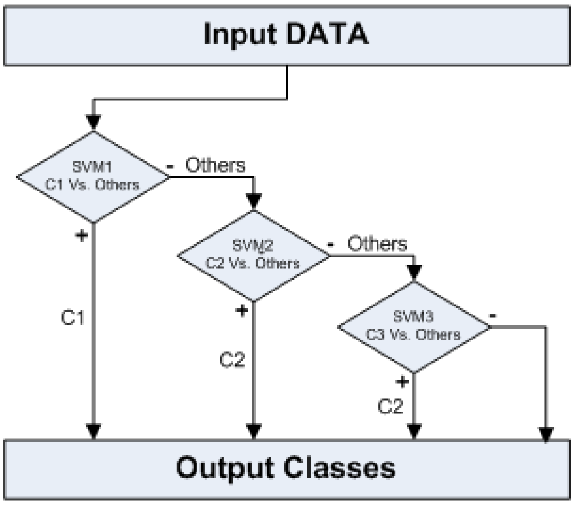


Figure 3.9: One-against-all approach of multi-class svm

One-against-one approach

In this approach, a SVM classifier is trained per each couple of classes. In other words, we obtain $K(K-1)/2$ classifiers for trained by each pair of data classes. The process of this approach is described by fig 3.10.

In the one-against-one approach the numbers of classifiers are usually larger than the number of classifiers in one-against-all technique. In OAO, for k individuals, it is required to evaluate the $K(K-1)/2$ classifiers, while only (k1) classifiers would be required in OAA. For example, if $k = 4$, the OAO approach requires six binary classifiers that should be trained while only three classifiers is sufficient for the OAA approach. However, the OAO approach needs more computational efforts and is much more time consuming at training stage; therefore, always employed for the problems with smaller number of faults. In order to classify a given pattern, it is necessary to evaluate all 6 binary classifiers and rank them according to their accuracy.

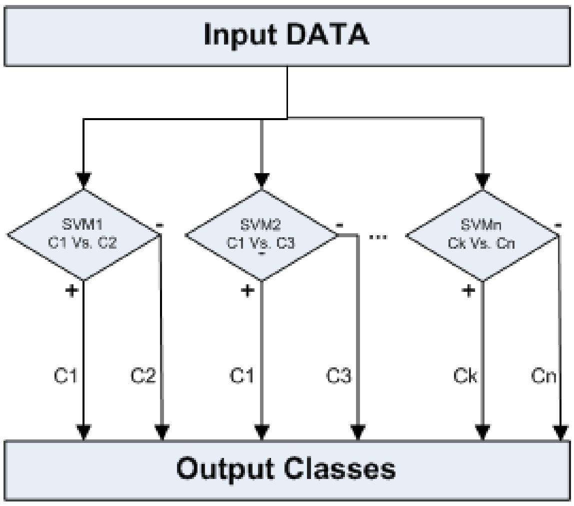


Figure 3.10: One-against-one approach of multi-class svm

3.4.4 K-fold cross validation

In order to train effectively the SVM classifier, the hyper-parameters need to be tuned and selected carefully. The k-fold cross-validation is a commonly used technique to evaluate the accuracy of SVMs with the selected hyper-parameters.

Cross-validation is a statistical approach to verify the classification performance of the model. It has attracted more and more attention in the ensemble of classifier models. The Cross-validation process can be executed through the following steps:

- Divide the dataset into K groups.
- Select some groups as training data to build the train model.
- Use the rest of the dataset as validation points (testing dataset) to verify the train model. Usually we use 20% of the whole dataset as testing data.
- Repeat the previous steps K times. Each subset is verified once. The results of the test are averaged or selecting the best verification result as a single estimation.

On these partitioned folds, training and testing is performed in k iterations. The accuracy obtained in each iteration is then averaged to get the model accuracy.

The advantage of the method lies in the repeating use of randomly generated sub samples for training and validation, and each result is verified once. Cross validation can prevent over fitting, and the disadvantage is time consumption.

3.4.5 SVM-based classification model

Some works from the literature propose to classify the network patterns using some mobile traffic signatures [124, 125, 13, 15]. Network signature is usually identified by its 'support', which defines the data aggregation level. Unlike the works mentioned above, we train and test our model with the normalized daily BS time-series, i.e the signature support is only for one day. Moreover, the BS profile classification is made for each day apart, so that we can detect whether the BS class changes throughout the days.

The SVM classification procedure considered in our research is based on the Multi-class SVM algorithm proposed by Weston and Watkins in [126].

Training step

The first step of the supervised classification consists in training the model with labeled data. Therefore, we use a training set of BS load time-series. The related class is defined with the help of the previous clustering analysis. The training vector (BS_i, d) (defined earlier), which represents the daily variation of the number of users (load) attached to BS_i, is then tagged with its adequate label y_i. Figure 3.11 shows some examples of training BS profiles corresponding to class 1, class 2 and class 3 respectively. We use the RBF kernel function because our training data are non-linear and it gives better results than other kernel functions. Globally, the training data (TD) are represented as following:

$$TD : (BS_1, y_1), (BS_2, y_2), ..., (BS_n, y_n)$$

where $BS_i \in \mathbb{R}^d$ represents the training vectors and d is the vector features number while $y_i \in \mathbb{R}$ represents the class labels.

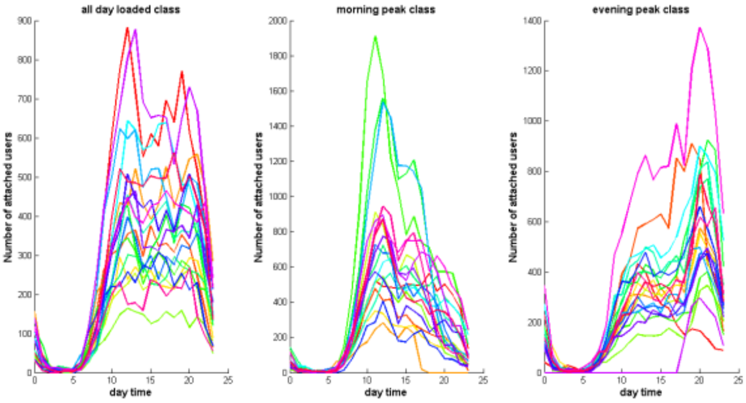


Figure 3.11: Training data for SVM classification

Testing step

Once the classification model is trained, we must ensure that it works well and has a good performance to classify on–the-fly new instances of BS load. The testing step is mandatory to fix and optimize the parameters of the SVM algorithm especially the choice of kernel function and its parameters. For that, we used another set of BSs whose class is predetermined to make sure that our model is capable to classify the remaining set. We use the cross validation technique during testing step to adjust the model with the best parameters. We consider the accuracy (Acc) metric as a criterion for the model optimality. The accuracy is defined as follows:

$$Acc = \frac{TP + TN}{N}$$

where TP is the total number of true positives, TN is the total number of true negatives and N is the total number of vectors used for testing. Hence, the objective of the testing step is to find the optimal values of the model parameters that maximize the accuracy.

3.4.6 K-means based classification

To validate the SVM classification model we should compare its performance to another classifier. We choose a modified K-means algorithm for this task.

Supervised K-means based model

We use k-means in a two-step manner. First, we run it on the training data used for SVM classification, under the constraint to obtain three clusters. The trained data are then classified into three reference clusters. We observed that we obtain almost the same classification as the one used to train SVM. Then, each cluster is averaged to obtain three reference time-series (Figure 3.12) which will be used as a centroid for the next step.

After this training phase, the second step consists in assigning a BS to a cluster by calculating its distance to the nearest reference of the three training clusters and then the classification is generated.

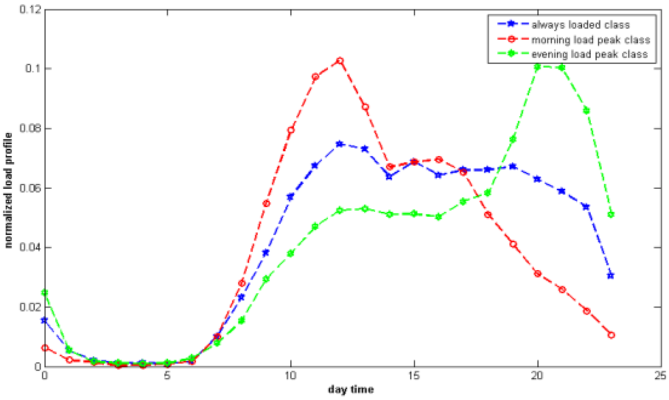


Figure 3.12: Normalized references of training clusters used for k-means classification

3.4.7 Simulation and classification performance results

The main objective of this part is to have an automatic and daily classification of all BS load time-series extracted from the whole dataset and also for new instances of BS load profiles. The results showed in this part corresponds to the classification of BSs installed in one dense district (Dakar plateau) of Dakar capital city. Figures 3.13-3.15 give a graphic comparison between the results obtained from the classification by k-means (left plot), SVM (middle plot) and real labeled traces (right plot) (These latter traces are extracted from the clustering step). Each line in each plot belongs to the load vector of one BS, i.e each color line in the left plot of figure 3.13 corresponds to a BS which is class 1 labeled by K-means. We present in these figures the normalized BS time-series using the "min-max" normalization. These figures depict the result visualization of one typical day (Monday) and they include the curves of all BS installed in 'Dakar Plateau' district. It is the most crowded district of Dakar with 72 BSs and represents 16% of all Dakar BSs).

Table 3.3 resumes the numerical comparison between both classifier results compared to the real classification. This results analysis represents an average performance result. We choose some statistical performance measurements extracted from the confusion matrix such as the *Precision*, the *Recall* and the *F1-score*. We notice from the table that SVM is more efficient than k-means for class 1 and class 3 results while both classifiers results are close for class 2, which means that SVM classification globally outperforms K-means.

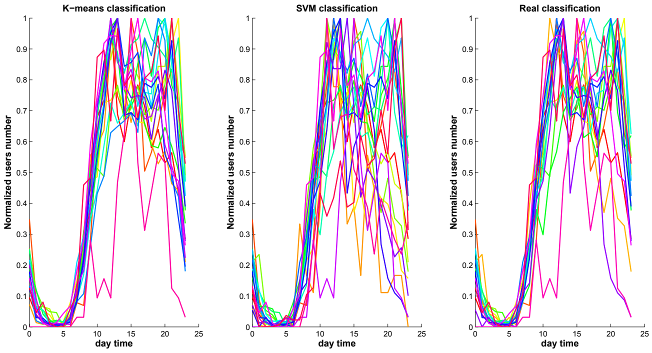


Figure 3.13: Classification visualization results for class 1

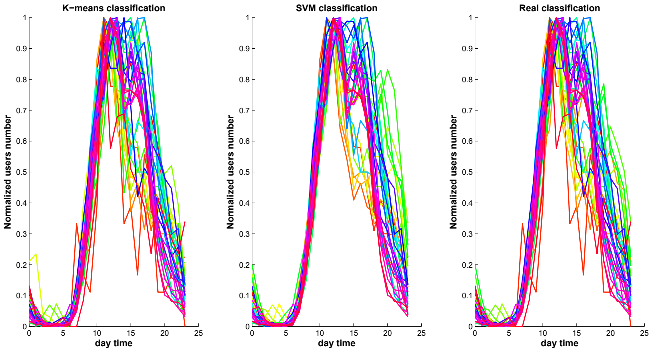


Figure 3.14: Classification visualization results for class 2

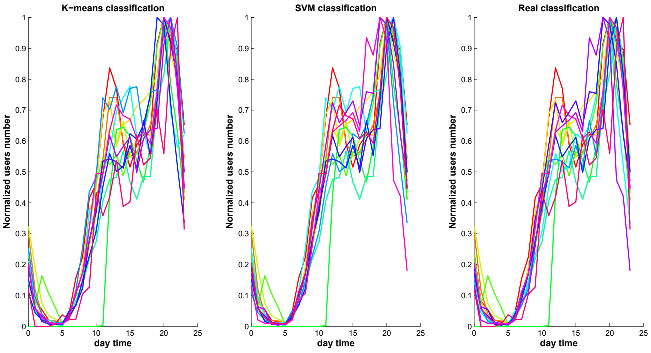


Figure 3.15: Classification visualization results for class 3

We notice that k-means results are close to SVM classification results and real ones. So, we can as well use k-means tool to construct the training set for SVM instead of preparing it by ourselves. However, we cannot use it to classify a big set of traces because it uses all data vectors to decide to compare the similarity and then decide the class of a new instance. Hence, k-means consumes more time and computational resource, unlike SVM which is more suitable for large scale data classification.

As shown, SVM is better suited to data classification, analysis and regression than K-means. The choice of SVM rather than K-means is motivated by the fact that network operators need an online and less time consuming solution to infer the daily class of their BS. K-means is a batch method that needs to take all the available data and classify them. Then we have to apply the previously explained variance coefficient to find the adequate number of clusters. So, K-means cannot be applied every time we have fresh arriving data. It is used only once to verify classification precision. Fresh online real-time data needs another classification tool, that is in our case, SVM. In terms of speed, we compared the time execution between the SVM model and the Kmeans to classify 1450 BS. SVM needs only $32 * 10^{-4}$ seconds to complete the classification while Kmeans need $29 * 10^{-3}$ seconds. So, the SVM gives faster results than the Kmeans.

Additional results are shown in figure 3.16. Figure 3.16(a) maps the classifications results on Dakar city chart where class 1 BSs are blue, class 2 BSs are green and class 3 BSs are red. The classification results allow us to divide Dakar into significant zones (Figure 3.16(b)) and also add semantic interpretation for these areas. The red-highlighted areas, for example, are almost dense with "Night-Peak" class BSs, so it can be assumed as residential areas. The green-highlighted area are dense with "Morning-peak" class BSs, it hence can stand to a business area. While the blue areas can correspond to mixed areas. The objective of the next section is to study the transitions between these areas and infers its impact on bandwidth management.

Table 3.4 resumes the average distribution of BSs w.r.t the identified 3 classes. We distinguish 2 distributions, one related to BSs mean percentage for weekdays and the second is related to weekends. We notice that the percentage of class 2 BSs (morning peak profiles) decrease during the weekend while the class 3 BS increase. This is due to fact the users tend to stay at home or in night-life location during the weekend.

Table 3.3: Performance comparison between SVM and K-means for each class

Performance	**Class 1**		**Class 2**		**Class 3**	
Classifiers	SVM	Kmeans	SVM	Kmeans	SVM	Kmeans
Precision	0.72	0.67	0.94	0.9	0.93	0.85
Recall	0.82	0.75	0.88	0.89	0.94	0.84
F1-score	0.78	0.68	0.91	0.92	0.94	0.84

Table 3.4: Classes Distribution Comparison between weekday and weekend

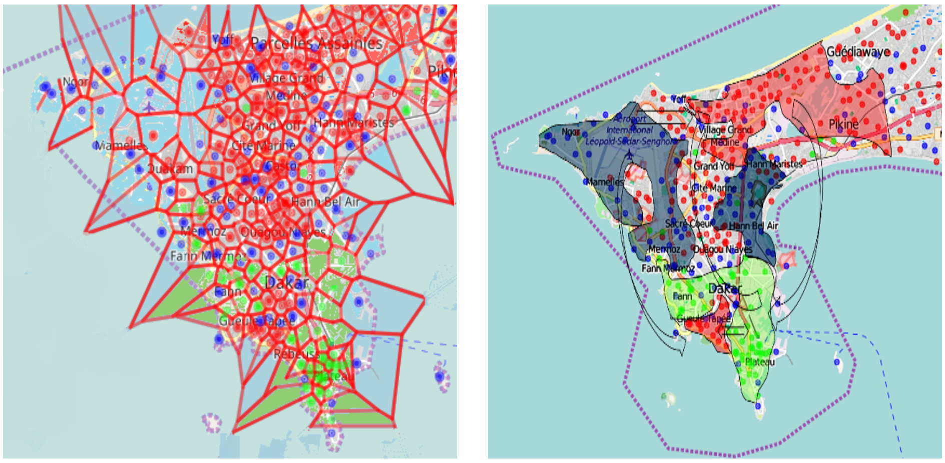
BS distribution	**Class 1**		**Class 2**		**Class 3**	
Day type	Weekday	Weekend	Weekday	Weekend	Weekday	Weekend
Percentage	26%	22%	14%	6%	60%	72%

3.5 Network load mobility analysis

In the third section, we automatically classify BS profile according to three class references in order to characterize the in-cell users consumption patterns. In this section, we investigate the user mobility behavior and study its relationship with the previous load classification in order to provide prediction for the temporal bandwidth mobility across the network.

Note that this set of traces does not give any fine grain network signaling data relative to cell handovers. We hence cannot detect phenomena such as "phantom movements" [127]. It would give us exact arrival and departure instants of each client. Each cellular dataset is different and has to be treated according to the available information.

We may consider in this study different users' mobility features like taking into account oscillations between BSs which does not mean effective users mobility but it expresses bandwidth mobility.



(a) Dakar Classified base station displaying with corresponding Voronoi cell mapping

(b) Potential transitions between classified areas

Figure 3.16: Classification results visualization over Dakar Map

3.5.1 Motivations

In order to enhance the continuity of network services, it is important to estimate with high confidence how the bandwidth demand on a BS at time t is shared among all the BSs in the following instants. In this section, we propose a method to derive a Temporal Origin-Destination Matrix (TODM) describing how bandwidth request at a BS at time t is spread over the other BSs in the next time instant. TODM is a temporal matrix because it depends obviously on the time of the day and the day of the week. Note also that it is very different from classical traffic [121] origin-destination matrices : it includes network load measurement and expresses the part of the load on BS i that moves at t towards BS j. Actually, when a user moves from a BS i to a BS j, network resources must be moved accordingly and the TODM is a tool intended for that. But, besides the continuity of service, bandwidth allocation on the BS requires to know the evolution of the bandwidth demand on this BS, taking into account also newly arrived users in the network and other ones which just left it. TODM helps to estimate the amount of bandwidth that moves from one place to another. However, it does not provide information about the total instant load locally (in each BS).

3.5.2 User mobility analysis

We start by extracting from the raw CDR dataset the daily trajectory time-series of each user. Thus, we obtain a set of time-stamped set containing, in a chronological order, the IDs of BSs where users were captured. For example, the trajectory time-series of user i Tr_i is as follows:

$$Tr_i : \{BS1, BS1, BS2, BS3, .., BS2, BS1\}$$

at the correspondent instant:

$$T_i : \{t_1, t_2, ..., t_k, ..., t_n\}$$

where t_k corresponds to the k^{th} position time-stamp. Figure 3.17 shows an example of a user trajectory made between 8am and 9am on a workday. As shown in this figure, the first BS in the trajectory is located in a residential area while the last one is located in a business area so that, it stands for an ordinary user trajectory while moving from his home to his work place. For this example, it is easy to infer the semantics of user trajectory semantic behavior, but there are many other users that are commuting between different places at different times during the day for many reasons, such as going to homes, schools, entertainment places, etc. This causes a diversity of users trajectories patterns, especially when we deal with an amount of 300.000 users. In order to understand these patterns, we start by a statistical study.

Figure 3.18 depicts the distribution of the number of places visited by users during one hour of time interval, i.e, the percentage of users in function of the number of visited places. We notice that almost 80% of users spend the time-interval in the same place. The box-plot in figure 3.19 shows the behavior of these non-moving (static) users at different one-hour time intervals. It represents the average percentage, over one week period of study, of the users that still immobile in the following time-intervals. We can notice from the plot that there is a little variation from the median values for all time intervals. This shows that most people in Dakar are non-moving while using mobile devices and this behavior is the same for a long period of the time. The remaining 20% of users represent the moving users that results on a bandwidth mobility.

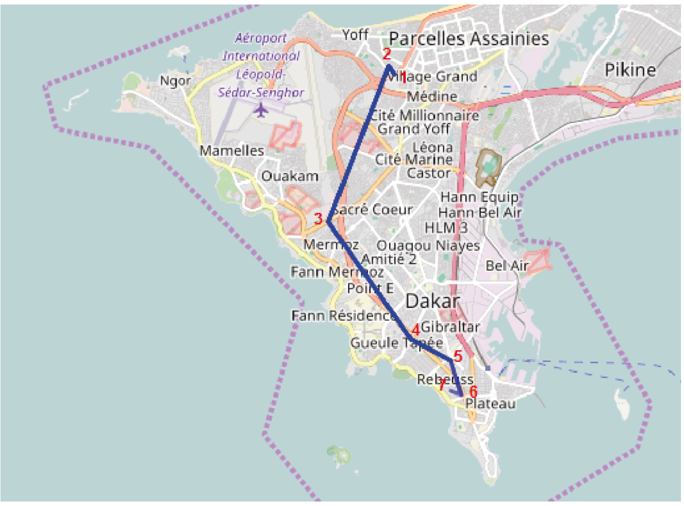


Figure 3.17: A sample of a user trajectory between 8am to 9am

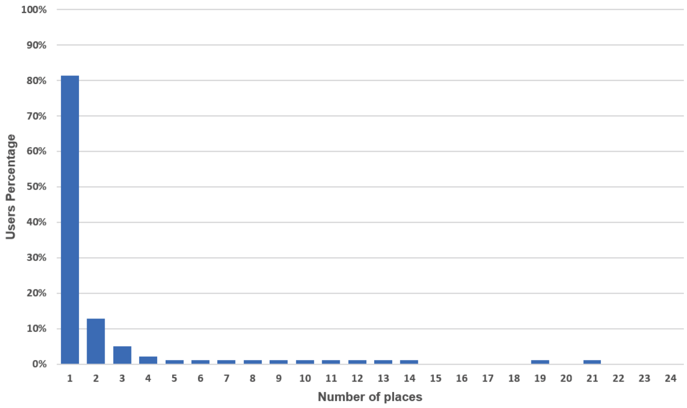


Figure 3.18: Distribution of the users in function of the number of visited places during one hour

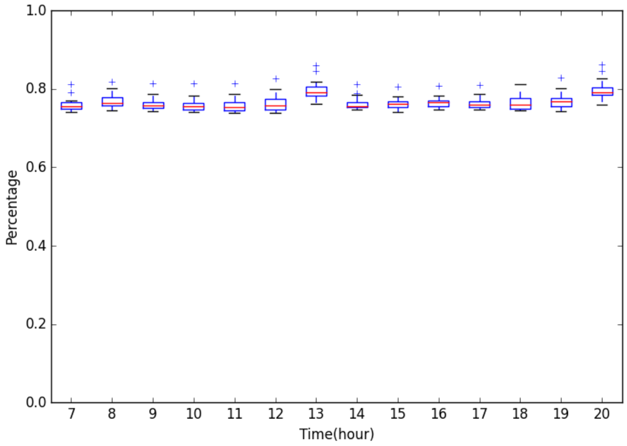


Figure 3.19: Evolution of static users percentage along the day and during one week

As we want to study the bandwidth propagation between different zones, we focus on users mobility w.r.t BS classification. We propose in the following parts a general method which infers the behavior of nomadic users and estimates the load transition between different areas. The method is general and can be easily used and adapted for other datasets.

Note that, we develop an algorithm here that catches automatically user mobility. In our traces of Dakar, it is not much seen but it does not mean that in other cities such as Paris and London, it has the same pattern.

3.5.3 Origin-Destination matrix estimation

The goal of this section is to infer the trends of users mobility while they move from a place to another in order to track the temporal propagation of the bandwidth. For this part also, we consider only the study-case of Dakar city, since it is the most crowded city in Senegal (population size in 2013 is equal to 3,137,196) with the highest cellular subscriber numbers and to also avoid some noisy patterns from rural areas.

Users mobility trends can be captured in “Origin and Destination” (OD) matrices which are extracted from users trajectory sets. They provide us with a global overview of the different flows of users’ transitions. An OD matrix contains the numbers of users moving from an “Origin” to a “Destination” BS. Origin BS can be homes while Destination BS can be work places, school, etc. Before extracting OD matrices, we must identify Origins and Destination BS from daily users trajectories sets. This step is explained in details in the next part.

Origin-Destination BS identification

We aim here to capture, from the set of daily user trajectories (extracted in section IV.A), each couple of *(Origin, Destination)* BSs to identify users transition flows. Each user trajectory consists of a chronological series of BS (geographical positions). *“Origin”* BS refers to the BS where a user starts a series of brief-time transitions (transitions occurring within few seconds) through some intermediate BS. *“Origin”* BS can also consists on the start position of a user sub-trajectory. *“Origin”* BS is easy to identify, it is generally the first BS of the whole daily trajectory or the first different BS, if it exists, after a *“Destination”* BS. *“Destination”* BS refers to a BS where users spend long time. The identification of a *“Destination”* BS belongs to a stop time threshold t_s, i.e, if the user spends a time higher than t_s in the same BS so this BS is considered as a *“Destination”* BS. We choose here t_s to be equal to 30 minutes. Figure 3.20 depicts an example of *“Origin”* and *“Destination”* BS identification process. Note that several *(Origin, Destination)* couples can be extracted from one user daily trajectory time-series. Algorithm 1 explains the procedure of extracting these *“Origin”* and *“Destination”* BS.

The output of algorithm 1 is then used to estimate the $n \times n$ origin and destination matrix M, where $M_{i,j}$ represents the number of users commuting from *Origin* BS_i to *Destination* BS_j. Figure 3.21 shows an example of a global Origin destination matrix which aggregates all users transitions during one day. Note that we neglect in this study all static users, i.e, the diagonal elements of the matrix are null.

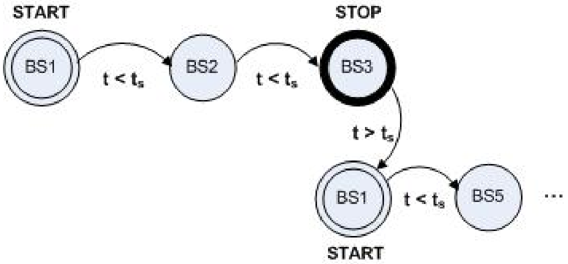


Figure 3.20: *Illustration of the "Origin"* & *"Destination"* BS identification process

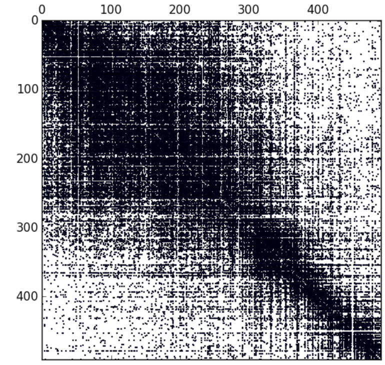


Figure 3.21: Example of a raw Origin-Destination Matrix

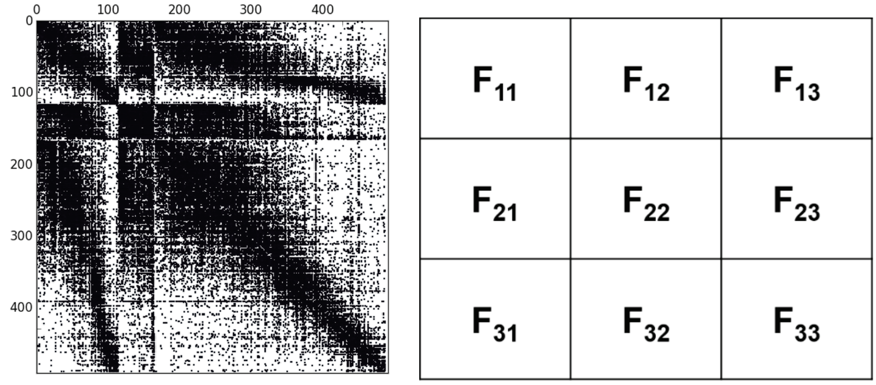
Algorithm 1 Start & Stop extraction

```
Input: Time-stamped Set of Trajectories Time-series TTS and a Stop time threshold $T_s$
Output: Time-stamped Set of (Start,Stop) tuple
procedure ODExtraction($TTS, T_s$)
    $StartBs \leftarrow pos[1]$
    $t_1 \leftarrow Tstamp(StartBS)$          ▷ Tstamp returns a position timestamp
    $j \leftarrow 2$
    while $j \leq length$ do
        $t_2 \leftarrow Tstamp(pos[j])$
        if $(t_2 - t_1) \leq t_s$ then
            $LastPos \leftarrow pos[j]$
            $j \leftarrow j + 1$
        else
            if $StartBs \neq LastPos$ then
                $[Start, Stop] \leftarrow (pos[1], LastPos)$
                $StartBS \leftarrow pos[j]$
                $j \leftarrow j + 1$
    return (Start,Stop)
```

Classified Origin-Destination Matrices

The global OD matrix depicted in figure 3.22 aggregates all transition flows in an unstructured manner, so it does not provide clear details about users mobility through the network. Thus, We can not directly infer daily user mobility patterns neither detect POI areas.

To get more details from the OD matrices, we exploit the classification results of section 3. We map the previous three classes results with the OD matrices and we re-arrange the global OD matrix rows and columns accordingly. Therefore, Class 1 BS rows are re-ordered in top rows of the matrix, then class 2 BS rows are re-ordered in the middle and finally class 3 BS rows are re-ordered in the bottom of the matrix. The same order is applied to columns. Figure 3.22(a) shows an example of a classified OD matrix where we notice clearly the separation between classified zones. We notice also that the classification permits to segment the OD matrix into 9 different blocs and each bloc aggregates transition flows of a specific type of user' mobility pattern. These blocs are illustrated in figure 3.22(b). In this figure, the notation $F_{i,j}$ specify the flows type and its direction, i.e $F_{1,1}$ bloc corresponds to transition flows between BSs belonging to the same class 1 areas, while $F_{2,3}$ bloc corresponds to transition flows from class 2 BSs areas (Business areas) toward class 3 BSs (Residential areas).



(a) Example of Classified Origin Destination matrix

(b) Flow blocs interpretation of the Classified OD matrix

Figure 3.22: Illustration of classified Origin and Destination matrices

The classified OD matrix provides more details about the transition flows across the city. These matrices aggregate flows in a spatial scale since they aggregate all transitions during the day and neglect the temporal aspects of these transitions. In the following part we propose to add the temporal scale to the classified OD matrix.

Spatio-Temporal OD matrices

Origin Destination matrices presented above capture all aggregated users commuting flows during the day. This representation permits to infer the spacial behavior of user mobility and allows to detect the *place of interest* (POI) [73, 128] within the studied city, thus, the most important places that people are moving between. But they omit the temporal scale of these flows, i.e, the temporal evolution of these transitions is not included, so the Time of Interest (TOI) cannot be evaluated. We mean by *Time of interest* here the important moments (or time) which characterize user mobility flows, i.e. when there is a transition peak between two given regions, or when a specific transition flow starts to become weaker (stagnant flow), etc.

As the goal of this study is to infer users mobility trends, not only in spatial context but also temporal scale and estimate the spatio-temporal evolution of transition flows, we propose then in this part to include the time scale to *Origin and Destination* matrices. For this need, a new dimension is added to the previous OD matrix to obtain a 3-Dimension (3D) array with $n \times n \times T$ dimension, where n represents the number of BSs (spatial dimension) and T represents the time dimension. The latter dimension is divided into T equal time-windows and each one represents an OD matrix, i.e M_t aggregates all transitions flows at time interval t, while $M_{i,j,t}$ captures the transitions flows from BS_i to BS_j at time t. The fact that we propose at section IV.B.1 to extract a time-stamped set of (Origin,Destination) couple makes the constitution of such a 3D array more obvious. This new representation of *Origin and Destination* matrix helps to estimate the instant propagation of users flows within the city and hence provides a spatio-temporal insight of bandwidth migration.

