

Universidad de Málaga

Escuela Técnica Superior de Ingeniería de Telecomunicación

Programa de Doctorado en Ingeniería de Telecomunicación



UNIVERSIDAD DE MÁLAGA

TESIS DOCTORAL

# DATA-DRIVEN SELF-MANAGEMENT OF CELLULAR RADIO ACCESS NETWORKS

Autora:

CAROLINA GIJÓN MARTÍN

Directores:

MATÍAS TORIL GENOVÉS


SALVADOR LUNA RAMÍREZ





UNIVERSIDAD  
DE MÁLAGA

AUTOR: Carolina Gijón Martín

 <https://orcid.org/0000-0001-6204-0604>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): [riuma.uma.es](http://riuma.uma.es)



## AUTORIZACIÓN PARA LECTURA DE LA TESIS DOCTORAL

Por la presente, Dr. D. Matías Toril Genovés y Dr. D. Salvador Luna Ramírez, profesores doctores del Departamento de Ingeniería de Comunicaciones de la Universidad de Málaga, certifican que la doctoranda Carolina Gijón Martín ha realizado en el Departamento de Ingeniería de Comunicaciones de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su TESIS DOCTORAL titulada

### **Data-driven self-management of cellular radio access networks**

Dicho trabajo ha dado lugar a las siguientes publicaciones en revistas y aportaciones a congresos que no han sido utilizadas en tesis anteriores:

1. C. Gijón, M. Toril, S. Luna, M.L. Marí, "A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks", *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414-9424, oct. 2019.
2. C. Gijón, M. Toril, M. Solera, S. Luna, L. Jiménez, "Encrypted traffic classification based on unsupervised learning in cellular radio access networks", *IEEE Access*, vol. 8, pp. 167252 - 167263, sep. 2020.
3. C. Gijón, M. Toril, S. Luna, J.L. Bejarano, M.L. Marí, "Estimating pole capacity from radio network performance statistics by supervised learning", *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2090 - 2101, dic. 2020.
4. C. Gijón, M. Toril, S. Luna, M.L. Marí, J.M. Ruiz, "Long-term data traffic forecasting for network dimensioning in LTE with short time series", *Electronics*, vol. 10, p. 1151, may. 2021.
5. C. Gijón, M. Toril, S. Luna, "Data-driven estimation of throughput performance in sliced radio access networks via supervised learning", *IEEE Transactions on Network and Service Management*, aceptada en sep. 2022.
6. C. Gijón, M. Toril, S. Luna, M. L. Marí, "A data-driven user steering algorithm for optimizing user experience in multi-tier LTE networks", 9th MC and scientific meeting of COST CA15104 (IRACON), Dublín (Irlanda), ene. 2019.
7. C. Gijón, M. Toril, S. Luna, M.L. Marí, "Mejora de la calidad de experiencia en redes LTE multi-portadora", XXXIV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2019), Sevilla (España), sep. 2019.
8. C. Gijón, M. Toril, S. Luna, J. L Bejarano, M. L. Marí, "Estimación de la capacidad en redes LTE mediante aprendizaje supervisado", XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020), Málaga (España), sep. 2020.
9. C. Gijón, M. Toril, S. Luna, "Modelling performance in sliced radio access networks with supervised learning", 1th scientific meeting of COST CA20120 (INTERACT), Bolonia (Italia), feb. 2022.

10. C. Gijón, M. Toril, S. Luna, "Modelado de rendimiento de segmento en redes de acceso radio mediante aprendizaje supervisado", XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022), Málaga (España), sep. 2022.

Por todo ello, consideran que esta Tesis es apta para su presentación al Tribunal que ha de juzgarla. Y para que conste a efectos de lo establecido, AUTORIZAN la presentación de esta Tesis en la Universidad de Málaga.

En Málaga, a 31 de enero de 2023.

Fdo.: Dr. D. Matías Toril Genovés

Fdo.: Dr. D. Salvador Luna Ramírez



## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña CAROLINA GIJÓN MARTÍN

Estudiante del programa de doctorado INGENIERÍA DE TELECOMUNICACIÓN de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: DATA-DRIVEN SELF-MANAGEMENT OF CELLULAR RADIO ACCESS NETWORKS

Realizada bajo la tutorización de MATÍAS TORIL GENOVÉS y dirección de MATÍAS TORIL GENOVÉS Y SALVADOR LUNA RAMÍREZ (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 31 de ENERO de 2023

Fdo.: CAROLINA GIJÓN MARTÍN Doctorando/a	Fdo.: MATÍAS TORIL GENOVÉS Tutor/a
Fdo.: MATÍAS TORIL GENOVÉS Y SALVADOR LUNA RAMÍREZ	





UNIVERSIDAD  
DE MÁLAGA



Escuela de Doctorado

Director/es de tesis

UNIVERSIDAD  
DE MÁLAGA



EFQM AENOR



Edificio Pabellón de Gobierno. Campus El Ejido.  
29071  
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10  
E-mail: doctorado@uma.es

*"A journey of a thousand miles begins with a single step."*



UNIVERSIDAD  
DE MÁLAGA



# Acknowledgements

The most rewarding about achieving goals is sharing happiness with people you appreciate. This thesis is the outcome of an intense four-year journey that I have been lucky to travel with fantastic people to whom I have much to thank.

First, I would like to express my foremost gratitude to my supervisors, Matías and Salvador, for their trust, guidance and support all these years and for pushing me to fulfill my professional goals. Salva introduced me to the exciting world of mobile communications. Had it not been for him, I wouldn't probably be writing these words today. Moreover, with his charisma, he manages to help me focus when I don't see the wood for the trees. Matías fills my mind with brilliant ideas in those inspiring and insightful discussions (or may I say lessons) and has devoted countless hours to help me whenever I got stuck. Both have played a crucial role in this work and in my (short) career as a researcher. No words can express my gratitude for that.

Second, I wish to thank my lab colleagues. To Luis and Luisi, who welcomed me into the research group with open arms. To Juanlu, for fruitful discussions on machine learning and for his infinite patience. To Joaquín, for making stressful days better with comforting hugs. To Nuria, Candela and Mara, for their generosity and for always considering my advice. Sharing the experience of pursuing a PhD with you has been a pleasure, guys.

I also want to thank those who made my research stay in London a life-enriching experience. To Toktam Mahmoodi, for her hospitality. She is an excellent referent of a woman leading a big team in the telecom world while devoting time to her students. To Omar and Wen, for amazing conversation. And to Manolín, a stranger that became a good friend in three months. Two different souls can definitely make a great team!

The financial support given by the FPU17/04286 grant from the Spanish Ministry of Education, Culture and Sport, research projects mentioned later in this document, the University of Málaga and Ericsson Spain is kindly acknowledged, since it has allowed

me to disseminate my research in journals, conferences and workshops.

To my friends: the teachers, the economist, the physiotherapist, the chemist, the psychologist, the musician and the engineers. What a great second family I chose! Their predisposition to reach their goals inspires me to pursue mine. To those of you who will become PhD soon, good luck!

Finally, I wish to thank my entire family. To my parents and brother, who celebrate my achievements as theirs. The person I am today is the result of all the love and unconditional support they have given me every day of my life. To Keko, for his love, admiration and ability to make me laugh even under stress, and for his understanding and patience these last months. And to the rest of my family, for being so proud of me (dear uncle, I hope you are happy to read these words wherever you are!). They are the most precious of my life, and to them I dedicate this thesis.

# Contents

<b>Abstract</b>	vii
<b>Resumen</b>	ix
<b>List of acronyms</b>	xi
<b>List of figures</b>	xvii
<b>List of tables</b>	xxi
<b>List of algorithms</b>	xxiii
<b>1 Introduction</b>	1
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	5
1.3 Research methodology . . . . .	7
1.4 Document structure . . . . .	9
<b>2 Technical background</b>	11
2.1 Machine learning . . . . .	11
2.1.1 Supervised learning . . . . .	11
2.1.2 Unsupervised learning . . . . .	17
2.1.3 Reinforcement learning . . . . .	19
2.2 Zero-touch networks . . . . .	20

2.2.1	SON use cases	20
2.2.2	Big-data-empowered SON	26
2.3	Network slicing	28
<b>3</b>	<b>Classification of encrypted traffic in cellular networks</b>	<b>31</b>
3.1	Related work	31
3.2	Problem formulation	33
3.2.1	Traffic descriptors from connection traces	33
3.2.2	Data encapsulation process	35
3.3	Classification method	38
3.4	Performance assessment	44
3.4.1	Assessment methodology	44
3.4.2	Results	45
3.4.3	Computational complexity	51
3.5	Conclusions	52
<b>4</b>	<b>Throughput estimation in cellular radio access networks</b>	<b>55</b>
4.1	Related work	55
4.2	Problem formulation	59
4.3	Throughput estimation method	61
4.3.1	Data collection	62
4.3.2	Data preprocessing	63
4.3.3	Model creation	63
4.3.4	Performance evaluation	66
4.4	Cell and user capacity estimation in HSDPA and LTE	66
4.4.1	Dataset description	66
4.4.2	Performance assessment	71
4.4.3	Conclusions	79
4.5	Cell and slice throughput estimation in sliced radio access networks	80

4.5.1	Dataset description	81
4.5.2	Performance assessment	84
4.5.3	Conclusions	99
<b>5</b>	<b>Long-term cell traffic forecasting</b>	<b>101</b>
5.1	Related work	101
5.2	Problem formulation	104
5.3	Forecasting method	106
5.4	Performance assessment	110
5.4.1	Dataset description	110
5.4.2	Assessment methodology	111
5.4.3	Results	115
5.4.4	Computational complexity	121
5.5	Conclusions	123
<b>6</b>	<b>Traffic steering in cellular networks</b>	<b>125</b>
6.1	Related work	125
6.2	QoE-driven traffic steering in multi-tier LTE networks	128
6.2.1	Problem formulation	129
6.2.2	Traffic steering strategy	131
6.2.3	Performance assessment	137
6.2.4	Conclusions	144
6.3	SLA-driven traffic steering in sliced radio access networks	144
6.3.1	Problem formulation	145
6.3.2	Traffic steering strategy	146
6.3.3	Performance assessment	151
6.3.4	Conclusions	160
<b>7</b>	<b>Conclusions</b>	<b>163</b>

7.1	Main contributions	163
7.1.1	Classification of encrypted traffic in cellular networks	165
7.1.2	Supervised learning for radio access network (re)dimensioning	166
7.1.3	Traffic steering in cellular networks	168
7.1.4	Discussion on model implementation	170
7.2	Future work	170
7.3	List of contributions	173
<b>Appendix A Simulation tool</b>		<b>179</b>
A.1	General structure	179
A.1.1	Work flow	179
A.1.2	Simulation scenarios	181
A.1.3	UE model	182
A.1.4	QoE model	182
A.2	Physical layer	185
A.2.1	Propagation model	185
A.2.2	Noise model	187
A.2.3	Interference model	188
A.3	Link layer	188
A.3.1	Link adaptation	188
A.3.2	Packet scheduling	189
A.3.3	Retransmission scheme	189
A.4	Network layer	189
A.4.1	Admission control	190
A.4.2	Handover scheme	190
A.5	Network slicing implementation	191
<b>Appendix B Summary (Spanish)</b>		<b>193</b>
B.1	Antecedentes y motivación	193

<b>B.2</b> Objetivos . . . . .	197
<b>B.3</b> Metodología de trabajo . . . . .	198
<b>B.4</b> Desarrollo de la investigación . . . . .	200
<b>B.4.1</b> Clasificación de tráfico encriptado en redes celulares . . . . .	200
<b>B.4.2</b> Estimación de rendimiento en redes de acceso radio celulares . . . . .	201
<b>B.4.3</b> Predicción de tráfico de celda a largo plazo . . . . .	204
<b>B.4.4</b> Reparto de tráfico en redes celulares . . . . .	205
<b>B.5</b> Conclusiones . . . . .	207
<b>B.6</b> Lista de contribuciones . . . . .	211
<b>References</b> . . . . .	217



UNIVERSIDAD  
DE MÁLAGA



# Abstract

In current years, cellular networks are experiencing profound changes to cope with the increasing demand for ever-diverse and ever-demanding services. As a result, the size and complexity of these networks has increased dramatically, evincing the need for zero-touch network and service management solutions. In the Radio Access Network (RAN), operators have already tackled the automation of management procedures in the past, giving rise to Self-Organizing Networks (SON). However, classical SON solutions are expected to be ineffective in next-generation networks offering services with extremely stringent and varying performance requirements. With the latest advances in information technology, it is now possible to leverage massive data collected in the Operations Support System (OSS) to develop advanced data-driven SON tools able to capture the peculiarities of each particular network. These new SON solutions must consider new features arising in 5G. One of these features is network slicing, allowing the coexistence of several separate logical networks operating simultaneously over the same physical infrastructure.

This thesis tackles the creation of data-driven self-management solutions for the RAN. Among existing SON use cases, the scope of this work focuses on two particular well-known self-planning and self-optimization use cases, namely RAN redimensioning and mobility load balancing. In both cases, solutions are proposed for legacy RANs, where all users share resources, and for new sliced RANs arising in 5G.

Regarding RAN redimensioning, this work explores the use of supervised learning over network data to derive performance models to detect potential capacity bottlenecks with radio planning tools. Models have been built for two purposes: estimating radio throughput metrics per cell/slice in different radio access technologies and forecasting cell traffic in the long term (i.e., months horizon).

Moreover, this thesis proposes two data-driven service-oriented mobility load balancing algorithms through handover parameter tuning. The main goal is to relieve

local congestion issues by sharing traffic with neighbor cells. In both proposals, traffic steering has been formulated as a control problem. The first algorithm deals with traffic steering among cells in different carriers with quality of experience criteria, whereas the second algorithm tackles slice-aware traffic steering to guarantee service level agreement compliance in new 5G sliced RANs.

It should be pointed out that service-oriented self-management solutions proposed in this thesis require prior knowledge of the application demanded per user. However, obtaining such information nowadays is not straightforward for operators due to traffic encryption. The task of classifying encrypted traffic per service type is also addressed in this work. Such a problem has been tackled through unsupervised learning over connection traces, circumventing the need for a labeled trace dataset or the installation of expensive probes in the core network.

All the solutions proposed in this thesis rely on data currently available in the OSS, thus requiring no change in network infrastructure. To support the significance of results, performance assessment is always carried out in a realistic environment, i.e., with data from commercial cellular networks when possible or with a simulation tool calibrated with configuration and performance data from live networks otherwise.

# Resumen

En la actualidad, las redes de comunicaciones móviles están experimentando cambios sustanciales para hacer frente a la creciente demanda de servicios móviles cada vez más diversos y exigentes. Como resultado, el tamaño y la complejidad de estas redes ha crecido dramáticamente, evidenciando la necesidad de soluciones de gestión de redes y servicios sin intervención humana. En la red de acceso radio, los operadores ya han abordado la automatización de los procedimientos de gestión en el pasado, dando lugar a las redes autoorganizadas. Sin embargo, es esperable que las soluciones clásicas no sean efectivas en las redes de nueva generación que ofrecen servicios con requisitos de rendimiento extremadamente exigentes y diversos. Con los últimos avances en tecnologías de la información, se puede aprovechar la ingente cantidad de datos que se recopila en el sistema de soporte a las operaciones de la red para desarrollar herramientas de gestión automática avanzadas basadas en datos, capaces de capturar las peculiaridades de cada red particular. Estas nuevas soluciones de gestión automática deben tener en cuenta las nuevas funcionalidades que surgen en 5G. Una de ellas es la segmentación de red, que permite la coexistencia de varias redes lógicas operando simultáneamente sobre la misma infraestructura física.

Esta tesis aborda la creación de herramientas basadas en el uso intensivo de datos para la gestión automática de redes de acceso radio. Entre los casos de uso de autogestión de redes celulares que existen, el alcance de este trabajo se centra en dos casos de uso de autoplanificación y autooptimización muy extendidos: el redimensionado de la red de acceso radio y el balance de tráfico por movilidad. En ambos casos, se proponen soluciones para las redes radio clásicas, en las que los recursos se comparten por todos los usuarios, y para las nuevas redes de acceso radio segmentadas que aparecen en 5G.

Para el redimensionado de la red radio, este trabajo explora el de modelos de aprendizaje supervisado para detectar potenciales cuellos de botella de capacidad con

herramientas de planificación radio. Se han creado modelos con dos objetivos: a) estimar diversas métricas de caudal (*throughput*) a nivel de celda y segmento en distintas tecnologías de acceso radio y b) predecir el tráfico de celda a largo plazo (es decir, con un horizonte en una escala temporal de meses).

En paralelo, esta tesis propone dos algoritmos basados en datos para el balance de tráfico por movilidad orientado al servicio mediante el ajuste de parámetros de traspaso. El objetivo es aliviar problemas de congestión locales a través del reparto de tráfico entre celdas vecinas. En ambas propuestas, el reparto de tráfico se ha formulado como un problema de control. El primer algoritmo distribuye el tráfico entre celdas que funcionan en distintas frecuencias portadoras con criterios de calidad de experiencia, mientras que el segundo algoritmo aborda la tarea de repartir el tráfico en las nuevas redes de acceso radio 5G segmentadas con el objetivo de garantizar el cumplimiento de los acuerdos de servicio.

Cabe destacar que las soluciones de gestión automática orientadas al servicio propuestas en esta tesis requieren conocer a priori el tipo de aplicación demandada por cada usuario. Sin embargo, en la actualidad esta información no es fácil de obtener por los operadores debido al encriptado del tráfico. La clasificación de tráfico encriptado por tipo de servicio también se aborda en esta tesis. Este problema se ha afrontado con el uso de aprendizaje no supervisado sobre trazas de conexión radio, que no requiere un juego de datos etiquetado ni la instalación de caras sondas de tráfico en el núcleo de la red.

Todos los métodos propuestos en esta tesis se basan en información actualmente almacenada en el sistema de soporte a operaciones, y por tanto no requieren cambios en la infraestructura de la red. Para avalar la importancia de los resultados, la evaluación del rendimiento siempre se lleva a cabo en un entorno realista, es decir, con datos de redes móviles comerciales cuando es posible o con una herramienta de simulación calibrada con datos de configuración y rendimiento de redes reales en caso contrario.

# List of acronyms

<b>3GPP</b>	3 <sup>rd</sup> Generation Partnership Project
<b>5QI</b>	5G Quality-of-Service Identifier
<b>ACK</b>	ACKnowledgement
<b>AdaBoost</b>	Adaptive Boosting
<b>Adam</b>	Adaptive moment estimation
<b>AHC</b>	Agglomerative Hierarchical Clustering
<b>AHW</b>	Additive Holt-Winters
<b>AI</b>	Artificial Intelligence
<b>AMR CS</b>	Adaptive Multi-Rate Circuit-Switched
<b>ANN</b>	Artificial Neural Network
<b>ARIMA</b>	Auto Regressive Integrated Moving Average
<b>BDA</b>	Big Data Analytics
<b>BLER</b>	BLock Error Rate
<b>CDF</b>	Cumulative Distribution Function
<b>CH</b>	Calinski–Harabasz
<b>CM</b>	Configuration Management
<b>CQI</b>	Channel Quality Indicator
<b>CTR</b>	Cell Traffic Recording

<b>DBSCAN</b>	Density-Based Spatial Clustering of Applications with Noise
<b>DCH</b>	Data CHannel
<b>DL</b>	Down Link
<b>DNN</b>	Deep Neural Network
<b>DRL</b>	Deep Reinforcement Learning
<b>DT</b>	Decision Tree
<b>E2E</b>	End-to-End
<b>eMBB</b>	enhanced Mobile BroadBand
<b>eNB</b>	evolved Node B
<b>ETU</b>	Extended Typical Urban
<b>EXP/PF</b>	EXPonential/Proportional Fair
<b>FE</b>	Feature Extraction
<b>FLC</b>	Fuzzy Logic Controller
<b>FoM</b>	Figure of Merit
<b>FS</b>	Feature Selection
<b>FTP</b>	File Transfer Protocol
<b>GARCH</b>	Generalized Auto-Regressive Conditionally Heteroskedastic
<b>gNB</b>	gNodeB
<b>HARQ</b>	Hybrid Automatic ReQuest
<b>HO</b>	HandOver
<b>HOM</b>	HandOver Margin
<b>HSDPA</b>	High-Speed Downlink Packet Access
<b>HTTP</b>	HyperText Transfer Protocol
<b>HW</b>	HardWare

---

<b>IP</b>	Internet Protocol
<b>KNN</b>	K-Nearest Neighbors
<b>KPI</b>	Key Performance Indicator
<b>KQI</b>	Key Quality Indicator
<b>L-BFGS</b>	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
<b>LSTM</b>	Long Short-Term Memory
<b>LTE</b>	Long-Term Evolution
<b>MAC</b>	Medium Access Control
<b>MANO</b>	Management ANd Orchestration
<b>MCS</b>	Modulation and Coding Scheme
<b>MDT</b>	Minimization of Drive Test
<b>MIMO</b>	Multiple-Input Multiple-Output
<b>ML</b>	Machine Learning
<b>MLB</b>	Mobility Load Balancing
<b>MLP</b>	Multi-Layer Perceptron
<b>MLR</b>	Multi-variable Linear Regression
<b>mMTC</b>	massive Machine Type Communications
<b>MNO</b>	Mobile Network Operator
<b>MOS</b>	Mean Opinion Score
<b>MSS</b>	Maximum Segment Size
<b>NE</b>	Network Element
<b>NF</b>	Network Function
<b>NGN</b>	Next-Generation Network
<b>NR</b>	New Radio

<b>NRT</b>	Non-Real Time
<b>NS</b>	Network Slicing
<b>OSS</b>	Operations Support System
<b>OTT</b>	Over The Top
<b>PBGT</b>	Power-BudGeT HandOver
<b>PDCP</b>	Packet Data Convergence Protocol
<b>PDSCH</b>	Physical Downlink Shared CHannel
<b>PF</b>	Proportional Fair
<b>PM</b>	Performance Management
<b>PRB</b>	Physical Resource Block
<b>QBGT</b>	Quality-BudGeT HandOver
<b>QCI</b>	Quality-of-service Class Identifier
<b>QoE</b>	Quality of Experience
<b>QoS</b>	Quality of Service
<b>RAN</b>	Radio Acces Network
<b>RAT</b>	Radio Access Technology
<b>RF</b>	Random Forest
<b>RFE</b>	Recursive Feature Elimination
<b>RL</b>	Reinforcement Learning
<b>RLC</b>	Radio Link Control
<b>ROP</b>	Reporting Output Period
<b>RRC</b>	Radio Resource Control
<b>RRM</b>	Radio Resource Management
<b>RSRP</b>	Reference Signal Received Power



<b>RSRQ</b>	Reference Signal Received Quality
<b>RSSI</b>	Received Signal Strength Indicator
<b>RT</b>	Real Time
<b>SARIMA</b>	Seasonal Auto Regressive Integrated Moving Average
<b>SFS</b>	Sequential Forward Selection
<b>SINR</b>	Signal-to-Interference-plus-Noise Ratio
<b>SL</b>	Supervised Learning
<b>SLA</b>	Service Level Agreement
<b>SON</b>	Self-Organized Networks
<b>SVC</b>	Support Vector Classifier
<b>SVR</b>	Support Vector Regression
<b>SW</b>	SoftWare
<b>TCP</b>	Transmission Control Protocol
<b>TI</b>	Tuning Interval
<b>TSA</b>	Time Series Analysis
<b>TTI</b>	Transmission Time Interval
<b>TTT</b>	Time To Trigger
<b>UDP</b>	User Datagram Protocol
<b>UE</b>	User Equipment
<b>UETR</b>	User Equipment Traffic Recording
<b>UL</b>	UpLink
<b>uRLLC</b>	ultra-Reliable Low Latency Communications
<b>USL</b>	UnSupervised Learning
<b>VNF</b>	Virtualized Network Function

<b>VoIP</b>	Voice over Internet Protocol
<b>VoLTE</b>	Voice over Long Term Evolution
<b>XGBoost</b>	eXtreme Gradient Boosting
<b>ZSM</b>	Zero-touch Network and Service Management

# List of figures

1.1	Data-driven network design and optimization system.	7
2.1	Taxonomy of machine learning algorithms.	12
2.2	Elements of support vector regression algorithm.	13
2.3	Example of decision tree.	14
2.4	Neural network elements.	16
2.5	Reinforcement learning agent.	19
2.6	Flow diagram of RAN redimensioning in radio planning tools.	22
2.7	Example of traffic steering by tuning handover margins.	25
2.8	Network slicing.	29
3.1	Example of packet encapsulation in the LTE user plane.	36
3.2	Traffic classification method.	39
3.3	Burst-level connection model in the radio interface considering last TTIs.	40
3.4	B-AHC performance with different number of clusters.	46
3.5	E-AHC performance with different number of clusters.	48
4.1	Throughput estimation method.	62
4.2	Cumulative distribution function of network indicators in HSDPA and LTE datasets.	70
4.3	MAPE evolution across sequential feature selection (FS-SFS) process when estimating DL cell throughput in HSDPA.	74

4.4	<i>MAPE</i> evolution across sequential feature selection (FS-SFS) process when estimating DL user throughput in HSDPA.	75
4.5	<i>MAPE</i> evolution across sequential feature selection (FS-SFS) process when estimating DL cell throughput in LTE.	77
4.6	<i>MAPE</i> evolution across sequential feature selection (FS-SFS) process when estimating DL user throughput in LTE.	77
4.7	Cumulative distribution function of number of active UEs across simulations – NS_SS scenario.	82
4.8	<i>MANE</i> evolution across RFE process when estimating DL cell throughput in single service NS scenario (NS_SS).	91
4.9	<i>MANE</i> evolution across RFE process when estimating DL cell throughput in multi-service NS scenario (NS_MS).	91
4.10	Distribution of absolute normalized error for best models when estimating DL cell throughput in single-service NS scenario (NS_SS).	92
4.11	<i>MANE</i> evolution across RFE process when estimating DL slice throughput in single-service scenario (NS_SS).	95
4.12	<i>MANE</i> evolution across RFE process when estimating DL slice throughput in multi-service NS scenario (NS_MS).	95
4.13	Distribution of absolute normalized error for best models when estimating DL slice throughput in multi-service NS scenario (NS_MS).	96
4.14	Values of features in samples with the largest error when estimating DL slice throughput with RFE–RF (NS_SS scenario).	97
4.15	Values of features in samples with the largest error when estimating DL slice throughput with RFE–SMLP (NS_MS scenario).	97
5.1	Evolution of monthly busy-hour traffic in a cell.	104
5.2	Autocorrelation of DL cell traffic.	106
5.3	Timeline of prediction algorithms in a generic case.	109
5.4	Timeline of prediction algorithms in cases 24–3 and 12–3 when forecasting cellular traffic in June 2017.	114
5.5	<i>MAE</i> evolution across different target months.	117

5.6	Example of the impact of replanning actions on cell traffic.	118
5.7	Prediction error vs. monthly busy-hour traffic (ANN-LSTM, case 12-3).	119
5.8	Error cumulative distribution functions for SL algorithms when forecasting traffic in high-traffic cells (case 12-3).	121
6.1	Typical handover scheme in a two-tier network.	130
6.2	Impact of cell load on RSRQ.	131
6.3	Quality-based handover scheme for a two-tier network.	132
6.4	Evolution of the overall QoE in the scenario.	140
6.5	Cumulative distribution function of user QoE for different services.	142
6.6	Evolution of absolute handover margin deviation from default values in tuned adjacencies per slice.	155
6.7	Evolution of the overall SLA compliance in the scenario.	156
6.8	Cumulative distribution of final SLA compliance per cell for slice 1 (FTP + LIVE VIDEO).	158
6.9	Cumulative distribution of final handover margin deviation from initial setting in tuned adjacencies for slice 3 (DRIVING).	159
A.1	Simulator workflow.	180
A.2	Scenarios implemented in the simulation tool.	183



UNIVERSIDAD  
DE MÁLAGA

# List of tables

2.1 Intra-frequency handover events. . . . .	25
3.1 Traffic descriptors at different protocol layers for different services in LTE. . . . .	37
3.2 Events in connection traces used for traffic classification. . . . .	45
3.3 Groups in B-AHC method. . . . .	46
3.4 Groups in E-AHC method. . . . .	48
3.5 Share of DL traffic volume. . . . .	51
4.1 Hyperparameters tuned for throughput estimation. . . . .	65
4.2 Candidate input features for estimating DL cell throughput in HSDPA. . . . .	68
4.3 Statistics of dataset A (HSDPA network). . . . .	68
4.4 Candidate input features for estimating DL cell throughput in LTE. . . . .	69
4.5 Statistics of dataset B (LTE scenario). . . . .	70
4.6 MAPE for estimating DL cell and user throughput in HSDPA [%]. . . . .	73
4.7 MAPE for estimating DL cell and user throughput in LTE [%]. . . . .	76
4.8 Training times for throughput estimation models in HSDPA and LTE [s]. . . . .	79
4.9 Candidate features for estimating DL cell throughput in sliced networks. . . . .	83
4.10 Candidate features for estimating DL slice throughput in sliced networks. . . . .	83
4.11 Statistics of cell-level datasets used to estimate DL cell throughput. . . . .	85
4.12 Statistics of slice-level datasets used to estimate DL slice throughput. . . . .	86
4.13 Correlation between candidate input features and DL cell throughput in different NS scenarios. . . . .	89

4.14 Model performance for estimating DL cell throughput in single-service	
NS scenario (NS_SS).	90
4.15 Model performance for estimating DL cell throughput in multi-service	
NS scenario (NS_MS).	90
4.16 Model performance for estimating DL slice throughput in single-service	
NS scenario (NS_SS).	94
4.17 Model performance for estimating DL slice throughput in multi-service	
NS scenario (NS_MS).	94
4.18 Performance per slice when estimating DL slice throughput with the	
best model.	96
4.19 Time complexity of FULL models when estimating DL slice throughput	
in single-service NS scenario.	99
5.1 Impact of data collection window for 3-month forecasting horizon.	116
5.2 Impact of data collection window for 6-month forecasting horizon.	116
5.3 Average performance of forecasting algorithms across different target	
months.	117
5.4 Performance of network-wide and specific forecasting models for high-	
traffic cells (case 12-3).	120
5.5 Execution times for forecasting models [s].	122
6.1 Initial performance of simulated two-tier LTE network.	140
6.2 Performance comparison of MLB strategies in a two-tier LTE network	
– case A (indoor/outdoor).	141
6.3 Performance comparison of MLB strategies in a two-tier LTE network	
– case B (outdoor).	143
6.4 Simulation set-up for assessing MLB strategies in a NS scenario.	152
6.5 Performance comparison of MLB strategies in a NS scenario.	157
A.1 Main simulation parameters.	181
A.2 Service model parameters.	184
A.3 ETU model [1].	187



# List of Algorithms

1	QoE-driven self-tuning algorithm.	136
2	Adjacency clustering algorithm.	147
3	SLA-driven slice-aware self-tuning algorithm.	150



UNIVERSIDAD  
DE MÁLAGA

# Chapter 1

## Introduction

In this opening chapter, the motivation for this thesis is first explained, objectives are then presented, the research methodology is next detailed and the structure of this document is finally broken down.

### 1.1 Motivation

Mobile communication networks have experienced several changes over the last few years. First, the explosive growth in wireless data traffic has forced Mobile Network Operators (MNOs) to increase network capacity. For this purpose, in the Radio Access Network (RAN), classical one-tier macro-cell networks are evolving towards multi-carrier systems with several operation bands and heterogeneous networks combining small cells and macro cells [2]. Moreover, the highly varying performance requirements and increasing user expectations have led to a change in network management procedures from a network-centric paradigm based on network performance to a user-centric approach focused on customer satisfaction (a.k.a. Quality of Experience, QoE) [3]. In parallel, the advent of 5G is expanding the business model of MNOs, who will provide vertical industries with enhanced Mobile BroadBand (eMBB), ultra-Reliable Low Latency Communications (uRLLC) and massive Machine Type Communications (mMTC) services, boosting a fully-connected world [4]. The 3<sup>rd</sup> Generation Partnership Project (3GPP) launched New Radio (NR) specifications for standalone and non-standalone 5G networks from release 15 onward. In the RAN, new frequency ranges (e.g., millimeter waves) and features (e.g., multi-connectivity, beamforming, massive Multiple-Input Multiple-Output – MIMO– schemes...) are introduced

to achieve ambitious 5G performance goals [5].

As a result of the above changes, the size and complexity of mobile networks have increased dramatically, evincing the need for network management tools with minimal human intervention to guarantee an efficient network operation. In the literature, plenty of self-planning, self-deployment, self-optimization and self-healing strategies have been proposed for the RAN in 2G (e.g., [6] [7] [8]), 3G (e.g., [9] [10] [11]) and 4G (e.g., [12] [13] [14]) networks. Classical solutions for Self-Organized Networks (SON) rely on analytical models and/or heuristic controllers designed according to the knowledge of experts. However, this approach is envisioned to be insufficient and ineffective in 5G and beyond systems (a.k.a. Next-Generation Networks, NGNs). for several reasons. First, the coexistence of services with strongly different performance requirements (e.g., energy efficiency, latency, reliability, data rate...) calls for a service-oriented self-management paradigm. Second, SON tools based on preset and fixed controllers will not make the most of network capabilities in NGNs with different service mix (e.g., smart city vs. industry 4.0), network topology (e.g., heterogeneous vs. macro-cell networks) and configuration (e.g., single-connectivity vs. multi-connectivity). Likewise, new 5G features such as network virtualization, cloudification, edge computing or End-to-End (E2E) network slicing must be considered when managing the network [5].

Network Slicing (NS) is a particularly remarkable 5G functionality due to its strong impact on network operation and performance. NS allows building separate logical networks tailored for specific purposes on top of a common physical infrastructure [15]. From a network management perspective, new Network Functions (NFs) arise in NS scenarios (e.g., capacity brokers), whose parameters can be self-configured and self-optimized. Moreover, several aspects must be considered when designing slice-aware self-management solution, such as: a) the split of network resources among slices, b) slice activation, deactivation or redimensioning of assigned resources, c) the possibility to tailor or even omit NFs per slice, and d) privacy issues that may prevent the central manager and orchestrator from accessing slice-level data managed by the slice tenant [16]. Additionally, note that a specific self-management solution may perform differently in distinct NS scenarios (e.g., multi-service slices leased by virtual MNOs vs. single-service slices for verticals).

To overcome the limitations of legacy SON schemes, with the latest advances in Big Data Analytics (BDA) and Artificial Intelligence (AI), it is now possible to develop fully automated data-driven SON tools leveraging massive data (e.g., alarms, connection traces, performance counters...) collected in the Operations Support System (OSS),

giving rise to a Zero-touch Network and Service Management (ZSM) paradigm [17]. Cutting-edge data-driven solutions for ZSM networks rely on machine learning techniques, able to capture the peculiarities of each particular network (e.g., type of scenario, network topology, radio resource management algorithms, service mix, NS set-up...) [18] [19]. The combination of NS and ZSM, creating logical networks managed without human intervention, has been recognized as the most efficient method to exploit network assets while guaranteeing customer satisfaction in NGNs [20].

A large number of SON use cases can be empowered with network data. References [17], [18] and [20] survey previous contributions proposing data-driven SON tools. This thesis focuses on two well-known self-planning and self-optimization use cases, namely RAN redimensioning and Mobility Load Balancing (MLB), for which some research gaps are still pending (e.g., the design of slice-aware solutions). Although these procedures are independent, their joint study is well-suited since, as explained later in this chapter, data-driven RAN redimensioning and MLB solutions can ease the implementation of a fully automated network design and optimization system.

RAN redimensioning is a critical task for MNOs to prevent capacity bottlenecks caused by changing mobile traffic patterns while avoiding unnecessary upgrades of network resources. To detect potential problems in advance, proactive network planning tools compare busy-hour traffic forecasts with estimates of network capacity, often measured as aggregated cell throughput. Traffic forecasting is the task of predicting expected traffic from historical data, which can be treated as a time series analysis problem. The use of supervised learning over historical network data for short-term (i.e., second- or minute-scale horizon) and mid-term (i.e., day-scale horizon) traffic forecasting in mobile networks has been extensively covered in the literature [21]. However, redimensioning actions can take up to several months (e.g., new cell deployment). It is still to be checked if supervised learning algorithms outperform classical time series analysis for long-term cellular traffic forecasting, which, as will be shown later, relies on noisy and short time series.

In contrast to traffic forecasting, performance estimation aims to predict Key Performance Indicators (KPIs) at a given time from other information about network state at the same time. Data-driven models have been proposed for estimating aggregated cell throughput in High-Speed Down link Packet Access (HSDPA) and Long-Term Evolution (LTE) networks. Most recent works rely on supervised learning, namely multi-variable linear regression [22] [23] [24] or complex deep neural networks [25], over data collected in the OSS. However, the performance of other non-linear models (e.g.,

ensemble models based on decision trees) less prone to overfitting than deep neural networks has not been checked in these Radio Access Technologies (RATs). Likewise, other throughput metrics with a higher impact on QoE (e.g., user throughput) should also be considered in the redimensioning process. Additionally, note that the correlation between network state and cell-level performance metrics may change when enabling NS. Thus, a separate analysis is required for sliced RANs. In these networks, NFs such as capacity brokers also require slice-level throughput estimates that complement cell-level and user-level metrics.

Some redimensioning actions cannot be implemented immediately. In the meantime, a cost-effective way of relieving capacity bottlenecks is sharing traffic demand between adjacent cells. Load balancing is a self-optimization use case that automatically offloads users between cells to deal with uneven traffic demand in a cellular network<sup>1</sup>. It ensures that every user is constantly served by the cell offering the best performance, thus strongly impacting user experience. Traffic steering can be addressed by adjusting antenna parameters such as transmit power [26] [27]. However, this approach may create coverage holes. Alternatively, most works tackle load balancing by optimizing mobility NFs (a.k.a. MLB), driven by logical parameters (i.e., timers, power offsets...) that can be cost-effectively and immediately tuned. Some authors opt for optimizing cell reselection parameters [28] [29]. However, the preferred option is tuning HandOver (HO) margins, since HO has a larger impact on network performance [30] [31] [32]. QoE-driven MLB algorithms have only been proposed to handle traffic in one-tier macro-cell LTE scenarios [33] [34]. To make the most of capacity in current multi-tier cellular networks, such an approach must be extended for self-optimizing inter-frequency traffic steering. Additionally, slice-aware MLB algorithms must be designed to guarantee Service Level Agreement (SLA) compliance in complex and highly dynamic sliced RANs [35].

It should be pointed out that service-oriented self-optimization algorithms, providing a customized handling per connection, assume prior knowledge of the type of application (e.g., voice call, media streaming, instant messaging...) demanded per user. Moreover, awareness of service mix can enhance performance models in radio planning tools. However, classifying connections per service class in cellular networks is a challenging task due to: a) traffic encryption, preventing MNOs from using deep packet inspection techniques [36], b) the reluctance of MNOs to install expensive probes to

---

<sup>1</sup>In this thesis, load balancing refers to any traffic steering strategy, even if the aim is not explicitly balancing cell load.

capture traffic flows in the core network, and c) the absence of labeled data required to train classifiers based on supervised learning. All these issues can be circumvented by performing classification with unsupervised learning over traffic descriptors derived from radio connection traces. Therefore, trace-based traffic classifiers are precious assets for MNOs.

## 1.2 Objectives

The aim of this thesis is to develop automatic data-driven procedures for the above-mentioned self-planning and self-optimization use cases that can be implemented in network management tools. Specifically, the goal of this thesis is threefold:

- O1.** Design a strategy for classifying encrypted traffic per service class in the radio interface, allowing a customized handling per connection in MLB algorithms and providing service mix information for performance estimation models.
- O2.** Explore the use of supervised learning over data collected in the OSS to enhance the performance of radio planning tools. To this end, three tasks are addressed:
  - O2.1.** Forecast monthly busy-hour cell traffic in the long term (i.e., months horizon) from short and noisy time series.
  - O2.2.** Estimate radio throughput metrics reflecting cell and user capacity in LTE and HSDPA networks.
  - O2.3.** Estimate radio throughput metrics reflecting cell and slice performance in sliced RANs.
- O3.** Develop data-driven service-oriented MLB algorithms for scenarios where such an approach has not been considered yet. Specifically, two use cases are covered:
  - O3.1.** Inter-frequency MLB for multi-tier LTE networks with QoE criteria.
  - O3.2.** Slice-aware MLB for 5G sliced RANs with SLA criteria.

The main contributions of this thesis are:

- 1) A system for encrypted traffic classification per service class in the RAN relying on unsupervised clustering over traffic descriptors computed from information in connection traces, including some novel features modeling connections at burst level. The unsupervised approach ensures that the method works in the absence

of network probes and labeled data, allows to identify new types of applications arising in the network, and can be easily adapted to different RATs.

- 2) A methodology for cell traffic forecasting in the long term based on supervised learning that outperforms classical time series analysis. This task has not been covered until now due to the scarcity of datasets comprising traffic measurements collected for years in live networks. However, results presented here evince the need for further research in this direction.
- 3) A methodology for estimating radio throughput indicators from data in the OSS based on supervised learning, which captures network peculiarities. The methodology is extended to estimate different throughput indicators (average user throughput and aggregated throughput) defined at different levels (cell and cell-slice) in different RATs (LTE and HSDPA) and scenarios (legacy and sliced RANs). The set of candidate predictors includes features derived from different data sources (cell counters and connection traces).
- 4) Two algorithms for service-oriented MLB by tuning HO margins driven by network data. The first algorithm steers traffic among carriers by tuning inter-frequency HO margins to improve user QoE in multi-service LTE networks. The second algorithm performs slice-aware MLB among intra-frequency neighbor cells to guarantee SLA compliance in all cells and slices. Both proposals rely on a proportional controller driven by indicators computed from connection traces.
- 5) A dynamic system-level simulator updated to emulate the activity of a realistic sliced RAN serving users demanding different uRLLC and eMBB services. In the absence of commercial networks with network slicing, this tool is a precious asset for validating slice-aware SON solutions.

All these contributions can be jointly used in a data-driven E2E network design and optimization system such as that illustrated in Fig. [1.1](#). The system consists of two automatic modules for proactive problem detection and problem avoidance, respectively, in the RAN. These modules are run recursively on a loop fed by network data. In the problem detection module, cell/slice throughput estimation and long-term cell traffic forecasting models allow the proactive detection of potential capacity bottlenecks on a cell or slice level. Then, in the problem avoidance module, service-oriented MLB algorithms can temporarily solve imminent foreseen problems until the normal network state is recovered or a more stable solution (e.g., capacity extension) is taken. In the latter case, cell/slice throughput estimation models can be used to measure the



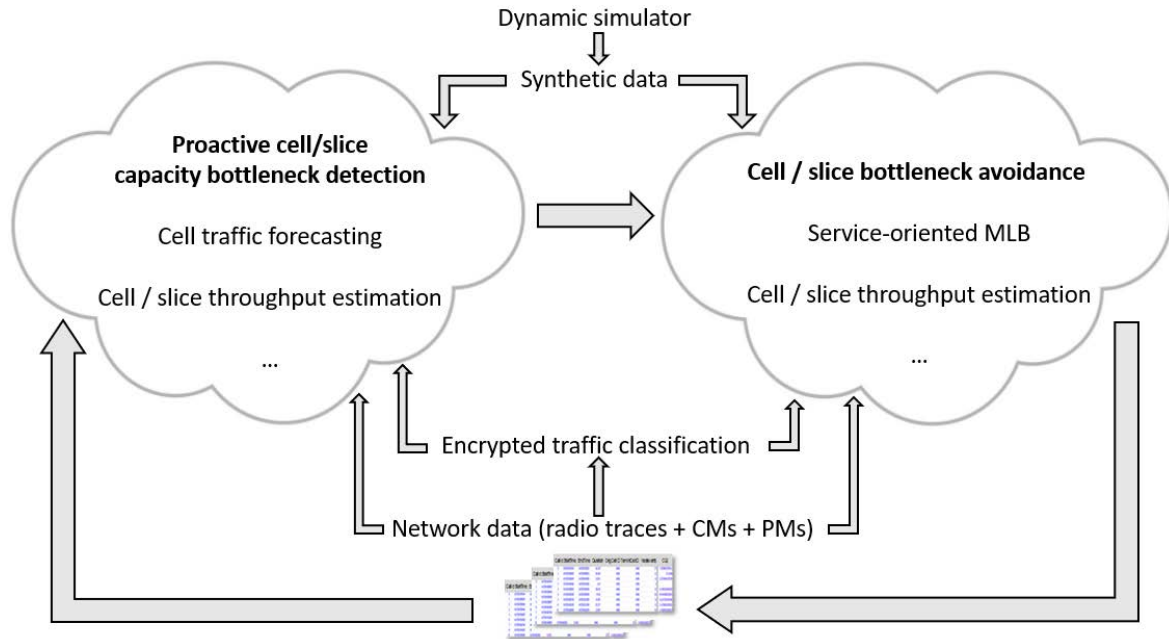


Figure 1.1: Data-driven network design and optimization system.

impact of candidate actions on network performance. The trace-based traffic classifier plays a crucial role in both modules, enabling the operation of MLB algorithms and providing input features for throughput estimation models. Similarly, the simulator enhanced as part of this thesis can be used as a digital twin to create realistic synthetic datasets or validate new SON tools while keeping the live network working in a safe operating area.

It should be pointed out that MNO constraints, often neglected in research work, have been taken into account in this thesis. All the proposed models and algorithms are centralized solutions conceived to make the most of connection traces and counters currently collected by MNOs in the OSS. Moreover, when using supervised learning, complex models based on deep neural networks have been avoided to consider the possibility of having limited datasets and the reluctance of operators to increase computational complexity in their network management tools. Finally, evaluation has been carried out over data from live networks whenever possible, and with realistic simulations otherwise.

### 1.3 Research methodology

Steps followed for the attainment of the defined objectives are described next, breaking down the specificities of each objective:

- a) *Problem selection and literature review.* First, the set of SON use cases to be tackled was identified. Then, the state of research in all the fields within the scope of this thesis was thoroughly revised. The main topics covered were: a) SON, to identify shortcomings in current RAN redimensioning and MLB solutions, b) machine learning, to get insight on data preprocessing and inference algorithms, and c) 5G technology and NS feature, to fully understand the concept and operation of a sliced RAN and hence guarantee the consistency of the subsequent implementation of this feature on a simulation tool.
- b) *Problem formulation and proposal.* Once research gaps were detected, the problems to be addressed were formulated and new data-driven solutions were proposed. For O1 and O2, contributions consist of a methodology for traffic classification, traffic forecasting or throughput estimation. For O3, proposals are new MLB algorithms for the considered scenarios.
- c) *Update of simulation tool.* An existing dynamic system-level LTE simulator coded in Matlab was updated to achieve O2.3 and O3. The most remarkable changes performed were: a) implementing a new realistic two-tier macro-cellular scenario to assess the MLB algorithm designed for O3.1, b) including the NS feature to create datasets for O2.3 and to validate the inter-frequency MLB algorithm proposed for O3.2, and c) adding new 5G service models with different target BLock Error Rate (BLER) and Quality of Service (QoS) requirements to enrich service diversity in tests for O3.2. These updates were validated by checking the consistency of results in long simulations (i.e., one hour of network activity). The resulting simulator is described in detail in appendix [A](#).
- d) *Data collection, preprocessing and analysis.* Datasets used for O1, O2.1 and O2.2 come from commercial cellular networks. The MNO was responsible for collecting and downloading data from the OSS. Once available, raw data was exported to a readable format using proprietary tools provided by the MNO and quickly inspected (e.g., to check names and meaning of available fields). Next, data was preprocessed (e.g., events in traces used for O1 were decoded and synchronized, time series for O2.1 were created, features were computed from raw data in O1 and O2.2, etc.). For O2.3, in the absence of public datasets from commercial 5G networks with NS, data was generated via simulation, and thus preprocessing was not necessary. No matter the data source, once the dataset was created, a preliminary statistical analysis was carried out (e.g., to check statistical feature distribution, correlation analysis...) and outliers are removed.

- e) *Performance assessment.* Validation was always carried out on a realistic environment, i.e., with data from commercial networks when possible and with a simulation tool calibrated with configuration and performance data from the emulated network otherwise. In all cases, proposed solutions were compared with state-of-the-art techniques, considered as a benchmark. Experiments related to each objective were run over varying software, namely Matlab (O1, O3.1 and O3.2), SPSS Modeler (O2.1) and Python (O2.2 and O2.3), whereas result analysis was always carried out in Matlab. Using different platforms for modeling tasks allowed to identify each tool's pros and cons and to provide recommendations to MNOs.

## 1.4 Document structure

Apart from this initial chapter, this document comprises six chapters and two appendices.

Chapter [2](#) introduces the basic principles within the scope of this thesis, namely machine learning, zero-touch networks and NS, defining key concepts and terminology used throughout the document.

Once the thesis is contextualized, chapters [3](#) to [6](#) correspond to the different objectives pursued. Specifically, chapter [3](#) tackles encrypted traffic classification in the RAN with unsupervised learning. Then, chapter [4](#) deals with long-term traffic forecasting in cellular networks through supervised learning. Such an approach is also considered in chapter [5](#) for throughput estimation in both legacy and sliced RANs. Finally, chapter [6](#) explores the development of service-centric data-driven MLB solutions for multi-tier and sliced RANs. For clarity, all these chapters share the same structure. First, a thorough revision of related literature is presented, highlighting the limitations of current solutions and the contributions of this thesis for the corresponding use case. Next, the problem to be solved is formulated by identifying available information, decision variables, constraints and objective function. Then, the proposed solution is detailed and the validation process is presented, including an analysis of results and computational complexity. Finally, the main conclusions are outlined.

Chapter [7](#) summarizes the main findings of this research, providing a list of original contributions and presenting possible future lines to extend the work carried out here.

Two appendices are included at the end of the document. Appendix [A](#) details the

operation of the simulation tool used to create some datasets used in chapter 4 and to validate the algorithms developed in chapter 6. Appendix B provides a summary of this thesis in Spanish.

# Chapter 2

## Technical background

This chapter outlines the technical aspects related to this thesis. First, section [2.1](#) focuses on machine learning, one of the enabling technologies for a ZSM framework. Next, section [2.2](#) introduces zero-touch networks, for which the solutions presented in subsequent chapters are conceived. Finally, section [2.3](#) describes the NS feature, key to ensure customer satisfaction in NGNs.

### 2.1 Machine learning

Machine Learning (ML) is a branch of AI that leverages data to create models able to predict outcomes without being explicitly programmed for that. A wide range of problems can be tackled through ML. Fig. [2.1](#) presents a taxonomy of ML algorithms, which are often divided into three broad groups: Supervised Learning (SL), UnSupervised Learning (USL) and Reinforcement Learning (RL). This section explains the basics, applications and types of algorithms within each category, focusing on how the specific algorithms used in this thesis (marked in gray in Fig. [2.1](#)) operate.

#### 2.1.1 Supervised learning

In SL, a learning algorithm infers a parameterized model from a labeled training dataset that predicts an output  $Y$  for a given input  $X$ , i.e.,  $\hat{Y} = f(X) + e$ , where  $e$  stands for prediction error. Both input and output feature spaces can be multidimensional, i.e.,  $X = (x_1, x_2, \dots, x_n)$  and  $Y = (y_1, y_2, \dots, y_m)$ . The training dataset,  $\mathcal{T}$ , consists on a set of  $N_d$  datapoints for which both explanatory (input) and response (output)

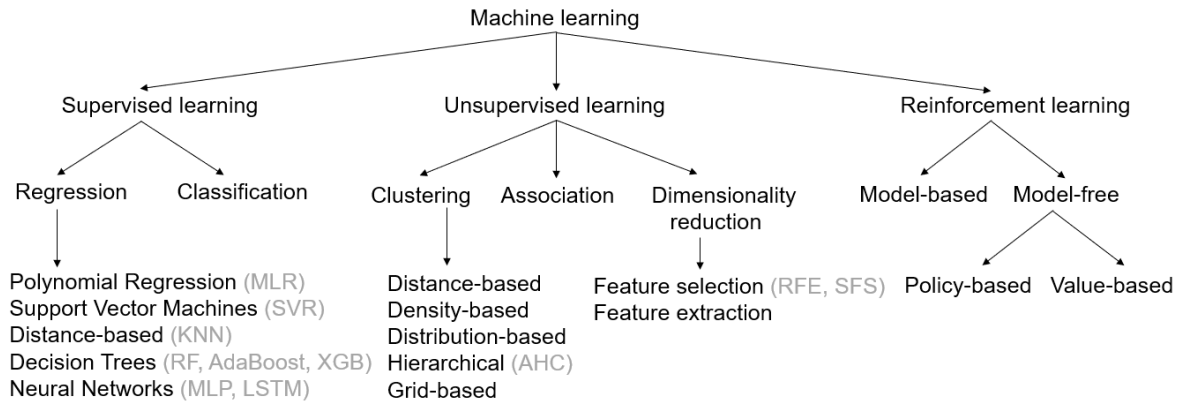


Figure 2.1: Taxonomy of machine learning algorithms.

variables are known, i.e.,  $\mathcal{T} = (X(d), Y(d))$ ,  $d = 1, 2, \dots, N_d$  [37].

SL can be used for classification (i.e., discrete  $Y$ ) or regression (i.e., continuous  $Y$ ). Classification assigns each datapoint to a category from a closed and predefined set of categories according to input features, whereas regression establishes the relationship between input features and continuous output variables. In this thesis, traffic forecasting and throughput estimation are tackled as regression problems. Thus, the explanation hereafter focuses on this application.

SL regression algorithms can rely on polynomials, support vectors, distance, Decision Trees (DTs), Artificial Neural Networks (ANNs) or ensemble techniques combining the output of several weak learners to perform a more robust regression [38]. These alternatives are described below.

### a) Polynomial models

These models capture the relationship between each response variable in  $Y$  and input features (a.k.a. predictors) in  $X$  by adjusting a  $p$ -degree polynomial in the input feature space, i.e.,

$$\hat{y}_i = \beta_{0i} + \beta_{1i}x_1^p + \beta_{2i}x_2^p + \dots + \beta_{ni}x_n^p, \quad i = 1, 2, \dots, m, \quad (2.1)$$

where  $\beta_{ni}$  is the slope of the regression surface of output variable  $y_i$  with respect to predictor  $x_n$  and  $\beta_{0i}$  is the intercept. The optimal solution is reached by ordinary linear least squares fitting, which adjusts regression coefficients to minimize the sum of squares of residuals between estimates and ground-truth data. For the particular case of  $p=1$ , this algorithm is referred to as Multi-Variable Linear Regression (MLR).

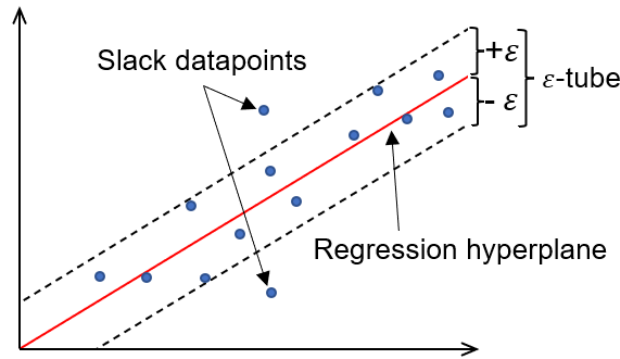


Figure 2.2: Elements of support vector regression algorithm.

### b) Support vector regression

Support Vector Regression (SVR) aims to find the hyperplane that best fits the training dataset. For this purpose, input features are mapped to a higher dimensional feature space by using a *kernel* function (e.g., linear, polynomial or radial basis), and the best regression hyperplane is constructed in that large and transformed version of the feature space [39]. Unlike MLR, SVR neglects all errors below a certain value controlled by the sensitivity parameter,  $\epsilon$ . As illustrated in Fig. 2.2, the best hyperplane minimizes the deviation of datapoints outside the insensitive  $\epsilon$ -tube (a.k.a. slack datapoints) to the maximum allowed error margin. Moreover, the regularization parameter,  $C$ , restricts the absolute value of regression coefficients. Both  $\epsilon$  and  $C$  parameters control the trade-off between regression accuracy and model complexity (i.e., the smaller  $\epsilon$  and larger  $C$ , the better the model fits the training data, but the higher risk of overfitting) [37].

### c) Decision trees

A DT is a flow-chart model that infers simple decision rules from the training dataset. Fig. 2.3 represents DT operation. In each node, the value of a specific input feature of the datapoint is compared to a certain threshold, and the left or right branch is chosen accordingly, leading to another node. This process is repeated until the datapoint reaches a leaf node. During training, the decision threshold of each node is adjusted to reduce the impurity of child nodes. To avoid overfitting, DTs are pruned when: a) a node has less than a minimum number of samples, b) a maximum depth is reached, or c) a new split does not lead to a significant decrease of impurity. Then, inference consists on passing inputs of the new datapoint through the DT until a leaf node is

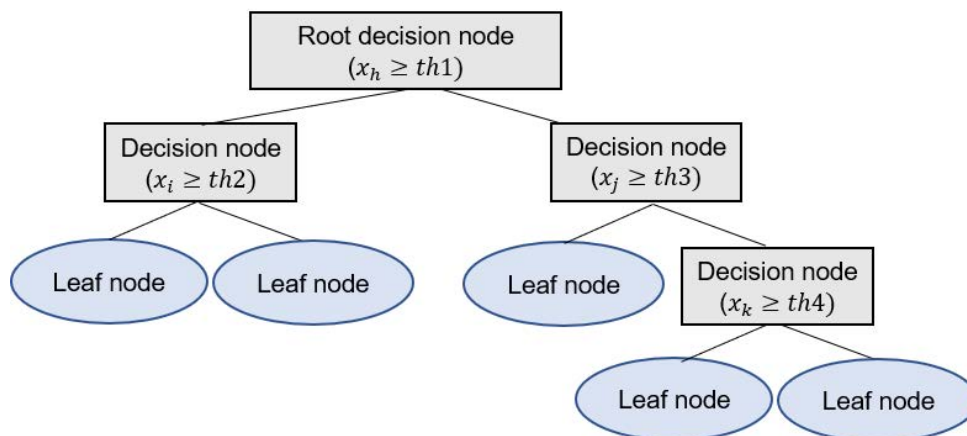


Figure 2.3: Example of decision tree.

reached. Output value for the new datapoint is computed as the average output values for the training datapoints belonging to that leaf node [37].

DTs are often weak learners by themselves, but powerful if used in ensemble methods [40]. Three ensemble methods based on DTs are considered in this thesis, namely Random Forest (RF), Adaptive Boosting (AdaBoost) and eXtreme Gradient Boosting (XGBoost). In RF, several independent DTs are trained with different subsets of datapoints (a.k.a. aggregated bootstrapping or bagging) and input features. Then, the outputs of all DTs are averaged to get the final output [41]. In contrast, in AdaBoost and XGBoost, DTs are sequentially created, so that  $DT_i$  tries to improve model performance obtained with  $DT_1$  to  $DT_{i-1}$  (a.k.a. boosting). AdaBoost considers one-level DTs, referred to as stumps.  $DT_1$  splits data based on the input feature providing the lowest prediction error over the training dataset. When training subsequent DTs, the weight of datapoints with high error in the previous DT is increased, whereas the weight of datapoints with low error in the previous DT is reduced. The weight of each DT prediction in the final output depends on its error rate [42]. Alternatively, in gradient boosting algorithms such as XGBoost,  $DT_1$  consists on a leaf with the mean of the output feature for training datapoints. Then, at each iteration, the gradient descent optimization algorithm is used to minimize a differentiable loss function over residuals from the previous DT. To avoid overfitting, the learning rate shrinks the contribution of each DT in the final output. XGBoost is an advanced version of gradient boost including L1 and L2 regularization and depth-first tree pruning (i.e., DTs are grown up to the maximum depth and then pruned backward until the improvement in loss function is below a threshold) [43].



#### d) $k$ -nearest neighbors

$k$ -Nearest Neighbors (KNN) is a non-parametric SL algorithm (i.e., it does not assume any specific form for the regression function) based on distance. It relies on the premise that observations with similar characteristics tend to have similar outcomes. To estimate the response variable of a new observation, KNN identifies its  $k$  nearest neighbors in the training dataset according to some previously defined distance metric (e.g., Euclidean, Chebyshev...) and a search algorithm (e.g., ball tree,  $k$ -dimensional tree...). Then, it computes the output as the (sometimes weighted) average of outputs for such neighbors [44].

The Euclidean distance, often used as distance metric for regression, between a pair of datapoints  $p$  and  $q$  is computed as

$$d_{euclidean}(p, q) = \sqrt{\sum_{k=1}^n (x_k(p) - x_k(q))^2}. \quad (2.2)$$

#### e) Artificial neural networks

ANNs rely on a statistical learning method inspired by the structure of the human brain. In ANNs, computation units (a.k.a. nodes or neurons) grouped in layers perform non-linear calculations through activation functions, capturing complex relations among input features [45]. Fig. 2.4.a) shows an example of ANN. Neurons in the input layer consist of the value of input features. Then, the datapoint passes through one (shallow ANN) or more (Deep ANN, DNN) hidden layers with a configurable number of nodes. The outcome is then processed by an output layer whose number of neurons is equal to the output size.

Several ANN architectures have been developed to solve different problems, surveyed in [46]. For instance, in feed-forward ANNs, information moves from the input to the output nodes through hidden nodes (if any). In contrast, in recurrent ANNs, connections between nodes form a graph along a temporal sequence, allowing to capture time dependencies in input data. Conversely, convolutional ANNs capture complex patterns in input features through convolutional kernels, compressing input data.

Two types of ANNs are considered in this thesis. The first architecture is the Multi-Layer Perceptron (MLP), widely used for regression. MLPs are fully connected feed-forward ANNs, i.e., every node in a layer is connected to all nodes in the subsequent

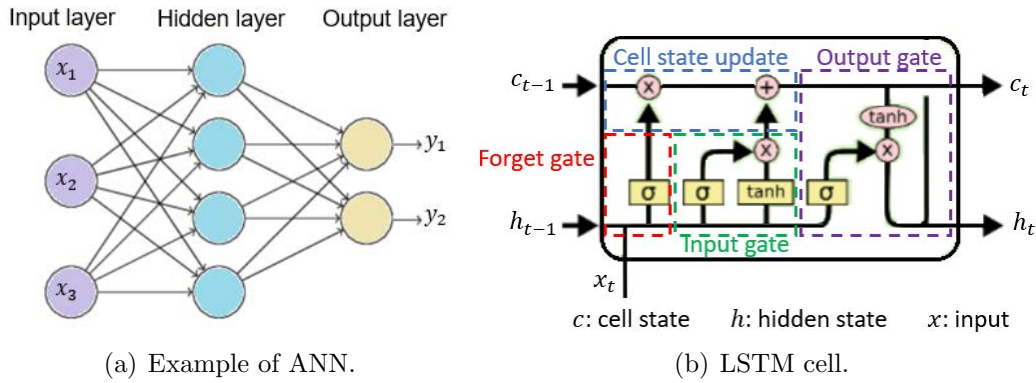


Figure 2.4: Neural network elements.

layer. In MLPs, nodes in hidden and output layers are perceptrons. A perceptron  $p$  performs a weighted sum of  $N$  inputs  $x_i$ ,  $i = 1, 2, \dots, N$ , plus a bias, and then passes the result through an activation function  $f_a$  (e.g., linear, sigmoid or hyperbolic tangent... [47]). Such a computation can be expressed as

$$Y(p) = f_a \left( b(p) + \sum_{i=1}^N x_i \cdot w_i(p) \right), \quad (2.3)$$

where  $w_i(p)$  and  $b(p)$  are the weights and bias, respectively (i.e., the trainable parameters) [48].

The second type of ANN used in this thesis is a recurrent network based on Long Short-Term Memory (LSTM) units (a.k.a. LSTM network), conceived to be used with time series data. LSTM networks comprise an input layer, one or more hidden layers with LSTM cells as nodes and an output layer made of perceptrons. A LSTM cell, illustrated in Fig. 2.4.b), provides a short-term memory that can last long, modeled as a cell state. Several perceptrons and activation functions are combined to create three gates regulating the flow of information into and out of the cell state. At a given step, the forget gate decides what information from prior cell state must be forgotten. Likewise, the input gate decides what new information must be included in current cell state. Finally, the output gate computes the current output of the LSTM cell, referred to as hidden state. Several LSTM hidden layers can allow capturing dependencies in different time scales. The reader is referred to [49] for a more detailed explanation of LSTM cells.

Once ANN architecture is defined, weights are initialized (e.g., zero-initialization or random Glorot initialization [50]). The training process consists in iteratively propa-

gating errors obtained for the training dataset with current weights back through the ANN, and tuning weights accordingly to minimize a predefined loss function, e.g., the root mean squared error or the mean absolute error. For this purpose, the training dataset is passed through the network several times, referred to as epochs. A solver algorithm (e.g., adaptive moment estimation – Adam– [51] or Limited-Memory Broyden–Fletcher–Goldfarb–Shanno –L-BFGS– [52]) orchestrates forward inference and backward gradients for weight updating. In each epoch, some optimizers (e.g., L-BFGS) process all training datapoints at once (i.e., an epoch comprises a single iteration), whereas others (e.g., Adam) process data in mini-batches with a fixed number of datapoints (i.e., an epoch may comprise several iterations). An early stopping condition is often set to avoid overfitting (e.g., loss function for the validation dataset is below a threshold or has not improved significantly over the last epochs) [53].

## 2.1.2 Unsupervised learning

USL seeks to identify unknown patterns in unlabeled data, which is useful for tasks such as clustering, dimensionality reduction and association. These applications are briefly introduced next.

### a) Clustering

Clustering aims to identify groups of datapoints based on the similarity between observations. Clustering algorithms can be classified into hierarchical, partition-based, density-based, distribution-based, grid-based and model-based [54]. In *hierarchical* methods (e.g., agglomerative hierarchical clustering [55]), clusters at one level are joined at the next level, creating a cluster tree. In *partition-based* algorithms (e.g., k-means or k-medoids [56]), spherical non-hierarchical clusters are created around a central structure (a.k.a. centroid). *Density-based* algorithms (e.g., Density-Based Spatial Clustering of Applications with Noise, DBSCAN [57]) identify high- and low-density regions and create arbitrary-shaped clusters accordingly. Distribution-based techniques (e.g., Gaussian mixture model [58]) compute several statistical distributions, so that clusters comprise datapoints belonging most likely to the same distribution. In *grid-based* strategies (e.g., entropy-based subspace clustering [59]), data is divided into a finite number of cells that form a grid structure on which all of the clustering operations are implemented. Finally, *model-based* approaches (e.g., self-organization and associative memory [60]) optimize the best fit between the given data and a mathematical model.

In this thesis, agglomerative hierarchical clustering is used for encrypted traffic classification. This algorithm groups datapoints in clusters based on their similarity. The algorithm starts by treating each datapoint as a singleton cluster. Then, (dis)similarity between every pair of datapoints in the dataset is computed with a given distance metric (e.g., Euclidean, Chebyshev...), and the two closest clusters are merged into a single cluster by a linkage function (e.g., unweighted average distance, inner squared distance...) based on such similarity information. This process is repeated until all clusters merge into one root cluster or until there are no two clusters with a similarity lower than a predefined cutoff threshold. The result is a tree-based representation of the data, referred to as *dendrogram*, representing the order of cluster merging and the distance between clusters.

## b) Dimensionality reduction

Dimensionality reduction compresses input data into a reduced set of new features, being helpful to visualize multi-dimensional data or as a preprocessing step before applying SL. For a given problem, an adequate dimensionality reduction often turns into similar or even better performance with simpler ML models. This task can be tackled through Feature Selection (FS) or Feature Extraction (FE) [61]. In FS, a subset of relevant features is selected according to a predefined criterion. FS comprises filtering, wrapper and embedded methods [62]. *Filtering* methods select features according to their variance or correlation with the outcome variable. These schemes are very efficient and independent of the subsequent SL algorithm and can thus be used as a preprocessing step. However, they might fail to find the optimal subset of features. Alternatively, *wrapper* methods select subsets of variables according to their usefulness for a given SL algorithm. Despite being computationally expensive, they provide the best subset of features per algorithm. Finally, *embedded* methods integrate feature selection in the learning process. In contrast, in FE, a new (reduced) set of features is built by combining the features from the original set. An extended FE method for numerical variables is principal component analysis, creating new features (a.k.a. principal components) computed as orthogonal linear combinations of original features [63].

In this thesis, different filtering and wrapper FS schemes are used to select a subset of representative input features to estimate throughput metrics in the RAN.

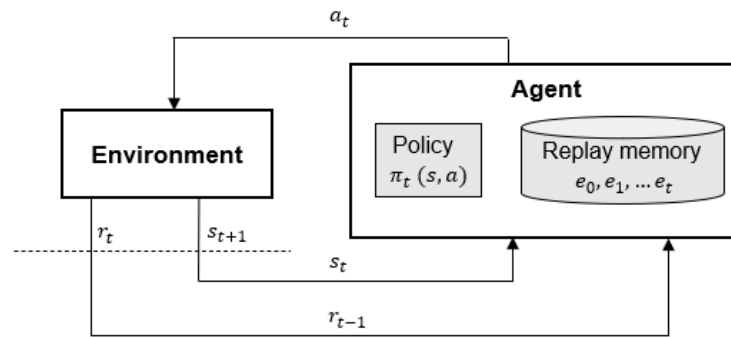


Figure 2.5: Reinforcement learning agent.

### c) Association

Association aims to find relationships between different features, usually represented as rules or frequent itemsets. Some examples of association algorithms are Apriori and Eclat algorithms [64]. These techniques rely on three key concepts: a) support, measuring how often an item appears in the dataset, b) confidence, indicating how often a rule is fulfilled, and c) lift, measuring the strength of a rule.

## 2.1.3 Reinforcement learning

RL is a ML technique where an agent takes actions in an environment aiming to maximize a cumulative reward. Fig. 2.5 sketches the agent-environment interaction. At every time step  $t = 0, 1, 2, \dots$  when the environment is in state  $s_t \in \mathcal{S}$ , the agent takes an action  $a_t \in \mathcal{A}$  according to some policy  $\pi_t(s, a)$ , for which receives a reward  $r_t$  and commutes to a new state  $s_{t+1}$ . If  $s_t$  retains all relevant past information, RL can be modeled as a Markov decision process.

RL algorithms can be model-based or model-free. The former model environment's transition function to make predictions about the consequences of taking actions (i.e., deductive approach), whereas the latter learn from experience (i.e., inductive approach). At the same time, model-free RL methods can be policy-based or value-based. Policy-based approaches learn a deterministic or stochastic optimal policy  $\pi^*$ . In contrast, value-based methods learn the optimal value function leading to the optimal policy. In all cases, the trend is incorporating DNNs into the solution (a.k.a Deep Reinforcement Learning, DRL), allowing agents to make decisions from unstructured input data without manual engineering of the state space. Some well-known DRL approaches are deep Q-learning or deep deterministic policy gradient. For further information on these and other RL algorithms, the reader is referred to [65].

## 2.2 Zero-touch networks

NGNs are expected to cope with a wide range of services, technologies, vertical industries and devices. The associated increase in network performance, flexibility and cost efficiency envisages an unprecedented complexity in network operation, management and orchestration. This fact, together with the latest advances in BDA and ML, have propelled the trend towards *zero-touch* networks with fully automated SON capabilities enabling an E2E closed-loop automation of network and service management operations.

In this section, SON is first introduced, focusing on the two specific use cases covered in this thesis: network redimensioning and MLB. Then, the basis of zero-touch networks is discussed, identifying different types of data available in the RAN to be used in big-data-empowered SON tools.

### 2.2.1 SON use cases

LTE entailed a growth in the number and types of cells (e.g., macro cells, small cells...) as well as in the set of parameters in base stations (a.k.a. evolved Nodes B, eNBs) compared to legacy networks with 2G and 3G RATs. Such an increase in RAN size and complexity boosted the interest of operators in SON techniques offering automatic planning, deployment, optimization and maintenance of network nodes. This automatic management paradigm speeds up network deployment, reduces capital and operational expenditures, ensures QoE provisioning, introduces processes too fast and/or too tedious to be implemented manually and releases radio engineers from repeating manual tasks in space and time, thus diminishing the impact of human errors on network performance.

Several procedures can be automated in cellular networks. SON use cases are often divided into four broad groups, namely self-planning, self-deployment, self-optimization and self-healing, described below [66]:

- a) *Self-planning* comprises all procedures related to the definition of a new Network Element (NE), excluding NE acquisition and preparation. This process implies planning NE location, radio and transport parameters and aligning data from all NEs in a RAN. Some examples of self-planning use cases are automatic site selection [67] or self-configuration of antenna power [68] and tilt [69], physical cha-

annel identifiers [70] or UpLink (UL) power control parameters [71]. Self-planning solutions are run: a) during the network planning stage, and b) when network redimensioning actions are executed during the operational stage.

- b) *Self-deployment* automates the process of bringing new NEs into operation, including automatic HardWare (HW) installation, node authentication, SoftWare (SW) download and self-test, among other tasks.
- c) *Self-optimization* aims to constantly make the most of network assets by self-tuning node parameters to adapt to traffic fluctuations (e.g., holidays periods, social events...), changes in network topology (e.g., deployment of new cell or carrier) or variations in radio channel conditions (e.g., new building, seasonal variations in propagation environment...) happening during the operational stage. For this purpose, user equipment and base station performance is monitored and analyzed, so that optimization actions are triggered on affected NEs when necessary. Some of the most extended self-optimization use cases are coverage and capacity optimization [72], mobility robustness optimization [73], inter-cell interference coordination [74], self-tuning of scheduling parameters [13] [75], self-optimization of tracking area list [76] or load balancing [33] [77].
- d) *Self-healing* automates troubleshooting, comprising fault detection and diagnosis. The former process seeks to find problematic cells, whereas the latter identifies fault causes based on symptoms (e.g., alarms) and decides the recovery action to be taken. Note that self-healing is a critical task in the RAN, since cells sometimes serve an area with little (or without) redundancy, and cell outage thus strongly degrades QoE. Some use cases are automatic alarm prioritization [8], cell outage detection [78], root cause analysis [14] and cell outage compensation [79].

Legacy SON functionalities initially conceived for LTE have been extended to other RATs and to optimize inter-RAT procedures (e.g., mobility). For instance, [19] discusses new SON use cases arising in 5G related to new features and NFs (e.g., optimization of spectrum sharing between slices in NS scenarios). A multi-RAT SON empowers operators with comprehensive, holistic and powerful tools, harmonizing the whole network management and optimizing operational efficiency.

The two SON use cases covered in this thesis, namely RAN redimensioning and MLB, are introduced next.

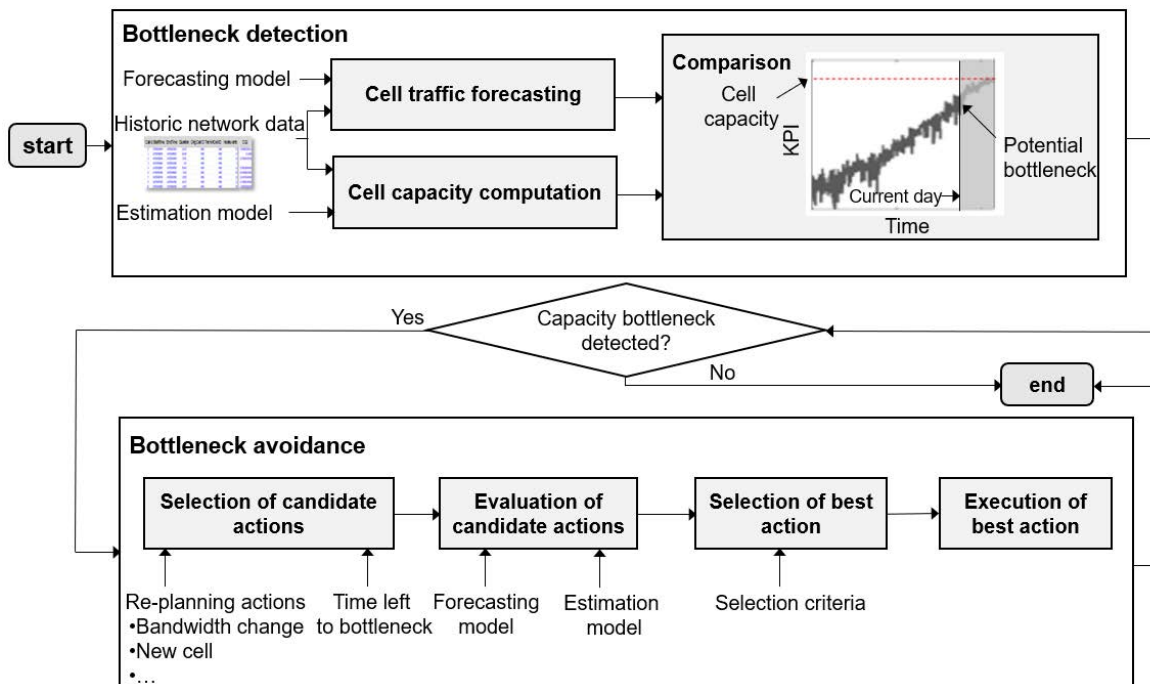


Figure 2.6: Flow diagram of RAN redimensioning in radio planning tools.

### a) RAN redimensioning

RAN redimensioning is a procedure within the scope of self-planning that aims to detect potential capacity bottlenecks in advance so that RAN configuration or equipment can be timely upgraded. This SON use case is critical to warrant service provisioning in NGNs, where the coexistence of services with very different requirements and the (de)activation of slices will lead to complex and changing traffic patterns and capacity demands.

Fig. 2.6 illustrates the workflow of RAN re-dimensioning in radio planning tools. The process comprises bottleneck detection and avoidance stages, detailed next:

- 1) To detect potential capacity bottlenecks, traffic forecasts per cell are compared to some predefined KPIs reflecting cell capacity (e.g., aggregated cell throughput in DownLink (DL) in high load conditions). With the user-centric network management approach currently preferred by MNOs, cell capacity KPIs should be complemented by other indicators that better reflect end-user performance in congestion scenarios (a.k.a. user capacity). For instance, DL user throughput is often regarded as a significant QoE metric for eMBB services, which are the first delivered in 5G networks. Thus, this metric should be considered for redimensioning purposes. It is expected that the higher the averaged user throughput in



the DL of a congested cell, the higher QoE level can be guaranteed for eMBB services in that cell.

- 2) If a potential future lack of resources is foreseen in the cell, an alarm is activated that triggers a bottleneck avoidance process. Otherwise, the procedure ends.
- 3) If bottleneck avoidance is required, a set of candidate actions is evaluated. The considered alternatives depend on how much in advance the problem is envisaged. Imminent issues detected with short-term traffic forecasts (e.g., several hours or days in advance) often trigger temporary changes in network parameters (e.g., a more efficient voice coding scheme [6], new HO margin settings for traffic sharing between adjacent cells [77] or naive packet schedulers for a lower computational load [80]). Such quick actions, dealing with fast fluctuations of traffic demand, act as temporary solutions until the normal network state is recovered or, if the problem persists, more stable solutions relying on network capacity extensions are taken. In contrast, if the lack of resources is foreseen several months ahead, more future-proof solutions can be implemented, such as bandwidth extension [81], license extension for the maximum number of channel elements and/or simultaneous users [9] or new carriers/co-sited cells. The assessment process seeks to estimate candidate actions' impact on cell capacity. Note that capacity indicators such as throughput may depend on many additional factors (e.g., traffic mix, terminal capabilities...), thus being extremely difficult to isolate the impact of a replanning action on these metrics. Alternatively, the impact of replanning actions on lower-level radio network performance indicators (e.g., bandwidth, channel quality indicator distribution, power) can more easily be predicted. Then, such predictions can be used as inputs to a capacity estimation model. Finally, the best action (e.g., the most cost-effective alternative that solves the capacity bottleneck) is selected and executed.

The above process is repeated periodically (e.g., daily) on a cell basis. Note that detecting fake potential bottlenecks could imply unnecessary investments in capacity extensions. Apart from the associated cost, these actions might degrade the performance of other cells in the scenario due to interference. Conversely, not detecting real potential capacity problems may degrade user experience. As a consequence, radio planning tools must rely on accurate forecasts of upcoming traffic demand and estimates of cell performance for a precise proactive bottleneck detection and action evaluation. For this purpose, cutting-edge tools rely on BDA and ML over data collected in the

OSS so as to perform these checks automatically [82] [83]. Such an approach is followed in this thesis.

## b) Mobility Load Balancing

MLB is one of the most extended self-optimization use cases in the RAN. It aims to alleviate congestion problems due to traffic fluctuations by offloading traffic from congested to underutilized cells through mobility parameter self-tuning. Since most MLB proposals rely on adjusting HO parameters, the HO procedure is described first and MLB is detailed later. The explanation focuses on LTE and NR technologies <sup>1</sup>, for which MLB algorithms in this thesis are conceived.

In the above-mentioned RATs, mobility of connected User Equipments (UEs) is handled by an event-based hard HO procedure. The UE measures signal level and quality received from the serving cell and a set of neighbor cells. Measurement reports are then sent to the serving base station (i.e., eNB in LTE or gNodeB –gNB– in NR) either periodically or triggered by an event. Finally, the base station makes the HO decision based on a predefined HO triggering event. Table 2.1 summarizes events defined for intra-RAT measurement reporting and HO in the absence of multi-connectivity [84] [85]. Those events can be evaluated with different report quantities, namely Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ) or Signal-to-Interference-plus-Noise Ratio (SINR). For instance, event A3 is triggered for a UE  $u$  when the selected report quantity  $Meas$  received from a neighbor cell  $j$ ,  $Meas_u(j)$ , exceeds that received from the serving cell  $i$ ,  $Meas_u(i)$ , by a certain HO Margin (HOM) defined per adjacency,  $HOM(i, j)$ , expressed as

$$Meas_u(j) > Meas_u(i) + HOM(i, j) \quad \forall t = 1, 2, \dots, TTT(i, j) \quad , \quad (2.4)$$

where all terms are expressed in logarithmic scale. To avoid unnecessary ping-pong HOs leading to signaling overload and user experience degradation, a UE only performs a HO if the event persists over a time interval referred to as Time To Trigger (TTT), also defined per adjacency (i.e.,  $TTT(i, j)$ ).

Regardless of the selected HO scheme (defined by the combination of triggering event and report quantity), offsets/thresholds in Table 2.1 and TTT are key parameters that can be tuned per adjacency for optimization purposes. Offsets/thresholds

<sup>1</sup>This thesis covers 5G networks without multi-connectivity.

Table 2.1: Intra-frequency handover events.

Event	Description
A1	Serving cell becomes better than threshold
A2	Serving cell becomes worse than threshold
A3	Neighbor cell becomes offset better than serving cell
A4	Neighbor cell becomes better than threshold
A5	Serving cell becomes worse than threshold1 and neighbor cell becomes better than threshold2

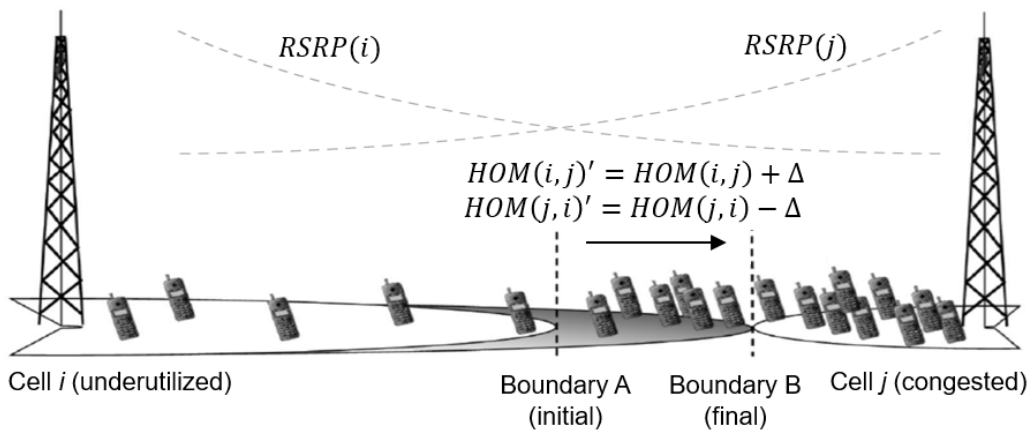


Figure 2.7: Example of traffic steering by tuning handover margins [87].

determine the specific condition triggering the HO procedure, thus being very powerful for MLB [86] [33]. In contrast, TTT ensures that the triggering condition lasts for a while, avoiding HOs caused by an instantaneous degradation in radio channel quality that may trigger mobility events by chance. As a consequence, TTT is often tuned for mobile robustness optimization (e.g., to avoid ping-pong HOs) [73].

To illustrate how MLB can be tackled through HO parameter tuning, Fig. 2.7 depicts two neighbor cells  $i$  and  $j$  in a mobile network. As usual, intra-frequency mobility is handled through HOs triggered by event A3 driven by RSRP. In this process, HOMs are set to place cell boundary at point A, so that both cells have a similar service area (in a live network, there should be a hysteresis area that has been neglected in the example for simplicity). With this set-up, due to the uneven distribution of UEs in the network, cell  $i$  is underutilized, whereas cell  $j$  is congested. However, if  $HOM(i, j)$  is increased by  $\Delta$  dB and  $HOM(j, i)$  is decreased by  $\Delta$  dB, displacing cell boundary to point B, all UEs in the grey area are handed over from cell  $j$  to cell  $i$ , thus offloading cell  $j$ . In MLB algorithms, such checks and decisions are performed automatically.

### 2.2.2 Big-data-empowered SON

ML and BDA techniques are envisioned as key enablers for autonomous network and service management [88]. Zero-touch networks rely on fully automated SON capabilities to: a) boost the efficiency of service delivery, b) reduce the operational expenditures, c) lessen or even omit engineer intervention in network management procedures, and d) ease coordination among SON functions with conflicting goals, dynamically determining the operating point providing the best performance trade-off [17]. ZSM architectures will rely on BDA platforms to handle the vast amount of data (i.e., configuration data, call traces, logs...) gathered in current and future cellular networks during normal operation. Additionally, ML will bring intelligent decision-making to network management through SON tools using network data to acquire knowledge learned from experience.

References [17], [18] and [20] present comprehensive surveys of ML-based SON solutions, including tasks related to the use cases considered in this work (e.g., RAN dimensioning in [82] [89], MLB in [90] [91] and encrypted traffic classification in [92] [93]). Problems that require estimating, forecasting and classifying variables (e.g., performance estimation of traffic forecasting) can be tackled through SL. In contrast, USL is helpful for pattern recognition (e.g., traffic classification). Finally, RL is the preferred option to address issues requiring network parameter control in complex scenarios (e.g., HOM tuning).

Current mobile networks generate a vast amount of data in the form of measurements and interaction registers that can be used in data-empowered SON, e.g., Minimization of Drive Test (MDT) data, charging data records, Configuration Management (CM) data, Performance Management (PM) data or connection traces [94]. All the solutions proposed in this thesis rely on the three latter data sources, which are introduced next.

#### a) Configuration and performance management data

CMs and PMs provide information about the state of NEs (e.g., base stations) during a time interval referred to as Reporting Output Period (ROP). CMs consist of network parameter settings (e.g., cell bandwidth or transmit power), whereas PMs are counters that provide aggregated measurements reflecting the performance of the NE (e.g., total data volume, number of radio resources scheduled in data channels...) during the

ROP [95] [96]. Higher-level KPIs and Key Quality Indicators (KQIs) better reflecting network performance (e.g., physical resource block utilization) and user satisfaction (e.g., average user throughput) are computed from CMs and PMs.

PMs and CMs are collected in NEs and then sent to the OSS at the end of the ROP, where they are gathered for a more extended period (e.g., months or even years) for network management purposes. In current networks, the ROP is often set to 15 min [97]. In NGNs, this value is expected to be shortened to cope with the higher dynamism and ambitious performance goals. Note that storing PMs/CMs of every NE with a high frequency implies deploying large databases. To circumvent this issue, in the past, operators often opted for storing PMs/CMs on a ROP basis for a while, but only keeping daily or monthly busy-hour information in the long term (i.e., years). Such a trend is disappearing with the latest advances in data storage and processing.

Most classical SON tools use PMs and CMs as inputs to heuristic controllers designed from expert knowledge. However, such data can also be exploited to train ML models driving decision making. Nonetheless, NE-aggregated information may not suffice for some service-oriented NFs, requiring UE-level information only available in radio connection traces.

## b) Traces

Radio connection traces (a.k.a. traffic recordings) contain signaling events in the radio interface [98]. In this context, an event is a report including measurements and performance information (e.g., signal level, bit rate...). Events are grouped into two categories: internal and external events. Internal events are generated by base stations and are specific to each vendor. In contrast, external events include signaling messages that the base station exchanges with other NEs via standard protocols, such as Radio Resource Control (RRC) or S1 application protocol. Events selected by the MNO are recorded in a trace file per cell generated periodically after each ROP, which is then sent to the OSS. Two types of trace files are distinguished: Cell Traffic Recording (CTR) and UE Traffic Recording (UETR). While CTRs include events of all users in the cell anonymously, UETRs store information of a specific user selected by the operator [99].

Radio traces allow to generate new indicators different from counters provided by vendors, thus being precious for MNOs [100]. Moreover, with MDT feature launched in release 10, information in radio traces can be geositioned [101]. Several trace-based SON tools have been proposed in the literature. For instance, traces can be used

in network planning to derive spectral efficiency curves required in cellular planning and simulation [102] or the spatiotemporal distribution of radio resources in a live network [103]. Likewise, traces can be used in the operational stage to check the performance of certain types of connections for benchmarking purposes [104], tune network parameters (e.g., link adaptation offset [105] or antenna tilt angle [72]) or find the root cause of problems (e.g., dropped connections [106]).

## 2.3 Network slicing

NS is a new feature in 5G networks allowing the coexistence of multiple logical networks tailored for a specific application or tenant (e.g., Over The Top –OTT– service providers or virtual operators) operating simultaneously over a common physical network infrastructure [15]. Together with other features such as software-defined networks or multi-edge computing, NS provides the flexibility and scalability required to handle the diverse end-users (i.e., humans or machines), devices (e.g., smartphones, vehicles, wearables, sensors...) and services with highly diverging QoS requirements (i.e., energy efficiency for mMTC, E2E latency-reliability for uRLLC, peak data rate for eMBB...) coexisting in NGNs [4]. As a result, NS maximizes revenues for infrastructure owners due to the efficient usage of network assets while opening up new go-to-market models for vertical industries [107].

Slice life cycle comprises four stages. In the preparation phase, the infrastructure owner (often a MNO) and the future tenant reach a SLA. Then, the new slice instance is designed and set up in the commissioning stage. As illustrated in Fig. 2.8, an E2E slice comprises HW, SW and radio resources together with a collection of Virtualized NFs (VNFs). Such assets provide storage, processing and networking capabilities required to comply QoS, security, mobility and availability conditions specified in the SLA during the operation phase. Finally, in the decommissioning step, slice resources are released [15]. A central MANager and Orchestrator (MANO) manages the life cycle of all slices operating in a network. Among other tasks, the MANO splits resources among slices, decides which VNFs (e.g., UE access control, HO...) are common to all/multiple slices, which VNFs are tailored or omitted per slice, and sets up VNF parameters [108].

In the literature, several works have covered different aspects related to NS. For instance, [109] surveys works focused on the administrative aspect, where the use of distributed ledger technologies (e.g., blockchain) has been recognized as as powerful

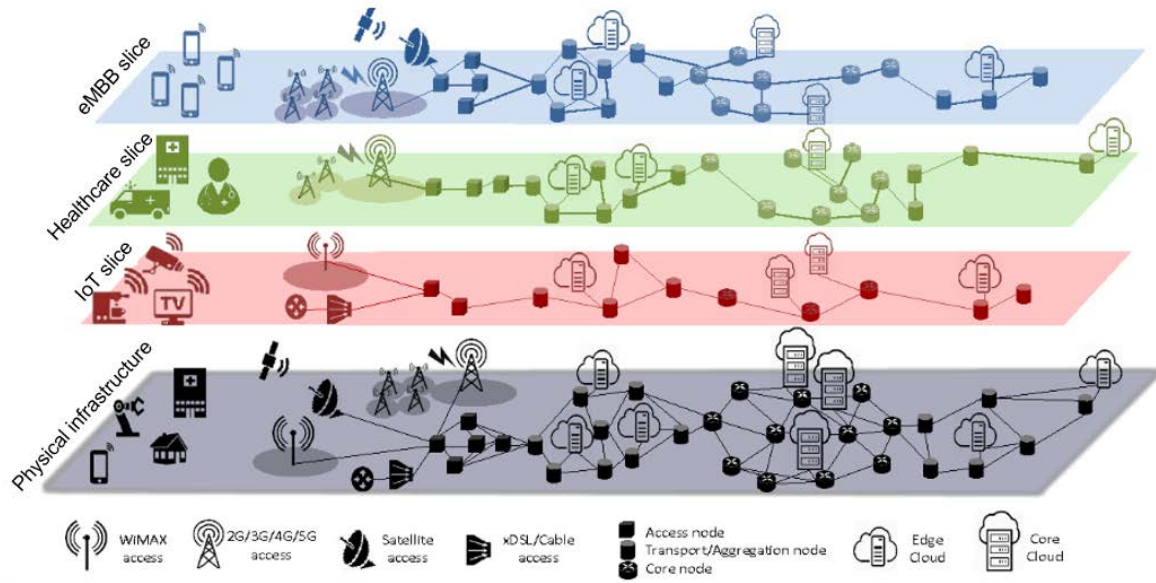


Figure 2.8: Network slicing [118].

tool. In [110], several architectures to provide E2E NS are surveyed. The information model required for NS in RAN, core and transport network domains is described in [111]. Other works focus on MANO, defining slice management policies for tasks such as slice admission control [112] or resource orchestration [113] [114], often relying on ML [115]. Regarding resource orchestration, ideally, slices must be self-contained and logically isolated, which increases robustness (i.e., faults in one slice do not affect other slices), improves security (i.e., an attack to a slice does not affect other slices) and reduces time-to-market due to few dependencies on external NFs. In the RAN, designing isolated slices is specially challenging due to the inherently limited and shared nature of spectrum. In [116], four different strategies to split radio resources among slices are compared in terms of spatial, temporal and frequency granularity of assignment, traffic and radio isolation, and customization. Several capacity brokers have been proposed following these strategies, sometimes performing joint spectrum split and access control [117].

It should be pointed out that enabling NS poses additional changes and challenges from a network management perspective. The inclusion of the new slice domain entails: a) the arise of new SON use cases related to slice (de)commissioning and maintenance (e.g., automatic capacity brokers), and b) the need for new slice-aware solutions for classical SON use cases (e.g., slice-aware MLB). To support these NFs, data such as PMs, CMs and connection traces must be collected not only per cell, but also on a slice basis. Moreover, privacy issues may prevent the central MANO from accessing slice-

level data managed by tenants. Additionally, NS entails a higher network dynamism due to the activation, deactivation and redimensioning of slices, thus requiring a more frequent and faster operation of SON NFs [119]. All these factors must be considered when extending SON use cases to NS scenarios, including those addressed in this thesis. Regarding RAN dimensioning, legacy cell performance models may fail in a NS scenario due to: a) the coexistence in the same geographical area of multiple slices with very different behaviors over the spatial and temporal domains, and b) the split of radio resources among slices, which may prevent the efficient use of cell bandwidth. As a result, a new slice dimensioning NF arises, which requires slice-level models to map specific slice characteristics (e.g., traffic type and demand, bandwidth, spectral efficiency...) to performance metrics. These slice-level models are also continuously exploited during the slice operation phase to check when spectrum sharing has to be reconfigured to meet the SLA while minimizing resource over-provisioning [120]. Likewise, new slice-aware algorithms for self-optimization use cases such as MLB must be developed considering slice-specific SLA aspects in the decision-making process. All these tasks are addressed in this thesis.



# Chapter 3

## Classification of encrypted traffic in cellular networks

This chapter tackles the problem of classifying connections carrying encrypted traffic per application type in the RAN. An accurate traffic classification can benefit many network management tasks, including capacity planning, troubleshooting, QoE management, slice design or NF optimization. In fact, some of the self-management solutions presented later in this thesis assume prior knowledge of the traffic mix in a cell. The chapter is divided into five sections. Section [3.1](#) reviews the related literature. Section [3.2](#) introduces some key concepts to understand the proposed classification method, described later in section [3.3](#). Section [3.4](#) assesses method performance over a trace dataset from a live LTE network. Finally, section [3.5](#) summarizes the main conclusions.

### 3.1 Related work

Traffic classification aims to associate network traffic with the underlying generating application. For this purpose, in LTE, each connection has a QoS Class Identifier (QCI) used to prioritize services [\[121\]](#). Similarly, a 5G QoS Identifier (5GQI) is assigned per traffic flow in NR [\[122\]](#). Such information is registered in measurements collected by radio NEs. However, even if some of these identifiers are associated with a single service, others comprise services of very different nature. For instance, in LTE, QCIs 6, 8 and 9 contain a mixture of multimedia, interactive and Transmission Control Protocol (TCP)-based services, namely instant messaging, streaming, web surfing or

app download. Such a coarse granularity complicates any application-oriented task. As such, more precise traffic classification methods are required.

In legacy Internet Protocol (IP)-based networks, traffic was classified in the past by port number [123]. Such an approach is unreliable today due to the proliferation of new applications with non-standard or randomly generated ports [124]. As an alternative, payload-based methods (e.g., deep packet inspection) match the IP packet payload with a set of stored signatures to classify network traffic [125]. However, this strategy is useless for encrypted traffic [36]. To solve these limitations, several works tackle traffic classification by analyzing payload-independent flow characteristics, relying on the premise that traffic from different applications typically has distinct flow patterns (a.k.a. app fingerprints). In fixed networks, several flow-based methods have been proposed to classify traffic in real time by using the first packets of the flow (early classification) [126] [127] or offline based on the whole flow (late classification) [125]. These approaches have also been extended to wireless networks by leveraging the ability of SL to identify app fingerprints. In [128], a device-fingerprinting scheme based on learning traffic patterns of background activities is proposed. The method uses a Support Vector Classifier (SVC) and KNN, trained with data from 20 users with different combinations of apps connected to a 3G network. In [129], six types of mobile applications are identified by analyzing the packet size and transmission direction of the first 20 packets as input features of a hidden Markov model. In [130], a framework for fingerprinting and identification of mobile apps is presented based on DTs and SVC trained with statistical flow features grouped based on timing and destination IP address/port. In [131], the same framework is used to assess the degradation of classification performance due to changes in app fingerprints. In [132], an ensemble approach combining different state-of-the-art classifiers is proposed. Four classes of combination techniques are compared, differing in accepted classifiers' outputs, training requirements and learning scheme. Validation on a dataset of real user activity shows higher accuracy compared to individual classifiers.

App fingerprints vary significantly with time due to terminal evolution, application updates, user behavior, etc. To overcome this issue, other works propose classifiers based on deep learning, that work directly on input data by automatically distilling structured and complex feature representations at the expense of a higher training complexity [133]. In wireless networks, this approach has been considered via variational autoencoder networks [134], convolutional networks [92] or multi-modal classifiers [135] [93]. However, deep-learning classifiers present two disadvantages: a) they

exclusively consider services included in the training dataset, being unable to identify new services arising in the network, and b) they require large quantities of labeled data, which are difficult to obtain. For these reasons, the design of semi-supervised [136] or unsupervised [137] schemes is considered as a promising research direction [133]. Additionally, in the particular case of mobile networks, flow-based traffic classification requires probes that analyze traffic in the core network. In practice, operators are reluctant to install such probes because of the high associated costs. As an alternative, it is possible to process connection traces collected in the radio interface by means of BDA techniques. Such very detailed information can be used to classify traffic without investing in network probes. To the authors' knowledge, no traffic classification method based on USL over performance indicators in radio connection traces has been proposed.

This chapter presents an offline method for coarse-grained encrypted traffic classification in cellular RANs. The method relies on USL to classify traffic into broad service classes. Unlike classical approaches, based on IP traffic analysis by probes in the core network, the proposed method uses traffic descriptors from connection traces in the radio interface to perform the classification. Likewise, it can be applied in the absence of labeled data (seldom available in mobile networks) and identify new types of services launched in the network. Validation is performed over a dataset from a live LTE network. The main contributions of this work are: a) the definition of a set of connection descriptors to characterize traffic in the radio interface, and b) a method for encrypted traffic classification in the absence of labeled data based on such descriptors.

## 3.2 Problem formulation

In this section, some key concepts for the proposed classification system are explained. First, some traffic descriptors from radio connection traces are introduced. Then, the impact of data encapsulation on such traffic descriptors is analyzed for different services.

### 3.2.1 Traffic descriptors from connection traces

Among the types of radio connection traces introduced in section 2.2.2, CTRs are considered, since they comprise data from all users in the network [100]. CTRs are binary files in ASN.1 format. To compute traffic descriptors for classification purposes,

these files must be first converted into a readable format (e.g., a comma-separated values file). Each file comprises several events from users demanding services in a cell. An event includes timestamp, user identifier, cell identifier, QCI and a set of traffic parameters that differ depending on the event type. For ease of analysis, data in each file is divided per event type and synchronized. Later, user and node identifiers are used to build individual connections. A connection comprises data from a user demanding a specific service in a particular cell. Such data includes user identifier, cell identifier and a set of traffic descriptors computed from information in events.

In this work, the following traffic descriptors are computed per connection  $k$  from CTRs:

- RRC connection time,  $T_{RRC}(k)$  [ms]. A RRC connection starts when a service is requested, and lasts until the user leaves the cell, the connection is closed explicitly or the user inactivity timer expires. In many networks, such a timer has a default value of 10 s [138]. Thus, in a RRC connection of 13 s, the user may transmit during the first 3 s and the inactivity timer expires 10 s later. The connection time excluding that timer (if that is the cause of connection release) is here referred to as effective connection time,  $T_{eff}(k)$  [ms].
- Total DL traffic volume at the Packet Data Convergence Protocol (PDCP) layer,  $V_{DL}(k)$  [bytes].
- Percentage of traffic volume carried in the UL,  $\rho_{UL}(k)$  [%], computed as

$$\rho_{UL}(k) = 100 \cdot \frac{V_{UL}(k)}{V_{UL}(k) + V_{DL}(k)}. \quad (3.1)$$

- Ratio of DL traffic volume transmitted in Transmission Time Intervals (TTIs) when the transmission buffer becomes empty (a.k.a. last TTIs [12]),  $\eta_{DL}^{lastTTI}(k)$ , computed as

$$\eta_{DL}^{lastTTI}(k) = \frac{V_{DL}^{lastTTI}(k)}{V_{DL}(k)}. \quad (3.2)$$

- DL activity ratio,  $\tau_{DL}^{active}(k)$ , computed as the ratio between active TTIs (i.e., those with data to transmit) and the effective duration of the connection,

$$\tau_{DL}^{active}(k) = \frac{T_{DL}^{active}(k)}{T_{eff}(k)}. \quad (3.3)$$

- DL session throughput,  $TH_{DL}^{session}(k)$  [bps], computed as the volume transmitted

in the DL at PDCP layer divided by the effective duration of the connection,

$$TH_{DL}^{session}(k) = \frac{8 \cdot V_{DL}(k)}{T_{eff}(k)}. \quad (3.4)$$

As shown in [104], the above traffic descriptors can easily be computed per connection from information in common signaling events (e.g., connection setup, connection release, etc.). All of them are payload-independent, so they can be collected even if traffic is encrypted at the application level. Moreover, most are ratios, showing similar values regardless of encryption scheme. Nonetheless, some of these descriptors are strongly influenced by radio link and network conditions (e.g.,  $\eta_{DL}^{lastTTI}(k)$  and  $\tau_{DL}^{active}(k)$  depend on spectral efficiency, cell bandwidth, available user capacity and scheduling algorithm). Thus, connections of the same service might have different values of these descriptors. Likewise, these indicators might have similar values in connections of different services, making it difficult to isolate services. Hence, it is advisable to develop new traffic descriptors that are less dependent on network performance.

### 3.2.2 Data encapsulation process

To reduce design complexity, most networks are organized into protocol layers, each built upon the one below. As a result, data generated by applications go through an encapsulation process. Each layer adds a header and passes the data to the next layer until the lowest layer is reached, where actual communication occurs through the physical medium.

Fig. 3.1 shows an example of the encapsulation scheme in the user plane of the LTE radio interface. The upper level is the application layer, which contains application-specific protocols (e.g., Hypertext Transfer Protocol –HTTP–, File Transfer Protocol –FTP–, etc.). These protocols generate data packets of very different sizes. Below the application layer is the transport layer, which is responsible for transferring data between application peers. The primary two protocols on this layer are TCP and User Datagram Protocol (UDP). UDP is a stateless and connectionless option, providing fast, unreliable data transfer, suitable for streaming services. In contrast, TCP is stateful and connection-oriented, providing reliable transmission by guaranteeing in-order data delivery and retransmissions, suitable for web or file transfer. In both cases, application data packets are broken into smaller, more manageable pieces. The maximum size of these pieces (a.k.a. Maximum Segment Size, MSS) is usually restric-

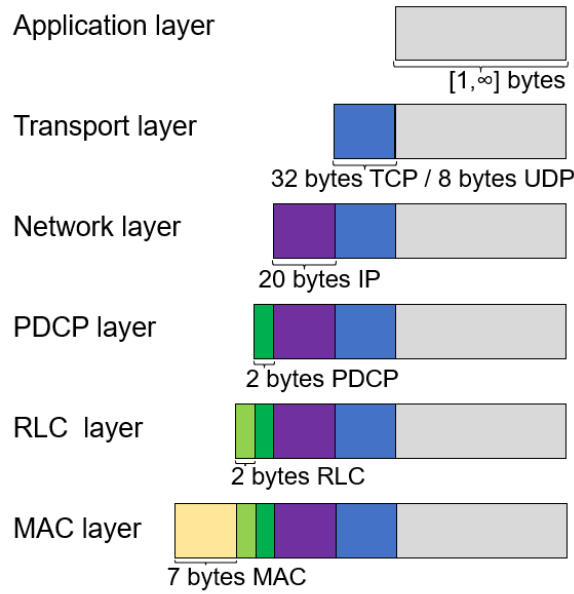


Figure 3.1: Example of packet encapsulation in the LTE user plane.

ted by the maximum transfer unit of the underlying network. In TCP, flow control uses a sliding window whose size limits how many bytes can be sent (one or more segments) until receiving an ACKnowledgment (ACK) packet. When a segment is correctly received, the receiver sends an ACK and informs about how many bytes can still be received. Below the transport layer is the network layer, responsible for connecting devices with IP protocol [139]. In the link layer, PDCP transports IP datagrams and provides header compression (if required), ciphering and integrity protection. Below PDCP, Radio Link Control (RLC) segments and concatenates PDCP packets to adapt them to the transport block size in the Medium Access Control (MAC) layer. RLC has three modes of operation: transparent mode, unacknowledged mode and acknowledge mode. The latter is often used to deliver packets through dedicated logical channels (i.e., user data traffic) [140].