Transition probability

In the previous section, we analyze the network activity patterns in a fine-grained (or micro) scale. Based on the used dataset, we find that network resource consumption may differ depending on day time and the BS profile. In this part of the study, we aim to study the network load transitions in a macro-scale according to previous classification which provides interesting perspectives to optimize cellular networks.

Once temporal classified OD matrices are extracted, we aggregate all transition flows of each bloc to obtain an aggregated 3D origin and destination array, but here origin and destination represent an aggregated zone rather than a single BS. We then calculate the *transition probability* flows between different zones at different time interval.

The transition probability is defined as follows:

$$TP_{i,j}(t) = N_{i,j}(t)/N_i(t-1)$$

where $TP_{i,j}$ is the transition probability from source i to destination j at time t, $N_{i,j}(t)$ represents the aggregated flow of users that are moving from classified zone i to classified zone j at time t and $N_i(t-1)$ represents the aggregated users' number that were present in zone i during the previous time slot $t-1$. The latter component is evaluated as follows:

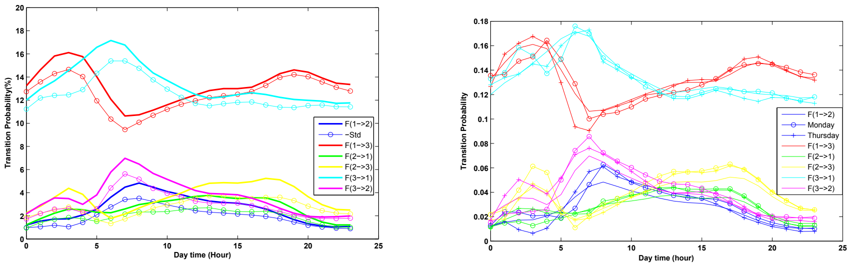
$$N_i(t-1) = \sum_j N_{i,j}(t-1)$$

Figures 3.23 shows the mean of transition probability of an aggregated data during a period of three weeks in addition to its standard deviation (figure 3.23(a)) and two examples of daily probability transition (figure 3.23(b)).

We notice from curves that the probability transition results are in accordance with the classification. For example, the curve $F_{1,3}$, which represents the evolution of probability transition flows from class 1 area to class 3 area, has low probability at the morning and then it increases at evening (from 7am). This is obvious since users tend to leave their residences at the morning (to go to work or study etc ...) and they return it back at night. Same interpretations are made for curve $F_{1,2}$. In the other hand, Curve $F_{3,2}$ shows the inverse, and this is due to the fact that class 2 zones contain BS with ”morning peak” load profile which can represents business areas.

Moreover, we can infer clearly from the figures the TOIs defined earlier. For example, at 8am there is a peak of transition of users moving from residential areas (Class 3) to business areas (Class 2) while transition flows toward residential areas ($F_{1,3}$, $F_{2,3}$) are represented with lower transition probability. Hence, extra network resource must be allocated to the business areas at this period. Besides, between 12am and 1pm the trend of user mobility starts to be inverted and users start to move to residential areas, i.e the slope of the $F_{.,3}$ transition probability curves is increasing while the complementary flow curves slope is decreasing. In the other hand, mobility flows toward residential areas reach their maximum transition probability between 7pm and 8pm. So, the bandwidth must be moved from class 2 areas to class 3 areas in order to serve this high rate demand. After 8pm, most of users are in their homes, so that their mobility is limited and that explains why all transition flows are decreasing.

We notice also that the behavior still very similar during several weeks (low standard deviation). This information is crucial for the forecast of bandwidth mobility in the future, hence allows network operators to pro-actively manage their resource allocation sharing.



(a) Mean Transition probability and its standard deviation

(b) Mean Transition probability and random examples

Figure 3.23: Transition probability between each couple of classes

3.6 Discussion

The CDRs data set used in this work does not include information about device types and low level signaling information [127]. We can imagine the presence of 'light' and 'heavy' users that don't consume the same amount of data. Figure 3.3 for example gives the average amount of a “heavy” user with a typical smartphone. All users in our study belong to the same usage profile. If this extra data is available from the core network, a small adjustment will be necessary in the pre-processing step of the framework. We hence can use our proposed framework by weighting the mobility values with different classes of cellular users (i.e. heavy or light). Moreover, we can also have a different classification per device type.

3.7 Conclusion

We show in this first chapter a general process about how to exploit real data-set of CDR to analyze the network subscribers and network usage behaviors. We propose algorithms that aim to analyze large scale cellular networks data traffic, to classify this traffic and to infer the bandwidth mobility behavior. We propose a classification model based on SVM to classify the BSs into three principal classes according to daily load profile. The SVM algorithm allows to efficiently process and classify large datasets on-the-fly. We also use the Dynamic Time Warping distance instead of the Euclidean distance which is more suitable for time series similarity measurement. We demonstrate, with multiple simulation results, the efficiency of our DTW choice. Results show also the high accuracy of the automatic classification algorithm.

We then study the user mobility behavior from a bandwidth management perspective. This helps to forecast the network load transitions and to estimate the amount of data migration, or in other term, the bandwidth mobility, between zones throughout a day.

Dynamic allocation network techniques such TV-white space spectrum allocation [19] and drone-assisted networks may exploit this bandwidth mobility information to share bandwidth accordingly between different areas in real-time (to ensure service continuity for moving users).

The sequence that we adopted in this chapter may become the skeleton of a precise and scalable methodology to analyze cellular network traffic and to dynamically plan resource provisioning on access networks. Moreover, the process we presented in this chapter will take part of a global framework, that we present later in this thesis, that allows to dynamically plan and allocate network resources.

Chapter 4

Base Station load prediction

4.1 Introduction

Most of the techniques for network resource allocation are based on a static sharing of the licensed spectrum, so that the access network can not efficiently manage instantaneous dynamics of user demands,i.e. when the number of users increases quickly, the access network starts to the reject some of them and the quality of service decreases drastically. Therefore and with the help of our field classifications, we can say that this allocation seems to have some waste for the radio resource as some base stations (BS) may not consume all the resource allocated because the low number of its attached users. In the same time, there can be some BSs where the number of users is too high and then it demands extra resources to satisfy all the users without affecting the quality of service. This hypothesis is proved by the CDR analysis and the BS profile classification of the previous chapter. Dynamic Techniques for radio resource allocation must substitute the classic ones, so that we can exploit dynamically the unused resources on a given period of time on another places and allocate them to BS when there can be a bottleneck at that moment.

Network operators can use innovative techniques to make more dynamic and efficient resource allocation policies such as machine learning techniques for on-line load prediction. Predicting the near future load of the cell allow to infer the profile class for which the BS will belong and then dynamic decision for resources allocation would be possible.

The previous chapter addresses the topic of network load profile classification and bandwidth mobility between classified areas. In this chapter, we propose a proactive techniques based on SVR to make an online prediction for the load of BS according to its history. Results from both sections combined give the precise global bandwidth dynamics.

We propose in this chapter a machine learning technique based on (SVR) to predict the load of each cell. Hence, we propose a framework that allows network operators to monitor their network resources and to provide new mechanisms that take into account the dynamics of user data demands and adapt the network resource allocation accordingly to enhance the QoS and users' QoE. Prediction outputs make using dynamic resource allocation techniques, such as TV-white space spectrum allocation [19] and drone-assisted networks, more accurate.

Moreover, on-line load prediction provides a pro-active tool that ensures a bandwidth sharing mechanisms between areas and to dynamically allocate resources for new attaching users.

The chapter is organized as follows. Section 2 introduces the support vector regression algorithm. Section 3 presents the data analysis results and the load prediction model. The comparison and performances evaluation of our model are presented in section 4 and we conclude in section 5.

4.2 Prediction models

4.2.1 Support Vector Regression

We propose a model based on Support Vector Regression (SVR) algorithm [110] for BS load prediction. Let $(x_{i,t}, y_{i,t})^N$ be a time-series training sample , where$x_{i,t}$ stands for the features vector that describes our data at timestamp t ($t \in 1, ..., N$) and for BS i, $y_{i,t}$ stands for the label of each features vector or, in our case, the instant users' density at timestamp t and N_i stands for the number of timestamps of data collected from the BS i used for training.

Primal formulation

The SVR model aims to find a linear function (or hyperplane) f which maps $x_{i,t}$ with $y_{i,t}$ in a feature space $\mathbf{F}$ (usually with higher dimension) that provides a linear projection of the data. Hence, the linear function f will be the solution of the optimization problem into the $\mathbf{F}$. The function is formulated as follows:

$$f(x) =< W^T.\phi(x) > +b\,. \tag{4.1}$$

W is a weighting vector in $\mathbf{F}$ space, b is a bias and ϕ is the mapping function corresponding to $\mathbf{F}$. To find this function, the following risk function should be minimized as stated in by this primal problem formulation:

$$\underset{W,b}{min}\ \frac{1}{2}\|W\|^2+C\sum_{i=1}^{N}(\xi_i+\xi_i^*)\,. \tag{4.2}$$

$$\begin{aligned} s.t.\ & (< W.x_i > +b) - y_i \leq \varepsilon + \xi_i, i = 1, 2, \ldots, N \\ & y_i - (< W.x_i > +b) \leq \varepsilon + \xi_i, i = 1, 2, \ldots, N \\ & \xi_i^* \geq 0, i = 1, 2, \ldots, N\,, \end{aligned}$$

where ξ_i and ξ_i^* are slack variables introduced to deal with prediction errors higher than the insensitive loss parameter ϵ and C is the penalty parameter.

The above function is called the Structural Risk Minimization (SRM) function. SVR is different from conventional regression techniques because it uses SRM, which was proven to be more efficient than Empirical Risk Minimization (ERM) used in neural networks [129] and which is equivalent to minimizing an upper bound on the generalization error and not the training error. Using SRM results in better performance of generalization and performs better than conventional techniques which may suffer from possible over fitting.

Dual formulation

To solve the quadratic programming of the primal formulation, Lagrange multipliers,(α_i), are introduced and problem formulation become as follows:

$$\begin{aligned}\min_{\alpha,\alpha^i} \; & \frac{1}{2}\sum_{i}^{N}\sum_{j}^{N} K(x_i,x_j)(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*) \\ & +\varepsilon\sum_{i}^{N}(\alpha_i+\alpha_i^*)-\sum_{i}^{N} y_i(\alpha_i-\alpha_i^*)\end{aligned} \tag{4.3}$$

$$\begin{aligned} s.t. \; & \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0, \\ & 0\le \alpha_i,\alpha_i^* \le \frac{C}{N}, i=1,2,\ldots,N\end{aligned}$$

The optimal prediction function obtained after resolving the previous optimization problem is as follows:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i,x)+b\,, \tag{4.4}$$

where $K(x_i,x_j)$ is the Kernel function and its equation is as follows:

$$K(x_i,x_j)=\phi(x_i)^T\phi(x_j)$$

Table 4.1 depicts some examples of kernel functions that are used in the literature.

Table 4.1: Examples of SVR kernel functions

Kernel function	**equation**
Linear	$K(x_i,x_j) =< x_i.x_j > +b$
Polynomial	$K(x_i,x_j) =< x_i.x_j >^d +b$
Sigmoid	$K(x_i,x_j) = tan(< x_i.x_j >) + b$
RBF	$K(x_i,x_j) = \exp(-\gamma\|x_i-x_j\|)^2$

4.2.2 Auto-regressive Integrated and Moving Average model

In order to provide a reliable evaluation of our prediction model based on SVR, we provide a benchmarking analysis and comparison between our model and other prediction technique from literature.

We choose to evaluate our SVR-based model against Auto-regressive and Integrated Moving Average (ARIMA) models which is, according to the literature , one of the most used and efficient technique for time-series forecasting.

ARIMA forecasting models is an extension for the classical Auto-Regressive and Moving Average (ARMA). The ARMA model was developped for the stationnary data prediction, wheras the ARIMA model is proposed as a generalization to the non-stationnary time-series. Moreover, BS load data are presented by non-stationary time-series that can be modeled by an ARIMA model.

The ARIMA models are generally denoted $ARIMA(p, d, q)$ where parameters p, d, and q are non-negative integers, p is the order of the auto-regressive model, d is the degree of differencing, and q is the order of the moving-average model. Given a time-series $\{X_t : 1 \leq t \leq n\}$, the ARIMA model is written as follows:

$$\phi(B)\nabla^d X_t = \theta(B)\mathcal{E}_t \tag{1}$$

where B represents the backshift operator and it is expressed as follows:

$$BX_t = X_{t-1} \tag{2}$$

while $\phi(B)$ is the auto-regressive operator, represented as a polynomial in the back shift operator:

$$\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p \tag{3}$$

and $\theta(B)$ is the moving-average operator, represented as a polynomial in the back shift operator:

$$\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q \tag{4}$$

$\mathcal{E}_t$ is the independent disturbance, also called the random error. It is assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2 (white noise). The roots of $\phi(X) = 0$ and $\theta(X) = 0$ should all lie outside the unit circle. ∇^d describes differencing operation to data series to make the data series stationary, and d is the number of differencing.

4.3 Load prediction model

4.3.1 Periodicity analysis

Power spectral density estimation models

A primordial step toward time-series prediction is to study and analyze their periodicity. Periodicity analysis is made by studying the Power Spectral Density (PSD) of the time-series.

FFT-based estimation : One of the common methods is to applying Fast Fourier Transform (FFT) that maps the time-series into the frequency domain so that we can extract the most significant frequencies that describe the BS load cycles. FFT is suitable for our periodicity analysis because it is a non-parametric method to extract periodicity and provides a periodogram that is easily interpretable. The periodogram based on FFT is expressed by the following equation:

$$S_n(\alpha) = \frac{1}{n}(\sum_{0 \neq i \leq n} (X(i)\exp(-ji\alpha))^2) \tag{4.5}$$

The issue of this periodogram is that it represents a biased periodogram of the BS time-series PSD estimation.

Welch estimation method : The Welch's periodogram [130] is a modified version of the previous mentioned method and it was proposed to come up with the bias of the FFT method. Welchs method is also called the weighted overlapped segment averaging (WOSA) method and periodogram averaging method. The method is carried out by dividing the time signal into successive blocks, forming the periodogram for each block, and averaging.

Let denote the m^th windowed, zero-padded frame from the BS_i time-series by:

$$X_m(i) = w(i)X(i + mR), 0 \geq i \leq n - 1, 0 \geq m \leq K \tag{4.6}$$

where $w(i)$ represents the window function. R is defined as the window hop size, and K denotes the number of available frames. Then the Welch's power spectral density estimation is given by the following equation:

$$S^W_{per}(w) = \frac{1}{K}\left(\sum_{0 \neq m \leq K-1} P_{X_m,M}(\alpha_k)\right) \tag{4.7}$$

where $P_{X_m,M}(\alpha_k)$ represents the FFT periodogram of the $m - th$ block. In other words, it's just an average of periodograms across time.

Base station load PSD estimation

Since the classical periodogram based on FFT provides a biased estimator of the BS load PSD, we adopt here the Welch's method to estimate the time-series PSD and analyze their periodic patterns. Welch's periodogram is an improvement of the standard periodogram spectrum estimation method that it reduces the noise in the estimated power spectrum density.

Let $\{BS_i : \{X(1), ..., X(n), n = 21024\}, i \in B_d\}$ be the set of all Dakar BS (B_d) time-series. BS_i represents the load time-series of the i^{th} BS at each time interval of ten minutes granularity, for a period of 4 months.

At first, we normalize the whole time-series of the set using the "min-max" normalization method. We then divide each BS_i time-series from the previous set into equal blocks of one week window size, i.e M is equal to 1008. Finally, we applied the welch's estimation method to infer the PSD signal of each BS_i load time-series. The normalized PSD Welch's estimation is represented in figure 4.1.

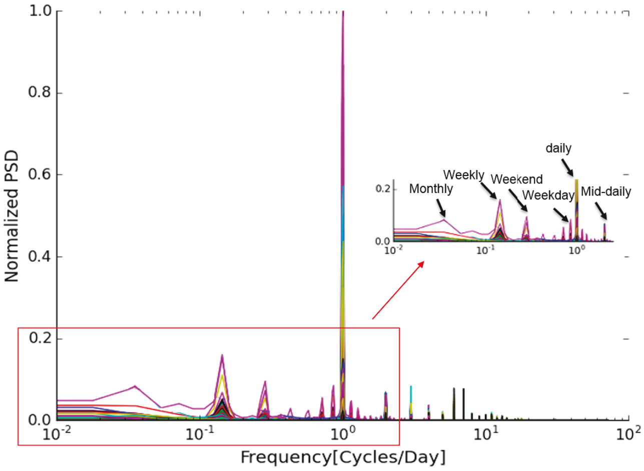


Figure 4.1: Welch's periodogram: Time-series periodicity analysis

According to this figure, we obviously notice some frequencies with high amplitude values. These relevant frequencies belong to the cycles that can model the periodic component of the BS time-series. These cycles are highlighted by the zoomed-in box in the figure and are interpreted as follows:

- Daily periodicity: corresponds to the peak at 10^0 which describes one cycle per day.
- Mid-day periodicity: corresponds to the peak at $10^{0.3}$ which describes two cycle per day.
- Weekend periodicity: corresponds to the peak at $10^{-0.54}$ which describes one cycle per weekend.
- Weekday periodicity: corresponds to the peak at $10^{-0.14}$ which describes one cycle per weekend.
- Weekly periodicity: corresponds to the peak at 10^{-85} which describes one cycle per week,i.e, 0.14 cycles per day.
- Monthly periodicity: corresponds to the peak at $10^{-1.47}$ which describes one cycle per month.

This analysis proves that BS activity has a periodicity on different scales and these information will be exploited latter to build the prediction model.

On the other hand, study in [81] shows that the BS load time-series is made with two components: one periodic component, which can be re-constructed from the previous cycles. And another stochastic component, that it is hard to model and predict and which needs advanced algorithm such as SVR model.

4.3.2 SVR-based prediction model description

For our prediction model, we use SVR to predict the load of each BS. Similarly, to the classification model, there are two steps for the prediction model: the training and the testing steps and the principles of those steps are very similar to the classification ones. The difference here concerns the features of the training vectors. For the classification, we used the daily BS load time-series which is extracted from the D4D traces data-set. For this purpose, we defined a set of five pertinent features for each BS to predict its load. The choice of features depends on the periodicity analysis results. For our dataset study-case, the periodicity analysis reveals four dominant cycles, so that we choose the following features.

Each training vectors $X_{i,t} \in \mathbb{R}$ is described as follows, $X_{i,t} = \{x^1_{i,t}, x^2_{i,t}, x^3_{i,t}, x^4_{i,t}, x^5_{i,t}\}$, where:

- $x^1_{i,t} \in \{1, 2 \ldots 144\}$ stands for the chronological order of the ten-minute time interval within a day. The choice of this feature is based on the daily cycles inferred from the periodicity analysis.

- $x^2_{i,t} \in \{0, 1\}$ indicates whether the training data belongs to weekend ($x^2_{i,t} = 1$) or a weekday ($x^2_{i,t} = 0$). The choice of this feature is based on the weekend cycle extracted from the PSD periodogram.

- $x^3_{i,t} \in \{1, 2 \ldots 7\}$ indicates the weekday standing for the training data. We choose this feature according to the weekly cycles inferred from the periodicity analysis made previously.

- $x^4_{i,t} \in \{1, 2 \ldots 52\}$ represents the chronological order of training data week. We choose this feature because some yearly behavior can occurred like holidays or seasonal periods, etc.

- $x_i^5 \in \{1, 2 \ldots N\}$ corresponds to the year of the training day. N can vary according to the number of years with which to train the prediction model. In our case we have just a one year of traces.

- t stands for the timestamp of the time-series data, i.e, $t \in N$ as defined earlier.

- i stands for the index of the BS concerned by the prediction.

The choice of these features is made based on the periodicity analysis presented previously in this chapter, where we find daily and weekly cycles. In other cases, datasets may contain more than four cycles and then SVR training model must be adapted. Note that we neglect the mid-day and monthly cycles because are less representative in the periodogram and has less weight on the prediction compared to other cycles.

Like the classification model, each training vector must be associated with its corresponding label. In the prediction case, the label y_i represents the load of the BS at the time interval x_i^1. To find the optimal value for parameters C, γ (for RBF) and ϵ we used the cross validation technique. Then we adjust the SVR model with these parameters and test its efficiency.

4.3.3 model tuning and prediction Results

To evaluate our prediction model, we test it with the D4D traces dataset and considered the Mean squared Error (MSE) as a criterion for the evaluation. The mean squared error is measured between the predicted BS instant load and the real data.

Choice of kernel function

Table 4.2 shows a comparison between the mentioned kernel function. We use here the MSE as a criterion for comparison. All compared kernel functions are used with the same training dataset.

The table results prove that SVR with RBF has a larger accuracy compared than other kernel functions. These results validate our choice of RBF as kernel function for our prediction model.

Table 4.2: comparison between SVR kernel functions

Kernel function	**MSE**
RBF	$4.3 * 10^{-4}$
Linear	$44 * 10^{-4}$
Polynomial	$24.06 * 10^{-4}$
Sigmoid	$43.8 * 10^{-4}$

model parameter optimization

We adopt the grid-search algorithm combined to the cross-validation algorithm to select the optimal combination of SVR and RBF hyperparameters, such as the penalty parameter C and RBF γ parameter as well as defining the optimal training dataset size.

The principal of grid-search algorithm is to set the parameters of C, γ and a steps size, so as to form a grid by search points. Thus, the SVR model is trained for each pair of hyperparameters within the grid. The best training model is selected in manner that it provides the lower MSE value. An improved grid-search algorithm is used in this chapter, the search is performed in multiple phases. At first, we set the parameters range and an initial higher steps size for coarse search, and then evaluate the MSE for each step and select a smaller parameters range that provide the lowest MSE. The following phases consists on fine ranging the search by training the SVR model with the values range selected in the previous phase but with smaller steps size. In our algorithm the step size is divided by ten at each phase. Thus, the computing cost can be shortened while the prediction accuracy can be improved. The steps of the grid-search algorithm are detailed as follows:

1 Set the initial range values of the couple (C, γ)

2 Set the initial steps size

3 Train iteratively, according the steps size, SVR model with each couple of (C, γ) chosen from the respective ranges and calculate the MSE at each step.

4 Select the range of parameters (C, γ) that provides lower MSE

5 Divide the steps size by ten and re-train the SVR models according the new selected ranges and steps size

6 Repeat the steps (3), (4) and (5) until the convergence to a lowest MSE.

To validate this tuning parameters algorithm, we evaluated against different training dataset using the cross-validation method (explained in the previous chapter). Although, for the SVR prediction model, it is important to respect the order of samples, instead of the classification task where we divide the training set into k-folds and evaluate the classification model against a mixed selection of folds. Hence, for the SVR model, we use a sliding-window cross validation, which accounts for data order and preserves the notion of history data for predicting future data.

We initialize the grid-search algorithm with the following parameters range $[0, 1]$ and $[0, 100]$ and steps size 0.1 and 1 for γ and C parameters respectively. Figure 4.2 depicts an example of the evolution, during the grid-search selection algorithm, of the MSE in function of γ parameter using three weeks as training data size. The figure shows that the optimal value at the convergence of the algorithm is $1.4 * 10^{-3}$.

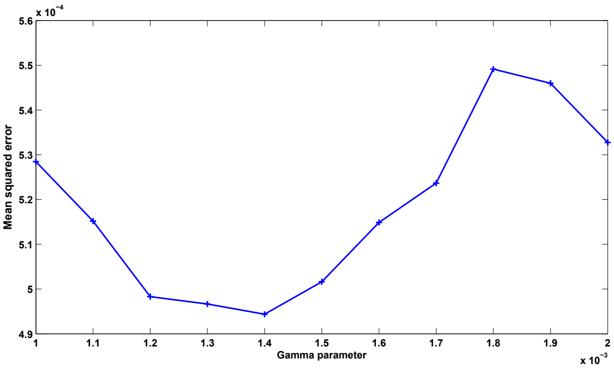


Figure 4.2: Variation of MSE in function of γ parameter

Figure 4.3 shows also the evolution of MSE in function of C parameter during the execution of the tuning parameters algorithm. The figure shows that the the optimal C is equal to 85.

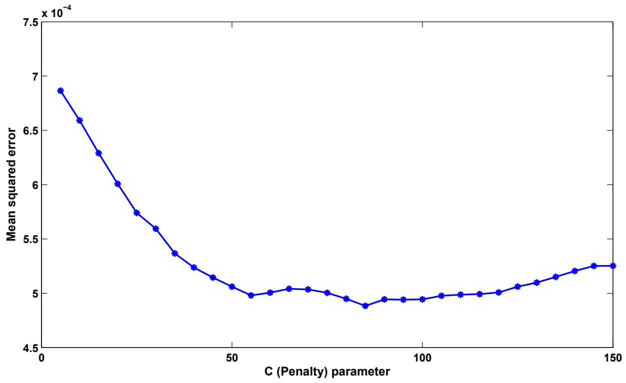


Figure 4.3: Variation of MSE in function of γ parameter

Once the SVR model optimal parameter are selected, we aim to select the best training data size that provide better prediction performance. Figure 4.8 shows the evolution of MSE in function of the training data size. We notice that from eleven weeks, the MSE start to be stable and the deviation, using higher size, is negligible. Then, we choose to fix the training window on eleven weeks. This choice will be used for prediction in this chapter and other chapters.

4.4 Benchmarking analysis and performance evaluation

Once the optimal parameters of the SVR model are selected as well as the optimal training set size is inferred, we train our prediction model we the selected hyperparameters and evaluate in this section its performance.

Figures 4.4, 4.5 and 4.6 show the comparison between the real data (in red) and the predicted one (in blue) for constant load, day-peak load and night-peak load class profiles respectively.

We notice clearly from these figures that the prediction is very close to the real load profile of the BS, with an average MSE equal to $4.82 * 10^{-4}$ for the three different class of weekly time-series. This MSE value is a global value for the whole set of BS time-series.

We notice also that the SVR-based model predicts correctly the class profile, i.e for example, in figure 4.5 the weekdays BS profile stands for the day-peak load class, whereas on weekends and especially on Sundays, the profile changes and corresponds to the constant load profile.

Despite the accurate prediction of the whole bulk of time-series, the issue with this prediction models that it under-predicts the peaks. In order to come up with this issue (especially noticed in figure 4.4), we used the max error (ME) metric as an optimization criterion to adjust the SVR parameters and the prediction results is presented in figure 4.10. We notice that the peaks are well predicted whereas the predicted load is over-fitted in almost the time which may cause an over-sized network. So we can adopt both metric (The MSE and the ME) according to the objective of the network operator: either predicting with high accuracy the load out of peak hours or predicting the load spikes.

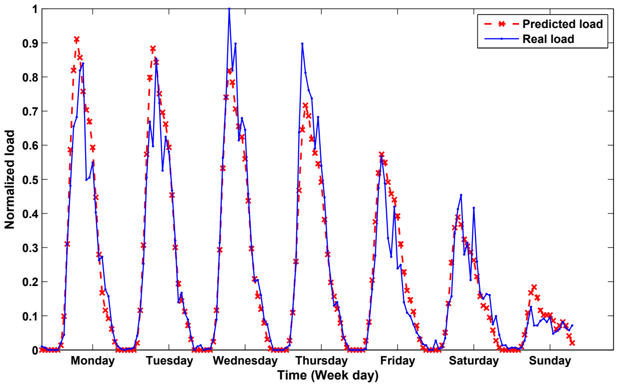


Figure 4.4: Comparison between real BS load and SVR prediction based on ME: "Morning-Peak" profile use-case

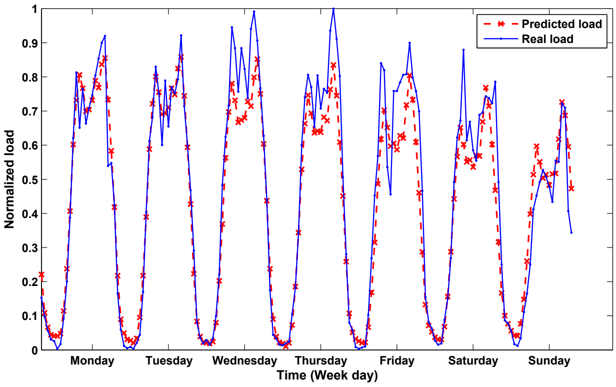


Figure 4.5: Comparison between real BS load and SVR prediction based on ME: "Almost-constant" profile use-case

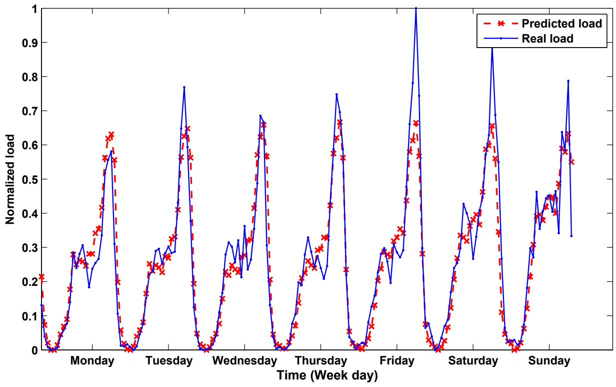


Figure 4.6: Comparison between real BS load and SVR prediction based on ME: "Night-Peak" profile use-case

We go further in our study and we provide a benchmarking analysis. Figures 4.7 and 4.8 give a comparison between our SVR prediction model and several ARIMA models with different orders. We use a deterministic pattern extracted from the same SVR training time-series to fit the ARIMA models. Figure 4.7 shows the evolution of MSE in function of ARIMA order p (A comparison between SVR model and ARIMA models with different order). We can notice that MSE decreases as the ARIMA order increase and become stable from the $10th$ order. The figure also shows that the ARIMA MSE still higher than SVR MSE especially for lower order ($k \leq 5$) where the difference is more than 35%. Figure 4.8 shows a comparison between SVR models and ARIMA model using order equal to 10, using different size for training data. We notice that the SVR' MSE decrease when training set size increase and it still always lower than ARIMA MSE. Whereas, ARIMA MSE still almost constant and the difference error is about 30%. We conclude from these two figures than SVR is more efficient than ARIMA models even with higher orders.