The performance of the above protocols is strongly influenced by the type of service requested by the user. Different applications have distinct traffic characteristics and communication patterns. For instance, app or file downloads generate large packets, while messaging services generate infrequent small packets. To support this statement, Table 3.1 breaks down traffic descriptors at different protocol layers for four well-demanded services nowadays, namely instant messaging via WhatsApp, web browsing (on two different websites), video streaming via YouTube and app download via Google Play Store. Data in the table is obtained by analyzing traffic from live applications captured in a mobile terminal connected to a commercial LTE network. As

Table 3.1: Traffic descriptors at different protocol layers for different services in LTE.

		Measured					Theoretical
Service		Instant messaging	Web (small objects)	Web (large objects)	Video streaming	App download	Full buffer
Provider		WhatsApp	Freepik	Vimeo	YouTube	Google Play Store	–
Transport	Protocol	TCP	TCP	TCP	UDP	TCP	–
	Header [bytes]	32	32	32	8	32	32
	Max. DL payload [bytes]	147	1348	1348	1350	1348	1348
	Avg. DL packet size [bytes]	71	1139	1396	1189	1391	1348
	Max. DL packet size [bytes]	179	1380	1380	1358	1380	1380
	DL packets with MSS [%]	0	73	99	86	99	100
	No. of DL packets	27	56	2369	1988	30754	$N_p$
	No. of UL packets	30	39	1156	313	10991	$N_p$
	Ratio DL/UL packets	0.90	1.44	2.05	6.35	2.80	1
IP	Protocol header [bytes]	20	20	20	20	20	20
	Max. packet size [bytes]	199	1400	1400	1378	1400	1400
PDCP	Total DL volume [kB]	1.9	63.8	3306.2	2362.9	42770.9	$1400 \times N_p$
	Total UL volume [kB]	2.5	4.9	61.9	80.08	571.5	$52 \times N_p$
	$\rho_{UL}(k)$ [%]	56.6	7.2	1.8	3.2	1.3	3.58

expected, WhatsApp reports the lowest transport packet size, with an average packet size of 71 bytes. In fact, no packet fills the transport MSS. For the rest of services (i.e., data-hungry services), the percentage of packets that fill the transport MSS varies. In app download, video streaming and web with large objects, application data chunks are large enough to fill payload in most transport packets ( $\geq 86\%$ ). In contrast, in the case of web browsing in simple webs, only 73% of packets fill TCP payload, revealing the presence of some application data chunks with smaller size (e.g., small objects).

The different packet sizes of data-hungry services impact the value of the descriptor reflecting the ratio of UL volume,  $\rho_{UL}$ . This indicator reflects in which direction (i.e., UL, DL or both) data traffic is transmitted in a connection. Connections with  $\rho_{UL}(k)$  close to 0%/100% belong to asymmetric download/upload services, respectively, while connections with  $\rho_{UL}(k)$  close to 50% correspond to symmetric services. For download connections, the value of  $\rho_{UL}(k)$  can be approximated analytically by considering a connection with arbitrarily large application data chunks, where all transport packets are completely filled (i.e., a full-buffer service). Such an example is included in 'Full Buffer' column in Table 3.1.  $V_{DL}(k)$  at PDCP level is computed as the maximum

TCP payload (i.e., 1348 bytes in LTE, according to measurements in Table 3.1) plus 32+20 bytes of TCP and IP headers. Likewise,  $V_{UL}(k)$  is approximated by the size of an ACK packet (52 bytes). Thus,  $\rho_{UL}(k) = 100 \cdot \frac{52}{1400+52} = 3.58\%$ . Connections with  $\rho_{UL}(k)$  less than that value belong to download services characterized by large data chunks (e.g., app download). In contrast, connections with a higher  $\rho_{UL}(k)$  correspond to upload services (e.g., file upload), symmetric services (e.g., video conference) or download services with smaller data chunk size (e.g., web browsing with small objects). Such a statement is supported by measurements in Table 3.1. It is observed that Google Play Store, YouTube and Vimeo (web with large objects) show  $\rho_{UL}(k)$  below 3.58%. In contrast, the simple web shows  $\rho_{UL}(k)$  above 3.58%, and WhatsApp has  $\rho_{UL}(k) \approx 50\%$ , since it is a symmetric service.

It should be pointed out that, in the analytical bound obtained for full-buffer services, it is assumed that: a) there is no header compression in PDCP, which is valid for most data traffic in LTE [140], b) TCP protocol is used in the transport layer, and c) each TCP packet is acknowledged by an ACK. The latter assumptions are not always true in current networks. For instance, results for app download service in Table 3.1 show that 30,754 packets are sent in the DL and only 10,991 ACKs are sent in the UL (i.e., 1 UL ACK message acknowledges 2.8 DL packets on average). Likewise, YouTube sometimes uses UDP protocol [141]. If some of these conditions do not hold (e.g., there is header compression, less ACKs are sent, and/or a different transport protocol is used), a lower value of  $\rho_{UL}(k)$  will be obtained. Nonetheless, it can be stated that connections filling most transport packets cannot have  $\rho_{UL}(k)$  higher than 3.58% in LTE. This threshold will be used to isolate different types of services.

### 3.3 Classification method

This section describes the proposed traffic classification scheme. The aim of the method is: a) to divide traffic into broad application groups (e.g., messaging services, web browsing, streaming services, etc.) using information from radio connection traces provided by network operators, and b) to analyze the main features of each group.

Method structure is shown in Fig. 3.2. Once radio traces are collected and processed as explained in the previous section, the connection dataset is broken into disjoint groups by the five-step procedure illustrated in the right box of Fig. 3.2). First, a new set of traffic descriptors modeling radio connections at burst level is computed per connection and added to the dataset. It will be shown later that services offered



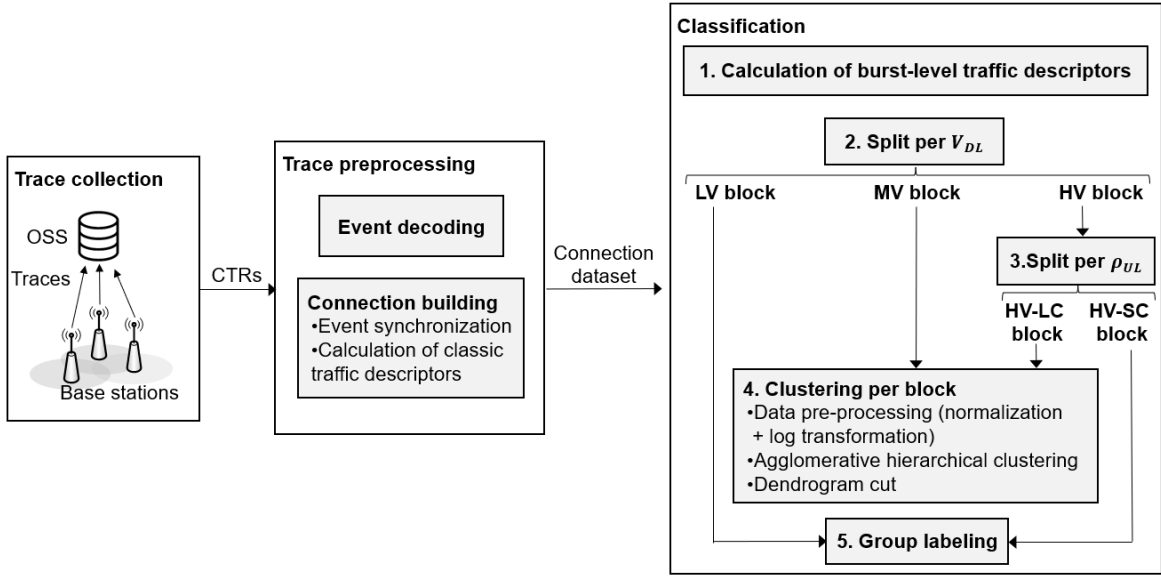


Figure 3.2: Traffic classification method.

in mobile networks are unevenly demanded (e.g., instant messaging is more demanded than file download). Performing clustering over an imbalanced dataset can lead to the classes with fewer members being shadowed by those with more members [142]. To circumvent this issue, the connection dataset is next split into broad service classes (hereafter referred to as connection blocks) based on prior knowledge on  $V_{DL}$  and  $\rho_{UL}$  descriptors. Then, connections in each block, from services with comparable demand, are divided into groups through unsupervised clustering. Finally, the obtained groups are labeled manually by analyzing their properties. A more detailed explanation of each step is given below.

### a) Computation of burst-level traffic descriptors

Traffic carried during a connection consists of one or more data chunks sent from/to the network. As explained above, chunks generated at the application layer can be segmented into smaller packets in lower layers. Then, as a result of packet scheduling, packets belonging to the same data chunk can be transmitted in several data bursts over the radio interface, where traces are collected [140]. Thus, a connection in the radio interface consists of a series of data bursts, characterized by three parameters: the number of bursts,  $N_{DL}^{burst}(k)$ , the duration per burst,  $T_{DL}^{burst}(k, n)$ , and the volume per burst,  $V_{DL}^{burst}(k, n)$  (where  $n$  denotes the burst index, since burst duration and volume may vary across bursts). Those parameters strongly depend on the service.

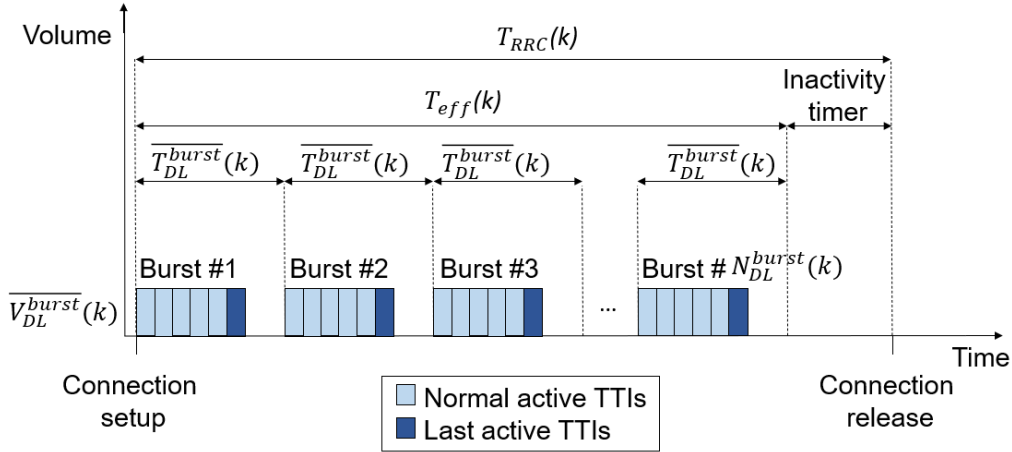


Figure 3.3: Burst-level connection model in the radio interface considering last TTIs.

For instance, when downloading a large file, a single data chunk is available at once at the application layer, so less bursts are likely to be transmitted than when downloading a web page comprising many small objects. Hence, the values of the above parameters can be used to isolate different services in the radio interface.

Unfortunately, radio connection traces do not explicitly register information at a burst level. As an alternative, burst-level parameters can be estimated per connection from the set of traffic descriptors introduced in section 3.2.1 by assuming that all bursts are equal (i.e., have the same burst volume and duration), as shown in Fig. 3.3. First, the activity ratio of a connection  $k$  in the DL,  $\tau_{DL}^{active}(k)$ , is expressed as

$$\tau_{DL}^{active}(k) = \frac{T_{DL}^{active}(k)}{T_{eff}(k)} = \frac{N_{DL}^{burst}(k) \overline{N_{burst\_DL}^{activeTTI}}(k)}{T_{eff}(k)} = \frac{\overline{N_{burst\_DL}^{activeTTI}}(k)}{\overline{T_{DL}^{burst}}(k)}, \quad (3.5)$$

where  $\overline{N_{burst\_DL}^{activeTTI}}(k)$  is the average number of active TTIs per burst in DL in connection  $k$  (e.g.,  $\overline{N_{burst\_DL}^{activeTTI}}(k)=6$  in Fig. 3.3). Likewise, by assuming that all the  $\overline{N_{burst\_DL}^{activeTTI}}(k)$  active TTIs in DL in a connection transmit the same data volume,  $V_{DL}^{TTI}(k)$ , the total data volume transmitted in last TTIs in DL in the connection can be expressed as

$$V_{DL}^{lastTTI}(k) = N_{DL}^{burst}(k) V_{DL}^{TTI}(k) = N_{DL}^{burst}(k) \frac{V_{DL}(k)}{\overline{N_{burst\_DL}^{activeTTI}}(k)} = N_{DL}^{burst}(k) \frac{V_{DL}(k)}{N_{DL}^{burst}(k) \overline{N_{burst\_DL}^{activeTTI}}(k)} = \frac{V_{DL}(k)}{\overline{N_{burst\_DL}^{activeTTI}}(k)}, \quad (3.6)$$

where it has been taken into account that there is only 1 last TTI per burst, and hence the number of last TTIs in DL in a connection is  $N_{DL}^{burst}(k)$ . Thus, the share of volume

in last TTIs is given by

$$\eta_{DL}^{lastTTI}(k) = \frac{V_{DL}^{lastTTI}(k)}{V_{DL}(k)} = \frac{\frac{V_{DL}(k)}{N_{burst\_DL}^{activeTTI}(k)}}{V_{DL}(k)} = \frac{1}{N_{burst\_DL}^{activeTTI}(k)}. \quad (3.7)$$

By replacing (3.7) in (3.5), the average burst duration can be computed as

$$\overline{T_{DL}^{burst}}(k) = \frac{1}{\tau_{DL}^{active}(k) \eta_{DL}^{lastTTI}(k)}. \quad (3.8)$$

Then, the number of bursts is estimated as

$$N_{DL}^{burst}(k) = \frac{T_{eff}(k)}{\overline{T_{DL}^{burst}}(k)}, \quad (3.9)$$

and finally the average burst size is computed as

$$\overline{V_{DL}^{burst}}(k) = \frac{V_{DL}(k)}{N_{DL}^{burst}(k)}. \quad (3.10)$$

In the above equations, it is assumed that: a) every burst has the same number of active TTIs, and b) every active TTI transmits the same volume (including last TTIs). Both statements may not be true for some connections due to changing radio conditions, TCP ramp-up or services with varying burst size (e.g., multiple objects in a web page). Nonetheless,  $N_{DL}^{burst}(k)$ ,  $\overline{T_{DL}^{burst}}(k)$  and  $\overline{V_{DL}^{burst}}(k)$  capture the general behavior of the connection, which should be enough to identify the type of service it belongs to. These descriptors, less dependent on network performance than those introduced in section 3.2.1, are computed per connection and added to the dataset.

### b) Split per DL volume ( $V_{DL}$ )

The total DL volume in a connection,  $V_{DL}(k)$ , allows to separate data-hungry services from those requiring a low bit rate. Specifically, connections can be split into three blocks:

- 1) High Volume (HV) block, comprising connections with  $V_{DL}(k) \geq 256$  kB, belonging to data-hungry services. Such a threshold is the 5<sup>th</sup> percentile of web page size in mobile version according to a comprehensive analysis of the 400 top websites in Alexa ranking [143] performed with the WebPageTest tool [144]. Moreover, such a threshold is below the size of the initial data chunk of any audio

or video in major streaming platforms [145] [146].

- 2) Medium Volume (MV) block, made of connections with  $300 \text{ B} < V_{DL}(k) < 256 \text{ kB}$ . This block contains connections from applications consuming less data. The lower threshold is the minimum data volume exchanged by applications providing instant messaging service (Telegram, Viber, etc.), which is the less data-demanding of the most popular services in current mobile networks [147]. Such a threshold is also the maximum size of push notifications used by mobile applications to inform users of new events and updates [148].
- 3) Low volume (LV) block, comprising connections with  $V_{DL}(k) \leq 300 \text{ B}$ . This block contains traffic from signaling or push-up notifications.

### c) Split per transport segment size ( $\rho_{UL}$ )

Different data-hungry services have different data chunk size at the application layer. As explained in section 3.2.2, such behavior has an impact on the UL/DL volume ratio. Thus, the share of UL volume,  $\rho_{UL}(k)$ , can be used to split connections from HV block in two sub-blocks: a) HV-LC block, comprising connections with Heavy data Volume and Large data Chunks that tend to make the most of payload size at the transport layer, and b) HV-SC, comprising connections with Heavy data Volume and some Small data Chunks that may not fill transport packets. In section 3.2.2,  $\rho_{UL}(k) \approx 3\%$  was computed as an upper bound for the former services.

### d) Clustering per block

Connections in LV block consist of signaling and notifications, often neglected in network dimensioning and service-oriented self-management tools. Likewise, HV-SC block is expected to include a mix of services whose traffic patterns are not distinguishable by information in traces. However, a more fine-grained classification can be performed over connections in MV and HV-LC blocks. For this purpose, Agglomerative Hierarchical Clustering (AHC) is used. Among the existing clustering algorithms, AHC has been selected because: a) it can manage datasets with clusters of different sizes (remember that, in mobile networks, services are unevenly demanded) and density (connections from a service type can have very similar traffic descriptors or not), b) it does not require to specify the number of clusters in advance (in the considered problem, such information is unknown), and c) the dendrogram itself is valuable to understand the data.

Most clustering algorithms do not work effectively in high dimensional space due to the so-called *curse of dimensionality* [149]. Moreover, in clustering algorithms based on distance such as AHC, as the number of input features grows, the distances among datapoints become all approximately equal, and no meaningful clusters can be formed [150]. To avoid these undesirable effects, a reduced subset of the considered traffic descriptors is used as input features to AHC. Ideally, the selected traffic descriptors must fulfill that: a) they take different values for different services, b) they are insensitive to network conditions, and c) they do not provide redundant information. A preliminary analysis of traffic descriptors (not shown here for brevity) reveals that the subset comprising  $T_{RRC}(k)$ ,  $\overline{V_{DL}^{burst}(k)}$  and  $N_{DL}^{burst}(k)$  fulfills these criteria. Then, only these 3 traffic descriptors are used as input features to AHC.

AHC assumes normally distributed data. A log transformation is performed over the 3 input features to reduce data skewness. Moreover, traffic descriptors show very different ranges of values, which is an issue for distance-based clustering algorithms. For better performance, log-transformed data is normalized, so all input features are comparable. For this purpose, a min-max scaling method is used [151]. The normalized value of each descriptor  $f$  in datapoint  $d$ ,  $f_{\text{norm}}(d)$ , is computed as

$$f_{\text{norm}}(d) = \frac{f(d) - f_{\min}}{f_{\max} - f_{\min}}, \quad (3.11)$$

where  $f(d)$  is the value of the descriptor after log-transformation and  $f_{\max}$  and  $f_{\min}$  are the maximum and minimum values of the descriptor in the dataset. Normalization must be performed separately in each block of connections (i.e., MV and HV-LC).

For robustness, the optimal point to cut the dendrogram (i.e., the best number of clusters,  $N_{\text{clust}}$ ) is found per block by checking the average silhouette score [152] and the Calinski–Harabasz (CH) score [153] across a wide range of cut points. Silhouette score assigns a mark between -1 and 1 to each sample in the dataset. Positive values show that a datapoint is well classified, whereas negative values indicate that the datapoint is more similar to a different cluster. In contrast, CH score computes the ratio between the within-cluster dispersion and the between-cluster dispersion. In both cases, the higher value, the better.

### e) Group labeling

Finally, the services included in each group are deduced by manually analyzing the median value of traffic descriptors for connections in the group.

## 3.4 Performance assessment

The proposed classification method is validated using connection traces from a live LTE network. For clarity, assessment methodology is first described, results are presented later and computational complexity is finally discussed.

### 3.4.1 Assessment methodology

The dataset is generated from CTRs collected from 10 am to 11 am (busy hour) in 145 cells covering 125 km<sup>2</sup> in an urban area of a live LTE network. This data should be representative of the entire network traffic because: a) the time period represents a significant share of daily network traffic, and b) the area includes financial, residential and recreational districts with different user profiles, which should reduce the influence of time of day. Table 3.2 presents trace events provided by the vendor used to compute all the considered traffic descriptors.

Event decoding is performed by a proprietary tool provided by the network operator, and then connection building is carried out in Java for computational efficiency. The resulting dataset consists of 184,349 connections. It is expected that most traffic is encrypted by the time the dataset was collected based on reports published by popular content providers (e.g., Google [154]). As a consequence, QCI is the only information available regarding service type. The dataset comprises 11.5% of connections with QCI 1 (Voice over LTE, VoLTE), 0.1% with QCI 5 (IP Multimedia Subsystem signaling) and 88.4% with QCIs from 6 to 9 (multimedia and TCP-based services). The latter class, comprising 162,965 connections, is divided into application groups by the proposed classification method. Such a method, hereafter referred to as Enhanced Agglomerative Hierarchical Clustering (E-AHC), is compared with a naive method, referred to as Basic Agglomerative Hierarchical Clustering (B-AHC). In B-AHC, AHC is directly applied over all connections with QCIs from 6 to 9 (i.e., without any previous split per  $V_{DL}(k)$  or  $\rho_{UL}(k)$ ). This approach, considered a benchmark, may be taken by a data scientist with no prior knowledge of mobile networks.

Table 3.2: Events in connection traces used for traffic classification.

Event name	Description
INTERNAL_PROC_INITIAL_CTXT_SETUP	Event reporting connection start time
INTERNAL_PROC_UE_CTXT_RELEASE	Event reporting connection release time and cause
INTERNAL_PER_UE_TRAFFIC_REP	Periodic event reporting the active number of TTIs in both UL and DL
INTERNAL_PER_UE_RB_TRAFFIC_REP	Periodic event with total data volume in UL and DL and data volume transmitted in last TTIs

AHC is implemented with the *Cluster Analysis* toolbox in Matlab [155]. In both B-AHC and E-AHC, a *ward* linkage function is used, which minimizes the total within-cluster variance by merging the pair of clusters with minimum between-cluster distance at each step. The Euclidean distance is used as distance metric [156]. In the absence of labeled data, which would require using network probes, the method is validated by checking that the groups created are consistent with the typical mobile traffic mix reported by a vendor the year when traces were collected [157].

### 3.4.2 Results

Results are presented next, broken down per classification algorithm for clarity.

#### a) B-AHC

Fig. 3.4 shows the average silhouette and CH scores obtained with the classical B-AHC for different cuts in the dendrogram (i.e.,  $N_{clust}$  choices). For better visualization, values for each indicator are normalized by their maximum value. It is observed that, in general, the value of both metrics tends to decrease as the number of clusters increases. The higher (i.e., the best) value of CH index is obtained when  $N_{clust}=4$ , whereas the silhouette value for this set-up is close to the best value (i.e., the relative value is 0.86). Thus, the connection dataset is split by AHC into 4 service groups.

Table 3.3 breaks down the properties of groups (clusters) obtained by for B-AHC with  $N_{clust}=4$ . For each group, the following information is provided: a) the number of connections, b) the median value of traffic descriptors of connections in the group<sup>1</sup>,

<sup>1</sup>The median operation is expressed by eliminating the dependence of  $k$  (e.g.,  $V_{DL}$  is the median of  $V_{DL}(k)$  for all connections  $k$  in a group)

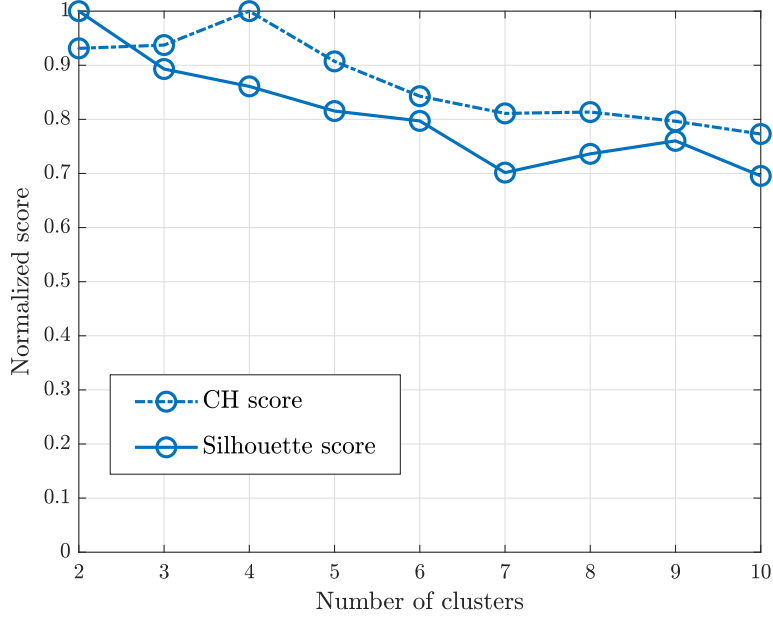


Figure 3.4: B-AHC performance with different number of clusters.

Table 3.3: Groups in B-AHC method.

Group	Group 1	Group 2	Group 3	Group 4
No. of connections	35488	55224	51782	20471
$T_{RRC}$ [ms]	10618	10537	14148	28460
$V_{DL}$ [bytes]	211	288	6111	243493
$\rho_{UL}$ [%]	65.1	47.3	36.5	10.7
$\eta_{DL}^{lastTTI}$	1	1	1	0.33
$\tau_{DL}^{active}$	0.024	0.023	0.012	0.019
$TH_{DL}^{session}$ [kbps]	2.27	3.41	13.97	132.04
$N_{DL}^{burst}$	7	9	35	83
$\overline{T}_{DL}^{burst}$ [ms]	132	80	191	114
$\overline{V}_{DL}^{burst}$ [bytes]	36	32	173	2925
% of total DL volume	0.13	0.06	1.29	98.52

and c) the percentage of DL volume carried by connections in the group. Results show that connections in groups 1 and 2 present very similar characteristics (short connections with reduced volume transmitted in last TTIs). Thus, all these connections should have been grouped into a single cluster. Moreover, group 4, comprising long data-intensive connections, has 98.52% of the total carried traffic in the DL. According to [157], no service had such an amount of traffic by the time the dataset was collected (nor currently). The large number of connections in this group (e.g., 12.56% of the total) suggests that it contains connections from several data-hungry services. These



inconsistencies point out that, as expected, AHC is not performing well because the number of connections in some services is vast, causing that clustering focuses only on that particular service. To confirm that bad results are not due to the choice of AHC as clustering algorithm, the experiment is repeated with k-means and DBSCAN [158], obtaining similar performance.

The above shortcomings are solved by the proposed E-AHC method by dividing the dataset into blocks of connections based on prior knowledge.

## b) E-AHC

In E-AHC, the dataset is first divided into three blocks based on connection data volume in the DL (LV, MV and HV blocks). This split results in MV block (medium volume) with the highest number of connections (104,227 connections, 63.99% of the total), LV block (low volume) with 48,615 connections (31.69% of the total) and HV block (high volume) with the lowest number of connections (7,032, a 4.32% of the total). Then, the latter block is split according to  $\rho_{UL}(k)$  value in two sub-blocks: HV-SC (small data chunks), comprising 7,032 connections, and HV-LC (large data chunks), with only 3,091 connections. Finally, MV and HV-LC sub-blocks are divided into clusters by means of AHC.

Fig. 3.5 shows the relative average silhouette and CH scores obtained when cutting the dendrograms of MV block (blue curves) and HV-LC block (orange curves) with different numbers of clusters. The best number of clusters is supposed to provide the highest score values. Following this rationale, the optimal point should be  $N_{clust}=2$  for both MV and HV-LC blocks. However, an analysis of within-cluster sum of distances (not shown here) reveals that, in both cases, this solution creates non-compact clusters, providing a too coarse classification. Thus,  $N_{clust}=2$  is discarded. For  $N_{clust}=4$ , CH score in MV block has a value of less than 0.6 compared to the maximum, which is unacceptable. Similarly, in HV-LC block, individual silhouette scores per datapoint (not shown here) reveal that the number of samples with a negative silhouette score value (i.e., with very different features to the typical behavior of the corresponding cluster) strongly increases when  $N_{clust}=4$ , which is undesirable. In both cases, a larger number of clusters leads to worse performance. Therefore,  $N_{clust}=3$  is selected as the cut point for both MV and HV-LC blocks. This solution provides a trade-off between cluster compactness and acceptable CH and silhouette scores.

Table 3.4 presents the eight connection groups obtained at the end of the classi-

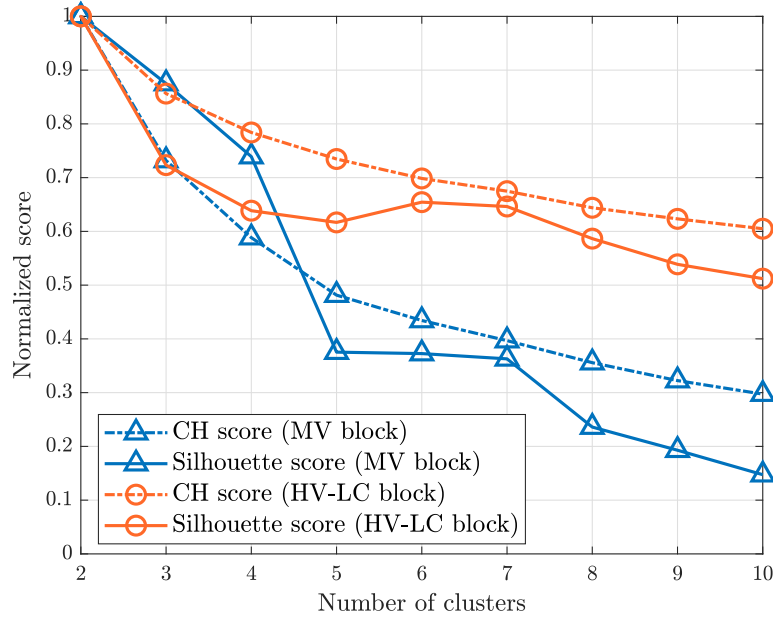


Figure 3.5: E-AHC performance with different number of clusters.

Table 3.4: Groups in E-AHC method.

Block	LV	MV			HV-LC			HV-SC
Group	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8
No. of connections	48615	52798	37624	13805	834	1205	1052	7032
$T_{RRC}$ [ms]	10458	11248	17890	12337	62555	18279	21404	46335
$V_{DL}$ [bytes]	144	797	11220	8026	$11.96 \times 10^6$	$1181 \times 10^6$	$2.11 \times 10^6$	97390
$\rho_{UL}$ [%]	5	48.9	35.5	25.4	2.3	2.4	2.3	7.2
$\eta_{DL}^{lastTTI}$	1	1	1	0.40	0.24	0.03	0.17	0.23
$\tau_{DL}^{active}$	0.026	0.017	0.01	0.02	0.092	0.104	0.067	0.023
$TH_{DL}^{session}$ [kbps]	2.25	5.44	16.46	37.73	2146.53	2319.8	1480.3	251.1
$N_{DL}^{burst}$	4	18	54	18	820	24	136	165
$T_{DL}^{burst}$ [ms]	127	67	125	144	57	363	80	203
$V_{DL}^{burst}$ [bytes]	33	40	199	496	12793	84223	16305	6487
% of DL volume	0.02	0.15	2.57	0.74	41.81	9.21	8.33	37.17
Service	Push notifications	Instant messaging	Instant messaging	File sharing	Streaming	Full buffer services	Web browsing	Web browsing and RSS

fication process (one cluster from LV group, three clusters from MV group and four clusters from HV group). For each group, it breaks down: a) the block to which the group belongs, b) the number of connections, c) the median value of traffic descriptors of connections in the group, d) the percentage of the total DL volume carried by connections in the group and e) the underlying service, guessed by analyzing such values. Groups are analyzed next.

LV block makes up group 1, comprising very short connections ( $T_{RRC} < 11$  s and, thus,  $T_{eff} < 1$  s, since default inactivity timer is 10 s) with few data ( $\approx 150$  B in both

UL and DL), all transmitted in last TTIs ( $\eta_{DL}^{lastTTI}=1$ ). Due to the low transmitted data volume, session throughput is very low ( $\approx 2$  kbps). Such a description fits with push notifications, consisting of lightweight audio or visual cues sent by specific servers (e.g., Google Cloud Messaging Server) to inform users about unread messages or updates in applications [148]. This group may also include some radio connections comprising only a TCP FIN or RESET packet, appearing when these packets are delayed more than the user inactivity timer [138]. In this case, a TCP connection is split into two connections over the radio interface (one with the main TCP data flow and another only with the FIN or RESET message). Note that this group is the second largest in the mobile network under analysis, comprising approximately 30% of connections in the dataset.

MV block is split into groups 2 to 4. Group 2 presents the highest number of connections (about 33% of the total) with a short RRC connection time ( $\approx 11$  s), low data volume ( $\approx 800$  B) and 100% of data transmitted in last TTIs. The fact that  $T_{RRC}$  is very close to the inactivity timer suggests that these connections consist of a single data chunk at the application layer. Moreover,  $\rho_{UL}=49\%$ , revealing that connections belong to a symmetric service, i.e., users send and receive data. All these characteristics can be associated to instant messaging services (e.g., WhatsApp) [147].

Group 3 has fewer connections than group 2 (23% of the total) with a longer duration ( $\approx 8$  s without considering the inactivity timer) and a higher but still limited data volume ( $\approx 11$  kB). The fact that data is transmitted in last TTIs and the extremely low activity ratio in the DL (median of 1%) show that data consists of small data chunks scattered in time (in fact,  $N_{DL}^{burst}=54$ ). Since  $\rho_{UL}=35\%$ , a considerable amount of the total data is transmitted in the UL. Thus, these connections are likely due to several interactions between user and network. This behavior is also typical of instant messaging services, where several messages are received/sent before the inactivity timer expires and thus all those messages are part of the same radio connection. Note that connections in groups 2 and 3 make up 56% of samples in the dataset, which is consistent with the fact that instant messaging services are the most demanded services in mobile networks nowadays [159].

Connections in group 4 are shorter than those of group 3 ( $T_{RRC}=12.3$  s), with similar DL volume (8 kB) but lower UL volume ratio ( $\approx 25\%$ ). The average burst volume is much higher than in group 3 ( $\overline{V_{DL}^{burst}}=199$  B and 496 B in groups 3 and 4, respectively), showing an increase in data chunk length. In fact, only 40% of data is transmitted in last TTIs. This group may be associated with small data files (e.g., images, audio

recordings, documents, etc.) commonly shared by e-mail, messaging applications or social networks.

HV-LC block, comprising data-hungry services with large data chunks at the application layer (i.e.,  $\eta_{DL} < 3\%$ ), is divided into groups 5 to 7. Group 5 presents the lowest number of connections in the dataset (0.05% of the total) with the longest length ( $T_{RRC} \approx 62$  s) and the highest DL data volume (12 MB), which is transmitted in many bursts. In fact, despite the reduced number of connections, this group accounts for 41.81% of the total download traffic in the network. The large duration and DL volume and the presence of bursty traffic suggest that this group includes connections from audio and video streaming applications (e.g., YouTube, Netflix, Spotify, etc.). It is worth noting that the median value of  $TH_{session}^{DL}$  in this group (2146.53 kbps) is higher than expected (150 kbps is approximately the rate of high-definition video [160]). It should be pointed out that, at the initial phase of a streaming session, a significant part of the video/audio file (e.g., 40 s) is downloaded at full speed to avoid rebuffering events. Then, download speed decreases, approaching the playout rate [145] [146]. Thus,  $TH_{session}^{DL}$  for short videos can be considerably higher than the playout rate. A deeper analysis of data reveals that  $TH_{session}^{DL}$  for connections in this group tends to decrease as  $T_{RRC}$  increases, which is consistent with the fact that, in longer videos, download speed tends to playout rate.

Groups 6 and 7 comprise shorter connections than group 5 ( $T_{RRC} \approx 20$  s) with lower  $V_{DL}$  ( $\approx 2$  MB). The new burst traffic descriptors reveal that, for connections in group 6, data is transmitted in a few very long bursts over the air interface (the heaviest in the dataset). As a consequence, the activity ratio in the DL and session throughput are the highest (10.4% and 2.3 Mbps, respectively). These features fit with full-buffer services, such as app download, software update or large file download via FTP, where the user demands as many resources as possible until all the data is transmitted. In contrast, group 7 comprises connections with a large number of bursts ( $N_{DL}^{burst} = 136$  in group 7, compared to 24 in group 6) and lower DL activity ratio (6.7%) and session throughput ( $\approx 1.48$  Mbps). The higher ratio of last TTIs (0.17 in group 7, compared to 0.03 in group 6) points out the presence of small data bursts, which is confirmed by the lower  $\overline{V_{DL}^{burst}}$  (6.5 kB in group 7 vs. 16.3 kB in group 6). Because of the presence of bursts with different sizes and the median value of  $V_{DL}$ , very similar to the median size of mobile web pages in Alexa ranking [161], this group is labeled as web browsing.

Finally, HV-SC block corresponds to group 8. Since  $\eta_{DL}^{lastTTI} = 0.23$ , it is deduced that connections in this group have medium size data chunks. The median value of

Table 3.5: Share of DL traffic volume.

Service	Vendor report	Proposed method
Streaming	54.8 %	41.8 %
Web browsing	5.9 %	8.3 %
Full-buffer services	6.8 %	9.2 %
Social networks & others	32.6 %	40.7 %

$T_{RRC}$  is 46 s. The reduced DL activity ratio (2.3%) and the low session throughput ( $\approx 250$  kbps) indicate that such a duration is due to several user interactions. This group may contain a mix of services, such as web browsing (e.g., web with many small objects or multi-page sessions) or social networks, where a wide range of services (e.g., instant messaging, file sharing, short video streaming, etc.) can be demanded in a single connection.

In the absence of labeled data, the classification shown in Table 3.4 is validated by comparing the results with mobile traffic statistics published by a vendor [157]. Table 3.5 shows the percentage of traffic per application type carried worldwide in 2016 [157] (i.e., when traces were collected) and that obtained by E-AHC. According to [157], audio/video streaming services carry most of the traffic (54.8%) in current networks. This figure is consistent with results from E-AHC, which ascribe 41.8% of traffic to these services (group 5). In [157], 5.9% of traffic is assigned to web browsing, whereas the proposed classification system assigns 8.3% of traffic to this service (group 7). Software update, application download and file sharing services comprise 6.8% of traffic in [157], compared to the 9.2% of traffic assigned to full-buffer services (group 6) by E-AHC. Finally, [157] includes two groups called *Social Networks* and *Others* carrying 32.6% of traffic. Both groups include traffic of a different nature (e.g., instant messaging, short videos, small file sharing, etc.), equivalent to groups 1, 2, 3, 4 and 8 in A-EHC, carrying 40.7% of volume in the DL. Nonetheless, note that the classification performed here is based on traces from a particular network, and percentages may slightly differ from those reported worldwide by the vendor.

### 3.4.3 Computational complexity

The proposed classification system is conceived to be executed offline, and thus computational efficiency is not critical. The only tasks that must be performed manually by an expert are dendrogram cutting and group labeling. Once service groups in the

network have been identified, there are two options to classify new datapoints (i.e., connections): a) compute cluster centroids and assign every connection to the cluster with the nearest centroid, or b) train a SL-based classifier with a dataset labeled with the proposed system. No matter the selected option, the most time-consuming task for model exploitation is preprocessing connection traces. For time-constrained applications, this task can be accelerated by using parallelization.

The whole process must be repeated: a) when a new service is launched in the network (such an event may require adding a new service group) and b) periodically, to capture significant changes in network protocols or user behavior affecting the value of the considered traffic descriptors (as such an event may change group centroids or require updating  $V_{DL}$  or  $\rho_{UL}$  thresholds used to create connection blocks). In current mobile networks, these events take place at most with a monthly resolution. Therefore, performing a new service classification (which takes at most several hours) should not entail a problem for MNOs.

### 3.5 Conclusions

This chapter has dealt with the problem of classifying connections per service type in mobile networks, which is a key task for the proliferation of service-oriented NFs. A novel scheme for coarse-grained encrypted traffic classification has been proposed. Unlike previous flow-based approaches, relying on expensive traffic probes in the core network, the proposed method is based on traffic descriptors computed from connection traces collected on the air interface. To avoid the influence of network conditions, a new set of network-independent indicators characterizing connections at burst level has been developed. Since the model relies on USL, namely agglomerative hierarchical clustering, it can be applied in the absence of labeled data.

Validation has been performed with a dataset from a live LTE network. Results have shown the potential of burst-level traffic descriptors to cluster connections per service type. Nonetheless, even with the adequate set of input features, unsupervised clustering algorithms perform poorly when applied directly over all connections in CTRs due to the uneven demand of services in mobile networks, where some services (e.g., instant messaging) prevail over others. To circumvent this problem, it is essential to exploit prior knowledge to create broad connections blocks, and then apply USL separately over each block for a finer-grained clustering. The classification performed by the proposed method is consistent with the traffic share reported for live networks

the year data was collected, supporting the reliability of results.

The proposed classification scheme can easily be extended to other RATs and is especially suitable for 5G networks, where highly differing services coexist and hence the development of service-oriented NFs is key to warrant customer satisfaction.



UNIVERSIDAD  
DE MÁLAGA



# Chapter 4

## Throughput estimation in cellular radio access networks

This chapter addresses the issue of estimating radio throughput indicators in HSDPA, LTE and upcoming sliced RANs by applying SL over data collected in the OSS. In all these networks, cell-level performance estimates are key for detecting and solving capacity problems in the RAN. Likewise, slice-level performance estimates are required for slice (re)dimensioning purposes in NS scenarios. Among existing modeling approaches, using SL over network data is a promising solution to derive performance models tailored to specific network peculiarities (e.g., architecture, Radio Resource Management –RRM– algorithms or NS set-up).

Content in this chapter is organized as follows. Section [4.1](#) revises related work. Section [4.2](#) formulates the problem of estimating radio throughput indicators from statistical measurements. Section [4.3](#) details the proposed generic estimation methodology, that can be applied to different RATs. Then, section [4.4](#) presents method assessment when estimating DL cell and user throughput during busy hours in HSDPA and LTE networks. Finally, section [4.5](#) extends the analysis to DL cell and slice throughput estimation in sliced RANs.

### 4.1 Related work

Performance estimation aims to predict some KPI at a given time from other known or predicted information about network state at that time (unlike performance forecasting,

where predictors are taken from a different time instant). Estimating cell capacity is key for RAN dimensioning tasks, and has therefore been extensively covered in the literature. Different metrics have been considered as capacity indicators. In [162], the authors present an admission control policy driven by an analytical model based on a multidimensional continuous-time Markov chain to estimate the varying capacity of cells in LTE caused by user mobility in terms of session blocking probability. In [163], the available bandwidth (i.e., channel spare capacity) is estimated from measurements taken in drive tests in MONROE 3G/4G testbed, and the relationship between available bandwidth and achievable throughput is analyzed. The MONROE platform is also used in [164] to characterize cell capacity offered by 11 operators in 4 different countries, measured as maximum throughput at the application layer. In [165], cell capacity for VoLTE service is measured as the maximum number of simultaneous active users that can be served by a cell. Then, an analytical model is proposed to estimate cell capacity in cell areas where users report different channel quality information. In [166], a model based on linear regression is proposed to measure the maximum allowed traffic in Erlangs in a multi-service HSDPA network for different transmit powers and QoS requirements from network performance indicators.

A common approach is to measure cell capacity as DL cell throughput in high load conditions. Several analytical models have been developed to estimate cell throughput considering different MIMO antenna schemes [167], scheduling algorithms [168] and traffic classes [169]. However, cell throughput is highly dependent on multiple factors, such as service mix, terminal capabilities or propagation environment, which change with time and location. To deal with this diversity, some studies estimate cell throughput via simulations [170] [171]. Nonetheless, it is virtually impossible to simulate all possible combinations of the above-mentioned factors. Alternatively, some works propose models tuned with real network statistics (e.g., CM, PM, traces...) collected in the OSS. An interesting approach is to estimate cell performance with regression models based on SL, able to capture the peculiarities of each particular network (e.g., packet scheduler). The earliest works rely on Multi-Variable Linear Regression (MLR). In [22], a performance model based on MLR is derived to estimate DL cell throughput in the busy hour in a live HSDPA network from code-related, quality-related and power-related indicators computed from PMs and CMs collected on a cell basis. In [23], it is shown with real data that MLR can estimate cell throughput reasonably well in a multi-service LTE network, but not packet delay statistics of VoIP users. In [24], delay in connection setup is also considered an input to the linear model.

As explained in chapter 2, current radio planning tools should not rely only on metrics reflecting aggregated cell performance but also consider user performance metrics. In [172], SL algorithms are applied over data collected in a crowdsourced speed test to estimate DL user throughput. Such tests collect terminal- and network-related data through a large number of over-the-air transmissions. As a consequence, they can overload the radio interface and drain user limited data plans, which is undesirable. Alternatively, in [173], an analytical model is proposed to estimate DL user throughput using drive test data collected by a radio frequency scanner. However, drive tests are time-consuming and imply high operational costs, since they must be performed periodically to adapt to events in the area or in the network affecting radio frequency measurements (e.g., new building or new cells, respectively) [174]. In [12], an analytical performance model is presented to estimate radio user throughput for packet scheduling purposes in a multi-service scenario. Model parameters are adjusted with information from radio connection traces. A more efficient approach for operators is to estimate both aggregated cell throughput (hereafter, cell throughput) and average user throughput per cell in the DL from the same set of cell-level measurements gathered in the OSS during normal network operation. Unlike cell performance, user performance may not be linearly related to cell-level indicators, suggesting the use of non-linear SL algorithms. For instance, in [25], DL cell and user throughput in LTE are estimated with a DNN from a labeled dataset. The authors consider a set of 13 CMs and PMs collected in a live network hourly for two months to train the model. However, network operators are reluctant to use complex deep learning models with thousands of hyperparameters in their network management tools, since these models are difficult to configure and interpret and must be trained with extensive training datasets (tens of thousands of samples) to avoid overfitting. Under this premise, it is appropriate to check if simpler classical SL algorithms perform well for the tackled problem.

This chapter presents a comprehensive analysis comparing the performance of well-known SL algorithms for DL cell/user throughput estimation in busy hours from cell-level PMs/CMs collected in the OSS. Two different RATs are considered, namely HSDPA and LTE. For this purpose, two datasets with the most relevant performance indicators in each RAT are collected from live networks. The main contributions of this analysis are:

- a) Presenting the first comparison of non-deep SL schemes for estimating DL cell/user throughput in busy hours from network measurements in LTE. The considered approaches include RF, MLP, SVR and KNN. These algorithms are compared

with DNN and MLR techniques proposed in previous works [23] [25] [24].

- b) Extending the analysis to HSDPA, where previous works only covered DL cell throughput estimation with MLR [22].
- c) Identifying a minimal set of key network performance indicators to be stored in the OSS to estimate throughput indicators in both technologies.

In 5G systems, slicing the RAN implies significant changes that, as will be shown later in this chapter, alter the correlation between network indicators and throughput. Thus, estimating DL cell throughput in NS scenarios requires a separate analysis. Moreover, new NFs (e.g., capacity brokers) arise that require slice-level performance estimates to guarantee SLA fulfillment while ensuring an efficient use of system bandwidth. In [175], an analytical model is presented to estimate user blocking probability in a cell serving guaranteed-bit-rate slices from channel quality information. The model is based on a multi-dimensional Erlang-B system, insensitive to session duration distribution. An analytical approach is also considered in [176] to estimate the required capacity per slice on a cell and pixel basis for redimensioning purposes. The model is fed with cell configuration, channel quality information and traffic information (i.e., spatial distribution and volume) per active slice. For slices serving eMBB traffic, throughput is often the most highly-demanding performance requirement among those included in SLAs while strongly impacting user experience. As a consequence, the development of slice-level models to estimate indicators such as the aggregated slice throughput per cell in the DL (hereafter, DL slice throughput) has gained interest for MNOs.

The ability of ML to capture network peculiarities is key when managing complex sliced RANs. As a consequence, ML-based solutions have been proposed for resource split among slices [117] [177], slice admission control [117], user-centric slice design [178] or slice classification per service type [179], among other tasks. Closer to this thesis, in [180], SL is used to estimate application-level video requirements from low-layer network measurements to improve the slice negotiation phase. In [181], a digital twin network model relying on graph ANNs is used to predict end-to-end packet latency in three different NS scenarios, capturing intertwined relationships among slices. However, the performance of SL models to estimate cell or slice throughput in sliced RANs has not been assessed yet.

This chapter also presents an analysis of the performance of well-known SL algorithms to estimate DL cell and slice throughput in sliced RANs from data in the OSS. The considered approaches comprise all the algorithms tested in the analysis for

non-sliced RANs (except MLR), plus AdaBoost and XGBoost. The analysis focuses on eMBB services, for which throughput is key to ensure user satisfaction. For this purpose, synthetic measurement datasets have been created with a dynamic system-level simulator emulating the activity of a live cellular network. The main contributions are:

- a) Presenting the first study assessing the performance of well-known SL models to estimate DL slice throughput in sliced RANs from data in the OSS.
- b) Assessing the performance of these algorithms to estimate also DL cell throughput in sliced RANs. To justify the need for this contribution, an analysis of the impact of enabling NS on the correlation between network indicators and DL cell throughput is presented.
- c) Extending the comparative analysis to two different NS scenarios, consisting of single-service and multi-service slices serving eMBB users.
- d) Identifying a minimal set of network performance indicators to be stored in the OSS for the above tasks. A novelty here is the inclusion of features derived from radio connection traces, not considered in the analysis performed over non-sliced network.

It should be pointed out that all SL techniques considered here are included in most data analytics packages and have already been used in several fields. Hence, the main novelty is the assessment of well-established SL methods for new use cases related to radio throughput estimation in cellular networks.

## 4.2 Problem formulation

This chapter tackles the problem of developing models to estimate radio throughput indicators at a given time  $t$  from information (real or hypothetical) on network state at time  $t$ . These models are key for an efficient network (re)dimensioning since they allow: a) to analyze a worst-case scenario for the current network set-up, and b) to assess the impact of redimensioning actions (e.g., cell/slice bandwidth extension/reduction, deployment or temporal switch-off of a cell...) on network performance. To capture network peculiarities, models are built from data gathered in the OSS.

The estimation of DL throughput of an entity  $k$  in the RAN of a cellular network,  $TH(k)$ , from data collected in the OSS can be tackled as a regression problem. Throughput depends on many factors related to radio channel conditions (e.g., in-

door/outdoor environment, inter-site distance...), network configuration (e.g., packet scheduling algorithm, radio resource utilization threshold...) and user profile (e.g., traffic mix, terminal capabilities...). As a consequence, complex regression models with dozens of predictors can be derived, some of which may be RAT-specific features (e.g., code-related features in HSDPA). In the simplest model, DL throughput estimation is formulated as

$$\widehat{TH}(k) = f(C(k), C_{util}(k), SE(k)), \quad (4.1)$$

where  $C(k)$  denotes the capacity of entity  $k$ ,  $C_{util}(k)$  is the amount of used capacity, and  $SE(k)$  is spectral efficiency, reflecting how much data can be transmitted per capacity unit with radio link conditions experienced in the DL of entity  $k$ .

The activation of NS feature can have a strong impact on network performance. Cell bandwidth, Physical Resource Block (PRB) utilization ratio in the Physical DL Shared CHannel (PDSCH) and Channel Quality Indicator (CQI) statistics are often considered as capacity, used capacity and spectral efficiency indicators, respectively, when estimating DL cell throughput. In live networks, cell bandwidth determines the maximum achievable cell throughput, whether NS is enabled or not. In legacy networks without NS, all users share the spectrum, leading to a high PRB utilization ratio in peak periods in the presence of users demanding data-hungry services. As a consequence, for a specific cell bandwidth, DL cell throughput strongly depends on spectral efficiency (i.e., CQI reported by the UE), which determines the amount of bits that can be transmitted per PRB. In contrast, in NS scenarios, the split of radio resources among slices may prevent the packet scheduler to make the most of cell bandwidth. If this is the case, the PRB utilization ratio becomes a relevant indicator to estimate DL cell throughput. These differences suggest that separate cell-level performance models must be derived for non-sliced and sliced scenarios.

Apart from cell-level models, in sliced RANs, some advanced RRM tasks require estimating system performance per slice. The model in (4.1) can be extended to estimate DL throughput per cell and slice by defining slice-level inputs (e.g.,  $C(k)$  may be the average no. of PRBs allocated to an slice per cell in the considered ROP). Note that slices can either only serve a type of service (e.g., video streaming slice) or a mix of services (e.g., all traffic belonging to a virtual MNO). To make it easier for operators to select the best slice set-up, a comprehensive analysis must be carried out to check the impact of service mix when estimating slice performance by comparing results from

scenarios with single-service and multi-service slices.

In 5G systems, the increase of bursty data from services with small packet size (e.g., mMTC services) may alter the correlation between network performance indicators (e.g., number of simultaneous users) and DL slice throughput. Previous cell performance models have been developed using data from live 3G or 4G networks, where most connections belong to data-hungry services [182]. To deal with service diversity, it could be necessary to include features reflecting the traffic mix in network performance models at cell and slice level.

Note that, in the RAN, several important aspects affecting slice definition are up to MNOs. For instance, radio resource split among slices might be hard (i.e., dedicated radio resources per slice) or flexible (i.e., slices share radio resources). Moreover, in the latter case, resource allocation per slice can be static or dynamic (e.g., slice resource reallocation every minute). Different NS settings lead to different radio-electrical and traffic isolation between slices [116]<sup>1</sup>. Likewise, RRM algorithms (e.g., access control or packet scheduling) can be customized per tenant, leading to different slice behavior. More importantly, even in the absence of NS, each mobile network has its own peculiarities (e.g., topology or RRM algorithms). In this context, empirical SL models can capture non-linear relationships among features and peculiarities of each specific scenario when estimating cell or slice performance. In this process, the selection of adequate predictors is key for model accuracy and generalization.

In this chapter, all these aspects are considered by developing separate SL models to estimate different throughput KPIs (i.e., average user throughput, aggregated throughput) with different granularity (i.e., cell- or slice-level) in different RATs (i.e., LTE and HSDPA) and scenarios (i.e., legacy and sliced RANs) from a set of candidate input features derived from multiple data sources (i.e., cell counters and connection traces).