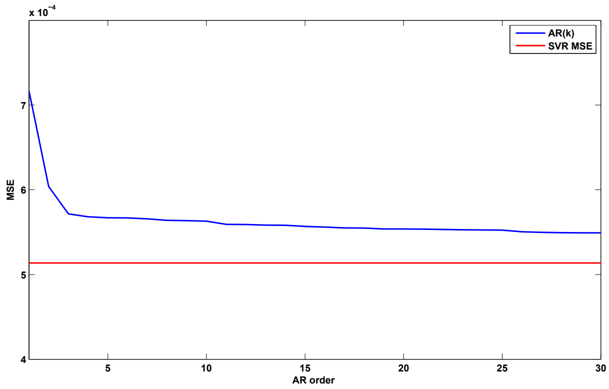


Figure 4.7: MSE comparison between ARIMA models and SVR

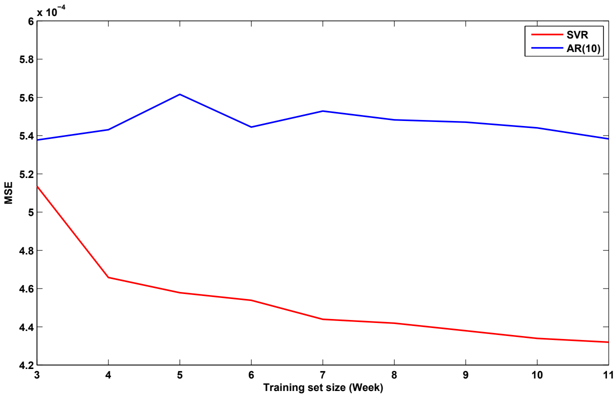


Figure 4.8: Training set size evaluation: comparison between ARIMA(10) and SVR

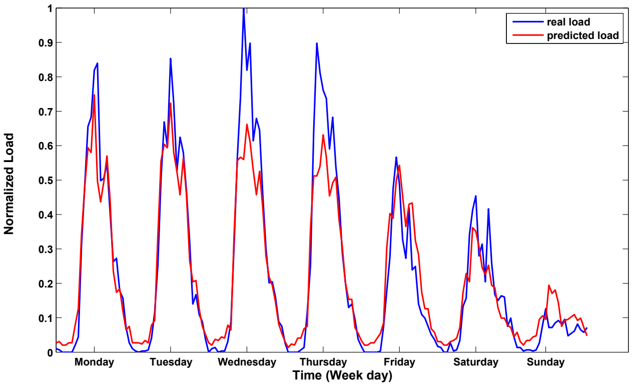


Figure 4.9: Comparison between real BS load and SVR prediction based on MSE

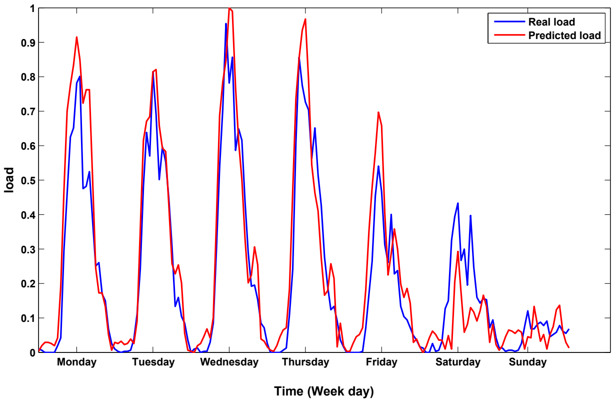


Figure 4.10: Comparison between real BS load and SVR prediction based on ME

4.5 Conclusion

We design in this chapter a prediction algorithm to perform online and local load prediction for each BS. Our proposed model is based on SVR. It helps to complete the framework with the adequate tool to predict the total cell load. we validate our algorithm with the D4D dataset and we evaluate its performance against an ARIMA model. The obtained results show the high accuracy of our model compared to the other prediction method.

The model that we present in this chapter, as well as the previous chapter, will be combined into a framework that allows to analyze cellular network traffic and to dynamically plan resource provisioning on access networks. But before, we exploit the SVM and SVR based models to incorporate a real bandwidth allocation scenario in a wireless mesh networks with fault-tolerance ability.

Chapter 5

Fault-Tolerant and Dynamic Planning for Wireless Mesh Networks

5.1 Introduction

In the previous chapters, we present a methodological data mining process that , firstly, analyze real data-sets of cellular network call detail records (CDR), classify the base station profiles into three relevant classes as well as the inference of the bandwidth mobility. The process allows also to predict efficiently the base stations' load.

We exploit in this chapter that process to present our framework for network dynamic planning. This chapter will demonstrate an example of direct application for the framework on a real-world architecture of wireless mesh network.

Firstly, We validate our framework on an innovative architecture that combines WMN with cellular femto-cells (3GPP based) to deploy cellular services for small MVNO in developing countries or for special periodic mass events. Wireless mesh networks (WMN) [4] are an easy and low cost alternative for network operators to provide a high speed connectivity access to some areas where installing a cellular access network is difficult. Thus, WMN can extend cellular network femtocell-based coverage areas with a backhaul internet dorsal and then satisfy more users anywhere.

The framework is made with the proposed classification and prediction algorithm proposed in previous chapter, as well as the planning algorithm. The planning algorithm is based on a multi-integer linear programming model, that takes as input, the results of the previously presented algorithm to deliver a dynamic WMN planning decisions and optimize the initial main topology.

Furthermore, WMN reliability constitutes a major concern for researchers. In this context, we propose, in addition to the main topology optimization, a dynamic optimized backup topology in case of mesh node failure. Hence, fault-tolerance is also supported by our framework.

The rest of the chapter is organized as follows. Section 2 presents the framework and the context of the study. Section 3 presents in details our proposed network planning tool and the optimization problem. Section 5 presents the simulation and planning results with various scenarios and we conclude in section 6.

5.2 Topology and context

5.2.1 LCI4D network architecture

Our contribution constitutes a part of a research and development project; LCI4D (Low cost infrastructure for development). LCI4D is a French project, which aims to deploy, as the name indicates, a low cost wireless network infrastructures in developing countries (like those in Africa). The objective of the project is to provide areas, where no network infrastructure is already installed, with a high-speed cellular network using femto-cells, interconnected through a mesh topology. The project proposes a core network for the cellular architecture that does not conform to the standards found in the 3GPP.

The project architecture is presented in figure 5.1. It is essentially based on femtocells, using an LTE base station with a small coverage, offering high speed access. The access network is composed of the following parts:

- An "inter-home" backhaul based on a wireless mesh network. The backhaul mesh routers are equipped, in addition to the mesh interface, with a second interface (wireless or wired) to ensure the connection with the femtocell-based access network part.

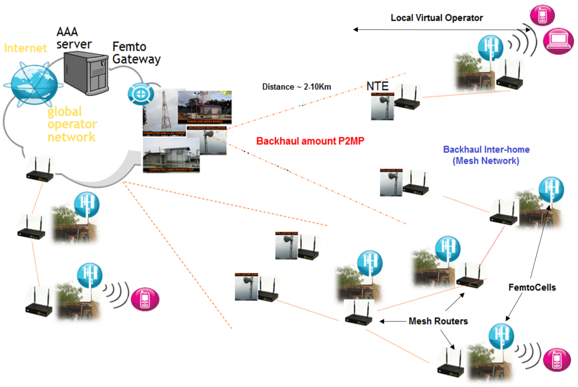


Figure 5.1: LCI4D Network Architecture

- An "Amount backhaul" which supports the "inter-home" backhaul. It ensures a point-to-multi-point link with mesh routers providing them with high speed connectivity with the core network. This part of the architecture replaces the classic wired backbone usually used in wireless mesh networks.

In our contribution we focus on the inter-home backhaul part and we propose a dynamic planning solution for its wireless mesh network.

5.2.2 Wireless mesh network

The project proposes to deploy an inter-home network layer with wireless mesh nodes as a backhaul relay to support the main access network made with multiple femtocells. The technologies defined by the 3GPP (i.e. X2 interface) and in IEEE 802.11s [4] can be used. The latter presents a low-cost alternative for broadband network interconnection with high speed connectivity but may suffer from interference issues. Wireless mesh nodes ensure a multi-hop connectivity based on routing protocols like AODV [131], OLSR [132], BATMAN [133] (an open source routing protocol which operates not only on OSI layer 3 but also on layer 2) etc. Note that in this work, we are not focusing on mesh routing issues as we focus on mesh network planning tasks and energy efficiency, but we show that our model also optimizes the routing paths while always guaranteeing a better network efficiency.

On the other hand, Mesh nodes also provide other functionalities such as dynamic configuration, neighbor discovery and self-healing. Mesh nodes adopt self-healing techniques in order to select alternative paths if one path fails. But an important question still remains: how can the network react if one of the installed mesh nodes fails and the available active nodes and paths are inefficient to serve the required data traffic and its routing to the final destination?

In our contribution we present an answer to that question by proposing a dynamic planning solution that allows the selection of an optimal main topology made of mesh nodes to serve each femtocell (FC) demand, and guarantees at the same time the reliability of the network if one mesh node fails by proposing an optimized backup topology.

5.2.3 Fault-Tolerant inter-home backhaul

LCI4D has the main objective of providing emerging countries in Africa with a low-cost wireless access network topology guaranteeing high speed connectivity based on wireless mesh routers and femtocells. Some of these countries can have bad meteorological conditions or insecure zones that may result in damaging the wireless mesh routers installed. Therefore, one of the challenges for the operators on this project is how to ensure the reliability of the network in case of mesh router failure.

In our work, we propose a fault-tolerant planning model that optimizes the main topology even when one of the mesh routers fails. Thus the model guarantees the service stability in case of node failure. Note that Optimizing under fault-tolerance constraints also serves to avoid lengthy path routing when node failures occur.

We assume that only one mesh router can fail during a time interval T_t, which is a reasonable assumption since usually during that time it will be repaired or replaced, and the probability that another node fails during this time is very small, especially if the number of mesh routers covering a defined service area cannot be very large. Hence, considering this assumption will keep the cost of topology low. Allowing the failure of more than one mesh router will require the installation of additional nodes and increases the total cost, but can easily be integrated into our model, albeit at an increase in the computational complexity.

5.3 Wireless mesh network planning tool

We assume that, within the service area, there is a set of potential wireless access locations where femtocells (FCs) are installed. Let F be the set of all the FCs installed in the service area and $r_{i,t}, r^k_{i,t},\ i \in F, k \in M, t \in T$, represent respectively, the rate demands of each FC for the main topology (i.e. with no failures) and its rate in the case of failure of mesh router k.

5.3.1 Network planning tool scheme

In previous chapters, we proposed machine learning models that are able to enhance the performance of the network and guarantees an efficient energy-saving for the system. The output of these algorithms allows to define dynamic network resource allocation strategies such as turning on/off remotely the network access nodes or to allocate additional channels [19] to the system in order to enhance the connectivity.

We are focusing in this work on the first case, where the tool must take as input a set of wireless nodes with the capability to be turned on or off. We propose to test the performance of our tool on a topology as proposed on LCI4D architecture, more precisely on the wireless mesh backhaul.

We propose in this chapter the network dynamic planning framework whose the scheme is shown in figure 5.2. The framework includes, in addition to the network classification and load prediction algorithms, a planning algorithm component that is presented in this chapter. We implement for this part an optimization algorithm which, given a set of candidate mesh routers, selects an optimal number to be active during a given time interval T_t and then the network is able to serve the variable demand of femtocells to be managed dynamically. Hence, the tool permits to a network composed of a set of candidate mesh routers. The algorithm proposes also a proactive solution in case of mesh router failure. The planning algorithm considers in parallel the traffic profiles, delivered by the classification and prediction algorithms, in order to deal with the users' demand dynamics during the day so that it provides a more realistic optimization. The optimization algorithm is explained in detail in the ”optimization model” section.

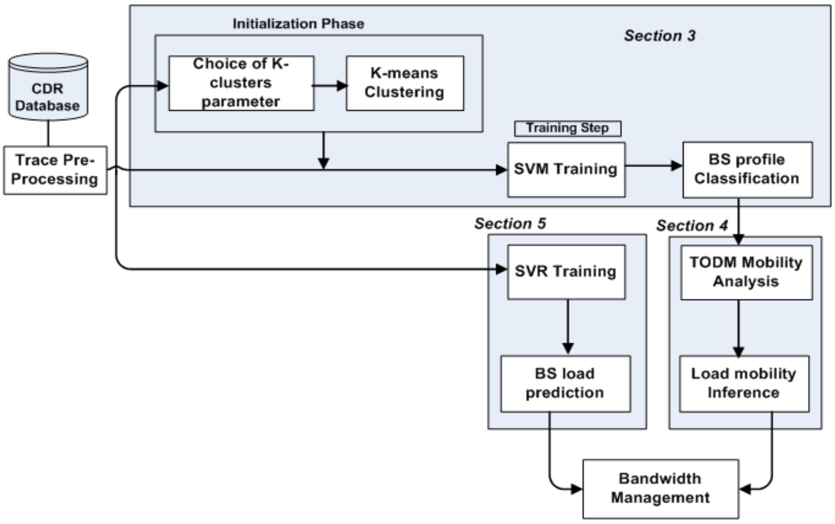


Figure 5.2: Framework scheme and methodological process

5.3.2 Optimization model

The planning algorithm component included in our network planning framework is based on an optimization algorithm that can be applied to different types of wireless networks, so it remains useful for different situations and topologies. In this chapter, we test it with a wireless mesh backhaul network proposed in the LCI4D architecture. The model not only proposes a dynamic optimization for the planning and placement of the main topology Mesh Routers, but also a dynamic backup topology in case of mesh node failure.

The model takes as input the locations of the initial main topology’s Mesh Routers (MRs), Gateways (GWs), and femtocells (FCs). It also takes the load profiles which indicate the femtocells data request rate in Mbits/s at each time interval (this information is provided by the load predictor and BS classifier components). Finally, it takes other wireless nodes parameters like transmission power, noise level, path loss exponent or maximum link capacity in the case where the interference is neglected. The output will indicate the state of every mesh router (active or not), thus the optimal number and placement for main and backup topologies. It indicates also the state of each link and the flow transmitted through it. The possible links in our topology are presented as follows:

- GW-to-MR: link between a gateway and a mesh router.
- MR-to-MR: link between two mesh routers.
- MR-to-FC: link between a mesh router and a femtocell.

Note that we suppose that there is no possible link between a gateway and a femtocell. Note also that we only consider the downlink traffic, since loads over uplinks are often lower and that downlinks' optimization is still valid for uplinks.

Decision variables

In this part we present the problem decision variables for the main and backup topologies. Decision variables for the main topology are as follow:

$$dR_{i,t} = \begin{cases} 1 & \text{if } MR_i \text{ is active at } T_t \text{ ; } i \in M \\ 0 & \text{Otherwise.} \end{cases}$$

$$dGR_{i,j,t} = \begin{cases} 1 & \text{if a link is active between } GW_i \text{ and } RS_j \text{ at} \\ & T_t;\ i\ \in G\ ;\ j\ \in M \\ 0 & \text{Otherwise.} \end{cases}$$

$$dRR_{i,j,t} = \begin{cases} 1 & \text{if a link is active between } MR_i \text{ and } MR_j \text{ at} \\ & T_t;\ i\ \in M\ ;\ j\ \in M \\ 0 & \text{Otherwise.} \end{cases}$$

$$dRT_{i,j,t} = \begin{cases} 1 & \text{if a link is active between } MR_i \text{ and } FC_j \text{ at} \\ & T_t;\ i\ \in M\ ;\ j\ \in F \\ 0 & \text{Otherwise.} \end{cases}$$

where G, M, F are respectively the index set of gateways, mesh routers and femtocells. Backup topology decision variables are similarly defined as above with the following notation: $dR^k_{i,t}$, $dGR^k_{i,j,t}$, $dRR^k_{i,j,t}$ and $dRT^k_{i,j,t}$ where $k \in M$, represents the index of failed MR at time interval T_t. It is assumed that there is an MR failure detection mechanism that identifies the failed MR.

In addition to defining decision variables corresponding to node (or links) states at different time slots, we also define other decision variables corresponding to data flows over each active link. These variables are defined as follows: $fGR_{i,j,t}$ and $fRR_{i,j,t}$ correspond respectively to the link flow from node GW_i to node MR_j at time T_t and the link flow from MR_i to MR_j. Similarly, we define the decision variables of link flows for the backup topology as $fGR^k_{i,j,t}$ and $fRR^k_{i,j,t}$.

These flow variables may depend on the SINR level of each link and the requested rate $r_{i,t}$ $i \in F$; of each femtocell. This point is explained in the next subsection.

Topology constraints

In this part, we begin by defining constraints related to topology issues. These constraints allow us to identify when an MR_i should be turned to active mode or when to activate the link between two different nodes.

The following constraint ensures that when a link is active between a gateway GW_i and a mesh router MR_j at time T_t, node MR_j should be active:

$$dGR_{i,j,t} \leq dR_{j,t};\ \forall i \in G\ ;\ \forall j \in M\ ; \tag{5.1}$$

If at T_t, the link between MR_i and MR_j is active, then these two mesh routers should be in active mode and thus, we can define the following constraint to ensure that:

$$dRR_{i,j,t} \leq 1/2\ (dR_{i,t} + dR_{j,t});\ \forall i,\ j \in M\ ; \tag{5.2}$$

When a link is active between MR_i and FC_j at instant T_t , the femtocell FC_j should be attached to the selected mesh router, and the latter's active mode must be turned on. This is ensured by the following constraint:

$$dRT_{i,j,t} \leq dR_{i,t};\ \forall i \in M\ ;\ \forall j \in F\ ; \tag{5.3}$$

As our model allows fault-tolerance for the main topology, we should define as well topology constraints for the backup topology when an active MR node at T_t fails. Similar to the previous constraints, the following constraints correspond to the backup topology where $k \in M$ designate the failed MR.

$$dGR^k_{i,j,t} \leq dR^k_{j,t};\ k \neq j\ \forall\ i \in G\ ;\ \forall k,\ j \in M\ ; \tag{5.4}$$

$$dRR^k_{i,j,t} \leq 1/2\ (dR^k_{i,t} + dR^k_{j,t});\ k \neq i,j\ \forall k,\ i,\ j \in M\ ; \tag{5.5}$$

$$dRT^k_{i,j,t} \leq dR^k_{i,t};\ ;\ k \neq i\ \forall k,\ i \in M\ ;\ \forall j \in F\ ; \tag{5.6}$$

As each active femtocell can be attached to only one MR, we define the following constraints that ensure this fact and guarantee the sending of the FC requested data through one MR link:

$$\sum_{i \in M} dRT_{i,j,t} = 1;\ \forall j \in F \tag{5.7}$$

$$\sum_{i \in M} dRT^k_{i,j,t} = 1;\ k \neq i \forall j \in F,\ \forall k \in M \tag{5.8}$$

Flow balance constraints

Flow constraints guarantee the conservation of data flow, from the gateways to all users, distributed over the different links of the network and ensure that these flows are sufficient to serve the demand of all the femtocells.

Flow balance at gateways The following constraints guarantee that all the traffic going out from all the Gateways deployed on the network should be equal to the sum of the data rate requested by all the installed femtocells:

$$\sum_{i \in G} \sum_{j \in M} fGR_{i,j,t} = \sum_{l \in F} r_{l,t} \tag{5.9}$$

$$fGR_{i,j,t} \leq Q.dGRi,j,t;\ i \in G, j \in M \tag{5.10}$$

where Q is evaluated as follow :

$$Q = \sum_{l \in F} r_{l,t}. \tag{5.11}$$

The same flow balance is applied to the backup topology as follows:

$$\sum_{i\in G}\sum_{j\in M} fGR^k_{i,j,t} = \sum_{l\in F} r^k_{l,t}\ \forall k \in M \tag{5.12}$$

$$fGR^k_{i,j,t} \leq Q^k.dGR^k_{i,j,t}\ \forall k \in M \tag{5.13}$$

where Q^k is defined similarly to Q.

Flow balance at MR nodes: Flow balance at an MR guarantees that the total incoming and outgoing amount of data must be equal. For example at the MR_j, the sum of the incoming traffic from all the gateways installed in the network and which have a direct link with the mesh router, and the incoming traffic from the neighboring MRs is equal to the sum of the traffic transmitted to its neighboring MR and its attached femtocells. The following constraints ensure this:

$$\sum_{i\in G} fGR_{i,j,t} + \sum_{n\in M n\neq j} fRR_{n,j,t} = \sum_{n\in M,n\neq j} fRR_{j,n,t} + \sum_{l\in F} r_{l,t};\ \forall j \in M \tag{5.14}$$

$$fRR_{i,j,t} \leq Q.dRRi,j,t,\ \forall i,j\ \in M \tag{5.15}$$

For backup topology, the flow conservation is ensured by the following constraint:

$$\sum i \in G fGR^k_{i,j,t} + \sum_{n\in M n\neq j} fRR^k_{n,j,t} = \sum_{n\in M n\neq j} fRR^k_{j,n,t} + \sum_{l\in F} r^k_{l,t};\ \forall j \in M \tag{5.16}$$

Where $fRR^k_{i,j,t},\ i,j\ \in M$, is defined similar to that in equation 15 for the main topology.

FLow balance at FCs These constraints mean that the traffic coming to the FC should be delivered by just one MR, i.e the FC_i is attached to just one mesh router, and the amount of traffic delivered by the serving MR should be equal to the full required rate r_i. This is formulated by the following constraints:

$$\sum_{j \in M} fRT_{j,i,t} = r_i;\ \forall i \in F \tag{5.17}$$

$$fRT_{j,i,t} \leq Q.dRT_{j,i,t}\ \forall i \in F;\ \forall j \in M \tag{5.18}$$

Similarly for backup topology, we should have the following constraint:

$$\sum_{j \in M} fRT^k_{j,i,t} = r^k_i;\ \forall i \in F;\ \forall k \in M \tag{5.19}$$

$$fRT^k_{j,i,t} \leq Q^k.dRT^k_{j,i,t}\ \forall i \in F;\ k \neq j;\ \forall k, j \in M \tag{5.20}$$

link capacity constraints

In the above subsections, we presented constraints related to the states of the links and nodes and the conservation of the traffic transmitted into the deployed network, but no constraints ensure or indicate the quantity of the flow over each active link. These constraints are presented in this subsection. The flow over each link depends on the maximum capacity of this link that should be calculated first.

In our model, we study two cases of mesh link capacity. In the first case, the effect of interference is neglected and the link capacity is fixed, i.e. orthogonal channel allocation is used, while the second case takes into consideration the interference, which is more realistic. For the first case we have the following constraints:

$$fGR_{i,j,t} \leq C_{i,j,t};\ \forall i \in G,\ \forall j \in M \tag{5.21}$$

$$fRR_{i,j,t} \leq C_{i,j,t};\ \forall i, j \in M \tag{5.22}$$

$$fRT_{i,j,t} \leq C_{i,j,t};\ \forall i \in M,\ \forall j \in F \tag{5.23}$$

Where $C_{i,j,t}$ is the static link capacity of the corresponding link.

For the second case, which models frequency reuse, link capacity should be calculated according to the measured interference level and SINR at each receiver mesh node. The SINR measured at a node i when receiving transmission from a node j, is calculated by the following expression:

$$SINR_{j,i} = \frac{\beta P_{j,i}}{d^{\alpha}_{j,i}(N_0 + I_{j,i})} \tag{5.24}$$

where $P_{j,i}$ corresponds to the transmission power of the node j, $d_{j,i}$ is the distance between two nodes, α is the path loss exponent, β is the antenna gain, N_0 is the noise power and the term $I_{j,i}$ represents the interference value caused by other nodes to the signal received at node i.

As we focus here only on downlinks, we will consider evaluating the SINR just at MRs nodes while receiving transmissions from other nodes whether is a mesh router or mesh gateway.

Let $\boldsymbol{I}^{(i)}$ be the set of mesh nodes (mesh routers or mesh gateways) causing interference at MR_i when receiving traffic from mesh node j (it can be MR or GW). The interference term $I_{j,i}$ can be evaluated as the sum of all transmissions made by nodes which belong to the set $\boldsymbol{I}^{(i)}$ to obtain the following expression which should be replaced in eq.24:

$$I_{j,i} = \frac{P_{l,i}}{\sum\limits_{l \in I^{(i)},\ l \neq j} d^{\alpha}_{l,i}} \tag{5.25}$$

Afterward, we calculate the maximum capacity over this link modeled with the Shanon-Hartley equation [134] and given as follows:

$$C_{j,i} = B. \log\left(1 + SINR(j,i)\right) \tag{5.26}$$

where B represents the bandwidth.

The later equation is general and allows the link capacity to be calculated without considering the activity state of interferers' nodes. We propose in our work to modify the equation by adding a new parameter, "Activity percentage" (Ap(t)), to evaluate the link capacity. As known, within a time interval T_t the mesh node can-not transmit signals over all this interval (typically it is controlled by a CSMA-like or slotted mechanism), even if it is active, but it uses a specific time slot to deliver data traffic. Thus, the effect of the interference over the current link can not be calculated for the whole interval, and the "activity percentage" allows us to determine the real transmission percentage of the mesh during the given time interval. This parameter indicates the global probability of interferers' node transmission during T_t, then, it is proportional to the requested rates of femtocells. Therefore, the link capacity becomes as follows:

$$C_{i,j,t} = B.\log\left(1 + Ap(t).SINR(i,j)\right) \tag{5.27}$$

After calculating the maximum capacity over each link, we are now able to define the constraints related to the maximum flow that can be transmitted over the link. This maximum flow is limited by the transmission power, interference level and link distance, i.e. the maximum flow assigned to the link from the GW_j to RS_i at time T_t is limited by the maximum capacity $CGR_{i,j,t}$ of this link. A similar notation is used for all the other links and the constraints that ensure the upper bound are given as follows:

$$fGR_{i,j,t} \leq CGR_{i,j,t};\ \forall i \in G;\ \forall j \in M \tag{5.28}$$

$$fGR^k_{i,j,t} \leq CGR^k_{i,j,t};\ \forall i \in G;\ k \neq j \forall j,k \in M \tag{5.29}$$

$$fRR_{i,j,t} \leq CRR_{i,j,t};\ \forall i,j \in M \tag{5.30}$$

$$fRR^k_{i,j,t} \leq CRR^k_{i,j,t};\ k \neq i,j;\ \forall i,j \in M \tag{5.31}$$

$$fRT_{i,j,t} \leq CRT_{i,j,t};\ \forall i \in M;\ \forall j \in F \tag{5.32}$$

$$fRT^k_{i,j,t} \leq CRT^k_{i,j,t};\ k \neq i;\ \forall k,i \in M;\ \forall j \in F \tag{5.33}$$

Due to the log operator in capacity link formula, the previous constraints become non-linear. As we are interested in keeping the system linear, we propose a piece-wise approximation to the link capacity curves. Piece-wise approximation consists in approximating the link capacity curve by a limited number, P, of linear function defined at a segment Sg_p, as we see in figure 5.3), so that the primal link capacity function is replaced by a set of linear functions for each segment $p \in 1,...,P$. The function is represented as follows:

$$CXX_{p,i,j,t} = b_{p,i,j,t} + a_{p,i,j,t}.s_{p,i,j,t};\ \forall p \in \{1,...,P\} \tag{5.34}$$

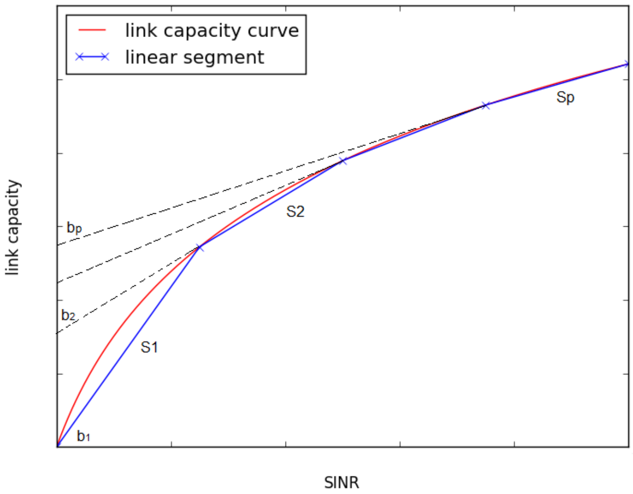


Figure 5.3: An illustration of piece-wise approximation for the link capacity curve

where $XX \in \{GR, RR, RT\}$, $a_{p,i,j,t}$ is the line slope at segment Sg_p, $s_{p,i,j,t}$ is a variable representing the SINR level and $b_{p,i,j,t}$ is a constant.

Once the approximation is made, link capacity constraints for a segment Sg_p can be written as follows:

$$fXX_{p,i,j,t} \leq CXX_{p,i,j,t};\ \forall p \in \{1, ..., P\} \tag{5.35}$$

Similar to the backup topology, the constraints become as follows:

$$fXX^k_{p,i,j,t} \leq CXX^k_{p,i,j,t};\ \forall p \in \{1, ..., P\} \tag{5.36}$$

The linear system can now be resolved at each segment with a simple MILP.

Objective function

The main objective in this section, is implementing an optimization model aiming to dynamically minimize the number of active MRs and take also into consideration different base station profiles (defined in Section 3) that deliver a variable level of rates requested by the installed femtocells during the whole period of the day. So our optimization reduces the OPEX charges by selecting at each time interval the essential number of nodes to be active and turns off the other ones.

We define the variable S_i which indicates if an MR_i is active either in the main topology or in the backup topology:

$$S_{i,t} \geq dR_{i,t};\ \forall i \in M \tag{5.37}$$

$$S_{i,t} \geq dR^{k}_{i,t};\ k \neq i;\ \forall i \in M \tag{5.38}$$

To minimize the total number of MRs that are active, $\sum_{i \in M} S_{i,t}$ should be minimized.

We also aim at reducing the number of MRs used in the backup topology. Minimizing the number of MRs used in every topology allows us to remove lengthy paths. For example, if an FC can connect to the network by going through one MR only, it is better not to use two MRs for this FC. Thus, minimizing the number of MRs in the backup topology will make an FC use the minimum necessary number of MRs it needs to connect. The variable $U_{k,t}$ designates the number of MRs used in the backup topology when MR^k fails. We have the following constraint:

$$U_{k,t} \geq \sum_{i \in M,\ i \neq k} dR^{k}_{i,t};\ \forall k \in M \tag{5.39}$$

Similarly, we aim to minimize the number of MRs used in the main topology, designated by the term V. We have the constraint:

$$V \geq \sum_{i \in M} dR_{i,t} \tag{5.40}$$

The objective function, which is denoted by *Obj*, combines the terms above. The main term of the optimization problem is $\sum_{i \in M} S_{i,t}$ because it gives, for both topology, the total number of MRs that should be active. Then, it should be given a higher weight than the other terms. The maximum value of $\sum_{k \in M} U_{k,t}$ is $|M^2|$ and the maximum value of V is $|M|$ ($|M|$=N). Thus, we assign the weight $N^2 + N$ to the term with S_i:

$$obj = (N^2 + N) \sum_{i \in M} S_{i,t} + \sum_{k \in M} U_{k,t} + V. \tag{5.41}$$

We then obtain the objective function as:

$$Minimize\ Obj \tag{5.42}$$

This model is evaluated in the next section.

5.4 Simulations and results

5.4.1 Implementation and development tools

The first part of our implementation concerns the initialization of the main network topology. We implement a topology generator tool in JAVA that takes as input the maximum number of gateways, mesh routers, femtocells and their positions. The tool also takes the power transmission of each gateway and mesh router, other parameters like the path-loss, and the instantaneous "Activity Percentage" parameter, Ap(t), defined earlier. This topology generator allows us to calculate the amount of interference caused at each receiver mesh node as well its SINR. Taking into consideration these parameters, it evaluates then each link capacity which is useful for the optimization algorithm.

For the optimization part, the formulation of the model constraints and objective functions were stated in a mathematical language and solved with CPLEX which is an optimization software package adequate for MILP problems. Note that we propose in this work a Mixed Integer Programming (MIP) algorithm with some non-linear constraints. The MIP model is an NP-hard algorithm. We then propose a relaxation for the non-linearity issues by adopting a piece-wise linear approximation to make these constraints linear so that the problem becomes a MILP (Mixed Integer Linear Programming). The relaxation of the constraints reduces the complexity of the problem and makes it easier to solve with CPLEX. Note that, the run time in average of the program for a one-day planning optimization is almost 11.37 seconds (in the context of real deployment we consider this time to be acceptable).

5.4.2 Simulation scenarios

For all simulation scenarios, we choose a simple topology where the MRs are deployed in a square grid. The set of MRs used in the topology are supposed to be outdoor WiFi routers which can provide a large coverage for the network. Figure 5.4 shows the initial deployed topology which is composed of 8 femtocells, 8 mesh routers and 2 mesh gateways.