### 4.3 Throughput estimation method

Fig. 4.1 outlines the estimation process followed once the target throughput metric and the set of candidate predictors are defined. First, data from all cells in the network is gathered in the OSS to build the dataset, which is then preprocessed to normalize the values of input features and create training and test datasets. Next, the training dataset

<sup>1</sup>Radio-electrical isolation refers to the absence of mutual interference at the air interface among different slices. In contrast, traffic isolation refers to the impossibility of a slice to transmit in PRBs allocated to other slice in a given cell.

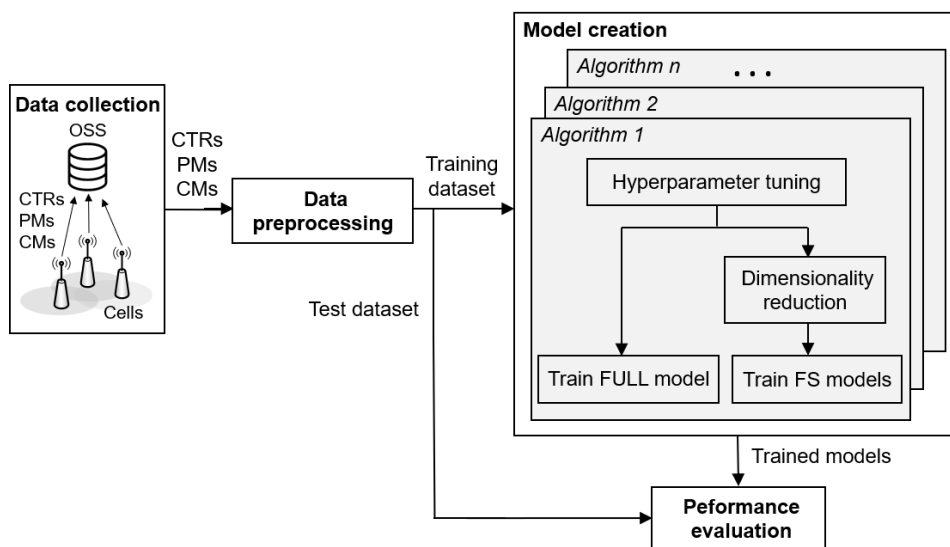


Figure 4.1: Throughput estimation method.

is used to build performance models. In a given network, separate models are created to estimate each throughput metric. Different FS techniques are tested to reduce the number of predictors. In all cases, hyperparameters of SL algorithms are adjusted to avoid overfitting. Finally, model performance is assessed on the corresponding test dataset. A more detailed explanation of each step is given next.

### 4.3.1 Data collection

This thesis tackles the estimation of three different radio throughput indicators:

- a) DL cell throughput [kbps], defined as the total data volume transmitted per second at the PDCP layer in active periods in the DL of a cell [104].
- b) DL user throughput [kbps], defined as the average data volume transmitted per second to each active user at the PDCP layer in the DL of a cell, excluding last TTIs [104].
- c) DL slice throughput [kbps], defined as the total data volume transmitted per second at the PDCP layer in active periods in the DL of a cell in PRBs assigned to a specific slice.

The specific set of considered candidate predictors for estimating these throughput indicators per RAT will be defined later in this chapter. In HSDPA and LTE, all input and output features can be computed from PMs aggregated on a cell basis and CMs. In contrast, in complex NS scenarios, the considered predictors are obtained from PMs



and CMs gathered on a cell and slice basis and radio traces. In live networks, all this data can be stored in the OSS after each ROP.

### 4.3.2 Data preprocessing

To ensure high accuracy and faster convergence of SL algorithms, input features are normalized, so that all have comparable ranges. For this purpose, a Z-score standardization method is used [151]. The scaled value of feature  $f$  for datapoint  $d$ , denoted as  $f_{\text{scaled}}(d)$ , is computed as

$$f_{\text{scaled}}(d) = \frac{f(d) - \mu_f}{\sigma_f}, \quad (4.2)$$

where  $\mu_f$  and  $\sigma_f$  are the mean and standard deviation for feature  $f$  in all datapoints in the dataset.

After data normalization, the  $N_s$  samples in the dataset are split into training and test subsets by creating a random partition. To avoid overfitting, the number of samples in the training dataset must be higher than the number of trainable model parameters.

### 4.3.3 Model creation

The conducted analysis seeks to find the best model to estimate a particular throughput indicator (i.e., DL cell, user or slice throughput) in all the entities (i.e., cell or cell-slice) in the same RAT within a network. Modeling is performed through SL. Different regression algorithms among those described in section 2.1.1 are tested, based on linear regression (MLR), support vectors (SVR), distance (KNN), DTs (XGBoost, AdaBoost and RF) and ANNs (MLP). Several different models are trained per algorithm and throughput metric: a) a full model (FULL) considering all candidate input features and b) several FS models considering a relevant subset of input features.

Two key aspects in modeling, namely hyperparameter optimization and dimensionality reduction, are detailed next.

### a) Dimensionality reduction

It is expected that the larger the number of input features (i.e., network indicators), the better estimation of the output feature (i.e., throughput metric). However, SL algorithms may underperform when input features are strongly correlated or are not relevant for the predicted variable. Moreover, when it comes to cellular networks, it is preferred to gather only useful data in the OSS to avoid: a) congestion problems in the backhaul due to the flow of data sent from base stations, b) unnecessary investment in large databases and processing platforms with a large computational power, and c) large data preprocessing and model training times, which can be critical for real-time applications. Additionally, collecting some data (e.g., connection traces) in NS scenarios may require an agreement between tenants and the infrastructure owner. Hence, dimensionality reduction is a key aspect in this analysis. The FS approach is selected for this purpose, since it eliminates the need for gathering irrelevant indicators in the OSS (note that this issue is not solved by FE).

Three different FS methods are tested, namely CORrelation-based FS (COR), Sequential Forward Selection (SFS) and Recursive Feature Elimination (RFE). COR is a simple filtering method that considers as relevant those features whose linear correlation,  $\rho$ , with the response variable is high, i.e.,  $|\rho| > 0.5$ . SFS is a wrapper method that starts with an empty model. Then, the most relevant features according to a predefined loss function are sequentially added until adding an additional feature does not significantly improve a predefined loss function. Oppositely, RFE is a wrapper method that starts with a model including all the candidate input attributes and sequentially removes the least relevant feature until an empty model is created [61].

Note that the set of candidate features differs per RAT. Moreover, some features may be relevant for estimating a certain throughput indicator, but negligible for others. Thus, COR must be performed per RAT and output feature, whereas SFS and RFE must be executed per RAT, output feature and SL algorithm.

### b) Hyperparameter tuning

Hyperparameters are internal model parameters controlling the learning process in ML algorithms. An adequate hyperparameter configuration is key to make the most of SL models. However, SL algorithms often have dozens (or even hundreds) of hyperparameters, and thus fine-grained tuning increases training time exponentially. For simplicity, in this thesis, the less influential parameters for each algorithm are fixed,

Table 4.1: Hyperparameters tuned for throughput estimation.

	Hyperparameter name	Parameter space
MLR	Fit intercept	True (fixed)
SVR	Sensitivity, $\epsilon$	[0.05, 0.4]
	Regularization, $C$	[10, 100]
	Kernel function	{linear, radial basis, polynomial}
KNN	No. of neighbors	[4, 20]
	Distance metric	Euclidean (fixed)
XGBoost	No. of trees	[50, 200]
	Maximum tree depth	[5, 10]
	No. of features per tree	No. input features (fixed)
	Loss function	Squared error (fixed)
	Learning rate ( $\eta$ )	[0.01, 0.1]
	L1 regularization term ( $\alpha$ )	[0.01 100]
	L2 regularization term ( $\lambda$ )	[0.01 100]
Minimum loss reduction for splitting	0 (fixed)	
AdaBoost	No. of trees	[10, 50]
	Loss function	{linear, square, exponential}
	Learning rate	[0.1, 0.7]
RF	No. of trees	[30, 100]
	Maximum tree depth	[5, 50]
	Minimum node size to split	2 (fixed)
	No. of features per tree	{ $\sqrt{N_f}, N_f$ } ( $N_f$ stands for no. of features)
	Bagging	Enabled (fixed)
	Criterion to measure split quality	Mean absolute error (fixed)
	Minimum impurity reduction for splitting	0 (fixed)
MLP	No. of layers	[3, 10]
	No. of neurons per hidden layer	[5, $N_f$ ]
	Weight initialization	Glorot []
	Activation function in hidden layers	{Hyperbolic tangent, linear, sigmoid, rectified linear unit}
	Activation function in output layer	Rectified linear unit (fixed)
	Optimization algorithm	{Adam, L-BFGS}
	Mini-batch size (Adam optimizer)	64 (fixed)
	Loss function	MAE (fixed)
	Max. no. of epochs	1000 (fixed)
	Train / validation split	70% / 30% (fixed)
	Early stopping condition	Accuracy in the validation dataset does not improve in 3 epochs

and only the most influential parameters according to previous works [183] are tuned through a random grid search in the parameter space [184]. Table 4.1 breaks down the main hyperparameters for the considered SL algorithms, together with the configured fixed value or parameter space. The reader is referred to [37] for a detailed explanation of these hyperparameters. The best hyperparameter value (or tuple) is that minimizing a given FoM (e.g., mean absolute error, mean percentage error...). Since the best setting strongly depends on the problem and set of predictors, the tuning process is performed separately for each RAT, output feature and selected predictors among candidates when performing FS (e.g., at each step of the SFS/RFE process).

#### 4.3.4 Performance evaluation

Model performance is assessed over the test dataset with different Figures of Merit (FoMs). Accuracy FoMs defined for non-sliced and sliced networks will be presented in subsequent sections 4.4.2 and 4.5.2, respectively. In both cases, the number of input features per model is also considered as a proxy of required storage capacity in the OSS and load in the backhaul due to data exchange. Finally, training time is measured as a measure of computational complexity. The best model (i.e., SL algorithm and FS technique) is selected as a trade-off of all these FoMs.

## 4.4 Cell and user capacity estimation in HSDPA and LTE

This section tackles the problem of estimating radio throughput indicators reflecting cell and user capacity in HSDPA and LTE with SL over data collected in the OSS. Assessment is carried out over two datasets obtained in commercial networks. For clarity, datasets are first described, performance assessment is detailed next and the main conclusions are finally exposed.

### 4.4.1 Dataset description

Two different datasets are collected from a live HSDPA network and a live LTE network. The main characteristic of these networks and the dataset creation process are outlined next.

### a) Dataset A – HSDPA

The first dataset is collected in a live 3G network serving an entire country (approximately 10,000 km<sup>2</sup>), comprising 12,318 cells of very different sizes and environments. Two carrier frequencies are deployed per cell. A first carrier is used for Adaptive Multi-Rate Circuit-Switched (AMR CS) calls and non-HSDPA packet-switched traffic, while a second carrier is used for HSDPA traffic and AMR CS calls when the first carrier is full. The analysis is focused on the second carrier (i.e., HSDPA capacity), for which capacity estimation is more difficult.

CMs and PMs are collected on a cell and hourly basis for a whole day in the OSS. In HSDPA, the maximum cell/user capacity is defined as DL cell/user throughput metrics introduced in section 4.3.1 measured when the TTI utilization ratio,  $TTI_{util\_rat}$ , is high. Thus, only data from highly loaded cells is considered in the analysis. The selection of such cells is carried out through the observation of the cell busy hour, defined as the hour with the largest average number of active UEs (i.e., with data to transmit) over HSDPA. Analysis is restricted to those cells with  $TTI_{util\_rat} > 50\%$  during the busy hour. This filter results in a dataset comprising 1,095 datapoints with the following features:

- a) Cell identifier.
- b) Date (format DD/MM/YYYY HH:HH).
- c) TTI utilization ratio in HSDPA,  $TTI_{util\_rat}$ , as a measure of cell load.
- d) A set of 12 network indicators, shown in Table 4.2, as candidate input features for capacity estimation. To allow the comparison with previous approaches, the considered input features are the same as in [22], including code-related indicators (e.g., no. of codes used in HSDPA), traffic-related indicators (e.g., no. of active UEs), power-related indicators (e.g., avg. DL transmit power for HSDPA) and quality-related indicators (e.g., median CQI).
- e) DL cell throughput in HSDPA,  $TH_{cell}^{HSDPA}$ , defined in section 4.3.1, as a measure of cell capacity (i.e., variable to be predicted).
- f) DL user throughput in HSDPA,  $TH_{user}^{HSDPA}$ , defined in section 4.3.1, as a measure of user capacity per cell (i.e., variable to be predicted).

Table 4.3 presents the minimum, maximum and mean value and the standard deviation of input and output features in the HSDPA dataset.

Table 4.2: Candidate input features for estimating DL cell throughput in HSDPA.

Type	Name	Description
Power	<i>Avg_R99_DL_power</i> [mW]	Average DL transmit power for Data CHannel (DCH)
	<i>Avg_HSDPA_DL_power</i> [mW]	Average DL transmit power for HSDPA
Traffic	<i>Avg_activeUE</i>	Average number of HSDPA active UEs per TTI in DL
Code	<i>Avg_codes_used_HSDPA</i>	Average number of codes used in HSDPA
	<i>Avg_SF16_codes_HSDPA</i>	Avg. number of codes with spreading factor 16 reserved for HSDPA
	<i>Avg_codes_HSDPA_UE</i>	Average number of codes used per HSDPA user
	<i>Code_Load</i> [%]	Percentage of channelization codes used in both DCH and HSDPA
Quality	<i>CQI_class_p50</i>	Median DL CQI
	<i>CQI_class_p80</i>	80th-tile of DL CQI distribution
	<i>16QAM_usage</i> [%]	Usage of 16QAM modulation (as opposed to QPSK)
	<i>RLC_retx_ratio_DL</i>	Ratio of RLC retransmissions in DL
	<i>PDU656_usage</i> [%]	Percentage of packet data units with size 656 B (as opposed to 310 B)

Table 4.3: Statistics of dataset A (HSDPA network).

Indicator	Min.	Max.	Mean	Std. deviation
<i>Avg_R99_DL_power</i> [mW]	2,014	15,777	7,512	1,844
<i>Avg_HSDPA_DL_power</i> [mW]	1,200	16,060	5,684	1,368
<i>Avg_activeUE</i>	0.85	62.68	20.43	10.85
<i>Avg_codes_used_HSDPA</i>	1.10	9.20	4.30	1.10
<i>Avg_SF16_codes_HSDPA</i>	5	14	9.48	1.93
<i>Avg_codes_HSDPA_UE</i>	7.15	11.40	9.29	0.67
<i>Code_Load</i> [%]	47.62	96.66	88.16	3.59
<i>CQI_class_p50</i>	6	22	15.06	1.90
<i>CQI_class_p80</i>	9	26	19.47	1.86
<i>16QAM_usage</i> [%]	0.10	89.20	22.09	14.41
<i>RLC_retx_ratio_DL</i>	0.02	1.34	0.14	0.07
<i>PDU656_usage</i> [%]	0	135.80	32.82	26.02
$TH_{cell}^{HSDPA}$ [kbps]	252.82	4768.21	1859.92	648.27
$TH_{user}^{HSDPA}$ [kbps]	11.43	1771.50	90.27	112.09

## b) Dataset B – LTE

The second dataset is collected in a live LTE network comprising 656 cells covering urban and residential areas. In this network, two carriers are deployed at 700 MHz and 2100 MHz with a system bandwidth of 10 MHz and 5 MHz, respectively. In this case, the analysis includes both carriers. To obtain the dataset, CMs and PMs are gathered on an hourly and cell basis for 6 days (note that the smaller size of the

Table 4.4: Candidate input features for estimating DL cell throughput in LTE.

Type	Name	Description
Quality	<i>Avg_CQI</i>	Average DL CQI
	$\sigma_{CQI}$	Standard deviation of DL CQI distribution
	<i>CQI_class_p5</i>	5th-tile of DL CQI distribution
	<i>CQI_class_p10</i>	10th-tile of DL CQI distribution
	<i>HARQ_fail_ratio_DL</i>	Hybrid Automatic Repeat reQuest (HARQ) failure ratio in DL
	<i>RLC_retx_ratio_DL</i>	Ratio of RLC retransmissions in DL
	<i>DL_assign_Ack</i>	Ratio of correct resource assignments in DL control channel
Traffic	<i>Avg_activeUE</i>	Average number of active UEs per TTI in DL
CMs	<i>BW [MHz]</i>	LTE system bandwidth
	<i>PUCCH_SR_users</i>	Max. number of UEs allowed to send Scheduling Request in UL

network allowed a longer data collection period compared to the HSDPA case). Again, to obtain reliable capacity estimates, the analysis is restricted to those datapoints from highly loaded cells, i.e., those with DL PRB utilization ratio,  $PRButil_{rat}$ , higher than 50% in the daily busy hour. This filter results in a dataset with 2,141 datapoints with the following information:

- a) Cell identifier.
- b) Date (format DD/MM/YYYY HH:HH).
- c) DL PRB utilization ratio,  $PRButil_{rat}$ , as a measure of cell load.
- d) A set of 10 network indicators, shown in Table 4.4, as candidate input features. These include network settings (e.g., system bandwidth), quality-related statistics (e.g., average CQI) and traffic-related statistics (e.g., no. of active UEs) provided by most vendors and used in previous studies for capacity estimation in LTE [23].
- e) DL cell throughput,  $TH_{cell}^{LTE}$ , defined in section 4.3.1, as a measure of cell capacity (i.e., variable to be predicted).
- f) DL user throughput,  $TH_{user}^{LTE}$ , defined in section 4.3.1, as a measure of user capacity per cell (i.e., variable to be predicted).

Table 4.5 presents the minimum, maximum and average value and the standard deviation of input and output features in the LTE dataset.

From the comparison of Tables 4.2 and 4.4, it is observed that some of the considered indicators provide similar information in both technologies (e.g., number of active UEs, RLC retransmissions, CQI, etc.), and, thus, both analyses rely on simi-

Table 4.5: Statistics of dataset B (LTE scenario).

Indicator	Min.	Max.	Mean	Std. deviation
<i>Avg_CQI</i>	5.50	12.21	7.81	0.95
$\sigma_{CQI}$	0.29	3.13	0.85	0.19
<i>CQI_class_p5</i>	1.31	7.12	3.48	0.68
<i>CQI_class_p10</i>	1.69	8.29	4.17	0.74
<i>HARQ_fail_ratio_DL</i>	0.05	0.11	0.07	$7.3 \cdot 10^{-3}$
<i>RLC_retx_ratio_DL</i>	$1.2 \cdot 10^{-5}$	0.05	$7.2 \cdot 10^{-4}$	$1.6 \cdot 10^{-3}$
<i>DL_assign_Ack</i>	0.26	0.99	0.96	0.07
<i>Avg_activeUE</i>	0.30	16.97	1.69	1.06
<i>BW</i> [MHz]	5	10	9.44	1.57
<i>PUCCH_SR_users</i>	560	730	646.35	34.06
$TH_{cell}^{LTE}$ [kbps]	2441.96	22967.20	8471.62	2112.73
$TH_{user}^{LTE}$ [kbps]	514.31	16004.31	4691.54	1804.45

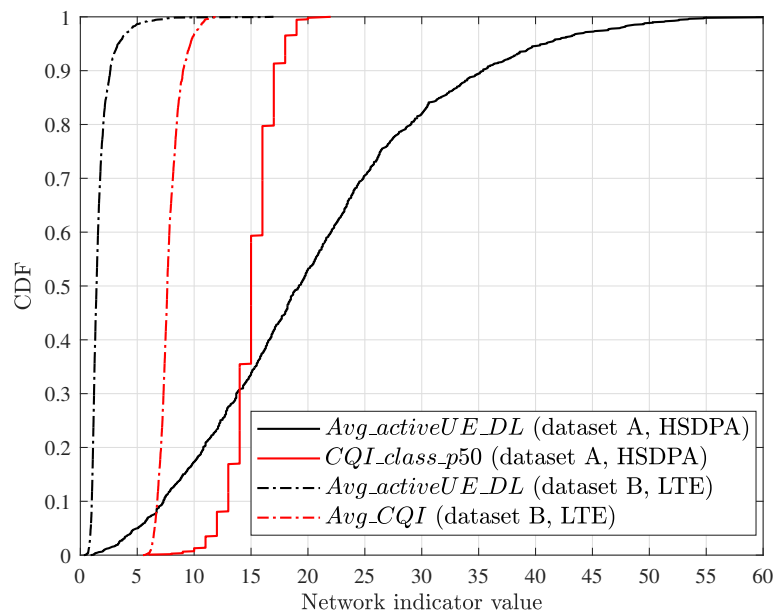


Figure 4.2: Cumulative distribution function of network indicators in HSDPA and LTE datasets.

lar initial information. Nonetheless, other indicators are distinctive of the technology (e.g., code-related indicators in HSDPA), so that technology-specific information is also considered. For a deeper analysis, Fig. 4.2 shows the Cumulative Distribution Function (CDF) of *Avg\_activeUE* and *CQI\_class\_p50* indicators in dataset A (solid lines) and *Avg\_activeUE* and *Avg\_CQI* indicators in dataset B (dashed lines). Note that, although only highly-loaded cells are considered in both technologies, *Avg\_activeUE* in



LTE is lower than in HSDPA, revealing that users in LTE demand more data-hungry services. Likewise, the highest values of CQI measured per cell (i.e., 22 in HSDPA scenario and 12 in LTE scenario according to Tables 4.2 and 4.4, respectively) are below the maximum CQI value defined in each RAT (i.e., 30 in HSDPA and 15 in LTE [140]).

Both datasets A and B combine a large geographical area (hundreds of cells) with an adequate time resolution (hour), similarly to those used by operators for capacity estimation purposes. This fact guarantees the reliability and significance of results. Note that, because of the filtering based on busy hour and TTI utilization ratio, these datasets have a reduced number of samples. This property increases the interest of assessing the performance of non-deep SL algorithms, less prone to overfitting than DNNs when trained with reduced datasets, for cell/user capacity estimation.

## 4.4.2 Performance assessment

This section presents the assessment of the estimation methodology described in section 4.3 over the datasets introduced in section 4.4.1. For clarity, analysis set-up is first explained. Then, results are presented, broken down per RAT. Finally, computational complexity is discussed.

### a) Analysis set-up

The proposed estimation methodology is particularized as follows. Regarding data pre-processing, for each dataset, 80% of datapoints are used for training and the remaining 20% are used for test. Regarding modeling, six regression algorithms are compared, namely MLR, SVR, RF, KNN and two MLPs differing in the number of hidden layers. The first one, denoted as MLP-SNN, has a single hidden layer (a.k.a. shallow ANN) whose number of units is determined by grid search as set in Table 4.1. The second one, denoted as MLP-DNN, is a DNN based on that tested in [25] to estimate DL cell and user throughput in LTE. The number of hidden layers is 3/4 for cell/user capacity estimation (values obtained through grid search) with as many units as input features. All SL algorithms are implemented with *scikit-learn* and *Keras*, two ML libraries for Python extensively used in several fields. The reader is referred to [185] [186] for further information on the implementation of SL algorithms in these libraries.

Three models are derived with each regression algorithm: a full model with all candidate predictors (FULL model), a simplified model with a subset of input features

selected by COR method (FS–COR model) and a simplified model with predictors selected by SFS (FS–SFS model). Thus, 18 regression models are tested. For each model, the best hyperparameter value (or tuple) is that minimizing the Mean Absolute Percentage Error,  $MAPE$ , in the training dataset.  $MAPE$  is computed as

$$MAPE = \frac{1}{N_s} \sum_{i=1}^{N_s} \left| 100 \cdot \frac{\hat{y}(i) - y(i)}{y(i)} \right|, \quad (4.3)$$

where  $N_s$  is the number of datapoints, and  $y(i)$  and  $\hat{y}(i)$  are the measured and estimated values of the output feature in datapoint  $i$ , respectively. Such a metric is also the loss function in the SFS process, where the condition to select the optimal number of features,  $N_f^{opt}$ , is when the decrease in  $MAPE$  after adding a new feature is lower than 1% provided that  $MAPE < 10\%$ . To prevent overfitting, a 5-fold cross validation is performed over the training dataset when tuning hyperparameters and over the whole dataset at each step of the RFE process [37].

Performance evaluation is based on  $MAPE$ , complemented by the number of input features and training time as a proxy to storage and computational efficiency. A model (i.e., combination of SL algorithm and FS scheme) is considered acceptable to estimate DL cell/user throughput if  $MAPE < 10\%$ . This value has been considered as an acceptable error in previous works [24], since it provides a trade-off between model complexity and accuracy. A more restrictive  $MAPE$  threshold can only be achieved by complex models requiring large training datasets and higher training times. In network planning tools, such an increase in complexity does not pay off, since operators have to take the same replanning actions whether capacity problems are detected with a  $MAPE$  of 5% or 10%. On the contrary, a too relaxed threshold can lead to unnecessary investments (e.g., bandwidth extension licenses) if capacity is underestimated, or to capacity bottlenecks (e.g., underprovision of radio resources) if capacity is overestimated. In live networks, the accuracy threshold is up to the MNO. No matter the set value, the worst case is not detecting problems due to overestimating capacity, since user experience may be degraded. To ensure that all potential issues are detected, parameters in redimensioning NFs (e.g., thresholds to trigger minor, major or critical alarms) must be set considering the expected model error.

The best model for each throughput indicator (output feature) is that with a  $MAPE$  comparable to the best model (i.e., difference lower than 2% in absolute terms) and the lowest number of input features.

Table 4.6: *MAPE* for estimating DL cell and user throughput in HSDPA [%].

Model	$TH_{cell}^{HSDPA}$			$TH_{user}^{HSDPA}$		
	FULL	FS-COR	FS-SFS	FULL	FS-COR	FS-SFS
$N_f$	12	4	—	12	4	—
MLR	7.37	9.39	8.13 ( $N_f^{opt} = 4$ )	43.11	49.15	41.24 ( $N_f^{opt} = 4$ )
SVR	7.36	9.34	8.17 ( $N_f^{opt} = 4$ )	13.31	21.75	11.03 ( $N_f^{opt} = 4$ )
RF	6.69	9.06	7.04 ( $N_f^{opt} = 4$ )	13.42	22.44	12.11 ( $N_f^{opt} = 4$ )
KNN	10.03	8.46	8.27 ( $N_f^{opt} = 3$ )	17.09	22.52	12.33 ( $N_f^{opt} = 4$ )
MLP-SNN	5.60	9.33	6.80 ( $N_f^{opt} = 4$ )	10.64	21.68	9.23 ( $N_f^{opt} = 5$ )
MLP-DNN	5.81	8.99	7.13 ( $N_f^{opt} = 4$ )	8.79	20.95	10.26 ( $N_f^{opt} = 10$ )

## b) Results – HSDPA

Table 4.6 breaks down the results obtained for the considered regression algorithms when estimating  $TH_{cell}^{HSDPA}$  and  $TH_{user}^{HSDPA}$  with the FULL, FS-COR and FS-SFS models.  $TH_{cell}^{HSDPA}$  results are analyzed first and  $TH_{user}^{HSDPA}$  is considered later.

Results from FULL models show that, as stated in [22], MLR achieves an adequate accuracy (i.e.,  $MAPE=7.37\%$ ) when estimating  $TH_{cell}^{HSDPA}$ . This result points out that some input features have a strong linear relationship with the output variable. RF, MLP-SNN and MLP-DNN improve MLR accuracy, with a  $MAPE$  of 6.69%, 5.60% and 5.81%, respectively. KNN shows the worst results, although its  $MAPE$  ( $=10.03\%$ ) is still acceptable. An analysis of the Pearson correlation coefficients (not shown here) reveals that  $CQI\_class\_p50$ ,  $CQI\_class\_p80$ ,  $Avg\_codes\_used\_HSDPA$  and  $16QAM\_usage$  features are linearly correlated with  $TH_{cell}^{HSDPA}$  (i.e.,  $|\rho|>0.5$ ). This is reinforced by the fact that FS-COR models, which use only those 4 indicators as input features, achieve a  $MAPE$  below 10% for all algorithms. Fig. 4.3 depicts the evolution of the  $MAPE$  obtained across FS-SFS process. As expected, in general, the larger number of features, the higher accuracy. However, KNN performance degrades progressively when the number of features grows above  $N_f=5$ . This unexpected behavior reveals that KNN is suffering the so-called *curse of dimensionality*, since it requires all neighbor datapoints to be close in all dimensions of the data space, which becomes more difficult as the input feature space grows [187]. MLP-SNN and RF perform similarly, providing the best results with a low number of input features. Table 4.6 includes, in FS-SFS column, the  $MAPE$  obtained for each algorithm with  $N_f^{opt}$  selected with the predefined convergence criterion.  $MAPE$  values show that FS-SFS models reduce the required storage capacity compared to the FULL models at the expense of a negligible degradation in  $MAPE$  ( $\lesssim 1\%$  in absolute terms for all algorithms). In

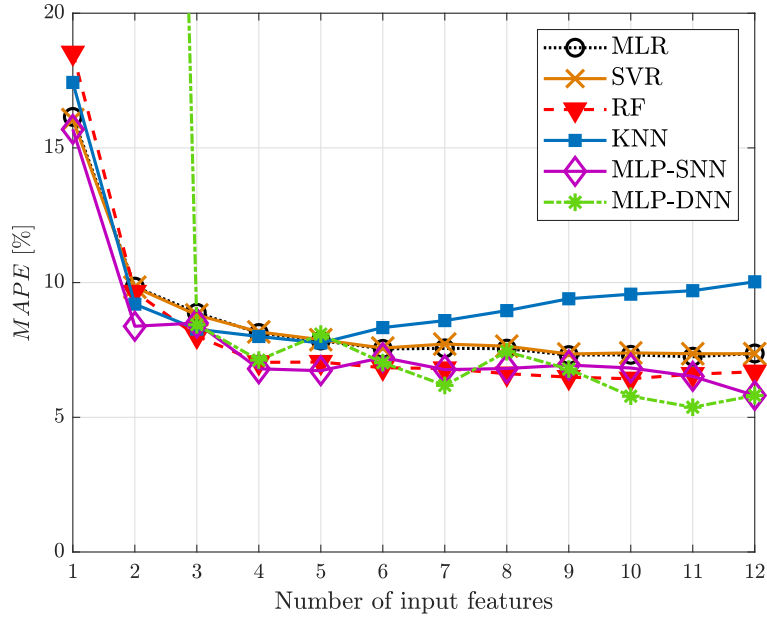


Figure 4.3:  $MAPE$  evolution across sequential feature selection (FS-SFS) process when estimating DL cell throughput in HSDPA.

KNN, FS-SFS model is more accurate than the FULL model (i.e.,  $MAPE=10.03\%$  with FULL model, and  $8.27\%$  for FS-SFS model) for the above reasons. Overall, the best model is MLP-SNN with FS-SFS, since it achieves a  $MAPE$  very close to the best model ( $6.80\%$ ) with only 4 input features ( $CQI\_class\_p50$ ,  $Avg\_SF16\_codes\_HSDPA$ ,  $Avg\_codes\_used\_HSDPA$  and  $16QAM\_usage$ ). Nonetheless, Fig. 4.3 reveals that an acceptable  $MAPE$  (i.e.,  $<10\%$ ) can be achieved with all non-deep SL algorithms by selecting a subset of only 2 features (specifically,  $Avg\_codes\_used\_HSDPA$  and  $16QAM\_usage$ ). Hence, it can be concluded that MLR is competitive with more sophisticated SL algorithms when estimating busy-hour cell throughput in HSDPA DL.

It is remarkable that the subset of features in the best option (MLP-SNN with  $N_f^{opt}=4$ ) differs in the number of features and in some of the selected features from the subset in [22], where  $TH_{cell}^{HSDPA}$  is estimated via MLR and feature selection is performed based on  $p$ -values. In that work, the authors propose a model with 5 input features:  $CQI\_class\_p50$ ,  $Avg\_codes\_used\_HSDPA$ ,  $Avg\_HSDPA\_DL\_power$ ,  $16QAM\_usage$  and  $PDU656\_usage$ . Thus, it can be concluded that, when estimating cell capacity, not only how many features but also which features must be stored in the OSS depend on the selected SL algorithm and FS approach. This fact justifies that feature selection and regression must be jointly analyzed.

When it comes to DL user throughput estimation, MLR does not perform well, with

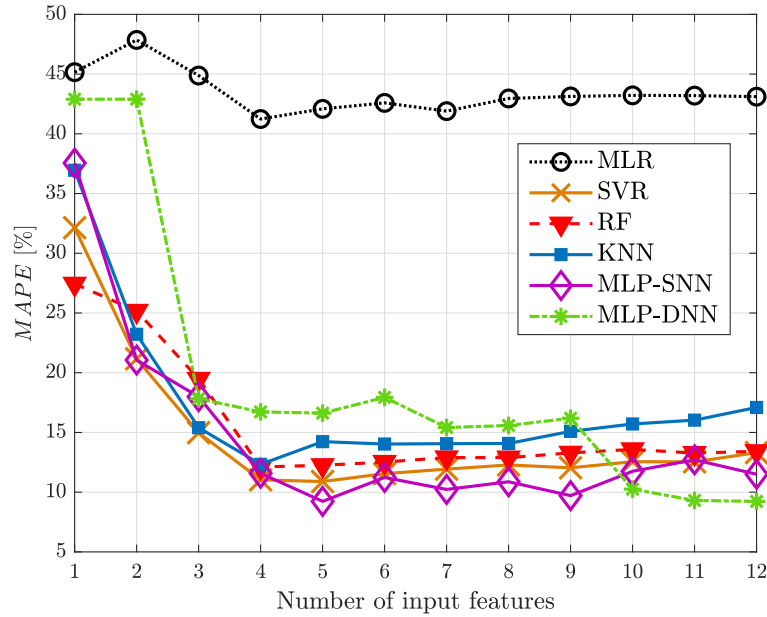


Figure 4.4:  $MAPE$  evolution across sequential feature selection (FS-SFS) process when estimating DL user throughput in HSDPA.

a  $MAPE$  of 43.41% for the FULL model. This poor performance suggests that there is a non-linear relationship between the input features and the output variable,  $TH_{user}^{HSDPA}$  (on the contrary, it is expected that  $TH_{user}^{HSDPA}$  is inversely proportional to the number of simultaneous active UEs,  $Avg\_activeUE$ ). FULL models created with all other algorithms outperform MLR, with  $MAPE$  values below 18%. Still, only MLP-DNN achieves a  $MAPE$  below the 10% threshold (8.79%). FS-COR models strongly degrade accuracy for all regression algorithms. For instance, in MLP-SNN,  $MAPE$  grows from 10.64% to 21.68% when comparing FULL and FS-COR models (i.e., an increase of 103% in relative terms). These numbers are consistent with the above statement about the non-linear relationship among input and output features, since FS-COR is a FS process based on linearity. Fig. 4.4 shows the evolution of  $MAPE$  across FS-SFS process. In this case, even for algorithms not based on distance, such as MLP-SNN, the larger number of features does not necessarily lead to a higher accuracy, revealing that some of the considered input features are irrelevant for estimating  $TH_{user}^{HSDPA}$  or provide redundant information. MLP-SNN achieves the best results when  $N_f$  is between 4 and 10, whereas MLR clearly shows the worst performance at every point. Again, the best point is  $N_f^{opt} \leq 5$  for all algorithms but MLP-DNN, whose best performance is with  $N_f^{opt} = 10$ .  $MAPE$  obtained over the test dataset at those points is shown in Table 4.6. Results reveal that, unexpectedly, most FS-SFS models outperform FULL

Table 4.7: *MAPE* for estimating DL cell and user throughput in LTE [%].

Model	$TH_{cell}^{LTE}$			$TH_{user}^{LTE}$		
	FULL	FS-COR	FS-SFS	FULL	FS-COR	FS-SFS
$N_f$	10	1	—	10	2	—
MLR	9.09	14.12	9.84 ( $N_f^{opt} = 5$ )	17.79	23.09	18.03 ( $N_f^{opt} = 5$ )
SVR	7.36	13.72	8.32 ( $N_f^{opt} = 5$ )	12.36	17.33	12.14 ( $N_f^{opt} = 5$ )
RF	7.25	16.29	7.21 ( $N_f^{opt} = 5$ )	10.04	16.72	10.04 ( $N_f^{opt} = 6$ )
KNN	7.64	15.58	8.86 ( $N_f^{opt} = 4$ )	10.13	16.59	9.62 ( $N_f^{opt} = 5$ )
MLP-SNN	6.96	13.93	8.98 ( $N_f^{opt} = 8$ )	7.95	15.17	8.86 ( $N_f^{opt} = 9$ )
MLP-DNN	6.86	14.11	8.34 ( $N_f^{opt} = 5$ )	8.86	17.73	8.73 ( $N_f^{opt} = 9$ )

models (e.g., in KNN, *MAPE* decreases from 17.09% to 12.33%). Overall, the best results are obtained with the combination MLP-SNN+FS-SFS, being the only one achieving a *MAPE* lower than 10% (9.23%) with a reduced subset of input features. The predictors selected in that model are *16QAM\_usage*, *Avg\_codes\_used\_HSDPA*, *Avg\_SF16\_codes\_HSDPA*, *Avg\_R99\_DL\_power* and *Avg\_activeUE*.

### c) Results – LTE

Table 4.7 summarizes the results obtained for the considered algorithms when estimating  $TH_{cell}^{LTE}$  and  $TH_{user}^{LTE}$  with the FULL, FS-COR and FS-SFS models in LTE. Again, MLR provides acceptable accuracy with FULL model when estimating  $TH_{cell}^{LTE}$  (*MAPE*=9.09%), but not when estimating  $TH_{user}^{LTE}$  (*MAPE*=17.79%). All other algorithms outperform MLR in both DL cell and user throughput estimations. When estimating  $TH_{cell}^{LTE}$  with FULL model, SL algorithms perform similarly (*MAPE*≈7%). However, when estimating  $TH_{user}^{LTE}$ , only MLP-SNN and MLP-DNN fulfill the 10% threshold (*MAPE*≈8% and 9%, respectively). FS-COR models degrade accuracy significantly, showing that the most relevant features do not have a strong linear relation to the output variables. In fact, an analysis of Pearson correlation coefficients (not shown here) reveals that only *HARQ\_fail\_ratio\_DL* has a significant linear correlation with  $TH_{cell}^{LTE}$ , and only *HARQ\_fail\_ratio\_DL* and *Avg\_activeUE* have a significant linear correlation with  $TH_{user}^{LTE}$ .

Fig. 4.5 and 4.6 show the *MAPE* evolution across FS-SFS process when estimating  $TH_{cell}^{LTE}$  and  $TH_{user}^{LTE}$ , respectively. In general, the larger number of features, the higher accuracy. Table 4.7 includes, in FS-SFS columns, the *MAPE* of each method with the selected  $N_f^{opt}$  value. When considering the trade-off between accuracy and number of predictors, KNN is the best option for estimating both  $TH_{cell}^{LTE}$

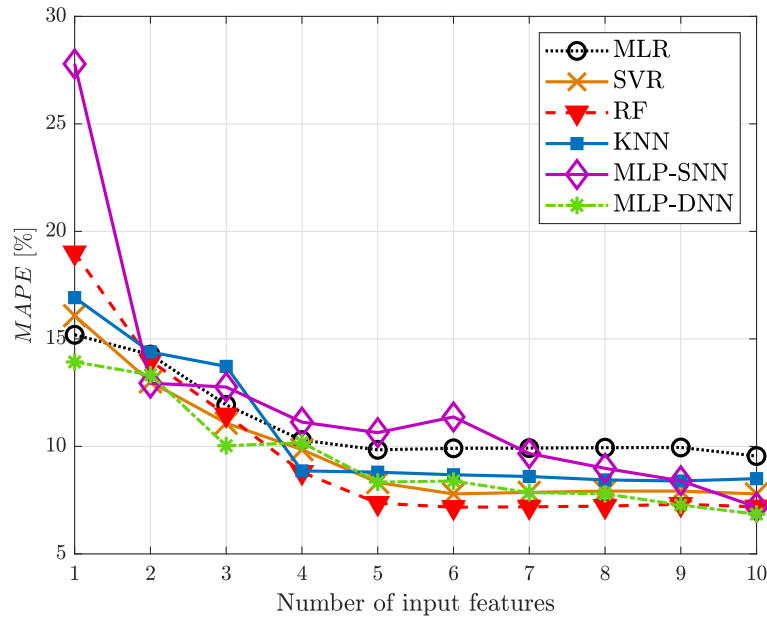


Figure 4.5:  $MAPE$  evolution across sequential feature selection (FS-SFS) process when estimating DL cell throughput in LTE.

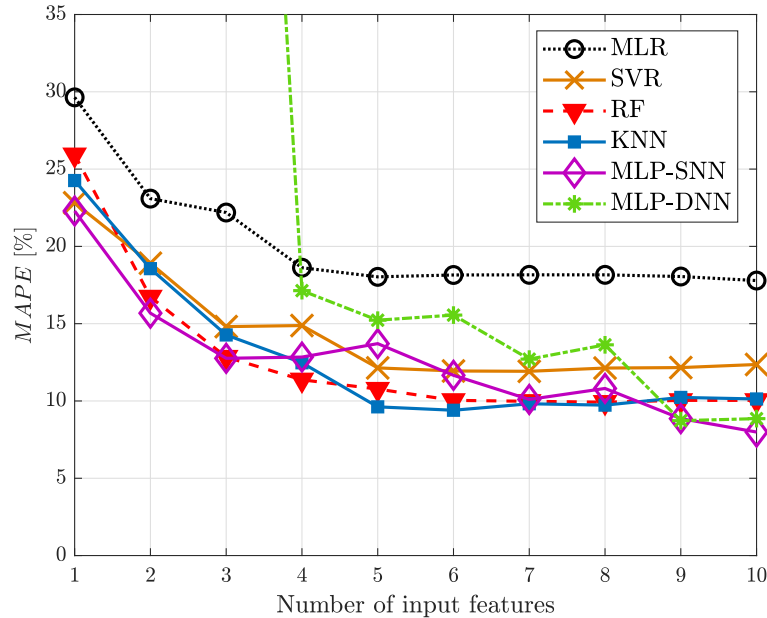


Figure 4.6:  $MAPE$  evolution across sequential feature selection (FS-SFS) process when estimating DL user throughput in LTE.

( $MAPE=8.86\%$  for  $N_f^{opt}=4$ ) and  $TH_{user}^{LTE}$  ( $MAPE=9.62\%$  for  $N_f^{opt}=5$ ). The most relevant input features for estimating cell throughput are  $Avg\_CQI$ ,  $DL\_assign\_ACK$ ,  $BW$  and  $HARQ\_fail\_ratio\_DL$ . Unlike [23],  $DL\_assign\_ACK$  is selected instead

of *Avg\_activeUE*. Likewise, the most relevant input features for estimating user throughput are *CQI\_class\_p10*, *Avg\_CQI*, *Avg\_activeUE*, *PUCCH\_SR\_users* and *DL\_assign\_ACK*.

It should be pointed out that, among the considered regression algorithms, MLP approaches have the largest number of hyperparameters. The optimal value of these hyperparameters may vary at each step of the SFS process. In this analysis, for efficiency, only the most relevant parameters have been tuned (as network operators do). This is probably the reasons for the unstable behavior of MLP approaches across SFS, translated into peaks in *MAPE* (e.g., MLP–SNN with  $N_f=6$  in Fig. 4.5 and  $N_f=5$  in Fig. 4.6) and severe performance degradation below a certain number of features (e.g., MLP–DNN with  $N_f \leq 2$  in Fig. 4.3 and 4.4, or  $N_f \leq 3$  in Fig. 4.6).

#### d) Computational complexity

The implementation of SL models for estimating throughput in radio planning tools entails: a) collecting and preprocessing data in the OSS, b) selecting the best model (i.e., combination of SL algorithm and set of input features) for the specific network, c) exploiting the model and d) retraining the model when necessary.

Data used to compute input features considered (i.e., PMs and CMs) is often collected and processed by MNOs for network management purposes, so that dataset creation should not entail a significant additional computational workload. If required, parallelization can be used to speed up data processing.

The most time-consuming task is finding the best model, since it implies carrying out the SFS process for several candidate SL algorithms. For MLR, training time grows linearly with the number of input features,  $N_f$ , and the number of samples in the dataset,  $N_s$ . Thus, the worst-case time complexity is  $\mathcal{O}(N_s \times N_f)$ . The back propagation algorithm used to train a MLP with 1 output and 3 layers has a worst-case time complexity of  $\mathcal{O}(N_s \times N_f \times N_l \times N_i)$ , where  $N_l$  is the size of the hidden layer and  $N_i$  is the number of iterations. Time complexity of sequential minimal optimization used to train SVR is quadratic with the training set size and linear with the number of features,  $\mathcal{O}(N_s^2 \times N_f)$ . Likewise, the worst-case time complexity of ensemble models based on DTs is given by the number of trees ( $N_t$ ) and time of building a tree, i.e.,  $\mathcal{O}(N_t \times N_f \times N_s \times \log N_s)$ . Finally, for KNN, the worst-case complexity is given by  $\mathcal{O}(N_f \times N_s \times k)$ , where  $k$  is the number of neighbors. For instance, Table 4.8 summarizes the time taken to train FULL models once hyperparameters have been



Table 4.8: Training times for throughput estimation models in HSDPA and LTE [s].

	$TH_{cell}^{HSDPA}$	$TH_{user}^{HSDPA}$	$TH_{cell}^{LTE}$	$TH_{user}^{LTE}$
MLR	<0.01	<0.01	0.02	<0.01
SVR	0.05	0.09	0.84	0.17
RF	1.14	2.53	3.02	3.89
KNN	<0.01	<0.01	<0.01	<0.01
MLP-SNN	0.23	0.71	0.66	0.95
MLP-DNN	22.92	23.72	32.47	33.28

fixed in a centralized server with Intel Xenon octa-core processor, clock frequency of 2.4 GHz and 64 GB of RAM. Results show that model training in HSDPA is faster than in LTE, possibly due to the highest number of datapoints in dataset B. For a given technology and regression algorithm, training when estimating cell throughput is faster than when estimating user throughput. MLP-DNN takes the largest execution time for every technology and output feature, whereas MLR and KNN show the lowest execution times. Nonetheless, even in the worst case (i.e., training a MLP-DNN model to estimate  $TH_{user}^{LTE}$ ), the obtained execution time is only 33 s. This time decreases significantly with FS-SFS models.

Once the best SL scheme is selected, exploiting the models is immediate (e.g., in this analysis, prediction time per datapoint is approximately 0.5 ms). This time meets the requirements even of the most stringent slice redimensioning NFs, typically working on a second or millisecond timescale.

Estimation models must be executed again after any significant change affecting input variables (e.g., change in traffic demand, radio channel conditions or cell bandwidth). Likewise, models must be retrained if an event changing the relationship between predictors and the output variable happens in the network (e.g., an update of packet scheduling algorithm, the launch of new services or the introduction of new terminal and base station capabilities). Different models must also be trained for different networks.

### 4.4.3 Conclusions

Accurate estimates of cell and user capacity in the RAN are key for smart radio planning tools. In this section, a comparative analysis has been presented assessing the performance of different SL algorithms to estimate cell and user throughput in the DL in busy hours from network measurements collected in the OSS. Model assessment has

been carried out with two datasets taken from live HSDPA and LTE networks. Four well-known non-deep SL methods have been compared with MLR and DNN models proposed in previous works.

Results show that MLR performs well when estimating DL cell throughput in both HSDPA and LTE ( $MAPE=7.37\%$  and  $9.09\%$ , respectively), but not when estimating DL user throughput ( $MAPE=41.14\%$  and  $17.79\%$ , respectively), probably due to the non-linear relationship between cell-level indicators and user-level metrics. Nonetheless, other SL approaches outperform MLR in terms of accuracy in both DL cell and user throughput estimation. The DNN achieves adequate accuracy (i.e.,  $MAPE<10\%$ ) in all cases when the full set of network indicators is available. However, its performance strongly degrades when the number of features decreases. Alternatively, with non-deep SL, it is possible to train models to estimate DL cell/user throughput with similar accuracy relying on reduced datasets (less than 2,000 samples and collection of 5 or 6 indicators in the OSS). To achieve this goal, a feature selection process must be performed by wrapper methods.

Considering the trade-off between accuracy and number of predictors, MLP–SNN has shown the best results in HSDPA, with  $MAPE=6.80\%$  with 4 input features when estimating DL cell throughput, and  $MAPE=9.23\%$  with 5 input features for DL user throughput. In contrast, in LTE, KNN has shown the best performance, with  $MAPE=8.86\%$  with 4 input features for DL cell throughput, and  $MAPE=9.62\%$  with 5 input features for DL user throughput.

## 4.5 Cell and slice throughput estimation in sliced radio access networks

This section addresses the problem of estimating radio throughput at cell and slice level in sliced RANs through SL over network data gathered in the OSS. The analysis focuses on eMBB services, for which throughput is a key performance metric. For clarity, content is structured as in section [4.4](#), i.e., datasets are first described, performance assessment is then presented and the main conclusions are finally summarized.

### 4.5.1 Dataset description

Since large-scale datasets from operational networks with NS are not available yet, datasets have been created with a dynamic system-level simulator emulating the activity of a LTE-Advanced cellular network with NS functionality. The simulation tool is thoroughly described in appendix [A](#). Among the two networks implemented in the simulator, network A (the largest one in terms of spatial dimension and cells) is selected, comprising 108 irregular cells in urban and sub-urban areas covering  $11 \times 23$  km<sup>2</sup>. VoIP, video streaming, web browsing and file download services are considered. Regarding user speed, 70% of UEs are static (e.g., indoor users), whereas the remaining 30% are pedestrians. The rest of simulation parameters are those in table [A.1](#).

The following three NS set-ups (referred to as NS scenarios) are considered:

- 1) Scenario with single-service slices (NS\_SS): in this scenario, all cells in the network allocate four slices, which remain active for the whole simulation. Each slice exclusively offers a single service (i.e., VoIP, video, file download or web browsing). This scenario is representative of a system where the MNO creates slices optimized to fulfill certain service requirements of specific clients or those of OTT service providers.
- 2) Scenario with multi-service slices (NS\_MS): in this scenario, there are also four slices whose operation areas cover the whole network. However, unlike in the previous case, all slices offer all services, emulating a network with 4 virtual MNOs operating on different slices over the same infrastructure.
- 3) Scenario without NS (noNS): a legacy network scenario where all UEs share the available bandwidth.

To generate datasets, 8 simulations with different traffic intensities have been performed for each of the three above-described NS set-ups and for two different system bandwidths (5 and 10 MHz), for a total of 48 simulations (=2 bandwidths  $\times$  8 traffic intensities  $\times$  3 NS scenarios). Relative UE spatial distribution and traffic mix per cell remain constant across simulations, whereas the UE generation rate per cell is altered to control traffic intensity. Specifically, the UE generation rate of a cell  $c$  in simulation  $i$ ,  $\lambda_i(c)$ , is computed as

$$\lambda_i(c) = k_i \lambda_{real}(c), \quad (4.4)$$

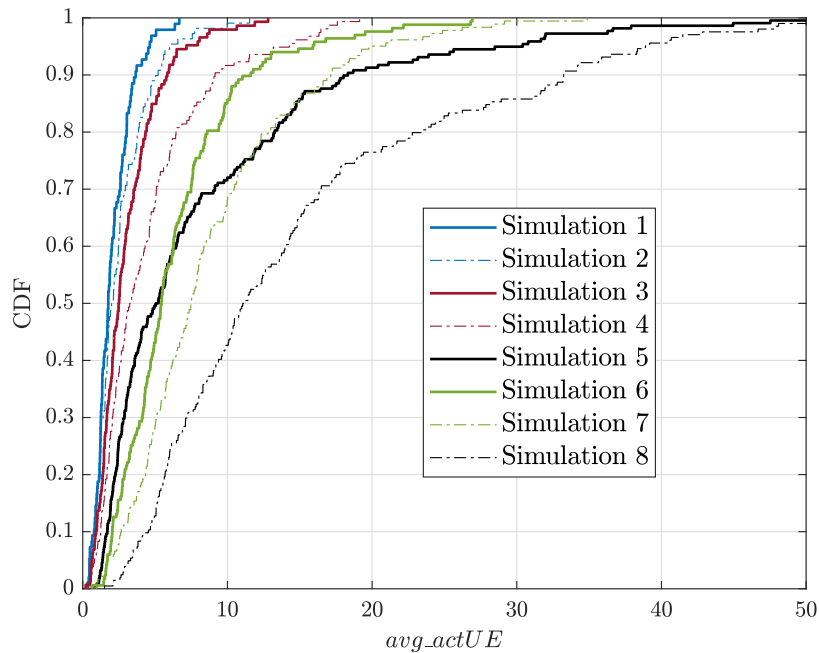


Figure 4.7: Cumulative distribution function of number of active UEs across simulations – NS\_SS scenario.

where  $\lambda_{real}(c)$  is the UE arrival rate of cell  $c$  in the live network and  $k_i$  modulates traffic intensity per simulation, ranging from 0.125 (in the simulation with the lowest traffic intensity for a 5-MHz system bandwidth) to 10 (in the simulation with the largest traffic intensity for a 10-MHz system bandwidth).

A single simulation reflects 15 minutes of network activity (i.e., typical ROP) emulated with a 10-ms time resolution to reduce computational load. To avoid the transient effects of a cold start, a longer period is simulated and statistic collection starts once the adaptive capacity broker has reached a steady state. As an example of the diverse network conditions considered, Fig. 4.7 shows the CDF of the number of active simultaneous UEs,  $avg\_activeUE$ , obtained per simulation in NS\_SS scenario with system bandwidth of 10 MHz. Each line comprises 108 points reflecting the average number of UEs per cell during 15 minutes of network time. It is clearly observed that cells in the scenario are unevenly loaded.

CTRs, CMs and PMs are gathered during simulations. CMs and PMs are collected on a cell basis (i.e., a value per cell) in all scenarios, and on a slice basis (i.e., a value per cell and slice) in NS scenarios. Data is grouped into five datasets depending on the scenario and granularity (i.e., cell or slice). These datasets are denoted as NS\_SS\_cell, NS\_MS\_cell, noNS\_cell, NS\_SS\_slice and NS\_MS\_slice, where the prefix denotes the NS

Table 4.9: Candidate features for estimating DL cell throughput in sliced networks.

	Feature name	Description
Quality	$avg\_CQI$	Average DL CQI in the cell
	$CQI\_class\_p50$	Median DL CQI in the cell
	$CQI\_class\_p5$	5th-tile of DL CQI distribution in the cell
Traffic	$avg\_actUE$	Avg. number of active UEs per TTI in DL in the cell
	$PRButil\_rat$	PRB utilization ratio in PDSCH in the cell
	$s\_UE\_rat \forall s \in \{VoIP, video, ftp, web\}$	Ratio of UEs in the cell demanding each service $s$ offered in the network
CMs	$cell\_BW$ [MHz]	Cell bandwidth
	$nPRB\_i \forall i = 1, 2, \dots, N_{slices}$	Number of PRBs allocated per slice in the cell in DL (only for NS scenarios)

Table 4.10: Candidate features for estimating DL slice throughput in sliced networks.

	Feature name	Description
Quality	$avg\_CQI\_slice$	Avg. DL CQI in the cell for UEs in the slice
	$CQI\_class\_p50\_slice$	Median DL CQI in the cell for UEs in the slice
	$CQI\_class\_p5\_slice$	5th-tile of DL CQI in the cell for UEs in the slice
Traffic	$avg\_actUE\_slice$	Avg. no. of active UEs per TTI in DL served by the slice in the cell
	$PRButil\_rat\_slice$	PRB utilization ratio in PDSCH considering only those PRBs allocated to the slice
	$s\_UE\_rat\_slice \forall s \in \{VoIP, video, ftp, web\}$	Ratio of UEs in the cell served by the slice demanding each service $s$ offered in the network
CMs	$cell\_BW$ [MHz]	Cell bandwidth
	$nPRB\_slice$	No. of PRBs allocated to the slice in the cell

setting (i.e., noNS, NS\_SS or NS\_MS) and the suffix indicates if the dataset contains cell-level or slice-level data.

Each cell-level dataset contains 1,728 datapoints (2 bandwidths  $\times$  8 simulations  $\times$  108 cells) with the following information:

- a) Simulation index.
- b) Cell identifier ( $cell\_ID$ ).
- c) DL cell throughput,  $TH_{cell}$ , defined in section [4.3.1](#), as the target KPI to be estimated.
- d) The set of 14 features shown in Table [4.9](#), as candidate input features.

Likewise, each slice-level dataset is made of 6,912 datapoints (i.e., 2 BWs  $\times$  8 simulations  $\times$  108 cells  $\times$  4 slices) including the following information:

- a) Simulation index.
- b) Cell identifier ( $cell\_ID$ ).
- c) Slice identifier ( $slice\_ID$ ).
- d) DL slice throughput,  $TH_{slice}$ , defined in section 4.3.1, as KPI to be estimated.
- e) The set of 10 features presented in Table 4.10, as candidate input features.

Features in both tables 4.9 and 4.10 are similar, but aggregated at different levels (i.e., a datapoint per cell in Table 4.9, a datapoint per cell and slice in Table 4.10). The considered set of predictors includes a) general configuration parameters (i.e., cell bandwidth), b) for NS scenarios, NS-related configuration parameters (i.e., number of PRBs allocated per slice), c) performance metrics related to spectral efficiency (i.e., CQI indicators) and d) traffic indicators (i.e., traffic mix and number of active UEs). All these features can be computed from cell-level or cell-slice-level CMs/PMs except traffic mix features, which must be derived by aggregating information in connection traces on a cell or cell-slice basis.

Tables 4.11 and 4.12 present a statistical summary of the 3 cell-level datasets (noNS\_cell, NS\_SS\_cell and NS\_MS\_cell) and 2 slice-level datasets (NS\_SS\_slice and NS\_MS\_slice), respectively. For each dataset, information includes the number of samples, and mean, standard deviation, maximum value and minimum value of the input and output features. A rough inspection of the table shows that the variance of all input features is large to better capture the impact of each predictor on the target variable. Note that  $nPRB_1$  to  $nPRB_4$  is the number of PRBs assigned to the slice of each service in NS\_SS, and to the slice of each MNO in NS\_MS. These figures are obtained with the adaptive capacity broker defining the share of bandwidth among slices in the steady state, which assigns each PRB in a cell to a specific slice. Thus, for a subcarrier spacing of 180 kHz, the sum of PRBs assigned to all slices in a cell,  $\sum_{s=1}^4 nPRB_s$ , is 25 and 50 for a system bandwidth of 5 MHz and 10 MHz, respectively.

## 4.5.2 Performance assessment

This section presents the comparative analysis of SL algorithms carried out over the above-introduced datasets. For clarity, analysis set-up is first explained. Then, results are presented, broken down per experiment. Finally, computational complexity is discussed.

Table 4.11: Statistics of cell-level datasets used to estimate DL cell throughput.

Dataset name	noNS_cell				NS_SS_cell				NS_MS_cell			
	Mean	Std. deviation	Min.	Max.	Mean	Std. deviation	Min.	Max.	Mean	Std. deviation	Min	Max
No. datapoints	1,728											
<i>avg_CQI</i>	10.46	2.85	1.97	15.99	10.84	2.63	3.20	15.97	10.45	2.61	3.28	15.87
<i>CQI_class_p50</i>	10.53	3.45	1	16	11.01	3.31	2	16	10.48	3.29	1	16
<i>CQI_class_p5</i>	5.26	3.08	1	16	5.64	2.87	1	16	5.22	2.75	1	15
<i>avg_actUE</i>	8.24	24.03	0.26	55.63	12.00	25.18	0.48	58.76	14.32	25.40	0.65	62.42
<i>PRButil_rat</i>	0.57	0.33	0.01	1	0.43	0.27	0.02	1	0.55	0.30	0.01	1
<i>voip_UE_rat</i>	0.22	0.09	0.04	0.59	0.22	0.09	0.04	0.59	0.22	0.09	0.04	0.59
<i>video_UE_rat</i>	0.25	0.10	0.05	0.67	0.25	0.10	0.05	0.67	0.25	0.10	0.05	0.67
<i>ftp_UE_rat</i>	0.26	0.10	0.02	0.67	0.26	0.10	0.02	0.67	0.26	0.10	0.02	0.67
<i>web_UE_rat</i>	0.27	0.11	0.03	0.69	0.27	0.11	0.03	0.69	0.27	0.11	0.03	0.69
<i>cell_BW</i> [MHz]	7.5	2.5	5	10	7.5	2.5	5	10	7.5	2.5	5	10
<i>nPRB_1</i>	-	-	-	-	3.74	1.12	3	8	9.99	4.31	3	24
<i>nPRB_2</i>	-	-	-	-	20.45	8.19	4	40	10.05	5.09	3	27
<i>nPRB_3</i>	-	-	-	-	5.78	4.68	3	22	9.76	5.17	4	28
<i>nPRB_4</i>	-	-	-	-	8.81	8.55	3	38	9.54	4.82	3	29
<i>TH<sub>cell</sub></i> [kbps]	14894.13	8119.25	1352.95	43882.93	8174.02	5088.90	393.24	29845.14	9973.97	6314.22	830.29	35819.48

Table 4.12: Statistics of slice-level datasets used to estimate DL slice throughput.

Dataset name	NS_SS_slice				NS_MS_slice			
No. datapoints	6,912				6,912			
Statistic	Mean	Std. deviation	Min.	Max.	Mean	Std. deviation	Min.	Max.
<i>avg_CQI_slice</i>	11.06	3.30	2	16	10.79	3.26	1.22	16
<i>CQI_class_p50_slice</i>	11.09	3.80	1	16	10.78	3.76	1	16
<i>CQI_class_p5_slice</i>	7.46	3.88	1	16	7.12	3.81	1	16
<i>avg_actUE_slice</i>	4.56	9.65	0.13	20.02	4.74	7.42	0.26	20.43
<i>PRButil_rat_slice</i>	0.52	0.34	0.05	1	0.54	0.35	0.06	1
<i>voip_UE_rat_slice</i>					0.21	0.22	0	1
<i>video_UE_rat_slice</i>					0.25	0.23	0	1
<i>ftp_UE_rat_slice</i>					0.25	0.24	0	1
<i>web_UE_rat_slice</i>					0.27	0.24	0	1
<i>cell_BW</i> [MHz]	7.5	2.5	5	10	7.5	2.5	5	10
<i>nPRB_slice</i>	10.94	9.57	3	40	9.89	4.82	3	29
<i>TH_slice</i> [kbps]	5389.22	5763.09	21.01	35353.03	4370.87	3047.14	16.35	21772.30

### a) Analysis set-up

The following three experiments are carried out:

- 1) *Experiment 1 – preliminary correlation analysis*: this experiment aims to justify the need for deriving specific models to estimate DL cell throughput in NS scenarios. For this purpose, the average Spearman’s rank correlation value,  $\rho$ , among several candidate input features and  $TH_{cell}$  is compared in noNS\_cell, NS\_SS\_cell and NS\_MS\_cell datasets. Recall that  $\rho$  assesses the strength and direction of monotonic association (whether linear or not) between two variables [188]. Significant changes in the correlation between predictors and response variable in different scenarios would point out the need to derive separate performance estimation models.
- 2) *Experiment 2 – estimation of cell throughput in NS scenarios*: the goal of this experiment is to assess the performance of SL algorithms to estimate  $TH_{cell}$  from predictors in Table 4.9 in the two considered NS scenarios (single-service and multi-service slices). For this purpose, the estimation methodology presented in section 4.3 is applied over NS\_SS\_cell and NS\_MS\_cell datasets.
- 3) *Experiment 3 – estimation of slice throughput in NS scenarios*: this experiment is similar to experiment 2, but aimed to estimate slice throughput from features in slice-level datasets (i.e., NS\_SS\_slice and NS\_MS\_slice). In each scenario, the slice-level model is trained with datapoints from all slices and cells (single output



model shared by all slices). This model is then exploited on a cell and slice basis.

In experiments 2 and 3, for each dataset, 70% of datapoints are used for training and the remaining 30% are used for test. Seven regression algorithms are compared: SVR, KNN, XGBoost, AdaBoost, RF, a shallow MLP (SMLP) and a Deep MLP (DMLP) with 2 hidden layers. In both ANNs, the number of neurons in hidden layers is established through grid search. Two models are derived with each regression algorithm: a) a FULL model with all candidate predictors and b) a simplified model with a subset of input features selected by RFE (hereafter, RFE model). Compared to the analysis presented in section 4.4, MLR and FS-COR models have been discarded due to their poor performance. Likewise, since RF showed acceptable accuracy in that analysis, two additional algorithms based on DTs have been considered, namely AdaBoost and XGBoost. These algorithms are implemented with *scikit-learn* and *XGBoost* Python libraries, respectively [185] [43]. Thus, a total of 56 models (=7 algorithms  $\times$  2 outputs  $\times$  2 scenarios  $\times$  2 feature sets) are tested in total (considering the RFE process as a single model).

Both hyperparameter optimization and RFE are executed with 5-fold cross validation considering the average error,  $MAE$ , as loss metric.  $MAE$  is defined as

$$MAE = \frac{1}{N_s} \sum_{i=1}^{N_s} |\hat{y}(i) - y(i)|, \quad (4.5)$$

where  $N_s$  is the number of samples in the dataset, and  $y(s)$  and  $\hat{y}(i)$  are the measured and estimated values of the output feature in sample  $i$ , respectively. In the RFE process,  $N_f^{opt}$  for a given SL algorithm is the minimum number of predictors achieving similar performance to the most accurate model (i.e., a difference of both  $mAPE$  and  $MANE$  lower than 2% in absolute terms).

Two FoMs are used to assess model performance in terms of accuracy. The first one is the median value of Absolute Percentage Error,  $mAPE$ , computed as

$$mAPE [\%] = \text{median} \left( 100 \cdot \left| \frac{\hat{y}(i) - y(i)}{y(i)} \right| \right) \quad \forall i \in \{1, 2, \dots, N_s\}. \quad (4.6)$$

The median (and not the mean) operation has been used to avoid that insignificant absolute errors in datapoints with low throughput lead to very high and misleading percentage errors (in section 4.4, this effect was negligible due to the selection of datapoints from busy hours). The second FoM is the Mean Absolute Error Normalized to

the maximum theoretical throughput in the cell/slice,  $MANE$ , defined as

$$MANE [\%] = \frac{1}{N_s} \sum_{i=1}^{N_s} (100 \cdot \left| \frac{\hat{y}(i) - y(i)}{TH_{max}(k_i)} \right|), \quad (4.7)$$

where  $TH_{max}(k_i)$  is the maximum achievable throughput in entity  $k$  (i.e., cell/slice) to which datapoint  $i$  belongs,  $k_i$ .  $TH_{max}(k_i)$  is computed assuming that all PRBs in a cell or slice are allocated to UEs with maximum CQI (i.e., 15). With the modulation and coding schemes considered in the simulator, the peak throughput is 1 Mbps per PRB [189]. Under this assumption,  $TH_{max}(k_i)$  can be computed from  $cell\_BW$  for cells, and from  $n\_PRB\_s$  for slices. Note that datapoints in the considered datasets have different system bandwidths, given by  $cell\_BW$  and  $n\_PRB\_slice$  in cell-level and slice-level datasets, respectively. These features limit the capacity of the entity (i.e., cell or cell-slice), and, thus, FoMs relying on absolute error (e.g., mean absolute error) would be dominated by datapoints from entities with the highest capacity, which is undesirable. The normalization performed in  $MANE$  circumvents this issue. As shown in (4.7),  $MANE$  is expressed as a percentage for an easier interpretation.

Following the same rationale as in the analysis over non-sliced networks, a model is considered acceptable to estimate DL cell/slice throughput if  $mAPE < 10\%$  and  $MANE < 10\%$ . For robustness, the best model for each output feature is selected as follows. First, models providing  $MANE$  similar to the most accurate model (i.e., difference lower than 1% in absolute terms) are selected as candidates. Then, models with a difference in  $mAPE$  higher than 1% compared to the best  $mAPE$  among candidates are discarded. Finally, the model with the lowest number of features among the remaining candidates is selected as best model. If several models satisfy this condition, the best model is that providing the best results for the worst samples (i.e., lowest 90<sup>th</sup> percentile of Absolute Normalized Error,  $ANE$ , computed as in (4.7)).

## b) Results – correlation analysis

Table 4.13 shows the average value of Spearman's correlation coefficient, computed between  $cell\_BW$ ,  $PRB_{util\_rat}$  and  $avg\_CQI$  and  $TH_{cell}$  in noNS-cell, NS\_SS-cell and NS-cell datasets (e.g., average correlation between  $cell\_BW$  and  $TH_{cell}$  in noNS-cell dataset is 0.59). Those features provide information about cell resources, radio resource utilization and spectral efficiency, respectively. For NS scenarios, the correlation between the number of PRBs allocated per slice,  $nPRB\_k \forall k \in [1, 4]$ , and  $TH_{cell}$

Table 4.13: Correlation between candidate input features and DL cell throughput in different NS scenarios.

KPI	noNS	NS_SS	NS_MS
<i>cell_BW</i>	0.59	0.69	0.70
<i>PRButil_rat</i>	0.04	0.32	0.36
<i>avg_CQI</i>	0.66	0.14	0.12
<i>nPRB_1</i>	–	0.66	0.67
<i>nPRB_2</i>	–	0.48	0.69
<i>nPRB_3</i>	–	0.55	0.68
<i>nPRB_4</i>	–	0.67	0.70

is also included as a metric of spectrum split. It is observed that *cell\_BW* is significantly correlated with cell throughput in all scenarios (i.e.,  $\rho \geq 0.59$ ). In contrast, the correlation of *PRButil\_rat* is much higher in NS scenarios than in noNS scenario (i.e.,  $\rho = 0.32$  and  $0.36$  in NS\_SS and NS\_MS scenarios, respectively, compared to  $0.04$  in noNS scenario). Likewise, *avg\_CQI* is correlated with cell throughput in noNS scenario (i.e.,  $\rho = 0.66$ ), but not in NS\_SS and NS\_MS scenarios ( $\rho = 0.14$  and  $0.12$ , respectively). These differences confirm that enabling the NS feature changes the relationships among network indicators, as anticipated in section 4.2, revealing the need for creating new performance models for NS scenarios.

It is also remarkable that similar correlation values are obtained in both NS scenarios for all features but *nPRB\_k*. This feature presents similar correlation values in all slices in NS\_MS scenario (slices offering a service mix), but not in NS\_SS scenario (slices offering a single service). To capture these peculiarities, specific performance models are derived per scenario in experiments 2 and 3.

### c) Results – cell throughput estimation

Tables 4.14 and 4.15 break down results obtained when estimating  $TH_{cell}$  in NS\_SS and NS\_MS scenarios, respectively. Performance from FULL models is first analyzed. KNN is the worst algorithm for both scenarios, with unacceptable  $mAPE$  values. XGBoost is the best ensemble method, whereas both ANNs (SMLP and DMLP) perform similarly. In NS\_SS scenario, XGBoost and ANNs are the best FULL models, with  $mAPE < 7\%$  and  $MANE < 2\%$ . In contrast, in NS\_MS scenario, ANNs outperform the rest of algorithms, with  $mAPE < 5\%$  and  $MANE < 1.5\%$ . When comparing scenarios, similar performance (i.e., differences smaller than 2% in absolute terms) can be obtained to estimate  $TH_{cell}$  in NS\_SS and NS\_MS scenarios with ANNs.

Table 4.14: Model performance for estimating DL cell throughput in single-service NS scenario (NS\_SS).

Model	FULL		RFE		
FoM	$mAPE$	$MANE$	$N_f^{opt}$	$mAPE$	$MANE$
SVR	8.97	2.33	3	8.02	1.96
KNN	15.06	3.88	3	7.74	2.01
XGBoost	6.30	1.86	3	7.83	2.32
AdaBoost	8.91	2.14	4	8.34	2.15
RF	8.75	2.23	4	7.61	2.01
SMLP	6.47	1.71	3	7.96	1.89
DMLP	6.62	1.72	3	7.63	1.96

Table 4.15: Model performance for estimating DL cell throughput in multi-service NS scenario (NS\_MS).

Model	FULL		RFE		
FoM	$mAPE$	$MANE$	$N_f^{opt}$	$mAPE$	$MANE$
SVR	6.94	1.97	8	6.41	1.85
KNN	14.19	4.23	4	6.83	2.06
XGBoost	6.65	1.92	4	6.41	1.80
AdaBoost	7.03	2.07	9	7.03	1.96
RF	7.04	2.02	5	6.30	1.84
SMLP	4.93	1.40	4	5.00	1.38
DMLP	4.88	1.35	5	5.41	1.57

Fig. 4.8 and 4.9 show  $MANE$  obtained across RFE process in NS\_SS and NS\_MS scenarios, respectively. For better visualization, only SVR, KNN and the best ANN and ensemble method are included.  $mAPE$  evolution, not shown here for brevity, is very similar. Again, KNN suffers the curse of dimensionality. Such a phenomenon is also observed in SVR for  $N_f \in \{3, \dots, 7\}$  in Fig. 4.9.

The last three columns in Table 4.14 summarize FoMs with the best RFE model for all tested algorithms in NS\_SS scenario.  $N_f^{opt}$  is selected with the convergence criteria described above, resulting  $N_f^{opt}=3$  for all algorithms but AdaBoost and RF, with  $N_f^{opt}=4$ . RFE models requiring 3 input features (i.e., all but AdaBoost and RF) are considered potential candidates to estimate  $TH_{cell}$ . Since the accuracy of both ANNs is similar and SMLP is faster to train and less prone to overfitting than DMLP, DMLP is discarded. For a deeper analysis, Fig. 4.10 depicts the CDF of  $ANE$  obtained with the remaining candidate models. All algorithms perform similarly in the lower part of the CDF, whereas SMLP performs best in the upper part, showing the lowest error. Thus, RFE-SMLP is considered the best model. The selected input features (ranked by relevance) are  $PRButil\_rat$ ,  $avg\_CQI$  and  $cell\_BW$ .

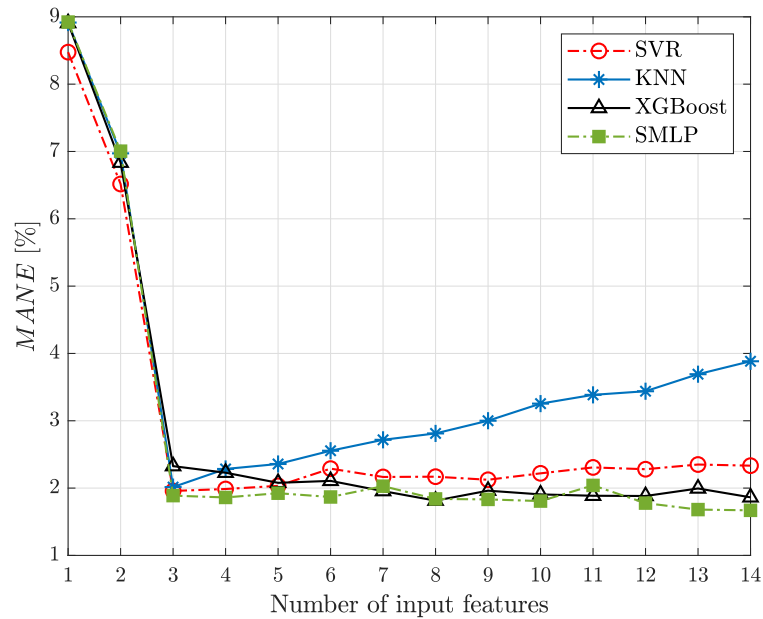


Figure 4.8:  $MANE$  evolution across RFE process when estimating DL cell throughput in single service NS scenario (NS\_SS).

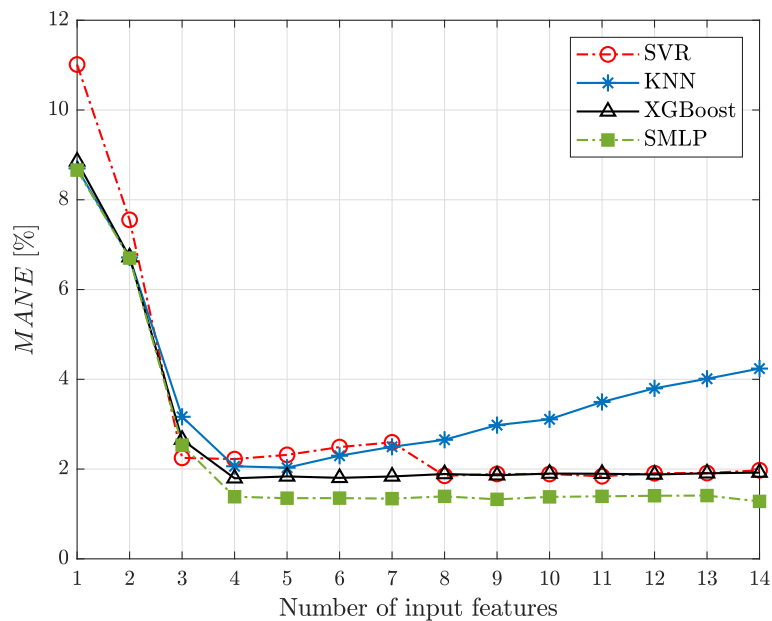


Figure 4.9:  $MANE$  evolution across RFE process when estimating DL cell throughput in multi-service NS scenario (NS\_MS).

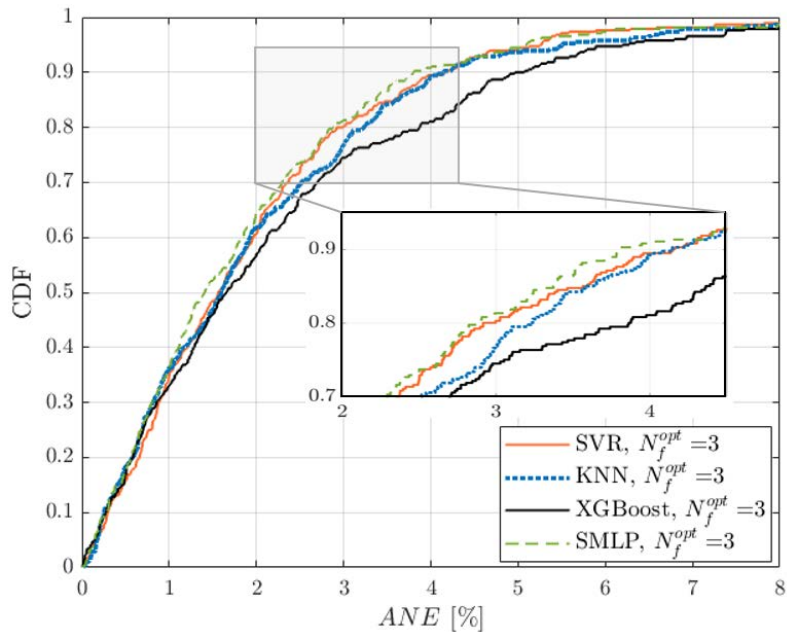


Figure 4.10: Distribution of absolute normalized error for best models when estimating DL cell throughput in single-service NS scenario (NS\_SS).

Similarly, the last three columns in Table 4.15 summarize FoMs obtained with the best RFE models in NS\_MS scenario. Unlike in NS\_SS scenario, the algorithms now have completely different  $N_f^{opt}$ , ranging from 4 to 9. The best model is RFE-SMLP, with the highest accuracy ( $mAPE=5\%$  and  $MANE=1.38\%$ ) and the lowest number of input features ( $N_f^{opt}=4$ ). Not shown in the table is the fact that some of the relevant input features are also different, with  $PRButil\_rat$ ,  $CQI\_class\_p50$ ,  $cell\_BW$  and  $CQI\_class\_p5$  (listed by relevance). The significant decrease in  $MANE$  for SMLP from  $N_f=3$  to  $N_f=4$  observed in Fig 4.9 confirms that, in NS\_MS scenario, the inclusion of spectral efficiency of cell-edge users through  $CQI\_class\_p5$  improves SMLP performance. Such an effect may be due to joint packet scheduling for users demanding different services in each slice, which favors cell-edge users from services with strict delay requirements (e.g., VoIP) at the expense of those with better channel conditions from services with loose delay constraints, decreasing cell throughput even with a high  $PRButil\_rat$ .

From the above results, it can be concluded that SMLP is an adequate SL algorithm to estimate cell throughput in both NS scenarios, providing acceptable accuracy with models requiring few input features. With an adequate FS, similar accuracy can be obtained for estimating  $TH_{cell}$  in NS scenarios with single-service or multi-service slices, with an error lower than 2% of the achievable cell throughput. It should be pointed out

that, in both scenarios, the input features to RFE–SMLP model are similar to those in classical models, shown in (4.1). Those features can be computed from PMs/CMs stored in the OSS by most network operators. Thus, it is unnecessary to store additional NS-specific data or collect connection traces to estimate cell throughput in NS scenarios. A deeper analysis of data shows that, in both scenarios, the worst data-points (i.e., those with the highest  $ANE$ ) correspond to underutilized cells with high spectral efficiency (i.e.,  $PRButil\_rat < 40\%$ ,  $avg\_CQI \geq 10$ ), where  $TH_{cell}$  tends to be underestimated.

#### d) Results – slice throughput estimation

Tables 4.16 and 4.17 summarize results when estimating  $TH_{slice}$  in NS\_SS and NS\_MS scenarios, respectively. Likewise, Fig. 4.11 and 4.12 present the evolution of  $MANE$  across RFE for all algorithms in these scenarios (DMLP lines have been omitted for a better visualization, since they overlap with SMLP).

For FULL models in both scenarios, only algorithms based on DTs and ANNs fulfill the accuracy threshold of 10% for both  $mAPE$  and  $MANE$ . RF provides the best FULL model in NS\_SS scenario ( $mAPE=5.53\%$  and  $MANE=2.57\%$ ), whereas both ANNs get the best results in NS\_MS scenario ( $mAPE\approx 8.85\%$  and  $MANE\approx 5.15\%$ ). When comparing FULL and RFE models for each scenario, it is observed that, again, for a given algorithm, similar performance can be obtained with simpler models with less input variables.

Regarding RFE, Fig. 4.11 and 4.12 reveal that KNN is again suffering the curse of dimensionality, since accuracy diminishes when increasing the number of features above  $N_f=3$ . It is also remarkable that the evolution of SVR performance in NS\_MS scenario, with  $N_f^{opt}=4$ , differs significantly from NS\_SS scenario, where only one feature can be extracted for an acceptable model performance (i.e.,  $MANE$  increases significantly below  $N_f=10$ ). Nonetheless, unlike when predicting  $TH_{cell}$ , SVR is not competitive with other SL algorithms when predicting  $TH_{slice}$ .

When considering a trade-off between accuracy and input size, RFE–RF is the best model in NS\_SS scenario ( $mAPE=6.26\%$  and  $MANE=2.75\%$  with  $N_f^{opt}=5$ ), followed by RFE–SMLP. In NS\_MS scenario, RFE models built with ANNs show the best accuracy (i.e.,  $mAPE\approx 9\%$  and  $MANE\approx 5.35\%$ ) and required information ( $N_f^{opt}=5$ ). Fig. 4.13 represents the CDFs of  $ANE$  obtained with these models. RFE–XGBoost model (i.e., the next model with better  $MANE$ ) is also included. A significant

Table 4.16: Model performance for estimating DL slice throughput in single-service NS scenario (NS\_SS).

Model	FULL		RFE		
FoM	$mAPE$	$MANE$	$N_f^{opt}$	$mAPE$	$MANE$
SVR	14.40	8.30	10	12.64	7.81
KNN	14.56	9.84	3	8.92	3.96
XGBoost	7.37	2.87	4	8.16	3.05
AdaBoost	6.44	2.86	7	8.15	3.21
RF	5.53	2.57	5	6.26	2.75
SMLP	7.70	3.27	4	7.80	2.97
DMLP	6.78	2.83	5	9.63	4.46

Table 4.17: Model performance for estimating DL slice throughput in multi-service NS scenario (NS\_MS).

Model	FULL		RFE		
FoM	$mAPE$	$MANE$	$N_f^{opt}$	$mAPE$	$MANE$
SVR	12.47	7.32	4	12.51	7.22
KNN	16.30	9.73	3	12.08	6.99
XGBoost	9.64	5.58	6	9.46	5.53
AdaBoost	10.24	5.77	8	9.86	5.53
RF	9.34	5.48	6	9.16	5.70
SMLP	8.82	5.10	5	8.78	5.25
DMLP	8.87	5.17	5	9.23	5.45

improvement of ANNs over XGBoost is observed for the largest error percentiles. Among ANNs, RFE–SMLP is the best option (lines are shifted to the left compared to RFE–DMLP). The input features in the best models (i.e., RFE–RF for NS\_SS scenario and RFE–SMLP for NS\_MS scenario) are  $cell\_BW$ ,  $nPRB\_slice$ ,  $PRButil\_rat\_slice$ ,  $CQI\_class\_p50\_slice$  and  $voip\_UE\_rat\_slice$  in both scenarios.

Table 4.18 shows the results for the best models (i.e., RFE–RF in NS\_SS scenario and RFE–SMLP in NS\_MS scenario) broken down per slice. Recall that, in NS\_SS scenario, slices 1 to 4 serve users demanding VoIP, video, file download and web browsing, respectively. In contrast, in NS\_MS scenario, slices serve a service mix changing with cell and tenant. It can be noticed that differences among slices are larger in NS\_SS than in NS\_MS scenario. For a more detailed analysis, Fig. 4.14 and 4.15 show the probability density function of  $PRButil\_rat\_slice$  and  $slice\_ID$  for 5% of samples with the largest error (i.e., highest  $MANE$ ) for the best models in NS\_SS and NS\_MS scenarios. In NS\_SS scenario,  $slice\_ID$  distribution reveals that more than 80% of worst samples



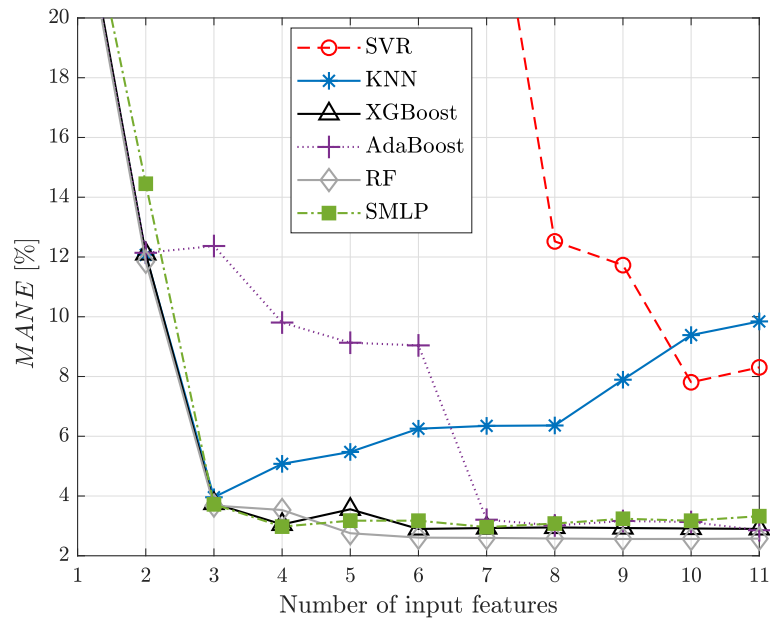


Figure 4.11:  $MANE$  evolution across RFE process when estimating DL slice throughput in single-service scenario (NS\_SS).

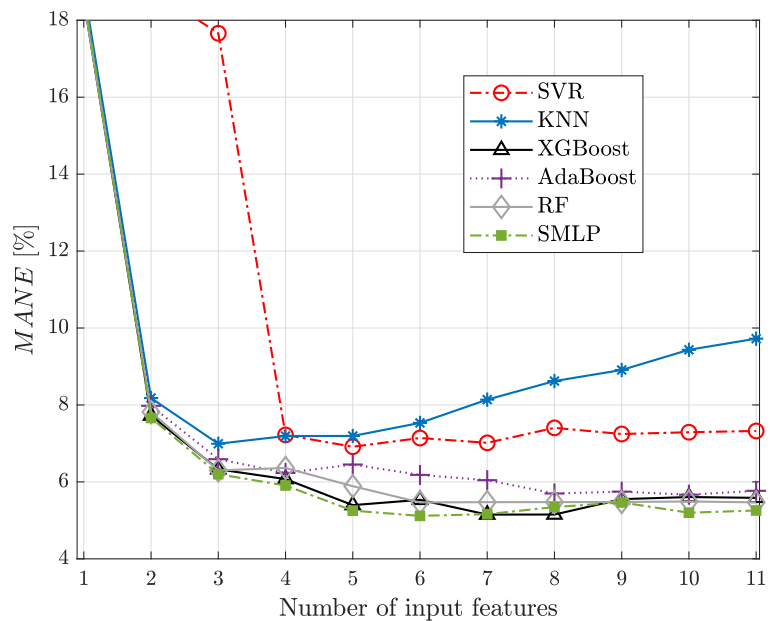


Figure 4.12:  $MANE$  evolution across RFE process when estimating DL slice throughput in multi-service NS scenario (NS\_MS).

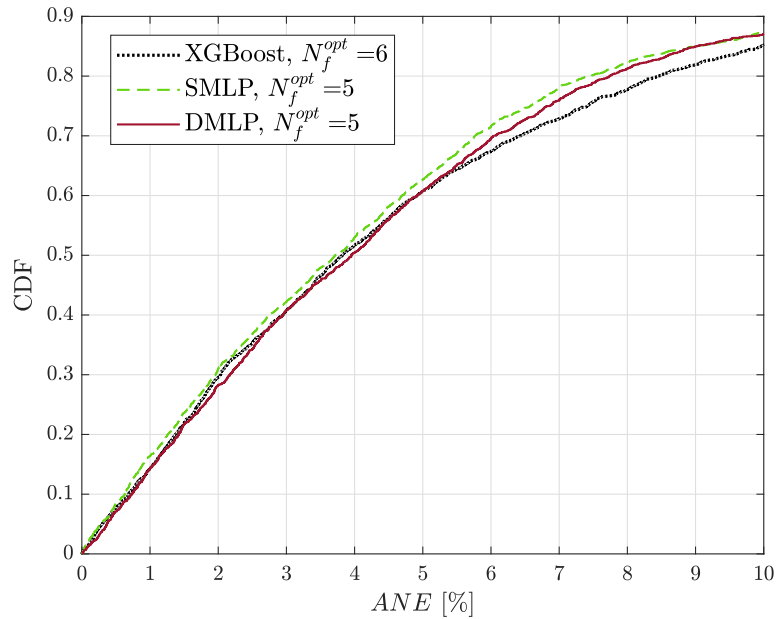


Figure 4.13: Distribution of absolute normalized error for best models when estimating DL slice throughput in multi-service NS scenario (NS\_MS).

Table 4.18: Performance per slice when estimating DL slice throughput with the best model.

Scenario	NS_SS		NS_MS	
Best model	RFE-RF		RFE-SMLP	
FoM	$mAPE$	$MANE$	$mAPE$	$MANE$
Slice 1	14.43	0.38	9.99	5.19
Slice 2	4.82	3.09	9.41	5.34
Slice 3	4.74	3.49	6.97	5.35
Slice 4	5.51	3.86	8.92	5.25

are from slices 3 to 4. Thus, RFE-RF provides the lowest accuracy when estimating throughput from slices serving web and file download users. This behavior is observed in most of the tested SL algorithms. Note that the data rate of users of these best effort services adapts to instantaneous slice capacity (i.e., allocated PRBs) and traffic (i.e., UEs to schedule) in the cell and is thus prone to fluctuate, being more difficult to estimate. This problem may be solved by creating per-service slice-level models at the expense of having less training datapoints per model. In NS\_MS scenario, worst offenders for RFE-SMLP are evenly distributed among slices. However, it is remarkable that 70% of these datapoints belong to slices with  $PRB_{util\_rat\_slice} \leq 20\%$ . Thus, it can be concluded that RFE-SMLP is less accurate when predicting the aggregate throughput of underutilized slices, no matter the slice service mix.

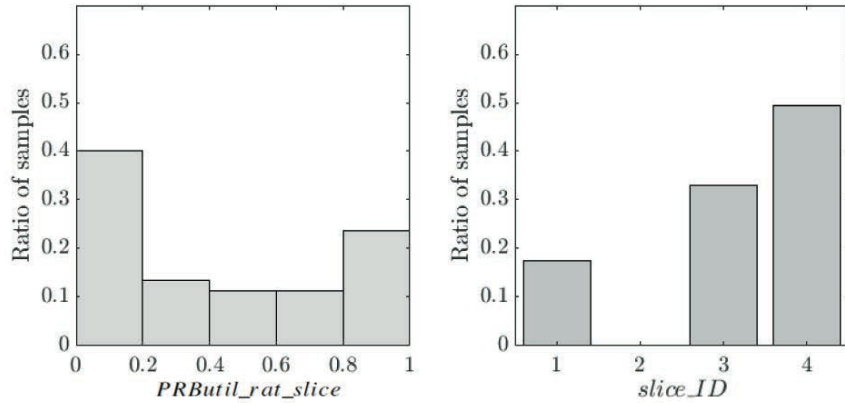


Figure 4.14: Values of features in samples with the largest error when estimating DL slice throughput with RFE–RF (NS\_SS scenario).

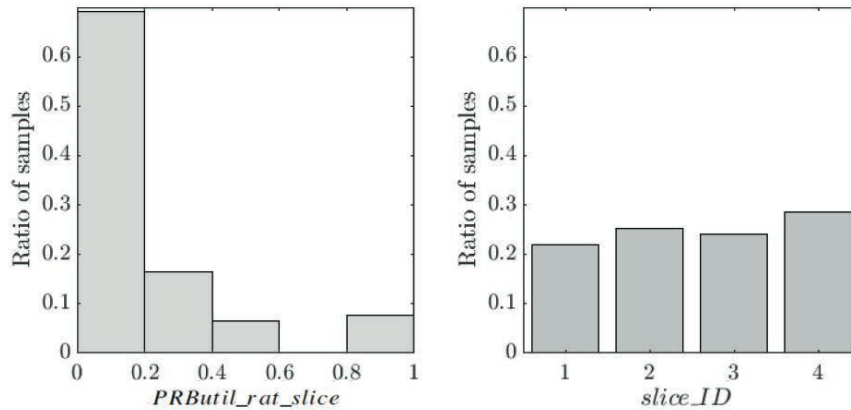


Figure 4.15: Values of features in samples with the largest error when estimating DL slice throughput with RFE–SMLP (NS\_MS scenario).

From the above results, it can be concluded that the best SL algorithm to estimate slice throughput depends on the NS scenario. Moreover, it is worth noting that, unlike for cell throughput, accuracy obtained for slice throughput with the best model is lower in NS\_MS scenario ( $MANE=5.25\%$ ) than in NS\_SS scenario ( $MANE=2.75\%$ ). This may be due to the coexistence of users with services with very different traffic patterns, which makes throughput calculations more complex. Likewise, the RFE process shows that using information about slice service mix improves  $TH_{slice}$  estimates. Specifically,  $voip\_UE\_rat\_slice$  has been selected as a key feature in this analysis. Note that VoIP is the service with the lowest data rate in the considered scenario. Thus, this feature provides information about the ratio of data-hungry UEs in the slice. This information may be useful to estimate throughput since bursty traffic degrades network spectral efficiency due to last transmission time interval data and outer loop

link adaptation [104]. In scenarios with single-service slices, obtaining slice service mix is straightforward (i.e., no mix). In scenarios with multi-service slices, service mix can be obtained by applying a traffic classification algorithm over radio connection traces even if traffic is encrypted, as explained in chapter 3.

### e) Computational complexity

A thorough discussion on computational complexity when estimating radio throughput indicators from network data has already been presented in section 4.4.2. Thus, this section focuses on particular aspects of NS scenarios.

Regarding data collection and processing, according to the results presented above, service mix information is relevant for slice-level performance models. To this end, in multi-service slice scenarios, connection traces must be collected and processed to obtain this data, which may be time-consuming. If required, parallelization can be used to speed up trace processing.

Table 4.19 shows training time for the FULL models (including hyperparameter optimization) when estimating  $TH_{slice}$  in NS\_SS scenario (i.e., the case with the largest number of datapoints) in a personal computer with Intel Core i7-8700 processor working at 3.2 GHz with a RAM of 16 GB. Training times range from a few seconds to near 9 minutes. Differences are due to algorithm complexity and especially to the varying number of hyperparameters tuned per algorithm. Training time decreases significantly with RFE models. Nonetheless, in case of tight time constraints, hyperparameter optimization can be accelerated via parallelization or relaxed (e.g., running less folds of data). Since the latter option may degrade model performance, the former alternative is preferred when necessary.

Both slice and cell level models must be executed again after any significant change affecting input variables (e.g., change in traffic mix). Moreover, slice-level estimates must be updated if PRB split among slices varies due to SLA violation, SLA redefinition or slice activation/de-activation. Likewise, new events can appear in NS scenarios (e.g., an update of capacity broker policy) altering the relationship between predictors and predicted variable. Thus, model retraining is likely to be more frequent than in legacy (i.e., non-sliced) networks.

Table 4.19: Time complexity of FULL models when estimating DL slice throughput in single-service NS scenario.

Algorithm	Training time [s]
SVR	114
KNN	0.6
XGBoost	458
AdaBoost	13
RF	27
SMLP	164
DMLP	864

### 4.5.3 Conclusions

In 5G radio access networks, the introduction of NS prevents the use of legacy network performance models. In this new scenario, slice-level and cell-level throughput estimates are required for network management purposes, such as cell capacity replanning or spectrum sharing among slices. This section has presented a comprehensive analysis of the performance of 7 well-known SL algorithms for estimating DL cell and slice throughput from CMs, PMs and radio traces collected in the OSS. Performance assessment has been carried out over cell-level and slice-level datasets built with a system-level simulator. This tool a) is dynamic (i.e., a simulation consists of a set of correlated snapshots emulating network activity along time on a 10-ms resolution), b) includes realistic traffic models from 4 different services, c) implements real RRM algorithms from vendors and d) considers a realistic scenario comprising 108 cells with uneven cell service area, traffic density and service mix. All these characteristics allow the creation of realistic datasets supporting the significance of results. Two different NS scenarios have been considered, with single-service slices and multi-service slices.

Results show that, with adequate feature selection, all tested algorithms achieve acceptable accuracy (i.e.,  $MANE$  and  $mAPE$  lower than 10%) when estimating cell throughput in NS scenarios, using similar information to models in non-NS scenarios. Moreover, all algorithms perform similarly in scenarios with single-service and multi-service slices. When considering the trade-off between accuracy and storage capacity, SMLP has shown the best results in both scenarios, with  $MANE=2\%$  and  $mAPE<8\%$ . The best models have at most 4 input features related to bandwidth, radio resource utilization and spectral efficiency. Such features can be computed from PMs/CMs collected on a cell basis in the OSS.

However, only ensemble methods and ANNs achieve acceptable accuracy when estimating slice throughput. Moreover, model accuracy is worse in multi-service slices, where users demand services with highly differing traffic patterns. RFE–RF has shown the best accuracy in single-service NS scenario ( $mAPE=6.26\%$ ,  $MANE=2.75\%$ ), whereas RFE–SMLP has performed best in multi-service NS scenario ( $mAPE=8.78\%$ ,  $MANE=5.25\%$ ). In both cases, the 5 input features in these models not only include indicators computed from PMs/CMs at cell level, but also indicators computed at slice level and information about the service mix per slice that must be derived from connection traces.

Analyses presented in this chapter have shown the potential of SL to estimate several radio throughput metrics in different RATs. It should be pointed out that, for a given analysis, all the considered SL algorithms have been trained with the same size-limited dataset. It can be ensured that models have not overfitted since: a) model architectures (e.g., number of layers/neurons in ANNs) have been chosen by keeping the number of trainable parameters lower than the number of training datapoints, and b) cross-validation has been performed during hyperparameter tuning and feature selection. However, a more extensive dataset may allow making the most of DNNs, increasing accuracy obtained here. Nonetheless, it can be stated that non-deep SL algorithms are the best option when limited data is stored in the OSS, providing an adequate estimation accuracy for (re)dimensioning purposes.

Throughput models proposed here can be used to detect resource overprovisioning, capacity problems or SLA violations. For instance, cell/slice performance in the worst conditions can be checked by setting input features to their worst value (e.g., highest allowed PRB utilization). Moreover, future throughput performance can be predicted by feeding the models with forecasts of input features to assess the impact of candidate replanning actions on network performance.

# Chapter 5

## Long-term cell traffic forecasting

This chapter deals with cell traffic forecasting in the long term (i.e., several months ahead) through SL. In radio planning tools, these traffic forecasts can be compared with cell capacity estimates such as those obtained in chapter 4 to detect potential capacity bottlenecks in advance. Content is organized as follows. Section 5.1 revises related work. Section 5.2 outlines the problem of predicting monthly busy-hour data traffic per cell, highlighting the properties of the time series involved. Section 5.3 outlines the considered forecasting methodology. Section 5.4 presents performance assessment over a dataset from a live cellular network. Finally, section 5.5 exposes the main conclusions.

### 5.1 Related work

Traffic forecasting in telecommunication networks can be treated as a time series analysis problem. The earliest works address circuit-switched traffic prediction by deriving statistical models based on historical data. Linear time series models, such as Auto Regressive Integrated Moving Average (ARIMA), capture trend and short-range dependencies in traffic demand. More complex models, such as seasonal ARIMA [190] [191] and exponential smoothing (e.g., Holt-Winters) [192] [193], include seasonality. To reflect long-range dependencies, these can be extended with non-linear models, such as Generalized Auto-Regressive Conditionally Heteroskedastic (GARCH) [194].

The previous works show that it is possible to predict cellular traffic at different geographical scales (e.g., network operator [190], province [191], cell [192]) and time resolutions (e.g., minutes [194], hourly [192], daily [190], monthly [191]), provided that

traffic is originated by circuit-switched services (e.g., voice and text messages). However, predicting packet data traffic is much more challenging [195]. As pointed out in [193], data traffic is more influenced by abnormal events and changes in network configuration than circuit-switched traffic. In [196], short-term traffic volume in a 3G network is predicted via Kalman filtering. In [197], ARIMA is used to predict the achievable user rate in four cells located in areas with different land use. In [198], application-level traffic is predicted by deriving a  $\alpha$ -stable model for three different service types, and then dictionary learning is implemented to refine forecasts. As an alternative, more modern approaches tackle data traffic forecasting with sophisticated models based on SL to take advantage of massive data collected in cellular networks. Most efforts have been focused on short-term prediction (seconds, minutes) to solve the limitations of Time Series Analysis (TSA) approaches to capture rapid fluctuations of the time series. A common approach is to use deep learning to model the spatiotemporal dependence of traffic demand. The temporal aspect of traffic variations is often captured with recurrent neural networks based on LSTM units [199] [200] [201] [202]. Alternatively, in [203], a deep belief network and a Gaussian model are used to capture temporal dependencies of network traffic in a mesh wireless network. The spatial dependence is captured by different approaches. In [199], the scenario is divided into a regular grid, and a convolutional ANN is used to model spatial traffic dependencies among grid points. A similar approach is considered in [204], where extra branches are added to the ANN for fusing external factors such as crowd mobility patterns or temporal functional regions. In [205], convolutional LSTM units and 3D convolutional layers are fused to encode the spatiotemporal dependencies of traffic carried in the grid points. Alternatively, other authors model spatial dependencies of traffic carried in different cells. In [202], a general feature extractor is used with a correlation selection mechanism for modeling spatial dependencies among cells and an embedding mechanism to encode external information. In [206], the spatial relevancy among cells is modeled with a graph ANN based on the distance among cell towers to deal with an irregular cell distribution. A graph-based approach is also considered in [207], where traffic is decomposed into inter-tower and in-tower traffic components. Deep learning schemes such as recurrent [208] [209] or convolutional ANNs [210] have also been applied to coarser time resolutions (i.e., an hour) to extend the forecasting horizon to several days.

The above works show that advanced deep learning models perform well if data is collected with fine-grained time resolution to build long time series (i.e., thousands



of samples) of correlated data. With these models, advanced dynamic radio resource management schemes and proactive self-tuning algorithms can be implemented [211]. However, some replanning actions (e.g., deployment of a new cell) may take several months to be implemented (e.g., radio frequency planning, site acquisition, civil works, licenses, installation/commissioning, pre-launch optimization...) [212]. Thus, the upcoming traffic must be predicted with much longer time horizons (i.e., several months in advance) [213]. For this purpose, a monthly traffic indicator is often computed per cell from busy-hour measurements, limiting the number of historical data samples used for prediction [214] [215]. Moreover, some studies [216] [217] have shown that the influence of past measurements quickly diminishes after a few weeks due to changes in user trends (e.g., new terminals, new hot spots...) and replanning actions by the operator (e.g., new site, equipment upgrades...). As a consequence, long-term traffic forecasting relies on short and noisy time series, which might prevent operators from using complex deep learning models. As an alternative, it must be checked if simpler SL algorithms outperform the classical TSA approach for time series with these constraints. For this purpose, a large dataset containing data collected per cell for years is required. Such information is a precious asset for operators, which is seldom shared. For this reason, to the authors' knowledge, no recent work has evaluated long-term traffic forecasting in mobile networks considering SL techniques.

This chapter presents a comprehensive analysis assessing the performance of SL against classical TSA schemes for predicting monthly busy-hour data traffic per cell in the long term. For this purpose, a large dataset is collected for 30 months from a live LTE network covering an entire country. All prediction techniques considered here are included in most data analytics packages and have already been used in several fields. Hence, the main novelty is the assessment of well-established SL algorithms for long-term data traffic forecasting based on short and noisy time series taken from current mobile networks offering a heterogeneous service mix. The impact of key design parameters has been checked, namely the data collection window, the prediction horizon and the number of models to be created (one per cell or one for the whole network). Throughout the analysis, algorithms are compared in terms of accuracy and computational complexity.

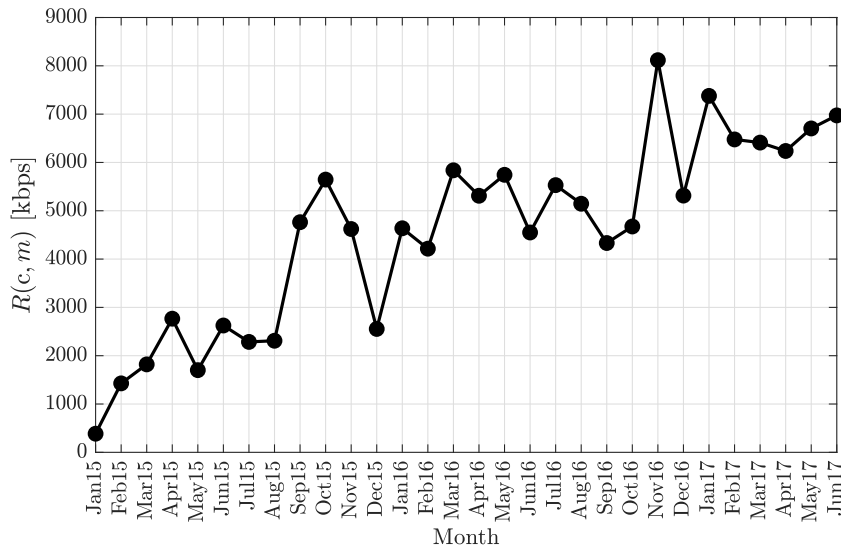


Figure 5.1: Evolution of monthly busy-hour traffic in a cell.