In fact, we do not focus especially in our study on the topology as much as we focus rather on how the topology components react to data demand fluctuations. Moreover, we mentioned that we suppose that we are using outdoor mesh nodes with a high coverage range (One of the requirements of the LCI4D project for the inter-home backhaul), which are able to provide subscribers with high speed connectivity in a large area, i.e. our proposed topology can cover a service area of more than 1400 m^2 like a university campus, a shopping mall or even a residential district. Note also that the inter-home backhaul serves users data requests via an access layer made up of several femtocells which must guarantee at least a downlink throughput of 36 Mbits/s (a requirement of the LCI4D project also). Therefore, we notice from our experiments that using a huge number of mesh nodes may cause a high level of interference and consequently poor network performance, so that the throughput requirements can-not be achieved. Thus, our proposed topology model may be a good compromise. On the other hand, the optimization model that we propose is scalable and can be used for other types of topology (pure Wifi, 5G, Smart-citie, LORA, etc.) with little modification.

Planning results without interference

We present here a mesh topology where the effect of interference caused by nearby nodes is minimal (for example, due to orthogonal channel allocation) and then the maximum link capacity is kept static over the simulation time. We assume that we use outdoor mesh nodes with high range and equipped with multiple antennas, so that each node uses different channels to transmit traffic to other mesh nodes in its vicinity in addition to an extra antenna to serve femtocell clients in its vicinity. Table 1 presents some parameters specific to this scenario.

Figure 5.5 represents an example of an input load profile which represents the instantaneous data rate requests in Mbits/s over each one hour time interval. This profile type is applied in this scenario to all femtocells in the topology. Figures 5.6 and 5.7 show respectively the planning results and the energy gain of this scenario. We notice

Table 5.1: No interference scenario parameters

Description	Value
MR-MR link capacity	60 Mbits
MR-FC link capacity	45 Mbits
GW range	600m
MR range	300m
Energy consumption	15W

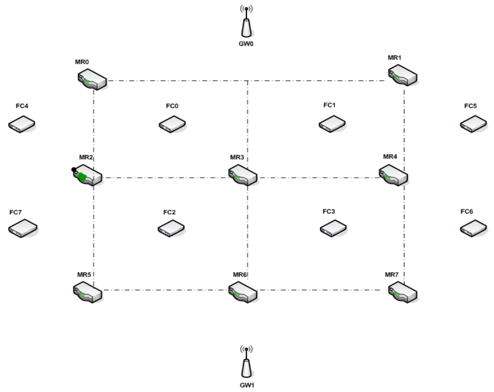


Figure 5.4: Main network topology

that the optimal number of mesh routers to be active also follows the data demand fluctuation indicated by the used traffic profile. The topology reaches a maximum number of active mesh nodes equal to six at peak hour while during very low request time interval we need only two MRs.

With this dynamic planning, the energy consumption also differs from the classic static planning where mesh nodes continue working all day and the network consumes the maximum amount of energy. To study the dynamics of energy consumption we use the following equation to determine the energy gain at each time interval: $G_t = (1 - P_t/P_f)$, where G_t represents the energy gain at interval T_t, P_t represents the power consumption by the total number of active MRs at time interval T_t and P_f represents the full energy consumption of all the MRs installed in the service area.

Figure 5.7 depicts the energy gain evolution throughout the day. The total average energy gain of this traffic profile is equal to 58.3%. The results show also that this evolution is inversely proportional to the traffic profile with a low energy gain at the peak hour and a high energy saving during low data requirements, especially between mid-

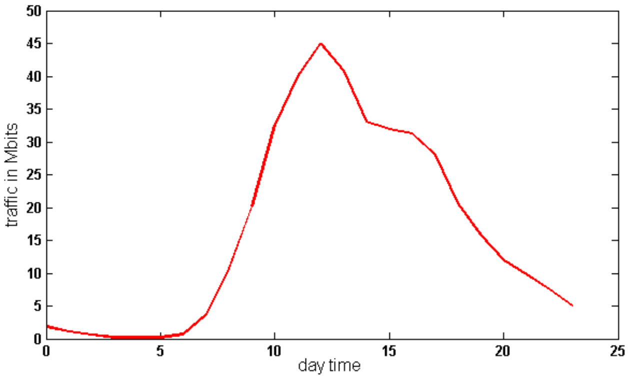


Figure 5.5: Traffic profile (demand in Mbits/s)

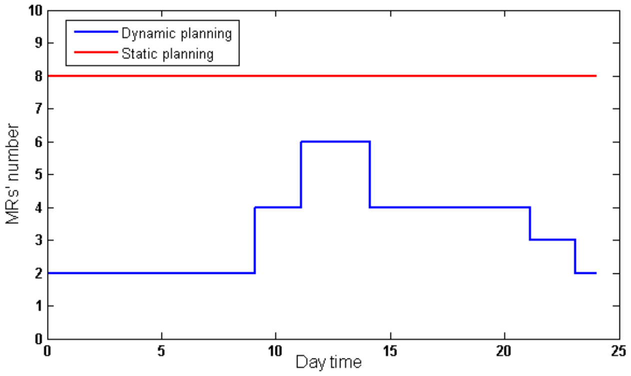


Figure 5.6: Optimal number of mesh routers

night and 8am.

Planning results with interference

In the previous section, we present planning results in the best radio conditions where interference has no effect on link capacity, but in reality, even using multiple antennas with dynamic channel assignment techniques, the interference may be reduce but is not completely canceled. Thus, a mesh router may share the same channel within a transmission time slot, with at least one of mesh routers in its range.

In this part we use the same topology presented in figure 5.5 and take into consideration the interference and the piece-wise approximation to measure the maximum link capacity. We assume that only the mesh routers installed within one unit grid may cause interference to the receiver mesh router. Unlike in the previous scenario, in this case

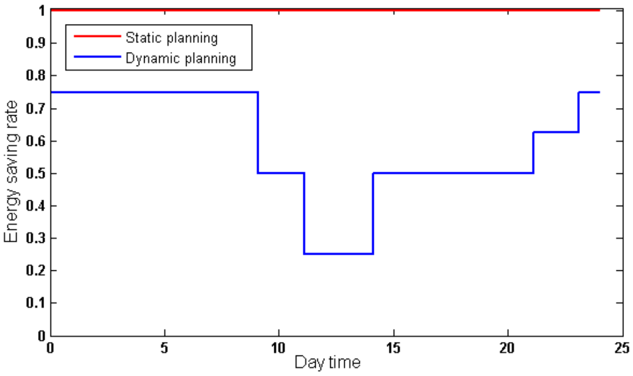


Figure 5.7: Energy gain

we fix the transmission power of mesh nodes.

Homogeneous demand profiles We suppose in this part that the set of femtocells installed in the service area has the same traffic profile type and we show the planning results for each traffic profile case. To simplify the problem, we normalized these profiles to obtain three traffic profile references which represent the percentage of data rate requests of each femtocell, then these profiles are multiplied by the maximum rate requested (in Mbits/s) by femtocell users. Note also that the traffic profiles used in our study has a granularity of one hour. Figure 5.8 shows an example of three types of traffic profiles that represent an instant data requests in Mbits/s at each time interval with a maximum data rate equal to 36.6 Mbits/s.

Homogeneous data rate requests

In this scenario, the transmission power is set respectively to 30dbm and 15 dbm for gateways and mesh routers. We also set the maximum data requests rate to 36.6 Mbits/s which is the same as all femtocells at each time interval.

Figure 5.9 shows respectively the planning results of the main topology respectively in case of “Morning-peak” profile, “constant load” profile and “night-peak” profile. These results show how dynamically the planning tool reacts with a variable traffic demand throughout the day.

We notice that the common characteristic of these three profiles is

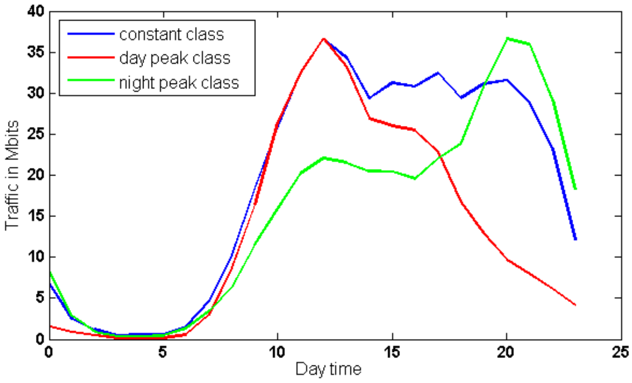


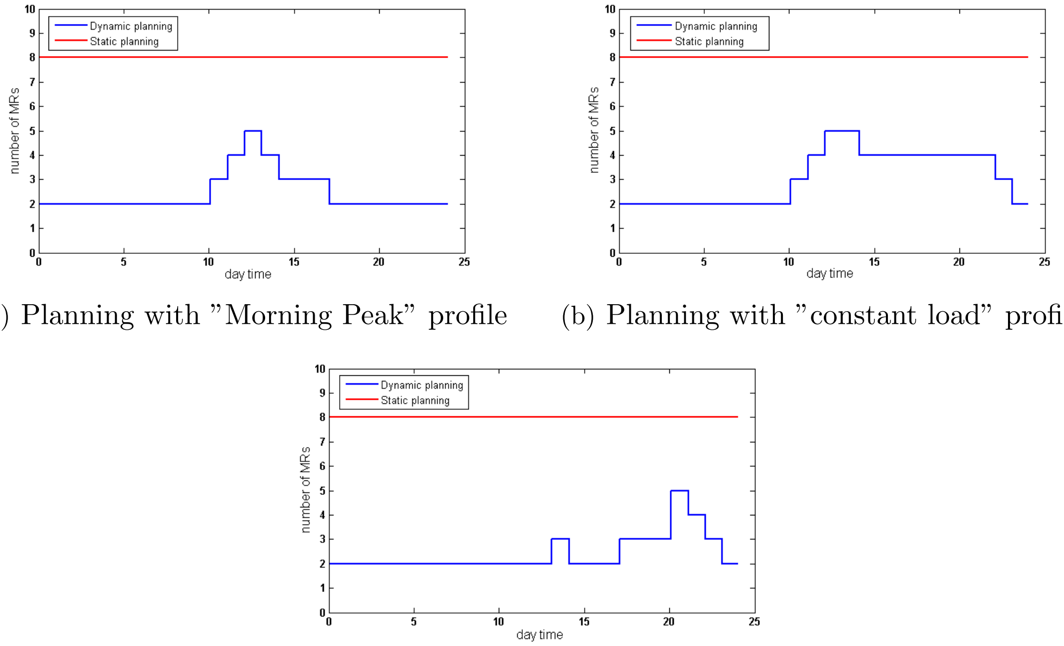
Figure 5.8: Traffic profiles

that between midnight and 10 am, only 2 active MRs must be active to serve all the femtocell demands and then we can save more energy (75% of energy during this period). Otherwise, during the peak hours of each traffic profile, just 5 out of 8 MRs installed are needed to be active. It is clear also from these figures that the evolution of active MRs number fits closely the traffic profiles trends for the three cases, i.e. the maximum number of active MRs occurs at each profile peak time (midday and 10 pm), while outside this period the number is reduced according to the demand.

The fact that all profiles have the same maximum data requests, lets us conclude that the ”always loaded” profile needs more active MRs for the whole day compared to the two other profiles and then the average energy gain is reduced. We show later, that this issue can be compensated by the two other profiles when they co-exist in the same service area.

Figure 5.10 shows the optimal topology at each profile peak hour. We notice that only MR_i, $i \in \{0, 2, 4, 5, 7\}$ should be active in the main topology. We notice also that each of the following RS: MR_2, MR_4 and MR_7 serve two FCs and MR_0 (respectively MR_5) serves just one FC.

Our planning algorithm also proposes a backup topology when one MR fails. The middle and right graphs in figure 5.10 show the optimal backup topology at the peak hour, with the same data request rates as the main topology, respectively when MR_0 and MR_7 fail. For both cases, MR_1 becomes active in the place of the failed MRs to handle



(a) Planning with "Morning Peak" profile (b) Planning with "constant load" profile

(c) Planning with "night peak" profile

Figure 5.9: Evolution of Optimal MRs number for homogeneous data rate

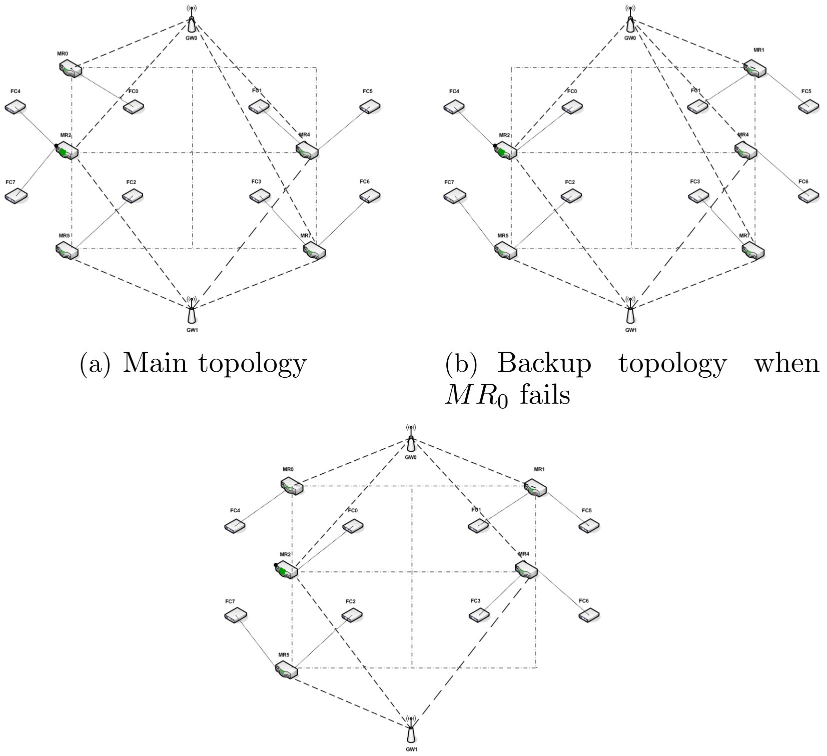
the data request of the femtocell in its vicinity; it serves FC_1 in the first case, while in the second case it serves FC_1 and FC_5. The figures also show how the femtocells' attachment change according to each case.

We notice for both cases that only 5 MRs should be active which is logical since the backup request rates remain unchanged. The same data rate service for backup topology is ensured due to the fact that there are still 3 more MRs available to be used in case of failure, which guarantees the same throughput for all femtocell attached users without the need to reduce it.

We mention that the common maximum rate of femtocells in this scenario for the main and backup topology is 36.6 Mbits/s. This common rate can not be increased since the maximum link capacity of some mesh nodes is reached and can not be exceeded due to the effect of interference. Therefore, when the interference effect is not taken into consideration the maximum rate achieves 45 Mbits/s.

Heterogeneous data rate requests

In this scenario, we aim to show a more realistic topology where

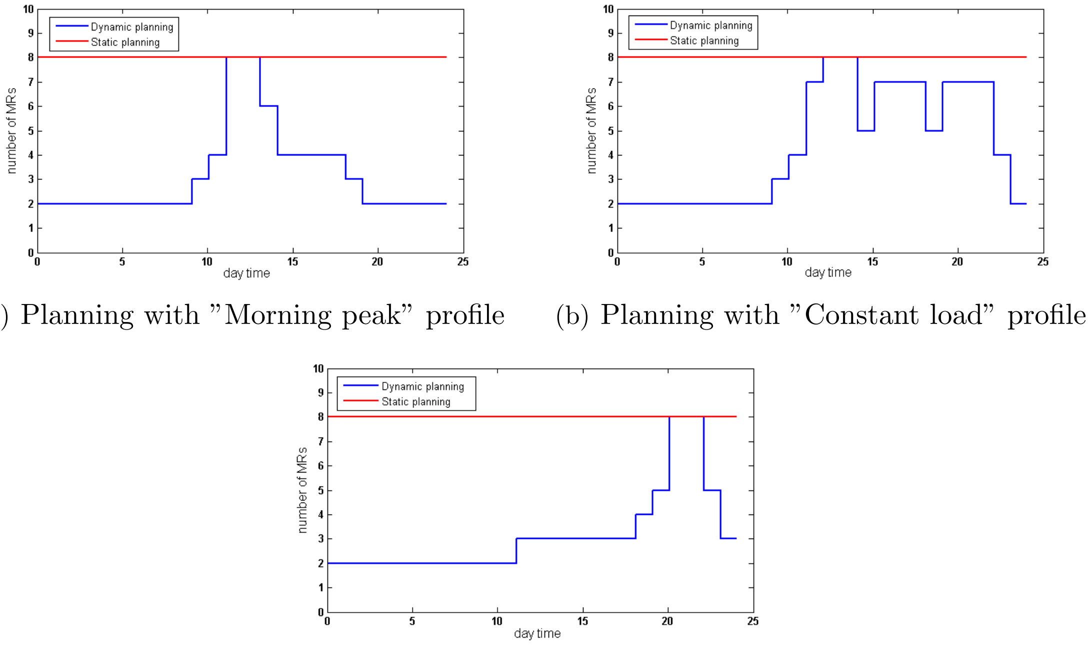


(a) Main topology

(b) Backup topology when MR_0 fails

(c) Backup topology when MR_7 fails

Figure 5.10: Topology placement results for homogeneous data rate request



(a) Planning with ”Morning peak” profile

(b) Planning with ”Constant load” profile

(c) Planning with ”Night peak” profile

Figure 5.11: Evolution of Optimal MRs number for heterogeneous data rate

maximum rate requests are not common for all femtocells. We continue using the same normalized profiles in this scenario but with heterogeneous maximum data requests. In fact, assigning different maximum data requests allows each femtocell to have different data requests compared to other femtocells in any time interval, unlike the previous scenario where femtocells have the same demand for all time intervals. Thus, each femtocell will have its own data request profile, even if the same normalized load profile is used. The maximum rates assignment is described in Table 2.

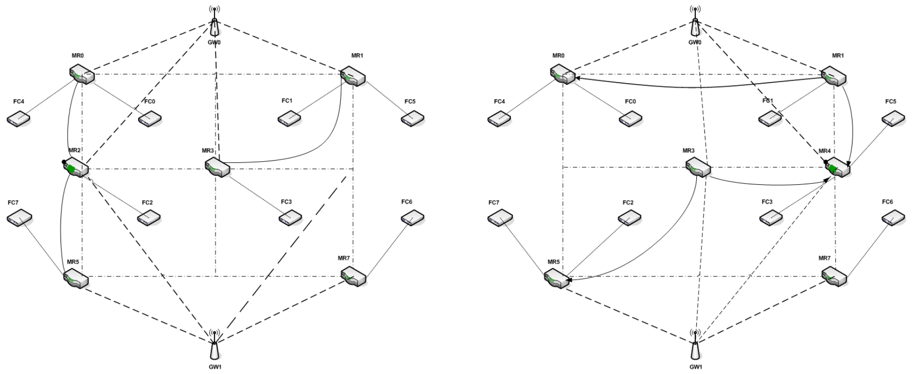
Figure 5.11 shows the variation of active MRs numbers all over the day for each traffic profile type. In this scenario, we notice that with the chosen maximum rate assignment policy, we obtain a full active topology in the peak hour of each profile, where each femtocell should be attached to its nearest MR to satisfy the high user demand, but this lasts for a brief time.

Table 5.2: Maximum requested data rate

Femtocell	rate (in Mbits/s)
FC0,FC1	40
FC2,FC3	36.6
FC4,FC5,FC6,FC7	44.5

The first graph of figure 5.12 shows an example of the optimal main topology between 1pm and 2pm of the ”morning peak” profile when the data rate request is heavy. Although not yet at its maximum, only 6 MRs are active. MR_0 and MR_1 are responsible for serving two FCs each, while the rest of the active MRs serve just one FC each. We notice also from the figure that some data flow are routed between MRs over an MR-MR active link, for example, a data flow is routed through MR_2 from MR_5 in order to assist MR_0 to serve its attached FCs, while MR_3 is assisting MR_1.

The second graph of figure 5.12 shows the backup topology when MR_2 fails within the same time interval mentioned above. Similarly, 6 active MRs are needed for this scenario. Therefore, femtocells are attached to only 5 MRs: MR_0, MR_1 , MR_4 (become active), MR_5 and MR_7 while MR_3 is needed to assist MR_4 and MR_5 and forward them the requested data flow routed from the GWs.



(a) Main topology (b) Backup topology when MR_2 fails

Figure 5.12: Topology placement results

To evaluate the impact of dynamic planning over the network, we investigate more the energy gain throughout the day. Figure 5.13 shows the energy gain of each traffic profile case.

We notice from these results that the gain is zero at the peak hours of each profile, which is logical, but this can be compensated by other periods of the day when the gain reaches a value of 75% for a large time interval from midnight to 10 am. Table 3 sums up the daily average energy gain of each profile.

Table 5.3: Profiles' Energy gain

Profile	Energy Gain
"Morning Peak load"	65.6%
"Constant load"	52%
"Night Peak load"	65.1%

We notice that the "Constant load" profile has the lowest energy gain compared to other profiles while the same requested rate distribution is assigned to the system. This low energy gain can be compensated by other service areas of the cellular network where the two other load profiles are assigned.

Finally, we conclude from this scenario that some femtocells can be served with higher rates, then satisfy more demand and enhance the QoS of their attached users. Moreover, by assigning a common rate to all FCs, the rate requests may reach a threshold that if exceeded, the optimization system may become unfeasible, but when assigning a heterogeneous rate some FCs can reach higher rate requests.

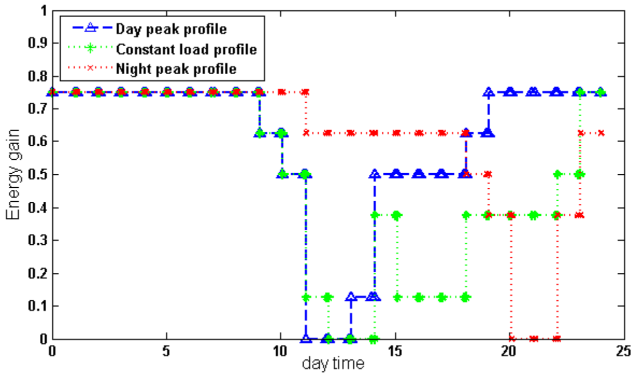


Figure 5.13: Energy gain

Heterogeneous traffic profiles Some service areas may cover a set of locations that may have different semantic annotations, for example, a campus university is composed of residential buildings, universities and entertainment areas. This diversity of locations causes an imbalance of user concentration over the time and space in such a manner that each location will be crowded by users at a specific period of the day. Thus, each location will have a specific demand profile. This part will look at such a scenario, where a specific traffic profile will be assigned to each femtocell in our topology according to the user activity it covers. Here we use the same network topology and configuration as in the previous scenario. The traffic profile assignment is described in table 4.

Table 5.4: Femtocell traffic profile types

Femtocell	Traffic profile type
FC1, FC3	”Morning peak” profile
FC5, FC6	”Constant load” profile
FC0, FC2, FC4, FC7	”Night peak” profile

Figure 5.14 shows the planning results for this scenario. We notice from the results that there are two peak traffic demand in the whole service area. During the first peak, which occurs at 12 pm, only 5 MRs should be active to serve the heavy request of FC1, FC3,FC5 and FC6. The second one occurs at 8 pm and 6 MRs should be turned on to satisfy the request of FCs having a “night peak” profile. We also notice that this topology reaches a maximum of 6 active MRs, while in the previous scenario 8 MRs must be active at the peak time, this

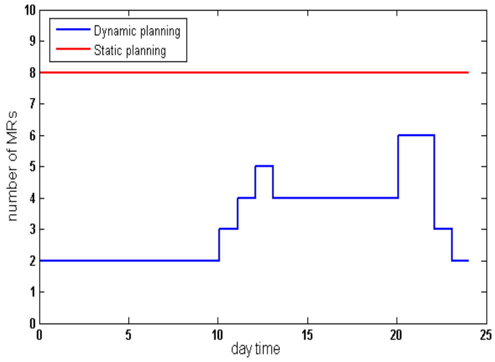


Figure 5.14: Planning results for heterogeneous profiles

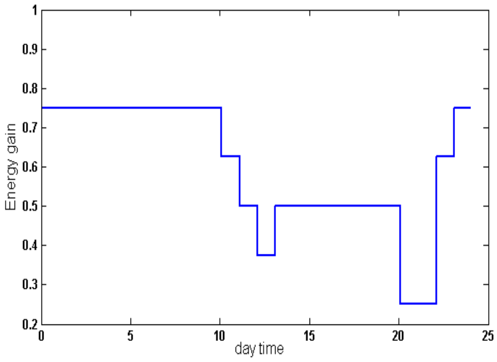
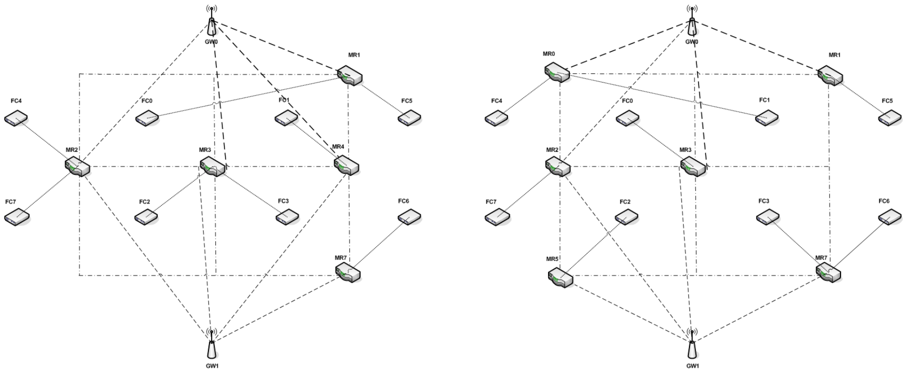


Figure 5.15: Energy gain for heterogeneous profiles

decrease in the number of MRs at peak hours is due to the diversity of data request behavior during the day over the geographical area of the network leading to a spatio-temporal load distribution over the network.

This also has an effect on the energy gain represented in figure 5.15 where the average energy gain for the whole day is evaluated at 66% which is greater than previous cases. This lets us say that the planning algorithm guarantees an efficient optimization for a service area with an heterogeneous traffic profile and thus a better bandwidth and resource management during the day in such an area.

The first graph of figure 5.16 depicts the placement of MRs that should be active at 12 pm (first peak hour). In this time period, the amount of data requested by the set of FCs: FC1, FC3, FC5 and FC6 is higher than other FCs, this justifies the fact that many MRs in the vicinity of these nodes are active. On the other hand, the second figure corresponds to the second peak hour, at 8 pm, and at this moment the set FC0, FC2, FC4 and FC7 has high data requests, which explains why more MRs are active in their vicinity. These results prove again that the dynamics of our proposed WMN planning follow the fluctuation of data requests throughout the day and deploy more resources in locations where more data is requested.



(a) planning results at midday (b) planning results at 8pm

Figure 5.16: Topology placement results for heterogeneous profiles

This scenario also shows that the bandwidth can be managed in a better way if load profiles in the service area are unbalanced; i.e for some periods of the day, the data demand is low for some FCs, and then the data flow serving these FCs will be reduced, hence more bandwidth and channels become available for the FCs with higher data requests and then the bandwidth will be divided dynamically according to the data request fluctuation.

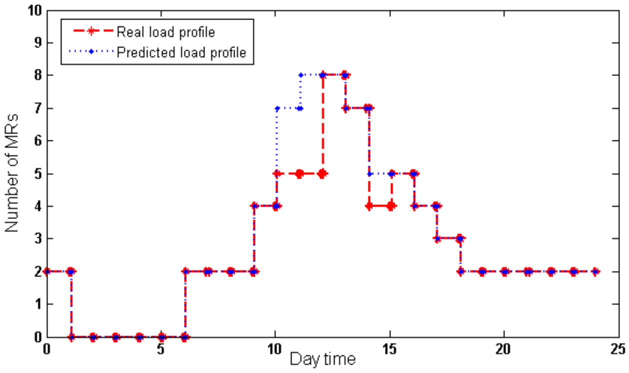


Figure 5.17: Planning results for the predicted profile against the real one

Prediction-based planning As presented in the framework scheme 5.2, our network planning tool includes a load prediction algorithm based on SVR. It provides the network operators with a proactive solution to dynamically manage their networks.

In this part, we aim to validate the prediction algorithms' performance in a dynamic network planning and management context. For this purpose, we use a real traffic profile, extracted from the Orange senegal dataset as well as its predicted load profile, resulting from SVR, as an input for the optimization algorithm in order to compare their planning results. We continue using the same topology and configurations as previously.

Figure 5.17 shows the comparison between the real and predicted dynamic planning. We notice that the planning profile corresponding to the prediction is very similar to the real one, i.e the optimal MRs number is the same for both profiles for most of the time. Therefore, there are just 3 time intervals where the planning results corresponding to the prediction exceed the real one. This similarity is confirmed also by the results of the energy gain which is equal to 63% for the predicted profile and 66.3% for the real profile. All these results encourage the use of the load prediction in order to design a wireless network dynamic planning and to provide proactive resource management.

5.5 Conclusion

We propose a dynamic planning tool for mesh networks with fault-tolerance capability considering instantaneous user demand provided by base station profiles predicted from real experiences. We propose a mixed-integer linear programming (MILP) algorithm to formulate the optimization problem that is resolved in two cases: with and without interference caused by mesh router nodes. We also propose a piece-wise linear approximation to divide the link capacity curve into linear functions with defined segments and then deal with non-linearity issues caused by interference. The model is applied to wireless mesh networks with different scenarios. The optimization results show how our model can efficiently deal with user demand dynamics throughout the day and prove that taking these dynamics into account allows an enhancement of network performance and provides better bandwidth management and energy saving, compared to the classical static planning, even with fault-tolerance considerations.

Chapter 6

Spatio-Temporal Anomaly Detection Framework for Mobile Networks

6.1 Introduction

Large Spatio-Temporal fluctuations in cellular network traffic may cause important network misbehavior and at least abnormal drops in quality of experience. These anomalous behavior may affect the efficiency of network prediction tools because they are very hard to predict earlier. Moreover, these unusual fluctuations affect certainly the mechanis of network resource allocation and drop drastically the performance of the network affecting the QoS and QoE.

In this context, an on-line dynamic prediction framework is presented to detect these network anomalies and to allow network operators to pro-actively monitor a variety of real-world phenomena and with less damage to the overall experience. It is important for a network manager to identify these anomalies; it can help him to protect, manage and gain insight into its network.

In the other hand, the expansion and development of new services and networks such Device-to-Device (D2D), Internet of Things (IoT) and eHealth applications consist a new and rich sources of data injected into the network that need extra bandwidth to be handled. These applications may change the bandwidth consumption behavior of customers and may add new dynamics and peaks on data requests. Hence, our framework helps on detecting these new and unusual data consumption pattern. It consists on analyzing these new behaviors, evaluate the readiness of existent network architectures and even prepare the integration of these new services on the network.

The framework is based on data mining techniques. It is a double-stage process which allows us to detect both spatial and temporal anomalies. The first stage, based on One-Class SVM (OCSVM) algorithm, is able to detect the cell presenting an anomaly, thus its geographical location. The second stage, based on SVR prediction algorithm, provide more precision to the framework and allow detecting the time-interval in which occurred the network anomaly. OCSVM and SVR are known to be efficient with large data processing since they do not need to use all the training dataset, but only support vectors.
On the other hand, combining both algorithms allow to implement more accurate framework for network anomaly detection, since we need to detect outliers in a spatio-temporal context. In fact, the OCSVM is insensitive to the temporal scale, but it allows to optimize the processing by detecting the anomalous base station activity within a large scale dataset. Thus the temporal anomaly detection is only performed on the detected set of anomalous activity instead of the entire large set. Moreover, using the SVR based prediction for the temporal detection avoids employing the OCSVM with a window-based approach [17] for anomaly detection for time-series. So it permits to save the computational resource since the window-based techniques divide the time-series into many sub-sequences to be processed. In addition, the choice of the window size and the manner to split the time-series is complex.