## 5.2 Problem formulation

The problem of forecasting traffic  $R$  carried in a cell  $c$  with a time horizon  $h$  at time  $t$  from historical data gathered during a collection period  $w_c$  can be formulated as

$$\hat{R}(c, t+h) = f(R(c, t), R(c, t-1), \dots, R(c, t-w_c+1)), \quad (5.1)$$

where  $t-w_c+1$  denotes the oldest available datasample.

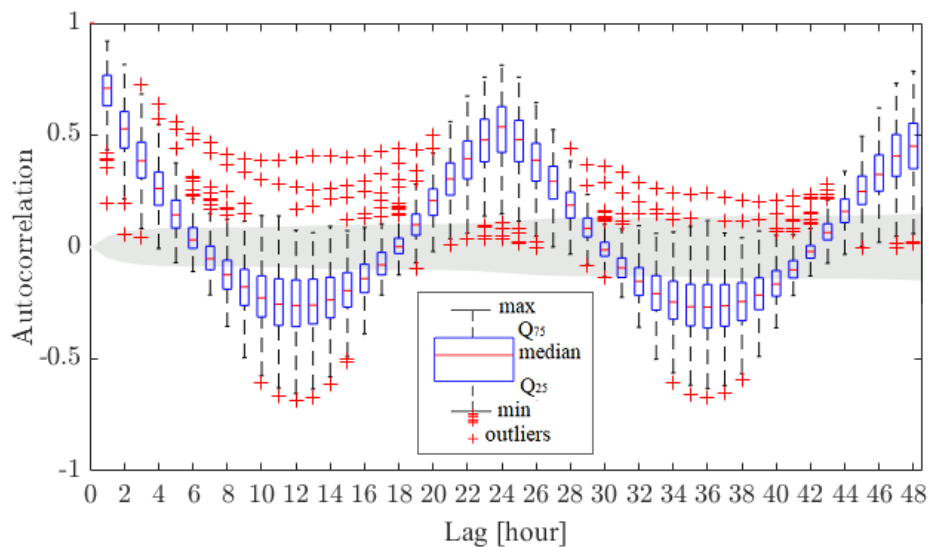
The way to tackle traffic prediction strongly depends on the time granularity of data, which determines two key factors. The first factor is the length of available time series,  $w_s$ . Note that, when training a cell-specific prediction model, the number of observations must be higher than the number of model parameters [218]. Thus, complex deep learning models with hundreds of internal parameters cannot be considered if a short time series (i.e., dozens of samples) is available due to data aggregation on a coarse time resolution (e.g., monthly data).

A second factor is data predictability, which may be degraded by the aggregation operation performed to obtain coarse-resolution time series from finer-resolution data. Such an undesirable effect appears when computing monthly busy-hour statistics from hourly data for RAN dimensioning purposes. To illustrate this fact, Fig. 5.1 shows the evolution of monthly busy-hour traffic in the DL from a live LTE cell. As described later in section 5.4.1, data aggregation is often performed by selecting the busy hour per week and then averaging traffic measurements per week in a month. Weekly and monthly

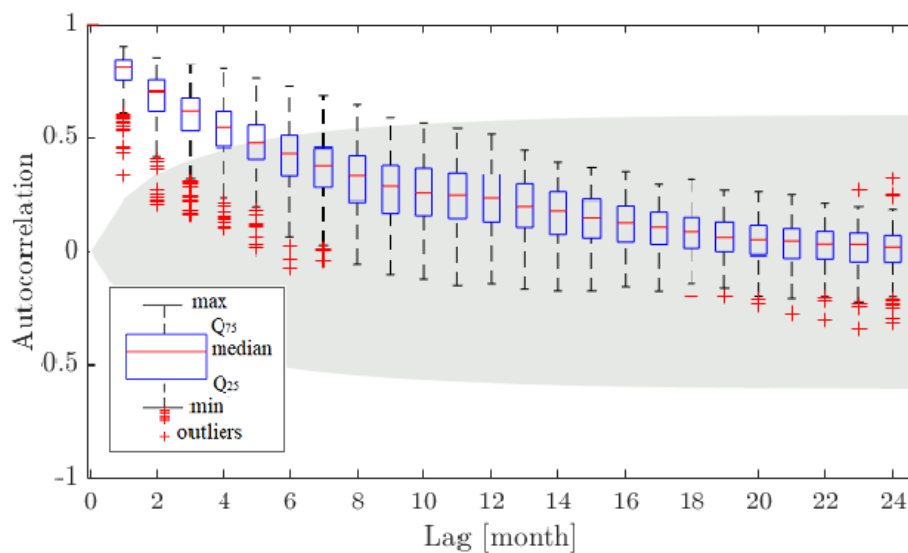
aggregation eliminates hourly/daily fluctuations and the impact of sporadic events (e.g., cell outage, cell barring. . .), leaving only monthly variations needed for detecting capacity issues. This variations consists of: a) a trend component, influenced by the traffic growth rate, b) a seasonal component, given by the month of the year, and c) a remainder component, due to abnormal events taking place locally (e.g., new nearby site, new hot spot. . .) or network-wide (e.g., launch of new terminals, change of network release. . .). In the example of the figure, it is observed that the trend component prevails over the seasonal component, causing the time series to be non-stationary. Also important, network events cause sudden trend changes, which decrease the value of past knowledge and, ultimately, degrade the performance of prediction algorithms.

For a deeper analysis of how time resolution affects predictability, Fig. 5.2.a) and b) depict a box-and-whisker diagram of the autocorrelation function of the DL traffic pattern in 310 cells from a live LTE network on an hourly and monthly basis, respectively. To ease comparison, two seasonal periods are represented in both cases (two days for hourly data and two years for monthly data). The shaded area depicts the 95% confidence interval for each lag, suggesting that values in the white area are very likely an actual correlation and not a statistical fluke. The smaller interval width for the hourly series is due to a larger number of samples stored in those series. In Fig. 5.2.a), the interquartile boxes show a cyclical pattern with maxima (strong positive autocorrelation) at lags 24 and 48 and minima (strong negative correlation) at lags 12 and 36. Such behavior reveals that hourly traffic is seasonal. Moreover, in most lags, the whole box is out of the shadowed area, suggesting that past information is relevant to the current traffic value. In contrast, Fig. 5.2.b) reveals that the autocorrelation of monthly traffic does not follow a seasonal pattern due to the absence of local maxima or minima. On the contrary, it quickly diminishes to 0, so that, in most cells, only information from lags 1 to 4 is significantly correlated with the current traffic value (i.e., boxes are outside the shaded area). This fact suggests that, even for time series with the same available time window (i.e., two periods), monthly busy-hour data is less predictable than hourly data. It is especially relevant that, unexpectedly, correlation with lags 12 and 24 (i.e., with data collected in the same month of previous years) is not significant, reducing data predictability. A closer analysis (not presented here) shows that seasonality is neither observed in the de-trended monthly traffic series.

The above analysis confirms that long-term traffic forecasting for network dimensioning, based on monthly busy-hour traffic data, requires a separate analysis from short-term traffic forecasting based on higher time resolution data.



(a) Hourly data.



(b) Monthly busy-hour data.

Figure 5.2: Autocorrelation of DL cell traffic.

## 5.3 Forecasting method

Overall, the considered forecasting methodology comprises the same stages as that proposed in section [4.3](#) for cell throughput estimation, namely data collection and pre-processing, model training and performance evaluation. Thus, for brevity, this section focuses on specific aspects of applying this generic methodology to forecasting, namely the considered prediction algorithms, the considerations to be taken into account when applying them for long-term cell traffic forecasting and the different model construction strategies.

### a) Prediction algorithms

Six forecasting algorithms are tested. A first group consists of two classical statistical TSA schemes, namely Seasonal Auto Regressive Integrated Moving Average (SARIMA) and Additive Holt-Winters (AHW). SARIMA computes the current value of a time series difference as the combination of previous difference values and the present and previous values of the series. As detailed in [219], a SARIMA process is described as  $SARIMA(p,d,q)(P,D,Q)_m$ .  $(p,d,q)$  describe the non-seasonal part of the model, where  $p$  is the auto-regressive order,  $d$  is the level of difference and  $q$  is the moving average order, with  $p$ ,  $d$  and  $q$  non-negative integers.  $(P,D,Q)_m$  describe the seasonal part of the model, where  $P$ ,  $D$  and  $Q$  are similar to  $p$ ,  $d$  and  $q$ , but with backshifts of the seasonal period  $m$  (e.g., for monthly data,  $m = 12$ ).

Holt-Winters calculates the future value of a time series with recursive equations by aggregating its typical level (average), trend (slope) and seasonality (cyclical pattern) [220]. These three components are expressed as three types of exponential smoothing filters with smoothing parameters  $\alpha$ ,  $\beta$  and  $\gamma$ , respectively. As in SARIMA, the seasonal period is denoted as  $m$ . Holt-Winters can be additive (when seasonality is roughly constant) or multiplicative (when seasonality is proportional to level). In this work, the former variant is chosen, since, as shown in Fig. 5.1, the seasonal effect is nearly constant through the time series.

A second group comprises state-of-the-art SL algorithms, namely RF, SVR and two different ANNs. The first ANN, denoted as ANN-LSTM, is a deep recurrent network with two hidden layers based on LSTM units capable of capturing long-term dependencies thanks to information control gates. The second one, denoted as ANN-MLP, is a shallow MLP that addresses forecasting as a time-independent regression problem.

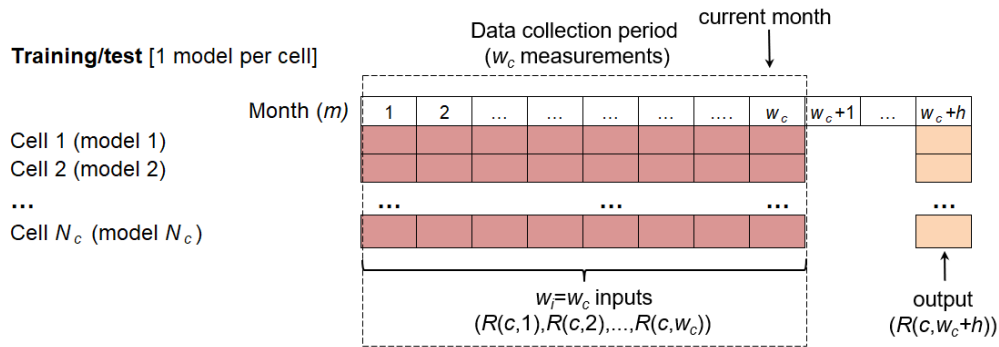
### b) Peculiarities of long-term traffic forecasting

Several issues must be considered when using the above algorithms for long-term cellular traffic forecasting in cellular networks, based on short and noisy time series:

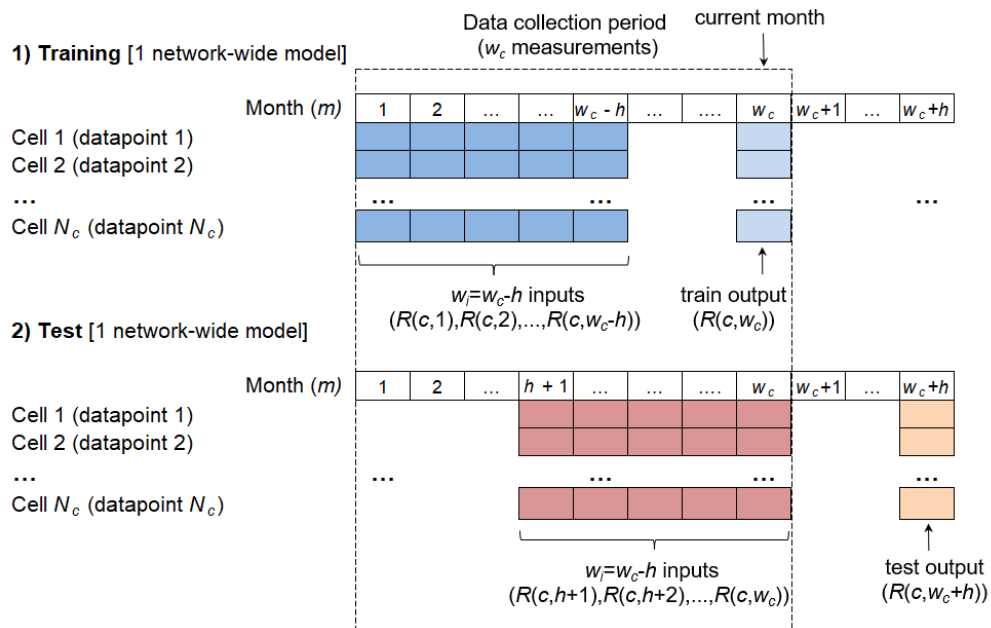
- 1) *Data collection window* ( $w_c$ ): the available period available in data warehouse systems of current cellular networks for long-term forecasting is typically less than 24 months. Moreover, time series from recently deployed cells may have even fewer monthly historical measurements. Thus, it is vital to check the capability of forecasting algorithms to work with small data collection windows. Such a

feature is especially critical during network deployment, when network structure constantly evolves (e.g., new cells are activated every month). At this early stage, robust traffic forecasting is crucial to avoid capacity problems and/or unnecessary investments by the MNO.

- 2) *Number of models*: recursive models such as SARIMA, AHW and ANN–LSTM are conceived to build a specific model per time series (i.e., cell) based on historical data of that particular cell. However, the short length of time series available for long-term traffic forecasting may jeopardize prediction capability with this approach, since it is always necessary to have more observations than model parameters to avoid model overfitting. As an alternative, with all SL algorithms (including ANN–LSTM), a single model can be derived for the whole network from historical data of all the cells. The latter ensures that enough training data is available to avoid model overfitting. Likewise, sharing past knowledge across cells in the system increases the robustness of predictions in cells with limited data or abnormal events.
- 3) *Forecasting horizon ( $h$ )*: The earlier a capacity bottleneck can be predicted, the more likely the problem will be fixed without any service degradation. Note that some network replanning actions (e.g., deploying a new site) may take several months. Such a delay forces operators to foresee traffic demand several months in advance (referred to as multi-step prediction). In classical TSA approaches, such as SARIMA and AHW, multi-step prediction is carried out by recursively using a one-step model multiple times (i.e., the prediction for the previous month is used as an input for predicting the following month). Such a recursive approach reduces the number of models needed, but quickly increases prediction errors originated by using predictions instead of observations as inputs [221]. This is a critical issue when using recursive algorithms for series with significant random components, such as those used in long-term forecasting. In contrast, SL algorithms can directly train a separate model for each future step. Such an approach does not entail an increase of computational load if the set of steps predicted is small (e.g., 3 and 6 months ahead) and/or parallelization is used.
- 4) *Interpretability*: Ideally, prediction models should be simple enough to have an intuitive explanation of their output values [222]. Models built with SARIMA and AHW are easier to understand, since their behavior is described by closed-form expressions, whereas SL models cannot be explained intuitively. Interpretability is not an issue in long-term traffic forecasting, thus being neglected here.



(a) Forecasting timeline for TSA algorithms (SARIMA and AHW).



(b) Forecasting timeline for SL algorithms (RF, ANN-MLP, ANN-LSTM and SVR).

Figure 5.3: Timeline of prediction algorithms in a generic case.

### c) Model construction

Fig. 5.3.a) and b) illustrate model training and test strategies considered for TSA (i.e., SARIMA/AHW) and SL (i.e., RF/SVR/ANN-MLP/ANN-LSTM) algorithms, respectively, conceived by taking into account all the above-mentioned peculiarities. Consider a network comprising  $N_c$  cells in which, in a given current month,  $w_c$  historical traffic measurements are available to predict cell traffic expected  $h$  months ahead,  $R(c, w_c + h)$ . For TSA approaches, as illustrated in Fig. 5.3.a), a different model is fitted per cell by using all historic measurements available as inputs (i.e., the input window to the model,  $w_i$ , is  $w_i = w_c$ ). That model is then applied recursively to predict traffic carried  $h' = 1, 2, \dots, h$  months ahead. This strategy results in  $N_c$  models (one

per cell). In contrast, for SL algorithms, a single model is created to predict traffic expected  $h$  months ahead in any cell of the whole network by using data from all cells, as shown in Fig. 5.3(b). First, the model is trained. The training dataset consists of  $N_c$  datapoints (one per cell) with  $w_c$  historical measurements. For each datapoint, the oldest  $w_i=w_c-h$  samples are used as predictors, whereas the most recent sample is used as output. Note that, unlike in TSA approaches, not all months can be used as predictors in the training stage to allow the  $h$ -month horizon. Once the model is trained, traffic expected  $h$  months ahead in each specific cell is computed by passing a new datapoint with the  $w_i$  most recent traffic measurements of the cell as predictors through the network-wide model.

It should be pointed out that, as stated above, ANNs based on LSTM units are conceived to build a model per time series (i.e., per cell). However, these models often have hundreds or even thousands of parameters (e.g., the ANN-LSTM model considered here has 1,331 parameters). To circumvent this problem, in this work, ANN-LSTM is used as the other SL algorithms, i.e., by creating a single model for all the cells in the network. To this end, a single time series is generated by concatenating time series from all cells in the network up to the considered current month, and only time lapses of  $w_c$  samples where the most recent sample corresponds to that current month are considered.

## 5.4 Performance assessment

This section presents the comparative analysis of forecasting schemes. For clarity, the dataset is first presented. Then, assessment methodology is detailed. Next, results are presented, broken down per experiment. Finally, computational complexity is discussed.

### 5.4.1 Dataset description

The proposed analysis requires a large dataset including data gathered from the largest number of cells during several years. Unfortunately, such data is not still available for 5G networks. As an alternative, a LTE dataset is considered that contains data collected from January 2015 to June 2017 (i.e., 30 months) in a large commercial network serving an entire country. The network comprises 7,160 eNBs covering a geographical area of approximately 500,000 km<sup>2</sup>, including cells of different sizes and



environments, with millions of subscribers. Analysis is restricted to the DL, as it has the largest utilization in LTE networks.

In the network, traffic measurements are gathered on a per-cell and hourly basis. Such raw data is preprocessed to obtain a single traffic measurement per cell/month to be stored in the long term for network dimensioning tasks. The resulting dataset consists of 7,160 time series (1 per cell) with 30 measurements (1 per month) of the monthly DL traffic volume in the busy hour, expressed as a rate (i.e., in kbps). The monthly DL traffic volume carried in a cell  $c$  and month  $m$  during the busy hour,  $R(c, m)$ , is calculated as follows:

1. The average DL traffic volume (in kbps) and the average number of active users are measured and collected per cell and hour.
2. The weekly busy hour is selected per cell as the hour with the highest number of active users in week  $k$ . Each week belongs to a month  $m$ . The DL traffic volume (in kbps) during that busy hour is selected as the weekly DL traffic volume per cell, week and month,  $R(c, k, m)$  (in kbps).
3. Finally, the monthly busy-hour DL traffic volume per cell and month,  $R(c, m)$ , is computed as the average of  $R(c, w, m)$  across weeks in month  $m$ , as

$$R(c, m) = \frac{1}{N_{week}(m)} \sum_{k=1}^{N_{week}(m)} R(c, k, m), \quad (5.2)$$

where  $N_{week}(m)$  is the number of weeks in month  $m$ . For simplicity, a week riding two months is considered to belong only to the month including more days.

The considered dataset combines a large temporal and spatial scale (30 months, entire country) with a fine-grained space resolution (cell), guaranteeing the reliability and significance of results. Moreover, it allows testing prediction algorithms with different data collection windows and time horizons, broadening the scope of the analysis.

## 5.4.2 Assessment methodology

Assessment is carried out with *IBM SPSS Modeler* [223], a commercial tool for predictive analytics extensively used in several fields [224] [225] [226]. The tool has a visual interface allowing users to use statistical and data mining algorithms without programming. Likewise, it offers an expert mode to find the optimal settings for certain

model hyperparameters. These features are extremely valuable for network planners without a deep knowledge of prediction algorithms. SPSS Modeler 18.1 includes all the forecasting algorithms considered in this work except for ANN–LSTM, which is implemented using *Keras* Python library. Regarding model hyperparameters, to get the best of classical TSA approaches,  $p$ ,  $d$ ,  $q$ ,  $P$ ,  $D$  and  $Q$  in SARIMA and  $\alpha$ ,  $\beta$  and  $\gamma$  in AHW must be set on a cell basis. For this purpose, the Expert mode offered by SPSS Modeler is used. In this mode, the input data series is first transformed when appropriate (e.g., differentiating, square root or natural log), and model parameters are then aromatically set to maximize accuracy in one-step ahead prediction. Such an Expert mode is also used to set the number of neurons in the hidden layer in ANN–MLP. The rest of hyperparameters are fixed or tuned following the random grid search in the parameter space identically as in chapter 4 considering the *MAE* defined in (4.5) as loss metric. The reader is referred to [227] and [186] for further information on the algorithms in SPSS Modeler and Keras, respectively.

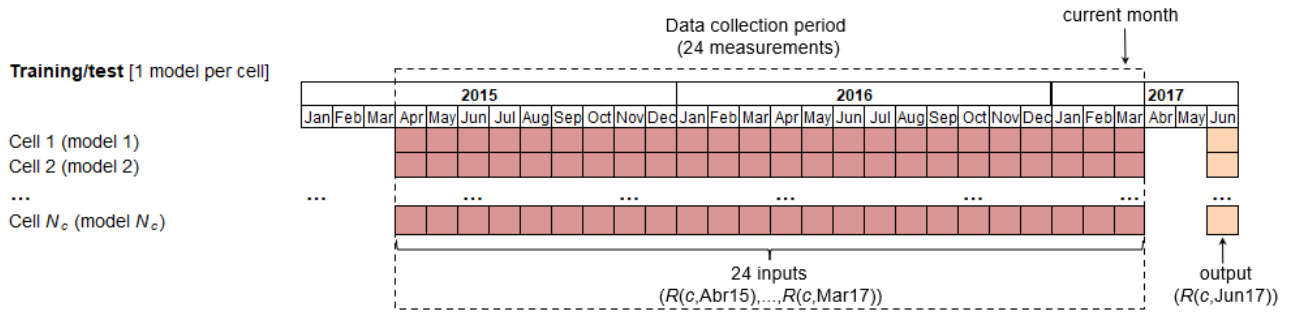
Four different cases are compared to check the sensitivity of forecasting algorithms to the data collection period ( $w_c$ ) and prediction time horizon ( $h$ ). Regarding  $w_c$ , algorithms are tested in two different cases. In the first case, it is assumed that the operator collects traffic data on a cell basis for 24 months. In this case, the six considered algorithms can be compared. In the second case, it is assumed that only data collected for last 12 months is available in the warehouse (e.g., network deployment stage). This reduction in the data collection window is an important constraint for SARIMA and AHW, which require more than 12 input samples to predict monthly data (13 for AHW and 16 for SARIMA) [218]. Hence, only SL algorithms are compared in the second case. To check the impact of prediction time horizon ( $h$ ) on model performance, two horizons are considered: 3 and 6 months (i.e., traffic is predicted 3/6 months in advance, respectively). The combination of  $w_c$  and  $h$  results in four cases, hereafter referred to as cases 12–3, 12–6, 24–3 and 24–6, where the first number denotes  $w_c$  and the second number denotes  $h$ . From the operator’s point of view, the less data stored and the more in advance traffic is forecast (i.e., case 12–6), the better, but, from an accuracy point of view, the opposite is likely true (i.e., case 24–3).

For clarity, Fig. 5.4 illustrates model training and test timelines for predicting traffic carried in June 2017 with  $h=3$  (i.e., prediction is made on March 2017) based on measurements from previous months. Specifically, Fig. 5.4.a) and b) present the timeline used for TSA and SL approaches, respectively, in case 24–3. In this case, since  $w_c=24$ , traffic in June 2017 is predicted in March 2017 based on historical data

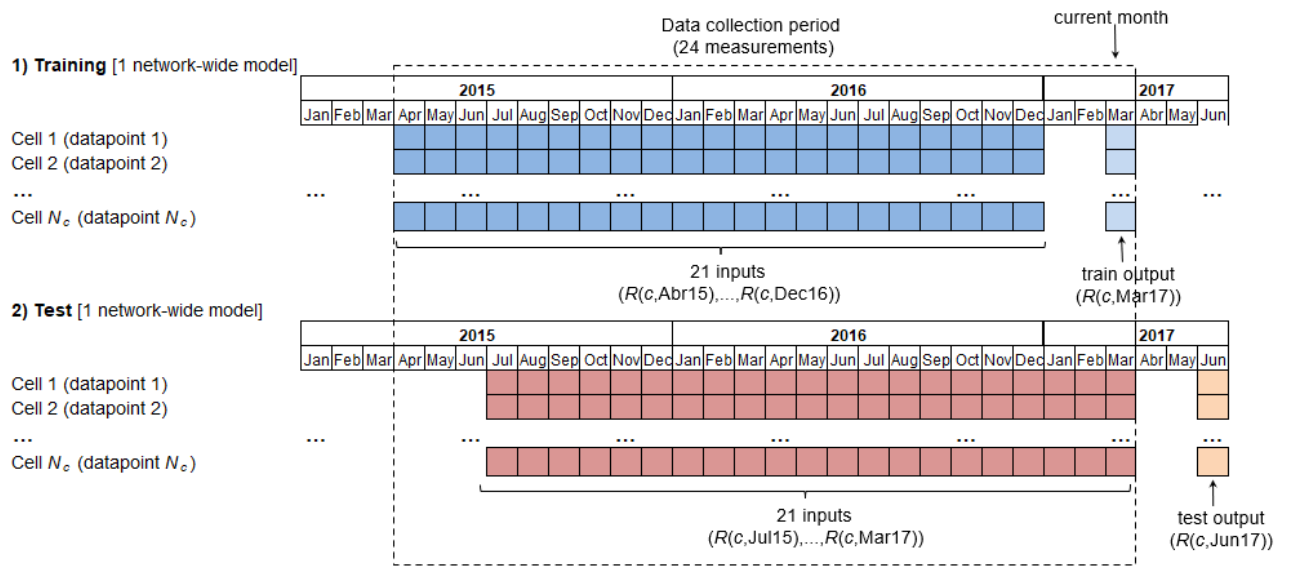
from April 2015 to March 2017. Similarly, Fig. 5.4c) corresponds to case 12–3, where only data from April 2016 to March 2017 is available. In the latter case, SARIMA and AHW algorithms cannot be used due to the limited length of the time series. Traffic forecasting for cases 24–6 and 12–6 follows a similar procedure with a 6-month gap between the end of the data collection window and the target month (e.g., in case 24–6, traffic in June 2017 is predicted in December 2016 based on measurements from January 2015 to December 2016).

To test the above-described cases, three experiments are performed sequentially. The results of each experiment motivate the execution of the following one.

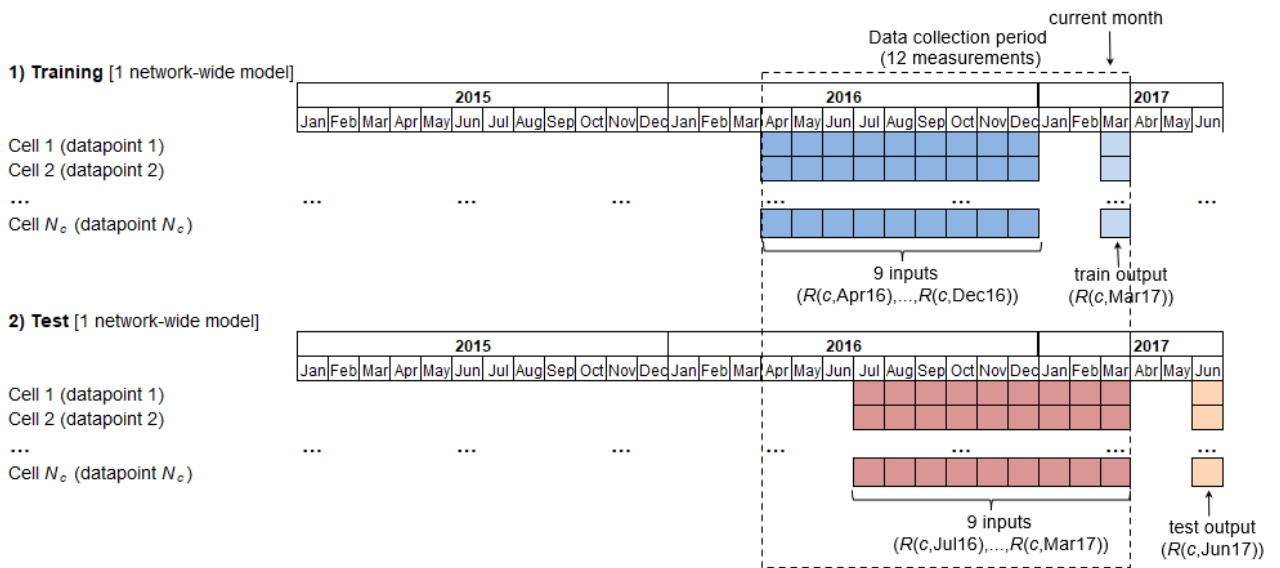
- *Experiment 1 – selection of data collection window.* The first experiment aims to determine how many months of data (i.e.,  $w_c=12$  or 24 months) are required to forecast cellular traffic accurately for the two considered prediction horizons (i.e.,  $h=3$  and 6 months). For this purpose, the six considered forecasting algorithms are evaluated in cases 12–3, 12–6, 24–3 and 24–6. The data to be predicted is cell traffic in June 2017, as shown in Fig. 5.4. Note that assessing (i.e., training and testing) case 24–6 (the most data-demanding case) requires collecting data 30 months in advance (i.e., dataset size), and thus June 2017 is the only possible target month for this experiment in the considered dataset.
- *Experiment 2 – algorithm comparison.* The second experiment aims to check: a) how much time in advance (3 or 6 months) prediction can be made with acceptable accuracy, b) the dependence of model performance on the target month, and c) which is the best prediction algorithm. For this purpose, cell traffic from July 2016 to June 2017 (i.e., for a year) is forecast considering cases 12–3 and 12–6. Cases 24–3 and 24–6 (and thus TSA approaches) are discarded according to results in experiment 1. For each SL algorithm, a different model is trained to predict traffic in each month as explained in Fig. 5.3, with the current month the closest to the target month for each horizon (e.g., in case 12–3, if the target month is May 2017, the data collection window starts in July 2016 and ends in February 2017).
- *Experiment 3 – creation of specific models for high-traffic cells.* In capacity planning, accurate traffic prediction is especially important in cells with high traffic, as these are more likely to suffer capacity problems. This experiment assesses the possibility of creating a differentiated model for such cells. The idea is to discard underutilized cells, often showing noisy traffic measurements,



(a) Case 24–3 for TSA approaches.



(b) Case 24–3 for SL approaches.



(c) Case 12–3 for SL approaches.

Figure 5.4: Timeline of prediction algorithms in cases 24–3 and 12–3 when forecasting cellular traffic in June 2017.

when training the model. To this end, a model to forecast traffic in June 2017 is trained only with data from high-traffic cells. In this work, a cell  $c'$  is considered as a high-traffic cell if its traffic in current month (e.g., in case 12–3, in March 2017) exceeds the 85<sup>th</sup> percentile of the monthly cell traffic in the network, i.e.,  $R(c', \text{Mar17}) > P_{85}^{R(c, \text{Mar17})}$ .

In all the experiments, prediction accuracy is measured with *MAPE* and *MAE*, defined in (4.3) and (4.5), respectively. Additionally, two secondary indicators are used for a more detailed assessment: a) the bias, computed as the mean error

$$\text{bias} = \frac{1}{N_c} \sum_c (\hat{R}(c, m) - R(c, m)), \quad (5.3)$$

and b) the execution time, as a measure of computational load.

### 5.4.3 Results

#### a) Experiment 1

Table 5.1 breaks down the performance of prediction algorithms in cases 12–3 and 24–3 (3-month horizon). Recall that SARIMA and AHW cannot be tested in case 12–3. Results show that SARIMA achieves the worst performance in case 24–3, with an extremely large *MAPE* (43.25%) and *MAE* (2069.72 kbps). AHW outperforms SARIMA, but still performs poorly (*MAPE*=29.28% and *MAE*=1780.71 kbps). Such a poor performance may be due to the recursive nature of these models, where noisy input data severely degrades the accuracy of predictions beyond the next step. All SL techniques outperform AHW, with *MAPE* below 28% and *MAE* below 1600 kbps. When comparing case 24–3 against case 12–3, it is observed that SL algorithms perform similarly or even better in case 12–3 (e.g., for ANN–MLP, *MAE* increases from 1023.55 kbps with 12 months to 1339.91 kbps with 24 months). This fact confirms that the influence of past measurements quickly diminishes in the long term due to changes in user trends and replanning actions by the operator.

Table 5.2 shows the comparison between cases 12–6 and 24–6 (6-month prediction horizon). In case 24–6, all SL techniques but SVR again outperform both SARIMA and AHW (SVR outperforms SARIMA, but not AHW). When comparing cases 12–6 and 24–6, all SL algorithms achieve a better *MAE* in case 12–6, as in cases 12–3 vs. 24–3 (e.g., in SVR, *MAE* decreases from 2517.43 kbps in case 24–6 to 1372.89 kbps in

Table 5.1: Impact of data collection window for 3-month forecasting horizon.

Case ( $w_c - h$ )	12-3		24-6	
FoM	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]
SARIMA	–	–	43.25	2069.72
AHW	–	–	29.28	1780.71
RF	23.75	1017.55	23.29	1236.44
SVR	25.78	1070.03	27.86	1572.81
ANN-MLP	24.28	1023.55	24.90	1339.91
ANN-LSTM	22.65	976.69	23.38	999.15

Table 5.2: Impact of data collection window for 6-month forecasting horizon.

Case ( $w_c - h$ )	12-3		24-6	
FoM	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]
SARIMA	–	–	590.17	16708.55
AHW	–	–	30.88	1902.44
RF	23.94	1048.63	22.76	1199.32
SVR	30.81	1372.89	37.66	2517.43
ANN-MLP	24.23	1055.93	23.58	1250.32
ANN-LSTM	22.23	1034.30	29.55	1253.69

case 12-6, a 45.47% decrease in relative terms).

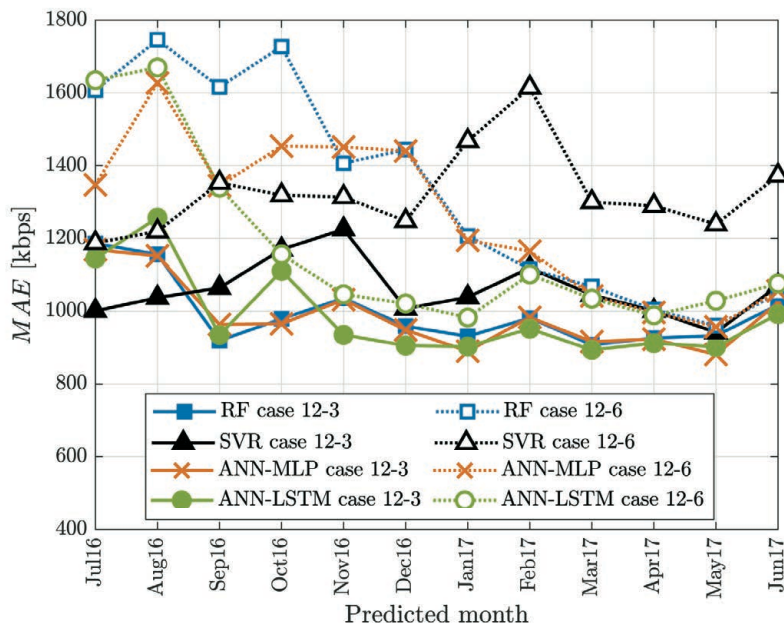
From the above results, it can be concluded that: a) SL approaches outperform SARIMA and AHW when predicting traffic in cellular networks in the long term, and b) there is not much benefit in storing traffic measurements for more than one year (unless SARIMA and AHW are the only options).

## b) Experiment 2

Table 5.3 presents the average *MAPE*, *MAE* and *bias* achieved for different target months for each algorithm and case. For a more detailed analysis, Fig. 5.5 breaks down *MAE* obtained in cases 12-3 (solid lines) and 12-6 (dotted lines) for each target month. Recall that cases 24-3 and 24-6 are not considered based on the conclusions in experiment 1. Table 5.3 shows that, in case 12-3, RF, ANN-MLP and ANN-LSTM perform similarly ( $\overline{MAPE} \approx 27\%$  and  $\overline{MAE} \approx 1000$  kbps), outperforming SVR ( $\overline{MAPE} = 30.26\%$  and  $\overline{MAE} = 1059.86$  kbps). Moreover, Fig. 5.5 reveals that prediction accuracy for most algorithms significantly degrades when predicting traffic in July and August 2016 (summer holidays) compared to the rest of months (working months). This might be due to isolated events taking place during summer months in the country where data

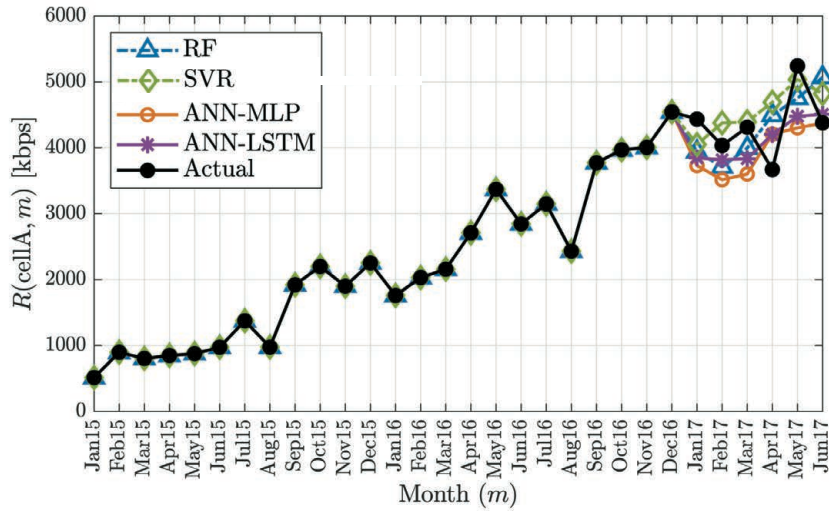
Table 5.3: Average performance of forecasting algorithms across different target months.

Case	12-3			12-6		
	$\overline{MAPE}$ [%]	$\overline{MAE}$ [kbps]	$\overline{bias}$ [kbps]	$\overline{MAPE}$ [%]	$\overline{MAE}$ [kbps]	$\overline{bias}$ [kbps]
RF	27.63	994.21	160.51	40.09	1329.31	719.20
SVR	30.26	1059.86	-251.32	36.71	1327.15	-531.02
ANN-MLP	27.73	987.28	134.61	38.71	1256.40	572.46
ANN-LSTM	26.37	986.88	134.27	31.87	1173.39	287.73

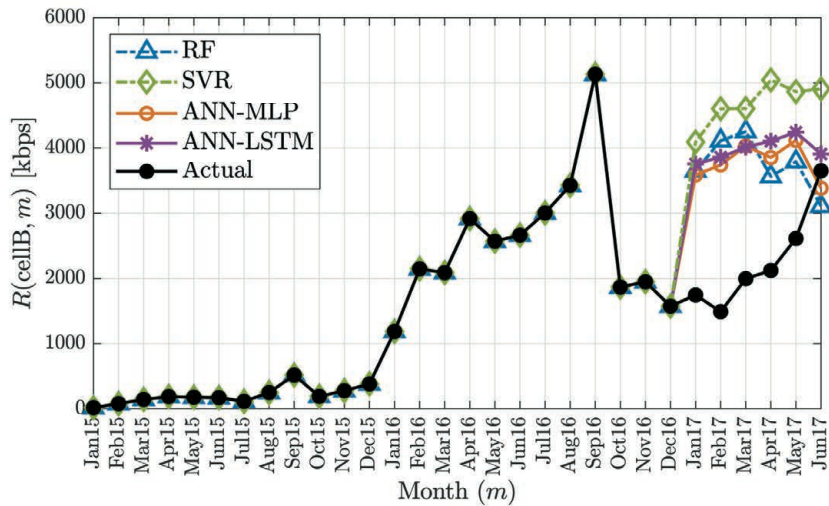
Figure 5.5:  $MAE$  evolution across different target months.

was collected (e.g., tourism, festivals, etc.) that change traffic patterns unpredictably, making data collected three months in advance not representative of the traffic in the months to come. In contrast, SVR shows a more stable behavior during summer months (i.e.,  $MAE$  does not degrade).

By comparing cases 12-3 and 12-6 in Table 5.3, it is observed that, for all algorithms, there is a significant degradation in accuracy if traffic predictions are made more than three months in advance (e.g., for ANN-LSTM,  $\overline{MAE}$  and  $\overline{MAPE}$  increase by 18.89% and 20.85% in relative terms, respectively). Moreover, dotted lines in Fig. 5.5 shows a substantial variation in  $MAE$  across months in case 12-6. Thus, it is recommended to use a 3-month prediction horizon when possible. ANN-LSTM shows the best overall results, with a  $\overline{MAPE}$  of 26.37% and a  $\overline{MAE}$  of 986.68 kbps in case 12-3, and the slightest degradation in accuracy from case 12-3 to case 12-6. Nonethe-



(a) Cell A (not affected by replanning action).



(b) Cell B (affected by replanning action).

Figure 5.6: Example of the impact of replanning actions on cell traffic.

less, for a 3-month horizon, ANN-MLP and RF algorithms can be used alternatively with similar accuracy ( $\overline{MAPE} \approx 27.70\%$  and  $\overline{MAE} \approx 990$  kbps).

It should be pointed out that, even for the best model (i.e., combination of algorithm and case), forecasts are not very accurate (i.e.,  $\overline{MAE} \approx 1000$  kbps, or, expressed more intuitively, a deviation of 0.39 GB per hour and cell). This fact confirms the unpredictability of busy-hour traffic metrics. A deeper analysis reveals that such a poor performance can be partially explained by replanning actions taken by the operator in the considered network during the data collection period, which lead to unpredictable traffic changes in neighbor cells. For a closer analysis, Fig. 5.6.a) and b) illustrate



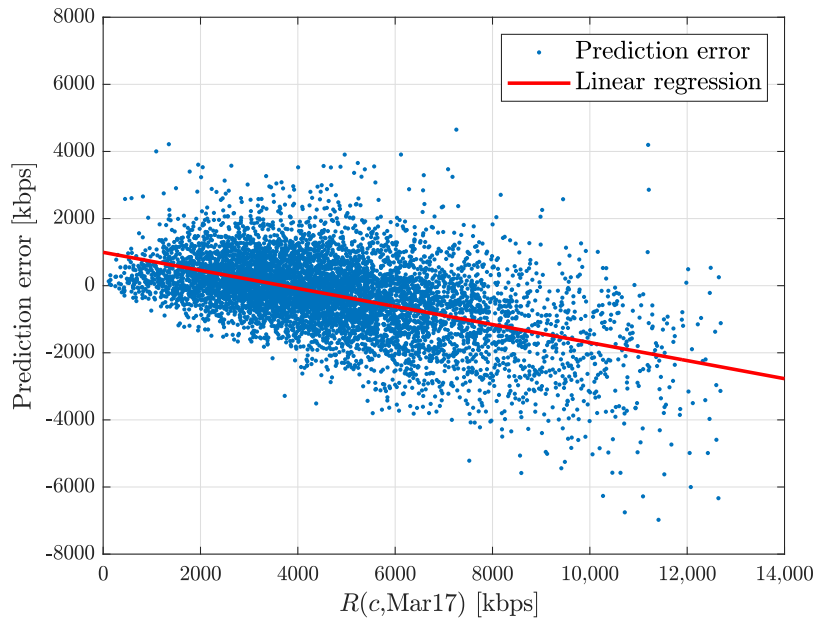


Figure 5.7: Prediction error vs. monthly busy-hour traffic (ANN-LSTM, case 12-3).

traffic prediction from January 2017 to June 2017 with a 3-month horizon for two cells, referred to as Cell A and Cell B, with the compared algorithms. No significant replanning actions were taken in the surroundings of Cell A during the data collection period, whereas Cell B is a cell with a new neighbor cell deployed in October 2016. In Fig. 5.6.a), it is observed that, for cell A, all algorithms predict real traffic quite well (e.g.,  $MAE$  fluctuates between 4 and 900 kbps for ANN-MLP, and between 314 and 824 kbps for RF). In contrast, in cell B, the abrupt decrease in traffic in October 2016, caused by the deployment of a nearby cell, leads to large prediction errors for all models.

It is also remarkable that  $\overline{bias}$  values in Table 5.3 for case 12-3 are negative or close to 0, i.e., models tend to underestimate traffic. This behavior is especially risky for high-traffic cells, which are more likely to suffer from capacity problems. For a closer analysis of bias, Fig. 5.7 depicts the scatter plot of prediction error,  $\widehat{R}(c, m) - R(c, m)$ , versus measured cell traffic obtained when predicting traffic carried in March 2017 with ANN-LSTM in case 12-3 (the combination of month/algorithm/horizon with the lowest  $MAE$  in this experiment). The regression line shows that the more loaded cells are, the more negative the error is. This trend points out the need for a more accurate model for high-traffic cells. This problem will be addressed in experiment 3.

Table 5.4: Performance of network-wide and specific forecasting models for high-traffic cells (case 12–3).

Model	Network-wide		Specific	
	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]
RF	12.46	1339.88	11.26	1212.18
SVR	20.44	2223.88	14.49	1725.20
ANN–MLP	12.31	1374.55	11.35	1232.98
ANN–LSTM	12.21	1311.72	13.27	1356.13

### c) Experiment 3

Table 5.4 breaks down *MAPE* and *MAE* in case 12–3 for the 1,074 (15%) cells with the largest traffic in March 2017, obtained with two different models: a) the model built in experiment 1 (denoted as network-wide model), and b) a specific model trained exclusively with data collected from these high-traffic cells (denoted as specific model). To isolate the effect of building a differentiated model, those high-traffic cells affected by replanning actions between March 2017 and June 2017 have been considered for model training, but not for model exploitation. The table shows that the specific model outperforms the network-wide model for RF, SVR and ANN–MLP algorithms. In contrast, the specific model created with ANN–LSTM does not improve the network-wide model built with this approach. The latter can be due to the lower number of training datapoints in specific models, which can lead to overfitting for models with a large number of internal parameters such as ANN–LSTM. SVR experiences the largest improvement with the specific model in absolute terms, decreasing *MAE* from 2223.88 kbps to 1725.20 kbps (22.42% in relative terms) and *MAPE* from 20.44% to 14.19% (29.11% in relative terms). Nonetheless, it is still the worst algorithm. Specific models created with RF and ANN–MLP show the best results, achieving  $MAPE \approx 11\%$  and  $MAE \approx 1200$  kbps. Nonetheless, CDFs show that error is still negative in many cells ( $\approx 45\%$  for RF, ANN–MLP and ANN–LSTM and  $\approx 75\%$  for SVR). This issue must be addressed by other means (e.g., models based on predictors other than cell traffic).

For a closer analysis, Fig. 5.8.a)–d) represent the error CDFs obtained with SL algorithms for high-traffic cells with the network-wide model (solid lines) and the specific model (dashed lines) in case 12–3 for the selected target month (June 2017). It is observed that, for all algorithms, error curves with the specific models are shifted to the right compared to those with the network-wide models, whose median values are

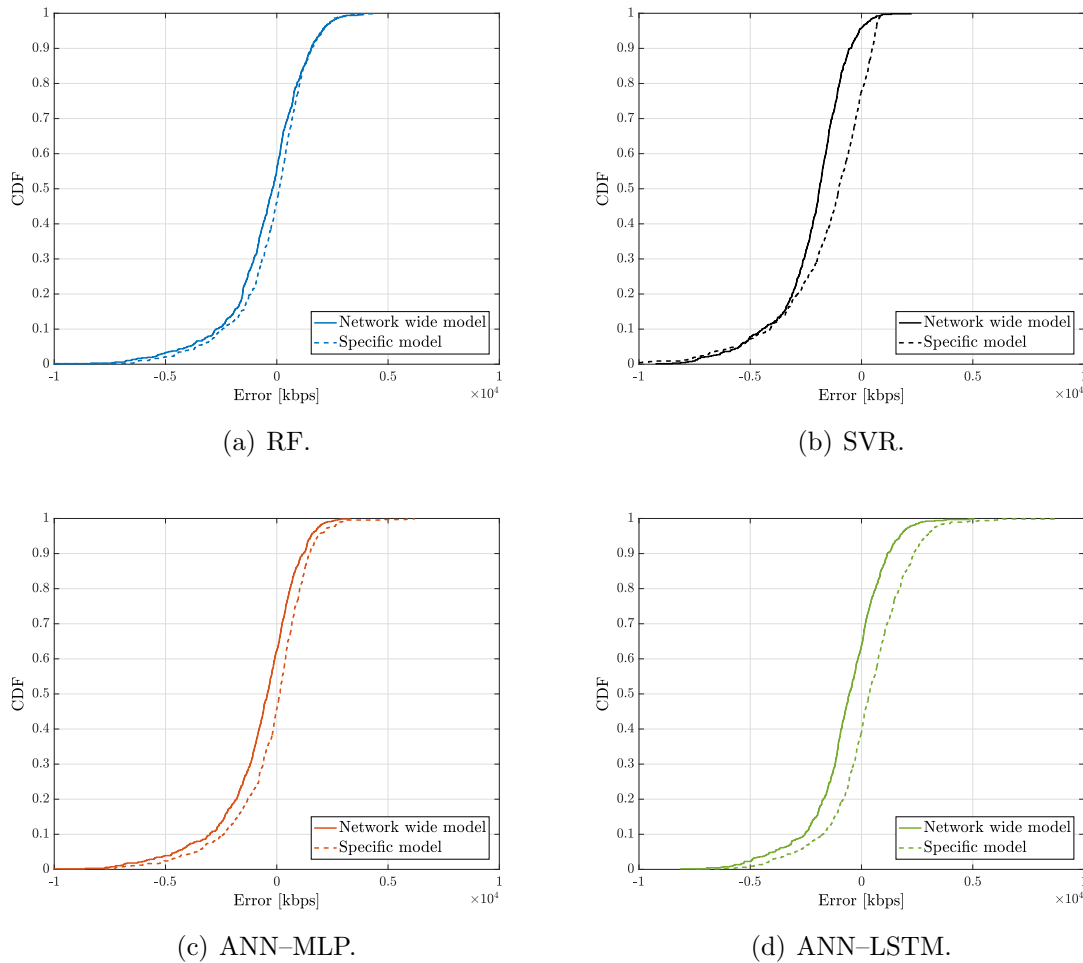


Figure 5.8: Error cumulative distribution functions for SL algorithms when forecasting traffic in high-traffic cells (case 12-3).

closer to 0. Thus, the specific models for high-traffic cells increase prediction accuracy while reducing bias. Nonetheless, CDFs show that error is still negative in many cells ( $\approx 45\%$  for RF, ANN-MLP and ANN-LSTM and  $\approx 75\%$  for SVR). This issue must be addressed by other means (e.g., models based on predictors other than cell traffic).

#### 5.4.4 Computational complexity

Cell traffic forecasting in radio planning tools entails: a) collecting and preprocessing data in the OSS, b) training the model (or set of models), c) exploiting the model and d) retraining the model when necessary. PMs used to compute aggregated cell traffic per hour are often collected by MNOs for network management purposes. Likewise, calculating busy-hour monthly traffic per cell from hourly traffic measurements is

Table 5.5: Execution times for forecasting models [s].

Case	12-3	12-6	24-3	24-6
Number of predictors				
TSA algorithms	–	–	24	24
SL algorithms	9	6	21	18
Execution times [s]				
SARIMA (per cell)	–	–	0.56	0.61
AHW (per cell)	–	–	0.65	0.72
RF (entire network)	5	4	8	6
SVR (entire network)	12	10	14	13
ANN-MLP (entire network)	3	2	6	4
ANN-LSTM (entire network)	140	98	313	245

simple; thus, dataset creation should not entail a significant computational workload. The most time-consuming task is model fitting. For SL algorithms, time complexity was discussed in section [4.4.2](#). For SARIMA and AHW, it is expected that running time grows linearly with the number of predictors (months),  $w_i$ , and the number of models built (cells),  $N_c$ . Thus, their worst-case time complexity is  $\mathcal{O}(N_c \times w)$ . Once trained, model exploitation is immediate (in this analysis, less than 0.1 ms per time series). The whole process must be repeated every month.

As an example of model computational complexity, Table [5.5](#) summarizes the execution time for the tested algorithms in experiment 1 (7,160 cells) in a centralized server with Intel Xenon octa-core processor, clock frequency of 2.4 GHz and 64 GB of RAM. For SARIMA and AHW, execution time comprises computing a cell-specific model and extending it until the target month, which must be repeated for all cells in the system. For SL algorithms, execution time comprises training a network-wide model with the series from all cells and computing predictions from historical traffic values (i.e., no model extension is needed). The number of predictors per case is shown in the upper part of the table. Results show AHW and SARIMA are the most time-consuming approaches in cases 24-3 and 24-6, since they require building a model per cell. Among SL algorithms, ANN-MLP and ANN-LSTM are the fastest and most time-consuming approaches, respectively. In SL algorithms, the longer data collection period, the larger number of predictors, and hence the larger runtime. For a certain collection period, the larger the time horizon, the lower number of predictors in the model, and thus the lower runtime. In contrast, in SARIMA and AHW, the longer

the time horizon, the larger runtime, since the model must be extended until the target month. Nonetheless, execution times in all cases are negligible for the considered application, where models must be trained and exploited once a month.

## 5.5 Conclusions

Accurate long-term traffic forecasting is crucial for replanning the RAN in cellular networks. However, monthly busy-hour time series often used for this purpose are short and noisy, making long-term prediction a challenging task. In this chapter, a comparative study has been conducted to assess different approaches for predicting cellular traffic in the long term (i.e., several months in advance). Six algorithms have been compared, including classical time series analysis schemes (SARIMA and AHW) and supervised learning algorithms (RF, ANN-MLP, ANN-LSTM and SVR). To this end, three experiments have been carried out with a dataset taken from a live LTE network covering an entire country (7,160 cells) and traffic data for two and a half years.

Results have shown that SL algorithms outperform classical TSA in terms of accuracy and required storage capacity. Specifically, with SL algorithms, traffic carried per cell can be predicted with a  $MAE \approx 1000$  kbps with a 3-month time horizon and a 12-month data collection period. It has also been shown that it is advisable to develop specific models for high-traffic cells, where prediction accuracy is critical. Overall, RF and ANN-MLP have shown the best results, providing acceptable accuracy ( $MAPE \approx 11\%$ ) to detect capacity bottlenecks in high-traffic cells with a 3-month prediction horizon by using data from the past 12 months. It is remarkable that these non-deep algorithms perform very similar to deep neural networks based on LSTM units, used to model time dependencies in short- and medium-term traffic forecasting. This is due to the monthly busy-hour aggregation of data, which reduces time series length and predictability compared to hourly or daily traffic series. Nonetheless, none of the considered algorithms is highly accurate, especially for summer months, due to changes in user trends, social events or temporary replanning actions by the operator.

Two work lines have been explored to improve model performance obtained in this analysis in the framework of a contract with a telecom vendor: a) preprocessing time series before fitting and exploiting forecasting models and b) refining/changing the training approach from that presented in Fig. 5.3. Although results cannot be detailed here for confidentiality reasons, some conclusions are outlined next. Regarding data

preprocessing, applying smoothing over time series leads to a better trend estimation at the expense of losing (or attenuating) the contribution of any other component (e.g., seasonality) to the series, which is paid off in most cells due to negligible seasonal component in monthly busy-hour traffic series. Alternatively, additive time series decomposition can be used to separate trend, seasonal and residual components and then train a different model to forecast each component. When recomposing the final forecast for a given cell, some components can be omitted if considered irrelevant. It should be pointed out that these advanced data preprocessing techniques pose a significant increase in computational complexity (especially the latter). No matter the selected preprocessing approach (i.e., smoothing, decomposing or leaving time series as they are), efficient outlier management improves forecasts significantly. Since cells in a network present significantly different traffic patterns and levels, outlier detection must not rely on network-wide statistical information, but be performed on a per-series basis.

Regarding model construction, a tested option is training a specific model for each month of the year by using datapoints with the same input and output months in past years. The possible benefit would be capturing relations between traffic carried in the network in specific months. As a side effect, training these models entails storing more historic traffic measurements per cell (at least  $w_c+12-h$  samples per time series). Another explored option is increasing the number of datapoints without changing the collection window ( $w_c$ ) by reducing the input window, i.e.,  $w_i < w_c + h$ . This approach helps to avoid model overfitting and may provide SL models with more traffic patterns at the expense of increasing computational complexity considerably. Preliminary tests reveal that any of these options improve model performance significantly, which is consistent with the analysis of autocorrelation presented in section [5.2](#) (i.e., lack of peaks in lags 12/24, fast decrease in autocorrelation values...).

Nonetheless, it is strongly recommended that operators store data with finer time resolution (e.g., daily busy-hour measurements) in the long term to make the most of SL models for traffic forecasting.

# Chapter 6

## Traffic steering in cellular networks

Once networks are deployed, an effective MLB strategy is key to relieving localized congestion problems, avoiding unnecessary re-planning actions while guaranteeing customer satisfaction. This chapter deals with the problem of performing MLB in multi-service cellular networks with different scenarios (i.e., multi-tier and sliced RANs). Content is organized as follows. Section [6.1](#) reviews state-of-the-art contributions. Then, section [6.2](#) presents a data-driven MLB algorithm for multi-tier networks with QoE criteria. Likewise, section [6.3](#) proposes novel solutions for MLB in sliced RANs.

### 6.1 Related work

In the literature, several algorithms have been proposed for MLB through HOM tuning in cellular RANs. Most contributions tackle HOM optimization as a control problem, for which different types of controllers have been proposed. The earliest works relied on proportional controllers driven by heuristic rules to perform intra-frequency MLB in macrocellular scenarios. In [\[30\]](#), an incremental controller tunes HOMs in fixed steps when the load difference of adjacent cells exceeds a threshold. Cell load is measured considering PRB utilization ratio and QoS requirements. In [\[228\]](#), the controller estimates the impact of changing HOMs on network performance with an analytical model, and tunes HOMs to maintain all cells under a preset load threshold. A similar approach is proposed for small-cell scenarios in [\[32\]](#), where an adaptive load threshold is considered, so that MLB can also act in non-congested cells with unevenly loaded cell boundaries. In [\[86\]](#), HOMs in a femtocell scenario are tuned with a Fuzzy Logic Controller (FLC) fed by current HOM values and blocking statistics to equalize the

blocking ratio among cells.

With the latest advances in information technology, cutting-edge load balancing solutions rely on artificial intelligence. [229] surveys ML-based load balancing schemes. Although some works are based on SL (e.g., MLR [230]), most MLB schemes rely on RL. Initially, RL was used to enhance the ability of classical controllers to adapt to changing environments. For instance, [231] improves the solution proposed in [86] for femtocell scenarios by using a Q-learning agent that customizes IF-THEN rules of the FLC driven by information from trial-and-error interactions with the network. In [232], this fuzzy Q-learning approach is tested in macrocellular scenarios, revealing the potential of readjusting FLC rules with constant exploration and exploitation to capture changes in network conditions. As an alternative, RL can be used to drive the control process. For instance, in [233], a Q-learning agent takes decisions per adjacency to equalize cell load from information about PRB utilization and cell-edge users.

In multi-tier networks, traffic steering becomes more complex due to the asymmetric signal and interference levels between cells of different layers (e.g., cells of different carriers, macro cells vs. small cells...) [234]. The earliest proposals dealt with user mobility in multi-band (or multi-carrier) cellular networks, consisting of co-located cells using different frequency bands [235]. Later schemes deal with user mobility in heterogeneous cellular networks, comprising overlapping cells of different sizes or technologies. In multi-tier networks, a common approach is to address inter-frequency load balancing in the cell (re)selection process. In [28], a heuristic algorithm that assigns cell-specific offsets to low-power nodes in a heterogeneous LTE network is proposed, so that more users can be associated with them during cell reselection (a.k.a. cell range expansion). In [236], an association scheme that jointly maximizes DL system capacity and minimizes mobile station UL transmit power is presented. In [29], the parameters in different cell (re)selection strategies are optimized with statistical information of radio propagation to achieve a target traffic distribution in a multi-carrier LTE network. Alternatively, other authors tackle traffic steering by adjusting the value of inter-frequency HO parameters. For instance, in [237], the optimal configuration of inter-RAT HOMs in a multi-RAT multi-layer wireless network is derived through a sensitivity analysis. In [31], a self-tuning algorithm based on a FLC adapted with RL is proposed to adjust inter-RAT HOMs to reduce call dropping ratio in heterogeneous LTE networks. In [238], cell-specific offsets are adjusted by taking into account target cells and their surroundings, reducing the number of unsatisfied users and HOs.

All the above traffic steering algorithms are driven by simple performance indica-



tors obtained from the aggregation of all connections in a cell. In legacy networks, where voice calls were the predominant service, these approaches achieved the best user performance. However, field trial results in [87] point out that balancing cell load in LTE networks and beyond supporting services of very different requirements does not guarantee the best overall user QoE. As an alternative, more recent works tackle MLB from a QoE perspective. In [33], the QoE of neighbor cells in a single-layer macrocellular scenario is balanced by tuning service-specific HOMs with a FLC. Other QoE-based works rely on optimization instead of using heuristic control rules. For instance, in [34], the impact of tuning HOM on system QoE is estimated with an analytical model adjusted with network data, and optimality is ensured with a gradient ascent scheme. In [239], an algorithm based on dynamic particle swarm optimization is centrally applied, which optimizes the overall QoE and reduces the number of users with poor QoE. QoE aspects have also been considered in RL-based MLB solutions as part of state [90] and/or reward [240] [241]. Closer to this work, in [242], a throughput-based traffic steering algorithm for heterogeneous LTE-Advanced networks is presented. Traffic steering decisions are evaluated by predicting whether forcing the HO of users may increase the overall system throughput. For this purpose, the maximum radio link throughput that each user could potentially achieve on each neighbor cell is estimated by the Shannon formula with a round-robin packet scheduler. The algorithm improves the overall user throughput, revealing the potential of evaluating HO impact on user performance for MLB. However, in current networks offering multiple services, end-user throughput strongly depends on the traffic mix and the packet scheduling algorithm. Thus, changing the scheduler or the demanded services might lead to inaccurate throughput estimations. Moreover, an increase in user throughput does not necessarily lead to a significant QoE improvement due to the non-linear mapping between QoS and QoE (e.g., logarithmic [243] or exponential [244]).

This chapter proposes a data-driven MLB algorithm for multi-carrier LTE networks to address these shortcomings. The algorithm aims to improve the overall system QoE. For this purpose, traffic steering is carried out by tuning RSRQ-driven HOMs in a classical inter-frequency HO scheme. The tuning process is driven by a novel indicator derived from individual connection traces that estimates the impact of HOs on end-user QoE. The algorithm is validated in a dynamic system-level simulator implementing a real multi-carrier scenario. The main contributions of this proposal are: a) a novel indicator derived from individual connection traces, showing the impact of HOs on end-user QoE, and b) a new self-tuning algorithm for steering users among carriers to

improve the overall system QoE.

Additionally to the need for service-oriented traffic steering, in NGNs, NS makes network management more complex, requiring self-optimization solutions ensuring SLA compliance in slices with different radio capacity and performance targets [245]. Few slice-aware self-optimization solutions have been proposed in the literature due to the absence of live networks with NS. Close to this work, a slice-aware mobility robustness optimization algorithm is presented in [246], which tunes slice-specific HOMs every 15 minutes with an actor-critic agent based on twin delayed deep deterministic policy gradient. However, to the author knowledge, the task of performing slice-aware MLB has not been covered yet.

To address the above gap, this chapter also proposes a slice-aware MLB algorithm aiming to improve SLA compliance in NS scenarios. For this purpose, traffic steering is carried out by tuning RSRP-driven intra-frequency HOMs in a slice-aware HO scheme. The tuning process is driven by a novel indicator derived from connection traces reflecting the imbalance of SLA compliance per slice between neighbor cells. To deal with the high dynamism of NS scenarios, the algorithm works on a time resolution finer than the legacy 15-min ROP. The main contributions of this second part are:

- a) Proposing the first slice-aware MLB scheme, driven by a novel SLA-based indicator derived from connection traces.
- b) Comparing its performance with other slice-aware and slice-unaware MLB solutions in a sliced RAN offering eMBB and uRLLC services.
- c) Analyzing the impact of key settings in the self-tuning process, such as time resolution or parallelization, on system performance.

Unlike most related contributions, the two algorithms proposed here are validated in a dynamic system-level simulator emulating the activity of realistic network scenarios taken from live networks, increasing the significance of results.

## 6.2 QoE-driven traffic steering in multi-tier LTE networks

This section deals with the problem of performing inter-frequency traffic steering with QoE criteria in LTE networks. For this purpose, section [6.2.1] formulates the problem

of optimizing the QoE in a multi-carrier LTE network by improving inter-frequency HO performance. Then, section 6.2.2 describes the proposed self-tuning algorithm, which is assessed via simulation in section 6.2.3. Finally, section 6.2.4 summarizes the main conclusions.

### 6.2.1 Problem formulation

As explained in section 2.2, UE mobility in LTE is handled by an event-based hard HO procedure often driven by RSRP and/or RSRQ measurements [247]. RSRP is defined as the linear average of the received power in the resource elements carrying cell-specific reference signals within the measurement frequency bandwidth. RSRQ is defined as the ratio

$$RSRQ = \frac{N_{PRB} \cdot RSRP}{RSSI}, \quad (6.1)$$

where  $N_{PRB}$  is the number of PRBs over the carrier bandwidth and  $RSSI$  is the E-UTRA carrier Received Signal Strength Indicator, providing information about total received wideband power, including all interference and thermal noise. Hence, RSRP is equivalent to signal strength, while RSRQ provides information about received signal quality and cell load. The type of measurement (i.e., RSRP or RSRQ) used to evaluate equations for event triggering is up to the operator.

In multi-carrier networks, it is essential to set a suitable HO scheme (i.e., triggering events, measurement type and HO parameters) to ensure efficient bandwidth use and guarantee end-user satisfaction. It is common practice for operators to configure both intra-frequency and inter-frequency HOs based on RSRP measurements [248]. Fig. 6.1 shows the typical HO scheme (hereafter referred to as Signal-Based HO scheme, SBHO) for handling mobility in a two-tier network. The bottom layer, comprising large cells (cells 1 and 2) working at a low carrier frequency, represents the coverage layer ( $L_{cov}$ ). The top layer, consisting of small cells (cells 3 and 4) with large bandwidth and a higher carrier frequency, works as a capacity layer ( $L_{cap}$ ). In both layers, intra-frequency HOs are triggered by event A3 introduced in (2.4) driven by RSRP (a.k.a. power-budget HOs), i.e.,

$$RSRP_u(j) \geq RSRP_u(i) + HOM_{intra}(i, j), \quad (6.2)$$

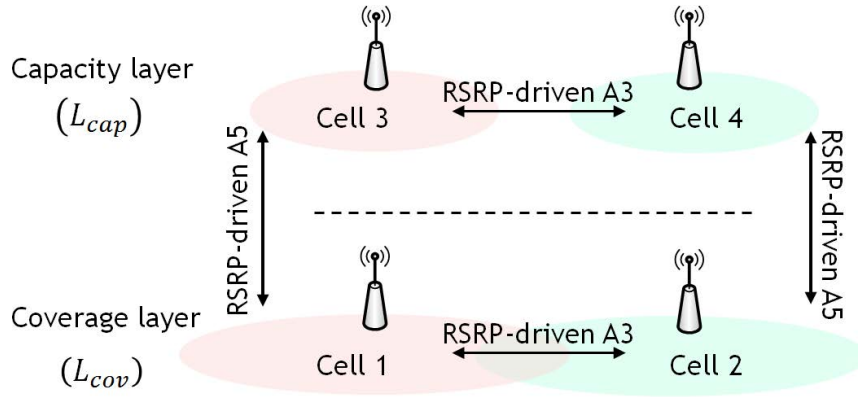


Figure 6.1: Typical handover scheme in a two-tier network.

where  $RSRP_u(i)$  and  $RSRP_u(j)$  are pilot signal levels received by user  $u$  from the serving cell  $i$  and neighbor cell  $j$ , respectively, and  $HOM_{intra}(i, j)$  is the intra-frequency HOM, defined on a per-adjacency basis. In contrast, inter-frequency HOs are triggered by RSRP-driven event A5, i.e.,

$$RSRP_u(i) \leq thd1(i), \quad (6.3)$$

$$RSRP_u(j) \geq thd2(j), \quad (6.4)$$

where  $thd1(i)$  and  $thd2(j)$  are absolute signal level thresholds, and  $i$  and  $j$  are inter-frequency neighbor cells (e.g., cells 1 and 3 in Fig. 6.1).

Since SBHO scheme is based on RSRP, it ensures that UEs are always connected to a cell with adequate received power. However, RSRP measurements do not reflect other factors affecting the radio link performance, such as noise, interference or cell congestion. As a consequence, network performance can be severely degraded if the coverage layer becomes congested due to its better propagation conditions. This problem can be solved by using RSRQ to trigger inter-frequency HOs.

To show the link between cell load and RSRQ, Fig. 6.2 represents an example of the evolution of RSRQ received by a LTE user from the serving cell as cell load (measured by the PRB utilization ratio,  $PRB_{util}$ ) changes in a simulation tool. Two RSRQ values are shown: instantaneous RSRQ,  $RSRQ_{inst}$ , and RSRQ averaged over a certain time window,  $RSRQ_{avg}$ , the latter reported in measurement reports [84]. It is observed that the value of  $RSRQ_{inst}$  strongly depends on cell load, i.e., the highest  $PRB_{util}$ , the lowest  $RSRQ_{inst}$ . Likewise, the left side of the figure shows that, even

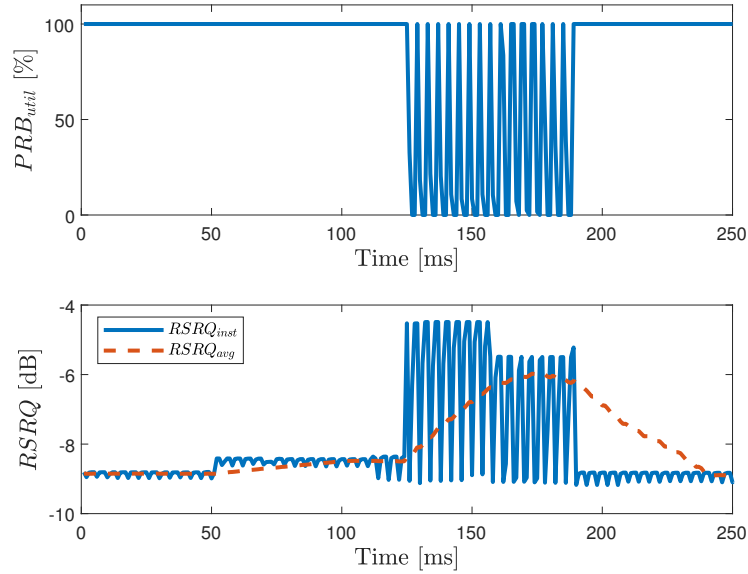


Figure 6.2: Impact of cell load on RSRQ.

when cell load is constant,  $RSRQ_{inst}$  varies due to desired signal strength and interference fluctuations. This rapid variation can turn into instabilities when evaluating HO triggering conditions. The averaging operation in  $RSRQ_{avg}$  smoothes out such fluctuations. Thus,  $RSRQ_{avg}$  may be an adequate measurement for triggering inter-frequency HOs in multi-carrier LTE networks, as traffic will be offloaded from coverage layers when capacity layers become underutilized. Some studies [249] [250] state that RSRQ-driven inter-frequency HOs lead to better performance in terms of packet delay, data throughput, number of HOs and UE power consumption. For this reason, RSRQ-driven inter-frequency HOs has been used as a passive traffic steering solution in multi-carrier scenarios [242]. Nonetheless, proper HOM settings must still be configured to ensure the best user experience. This is done by the proposed QoE-based MLB algorithm.

### 6.2.2 Traffic steering strategy

In this section, a novel strategy for traffic steering in multi-carrier LTE networks is presented. The aim is to improve the overall system QoE by redistributing users between carriers. For this purpose, a two-stage optimization process is carried out. First, a mobility scheme combining RSRQ and RSRP measurements to trigger inter-frequency HOs is activated. Then, inter-frequency HOMs are tuned per adjacency with a new MLB algorithm.

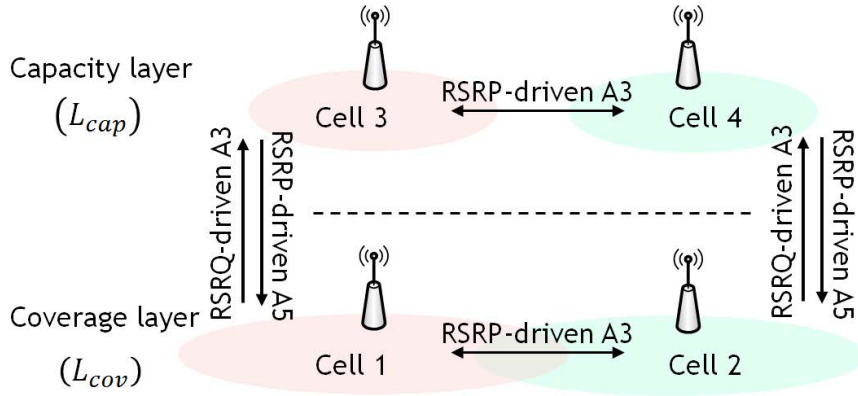


Figure 6.3: Quality-based handover scheme for a two-tier network.

### Stage 1: activation of RSRQ-driven inter-frequency HOs

First, the HO scheme illustrated in Fig. 6.3 is enabled, hereafter referred to as Quality-Based HO scheme (QBHO). Unlike SBHO scheme presented in Fig. 6.1, QBHO relies on RSRQ-driven HOs triggered by event A3 to handle mobility from coverage to capacity layer. Thus, a HO in this direction is triggered for user  $u$  when

$$RSRQ_u(j) \geq RSRQ_u(i) + HOM_{inter}(i, j), \quad (6.5)$$

where  $RSRQ_u(i)$  and  $RSRQ_u(j)$  are the RSRQ values received by user  $u$  from the serving and neighbor cells, respectively, and  $HOM_{inter}(i, j)$  is the inter-frequency HOM.

A default value of  $HOM_{inter}(i, j)=3$  dB is set for all adjacencies  $(i, j) | \{i \in L_{cov}, j \in L_{cap}\}$  [251]. Recall that RSRQ value depends not only on the received signal strength but also on DL interference and cell load. Consequently, even if signal strength received from  $L_{cap}$  is lower than that received from  $L_{cov}$ , some users will be offloaded from  $L_{cov}$  to  $L_{cap}$  when  $L_{cap}$  is underutilized. Moreover, any user experimenting bad coverage at  $L_{cap}$  will be reallocated to  $L_{cov}$  thanks to the coverage-based HO mechanism set for HOs from  $L_{cap}$  to  $L_{cov}$ . Hence, QBHO scheme guarantees service continuity while retaining a good signal quality.

### Stage 2: QoE-driven optimization of inter-frequency HOMs

Once QBHO scheme is activated, a novel MLB algorithm referred to as Optimized Experience (OE) is executed, which is the main contribution. The algorithm aims to improve the overall system QoE by finding the best share of users among cells of diffe-

rent carriers. This goal is achieved by tuning RSRQ-driven inter-frequency HOMs on an adjacency basis. Unlike classical traffic steering algorithms, where parameter tuning is driven by cell-level performance counters (e.g.,  $PRB_{util}$ ), the proposed algorithm relies on a new indicator that estimates the impact of HOs on end-user QoE. For clarity, the rationale of the algorithm is first explained, the indicator used to drive the tuning process is then defined and the controller is described later.

**Rationale of the algorithm** In LTE, an inter-frequency HO changes: a) the radio link conditions of the handed-over user and b) the number of simultaneous users in the source and target cells. These changes have an impact on received signal level of the handed-over user and cell loads, which may ultimately affect the QoE of every user in both cells.

From a QoE perspective, the optimum HO point is that maximizing the overall QoE of users in both source and target cells. Steering a user to the new cell too early might negatively affect the QoE of the handed-over user (e.g., due to low signal in the target cell) and that of users in the target cell (e.g., due to an earlier increase of cell load). In this early case, the overall QoE in the adjacency is expected to be degraded after the HO (i.e., the overall QoE is worse when the handed-over user is in the new cell). Conversely, steering a user to the new cell too late might negatively affect the QoE of the handed-over user (e.g., due to low signal in the source cell) and users in the source cell (e.g., due to a later decrease of cell load). In this case, the overall QoE in the adjacency is expected to improve after the HO (i.e., the overall QoE is worse when the handed-over user is in the old cell). In the optimal situation, the overall QoE in the adjacency should be the same just before and after the HO.

From the previous observation, it can be inferred that changes in the overall QoE measured after a HO event reflect the impact of changing the HO point. Since the HO point is displaced by tuning the HOM, such QoE differences before and after the HO can be used to derive the sign and approximate the magnitude of the gradient of the objective function (i.e., the overall system QoE,  $QoE_T$ ) with respect to the decision variables (i.e.,  $HOM(i, j)$ ),  $\frac{\partial QoE_T}{\partial HOM(i, j)}$ . This information can then be used to implement a gradient ascent method to optimize the overall system QoE. To that end, a self-tuning algorithm is proposed to adjust  $HOM(i, j)$  per adjacency so that the overall QoE in the adjacency is the same before and after HOs events on average. For this purpose, the QoE of individual connections around HO events must be estimated from connection traces.

**Description of the driver** a user  $u$  performing a HO  $k$  between two neighbor cells experiences a change of QoE defined as

$$\Delta QoE_u^{(k)} = QoE_{afterHO}^{(k)} - QoE_{beforeHO}^{(k)}, \quad (6.6)$$

where  $QoE_{beforeHO}^{(k)}$  and  $QoE_{afterHO}^{(k)}$  are the QoE experienced by the user just before and just after the HO, respectively, measured in MOS scale, i.e., ranging from 1 (bad) to 5 (excellent). For a user handed over from cell  $i$  to cell  $j$ ,  $QoE_{beforeHO}$  is measured in cell  $i$  and  $QoE_{afterHO}$  is measured in cell  $j$ . A positive value of  $\Delta QoE_u^{(k)}$  implies that user satisfaction improves after HO.

Both  $QoE_{beforeHO}^{(k)}$  and  $QoE_{afterHO}^{(k)}$  can be computed from user performance information in connection traces by using utility functions mapping objective QoS metrics (e.g., throughput) to Mean Opinion Score (MOS) values, ranging to 1 from 5. Examples of these utility functions are provided in appendix [A](#). For the sake of robustness, the time window used to measure the performance indicators in these equations must be long enough to provide representative information of the current user QoE in the serving cell, but short enough to isolate the impact of the HO from other events when computing  $QoE_u^{(k)}$ . In this work, a 500-ms window is established. This value is long enough to reduce the impact of the throughput ramp-up effect due to TCP slow-start [\[104\]](#) and outer loop link adaptation convergence [\[105\]](#) for throughput-sensitive services (e.g., web browsing or FTP).

In addition, any HO modifies the number of simultaneous users in both source and target cells (i.e., source cell loses a user, while target cell gains a user). As a result, the HO will also affect other users in such cells. Thus, the change in the overall QoE in the adjacency due to a HO  $k$  can be calculated as

$$\Delta QoE_T^{(k)} = \sum_{u \in \{i,j\}} \Delta QoE_u^{(k)}, \quad (6.7)$$

where  $u \in \{i, j\}$  represents all users served by cells  $i$  and  $j$  when the HO is executed and  $\Delta QoE_u^{(k)}$  is the change in QoE of user  $u$ , defined in [\(6.6\)](#). Note that, for the user performing the HO,  $QoE_{beforeHO}$  and  $QoE_{afterHO}$  in [\(6.6\)](#) are calculated in different cells, and the performance difference is due to the change of serving cell. In contrast, for the rest of users, both terms are calculated in the same cell (as their serving cell does not change), and the performance difference comes from the cell load change.



The indicator used as a driver to tune inter-frequency HOMs on an adjacency basis is the average QoE change after a HO in the adjacency, defined as

$$\overline{\Delta QoE_T}(i, j) = \frac{1}{N_{HO}(i, j)} \sum_{k=1}^{N_{HO}(i, j)} \Delta QoE_T^{(k)}, \quad (6.8)$$

where  $N_{HO}(i, j)$  is the number of HOs performed from cell  $i$  to cell  $j$  during a certain ROP.

A negative value of  $\overline{\Delta QoE_T}(i, j)$  indicates that, on average, the overall user satisfaction in cells  $i$  and  $j$  decreases when HOs are performed from  $i$  to  $j$ , and thus the number of these HOs must be reduced (i.e., HOMs must be more restrictive). In contrast, a positive value of  $\overline{\Delta QoE_T}(i, j)$  indicates that user satisfaction increases when HOs are performed from  $i$  to  $j$ , and thus the number of these HOs must be increased (i.e., HOMs must be less restrictive). The optimal HO point is given by the condition  $\overline{\Delta QoE_T}(i, j) = 0$ . At that point, on average, QoE does not experience any degradation because of HOs.

A proof of concept (not shown here for brevity) has been carried out in a pilot LTE network that confirms the feasibility of building  $\overline{\Delta QoE_T}(i, j)$  indicator by processing data in individual radio connection traces [252].

**Control algorithm** Algorithm 1 outlines the operation of the self-tuning algorithm. It is designed as a set of proportional controllers (1 per adjacency) that iteratively modify inter-frequency HOMs,  $HOM_{inter}(i, j)$ , based on the value of above-described indicator  $\overline{\Delta QoE_T}(i, j)$ . In each iteration, the value of  $HOM_{inter}(i, j)$  is tuned incrementally on a per-adjacency basis. Specifically, the increment/decrement in the HOMs,  $\Delta HOM_{inter}(i, j)$ , is computed from the value of  $\overline{\Delta QoE_T}(i, j)$  as

$$\Delta HOM_{inter}(i, j) = \begin{cases} 1 & \overline{\Delta QoE_T}(i, j) < \beta_1, \\ 0 & \beta_1 \leq \overline{\Delta QoE_T}(i, j) \leq \beta_2, \\ -1 & \overline{\Delta QoE_T}(i, j) > \beta_2, \end{cases} \quad (6.9)$$

where  $\beta_1$  and  $\beta_2$  are thresholds for triggering HOM changes so as to eliminate random actions due to small fluctuations of drivers. These parameters must be set to provide an adequate trade-off between optimality and complexity (in this work,  $\beta_2 = -\beta_1 = 0.05$ ). Larger absolute values reduce the number of iterations required to reach equilibrium

---

**Algorithm 1** QoE-driven self-tuning algorithm.

---

```

repeat
  Collect connection traces during ROP
  for all adjacencies  $(i, j) | \{i \in L_{cov}, j \in L_{cap}\}$  do
    Compute  $\overline{\Delta QoE_T}(i, j)$ 
    if  $\overline{\Delta QoE_T}(i, j) > \beta_1$  then
       $\Delta HOM_{inter}(i, j) = 1$ 
    else if  $\overline{\Delta QoE_T}(i, j) < \beta_2$  then
       $\Delta HOM_{inter}(i, j) = -1$ 
    else
       $\Delta HOM_{inter}(i, j) = 0$ 
    end if
    Update  $HOM_{inter}(i, j)$  value
  end for
until the predetermined number of loops is reached

```

---

at the expense of deteriorating network performance slightly, since the optimization process stops before  $\overline{\Delta QoE_T}(i, j) = 0$ .

The algorithm is executed a predetermined number of times (referred as to optimization loops). In every loop, connection traces are collected during a predetermined ROP (e.g., 15 min). Then, the algorithm computes the value of  $\overline{\Delta QoE_T}(i, j)$  in each inter-frequency adjacency  $(i, j)$  where HOs are RSRQ-driven. Finally, the new value of  $HOM_{inter}(i, j)$  is computed as

$$HOM_{inter}^{(n+1)}(i, j) = HOM_{inter}^{(n)}(i, j) + \Delta HOM_{inter}^{(n)}(i, j), \quad (6.10)$$

where superscripts  $(n)$  and  $(n + 1)$  denote iteration index.

Note that the chosen 1-dB step in (6.9) provides an adequate trade-off between fast convergence and stability. Lower values of  $\Delta HOM_{inter}(i, j)$  make optimization too slow, while too large values lead to abrupt changes in HOMs, both degrading network performance. Moreover, to guarantee an adequate HO performance,  $HOM_{inter}(i, j)$  values are limited to the range  $[-7, 7]$  dB [73]. The lower bound avoids too early HOs and ping-pong effect, while the upper bound avoids too late HOs, which can degrade user experience.

The proposed self-tuning algorithm performs small changes in the value of HOMs

iteratively (+/- 1 dB) until equilibrium is reached (i.e.,  $\overline{\Delta QoE_T}(i, j) = 0$ ). This equilibrium condition leads to a local maximum of the problem. However, due to the heuristic nature of the controller, convergence is not guaranteed. In practice, feedback loop gain is small enough to avoid oscillations in the system. In addition, thresholds  $\beta_1$  and  $\beta_2$  in (6.9) ensure that the controller stops when the value of  $\overline{\Delta QoE_T}(i, j)$  is small in every adjacency.

Note that every change performed by the algorithm only affects inter-frequency HOs from the coverage to the capacity layer. All other mobility mechanisms keep the default settings during the optimization process (e.g.,  $HOM_{intra}^{(n)}(i, j) = 3 \text{ dB } \forall i, j, n$ ). Moreover, these changes only affect HO triggering condition. Neither TTT nor HO execution procedures are modified at any time. As a consequence, the proposed self-tuning algorithm does not increase latency in the HO process, but just changes the condition that must be fulfilled to initiate the HO.

### 6.2.3 Performance assessment

This section presents the validation of the proposed traffic steering strategy through simulation. For clarity, assessment methodology is first described, results are presented later and computational complexity is finally discussed.

#### a) Assessment methodology

Validation is performed with the simulator described in [A], which emulates the activity of a live LTE-Advanced cellular network. Among the scenarios implemented in the tool, network B (i.e., multi-carrier network) is selected, comprising 48 cells distributed in 8 sites located in a dense urban area. Cells 1–24 work at 700 MHz (hereafter,  $L_{700}$ ) and cells 24 to 48 work at 2100 MHz (hereafter,  $L_{2100}$ ).  $L_{700}$  acts as a coverage layer, offering better propagation conditions but reduced cell bandwidth (1.4 MHz). In contrast,  $L_{2100}$  is a capacity layer with worse propagation conditions but a higher capacity (cell bandwidth of 5 MHz). Four different services are considered, namely VoIP, progressive video streaming (VIDEO), file download via FTP (FTP) and web browsing via HTTP (HTTP). User QoE is measured with the MOS models in (A.1)–(A.8), considering that indoor users have higher expectations than outdoor users. Network slicing feature is disabled (i.e., all users share radio resources). For further details on the simulation tool and service models, the reader is referred to appendix [A].

Four different mobility management methods are tested. A first method, referred to as Operator Solution (OS), considers mobility procedures configured in the live network, namely: a) idle users select carrier based on a token algorithm, b) cell (re)selection is then performed to select the best cell according to RSRP measurements), c) connected users are handed over according to the SBHO scheme shown in Fig. 6.1, and d) no MLB algorithm is enabled. Initial HO set-up is  $HOM(i, j)=3$  dB and  $TTT(i, j)=256$  ms for event A3 and  $thd1(i)=-115$  dBm and  $thd2(j)=-108$  dBm for event A5  $\forall i, j$ . This method is considered as a baseline. The other three methods consist of combinations of HO scheme and self-tuning algorithm. A first combination, denoted as SBHO+LB (Load Balancing), tackles traffic steering by executing a classical load balancing algorithm [30] to adjust  $HOM_{inter}(i, j)$  values in the SBHO scheme (i.e., RSRP-driven inter-frequency HOs). A second combination, denoted as QBHO+LB, executes the same load balancing algorithm to adjust  $HOM_{inter}(i, j)$  in the quality-based mobility scheme (i.e., RSRQ-driven inter-frequency HOs) presented in section 6.2.2. Finally, the third combination, denoted as QBHO+OE, is the two-stage strategy proposed in this work, modifying  $HOM_{inter}(i, j)$  values in QBHO scheme with OE algorithm. In all combinations, 10 optimization loops are simulated. Each loop consists of 15 min of network activity, which is the minimum ROP to collect traces in current LTE networks. It is checked a posteriori that 10 loops are enough to ensure that control system reaches steady state with the different self-tuning algorithms. For a fair comparison, every optimization loop is carried out under identical conditions by pre-generating all random variables. Thus, performance differences between loops are only due to the different mobility settings and not to the stochastic nature of simulation.

Two use cases are considered to check the impact of user context on the proposed strategy, referred to as cases A and B. In case A, 70% of users are indoors and 30% are outdoors. In case B, all users are outdoors. The four analyzed methods (OS, SBHO+LB, QBHO+LB and QBHO+OE) are tested in both scenarios.

The main FoM to assess method performance is the global QoE, computed as the average user QoE in the scenario,

$$QoE_{global} = \frac{1}{N_u} \sum_u QoE(u), \quad (6.11)$$

where  $N_u$  is the number of users in the scenario and  $QoE(u)$  is the session QoE experienced by user  $u$ , computed with the QoE models in (A.1)–(A.8).

Two secondary FoMs are considered for a more detailed assessment. The first one is

the average cell load in terms of PRB utilization ratio,  $\overline{PRB_{util}}$ , measured globally and on a per-layer basis. The second one is the average HOM deviation from the default settings caused by the corresponding MLB algorithm (i.e., LB or OE), computed as

$$\overline{\delta HOM_{inter}^{(n)}} = \frac{\sum_{(i,j)} \delta HOM_{inter}^{(n)}(i,j)}{N_a} = \frac{\sum_{(i,j)} \left( HOM_{inter}^{(n)}(i,j) - HOM_{inter}^{(0)}(i,j) \right)}{N_a}, \quad (6.12)$$

where  $N_a$  is the total number of adjacencies in the network where parameters tuning is performed, and  $HOM_{inter}^{(0)}(i,j)$  is the default inter-frequency HOM value at the beginning of the optimization process (i.e., iteration 0).

## b) Results

Table 6.1 shows some relevant performance metrics obtained with the initial HO settings (OS) in cases A and B. Recall that simulation parameters are set so that network performance resembles that of the live network. With the token-based cell (re)selection mechanism used by the operator, approximately 60% of users are served by cells at  $L_{2100}$  and 40% of users are served by cells at  $L_{700}$ , even if  $L_{2100}$  has larger bandwidth than  $L_{700}$ . Moreover, traffic demand is dominated by VIDEO users, followed by FTP users. In both cases, the average cell load in  $L_{2100}$  is less than that of  $L_{700}$  (about 30% in case A and 40% in case B). As a consequence of the high load in  $L_{700}$ , many users in this carrier experience poor QoE (on average, 1.91 MOS points in case A, and 2.49 MOS points in case B). These results point out the need for steering users from  $L_{700}$  to  $L_{2100}$ .

**Case A (indoor/outdoor users)** Fig. 6.4 shows the evolution of  $QoE_{global}$  across iterations in the tuning process. In all methods, loop 0 represents OS performance. In SBHO+LB, loops 1 to 9 show the behavior of the legacy approach that executes a classical LB algorithm in a SBHO scheme. In QBHO+LB curve, loop 1 shows the impact of enabling RSRQ-driven inter-frequency HOs and loops 2 to 9 show the impact of adjusting inter-frequency HOMs by LB in a QBHO scheme. Similarly, in QBHO+OE, loop 1 shows the effect of enabling RSRQ-driven inter-frequency HOs and loops 2 to 9 show the impact of adjusting inter-frequency HOMs based on the novel trace-based indicator in OE. Large markers indicate the final  $QoE_{global}$  achieved with each method. It is observed that SBHO+LB does not have a significant impact on QoE. In contrast, both QBHO+LB and QBHO+OE improve  $QoE_{global}$  compared to OS (3.75 and 3.91

Table 6.1: Initial performance of simulated two-tier LTE network.

FoM	$L_{700}$	$L_{2100}$	$L_{700}+L_{2100}$
Share of connections [%]	40.27	59.73	100
Data volume ratio VoIP [%]	$1.5 \cdot 10^{-5}$	$6 \cdot 10^{-5}$	$2.4 \cdot 10^{-3}$
Data volume ratio VIDEO [%]	26.51	57.27	53.86
Data volume ratio FTP [%]	50.98	32.11	32.14
Data volume ratio WEB [%]	22.50	10.62	13.99
$\overline{PRB}_{util}$ case A [%]	87.71	52.70	59.48
$QoE_{global}$ case A	1.91	4.35	3.32
$\overline{PRB}_{util}$ case B [%]	88.1	46.9	54.87
$QoE_{global}$ case B	2.49	4.38	3.57

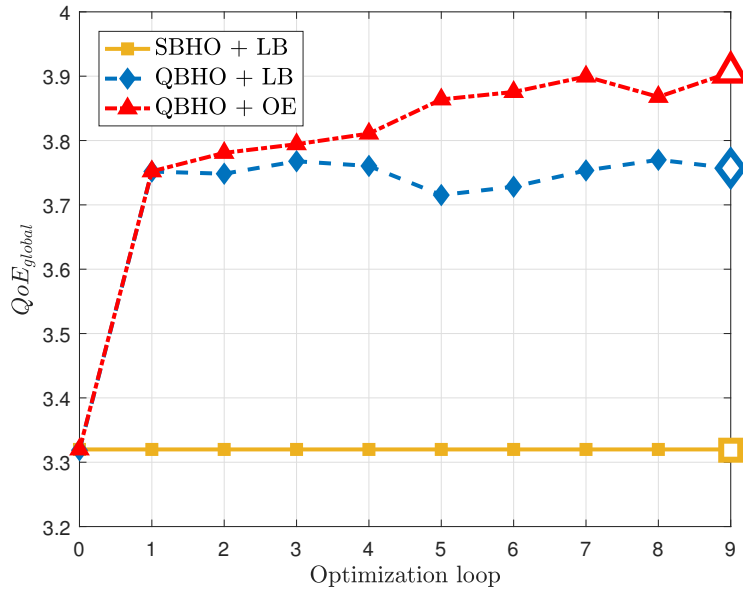


Figure 6.4: Evolution of the overall QoE in the scenario.

vs. 3.32 MOS points, respectively). The improvement obtained by QBHO+LB is essentially due to the activation of RSRQ-driven inter-frequency HOs, since no significant increase on  $QoE_{global}$  is shown from loop 1 onwards (i.e., again, LB does not improve  $QoE_{global}$ ). This result is consistent with the fact that LB does not aim to improve QoE, but to equalize cell load between layers. In contrast, QBHO+OE obtains an additional gain thanks to the OE algorithm, resulting in the highest  $QoE_{global}$ .

It should be pointed out that, even if OE aims to optimize user experience, a decrease of 0.03 MOS points in  $QoE_{global}$  is observed from loop 7 to loop 8 in QBHO+OE, due to the iterative nature of the controller. As in most closed-loop control systems,

Table 6.2: Performance comparison of MLB strategies in a two-tier LTE network – case A (indoor/outdoor).

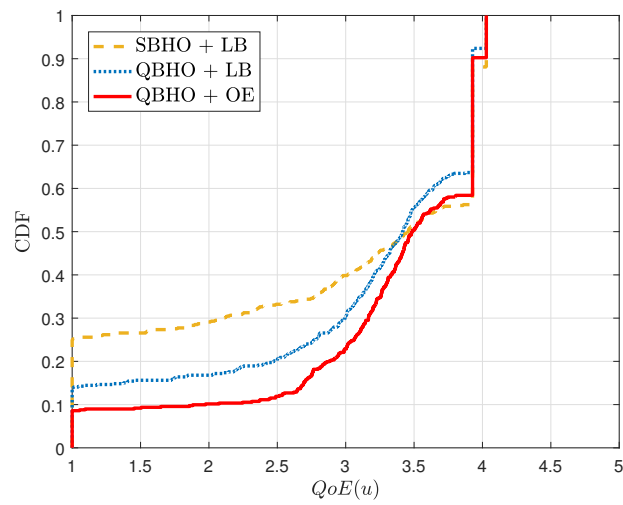
FoM	OS	SBHO+LB	QBHO+LB	QBHO+OE
$QoE_{global}$	3.32	3.32	3.75	3.91
Global $\overline{PRB}_{util}$ [%]	59.48	59.48	79.36	72.22
$\overline{PRB}_{util} L_{700}$ [%]	87.71	87.71	79.73	26.29
$\overline{PRB}_{util} L_{2100}$ [%]	52.70	52.70	77.79	83.25
$\delta HOM_{inter}^{(9)}$ [dB]	–	-5.97	-0.35	-3.38

small oscillations in system performance are observed when the controller reaches the steady state. Note that the decrease in  $QoE_{global}$  from loop 7 to 8 is negligible (0.03 MOS points) and is compensated in the following optimization loop.

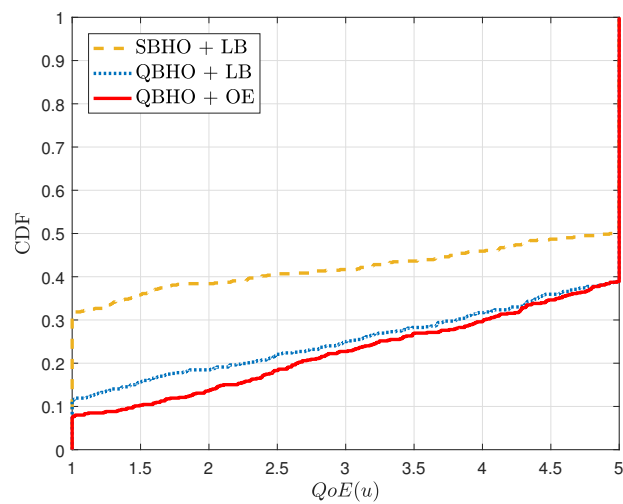
Fig. 6.5(a)–c) show the CDF of user QoE,  $QoE(u)$ , for VIDEO, FTP and WEB services with the different methods. VoIP is omitted since its traffic is negligible. Likewise, OS is not included, as its performance is identical to SBHO+LB. It is observed that, for all services, QBHO+LB and QBHO+OE improve QoE distribution compared to SBHO+LB, with QBHO+OE achieving the best QoE figures. In HTTP and VIDEO services, such an improvement is achieved at the expense of a slight decrease in  $QoE(u)$  for the best users. Note that many VIDEO users experience a  $QoE(u)$  value of 4.02 and 3.92, corresponding to the upper limiting values in the outdoor and indoor QoE models when the initial buffering time is 3 s (fixed value [146]).

For a more detailed analysis, Table 6.2 breaks down several statistics for the tested methods at the end of the optimization process (loop 9 in Fig. 6.4). OS is also included for comparison purposes. Regarding the main FoM,  $QoE_{global}$ , both QBHO+LB and QBHO+OE outperform OS, with QBHO+OE achieving the largest improvement (3.91 against 3.32, i.e., a 17.8% improvement compared to OS).  $\overline{PRB}_{util}$  values show that QBHO+OE obtains such a gain by offloading traffic from  $L_{700}$  to  $L_{2100}$ , since  $\overline{PRB}_{util}$  increases in  $L_{2100}$  (from 52.70% to 83.25%) and decreases in  $L_{700}$  (from 87.71% to 26.29%). This traffic steering is also confirmed by the negative value of  $\delta HOM_{inter}^{(9)}$  (i.e., -3.38 dB). As a side effect, QBHO+OE increases the global  $\overline{PRB}_{util}$  from 59.48% to 72.22% (i.e., a 12.74% increase in absolute terms).

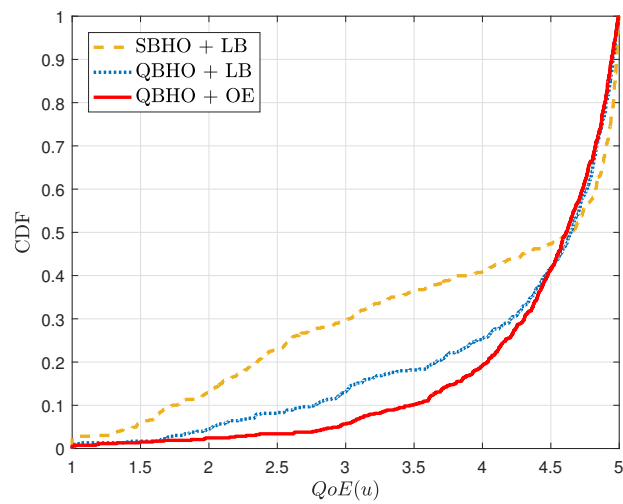
**Case B (outdoor users)** Table 6.3 summarizes the results when all users are outdoors. Most indicators show trends similar to those in case A. Again, in terms of  $QoE_{global}$ , both QBHO+LB and QBHO+OE outperform OS and SBHO+LB, with QBHO+OE achieving the largest improvement (4.25 vs. 3.57, i.e., a 19 % improve-



(a) VIDEO.



(b) FTP.



(c) WEB.

Figure 6.5: Cumulative distribution function of user QoE for different services.



Table 6.3: Performance comparison of MLB strategies in a two-tier LTE network – case B (outdoor).

FoM	OS	SBHO+LB	QBHO+LB	QBHO+OE
$QoE_{global}$	3.57	3.57	4.17	4.25
Global $\overline{PRB_{util}}$ [%]	54.87	54.87	70.99	68.11
$\overline{PRB_{util}}_{L_{700}}$ [%]	88.10	88.10	72.59	34.01
$\overline{PRB_{util}}_{L_{2100}}$ [%]	46.90	46.90	70.61	76.30
$\delta HOM_{inter}^{(9)}$ [dB]	–	-7.55	0.42	-3.09

ment compared to OS). In this case, differences between QBHO+OE and QBHO+LB are lower, since the value of  $QoE_{global}$  after enabling RSRQ-driven HOs in step 1 is already high (4.16 MOS points). Also, note that, from the  $\delta HOM_{inter}^{(9)}$  figures, it can be deduced that QBHO+LB and QBHO+OE steer traffic in opposite directions, leading to different traffic shares between layers. Specifically,  $\delta HOM_{inter}^{(9)}$  is negative in QBHO+OE (i.e., traffic is offloaded from  $L_{700}$  to  $L_{2100}$ ), whereas it is positive for QBHO+LB (i.e., traffic is offloaded from  $L_{2100}$  to  $L_{700}$ ).

It is worth noting that, in both cases A and B, even if QBHO+LB achieves a more evenly loaded scenario between the two tiers, it is QBHO+OE that reaches the best QoE. This is clear evidence that an evenly loaded network (which is the aim of classical MLB algorithms) does not lead to the best overall QoE, provided that system bandwidth is not the same in all cells [253]. Finally, it is also remarkable that no improvement in  $QoE_{global}$  is achieved with SBHO+LB, even if HO margins are shifted nearly  $-6$  dB in case A and  $-7.5$  dB in case B. This is due to the different propagation conditions in the two carriers that make it extremely difficult to trigger RSRP-driven event A3.

### c) Computational complexity

The proposed MLB method is designed as a rule-based controller and therefore has low computational complexity. The total execution time comprises the time required to process connection traces, the computation of the indicator reflecting the average impact of inter-frequency HOs on user QoE per adjacency,  $\overline{\Delta QoE_T}(i, j)$ , and the computation of the output of the controller. In practice, the total execution time in the above-described scenario in a computer with an Intel Xenon processor with a clock frequency of 2.4 GHz and 64 GB of RAM is 0.022 seconds per optimization loop (note that trace processing is not needed in simulations). The dominant operation is the com-

putation of  $\overline{\Delta QoE_T}(i, j)$ , with 0.016 seconds per optimization loop. The time taken by this operation grows linear with the number of users affected by HO events. Thus, the worst-case time complexity of the algorithm is  $\mathcal{O}(N_{HO} \times N_u)$ , where  $N_{HO}$  is the total number of HOs and  $N_u$  is the average number of active users.

### 6.2.4 Conclusions

In this section, a novel strategy for traffic steering in multi-tier LTE networks has been proposed to improve the overall system QoE. For this purpose, RSRQ-driven inter-frequency HOs are first enabled, and, later, a novel MLB algorithm that adjusts inter-frequency HOMs on a per-adjacency basis driven by QoE measurements is executed. In each adjacency, an independent controller increases (or decreases) the value of HOMs based on an indicator showing the impact of HOs on overall user satisfaction. Such an indicator is computed by processing data in connection traces.

Performance assessment has been carried out in a dynamic system-level simulator implementing a realistic scenario. Results have shown that the proposed algorithm outperforms classical load balancing techniques. Specifically, the overall QoE is improved by up to 19% compared to a traditional load balancing algorithm executed over a legacy RSRP-driven inter-frequency HO scheme. Such a performance gain is achieved by offloading traffic from coverage layers to capacity layers, so that users make the most of the large bandwidth available at capacity layers.

## 6.3 SLA-driven traffic steering in sliced radio access networks

The above-presented QBHO+OE algorithm is suitable for HSDPA and LTE networks, where the variety of services offered requires QoE-driven self-optimization solutions to guarantee customer satisfaction. In contrast, in NGNs with NS, performance targets per service are included in slice-specific SLAs, suggesting the design of SLA-driven SON tools. This section addresses slice-aware traffic steering in sliced RANs to improve SLA compliance. For this purpose, section [6.3.1](#) formulates the problem of performing slice-aware MLB. Then, section [6.3.2](#) presents a novel slice-aware traffic steering algorithm, assessed via simulation in section [6.3.3](#). Finally, section [6.3.4](#) summarizes the main conclusions.

### 6.3.1 Problem formulation

Consider a cellular network with NS where a set of  $N_s$  slices, denoted as  $\mathcal{S}$ , operate simultaneously. In the RAN, the network comprises  $N_c$  cells, denoted as  $\mathcal{C}$ , working at the same frequency band, so that every cell  $c \in \mathcal{C}$  may serve users from all active slices  $s \in \mathcal{S}$ . As typical in live networks, intra-frequency mobility is handled through power-budget HOs. Thus, the HO of a user  $u$  from serving cell  $i$  to a neighbor cell  $j$  is triggered by the condition (6.2) replacing  $HOM_{inter}(i, j)$  by  $HOM_{intra}(i, j)$ , i.e.,

$$RSRP_u(j) \geq RSRP_u(i) + HOM_{intra}(i, j). \quad (6.13)$$

In this scenario, a legacy MLB algorithm would adjust  $HOM(i, j)$  to steer traffic from congested to underutilized cells so that cell load is balanced. With the above HO scheme, parameter self-tuning can be performed on a per-adjacency basis. However, note that a certain HO set-up in a given adjacency may not lead to the same performance for all slices due to: a) the different traffic characteristics (e.g., user spatial distribution, mobility, performance requirements in SLA...) among slices, and b) the capacity broker, which may underestimate/overestimate resources required by a particular slice in a particular cell (or area), but not for others. This fact suggests the need for slice-aware MLB algorithms. For this purpose, a slice-aware HO scheme must be set first. The triggering equation for slice-aware RSRP-based HO event A3 can be expressed as

$$RSRP_u(j) \geq RSRP_u(i) + HOM(i, j, s_u), \quad (6.14)$$

where  $s_u$  is the slice to which user  $u$  belongs. With this new HO scheme, HOM tuning can be performed per adjacency and slice.