On the other hand, since anomalies can be hidden within large network datasets, we adopt OCSVM which is trained with normal data. It guarantees the ability to detect any type of outliers. We validate our framework with two CDRs datasets; one from Telecom Italia [135] and the second from Orange Senegal [12]. In fact, using real cellular network datasets increases the reliability of the model and allows to enhance its efficiency since they are richer than simulated data, used usually in other contributions, and provide various real-world patterns examples. We also compare it with Isolation Forest [136], an enhanced variant of random forest, which ensures low computational cost with linear time complexity. Random Forest is proved to be very efficient for outliers detection with less sensitivity to noisy patterns [107, 108]. Otherwise, our results outperform Isolation Forest which fails on detecting some anomalies as its performance is affected by noisy data.

We present at the end of this chapter a direct application for our spatio-temporal anomaly detection framework. We use the framework to detect anomalous bandwidth consumption pattern caused by eHealth-enabled data. Our framework helps to analyze the impact of these new sources of data on the normal behavior of the cellular networks.

The chapter is organized as follows. Section 2 presents our motivations behind proposing the STAD framework. The network outliers detection framework is presented in Section 3, while Section 4 shows the testbeds and presents the results of our experiments with performance comparisons. Section 5 presents a direct application of the STAD framework in a cellular network eHealth data enabled. Finally, we conclude in Section 6.

6.2 Motivations and context

6.2.1 Motivations

On July 2012, a general system failure occurred in the network of Orange Telecom, the historic national French operator. The breakdown was very severe and most of Orange subscribers (almost 26 million subscribers) were out of service during 9 hours. They were unable to make calls, texting or to use data services. The failure has also affected the Orange MVNOs (Mobile virtual network operators) and the interconnection with other Network operators. Due to the lack of adequate management tools that can rapidly detect this kind of anomaly, the operator was unable to avoid the blackout in its network or even reduce its impact. In addition, it was forced to deploy more resources to fix the failure and added extra cost in addition to its subscribers dissatisfaction. Investigations showed that an earlier update of a software stack was the origin of this blackout and the anomaly has not been notified by any alarm signal (Probably it was identified as a true positive alarm). From this incident and many others, it appears that there is a need to upgrade management and alarm systems with efficient automated techniques that, by the analysis of real-time traces, can detect on the fly network anomalies. These tools can also help the operators to monitor their infrastructures and more accurately manage their networks. Strong by their learning capabilities, they avoid the long and fastidious hand work to build evolving traffic profiles.

In fact, network outliers detection techniques aim to automatically identify and detect abnormal and anomalous patterns which differ from the normal behavior or may present a local deviation from the normal data. In our contribution, we address the problem of detecting outliers within radio access network. The exponential growth of mobile devices and users' demand dynamics, may create several spatio-temporal bandwidth consumption profiles [111], difficult to handle and manage by network operators. Next-generation cellular systems and cognitive networks aim to introduce more flexible techniques to better react to these dynamics. On the other hand, a major issue for network operators is to handle and detect sudden and local anomalous behavior within the network, whether it is a sharp peak of users demands (occurred during mass events for example), an abnormal brief decrease or even an non-common daily data consumption patterns. If the first anomaly type needs a fast reaction to guarantee network resilience and service survivability avoiding users rejections, the second one may be due to some technical issues of the network infrastructure that need an instant maintenance. These anomalies are also time-Dependant and need not only geographic identification but also temporal detection of the time interval in which they occur with high precision.

6.2.2 Network anomaly detection context

The application of anomaly detection may concern a wide variety of domains. Therefore, we can at first, classify its application into two principal categories:

1- Individual object applications

2- Ordered sequences applications

The first category covers domains such as image processing, genetics or medicine. It aim detect anomalous objects within a large testing dataset of objects, independently to their order or temporal aspects. While the second category is applied to a time-ordered sequences (or time-series) and it takes into consideration the time-order of the sequences. As we are interested in analyzing network time-series in this thesis, our application of anomaly detection take part into the second category.

Furthermore, the latter category can be also divided into four subcategories as follows:

- ***Sequence-Based approach:*** This approach consists on detecting an entire anomalous time-series from a dataset of testing time-series.
- ***Subsequence-Based approach:*** This category aims to detect anomalous contiguous sub-sequences within a large sequence. Generally this category takes as input a single large time-series extracted from only one source. As an example, it considers the load time-series of one base station for a whole day or week.
- ***Pattern frequency-based approach:*** This approach consists on detecting patterns in a test sequence with anomalous frequency of occurrence.
- ***Contextual anomaly detection approach:*** This approach detects a group of points within a period of time that are anomalous regarding to usual normal behavior of this period of time. As an example, the usual load of the base station installed in a beach resort in the summer is higher than other periods. So that, an peak of load out of this period may represents an anomalous behavior.

For our proposed anomaly detection framework, we choose to combine two different approaches, a first contextual anomaly detection approach for the geographic anomaly detection and a second subsequence-based approach for the temporal anomaly detection. Hence, our model consists on two stages of anomaly detection. The choice of this methodology is explained along this chapter.

6.3 Spatio-Temporal outliers detection model

The network anomaly detection model that we propose is a two stage anomaly detector. It is able to detect the BS with anomalous behavior (geographic anomaly) in the first stage and it goes further by detecting the time-interval when the anomaly occurred. As depicted in Figure 6.1. The first stage, based on One-class SVM [137] algorithm, aims to detect geographic anomalies. In case of anomaly detection, the second stage detects the anomaly temporal scale. It uses support vector regression (SVR).

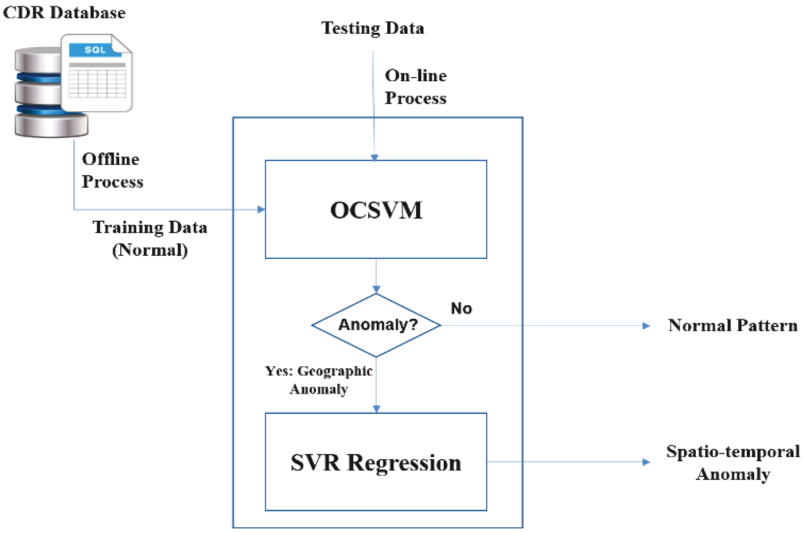


Figure 6.1: STAD: Spatio-Temporal anomaly detection framework scheme

One-Class SVM outliers detection

Anomalous or abnormal network activity patterns are not recurrent within network datasets. Thus modeling a machine learning algorithm to automatically recognize them may not be efficient. Even training a supervised classification model with both normal and outliers data will be a hard task since we need anomalous data as well as normal data. Since anomalous data may have several patterns, two-class or multi-class training models will only be able to detect the trained anomaly patterns Therefore, training a machine with only normal data, that is more available in the dataset, is more straightforward and convenient for our framework in the first stage. We propose to use the One-Class SVM classifier trained with normal network data to detect the outliers.

One-class SVM is a semi-supervised variant of SVM proposed by Scholkopf et *al.* [137] for outliers detection. One-class SVM attempts to find the decision boundary that achieves the maximum separation between the normal training samples and the origin (supposing that anomalies are close to the origin). To achieve that, it maps the data into a higher dimensional space using a transformation function $\Phi(.)$. The formulation of OCSVM model is a way different from standard SVM and SVR presented in chapters 3 and 4 respectively. The formulation aim to solving at first the following primal optimization problem:

$$\min_{\omega,\xi,\phi} \quad 1/_2||\omega||^2 + 1/_{\nu l}\sum_{i\leq l}\xi_i - \rho \tag{6.1}$$

$$s.t. \;\; (\omega.\Phi(x_i)) \geq \rho - \xi_i, \;\; \xi_i \geq 0 \; .$$

where ν is a regulation parameter, ξ_i is a slack variable for the i-th pattern. The constraint verifies that the pattern x_i belongs to normal data. To solve the quadratic programming of the primal model, Lagrangian multipliers, α_i, β_i are introduced to obtain the following dual problem:

$$\max_{\alpha} 1/2 \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(x_i, x_j) \tag{2}$$

$$s.t. \sum_{i=1}^{l}\alpha_i = 1, \;\; 0 \leq \alpha_i \leq \frac{1}{\nu l}.$$

where $K(x_i, x_j) = (\Phi(x_i)).\Phi(x_j)))$ is the kernel function. The solution of the problem is $\omega = \sum_{i=1}^{l}\alpha_i.\Phi(x_i)$ and the decision function is as follows:

$$f(x) = \sum_{i=1}^{l}\alpha_i K(x_i, \; x) - \rho.$$

The pattern x is considered as normal if $f(x) > 0$ and vice versa. For our model, we train the One-class SVM with the extracted normal network activities time-series using an RBF as kernel function that has the following equation:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|)^2 \tag{6.2}$$

Previous experiments (in chapter 4) show that RBF kernel function is much efficient with non-linear patterns as well network time-series used in our work to validate the model. We use cross validation technique with a testing dataset to fix the best OCSVM parameters, such as γ, ν and ξ.

In our model, the patterns x will be the base station load time-series extracted from the CDR datasets.

Outliers time detection

Once the spatial anomaly is detected, the anomalous time-series are handled by the second stage in order to detect precisely the time-interval of the anomaly. This stage is relevant especially in case of local anomaly that occurs in a limited subsequence of the BS time-series. We propose for this module, an algorithm based on SVR. Using history data as a training set, SVR aims to estimate an optimal function which fits the maximum of input samples.

In our study, we train the SVR model with the current BS past normal data in order to predict afterwards the appropriate future of daily BS users occupancy data. We continue here on using an RBF kernel function for the same reasons mentioned earlier. A testing step is crucial to fix the best parameters of the SVR model such as C (penalty parameter of the error term), γ (RBF kernel function coefficient) and ϵ (tolerance for termination criterion). We use for that normal patterns (testing data) compared to the predicted one using the MSE as criterion and best parameters that guarantee the minimal MSE are chosen. We use here the same tuning and optimization parameters methodology that we used in chapter 4.

Once the testing step is validated with the best combination of parameters, SVR model is applied now to anomaly patterns. We also compare the anomalous data with the predicted one using MSE measurement between each point of both time-series. Hence, a new time-series $E(t)$ is obtained with the temporal evolution of the MSE. Then, anomalous time-interval k is detected if $E(k)$ is higher than an appropriate error threshold chosen according each pattern.

6.4 Experiment testbeds and result comparison

6.4.1 Testbed datsets description

The anomaly detection framework we propose aims to be general, so that it can be applied to any dataset of users' activity times-series. The algorithm is embedded in a tool that takes as input, massive dataset records. We tested our method with data extracted from two different call detail records (CDRs) datasets: the first one is provided by *Telecom Italia* as part of the big data challenge context while the second dataset is provided by Orange for the *Senegal D4D challenge* context [12].

Milan dataset testbed

The *Telecom Italia* dataset is an open multi-source aggregation of weather, news, electricity and telecommunication data. We focus on the last data source which provide call, Short Messages exchanges (SMS) and data usage. The data provide, in a geographical square cell aggregation, the telecommunication activities of Milan city. In fact, the city is divided into 10,000 squares having a 235m x 235m size. Each square aggregates all network traffic data of the base stations installed into it, thus, no more information are given about the actual deployment of these base stations. The raw data contains information about subscribers' communications, such as the amount of in/out call, in/out SMS and data. It indicates also for each user' interaction, the geographical location (square ID), the time when it occurred and the country code. The collection of these data covers the time period from November 1st, 2013 to January 1st, 2014 and the measurements are aggregated in time slot of ten minutes granularity. To exploit these data, we extract at first a set of time-series for each square which indicates the daily evolution of each users' network interaction. We focus in our performance study on two use-case testbed studies:

- San Siro stadium use-case which represents an anomalous peak of consumption for some hours during the weekends. The time-series dataset used for for validation contains 4320 data points whose 72 are anomalous

- End of year festivities use-case which contains some anomalous patterns that occurred during the last week of the year (from Christmas to new year eve). The data-set used for this use-case contains time-series of users consumption in Duomo square, one of the most attractive touristic places in Milan. The dataset contains almost 42,000 data points whose 5184 are anomalous.

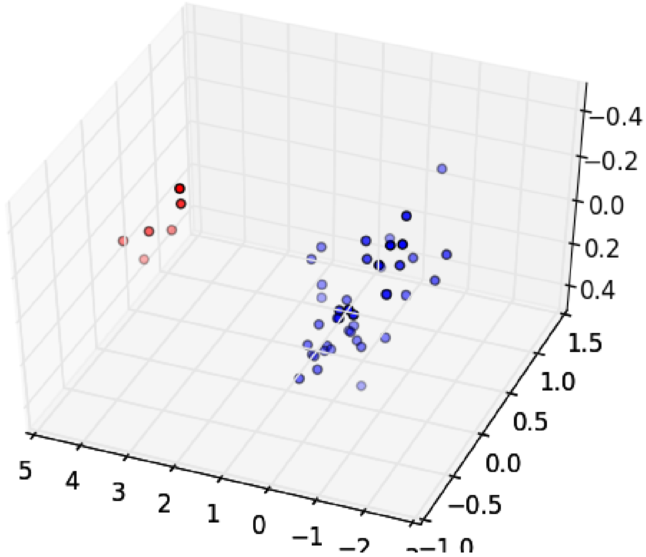


Figure 6.2: Example of an ACP study for Milan Christmas Eve anomalies testbed

Dakar dataset testbed

The second dataset we use to validate our framework is extracted from the *D4D-Senegal challenge* dataset [12]. This dataset collects call detail records of phone calls and SMS of about 9 million users for the year 2013. The dataset is divided into 3 sets: one set contains the antenna-to-antenna traffic for 1666 antennas on an hourly basis, another contains one year of coarse-grained mobility data at district level and a last one contains fine-grained mobility data on a rolling 2-week basis for a year with bandicoot behavioral indicators at individual level for about 300,000 randomly sampled users. We extract from the last set information allowing us to monitor the instant number of users in each cell and thus, estimate the instant instant BS load (or users' occupancy) for the whole day. Unlike the previous dataset, D4D CDRs set provides user' communication information at fine-grained space scale, i.e, at each base stations instead of geographic square aggregation. Hence, we can monitor each cell apart and detect network outliers precisely. Similarly to *Italia Telecom*, We extracted from these CDRs a dataset of times-series relative to Dakar city, whose each one describes the daily evolution of users occupancy within a BS cell. As Milan data-set use-case, We also focus here on two use-case testbed studies:

- Friday noon use-case which represents an anomalous decrease of consumption during some hours. The time-series dataset used to validate our framework with this testbed contains almost 17,000 data points whose 5760 are anomalous

- Tuesday the 5th of February use-case which present a sudden decrease of users consumption at night. The data-set used for validation contains contains almost 43,000 data points whose 720 are anomalous.

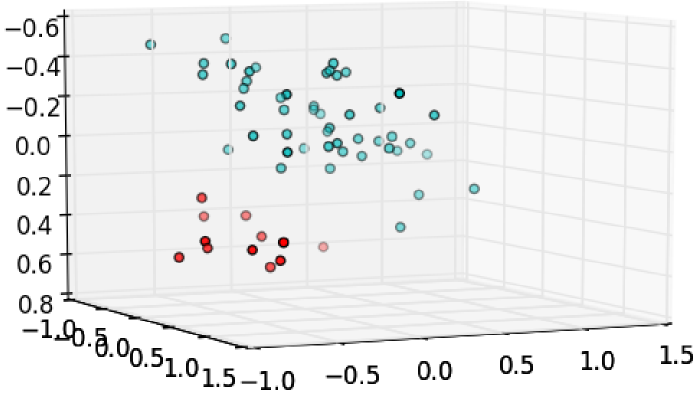


Figure 6.3: Example of an ACP study for Dakar Friday anomalies tesbed

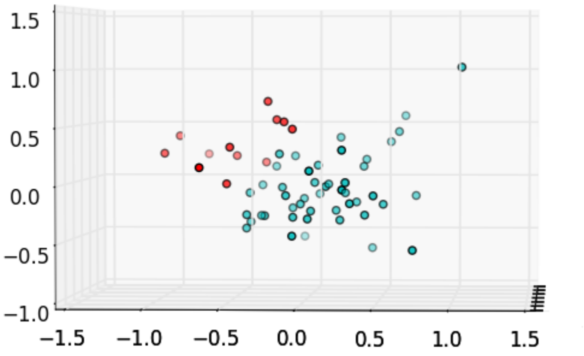


Figure 6.4: Example of an ACP study for Dakar Friday anomalies tesbed

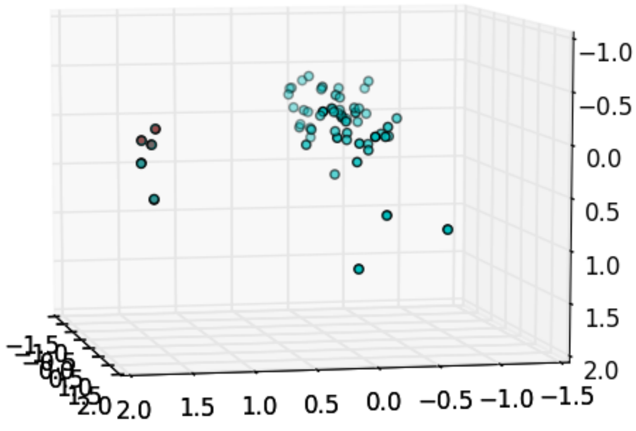


Figure 6.5: Example of an ACP study for Dakar 5th February anomalies testbed

6.4.2 Geographic anomaly detection results & performance evaluation

Milan dataset

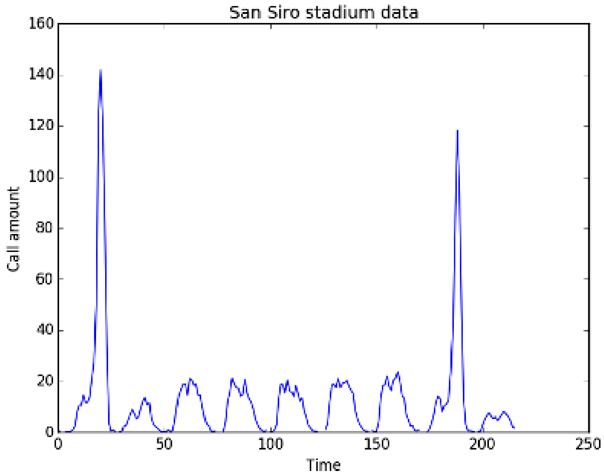


Figure 6.6: San Siro stadium square Call amount evolution from Sunday November 3th to Sunday November 10th

SanSiro stadium study case: While analyzing Milan dataset, using Principal Component Analysis (PCA) on network activity time-series, we discovered some abnormal behaviors within the geographical square where San Siro stadium is located. In fact, these anomalies consist on a huge increase of users network activity with a time-limited peak of SMS, call or even Internet packets. Figure 6.6 shows an example of Call activity evolution during one week in San Siro square: the first anomaly occurred on Sunday November 3th while the second occurred on Saturday 9th. These strange behaviors occur especially on the weekend and it last for almost 2 hours. We verify the time interval in which these heavy networks activities take place with San Siro sports events calendar and we realize that it is strongly correlated with football matches time. To detect automatically these anomalous patterns, we train our model first-stage with the current square workdays normal data, hence, OCSVM can find the appropriate normal data boundary.

To better evaluate the performance of our model, we used Isolation Forest (iForest) algorithm to compare both algorithms performances.

Table 6.1: Performance comparison between OCSVM and Isolation Forest for Milan dataset

Performance	**San Siro**		**Christmas eve**		**Christmas & Saint Silvester days**		**Saint Stephen's day**		**Global**	
Classifiers	OCSVM	iForest	OCSVM	iForest	OCSVM	iForest	OCSVM	iForest	OCSVM	iForest
Accuracy	100%	91.6%	98.2%	96.1%	100%	87%	100	97.9%	99.7%	92.6%
Sensitivity	100%	90%	97.8%	97.6%	100%	97.6%	100	97.6%	99.6%	93.8%
Specificity	100%	100%	100%	87.5%	100%	50%	100	100%	100%	86.5%
Precision	100%	100	100%	97.6%	100%	87.2%	100	100%	100%	97.3%
FOR	0%	33.3%	11.1%	12.5%	0%	14.2%	0	14.2%	1.8%	23.8%

Isolation Forest [136] is an enhanced variant of random forest machine learning which consists on generating a tree-classifiers ensemble based on training data and try to segment data until isolating anomalies. Both method are evaluated based on performance metrics extracted from the corresponding confusion matrix and numerical results are resumed in table 6.1. Results show that One-class SVM is able to detect correctly all tested anomalies with an accuracy of 100% unlike Isolation Forest which signals some false positive samples.

End of year festivities study case: PCA analysis reveals also some non-common patterns of users consumption which has occurred during the last week of the year. Figure 6.2 depicts a representation of a portion of the data (Christmas eve data) after applying a PCA decomposition. The red points represent the anomalous time-series which their projections are distant from the normal time-series cluster (Blue points). Some of The anomalous time-series are related to Christmas holidays, i.e. anomalies occurred during Christmas eve on Tuesday 24th, Christmas day on Wednesday 25th and also Saint Stephen's day on Thursday 26th. During these dates, a non-common decrease of users' network activity is noticed. Another types of anomalies is related to new year eve, with a brief peak of network activities during the first hour of 2014, and also Wednesday January 1st holiday with a decrease of network activities throughout the day. Figure 6.7 shows the call amount during November (top), December and January 1st 2014 (bottom) of the square where is located the touristic site of Duomo. We can notice clearly from these two figures the anomalous call amount patterns occurred on the mentioned days (red curves).

Unlike the first case study, we can not model here the OCSVM model with all workdays time-series as normal training data because each day has its own behavior (or profile class) [111]. Hence, w.r.t

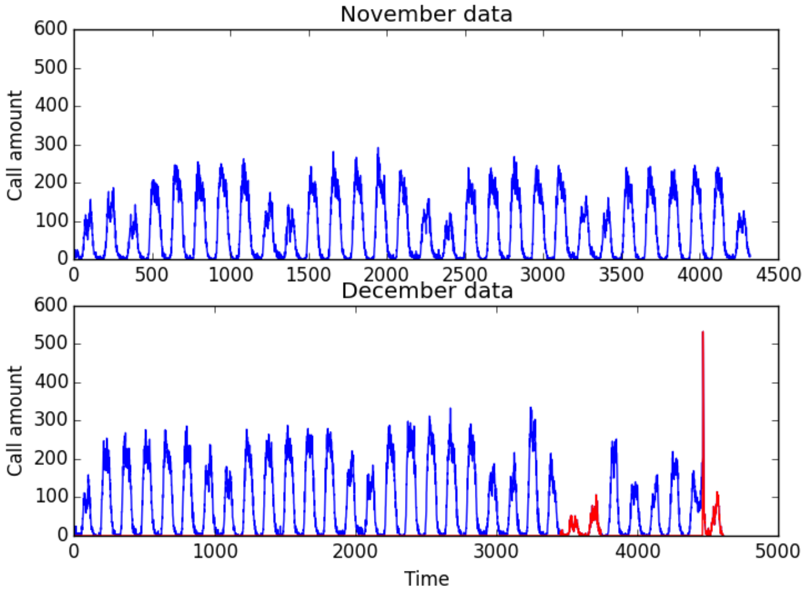


Figure 6.7: Duomo square Call amount evolution

the weekly cyclic characteristics, a given anomalous day is detected by training the model with its past normal data in a weekly basis. For example, we train the OCSVM model with all previous Tuesdays data from November 5th to December 17th in order to detect the Christmas eve anomaly. Although, the issue with this formulation is that the number of training instances are few (almost 7 days data for training) which may cause under-fitting problems. To handle it, we consider data from squares directly adjacent to the target square. We also verified that data from adjacent squares are very similar whether in normal days or outliers. This allows us to built the OCSVM model with more relevant training data which guarantees its efficiency. We compare also the performance of our model with iForest model using the same training dataset and results are resumed in table 6.1. It shows that SVM performances are better than Isolation forest which reaches a global accuracy of 99,7% against 92.6% for iForest.

Dakar Dataset study case

For Dakar use-case, the study is more refined since the data represents users' activity inside a cell instead of a larger square as Milan dataset and it is more accurate to analyze the network from a fine-grained perspective. Prior data analysis (also with PCA) allow us to capture some outliers behavior within Dakar dataset but due to space limitations we present some examples of these anomalies (Figure 6.3-6.5). Figure 6.3 and 6.4 depict an example of PCA decomposition of two different BS time-series data during the period of study (2 months). In the first example the separation between normal and anomalous data is clear while in the second example the separation is not so obvious and margin between the normal and anomalous clusters is low. Hence, a simple PCA is not very efficient to cluster the data and we will show in the results that the OCSVM provides high accuracy even with this kind of data.

Figure 6.8 outlines some examples of Dakar outliers patterns. A local decrease of users' number into the cell is noticed for some BSs which lasts for one hour usually between 12am and 1pm (top red curve of figure 6.8). These untypical behaviors, compared to other days (top blue curves of figure 6.8), occur usually on Fridays. Thus, we used workdays data except Fridays to train our first stage part which allow us to detect automatically these kind of outliers. Table 6.2 resumes the numerical results to compare our model performance against iForest. OCSVM model attempts an accuracy of 94.2% against 75% for iForest which confirm the efficiency of our method.

Another example of anomaly (Bottom graph of figure 6.8) is detected by our algorithm. We notice on Tuesday February 5th that the BS activity decrease heavily between 10pm and 11pm, and this abnormal observations is noticed for all BSs installed in Dakar, otherwise, this sudden decrease is not present neither in days of the same week or in other Tuesdays. This kind of anomalies may occur due to technical incidents in the network equipment or electricity issues. This anomaly is detected by training our model with the history of Tuesdays data anterior to February 5th. The results are compared to iForest and are presented on Table 6.2. Results shows that OCSVM is able to detect all anomalous BSs patterns with a full specificity and an accuracy of 95% while Random forest fails in detecting some of them and signals more false negative samples with an accuracy of 65%. Furthermore, for Dakar use-case the global accuracy of OCSVM reaches 93.5% against 70% for iForest.

We notice, for Dakar dataset, that iForest results decrease drastically compared to those of Milan dataset. This may be due to the data aggregation scale, whose is very small in the case of Dakar which may cause some noise in time-series. On the other hand, OCSVM results remain also accurate for Dakar dataset which confirm its ability of generalization and then avoid false alarms signaling.

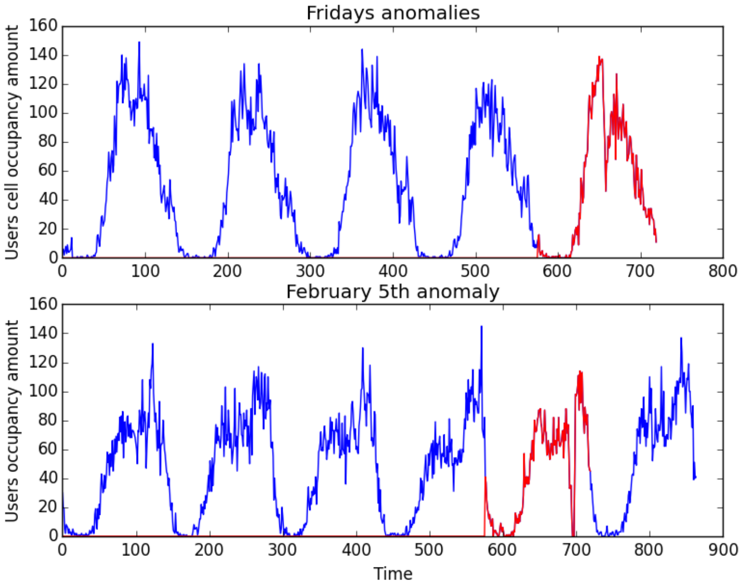


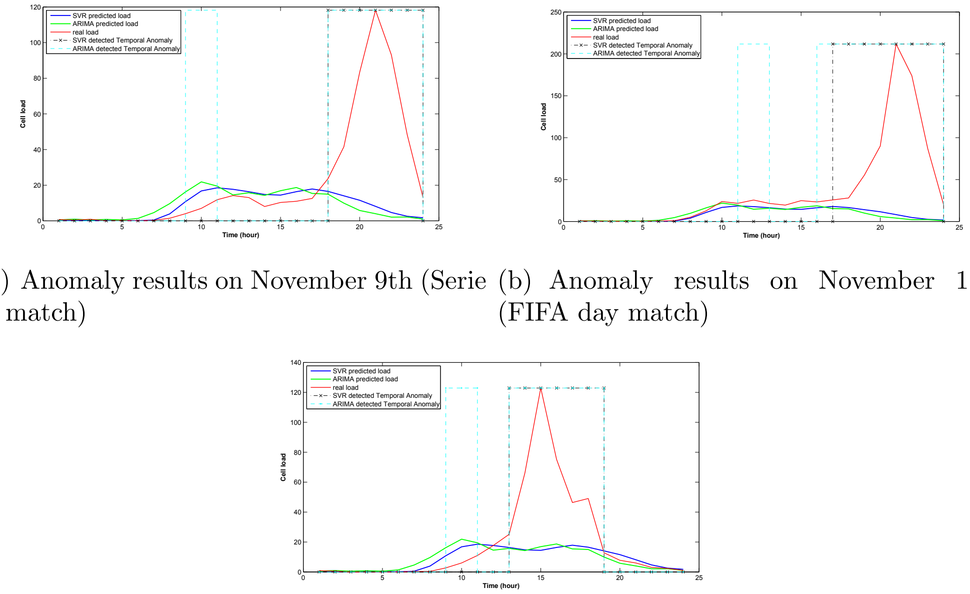
Figure 6.8: Examples of Dakar anomalies use-case: the top figure depicts an example of Friday anomaly (red curve) and its previous workdays normal data (blue curves). The bottom figure shows an BS examples of February 5th anomaly (red curve) and some other Tuesdays normal data

Table 6.2: Performance comparison between SVM and Isolation Forest for Dakar Dataset

Performance	**Fridays**		**February 5th**		**Global**	
Classifiers	OCSVM	iForest	OCSVM	iForest	OCSVM	iForest
Accuracy	92%	75%	95	65%	93.5%	70%
Sensitivity	94.8%	77%	94.6	63.3%	94.7%	70.1%
Specificity	81.5%	66.6%	100	80%	90.8%	73.3%
Precision	94.8%	90.2%	100	97.2%	97.4%	93.7%
FOR	18%	57.8%	37%	83%	27.5%	70.4%

6.4.3 Temporal anomaly detection results

In this part we evaluate the prediction based method for temporal anomaly detection based on SVR. In the study [82], authors investi-



(a) Anomaly results on November 9th (Serie A match)

(b) Anomaly results on November 15th (FIFA day match)

(c) Anomaly results on December 1st (Serie A match)

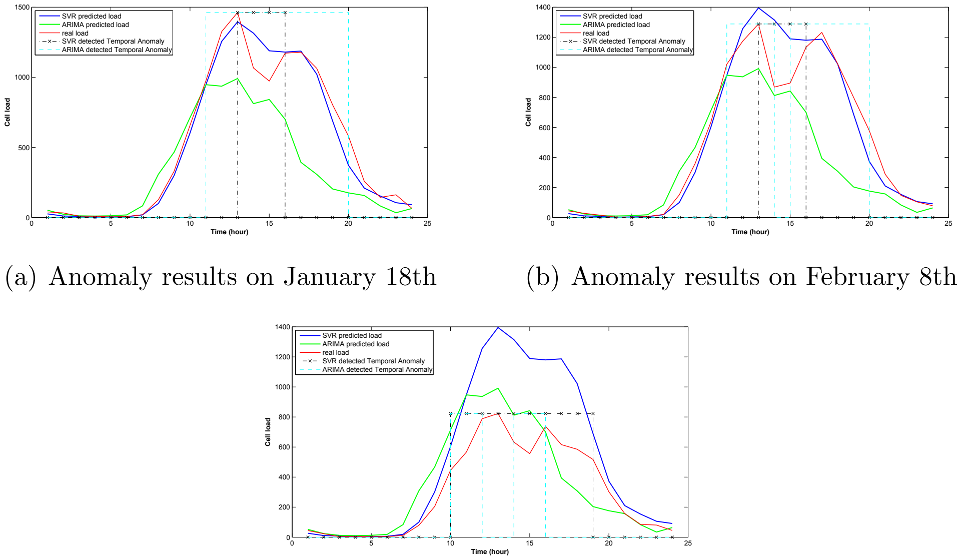
Figure 6.9: Temporal Anomaly detection result for SanSiro testbed

gate the efficiency of mobile network traffic prediction by comparing SVR to artificial neural networks (ANN). Results showed that SVR outperforms ANN for multidimensional data (which is our time-series case study). Furthermore, we present also in this part a comparison between our model and an ARIMA based detection anomaly model [138].

The second phase of our solution consists in enhancing the precision of outliers detection by pointing out the duration and the time-interval when the anomaly occurred. This phase consists on a prediction-based model, hence, we choose to compare our solution to ARIMA which take part of this category.

For each use-case studied previously, we use the same OCSVM training data to train also the SVR algorithm. Hence, the so-called normal behavior is predicted which is then compared to the current anomaly pattern in order to outline the temporal anomaly. The features model we adopted for the prediction is the same we used in a previous work [111].

We present here the results and comparisons of temporal anomaly detection between the two models and for both datasets, Milan and



(a) Anomaly results on January 18th (b) Anomaly results on February 8th

(c) Anomaly results on March 1st

Figure 6.10: Temporal Fridays Anomaly detection results for Dakar testbed

Dakar cellular networks. Figures 6.9 show some examples of San Siro network activity anomalous patterns. We notice from these figures that the anomalous time-interval detected (highlighted in green) by the SVR-based model is synchronized with the untypical peak of network activity while the ARIMA-based model is less accurate. In these figures, we notice that the ARIMA-base model signals a false anomaly around 10pm, contrary to the SVR-model, due to its overrated time-series prediction at this time-stamp. In addition, in figure 6.9(b) we see that the ARIMA model starts signaling the anomalous pattern prior its exact occurring time, while the SVR-based detection is well synchronized.

Figures 6.10 depicts some examples related to Dakar Fridays anomaly, and we can clearly notice that the detected anomalous time-interval is also synchronized with the time in which occur the abnormal decrease of BS users' occupancy and for the same duration. We notice also in figure 6.10(c) that the detected temporal anomaly lasts for more time compared to others (usually for one hour almost) and it covers almost the day activity time and this may be due to a special event in that area which cause a sudden decrease of users throughout the whole day.

In the other hand, the ARIMA-based model fails in detection the anomalous time window for all examples. Instead of detecting the drastic decrease of cell load occurring between 1pm and 4pm, it detects two false anomalies: the first one around 11am and the second one around 5pm.

6.5 Anomaly detection for eHealth-enabled Data

We show in this section a use case for anomaly detection applied on new applications such as IoT.

6.5.1 Context and motivations

We present in this section a direct application of our STAD framework. The objective of this part is to provide analysis study of the potential impact of eHealth and IoT application data on the future networks generation (such as cognitive networks, 5G networks, etc) and to evaluate the performance of our framework against a new source of data injected in the network. STAD framework is hence applied on a semi-synthetic dataset, combining the real load time-series presented earlier in this chapter and a simulated eHealth data generated from IoT applications.

In fact, Internet of Things (IoT) is an ever-growing technological paradigm that is expected to boost the development of a plethora of services and applications like eHealth services. The growth of such technologies will with no doubt impact the cellular networks. The massive amount of data generated by eHealth applications will be handled by the cellular architecture. Due to the additional eHealth data, cellular networks may suffer from some anomalies which need intelligent and autonomic mechanisms to be avoided. Network operators must integrate to their architecture pro-active tools able to detect and signal these anomalous patterns and then mitigate the issue of overloaded base-stations.

Moreover, this exponential growth of IoT application may generate new users' demand dynamics, and then create heavy spatio-temporal data consumption patterns in the network[111]. These data patterns can be difficult to be managed by network operators. On the other hand, a major issue for network operators is to handle and detect sudden and local anomalous behavior within the network, whether it is a sharp peak of users demands (occurred during mass events like a marathon), which needs intelligent tools to guarantee network resilience and service survivability avoiding users rejections. We present in this context our STAD framework as a solution able to provide network operators with an on-line and pro-active solution to face these anomalies. Our framework is applied to a pre-analyzed semi-synthetic data set of eHealth-enabled cellular data in the context of a mass event (Marathon event) in Milan city.

6.5.2 eHealth semi-synthetic data-set

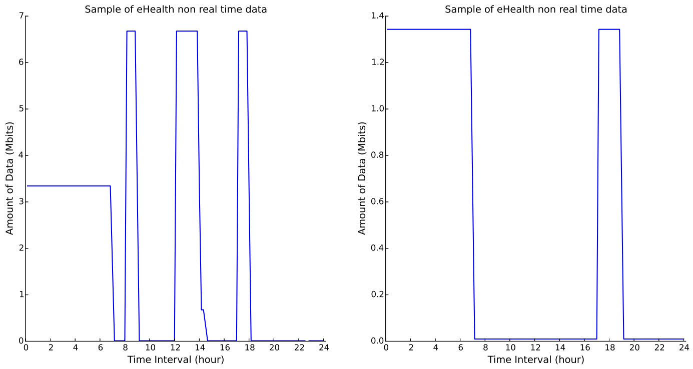
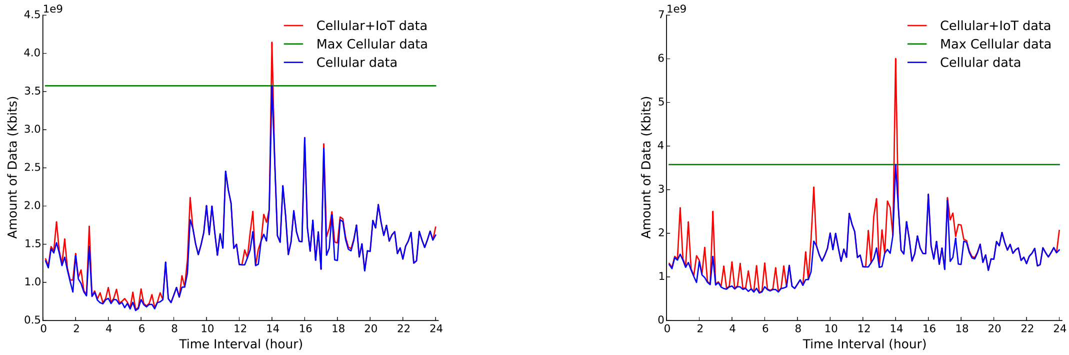


Figure 6.11: Sample of a daily eHealth-enabled application data

The concept of Information Communication Technology (ICT) passage in health care delivery popularly known as e-Health has attracted international initiatives with huge budgets, as an action for guaranteeing a global and equitable care delivery. The International Telecommunication Union (ITU) argue that successful wellness and eHealth deployment depends on three major links between patient and care provider telecommunication infrastructure, back-end computing/storage infrastructure and User-end platforms [139].

From gathered ehealth and wellness data application point of view, the nature of information exchange events (IEE) within the transactions was studied and graded in terms of its purpose(for example discharge summaries, prescriptions, conferencing streams, Radiology reports, etc.), the representation (video/image/text/etc.), the data size, the urgency of delivery (real time, offline) and mobility.



(a) Scenario: IoT users = 20% and NRT application = 50%

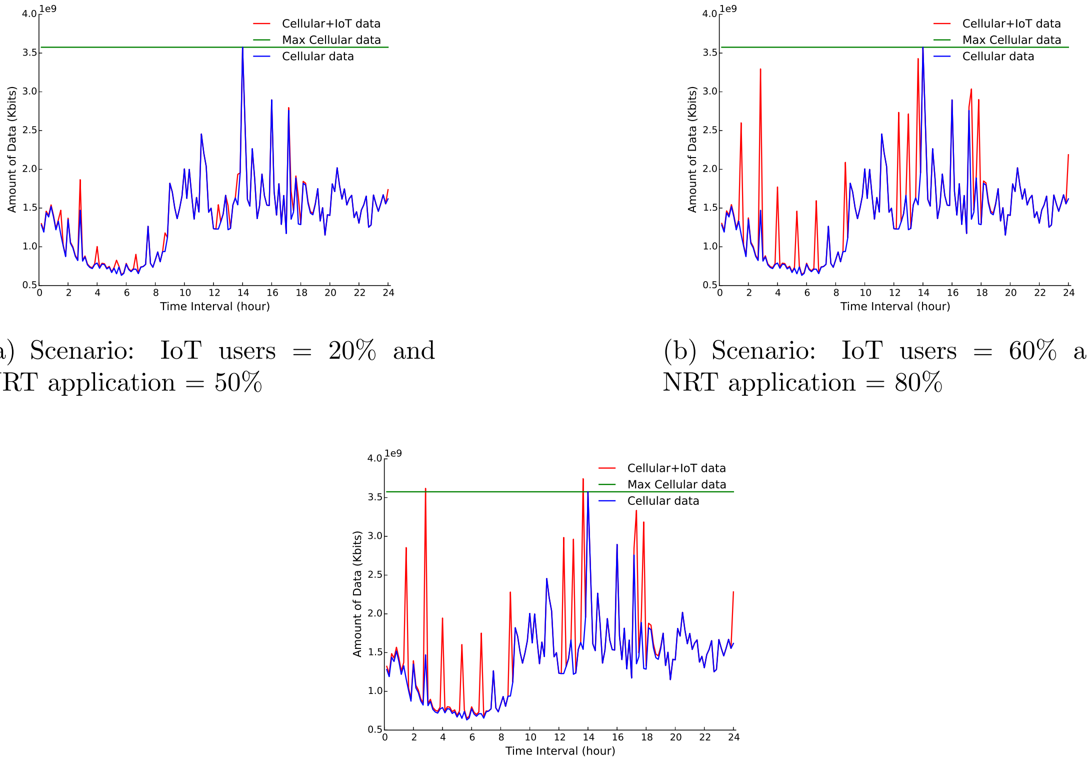
(b) Scenario: IoT users = 80% and NRT application = 80%

Figure 6.12: Simulation results for storage capacity = 10%

The ITU [139] paper exhibits that the distribution of information exchanges that require the two types of information transfer; store-and-forward real time data transfer (including buffered transmission) methods. It can be spotted that over 85% of the transactions can involve store and forward methods. Prescription, surveillance diagnostics and special reporting transactions and more then 80% of program management can be handled with store forward mode. On the other hand, disaster, ambulance services care and some wellness transactions are expected to happen in real-time mode. One can judiciously choose the mode of information exchange to save on the complexity and cost of infrastructure required for real time exchange. The patents wellness/medical records aforementioned in figure 6.11 are composed from a day records of real time and store-and-forward data applications. The gathered data concern ECG, EMG, Blood Saturation, Glucose Monitor, Temperature, Motion Sensor, Audio Medical Imaging Video, Voice, Capsule Endoscope, Artificial Retina and Cochlear Implant Internet of Things medical/wellness data.

6.5.3 Impact evaluation of eHealth data on cellular network: Testbed and Interpretations

We present in this part a testbed of the eHealth-enabled cellular data and we evaluate its impact on the network. The testbed is used later to validate our anomaly detection model. We investigate espe-



(a) Scenario: IoT users = 20% and NRT application = 50%

(b) Scenario: IoT users = 60% and NRT application = 80%

(c) Scenario: IoT users = 80% and NRT application = 70%

Figure 6.13: Simulation results for storage capacity = 25%

cially the effect of IoT data capacity storage, eHealth users percentage and non-real time (NRT), as well as real time (RT) eHealth application (for each eHealth user) percentage parameters. The storage capacity expresses the amount of NRT data to be stored in the smart-phone before forwarding it to the network, so it determines when these IoT data should be sent. It is chosen to be beyond the maximal smartphone storage capacity so that it does not affect its performance. Results are presented by Figures 6.12-6.14.

We notice from these figures that the IoT data traffic will add an important load to the normal cellular network and it drastically impact the network in some configurations.

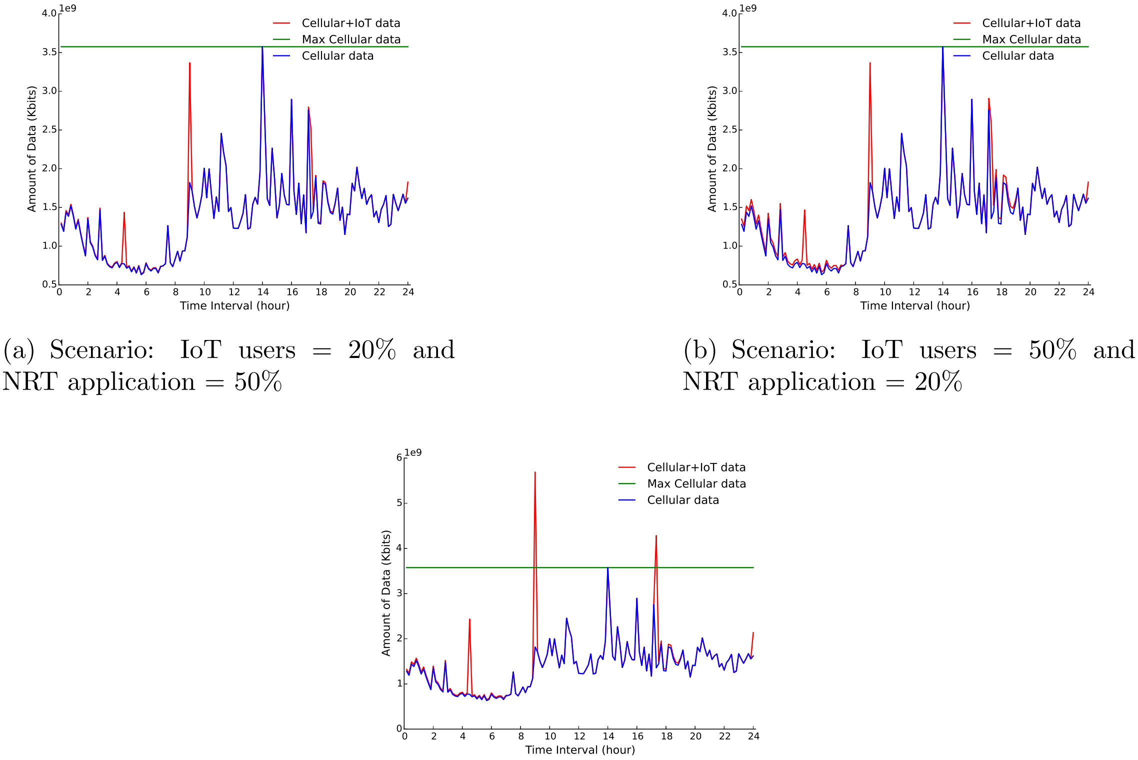
In figure 6.12, we are fixing the storage capacity for eHealth data to 10% out of the total capacity storage of the smartphone. We notice from the figure 6.12(a) that even for a low IoT users percentage (20%), the eHealth generated data cause an overrated peak of data (around 2 pm). This load peak impacts the radio channel occupancy causing then an anomaly. In figure 6.12(b), the eHealth users percentage is increased to 80% of the total users within the cell. We notice that we still have just one load peak (the same one as the first scenario, but with higher amplitude). However, the aggregated data load (eHealth and cellular data) increases by 40% (especially before the load peak hour). Otherwise, this extra load still less than the maximal cellular data consumption.

The network load peaks caused by eHealth applications can be avoided by increasing the capacity storage. This is shown in figures 6.13(a) and 6.14(a) where the storage capacity is increased to 25% and 60% respectively while keeping the same configuration as in 6.12(a). So, we notice that the eHealth data has no drastic impact on the cellular network and we also avoid to overload the network at peak hours. This due to the fact that the forward time for the non-real time application (the most data consumer in our case-study) does not coincide with the usual network peak hour. Figure 6.13(b) shows that in case of eHealth users and NRT percentage increasing, the global data load still beyond the maximal usual cellular data (without network anomalies), but the overall data consumption is importantly increased causing many data peak compared to the usual cellular data.

However, these load peaks has no big effect on the network since they cause no anomaly. In this storage capacity configuration, the eHealth data are impacting the cellular network from 80% of IoT users and 80% non-real time application (Fig 6.13(c)).

In figure 6.14, the storage capacity is fixed to 50% and we can notice that the load peak caused by the eHealth data (either beyond or above the usual maximal cellular data amount) are lesser. We notice from figure 6.14(b) that the eHealth users percentage can reach 50% without affecting drastically the network or causing a network anomaly (data peak above the maximal cellular data). But this is still true if the percentage of non-real time application still less than 20%. In figure 6.14(c), the storage capacity and the eHealth users percentage still unchanged but the NRT percentage is fixed to 50%. We notice that this configuration causes two network anomalies (around 9am and 5pm). In the first anomaly, the global amount of network data is doubled compared to the usual cellular amount. This fact could have a huge negative impact in the performance of cell and the network operator must find efficient solution to manage it.

We conclude from all these scenario that the choice parameters are inter-dependent. In some cases, the parameters are proportional as for the scenario in figures 6.13(a) and 6.13(b): The network operator can enable more eHealth users within the cell if their storage capacity is increased so that the forwarding of eHealth data occurs out of peak hours. In other cases, the parameters are inversely proportional like in scenarios depicted by figures 6.14(b) and 6.14(c). In these scenario, the network operator can enable more eHealth users but with lesser percentage of the non-real time eHealth applications.



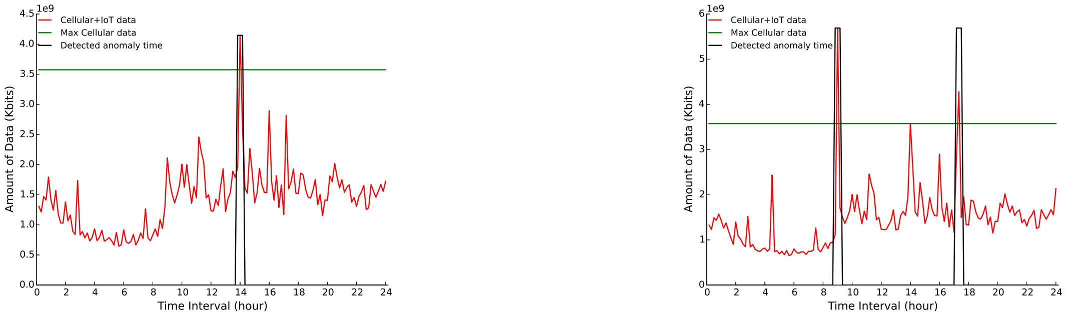
(a) Scenario: IoT users = 20% and NRT application = 50%

(b) Scenario: IoT users = 50% and NRT application = 20%

(c) Scenario: IoT users = 50% and NRT application = 50%

Figure 6.14: Simulation results for storage capacity = 45%

6.5.4 Anomaly detection results and network management recommendations



(a) Scenario 1

(b) Scenario 2

Figure 6.15: Anomaly detection results for eHealth enabled Cellular data

In this part we present results of our network anomaly detection applied to the semi-synthetic eHealth-enabled data that we analyzed earlier. This algorithm is based on SVR normal cellular data prediction. Based on the results of [111, 82], we choose to use SVR to integrate it into our model.

We use the history of the cell data to train the SVR algorithm at first. The features model we adopted for the SVR-based prediction is presented in a previous work [111]. After adjusting the optimal set of parameter for the SVR prediction model (discussed in section V.A.), the so-called normal cellular behavior is predicted. Then, the predicted data is compared to the instantaneous collected data (eHealth enabled cellular data) in order signal the network anomaly if it occurs.

We present here some results of network anomaly detection algorithm for eHealth semi-synthetic dataset. Figures 6.15(a) and 6.15(b) show some examples of anomalous patterns from the previous presented scenario. We notice from these figures that the anomalous time-interval detected (highlighted in black) is synchronized with the abnormal peak of network activity. We also notice in figure 6.15(b) that our model detects and distinguishes with precision two sparse anomalous load peak and it doesn't fail to signal the whole period between these anomalies as anomalous time interval even it contains a usual cellular load peak. Our model is able to detect other types of anomalies, but due to lack of space we omit extra results.

The efficiency of our model is proved by these results. Hence, it constitutes a accurate and autonomic tool for network operator to monitor and manage on-line their network even with eHealth-enabled data and then avoid pro-actively the network QoS degradation caused by the extra IoT data. Also, the network operator could recommend to use WiFi connection if it is available in order to forward eHealth data. So that allows to alleviate the cellular network resources. Note that WiFi will be enabled in next 5G networks.

6.6 Conclusion

Network anomalies are abnormal behaviors that occur suddenly affecting the performance of mobile networks and causing expensive lost for the operator. In this context, we propose a new efficient framework which provides an automatic and on-line detection of network spatio-temporal anomalies to prevent such aforementioned issues. This tool allows network operators to monitor their infrastructure dynamically and permits them to manage new challenges in next-generation networks. The framework combines two machine learning techniques: OCSVM for spatial detection and SVR for temporal anomalies detection. The framework is validated with real datasets of CDRs from two different urban cities and its high accuracy is proved. A comparison is also made with the *iForest* algorithm and results show that our model performance is better. The SVR part allows to predict accurately the normal behavior of the network in order then to detect the timestamps of the network outliers with high precision.

Predicting network outliers may be difficult since they are very diverse, but detecting them may help on predicting their propagation nearby. Ongoing work is carried on to study it.

We also investigate in this section the impact of the potential growth of data generated by eHealth/wellness applications on the cellular networks and we show that, in some cases, it could drastically degrade the network performances.

Machine Learning has been identified as the key tool to implement autonomous adaptability and take advantage from experience when making decisions.

Results show that our model allows to predict and detect anomalies with high precision. The proposed tool helps network operators to efficiently manage efficiently their infrastructure. It also allows them to implement self-organized and autonomous networks that can face the plethora of eHealth and IoT data.

Chapter 7

Drone-assisted cellular network optimization: A Multi-Agent Reinforcement Learning Approach

7.1 Introduction

The growth of data demands and the increase in wireless traffic rates in new mobile networks need intelligent and dynamic technologies for telecommunication management. Recent studies predict that the new generation cellular standards (like 5G) will rely much more heavily on a dense and less power consuming network to serve the dynamic users' data rate requests [3].

Small cells mounted on unmanned aerial vehicles (UAVs) or drones (we call them drone-cells hereafter) are proposed, as an alternative to fixed femto-cells, to support existing macro-cell infrastructure. The deployment of these mobile small cells consists on move these small cells toward a target position (usually within the range of the macro-cell to support) based on the decision made by a mobile network operator. The drone-cells movement toward adequate positions must be correlated with the amount of data requested, i.e. drone-cells should support overloaded cells (Figure 7.1). Hence, an intelligent entity may be added to the network that monitor instantly the network state and find the optimal decision to control drone-cells.

In the other hand, we presented, in the previous chapter, a framework able to detect pro-actively anomalous pattern of bandwidth consumption, either heavy load scenario like in a mass event or a drastic decrease of load that can be due to a network failure. Since the abnormal decrease needs some investigations to identify the network failure, the unexpected peak of load needs a fast deployment of extra resources to cover the users demand. In this chapter, we propose a dynamic network access topology based on drone-cells that assists the existent architecture and helps network operator to face unexpected heavy bandwidth demand in mass event scenarios. Due to the fast deployment and dynamic behavior these small cells can be deployed in an area of high users' traffic, a group of UAVs is instructed to occupy positions above high concentrations of User equipement (UE) and serve the UEs below.

Several studies have discussed the theoretical ability of cellular networks to use drone-cells to support their existent macro-cells [66, 64] Although, the readiness of networks to integrate such dynamic entities has not been discussed. For example, drone-cells require coordinated insertion to the network infrastructure while serving subscribers and smooth recess at the end of their activity (low battery, low data demand ...). This requires an efficient network configuration. It also adds management flexibility and self-organizing capabilities for the networks. Hence, updating the network, such as for integrating new tools, and technologies, becomes essential. Also, massive amounts of information about users and networks must be continuously collected and analyzed by intelligent algorithms.

In this work, we propose a dynamic solution for drone-cells networks that exploits real traces of demand profiles and adapts in real time the deployment of drones-cell according these demands. We propose to optimize the deployment using the machine learning programming instead of classical linear programming models. Our solution is based on a multi-agent reinforcement learning (MARL) approach which consists on two steps: an off-line step corresponding to exploration step and an on-line exploitation step. The off-line step consists especially on a learning phase where the drone-cells of the multi-agent topology learn, in a cooperative re-reinforcement learning context, how to react and move according to different scenario of bandwidth demands. Real Traces of network time-series load are used for this phase. In the on-line phase, the pre-learning agents demonstrate how they are adapted instantly on a real scenario of load peak signaled by the STAD framework (Framework of the chapter 6). The cooperative reinforcement learning is based on a joint action selection that aim to choose the optimal set of agents action. We propose on in this chapter our joint action selection algorithm and we compare it against the grid selection algorithm.

The chapter is organized as follows. Section 2 presents reinforcement learning approaches. Section 3 develops our MARL-based model. Section 3 presents simulations uses-cases and results and we conclude in section 4.

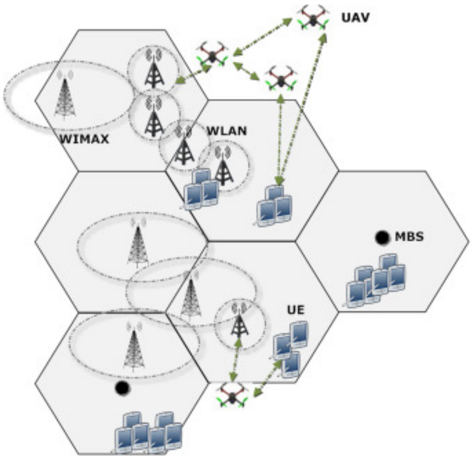


Figure 7.1: Graphical illustration of a Drone-assisted network

7.2 Reinforcement learning concept

In this section, we introduce the background on both “single-agent” and “multi-agent” reinforcement learning approaches. First, the single-agent approach is presented as well as its solution. Second, the multi-agent approach is defined. We also discuss the choice of using multi-agent approach in our contribution by presenting its benefits and advantages against the single-agent one.

7.2.1 Single reinforcement learning approach

Reinforcement Learning is a computational model for reward based learning implemented using the Markov Decision Process (MDP) framework. In the single agent reinforcement learning approach, an agent interacts with its environment, at each iteration, by choosing actions to execute. By executing the action, the agent moves to a new state and perceive a reward signal. The reward value may be positive indicating whether he reaches some goal. Negative reward value represent a penalization for the agent.

The single reinforcement learning approach is modeled by a finite Markov decision process which is represented by a tuple $< X, A, f, r >$, where X is the set of environment states, A is the finite set of agent actions, $f : X \times A \times X \rightarrow [0, 1]$ is the state transition probability function, and $r : X \times A \times X \rightarrow R$ represents the reward function. The environment is described, at each discrete time step k, by the state $x_k \in X$. The agent examines the state and selects an action $a_k \in A$. After executing the action $a_k \in A$ the state environment changes to $x_{k+1} \in X$ according to the transition probability given by $f(x_k, a_k, x_{k+1})$. Hence, the agent perceives a reward $r_{k+1} \in \mathbb{R}$ calculated by the reward function as follows: $r_{k+1} = r(x_k, a_k, x_{k+1})$. The reward value evaluates the effect of taking the action a_k when moving from the state x_k to the next state x_{k+1}.

The agent acts according to its policy which defines the behavior of the agent when choosing its actions in a given the state. The agents objective is to find a policy that maximizes, from every state x, the expected discounted return:

$$R^h(x) = E\left\{\sum_{i=1}^{\infty}(\gamma^k r_{k+1}|x_0 = x, h)\right\} \tag{7.1}$$

where $0 \geq \gamma < 1$ is the discount factor such that the expectation is taken over the probabilistic state transitions under the policy h. R is the return and it represents the reward accumulated by the agent during many iterations. The discount factor γ can be considered as encoding an increasing uncertainty about the future perceived rewards, or as a means to bound the sum which could increase unbounded.

The agent aims then to maximize its long-term return, while only receiving feedback (reward) about its immediate, one-step performance. This can be achieved by computing the optimal state-action value function, the $Q - function$ which gives the expected return obtained according to the policy h from any state-action pair:

$$Q^h(x,a) = E\left\{\sum_{i=1}^{\infty}(\gamma^k x_{k+1}|x_0 = x, a_k = a, h)\right\} \tag{7.2}$$

The optimal Q-function is defined as $Q^*(x,u) = \max_h Q^h(x,u)$ and it is evaluated so that it satisfies the Bellman optimality equation:

$$Q^*(x,a) = \sum_{x' \in X} f(x,a,x')[r(x,a,x') + \gamma \max_{a'} Q^*(x',u')] \tag{7.3}$$

This equation express that the optimal value of executing a in x is the expected immediate reward plus the expected (discounted) optimal value attainable from the next state (the expectation is explicitly written as a sum since X is finite).

Several studies addressed single-agent RL algorithms such as proposing model-free methods [140, 141]. Other models are based on dynamic programming [142, 143]. A well-known model derived from a model-free algorithms called Q-learning [144, 145, 146] which will be used also in this study.

Q-learning transforms the previous equation to an iterative approximation procedure. Q-learning updates after each transition the Q-function as follows:

$$Q_{k+1}(x_k,a_k) = Q_k(x_k,a_k) + \alpha_k[r_{k+1} + \gamma \max_{a'} Q_k(x_{k+1},a')Q_k(x_k,a_k)] \tag{7.4}$$

The second term is the temporal difference, i.e. the difference between the current state estimation $Q_k(x_k,a_k)$ of the optimal Q-value and the next state estimation: $r_{k+1} + \gamma \max_{a'} Q_k(x_{k+1},a')$.

7.2.2 Multi-agent reinforcement learning

As the wireless topology is composed of multiple drone-cells, and since modeling a single reinforcement learning model for each single drone-cell is complex and leads to an exponential explosion of the $(action, state)$ space, we propose to employ a multi-agent reinforcement learning model for the drone-cells network optimization and management.

The multi-agent reinforcement learning approach is a generalization of the single model and its modeled by a stochastic game. The multi-agent RL employs a joint action, which is the combination of actions to execute by each agent at state k.

The stochastic game is represented by the tuple:

$$< X, A_1, ..., A_n, f, r^1, ..., r^n >$$

with n is the total number of agents. A_i with $1 \geq i \leq n$ is the set of possible actions of agent i. So we can define the joint action as $A : A_1 \times A_2 \times ... \times A_n$. $r^i : X \times A \times X \rightarrow \mathbb{R}$ is the reward function of agent i that is assumed to be bounded. $f : X \times A \times X \rightarrow [0, 1]$ is the transition probability function. In the multi-agent approach, the transitions between states are the result of a joint action of all the agents which is expressed as: $ja_k = [a_{1,k}, ..., a_{n,k}]$ where $a_{i,k} \in A_n$. The reward $r_{i,k+1}$ $(r^i(x_k) = r_{i,k+1})$ resulting from executing the action $a_{i,k}$ is computed according to the joint policy jh and the joint expected return is expressed as follows:

$$R_i^{jh}(x) = E\left\{\sum_{i=1}^{\infty}(\gamma^k r_{k+1} | x_0 = x, jh)\right\} \tag{7.5}$$

The $Q - learning$ function depends also on the joint action and the joint policy and it is expressed as follows:

$$Q_{k+1}^{jh}(x_k, ja_k) = Q_k(x_k, ja_k) + \alpha_k[r_{k+1} + \gamma \max_{ja'} Q_k(x_{k+1}, ja')Q_k(x_k, ja_k)] \tag{7.6}$$

In our model, we are adopting the fully coordinated multi-agent RL, and in this case all reward functions are the same for all agents, $r^1 = ... = r^n = r$

7.2.3 Motivations of using MARL approach

Along with advantages due to the distributed feature of the multi-agent systems, as the acceleration becomes possible by parallelizing the computation, MARL approaches emerge with the benefit of sharing experiences between agents. Experience sharing helps agents to learn faster and reach better performances. Hence, the agents can communicate between each other to share experience and learning so that the better learned agents may speed up the learning phase of other agents. Moreover, multi-agent systems facilitate the insertion of new agents into the system which leads to came up with scalability issues affecting classic method such as linear programming.

Otherwise, in addition to challenges inherited from single-agent reinforcement learning such as the curse of dimensionality (due to the (action,state) space dimentionality) and the exploration-exploitation trade-off, other challenges arise in multi-agents reinforcement learning approaches. One of these challenges stands for the difficulty of specifying a learning goal, the non-stationarity of the learning problem, and the need for coordination. The need for coordination is motivated by the fact that the effect of any executed action by any agent on the environment depends also on other agents' actions. So, the agents action selection should be mutually consistent in order to achieve their goal. So, coordination is necessary in cooperative MARL models.

Due the advantages explained earlier and as our network model consists of employing dynamic cells-drones, cooperative and coordinated MARL model become essential to ensure an efficient and faster optimization. Coordinated MARL models will allow us to propose a general framework for deploying dynamic drone-cells topology due to its scalability feature.

7.3 Drone-cells network agent model

In this section we present our proposed model for drone-cells optimization and management and we explain the mapping between our model parameters and the theoretical model presented in the previous section.

7.3.1 Model framework description

Our drone-assisted network model is based on a centralized multi-agent reinforcement learning approach. The framework consists of a multi-agent system implementation based on MESA [147] package. It is composed of model entities that describes the system parameters and manage the interaction between the different agents. The framework is described in figure 7.2.

Since the approach is centralized, we need a coordinator agent that collects information from different agents and intervenes in the actions selection process. In our framework, this coordinator agent is represented by the network operator agent (which can be also a centralized entity in the network architecture).

The framework is composed also of a set other agents. We distinguish two type of agent: Active agent who executes the model actions and passive agent that participates in the system interaction but without executing any action. The active agents are the drone-cells that are represented by the "Drones" entity in our framework. The actions to be executed by a drone is whether to move to a cell location in order to serve it or to go back to the facility location (in case of battery running out or no cell left to serve). The "Drones" entity defines the battery level $BT_{d,t}$ of a drone d at time-stamp t, the bandwidth offer level $BW_{d,t}$, the status of drone d (whether is serving, idle or charging) and other parameters.

The passive agents are the cell (represented by "Cells" entity) and the facility (represented by "Facility" entity) agents. The cell agent entity defines the bandwidth demand rate at each time-stamp, the geographic location of the cell and the status of the cell (whether it is served or not). The facility agent constitutes the initial location of all drones and where drones can be charged. If a drone is not serving a cell, it should move to the facility location. We suppose that the facility location is situated in the middle of the service area in order to optimize the flight time toward cells.

7.3.2 Model states

The model states provide information about the network status at each time-stamp t such as drones parameters, facility instant capacity or cells demand rates. In our MARL model, each drone agent d has it own state S_t which describes the actual parameters at time t. These parameters are the drone battery level $BT_{d,t}$, the bandwidth offer capacity level $BW_{d,t}$.

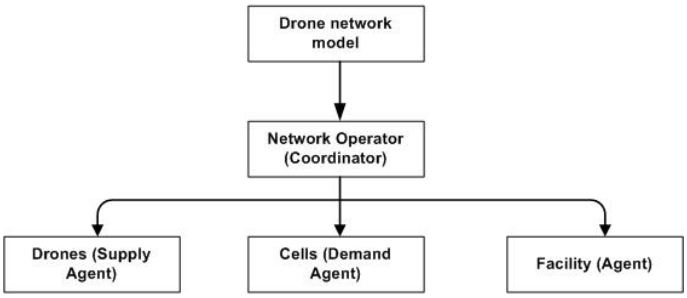


Figure 7.2: Graphical illustration of the MARL Drone-assisted network Framework and the relationship between the Agents

Moreover, each cell agent C state is represented by the aggregated throughput demand $Td_{C,t}$ at time t that exceeds the maximum capacity of the macro base station that covers it, and that should be assisted by drones. The facility agent state is also described by the current number of drones that are placed in it and either they are charged or not. All these information are gathered by the network operator agent (The coordinator agent) in order to choose the joint action to take for all drones in order to maximize the total network' QoS.

7.3.3 Model actions

The action that a drone agent can execute, at each state, is either to move and serve a cell and serve it or to move to the facility (to charge its battery or in case of no cell left to serve). Unlike single agent models, the action of each drone agent is taken by the coordinator (centralized approach) to maximize the global network QoS. Hence, it is not always straightforward that the chosen agent action maximize its own reward. Otherwise, the coordinator agent choose a joint action (set of action for all drones) that maximizes the global network performance.

7.3.4 Reward function

The reward function for each drone agent is measured at each state and after choosing the action to execute. The reward function measures the local network service ratio and it is the fraction of the served throughput by the drone d at time t and the throughput request of the cell to be served by d at the same time. Its expression is as follows:

$$r_{d,t} = \frac{Bandwidth\_Served_{d,t}}{Bandwidth\_Request_{C,t}}$$

7.3.5 Coordinated multi-agent RL

The multi-agent model we propose in this chapter is a fully coordinated model. The model needs a coordinator agent which plays the role of a network orchestrator. Hence, the coordinator agent ensures the communication between all drone-cells agents, collects the current status of each agent and the whole network (such as cells load). The coordinator also has access to the Q-table so that it ensures the selection of the joint optimal action at each iteration. We choose the coordinator to be a central network nodes such as a macro-cells that is able to communicate with all drone-cells agents and decides the actions of each agents at each state and updates the Q-tables entries.

Choosing the optimal joint action is a critical step for the multi-agent RL model. The criteria of choosing a better joint action selection algorithm is first, to reduce the computational cost of searching the joint action, which is exponential in the agents number and second, provide a better performance. We present in this section the hill climbing search algorithm (HCS) which is proposed in [148]. Then, we propose an enhancement of the Hill climbing search algorithm for an optimal joint action selection (eHCS) algorithm that speed up the actions search. The model is then compared to the HCS algorithm.

Hill climbing search algorithm

The main objective of hill climbing algorithm is to examine the neighboring agents one by one and to select the first neighboring agent action which optimizes the current reward as next node. The algorithm is resumed by the following steps:

1- Initialization step: Select an initial joint action JA by picking the action of each agent that maximizes the reward.

2- Randomly choosing an agent and its action in the previous JA is changed by a neighboring action.

3- The new formed action is evaluated. If it provides better network service ratio, it is stored as optimal JA.

4- Repeat steps 2 and 3 until testing all combinations.

Enhanced joint optimal action selection algorithm

Our multi-agent reinforcement learning model is based on a semi-centralized cooperative solution. The centralized part consists of adding a coordinator agent (Which is the network operator agent here) which is able to collect all drone agents and other model agents information in order to select the "joint optimal action" to be executed by drones. The centralized approach reduces the complexity of the system, speeds up the selection process and alleviates the system information sharing among all agent. Unlike the distributed approach which consists on forwarding all information between all agents, that may be a time consuming. Many centralized joint action selection algorithms exist in the literature such as hill climbing search [148], Stackelberg Q-Learning [149] etc. In our model we propose a modified version of hill climbing search algorithm which fits our drone actions selection problem.

The main objective of this algorithm is described as follows: at first, each drone agent, at a state S_t at time-stamp t, chooses its own optimal action to execute without taking into consideration other drones configuration. Then, the coordinator agent collects all drone information (Battery level, remaining bandwidth) as well as their chosen actions. It collects also cells and facility agents information. After that, the coordinator sorts decreasingly the set of cell demands (respectively the set of drone offers). Then, it tries, at each iteration, to find a match between the drone actions (the position to move toward) and the demand positions (cell positions) and assigns the adequate drone to the cell. Then, if the cell is totally served, the coordinator looks for drone agents whose action is identical to the served cell position and re-selects a new action for the drone according a new Q-table. The new Q-table is formed with the remained possible actions (except the selected prior).

The coordinator repeats the same process over all drones. This selection method avoids assigning many drones to the cells with higher demand and try to share drones all over the network.

The semi-centralized joint action selection algorithm is presents in algorithm 2.

Algorithm 2 Centralized Joint action selection

```
Input: (Action,State) space, Q-tables, ServedCell˙list
Output: Joint Action $JA_t$
$Collect\_system\_information$
for d in Drones do:
    if $Action_{d,t} \in ServedCell\_list$ then
        $ExtractNewQ - table(d)$
        $ActionSelection(NewQ\_table(d))$
    Verify NewAction(d,t)
    Find MatchAction($Cell$)
    if Matched Action then
        Actualize ServedCell_list
    Add NewAction(d) to JAction_List
return JAction_List
```

Note that the joint action selection algorithms are used only during the exploration phase (training phase). Once the Q-tables are calculated and their entries are the optimal values, they are used then for the on-line exploitation phase.

7.4 Simulation and results

7.4.1 Drone-cells network architecture

The architecture of the drone-cells network used here is composed of 2 kinds of infrastructure: A fixed infrastructure and a mobile infrastructure. The fixed infrastructure consists on six cells covering the areas represented by the figure 7.3. The area covers the Sansiro stadium and the car parks nearby the stadium. The infrastructure include the facility. The facility is placed in the center of the area. The facility is the initial location of the drones and is also used to store and charge the drones.

The mobile infrastructure is composed by the drone-cells. The drone-cells are either placed into the facility or fly over cells to serve the mobile users. The drone-cells can serve only one cell during a time-interval of one hour. The battery run out after serving during two time-intervals consecutively. In this case, the drone-cells should turn back to facility in order to charge its battery. The battery charge lasts for one time-interval, thus the discharged drone-cell stays non-operational during this time-interval.

7.4.2 Scenario use-case description

We validate our framework by using real data-set of cell demand time-series extracted from Milan CDRs dataset.

We simulate in this contribution the use-case of mass event where macro-cells are assisted by drone-cells managed by our framework in order to cover the unusual peaks of user demands. As shown and analyzed in the previous chapter, we detected some unusual behavior of demand around the SanSiro stadium during football match compared to workdays. Figure 7.3 represents the cells segmentation of the Sansiro stadium and areas around. Figure 7.4 depicts the daily demands time-series of these cells and we can notice that the SanSiro stadium cells have a high peak of user demand around 8pm (football match time) while nearby cells have two peaks: one before the match and another after it. This behavior may correspond to the arrival time of supporters and their departure time after the match. We can notice also that these cells are covering in a parking areas and some metro stations which make our assumptions logic.

We apply our framework to these demand time-series in order to manage the abnormal increase of users demand during this period of the day and we show results in the next part.

7.4.3 Results

The framework is applied on cell demands between 6pm and 11pm, the period of time when the abnormal demand peak are occurring. During this period, drone-cells will attempt to support the existent macro-cell and serve the additional demand. Table 7.1 resumes the simulation parameters.

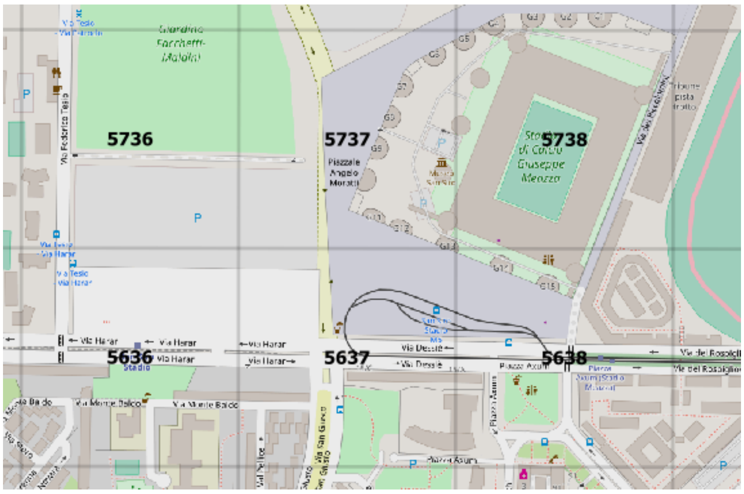


Figure 7.3: Graphical illustration of cell segmentation of SanSiro area

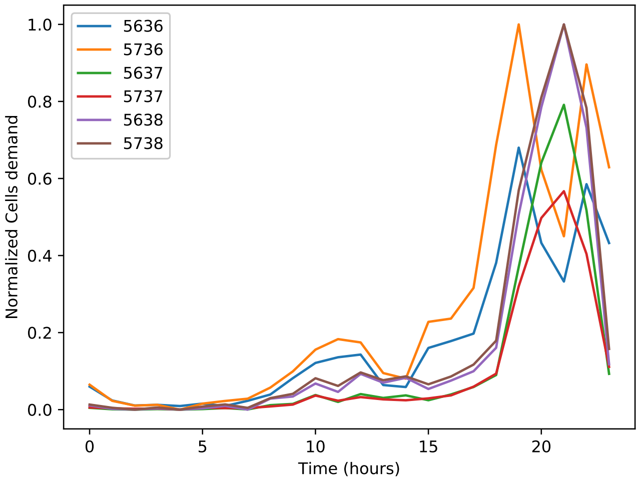


Figure 7.4: Example of demand time-series of SanSiro areas cells

Table 7.1: Simulation Parameters and Values

Parameters	**Values**
Initial Drone-cells battery level	100%
Drone-cells max rate	50 Mb/s
Cell max rate demand at peak hours	120 Mb/s
Battery life-time factor	0.5
Number of cells	6
Learning rate	0.75
Learning period (exploration)	300
exploitation period	50

As performance metric, we choose to use the network service ratio which is defined as follows:

$$Network\ service\ ratio = \frac{Total\ Bandwidth\ served}{Total\ bandwidth\ demand}$$

where $Total\ Bandwidth\ served$ is the sum of the served bandwidth by all drones and $Total\ bandwidth\ demand$ is the sum of all cell demands.

Our MARL model based on the enhanced hill climbing algorithm is compared to a MARL model with the classic hill algorithm. Figures 7.5-7.11 shows the simulations and comparison results between both models.

Figure 7.5 depicts the scenario simulation at 6pm (Fig 7.5(a)) and 7 pm (Fig 7.5(b)), the time of supporters arrival before starting the football match. We use in this scenario 8 drones to cover the area for both models. We can notice that after finishing the learning period (after 300 steps) the network service ratio converges to 1. During the learning period, the drones are exploring several options of action and communicating their decision to the coordinator. At the end of this period, the coordinator selects successfully the optimal joint action for all drones and then all cells are fully served. We also notice that 8 drones are sufficient, for our eHCS-based model, to serve the network. Whereas, the concurrent model based on HCS performs worse than our model. We notice that in this use-case scenario, that after the exploration phase the model fails to converge to 1 and the network service ratio converges to 0.9 and 0.6 at 6pm and 7pm respectively. Hence, the HCS based model performs 10% lower than our model at

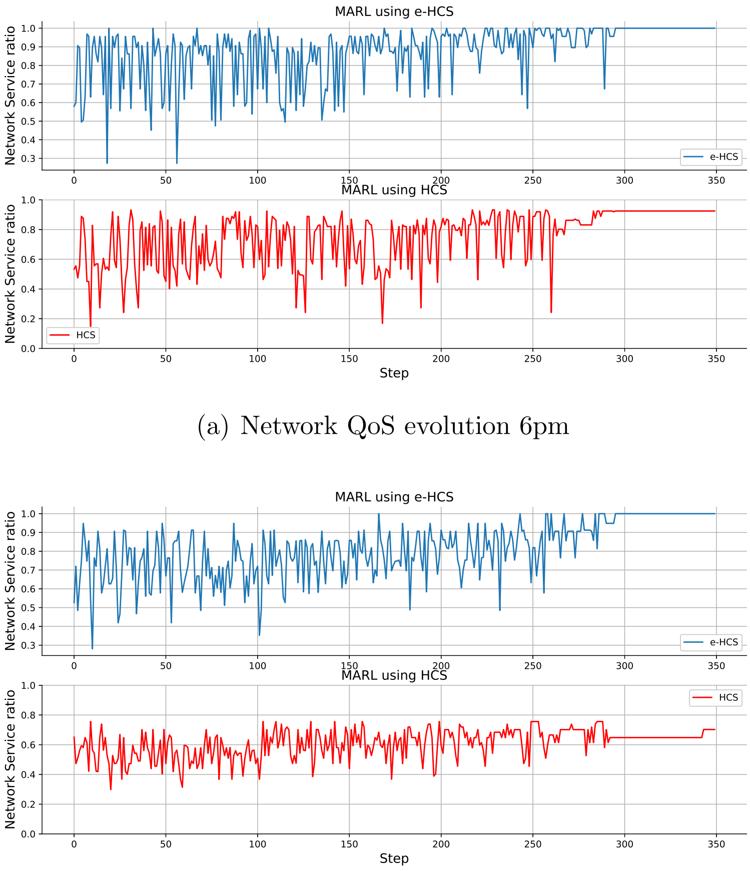
6pm and %40 lower at 7pm at which the demand is much higher. So, we can say that our model performs much better than the other model during peak hours. Moreover, the exploration period was not sufficient for the HCS based model to achieve the optimality and we conclude that our model is faster with 8 drones for the period between 6pm and 7pm.

Figures 7.6-7.11 show simulation and comparison results using 14 drones. We notice that for all time periods, the network service ratio converges to 1 after the learning period for our model. Whereas, the HCS based model success to converge to 1 only for three time periods; at 6pm, 7pm and 11pm while its network service ratio is lower than 1 for the rest of periods. Furthermore, for scenarios at 6pm, 7pm and 11pm, even that the HCS based model converges to 1 after the exploration period, it is slower that our model. Whereas, our model converges rapidly to one and even during the exploration period and this is due to the high number of drones and their availability at these periods.

Out of these periods, the HCS model fails to attempts optimality contrary to our model. The exploration period was again not sufficient to learn and find the optimal joint action at these periods. This is also due to the fact that most of drone batteries are discharged (Battery life-time is 2 slots) and they need one time slot to recharge. That is why the model re-reaches the optimality at 11pm again. Hence, in order to satisfy the global demand during the exploration period with the same period size, the HCS based model needs then more drones to serve all demands in shorter time, contrary to our model which success to serve the network efficiently with the same constraints.

Moreover, with this configuration, our enhanced model needs only 11 drones at total to cover the whole area during the whole football match event. These 11 drones alternate to serve the requested amount of data according to their battery life level.

Figure 7.12 depicts the deployment of drone-cells using our enhanced model at 9pm , time-slot corresponding to a heavy demand behavior. The colors represent the demand intensity at each cell. We can notice that 9 drones of 14 are deployed at cell locations that guarantee serving the total required bandwidth. Note that at this time 4



(a) Network QoS evolution 6pm

(b) Network QoS evolution 7pm

Figure 7.5: Network QoS evolution in function of execution steps

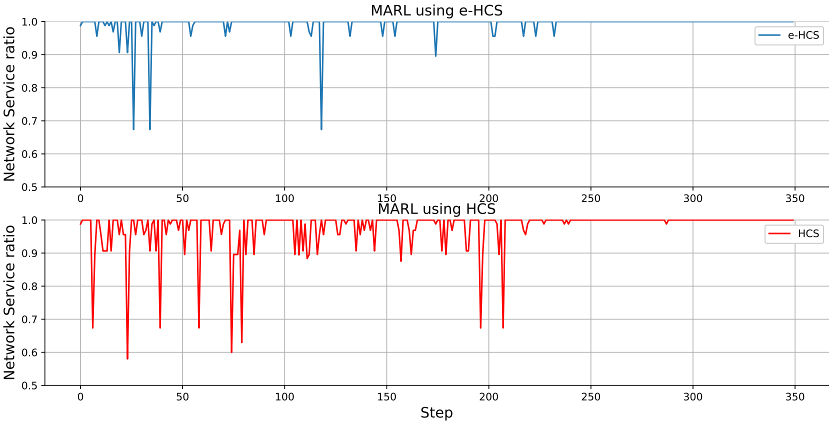


Figure 7.6: Network QoS evolution in function of execution steps with 14 drones and at 6pm

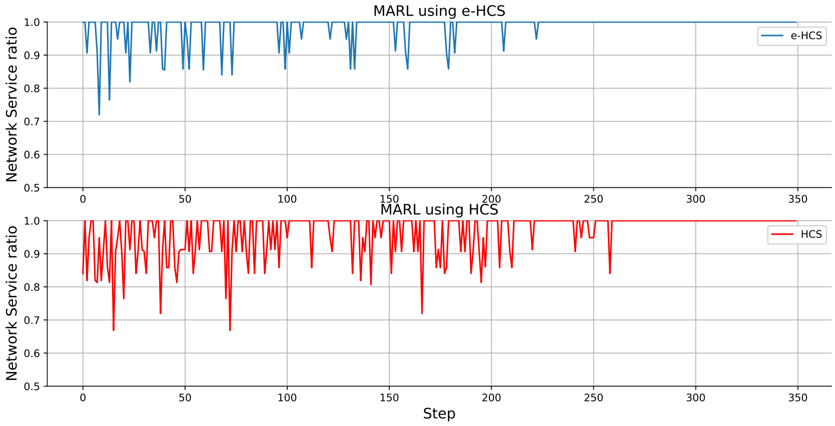


Figure 7.7: Network QoS evolution in function of execution steps with 14 drones and at 7pm

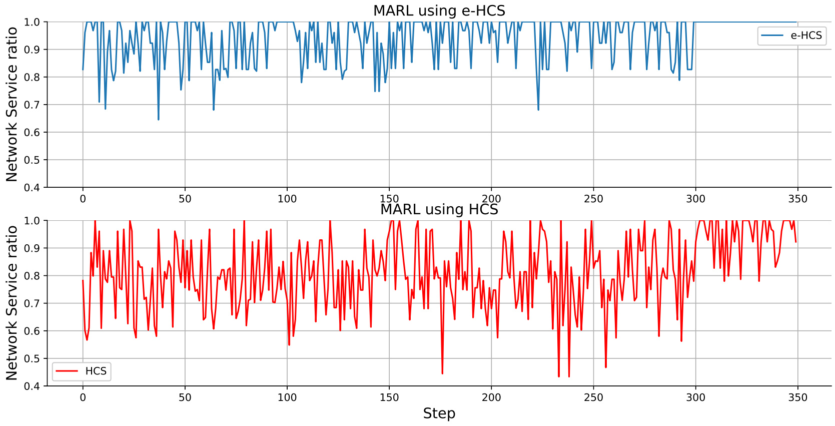


Figure 7.8: Network QoS evolution in function of execution steps with 14 drones and at 8pm

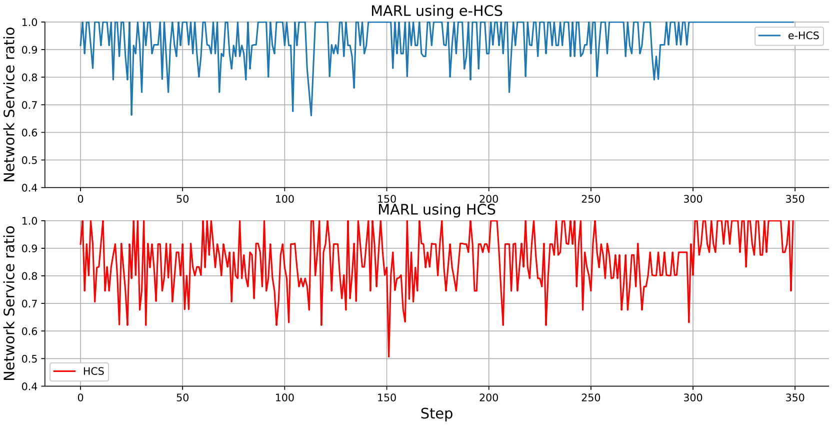


Figure 7.9: Network QoS evolution in function of execution steps with 14 drones and at 9pm

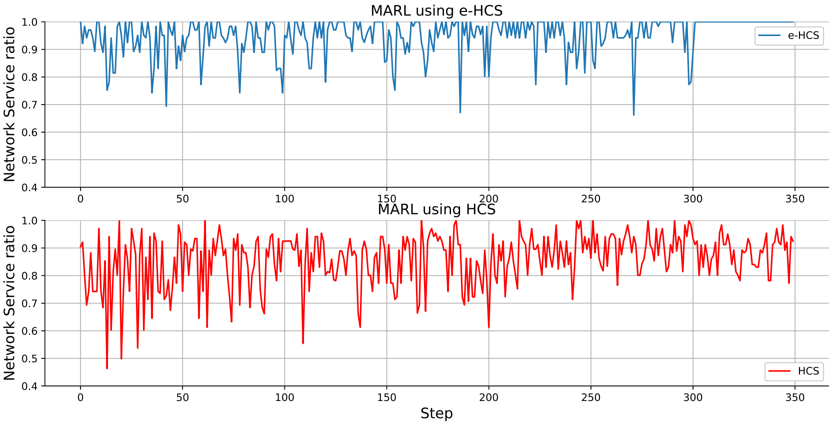


Figure 7.10: Network QoS evolution in function of execution steps with 14 drones and at 10pm

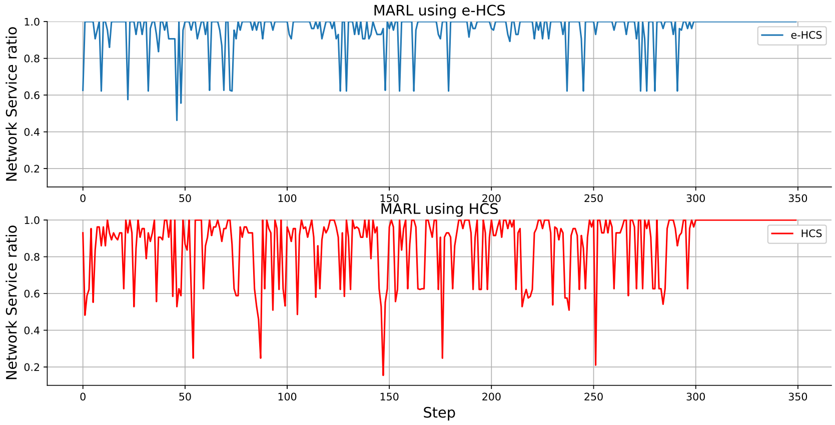


Figure 7.11: Network QoS evolution in function of execution steps with 14 drones and at 11pm

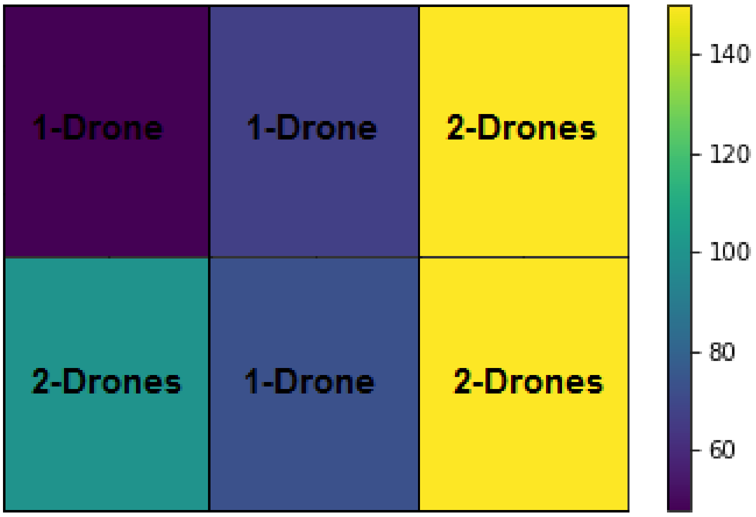


Figure 7.12: Illustration of drone-cells deployment at 9pm

drones are out of charge while 1 drone is not serving at all.

7.5 Conclusion

Drone-cells technology is emerging as a solution to support cellular network architectures. Drone-cells are flexible and provide a more dynamic solution for resource allocation in both scales, i.e, spatial and temporal. They allow to increase the bandwidth availability anytime and everywhere according the continuous rate demands. Their fast deployment provides network operators with a reliable solution to face sudden network overload or peak data demands during mass events, without interrupting services and guaranteeing better QoS. Although, their management is still complex and needs advanced and intelligent algorithms. Even their fast deployment, drone-cells networks suffer usually from coordination issues. In addition, battery technology is still limited and since drone-cells are equipped with battery that serves drone' movement and antenna transmission, energy presents a major constraints for their deployment.

We present in this chapter a solution based on drone-cells to support macro cells of the classic cellular network during mass event when data rate demand increases. To came up with management complexity we use a multi-agent reinforcement learning approach for this dynamic network deployment. We also propose an enhanced joint action selection algorithm to alleviate the coordination complexity between drone-cells agents and also speed up the search phase of the optimal joint action. Our model takes also into consideration the battery life constraints while aiming to maximize the network service ratio. Our solution is validated with real network traces and we provide a benchmarking analysis. Our model based on the enhanced joint action selection that we propose is compared with a model based on hill climbing search algorithm. Results show that our model outperforms the second model not only when the rate demand is lower but especially at peak time service.

Our model presents a better solution for network operators to manage their network dynamically and to provide efficiently a better QoS for users during mass events.

Chapter 8

Conclusions and Perspectives

8.1 Conclusions

This thesis constitutes a toolbox that consists on a collection of frameworks and tools for the mobile network data mining in such a manner that these frameworks complete each others. The main topic of the thesis is centered around the idea of real mobile network datasets analysis and network resource management and optimization. These datasets consist of CDRs metadata that include detailed information about users interactions with the networks. These datasets contain also important information for our analysis such as temporal and geographic scale information of mobile users activities. Given their large size and the fact that these are real-world datasets, information extracted from these datasets have intensively been used in this thesis to implement new algorithms that aim to enhance the network resource management and optimize the bandwidth allocation.

The issue with these CDR metadata is that they are provided in a raw format and the most relevant information are hidden within the large scale of datasets. This needs advanced tools, such as data mining technique and machine learning algorithms, to extract the relevant knowledge. In this context, we provide in this thesis a data mining study of a real-world CDRs dataset such as D4D challenge dataset provided by Orange Senegal and the big data challenge dataset provided by Telecom Italia. Our analysis method consists in clustering the base stations daily load time-series into relevant classes. We use for that a modified k-means clustering algorithm based on the dynamic time warping (DTW) distance. This clustering results in dividing the base station load time-series, extracted from the D4D challenge dataset, into three relevant classes. Each class belong to a specific base station load profile, such as a day-peak load profile, Constant load profile and Night-peak load profile. This first analysis phase permits to tag each base station with its corresponding profile class. The profiled data are used then to implement an automatic classification machine learning based on support vector machine (SVM). The classification algorithm allowed us to infer automatically the daily class of each base station time-series contained into the large-scale dataset. These information are important for network operators to propose dynamic algorithms for radio resource allocation that follow the instantaneous load fluctuation. To enhance the continuity of network services, it is important to estimate with high confidence how the bandwidth demand on a base station at a given time is shared among all the base stations in the following instants. We exploit then the classification of base stations profiles to analyze the mobility of the network bandwidth between areas. We use for this objective a novel form of the "origin-destination matrix based on the classification. This classified OD matrix provides aggregate information about the mobility of the load usage. In other words, it projects the mobility of the bandwidth between areas.

The second chapter of this thesis respond to the following question: Is it possible to use the CDRs dataset to implement an algorithm able to predict with higher accuracy the future network load? In the continuity of the first chapter, we address this issue in this thesis and we provide an analysis to study the characteristics of the base stations load time-series and we propose a prediction model based on support vector regression. Our solution is compared to other prediction techniques and the results proved the high efficiency of the SVR-based prediction model. We combine the network classification, bandwidth mobility and load prediction algorithms into a global framework that propose a dynamic network resource allocation techniques based on real data analysis. We evaluate the framework in the third chapter where we optimize the planning of a wireless mesh network proposed in the LCI4D project. In this chapter, we propose a MILP algorithm that provide a dynamic and fault-tolerant planning for a wireless mesh network that takes as input the cell load time-series resulting from the machine learning tools presented previously.

In the continuity of the CDRs dataset analysis and the load prediction, we propose in our thesis a second framework that consists on detecting pro-actively the anomalous load patterns of the network that may occur during mass events or network technical issues. Our anomaly detection framework is based on One-class SVM (OCSVM) and SVR algorithms. It is tested and validated with D4D challenge CDR and Italia telecom datasets. Comparison results show that our model outperforms other techniques. We use our framework to analyze the impact of the proliferous e-health data generated by the medical smart-phone applications.

To complete the on-line network anomalies detection framework, we propose another framework which consists on an innovative and dynamic network architecture based on drones-cells that is able to support the existent network macro-cells when abnormal bandwidth consumption occurs. Hence, we propose a dynamic solution for drone-cells networks that exploit real traces of demand profiles, output from the framework, and adapt in real time the deployment of drones-cell according these demands. In this part, we propose to optimize the deployment using the machine learning paradigm instead of classical linear programming models. Our proposal is based on a multi-agent reinforcement learning (MARL) approach. To provide a better MARL system, we need an efficient algorithm for joint action selection. Therefore, we propose a enhanced Hill Climbing search algorithm to speed up the selection of the optimal joint action and results show that our algorithm is much better and provide fast decision for drone-cells deployment.

To conclude this work, we remind that we integrate semi-synthetic data from our work into the network simulator NS3 and we published the code into an open-source platform [150] so that it is available and free-access for further research topics.

8.2 Perspectives

In this thesis, we are aiming to provide a general methodological process for analyzing CDR metadata and exploiting them for network optimization purposes. From the work presented here, several major lines of future work are available: First, we proposed a framework for classifying automatically base station profiles and we validate it using CDR from Dakar city. It is desirable to validate it also on traces and CDR dataset from other cities like London, New York etc. This we constitutes a more solid analysis study for the worldwide behavior of users bandwidth consumption.

In the other hand, our work focuses only on CDRs datasets from cellular networks. Therefore, we can also complete our research and analysis with other dataset such as WiFi consumption traces, HTTP-based datsets, social networks traces and combine them to infer a more detailed profiles of users bandwidth consumption behaviors.

In chapter 6 we provided an analysis study for the potential impact of eHealth data on cellular network but with used a simulated eHealth data. As the 5G network are under standardization and constitutes a hot topic of research, it will be interesting to enriches our datasets by a real eHealth data from real world applications and even other IoT applications data and D2D communications in order to prepare the road for the deployment of 5G networks.

Moreover, in chapter 7 we propose and evaluates a solution based on MARL for the dynamic deployment of Drone-cells networks. The solution is based on a machine learning programming and it is worth to evaluate it also against a linear programming solution. Linear programming algorithms provides usually the exact solution of the optimization problem but they suffers from processing time issues. Ongoing works are carried on to explore this direction.

Finally, we are working in integrating the framework developed in this thesis into a network simulator to provide researchers with a complete toolbox for networks analysis and optimization.

Bibliography

[1] Statista. Mobile internet usage worldwide - statistics facts:. `https://https://www.statista.com/topics/779/mobile-internet/`.

[2] Zenith Media. `https://https://www.zenithmedia.com/mobile-forecasts-75-internet-use-will-mobile-2017/`.

[3] Jeffrey G Andrews, Stefano Buzzi, Wan Choi, Stephen V Hanly, Angel Lozano, Anthony CK Soong, and Jianzhong Charlie Zhang. What will 5g be? *IEEE Journal on selected areas in communications*, 32(6):1065–1082, 2014.

[4] Guido R Hiertz, Dee Denteneer, Sebastian Max, Rakesh Taori, Javier Cardona, Lars Berlemann, and Bernhard Walke. Ieee 802.11 s: the wlan mesh standard. *Wireless Communications, IEEE*, 17(1):104–111, 2010.

[5] Srikant Manas Kala, M Reddy, and Bheemarjuna Reddy Tamma. Predicting performance of channel assignments in wireless mesh networks through statistical interference estimation. *arXiv preprint arXiv:1503.08687*, 2015.

[6] Farah Kandah, Weiyi Zhang, Yashaswi Singh, and Juan Li. Interference-aware robust wireless mesh network design. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[7] Yonggyu Kim, Doyoung Jung, Youngdoo Kim, Sungchang Choi, and Joongsoo Ma. Efficient interference-aware channel allocation in multi-radio wireless mesh networks. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 920–925. IEEE, 2012.

[8] DG Narayan, M Uma, Gabriele Pavan, and S Suraj. Cl-ild: A cross layer interference-load and delay aware routing metric for multi-radio wireless mesh network. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 181–186. IEEE, 2013.

[9] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223–2234. IEEE, 2005.

[10] Elham Kalantari, Muhammad Zeeshan Shakir, Halim Yanikomeroglu, and Abbas Yongacoglu. Backhaul-aware robust 3d drone placement in 5g+ wireless networks. *arXiv preprint arXiv:1702.08395*, 2017.

[11] GDH. Mobile phone and tablet forensics:. `https://https://globaldigitalforensics.com/computer_forensics_services/mobile-forensics/`.

[12] Orange. D4d challenge:. `https://http://www.d4d.orange.com`.

[13] Sébastian Grauwin, Stanislav Sobolevsky, Simon Moritz, István Gódor, and Carlo Ratti. Towards a comparative science of cities: Using mobile traffic records in new york, london, and hong kong. In *Computational approaches for urban environments*, pages 363–387. Springer, 2015.

[14] Jameson L Toole, Michael Ulm, Marta C González, and Dietmar Bauer. Inferring land use from mobile phone activity. In *Proceedings of the ACM SIGKDD international workshop on urban computing*, pages 1–8. ACM, 2012.

[15] Angelo Furno, Marco Fiore, Razvan Stanica, Cezary Ziemlicki, and Zbigniew Smoreda. A tale of ten cities: Characterizing signatures of mobile traffic in urban areas. *IEEE Transactions on Mobile Computing*, 2016.

[16] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001.

[17] Nitin Kumar, Venkata Nishanth Lolla, Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, and Li Wei. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 531–535. SIAM, 2005.

[18] Et Fcc. Docket no 03-222 notice of proposed rule making and order, 2003.

[19] Alexis Sultan, Majed ElKouki, Hossam Afifi, Vincent Gauthier, and Michel Marot. A dynamic femto cell architecture using tv whitespace improving user experience of urban crowds. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 886–891. IEEE, 2015.

[20] Milind M Buddhikot. Understanding dynamic spectrum access: Models, taxonomy and challenges. In *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 649–663. IEEE, 2007.

[21] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer networks*, 50(13):2127–2159, 2006.

[22] Qing Zhao and Ananthram Swami. A survey of dynamic spectrum access: Signal processing and networking perspectives. In *Acoustics, speech and signal processing, 2007. ICASSP 2007. IEEE international conference on*, volume 4, pages IV–1349. IEEE, 2007.

[23] Qing Zhao and Brian M Sadler. A survey of dynamic spectrum access. *IEEE signal processing magazine*, 24(3):79–89, 2007.

[24] Joseph Mitola. Cognitive radio—an integrated agent architecture for software defined radio. 2000.

[25] Mishra Vishram, Lau Chiew Tong, and Chan Syin. List multi-coloring based fair channel allocation policy for self coexistence in cognitive radio networks with qos provisioning. In *Region 10 Symposium, 2014 IEEE*, pages 99–104. IEEE, 2014.

[26] Yuan Lu and Alexandra Duel-Hallen. A sensing contribution-based two-layer game for channel selection and spectrum access in cognitive radio ad-hoc networks. *IEEE Transactions on Wireless Communications*, 2018.

[27] Vani Shrivastav, Sanjay K Dhurandher, Isaac Woungang, Vinesh Kumar, and Joel JPC Rodrigues. Game theory-based channel allocation in cognitive radio networks. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–5. IEEE, 2016.

[28] Tianqi Zhou, Bing Chen, Chunsheng Zhu, and Xiangping Zhai. Tpahs: A truthful and profit maximizing double auction for heterogeneous spectrums. In *Trustcom/BigDataSE/I SPA, 2016 IEEE*, pages 27–33. IEEE, 2016.

[29] Long Chen, Liusheng Huang, Zehao Sun, Hongli Xu, and Hansong Guo. Spectrum combinatorial double auction for cognitive radio network with ubiquitous network resource providers. *IET Communications*, 9(17):2085–2094, 2015.

[30] Yong Xiao, Tim Forde, Irene Macaluso, Luiz A DaSilva, and Linda Doyle. Spatial spectrum sharing-based carrier aggregation for heterogeneous networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 2591–2596. IEEE, 2012.

[31] Yufeng Wang, Feilong Tang, Yanqin Yang, Jie Li, Wenchao Xu, and Jinsong Wu. A qos-guaranteed adaptive cooperation scheme in cognitive radio network. In *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*, pages 516–523. IEEE, 2016.

[32] Mikio Hasegawa, Hiroshi Hirai, Kiyohito Nagano, Hiroshi Harada, and Kazuyuki Aihara. Optimization for centralized and

decentralized cognitive radio networks. *Proceedings of the IEEE*, 102(4):574–584, 2014.

[33] Juan Carlos Merlano-Duncan, Shree Krishna Sharma, Symeon Chatzinotas, Björn Ottersten, and Xianbin Wang. Multi-antenna based one-bit spatio-temporal wideband sensing for cognitive radio networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[34] Md Zulfikar Alom, Tapan Kumar Godder, Mohammad Nayeem Morshed, and Asmaa Maali. Enhanced spectrum sensing based on energy detection in cognitive radio network using adaptive threshold. In *Networking, Systems and Security (NSysS), 2017 International Conference on*, pages 138–143. IEEE, 2017.

[35] Xiangxia Sun, Tengyi Zhang, and Danny HK Tsang. Optimal energy-efficient cooperative sensing scheduling for cognitive radio networks with qos guarantee. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1825–1830. IEEE, 2011.

[36] Woongsoo Na, Jongha Yoon, Sungrae Cho, David Wesley Griffith, and Nada Golmie. Centralized cooperative directional spectrum sensing for cognitive radio networks. *IEEE Transactions on Mobile Computing*, 2017.

[37] Yifan Wang, Husheng Li, and Lijun Qian. Belief propagation and quickest detection-based cooperative spectrum sensing in heterogeneous and dynamic environments. *IEEE Transactions on Wireless Communications*, 16(11):7446–7459, 2017.

[38] Wenchi Cheng, Xi Zhang, and Hailin Zhang. Full-duplex spectrum-sensing and mac-protocol for multichannel nontime-slotted cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 33(5):820–831, 2015.

[39] Yi Liu, Rong Yu, Miao Pan, Yan Zhang, and Shengli Xie. Sd-mac: Spectrum database-driven mac protocol for cognitive machine-to-machine networks. *IEEE Transactions on Vehicular Technology*, 66(2):1456–1467, 2017.

[40] Ying He, Beeshanga Abewardana Jayawickrama, Eryk Dutkiewicz, Srikathyayani Srikanteswara, and Markus Dominik Mueck. Priority access and general authorised access interference mitigation in spectrum access system. *IEEE Transactions on Vehicular Technology*, 2018.

[41] Cong Xiong, Lu Lu, and Geoffrey Ye Li. Energy-efficient spectrum access in cognitive radios. *IEEE Journal on Selected Areas in Communications*, 32(3):550–562, 2014.

[42] Gozde Ozcan and M Cenk Gursoy. Energy-efficient power adaptation for cognitive radio systems under imperfect channel sensing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 706–711. IEEE, 2014.

[43] Cong Xiong, Geoffrey Ye Li, Shunqing Zhang, Yan Chen, and Shugong Xu. Energy-and spectral-efficiency tradeoff in downlink ofdma networks. *IEEE transactions on wireless communications*, 10(11):3874–3886, 2011.

[44] Salim Eryigit, Gurkan Gur, Suzan Bayhan, and Tuna Tugcu. Energy efficiency is a subtle concept: Fundamental trade-offs for cognitive radio networks. *IEEE Communications Magazine*, 52(7):30–36, 2014.

[45] Haijun Zhang, Chunxiao Jiang, Norman C Beaulieu, Xiaoli Chu, Xianbin Wang, and Tony QS Quek. Resource allocation for cognitive small cell networks: A cooperative bargaining game theoretic approach. *IEEE Transactions on Wireless Communications*, 14(6):3481–3493, 2015.

[46] Jianchao Zheng, Yueming Cai, Ning Lu, Yuhua Xu, and Xuemin Shen. Stochastic game-theoretic spectrum access in distributed and dynamic environment. *IEEE transactions on vehicular technology*, 64(10):4807–4820, 2015.

[47] *White space devices (WSD); wireless access systems operating in the 470 MHz to 790 MHz TV broadcast band; harmonized EN*

covering the essential requirements of article 3.2 of the RTTE directive. ETSI EN 301 598., 2014.

[48] *The unlicensed spectrum usage for future IMT technologies effecient LTE technologie enables better performance and experience.* HUAWEI, 2014.

[49] Wenjie Zhang, Guanglin Zhang, Yifeng Zheng, Liwei Yang, and Yeo Chai Kiat. Spectrum sharing for heterogeneous networks and application systems in tv white spaces. *IEEE Access*, 2018.

[50] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11 b mesh network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42. ACM, 2005.

[51] Edoardo Amaldi, Antonio Capone, Matteo Cesana, Ilario Filippini, and Federico Malucelli. Optimization models and methods for planning wireless mesh networks. *Computer Networks*, 52(11):2159–2171, 2008.

[52] Djohara Benyamina, Abdelhakim Hafid, Michel Gendreau, and Nasreddine Hallam. Optimization models for planning wireless mesh networks: a comparative study. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6. IEEE, 2009.

[53] Chun-cheng Chen, Chandra Chekuri, and Diego Klabjan. Topology formation for wireless mesh network planning. In *INFOCOM 2009, IEEE*, pages 2671–2675. IEEE, 2009.

[54] Ting-Yu Lin, Kai-Chiuan Hsieh, and Hsin-Chun Huang. Applying genetic algorithms for multiradio wireless mesh network planning. *Vehicular Technology, IEEE Transactions on*, 61(5):2256–2270, 2012.

[55] Felix Juraschek. Interference-aware wireless mesh networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–2. IEEE, 2013.

[56] Habiba Skalli, Sudip Ghosh, Sajal K Das, Luciano Lenzini, and Marco Conti. Channel assignment strategies for multiradio wireless mesh networks: issues and solutions. *Communications Magazine, IEEE*, 45(11):86–95, 2007.

[57] K Venkata Subbaiah and MM Naidu. An efficient interference aware channel allocation algorithm for wireless mesh networks. In *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*, pages 416–420. IEEE, 2015.

[58] Nachwan Mufti Adriansyah, Muhamad Asvial, and Bagio Budiardjo. Decision-based link scheduling approximation algorithm with sinr relaxation for wireless mesh network. In *Computer, Control, Informatics and its Applications (IC3INA), 2015 International Conference on*, pages 99–103. IEEE, 2015.

[59] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM, 2004.

[60] Svilen Ivanov, Edgar Nett, and Ralf Schumann. Fault-tolerant base station planning of wireless mesh networks in dynamic industrial environments. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8. IEEE, 2010.

[61] Junzhou Luo, Wenjia Wu, and Ming Yang. Interference-aware gateway placement for wireless mesh networks with fault tolerance assurance. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2373–2380. IEEE, 2010.

[62] Antonio Capone, Filippo Malandra, and Brunilde Sansò. Energy savings in wireless mesh networks in a time-variable context. *Mobile Networks and Applications*, 17(2):298–311, 2012.

[63] S Hammami, H Afifi, M Marot, and V Gauthier. Network planning tool based on network classification and load prediction.

In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2016.

[64] Elham Kalantari, Halim Yanikomeroglu, and Abbas Yongacoglu. On the number and 3d placement of drone base stations in wireless cellular networks. In *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*, pages 1–6. IEEE, 2016.

[65] Seongjoon Park, Hyunsoon Kim, Kangho Kim, and Hwangnam Kim. Drone formation algorithm on 3d space for a drone-based network infrastructure. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–6. IEEE, 2016.

[66] Ahmad Alsharoa, Hakim Ghazzai, Abdullah Kadri, and Ahmad E Kamal. Energy management in cellular hetnets assisted by solar powered drone small cells. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[67] Margot Deruyck, Jorg Wyckmans, Luc Martens, and Wout Joseph. Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*, pages 1–7. IEEE, 2016.

[68] Aymen Jaziri, Ridha Nasri, and Tijani Chahed. Congestion mitigation in 5g networks using drone relays. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 233–238. IEEE, 2016.

[69] Xu Li, Dongning Guo, Huarui Yin, and Guo Wei. Drone-assisted public safety wireless broadband network. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*, pages 323–328. IEEE, 2015.

[70] Milan Erdelj, Enrico Natalizio, Kaushik R Chowdhury, and Ian F Akyildiz. Help from the sky: Leveraging uavs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32, 2017.

[71] Myungsoo Jun and Raffaello DAndrea. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In *Cooperative control: models, applications and algorithms*, pages 95–110. Springer, 2003.

[72] Honghui Dong, Xiaoqing Ding, Yan Shi, Limin Jia, Yong Qin, Lianyu Chu, et al. Urban traffic commuting analysis based on mobile phone data. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 611–616. IEEE, 2014.

[73] Minjie Wang, Su Yang, Yi Sun, and Jun Gao. Predicting human mobility from region functions. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on*, pages 540–547. IEEE, 2016.

[74] Thomas Louail, Maxime Lenormand, Oliva G Cantu Ros, Miguel Picornell, Ricardo Herranz, Enrique Frias-Martinez, José J Ramasco, and Marc Barthelemy. From mobile phone data to the spatial structure of cities. *Scientific reports*, 4, 2014.

[75] Michele Berlingerio, Francesco Calabrese, Giusy Di Lorenzo, Rahul Nair, Fabio Pinelli, and Marco Luca Sbodio. Allaboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 663–666. Springer, 2013.

[76] Sonia Khetarpaul, Rashmi Chauhan, SK Gupta, L Venkata Subramaniam, and Ullas Nambiar. Mining gps data to determine interesting locations. In *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*, page 8. ACM, 2011.

[77] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[78] Wenhao Huang, Man Li, Weisong Hu, Guojie Song, and Kunqing Xie. Automated urban location annotation on mobile records. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on*, pages 951–956. IEEE, 2013.

[79] Diala Naboulsi, Razvan Stanica, and Marco Fiore. Classifying call profiles in large-scale mobile traffic datasets. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1806–1814. IEEE, 2014.

[80] Ionut Trestian, Supranamaya Ranjan, Aleksandar Kuzmanovic, and Antonio Nucci. Measuring serendipity: connecting people, locations and interests in a mobile 3g network. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 267–279. ACM, 2009.

[81] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE Transactions on Services Computing*, 9(5):796–805, 2016.

[82] Ali Yadavar Nikravesh, Samuel A Ajila, Chung-Horng Lung, and Wayne Ding. Mobile network traffic prediction using mlp, mlpwd, and svm. In *Big Data (BigData Congress), 2016 IEEE International Congress on*, pages 402–409. IEEE, 2016.

[83] M Zubair Shafiq, Lusheng Ji, Alex X Liu, and Jia Wang. Characterizing and modeling internet traffic dynamics of cellular devices. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 305–316. ACM, 2011.

[84] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, 2012.

[85] Sunhee Baek, Donghwoon Kwon, Jinoh Kim, Sang C Suh, Hyunjoo Kim, and Ikkyun Kim. Unsupervised labeling for supervised anomaly detection in enterprise and cloud networks. In *Cyber*

Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on, pages 205–210. IEEE, 2017.

[86] Juliette Dromard, Gilles Roudiere, and Philippe Owezarski. Online and scalable unsupervised network anomaly detection method. *IEEE Transactions on Network and Service Management*, 14(1):34–47, 2017.

[87] Sergey Chernov, Michael Cochez, and Tapani Ristaniemi. Anomaly detection algorithms for the sleeping cell detection in lte networks. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–5. IEEE, 2015.

[88] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2004.

[89] Eamonn Keogh, Stefano Lonardi, and Bill'Yuan-chi' Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556. ACM, 2002.

[90] Varun Chandola, Varun Mithal, and Vipin Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 743–748. IEEE, 2008.

[91] David Goergen, Veena Mendiratta, Thomas Engel, et al. Identifying abnormal patterns in cellular communication flows. In *Proceedings of Principles, Systems and Applications on IP Telecommunications*, pages 1–6. ACM, 2013.

[92] Robert Gwadera, Aristides Gionis, and Heikki Mannila. Optimal segmentation using tree models. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 244–253. IEEE, 2006.

[93] Robert Gwadera, Mikhail J Atallah, and Wojciech Szpankowski. Reliable detection of episodes in event sequences. *Knowledge and Information Systems*, 7(4):415–437, 2005.

[94] Li Wei, Nitin Kumar, Venkata Nishanth Lolla, Eamonn J Keogh, Stefano Lonardi, and Chotirat (Ann) Ratanamahatana. Assumption-free anomaly detection in time series. In *SSDBM*, volume 5, pages 237–242, 2005.

[95] Christoph C Michael and Anup Ghosh. Two state-based approaches to program-based anomaly detection. In *Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference*, pages 21–30. IEEE, 2000.

[96] Carla Marceau. Characterizing the behavior of a program using multiple-length n-grams. In *Proceedings of the 2000 workshop on New security paradigms*, pages 101–110. ACM, 2001.

[97] Siyuan Peng, Guoyin Wang, Zhixing Li, and Jie Yang. The application of markov model based equivalence class generalization in network anomaly detection. In *Cloud Computing and Big Data Analysis (ICCCBDA), 2017 IEEE 2nd International Conference on*, pages 389–393. IEEE, 2017.

[98] João BD Cabrera, Lundy Lewis, and Raman K Mehra. Detection and classification of intrusions and faults using sequences of system calls. *Acm sigmod record*, 30(4):25–34, 2001.

[99] Dipankar Dasgupta and Nivedita Sumi Majumdar. Anomaly detection in multidimensional data using negative selection algorithm. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1039–1044. IEEE, 2002.

[100] Mikhail Atallah, Wojciech Szpankowski, and Robert Gwadera. Detection of significant sets of episodes in event sequences. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 3–10. IEEE, 2004.

[101] Koosha Golmohammadi and Osmar R Zaiane. Time series contextual anomaly detection for detecting market manipulation in stock market. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

[102] Péter Szilágyi and Szabolcs Novaczki. An automatic detection and diagnosis framework for mobile communication systems. *IEEE transactions on Network and Service Management*, 9(2):184–197, 2012.

[103] Rana M Khanafer, Beatriz Solana, Jordi Triola, Raquel Barco, Lars Moltsen, Zwi Altman, and Pedro Lazaro. Automated diagnosis for umts networks using bayesian network approach. *IEEE Transactions on Vehicular Technology*, 57(4):2451–2461, 2008.

[104] Samira Rezaei, Hamidreza Radmanesh, Payam Alavizadeh, Hamidreza Nikoofar, and Farshad Lahouti. Automatic fault detection and diagnosis in cellular networks using operations support systems data. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 468–473. IEEE, 2016.

[105] Evgeny Burnaev and Dmitry Smolyakov. One-class svm with privileged information and its application to malware detection. *arXiv preprint arXiv:1609.08039*, 2016.

[106] G Prashanth, V Prashanth, P Jayashree, and N Srinivasan. Using random forests for network-based anomaly detection at active routers. In *Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on*, pages 93–96. IEEE, 2008.

[107] Kabir Yunus Peerbhay, Onisimo Mutanga, and Riyad Ismail. Random forests unsupervised classification: The detection and mapping of solanum mauritianum infestations in plantation forestry using hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3107–3122, 2015.

[108] R Ismail and O Mutanga. Discriminating the early stages of sirex noctilio infestation using classification tree ensembles and shortwave infrared bands. *International journal of remote sensing*, 32(15):4249–4266, 2011.

[109] Zhiguo Ding and Minrui Fei. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17, 2013.

[110] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[111] Seif Eddine Hammami, Hossam Afifi, Michel Marot, and Vincent Gauthier. Network planning tool based on network classification and load prediction. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–6. IEEE, 2016.

[112] Santi Phithakkitnukoon, Teerayut Horanont, Giusy Di Lorenzo, Ryosuke Shibasaki, and Carlo Ratti. Activity-aware map: Identifying human daily activity pattern using mobile phone data. In *International Workshop on Human Behavior Understanding*, pages 14–25. Springer, 2010.

[113] Shang Weixiong Zhu Yanfeng, Zhou Jin, and Ying Chun. Collecting and analyzing mobility data from mobile network. In *Broadband Network & Multimedia Technology, 2009. IC-BNMT'09. 2nd IEEE International Conference on*, pages 810–815. IEEE, 2009.

[114] Sibren Isaacman, Richard Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2012.

[115] Yang Ye, Yu Zheng, Yukun Chen, Jianhua Feng, and Xing Xie. Mining individual life pattern based on location history. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 1–10. IEEE, 2009.

[116] Sahar Hoteit, Guangshuo Chen, Aline Viana, and Marco Fiore. Spatio-temporal completion of call detail records for human mobility analysis. In *Rencontres Francophones sur la Conception de Protocoles, lÉvaluation de Performance et lExpérimentation des Réseaux de Communication*, 2017.

[117] Pu Wang, Timothy Hunter, Alexandre M Bayen, Katja Schechtner, and Marta C González. Understanding road usage patterns in urban areas. *Scientific reports*, 2:1001, 2012.

[118] Vincent D Blondel, Adeline Decuyper, and Gautier Krings. A survey of results on mobile phone datasets analysis. *EPJ Data Science*, 4(1):1, 2015.

[119] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

[120] Diala Naboulsi and Marco Fiore. Characterizing the instantaneous connectivity of large-scale urban vehicular networks. *IEEE Transactions on Mobile Computing*, 16(5):1272–1286, 2017.

[121] Siyang Qin, Youchen Zuo, Yaguan Wang, Xuan Sun, and Honghui Dong. Travel trajectories analysis based on call detail record data. In *Control And Decision Conference (CCDC), 2017 29th Chinese*, pages 7051–7056. IEEE, 2017.

[122] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[123] James MacQueen et al. Some methods for classification and analysis of multivariate observations. 1967.

[124] Víctor Soto and Enrique Frías-Martínez. Automated land use identification using cell-phone records. In *Proceedings of the 3rd ACM international workshop on MobiArch*, pages 17–22. ACM, 2011.

[125] Blerim Cici, Minas Gjoka, Athina Markopoulou, and Carter T Butts. On the decomposition of cell phone activity patterns and their connection with urban ecology. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 317–326. ACM, 2015.

[126] Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

[127] Alexis Sultan. *Méthodes et outils d'analyse de données de signalisation mobile pour l'étude de la mobilité humaine.* PhD thesis, Institut National des Télécommunications, 2016.

[128] Jingtao Ma, Huan Li, Fang Yuan, and Thomas Bauer. Deriving operational origin-destination matrices from large scale mobile phone data. *International Journal of Transportation Science and Technology*, 2(3):183–204, 2013.

[129] Xinxing Pan and Brian Lee. A comparison of support vector machines and artificial neural networks for mid-term load forecasting. In *Industrial Technology (ICIT), 2012 IEEE International Conference on*, pages 95–101. IEEE, 2012.

[130] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[131] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[132] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol (olsr). 2003.

[133] Open-Mesh. B.a.t.m.a.n. advanced:. `https://www.open-mesh.org/projects/batman-adv/wiki/Wiki`.

[134] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[135] Telecom Italia. Telecom italia big data challenge. (2015):. `https://http://www.telecomitalia.com/tit/en/bigdatachallenge.html`.

[136] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008.

[137] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.

[138] Eduardo HM Pena, Sylvio Barbon, Joel JPC Rodrigues, and Mario Lemes Proença. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1–6. IEEE, 2014.

[139] ITU. Scaling e-health services in step with ict transformation:. `https://https://www.itu.int/ITU-D/cyb/app/docs/Scaling%20e-Health-E.pdf`.

[140] Jing Peng and Ronald J Williams. Incremental multi-step q-learning. In *Machine Learning Proceedings 1994*, pages 226–232. Elsevier, 1994.

[141] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[142] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[143] Alborz Geramifard, Thomas J Walsh, Stefanie Tellex, Girish Chowdhary, Nicholas Roy, Jonathan P How, et al. A tutorial on linear function approximators for dynamic programming and

reinforcement learning. *Foundations and Trends® in Machine Learning*, 6(4):375–451, 2013.

[144] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[145] Francesc Wilhelmi, Boris Bellalta, Cristina Cano, and Anders Jonsson. Implications of decentralized q-learning resource allocation in wireless networks. *arXiv preprint arXiv:1705.10508*, 2017.

[146] Ning Xu, Huyin Zhang, Fang Xu, and Zhiyong Wang. Q-learning based interference-aware channel handoff for partially observable cognitive radio ad hoc networks. *Chinese Journal of Electronics*, 26(4):856–863, 2017.

[147] David Masad and Jacqueline Kazil. Mesa: An agent-based modeling framework:. `https://http://conference.scipy.org/proceedings/scipy2015/pdfs/jacqueline_kazil.pdf`.

[148] Scott Proper and Prasad Tadepalli. Scaling model-based average-reward reinforcement learning for product delivery. In *European Conference on Machine Learning*, pages 735–742. Springer, 2006.

[149] Chi Cheng, Zhangqing Zhu, Bo Xin, and Chunlin Chen. A multi-agent reinforcement learning algorithm based on stackelberg game. In *Data Driven Control and Learning Systems (DDCLS), 2017 6th*, pages 727–732. IEEE, 2017.

[150] Seif Eddine Hammami and Aziza Ben Mosbah. Ns3-realistic traffic generator:. `https://https://github.com/SHM91/NS3-RealisticTrafficGen1`.