The aim of slice-aware MLB algorithms must be guaranteeing SLA compliance (and thus both tenant and end-user satisfaction) for all slices. However, as stated in [87], an evenly loaded scenario does not ensure that all cells offer the same performance (e.g., due to different radio link conditions). This behavior is expected to worsen in sliced RANs. In these networks, equalizing global load of neighbor cells can have a negligible (or even negative) impact on SLA compliance for slices offering services with low rate, whose performance is jeopardized by eMBB slices with a larger radio resource allocation. Moreover, even for eMBB slices, each slice may access to a different amount

of PRBs in each cell, and thus the difference in spare PRBs per slice among neighbor cells may be different from the global load imbalance. As a consequence, slice-aware MLB strategies must be SLA-driven (and not load-driven, as legacy solutions).

When designing slice-aware SLA-driven MLB algorithms, it should be taken into account that optimizing network performance globally sometimes compromises cell-edge users (i.e., those with the highest risk of violating SLA). Such an issue is circumvented by approaches that equalize performance among cells, as that presented in [33]. Likewise, a high dynamism is expected in sliced 5G networks due to slice activation, deactivation and resource reallocation [35]. As a consequence, slice-aware traffic steering must operate in a time resolution finer than legacy schemes, where HOMs are tuned based on performance counters updated every 15 minutes at most. Due to such dynamism, indicators driving the MLB process must reflect slice performance in the last few seconds, which can only be obtained by processing connection traces. All these aspects are considered by the traffic steering algorithm proposed here.

### 6.3.2 Traffic steering strategy

This section, a novel slice-aware MLB algorithm is presented. The algorithm aims to equalize the level of SLA compliance per slice across the scenario by steering traffic among cells working at the same frequency band. For this purpose,  $HOM_{intra}(i, j, s)$  in the slice-aware HO scheme presented in (6.14) is self-tuned on a per-adjacency-and-slice basis. As a novelty, the driver indicator reflects the imbalance of SLA compliance per slice in neighbor cells.

When enabling the slice-aware intra-frequency HO scheme in (6.14), an initial value of  $HOM_{intra}(i, j, s)=3$  dB is set  $\forall i, j, s$ , as starting point for traffic steering. To prevent ineffective parameter changes, the MLB algorithm only operates on a subset of adjacencies denoted as  $\mathcal{A}$ , comprising a limited number of relevant adjacencies per cell. Moreover, to avoid that changing several HOMs for a cell simultaneously leads to excessive reduction/increase of cell area for a slice, HOM tuning is not performed simultaneously for all adjacencies in  $\mathcal{A}$ . For clarity, the adjacency selection and clustering strategy is first detailed and the self-tuning algorithm is presented later.

---

**Algorithm 2** Adjacency clustering algorithm.

---

```

Create  $\mathcal{A}$ 
Initialize empty  $\mathcal{G}$ 
 $k=1$ 
repeat
  Initialize empty group  $g_k$ 
  for all adjacencies  $a = (i \leftrightarrow j) \in \mathcal{A}$  do
    if any adjacency  $a' \in g_k$  contains cell  $i$  or  $j$  then
      Add  $a$  to group  $g_k$ 
      Remove  $a$  from  $\mathcal{A}$ 
    end if
  end for
  Add group  $g_k$  to  $\mathcal{G}$ 
   $k=k+1$ 
until  $\mathcal{A}$  is empty
 $N_g = k$ 

```

---

**Stage 1. Adjacency selection and clustering**

The subset of adjacencies in the whole network where MLB will operate,  $\mathcal{A}$ , is created as follows. For every cell  $c$  in the scenario, a fixed number of relevant neighbors cells,  $N_n$ , is selected. That set of  $N_n$  neighbors per cell  $c$ , denoted as  $\mathcal{N}(c)$ , includes: a) all co-sited cells, and b) the most interfering cells in the Down Link (DL) from nearby sites. Then, bidirectional adjacencies  $(c \leftrightarrow j) \forall j \in \mathcal{N}(c)$  are included in  $\mathcal{A}$ . After repeating this process for all cells in the scenario, the number of adjacencies in  $\mathcal{A}$  is  $N_c \times N_n$ . Then, duplicated adjacencies in  $\mathcal{A}$  (if any) are removed.

Next, adjacencies in  $\mathcal{A}$  are divided into a set of disjoint groups,  $\mathcal{G}$ . Clustering is performed with the heuristic scheme presented in Algorithm 2, inspired in [69]. Groups are created sequentially. For each group  $g_k$ , a random adjacency from  $\mathcal{A}$  is first selected as seed and removed from  $\mathcal{A}$ . Then, another adjacency in  $\mathcal{A}$  is randomly selected to be added to group  $g_k$  if it does not include any cell in the adjacency previously added to group  $g_k$ . More adjacencies are sequentially added to group  $g_k$  until no adjacency in  $\mathcal{A}$  comprises disjoint cells with all adjacencies already in the group. Then, a new group  $g_{k+1}$  is created. This process is repeated until  $\mathcal{A}$  becomes empty.

The subsequent MLB iteratively tunes HOMs. In each iteration  $k$ , only HOMs from adjacencies in group  $g_k$  are modified. As a consequence, the number of groups in

$\mathcal{G}$ ,  $N_g$ , determines how often parameters change per adjacency. Since  $N_g$  grows with  $N_n$ , to ensure fast and optimal convergence,  $N_n$  must have the lowest value allowing to include all relevant adjacencies per cell in  $\mathcal{A}$ . Nonetheless,  $N_g$  may vary in different executions if the random seed changes. It is recommended to perform multiple runs of the clustering algorithm with different seeds before optimization starts, and select the solution providing the lowest  $N_g$ .

It should be pointed out that HOM tuning reshapes cell serving area, and thus DL interference may change once the tuning process begins (e.g., due to cell load changes). However, it is strongly recommended to perform the above adjacency clustering process with a stable HOM set-up and redefine it only after a significant event altering radio link performance in the network (e.g., deployment of a new cell).

## Stage 2. SLA-driven HOM tuning

Once adjacency groups  $\mathcal{G}$  have been created, the slice-aware self-tuning algorithm detailed below is executed. For clarity, the indicator driving the tuning process is described first and the control algorithm is presented later.

**Description of the driver** The average level of SLA compliance for slice  $s$  in the DL of a given cell  $c$  during a certain period of time can be expressed as

$$\overline{SLA}(c, s) = \frac{1}{N_u(c, s)} \sum_{u=1}^{N_u(c, s)} SLA(u, c), \quad (6.15)$$

where  $N_u(c, s)$  is the number of users from slice  $s$  with relevant activity in the DL of cell  $c$ , i.e., those with data to be transmitted in at least 5% of transmission time intervals during the considered time period, and  $SLA(u, c)$  is the level of SLA compliance for user  $u$  belonging to slice  $s$  in cell  $c$ .  $SLA(u, c)$  is computed as

$$SLA(u, c) = \sum_{p=1}^{N_{KPI}(s_u)} w_p(s_u) SLA_p(u, c), \quad (6.16)$$

where  $N_{KPI}(s_u)$  is the number of KPIs included in the SLA for slice  $s_u$  to which user  $u$  belongs,  $w_p(s_u)$  is a weight factor showing the relative importance of KPI  $p$  for the performance of slice  $s_u$ , and  $SLA_p(u, c)$  is the level of SLA compliance related to KPI  $p$  for user  $u$  served by cell  $c$ .  $w_p(s_u)$  ranges from 0 to 1, so that  $\sum_{p=1}^{N_{KPI}(s)} w_p(s) = 1 \forall s$ .

Likewise,  $SLA_p(u, c)$  is calculated as

$$SLA_p(u, c) = \min \left( \frac{KPI_p(u, c)}{KPI_p^{tgt}(s_u)}, SLA_{max} \right), \quad (6.17)$$

where  $KPI_p(u, c)$  denotes performance of KPI  $p$  for user  $u$  in cell  $c$ ,  $KPI_p^{tgt}(s_u)$  is the performance target for KPI  $p$  in the SLA of slice  $s_u$ , and  $SLA_{max}$  is a maximum level of SLA compliance to avoid that users exceeding the SLA conceal those with worse performance in (6.15).

The indicator driving HOM tuning is the difference of SLA compliance levels for slice  $s$  in the two cells  $i$  and  $j$  of an adjacency,  $\overline{SLA}_{diff}(i, j, s)$ , defined as

$$\overline{SLA}_{diff}(i, j, s) = \overline{SLA}(j, s) - \overline{SLA}(i, s). \quad (6.18)$$

A negative value of  $\overline{SLA}_{diff}(i, j, s)$  indicates that, on average, the level of SLA compliance for slice  $s$  is better in cell  $i$  than in cell  $j$ , whereas a positive value of  $\overline{SLA}_{diff}(i, j, s)$  indicates the opposite. The HO point for a balanced scenario is given by the condition  $\overline{SLA}_{diff}(i, j, s)=0$ . At that point, on average, the level of SLA compliance for slice  $s$  is similar in both cells  $i$  and  $j$ .

**Control algorithm** Algorithm 3 outlines the operation of the self-tuning algorithm, designed as a set of proportional controllers (one per adjacency and slice) that iteratively modify  $HOM_{intra}(i, j, s)$  based on the value of  $\overline{SLA}_{diff}(i, j, s)$  indicator.

The algorithm is executed a predetermined number of optimization loops. Unlike in section 6.2, a loop comprises  $N_g$  iterations. The inter-iteration time (hereafter referred to as Tuning Interval, TI) must be short enough to reflect the current (and not past) network state, but long enough to get reliable computations of SLA compliance for services with bursty traffic (e.g., in this work, TI=5 s). In each iteration  $k$ , the HOM value for adjacencies in group  $g_k$  is tuned incrementally on a per-adjacency-and-slice basis. Specifically, the increment/decrement in HOM,  $\Delta HOM_{intra}(i, j, s)$ , is computed from the value of  $\overline{SLA}_{diff}(i, j, s)$  as

$$\Delta HOM_{intra}(i, j, s) = \begin{cases} 2 & \overline{SLA}_{diff}(i, j, s) < \alpha_1, \\ 0 & \alpha_1 \leq \overline{SLA}_{diff}(i, j, s) \leq \alpha_2, \\ -2 & \overline{SLA}_{diff}(i, j, s) > \alpha_2, \end{cases} \quad (6.19)$$

---

**Algorithm 3** SLA-driven slice-aware self-tuning algorithm.

---

Create  $\mathcal{A}$  and compute  $\mathcal{G}$  with Algorithm 2

**repeat**

**for all**  $k \in [1, N_g]$  **do**

    Wait for TI and collect connection traces

**for all** slices  $s \in \mathcal{S}$  **do**

**for all** adjacencies  $a = (i \leftrightarrow j) \in g_k$  **do**

        Compute  $\overline{SLA}_{diff}(i, j, s)$

**if**  $\overline{SLA}_{diff}(i, j, s) < \alpha_1$  **then**

$\Delta HOM_{intra}(i, j, s) = 2$

**else if**  $\overline{SLA}_{diff}(i, j, s) > \alpha_2$  **then**

$\Delta HOM_{intra}(i, j, s) = -2$

**else**

$\Delta HOM_{intra}(i, j, s) = 0$

**end if**

        Update  $HOM_{intra}(i, j, s)$  and  $HOM_{intra}(j, i, s)$  values

**end for**

**end for**

**end for**

**until** the predetermined number of loops is reached

---

where  $\alpha_1$  and  $\alpha_2$  are analog to  $\beta_1$  and  $\beta_2$  in (6.9) (in this work,  $\alpha_2 = -\alpha_1 = 0.05$ ). Then, the new value of  $HOM_{intra}(i, j, s)$  is computed as

$$HOM_{intra}^{(k+1)}(i, j, s) = HOM_{intra}^{(k)}(i, j, s) + \Delta HOM_{intra}^{(k)}(i, j, s). \quad (6.20)$$

To guarantee adequate HO performance,  $HOM_{intra}(i, j, s)$  values are limited to the range  $[-6, 12]$  dB. Finally, to avoid ping-pong effect, in all cases, a 6-dB hysteresis area is maintained by jointly setting HOMS in both directions of an adjacency so that  $HOM_{intra}(j, i, s) + HOM_{intra}(i, j, s) = 6$  dB.

Note that  $\overline{SLA}(c, s)$  for empty cells (i.e.,  $N_u(c, s) = 0$ ) must be set to a value higher than  $SLA_{max} + \max(|\alpha_1|, |\alpha_2|)$  to ensure that the MLB algorithm presented below offloads traffic to the empty cell. It is also remarkable that a larger step has been chosen in (6.19) compared to (6.9) (2 dB vs. 1 dB) since user density decreases when considering only traffic from a slice, and hence a higher change in HOM is required in slice-aware HO schemes to offload traffic from congested cells.



### 6.3.3 Performance assessment

This section presents the validation of the proposed slice-aware MLB algorithm. In the absence of commercial 5G networks with NS, method assessment is carried out with the simulation tool described in appendix [A](#). For clarity, the considered SLA definition is introduced first, assessment methodology is detailed next, results are presented later and computational complexity is finally discussed.

#### a) SLA definition

In the simulator, the contracted SLA is defined in terms of performance targets for the expected traffic in a given area. Two performance KPIs are considered, computed on a session level. The first KPI is DL session throughput,  $TH$ , defined as

$$TH(u) = \frac{V_{DL}(u)}{t_{session}(u)}, \quad (6.21)$$

where  $V_{DL}(u)$  is the total data volume transmitted to user  $u$  in the DL at PDCP layer, and  $t_{session}(u)$  is session duration. The second KPI is latency-reliability commitment,  $LR$ , defined as the ratio of packets transmitted in a session with an E2E latency below a predefined threshold [\[254\]](#), i.e.,

$$LR(u) = \frac{p_{succ}(u)}{p(u)}, \quad (6.22)$$

where  $p(u)$  is the total number of packets in the transmission buffer during the session of user  $u$  and  $p_{succ}(u)$  is the number of those packets fulfilling target E2E latency for slice  $s$  to which user  $u$  belongs<sup>1</sup>.

During experiments, the level of SLA compliance per user is computed in two different ways: a) per session, as a FoM to assess algorithm performance, and b) per session, cell and TI, to calculate  $\overline{SLA}_{diff}(i, j, s)$  indicator driving the HOM tuning process. In the latter case, with the above SLA definition, equation [\(6.16\)](#) can be particularized as

$$SLA(u, c) = w_{TH}(s_u)SLA_{TH}(u, c) + w_{LR}(s_u)SLA_{LR}(u, c). \quad (6.23)$$

<sup>1</sup>In the simulator, it is assumed that: a) a packet is a block of data to be transmitted, and b) E2E latency is the time from the packet arrives to transmission buffer until it is scheduled.

Table 6.4: Simulation set-up for assessing MLB strategies in a NS scenario.

Slice	Service	Speed	$TH^{tgt}(s)$	$LR^{tgt}(s)$
1	FTP LIVE VIDEO	3 km/h Static	1 Mbps	1 s for 90% of packets
2	HAPTIC	Static	400 kbps	10 ms for 99.9% of packets
3	DRIVING	30 km/h	16 kbps	10 ms for 99.9% of packets

Note that  $SLA_{TH}(u, c)$  and  $SLA_{LR}(u, c)$  must reflect SLA compliance in terms of  $TH$  and  $LR$  per user, cell and TI. In  $SLA_{TH}(u, c)$  calculation,  $TH(u, c)$  is computed as

$$TH(u, c) = \frac{V_{DL}(u, c, TI)}{t_{TI}(u, c)}, \quad (6.24)$$

where  $V_{DL}(u, c, TI)$  is the DL data volume transmitted to user  $u$  in cell  $c$  during the corresponding TI,  $TI$ , and  $t_{TI}(u, c)$  is the time period of  $TI$  where user  $u$  is served by cell  $c$ . Similarly, when computing  $SLA_{LR}(u, c)$ ,  $LR(u, c)$  only considers packets that arrive to the transmission buffer and are sent or dropped within the TI, i.e.,

$$LR(u, c) = \frac{p_{succ}(u, c, TI)}{p(u, c, TI)}. \quad (6.25)$$

For the above calculations, radio connection traces should be processed in a live environment.

## b) Assessment methodology

Validation is performed emulating the activity of network A (i.e., the largest scenario) with a system bandwidth of 10 MHz. Table 6.4 summarizes service and NS set-up. Three slices operate simultaneously in the network, which serve traffic from four different services, namely file download via FTP (FTP), live video streaming (LIVE VIDEO), haptic communications (HAPTIC) and autonomous driving (DRIVING). As in live networks, user speed depends on service. Slice 1 serves FTP and LIVE VIDEO users, with the highest  $TH$  requirement ( $TH^{tgt}(1)=1$  Mbps), but a relaxed target  $LR$  ( $LR^{tgt}(1)=1$  s for 90% of packets). FTP users are pedestrians moving at 3 km/h, whereas LIVE VIDEO users are static. Slice 2 serves HAPTIC traffic, generated from static users demanding a moderate  $TH$  (400 kbps) with stringent  $LR$  requirements (10 ms for 99.9% of packets). Finally, slice 3 serves DRIVING users moving at

30 km/h demanding a low rate (16 kbps), but with the same stringent  $LR$  requirements as HAPTIC users.

The proposed traffic steering algorithm, referred to as SLA-driven MLB over Slice-Aware HO scheme (SAHO+SLA) is compared with other three MLB strategies. The first, referred to as Load Balancing over Legacy HO scheme (LHO+LB), is a classical MLB algorithm that tunes HOMs per adjacency on a legacy (i.e., slice-unaware) HO scheme, whose aim is to balance PRB utilization across cells. The second, referred to as Load Balancing over Slice-Aware HO scheme (SAHO+LB), steers traffic on a per-adjacency-and-slice basis to balance PRB utilization of those PRBs assigned to each slice between adjacent cells. To justify the need for adjacency clustering in SAHO+SLA, a third strategy referred to as SLA-driven MLB with fast convergence over Slice-Aware HO scheme (SAHO+SLAfast) is considered, which applies the proposed slice-aware self-tuning algorithm, but omitting adjacency clustering (i.e., HOMs for all adjacencies in  $\mathcal{A}$  are tuned every TI). A simulation without MLB, referred to as No MLB, is also run as a benchmark.

For each of the above MLB strategies, 14 optimization loops (a total of 15 minutes of network activity) are simulated. In the starting point (i.e., TI=0), the adaptive capacity broker has already reached steady state. Therefore, resource allocation per slice remains fixed during the optimization process. The number of relevant neighbors per cell,  $N_n$ , is set to 6, with a total of 427 adjacencies ( $i \leftrightarrow j$ ) in the network to be optimized per slice. The adjacency clustering algorithm results in  $N_g=13$  groups of adjacencies. TI is set to 5 s. Recall that, in LHO+LB and SAHO+LB and SAHO+SLA, HOM is tuned once per optimization loop for each adjacency. With the above set-up, a loop lasts for  $5 \times 13=65$  s, which is a reasonable time to adapt to rapid changes in network conditions. Finally,  $SLA_{max}=1.2$  and  $w_{TH}(s) = w_{LR}(s) = 0.5 \forall s \in \mathcal{S}$ .

The main FoM to assess algorithm performance is the percentage of users complying SLA in terms of both  $TH$  and  $LR$ ,  $SLA_{global}$ , computed as

$$SLA_{global} = \frac{100}{N_u} \sum_u SLA_{bool}(u) \quad [\%], \quad (6.26)$$

where  $N_u$  is the number of users in the scenario and  $SLA_{bool}(u)$  is the boolean level of

SLA compliance per user  $u$ , computed as

$$SLA_{bool}(u) = \text{floor} \left( \frac{\text{floor} \left( \frac{TH(u)}{TH^{tgt}(s_u)} \right) + \text{floor} \left( \frac{LR(u)}{LR^{tgt}(s_u)} \right)}{2} \right). \quad (6.27)$$

This FoM is analyzed in absolute terms ( $SLA_{global}$ ) and relative to that obtained in the baseline case ( $SLA_{global}^{norm}$ ), i.e.,

$$SLA_{global}^{norm} = \frac{SLA_{global}}{SLA_{global \ baseline}}. \quad (6.28)$$

The overall SLA compliance per service,  $SLA_i \forall i \in \{FTP, LIVE \ VIDEO, HAPTIC, DRIVING\}$  is similarly computed.

Five secondary FoMs are also considered. The first is the final  $\overline{SLA}_{diff}(i, j, s)$  averaged for all the tuned adjacencies, showing the capacity of MLB strategies to balance SLA compliance among neighbor cells. The second is the final PRB utilization ratio across cells in the scenario,  $\overline{PRB}_{util}$ , as a proxy of resource usage. The third is the average absolute HOM deviation per slice from initial settings in the tuned adjacencies,  $|\overline{\delta HOM}_{intra}^{(n)}|(s)$ , computed as

$$\begin{aligned} |\overline{\delta HOM}_{intra}^{(n)}|(s) &= \frac{1}{N_a} \sum_{(i,j) \in \mathcal{A}} |\delta HOM_{intra}^{(n)}(i, j, s)| = \\ &= \frac{1}{N_a} \sum_{(i,j) \in \mathcal{A}} |HOM_{intra}^{(n)}(i, j, s) - HOM_{intra}^{(0)}(i, j, s)|, \end{aligned} \quad (6.29)$$

where  $n$  denotes optimization loop index,  $N_a$  is the number of adjacencies in  $\mathcal{A}$ , and  $HOM_{intra}^{(0)}(i, j, s)$  is the initial intra-frequency HOM value (i.e., in TI=0). Finally, the ratio between the number of HOs in a simulation compared to the baseline,  $nHO^{norm}$ , is also considered as a measure of the increase in signaling load caused by MLB.

### c) Results

To gain insight into how the different traffic steering algorithms work, Fig. [6.6](#)(a)–(c) show the evolution of  $|\overline{\delta HOM}_{intra}^{(n)}|(s)$  per slice across the optimization process obtained for all the tested MLB strategies. The (almost) stable level observed in the last optimization loops in all curves confirms that all algorithms converge for all slices. As

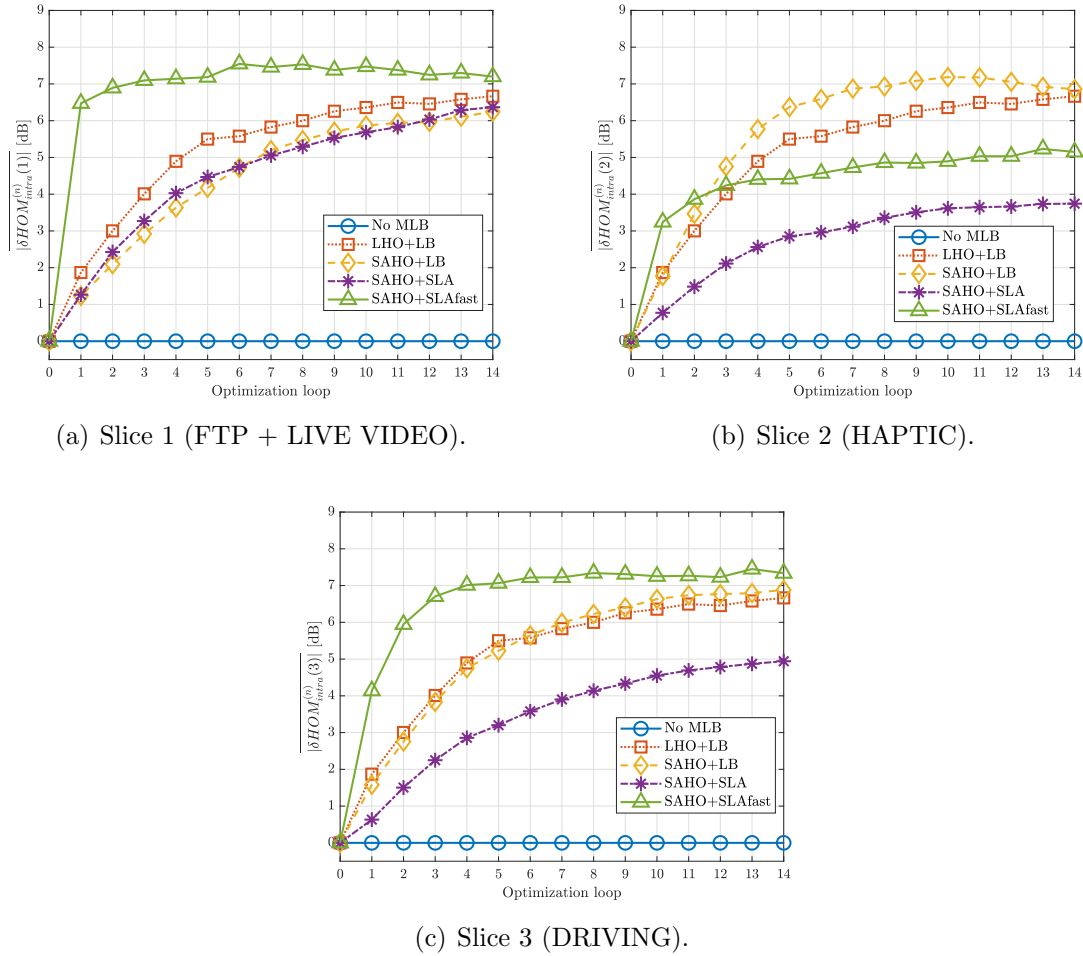


Figure 6.6: Evolution of absolute handover margin deviation from default values in tuned adjacencies per slice.

expected, SAHO+SLAfast shows the fastest convergence, since it tunes HOMs for all adjacencies in  $\mathcal{A}$  simultaneously every TI (i.e., 5 s). It should also be pointed out that, since LBO+LB relies on a slice-unaware HO scheme, red curves in Fig. 6.6.a) to c) are identical. For the remaining approaches, the evolution of HOM settings significantly differs per slice. This observation suggests that, at the beginning of the tuning process, performance (i.e., load for SAHO+LB, and SLA compliance for SAHO+SLA and SAHO+SLAfast) in neighbor cells varies per slice. This phenomenon may be due to: a) the different traffic distribution per slice, or b) a poor capacity broker performance for some slices in certain cells.

According to Fig. 6.6.a)–c), slice 1 presents the most similar final HOM settings across algorithms. In this slice, eMBB traffic requires a high PRB allocation per cell. Hence, slice 1 performance strongly impacts cell PRB utilization ratio. As a

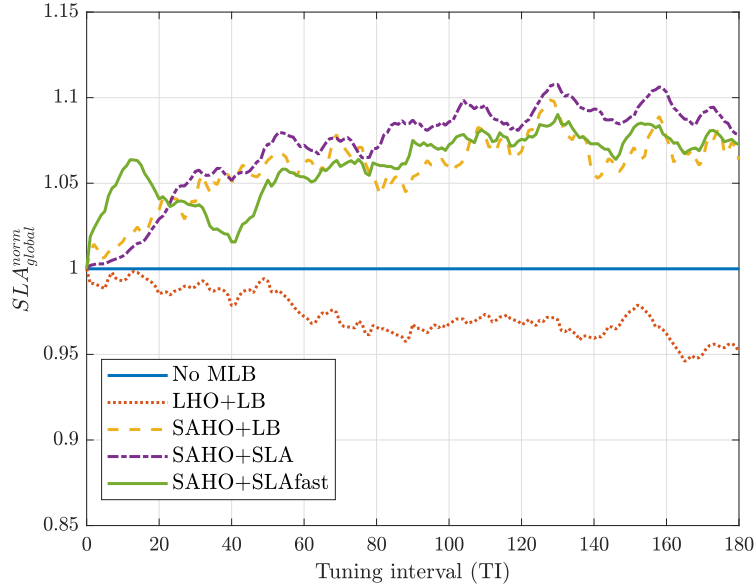


Figure 6.7: Evolution of the overall SLA compliance in the scenario.

consequence, LHO+LB may perform similarly to SAHO+LB. Moreover, since  $LR$  target for this slice is not too tight, the level of SLA compliance mainly depends on  $TH$  performance. As throughput is related to PRB utilization, load-based and SLA-based slice-aware approaches tend to tune HOMs in the same direction. Nonetheless, HOM set-up per strategy varies in many adjacencies, leading to different  $SLA_{FTP}$  and  $SLA_{LIVE\ VIDEO}$  FoMs, as will be shown later.

To illustrate the impact of MLB on network performance, Fig. 6.7 shows the evolution of  $SLA_{global}^{norm}$  for all the tested algorithms. It can be observed that, surprisingly, legacy LHO+LB algorithm presents the worst level of SLA compliance, even below the baseline case (i.e.,  $SLA_{global}^{norm} < 1$ ). In contrast to LHO+LB, all the remaining algorithms (i.e., SAHO+LB, SAHO+SLA and SAHO+SLAfast) outperform the baseline case in terms of  $SLA_{global}^{norm}$  across the whole tuning process (i.e., curves over 1 in Fig. 6.7). This behavior confirms the potential of slice-aware MLB schemes to improve SLA compliance in NS scenarios. It is remarkable that SAHO+SLA approach presents unstable  $SLA_{global}^{norm}$  evolution, with the best initial results due to fast HOM tuning followed by an undesirable strong performance degradation, compensated later.

For a deeper analysis, Table 6.5 summarizes the value of all the considered FoMs at the end of the tuning process (i.e., average FoM values in TIs belonging to the last optimization loop), computed globally and broken down per slice. SLA FoMs per slice reveal that LHO+LB has very poor performance in slice 3 (DRIVING), with a

Table 6.5: Performance comparison of MLB strategies in a NS scenario.

Slice	FoM	No MLB	LHO+LB	SAHO+LB	SAHO+SLA	SAHO+SLAfast
Global	$SLA_{global}$ [%]	66.75	63.54	70.43	72.61	71.65
	$\overline{PRB}_{util}$ [%]	56.25	59.71	62.07	62.28	63.71
	$nHO^{norm}$	1	3.32	3.39	2.41	3.82
Slice 1	$SLA_{FTP}$ [%]	39.16	45.41	50.95	57.37	59.07
	$SLA_{LIVE\ VIDEO}$ [%]	56.52	58.87	65.71	68.09	67.95
	Avg. $\overline{SLA}_{diff}(i, j, 1)$	0.39	0.36	0.29	0.25	0.25
	$\overline{\delta HOM}_{intra}^{(14)}(1)$ [dB]	0	6.67	6.24	6.37	7.20
Slice 2	$SLA_{HAPTIC}$ [%]	77.67	78.80	79.02	79.04	79.76
	Avg. $\overline{SLA}_{diff}(i, j, 2)$	0.29	0.29	0.26	0.25	0.25
	$\overline{\delta HOM}_{intra}^{(14)}(2)$ [dB]	0	6.67	6.86	3.74	5.15
Slice 3	$SLA_{DRIVING}$ [%]	74.84	62.60	73.33	76.18	73.02
	Avg. $\overline{SLA}_{diff}(i, j, 3)$	0.23	0.30	0.18	0.17	0.16
	$\overline{\delta HOM}_{intra}^{(14)}(3)$ [dB]	0	6.67	6.88	4.94	7.34

$SLA_{DRIVING}$  degradation of 12.24% in absolute terms compared to No MLB. Note that the low target  $TH$  for DRIVING users leads to a reduced PRB allocation per cell to slice 3, which therefore has a negligible impact on cell PRB utilization that drives the tuning process in LHO+LB. In contrast, for slices 1 (FTP + LIVE VIDEO) and 2 (HAPTIC), with a higher PRB allocation per cell, LHO+LB outperforms the baseline, with  $SLA_{FTP}$ ,  $SLA_{LIVE\ VIDEO}$  and  $SLA_{HAPTIC}$  higher than those of No MLB. Thus, it can be stated that balancing cell load in NS scenarios offloads congested cells only for slices accessing a significant number of PRBs. Even so, LHO+LB performance for slices 1 and 2 is still the worst among the tested MLB schemes.

Regarding slice-aware algorithms, PRB utilization values in Table 6.5 reveal that the improvement in  $SLA_{global}^{norm}$  shown in Fig. 6.7 comes along with a higher usage of radio resources due to the fact that traffic is offloaded from congested to underutilized cells. SAHO+SLA shows the best  $SLA_{global}$ , with a final improvement of 8.78% in relative terms compared to No MLB (i.e., 72.61% vs. 66.75%), followed by SAHO+SLAfast, with a  $SLA_{global}$  improvement of 7.34% compared to No MLB. These results prove that  $\overline{SLA}_{diff}(i, j, s)$  indicator is more powerful than PRB utilization ratio as a driver for MLB in sliced networks. Per-service SLA FoMs show that SAHO+LB is competitive to SAHO+SLA only for HAPTIC users served by slice 2, with  $SLA_{HAPTIC} \approx 79\%$ . Although the tested algorithms provide significantly different HOM settings for this slice (shown in  $|\overline{\delta HOM}_{intra}^{(14)}(2)|$  values), the high  $TH$  and  $LR$  requirements lead to moderate SLA improvements in all cases, with a maximum  $SLA_{HAPTIC}$  increase of 3%

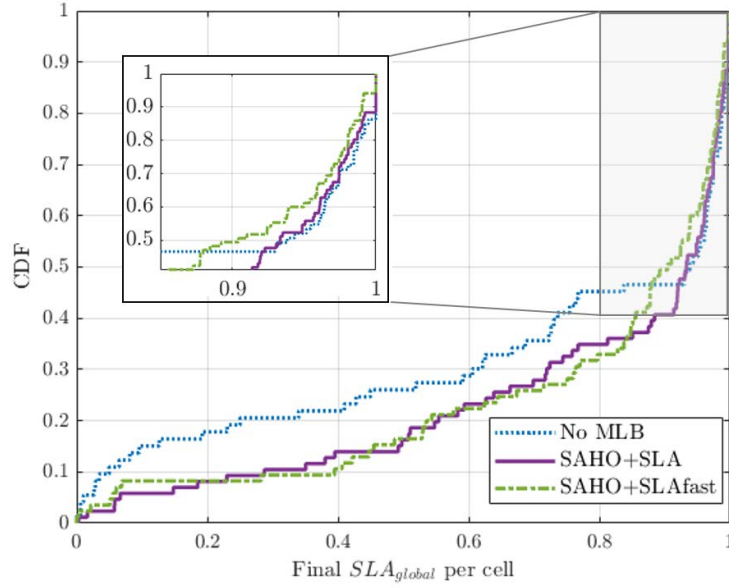


Figure 6.8: Cumulative distribution of final SLA compliance per cell for slice 1 (FTP + LIVE VIDEO).

in relative terms compared to No MLB.

To understand how SLA-driven algorithms obtain the above results, Fig. 6.8 shows the CDF of the final SLA compliance per cell in slice 1 for SAHO+SLA (solid line) and SAHO+SLAfast (dashed line), compared to No MLB (dotted line). Both MLB schemes show better SLA compliance in the worst cells at the expense of a slight performance degradation in the best cells. This behavior, also present in slices 2 and 3, is typical on self-tuning algorithms that balance a FoM across the scenario. In fact, according to  $\overline{SLA}_{diff}(i, j, s)$  figures in Table 6.5, SAHO+SLA and SAHO+SLAfast provide the best equilibrium of SLA compliance in neighbor cells, with a relative reduction of 35.9%, 10.4% and 26.1% in  $\overline{SLA}_{diff}(i, j, s)$  compared to No MLB for slices 1 to 3, respectively. Thus, balancing SLA compliance among cells on a per-adjacency-and-slice basis improves the overall system SLA compliance in NS scenarios.

When comparing SAHO+SLA and SAHO+SLAfast results in Table 6.5, it is observed that HOM deviations reached at the end of the tuning process are significantly different, even if both schemes have the same goal (i.e., equalizing SLA compliance per slice between neighbor cells). The highest variation appears in slice 3, with a difference of 2.4 dB in  $|\delta HOM_{intra}^{(14)}|(3)$  obtained with SAHO+SLAfast and SAHO+SLA. For a deeper analysis, Fig. 6.9 depicts the CDF of final absolute HOM deviation per adjacency in slice 3,  $|\delta HOM_{intra}^{(14)}|(i, j, 3)$ , in the tuned adjacencies for all the tested algorithms. It is observed that SAHO+SLA follows the most conservative tuning, leaving



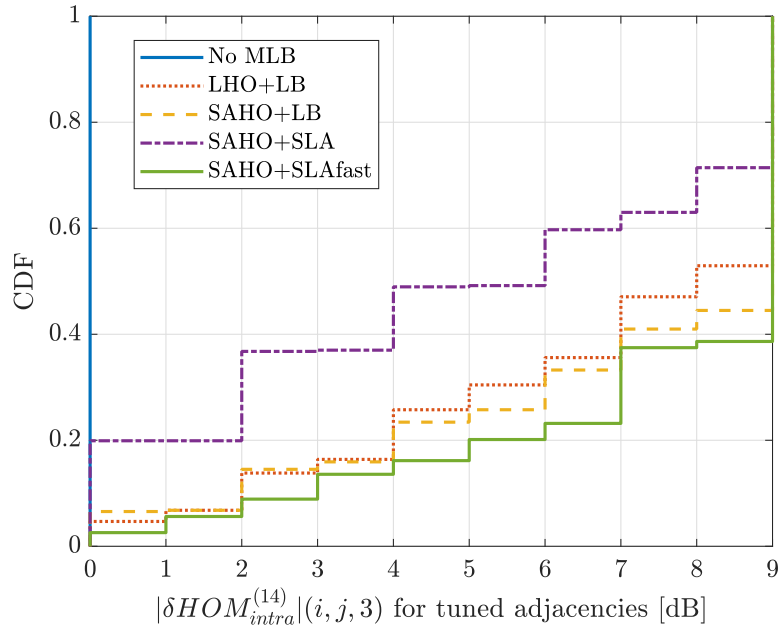


Figure 6.9: Cumulative distribution of final handover margin deviation from initial setting in tuned adjacencies for slice 3 (DRIVING).

20% of HOMs with the initial value. In contrast, SAHO+SLAfast performs the most aggressive HOM changes, with extreme HOM values in approximately 60% of adjacencies. According to Table 6.5, the conservative strategy followed by SAHO+SLA turns into the lowest increment in HOs triggered due to traffic steering, with  $nHO^{norm}=2.14$  (for all the remaining strategies,  $nHO^{norm}>3$ ). Thus, SAHO+SLA causes the lowest signaling overload and likelihood of dropped connections due to failures in the HO process.

The distinct HOM settings of SAHO+SLA and SAHO+SLAfast lead to a different final performance. SAHO+SLAfast only outperforms SAHO+SLA in more than 1% in absolute terms for FTP users, with  $SLA_{FTP}$  of 57.37% vs. 59.07% for SAHO+SLA and SAHO+SLAfast, respectively. Not shown in Table 6.5 is the fact that SAHO+SLAfast dramatically increases the number of HOs for slice 1 ( $nHO^{norm}=27.16$ ), which does not pay off. On the contrary, SAHO+SLA outperforms SAHO+SLAfast in slice 3, with  $SLA_{DRIVING}$  of 76.18% vs. 73.02% for SAHO+SLA and SAHO+SLAfast, respectively. More importantly, SAHO+SLAfast degrades performance for this slice compared to No MLB case. Actually, SAHO+SLA is the only strategy outperforming the baseline for slice 3. The poor LHO+LB performance, due to a reduced PRB allocation per cell, has been discussed above. This problem should be solved by SAHO+LB, taking into account slice-specific PRB utilization measurements. However, DRIVING users have

a bursty traffic profile consisting of small data chunks sent periodically that must be scheduled immediately. For a given cell-slice bandwidth, if data must be simultaneously transmitted to all DRIVING users,  $LR$  SLA may be violated even if PRB utilization remains low, since no data arrives to the transmission buffer until the next period. In contrast, if data bursts for DRIVING users in a cell must be transmitted at different time instants, average PRB utilization will be higher, but  $LR$  SLA is more likely to be complied. Hence, PRB utilization is not representative of SLA compliance for slices with low  $TH$  but stringent  $LR$  requirements. Finally, to understand the bad performance of SAHO+SLAfast in slice 3, note that DRIVING users move fast during long connections (unlike the other considered services). Due to the larger distance traveled, their radio conditions are subject to a wider range of variability. Consequently, for these users, aggressive cell area changes caused by the simultaneous modification of several HOMs in SAHO+SLAfast can lead to very poor radio conditions that temporarily prevent data transmission. For services with high latency and reliability requirements such as DRIVING, not transmitting a single packet strongly impacts the level of SLA compliance.

The above results confirm that the slice-aware MLB algorithm with adjacency clustering proposed in this work (SAHO+SLA) is the best option to enhance the level of SLA compliance while equalizing end-user satisfaction across the scenario and keeping a low increase in the number of HOs due to traffic steering.

#### d) Computational complexity

The proposed slice-aware MLB algorithm relies on a set of simple proportional controllers driven by an indicator computed from connection traces. Thus, discussion on computational complexity presented in section [6.2.3](#) also applies here. In this case, the worst-case time complexity of the algorithm, dominated by the computation of the indicator driving the tuning process,  $\overline{SLA}_{diff}(i, j, s)$ , is  $\mathcal{O}(N_u)$ , where  $N_u$  is the average number of active users.

### 6.3.4 Conclusions

In 5G and beyond systems with network slicing, new slice-aware self-optimization solutions are required to guarantee SLA compliance. In this work, a novel slice-aware MLB algorithm has been proposed. The algorithm adjusts intra-frequency handover margins on an adjacency-and-slice basis driven by an indicator reflecting the imbalance of SLA

compliance in neighbor cells per slice. In each adjacency and slice, an independent controller increments (or decrements) the value of handover margins based on that indicator, which can be computed by processing data in connection traces. To avoid ineffective actions and network instabilities, the algorithm operates only in the most relevant adjacencies per cell, and parameter tuning is performed simultaneously only in adjacencies comprising different cells. Moreover, to deal with the high dynamism of NS scenarios, MLB operates in a finer time resolution than in legacy MLB schemes.

Performance assessment has been carried out in a simulator emulating the activity of a realistic network with slices serving traffic from eMBB and uRLLC services. Results have shown the poor performance of slice-unaware MLB techniques in NS scenarios, specially for slices with busy traffic demanding a low data rate, which are neglected by legacy load balancing schemes. The proposed self-tuning algorithm has also outperformed a slice-aware load-driven MLB scheme, showing the potential of the proposed SLA-based indicator to drive the tuning process. Additionally, it has been proved that, even with the adequate driver indicator, tuning parameters too often (every 5 s) and in all adjacencies simultaneously dramatically increases the number of HOs, leading to signaling overload and possible dropped calls. In 15 minutes of network activity, the proposed algorithm has improved the overall SLA compliance by up to 8% compared to the case of not performing any MLB, while equalizing SLA compliance among neighbor cells. This improvement has been obtained with a significantly different final HOM set-up per slice.

It should be pointed out that the two traffic steering algorithms proposed in this chapter (QBHO+OE and SAHO+SLA) are conceived as centralized solutions to be run in the network management system, where connection traces from every cell in the network are collected. Likewise, recall that these algorithms are independent solutions designed for different RATs (QBHO+OE for 3G/4G, SAHO+SLA for 5G and beyond).



UNIVERSIDAD  
DE MÁLAGA

# Chapter 7

## Conclusions

This closing chapter summarizes the major findings of this thesis. Section [7.1](#) highlights the most relevant contributions. Then, section [7.2](#) outlines possible future research lines. Finally, section [7.3](#) provides a list of the publications arising from this work.

### 7.1 Main contributions

The high diversity, dynamism and complexity of upcoming cellular networks evince the need to develop advanced data-driven SON tools able to handle very different services and capture the peculiarities of each particular network. Moreover, in 5G, the new network slicing feature entails addressing new SON use cases (e.g., slice (re)dimensioning) and providing slice-aware solutions for legacy SON use cases. In this framework, this thesis has proposed data-driven solutions for two well-known SON use cases, namely RAN (re)dimensioning and MLB, relying on data gathered in the OSS.

Research has started with a thorough revision of literature in related topics. This initial stage has been essential for detecting research gaps and deciding how to formulate the problems to solve. First, the different types of machine learning algorithms have been presented to understand the existing alternatives for data-driven SON solutions. Then, the workflow of RAN (re)dimensioning and MLB procedures has been outlined to identify potential tasks to be enhanced and parameters to be optimized by using network data. Afterward, the types of data available in the OSS for this purpose have been explored. The knowledge of preprocessing complexity, available data and temporal/spatial granularity of different data sources is key for identifying the type of

data to be used in each SON solution. More importantly, the in-depth data analysis has confirmed that limited information regarding service type is available per user in current cellular networks. This is an issue for service-oriented SON solutions (including some models and algorithms developed in this thesis), assuming a prior knowledge of the application type demanded by the user. Finally, network slicing feature has been presented to identify its impact on the two tackled SON use cases.

Next, different analyses have been carried out. The main conclusions from these analyses, that can be transferred to other works (in some cases, even out of the telco scope) are:

- a) Radio connection traces are a powerful source of information for data-driven SON tools, and hence should be stored in the OSS. Although trace files are heavy and trace processing may be time-consuming, the investment pays off if such valuable information is used to empower several network management tools.
- b) Self-optimization solutions taking into account specific performance requirements per service are essential to guarantee end-user satisfaction in upcoming cellular networks offering extremely different services. Improving performance of services with stringent latency and reliability requirements arising in 5G is especially challenging due to the difficulty of estimating the impact of network parameter changes in latency.
- c) The definition of input features is key to make the most of clustering algorithms. The set of features must contain all relevant information while being orthogonal (i.e., two features must not provide the same information). The fewer features, the lower the possibility of suffering the curse of dimensionality. If dimensionality reduction is required, it should be considered that interpretable features derived with expert knowledge ease cluster interpretation compared to those obtained with feature extraction techniques (e.g., PCA). In addition, for imbalanced datasets, it is strongly recommended to divide datapoints into blocks with expert knowledge before performing clustering in order to prevent frequent data patterns from concealing less frequent patterns.
- d) When addressing time series forecasting with SL, the best prediction model (i.e., combination of SL algorithm, window observation and training strategy) is influenced by time series length, noise and seasonality strength. For short, noisy and non-seasonal series, simpler algorithms only capturing the general trend of time series tend to outperform complex recurrent ANNs such as LSTM.

- e) Although deep learning is powerful for regression, testing simpler SL algorithms is worth, since they can perform similarly or even better than DNNs while avoiding overfitting, specially for size-limited training datasets. No matter the selected algorithm, data preprocessing (i.e., feature normalization and outlier management) is critical to make the most of SL algorithms.

Experiments have always been carried out in realistic environment (in most cases, with data from commercial networks), which has entailed: a) a thorough read of vendor manuals to understand fields in CTRs and PM/CM files, and b) the development of parsing tools to preprocess raw data (e.g., to isolate connections from CTRs or calculate KPIs from PMs/CMs). The specific contributions and conclusions of each analysis are summarized next, broken down per topic.

### 7.1.1 Classification of encrypted traffic in cellular networks

The problem of classifying connections per service type in mobile networks has been addressed first. The idea of performing classification over information from radio traces (i.e., CTRs) has been explored for the first time. As part of problem formulation, an analysis of traffic captured from a mobile terminal connected to a commercial LTE network when demanding different live applications has been presented. This preliminary analysis has helped to establish theoretical bounds for some traffic descriptors in CTRs from full-buffer services. The dependence of other traffic descriptors on network conditions has also been pointed out.

Then, a novel scheme for coarse-grained traffic classification has been proposed. The method relies on agglomerative hierarchical clustering. Thus, it can be used in the absence of labeled data, seldom available in commercial mobile networks. To avoid the influence of network conditions, a new set of network-independent features characterizing connections at burst level has been analytically derived from information in CTRs. To circumvent the limitations of distance-based clustering algorithms to handle imbalanced datasets, broad connection blocks have been created based on expert knowledge before performing clustering. Validation has been carried out with a trace dataset from a live LTE network. Results have shown that the classification performed by the proposed method is consistent with the traffic share reported for live networks the year data was collected, confirming the potential of combining expert knowledge and USL over burst-level traffic descriptors to cluster connections per service type.

## 7.1.2 Supervised learning for radio access network (re)dimensioning

Next, work has focused on creating SL models for (re)dimensioning purposes. Two key tasks in this process have been covered: estimating radio throughput indicators at cell/slice level and forecasting cell traffic in the long term. Previous contributions on radio throughput estimation through SL consider either simple MLR models or complex DNNs. Likewise, long-term cellular traffic forecasting had not been tackled yet via SL due to the need for an extensive dataset comprising measurements collected for years. This thesis has compared the performance of well-known SL algorithms based on distance, vectors, decision trees and ANNs for these tasks.

### a) Throughput estimation in cellular radio access networks

The estimation of radio throughput indicators from data gathered in the OSS has been formulated as a regression problem. HSDPA and LTE networks have been first considered. In each RAT, a different set of input features built from CMs and PMs has been defined to estimate the aggregated cell throughput in the DL (DL cell throughput) and the average user throughput per cell in the DL (DL user throughput) in high load scenarios. Six well-known SL algorithms have been compared, namely MLR, SVR, KNN, RF, a shallow MLP and a deep MLP. Feature selection has been performed separately with a correlation-based method or with wrapper methods.

Assessment has been carried out over two datasets from a live HSDPA network and a live LTE network, respectively. Results have shown that wrapper feature selection methods outperform correlation-based schemes when finding the optimal subset of input features (determining data to be collected in the OSS). In both RATs, non-linear SL algorithms have outperformed classical MLR, especially when estimating DL user throughput. More importantly, some non-deep algorithms (e.g., shallow MLP) have shown similar performance to DNNs with fewer input features while being faster to train and less prone to overfitting. The best algorithms (i.e., shallow MLP for HSDPA and KNN for LTE) have shown error metrics lower than 10% with six input features at most.

Then, the analysis has been extended to network slicing scenarios arising in 5G. In the absence of large-scale datasets from operational networks with NS, assessment has been carried out over simulated data. For this purpose, network slicing has been



implemented in an existing simulation tool that emulates a realistic LTE-Advanced cellular network. The same set of simulations has been run in three scenarios: a non-sliced scenario and two NS scenarios with single-service and multi-service slices. As a result, three cell-level datasets (one per scenario) and two slice-level datasets (one per NS scenario) have been built. Datasets comprise input features computed from CTRs and CMs/PMs collected on a per cell and cell-slice basis. The target indicator is DL cell throughput for cell-level datasets and aggregated throughput per cell and slice (DL slice throughput) for slice-level datasets. In the light of results from the previous analysis, MLR algorithm and correlation-based feature selection have been omitted, and XGBoost and AdaBoost algorithms have been included.

A preliminary analysis of cell-level datasets has revealed significant differences in the correlation among some input features and DL cell throughput in each scenario, justifying the need for a separate analysis per scenario. The results have shown that, with adequate feature selection, all the tested algorithms achieve a similar and acceptable performance (i.e., error lower than 10%) when estimating DL cell throughput in the two tested NS scenarios. Models showing the best trade-off between accuracy and complexity are those based on a shallow MLP with four input features related to bandwidth, radio resource utilization and spectral efficiency. Such features can be computed from cell-level PMs/CMs. In contrast, only ensemble methods based on DTs and ANNs have reached acceptable accuracy when estimating DL slice throughput. As expected, model performance is worse in multi-service slices offering a mix of applications. It is also remarkable that, in both network slicing scenarios, the five input features to best models not only include features computed from cell-level PMs/CMs, but also features computed from slice-level PMs/CMs and information about service mix per slice derived from CTRs.

After this analysis, it can be concluded that non-deep SL techniques estimate radio throughput metrics in different RATs and scenarios with adequate performance. Moreover, the fact that slice-level models require information regarding service mix confirms the need for the traffic classification method previously developed.

## **b) Long-term cell traffic forecasting**

The task of predicting cell traffic several months in advance has been formulated as a time series problem. A preliminary analysis comparing the autocorrelation of hourly and monthly busy-hour cell traffic in two live LTE networks has revealed the challenges

of long-term traffic forecasting, based on short and noisy time series.

Then, a comparative study has been carried out assessing the performance of several SL algorithms not tested for this purpose so far against classical TSA approaches. To this end, three experiments have been carried out over a unique dataset comprising traffic measurements collected for two and a half years in a live LTE network covering an entire country.

Results have shown that SL algorithms outperform classical TSA approaches in terms of accuracy and required storage capacity. It has also been concluded that specific models must be developed for high-traffic cells, where prediction accuracy is critical. Unexpectedly, RF and a shallow MLP have shown the best results, with similar performance to DNNs based on LSTM units designed to model time dependencies. These results confirm the limited predictability of monthly busy-hour traffic compared to hourly or daily traffic series. None of the considered algorithms is extremely accurate, especially for summer months with holidays in the country where the network operates. Subsequent research (not shown here) has revealed that per-series outlier management combined with time series smoothing or additive decomposition can improve forecasts significantly.

### 7.1.3 Traffic steering in cellular networks

Finally, the task of designing service-oriented MLB algorithms driven by network data has been addressed. An in-depth review of related literature has pointed out the lack of MLB strategies to handle inter-frequency traffic steering QoE, and the absence of slice-aware MLB algorithm considering SLA aspects. These research gaps have been covered in this thesis. In both cases, MLB has been formulated as a control problem. Moreover, unlike legacy approaches driven by indicators computed from cell-level counters, the solutions proposed here rely on novel indicators reflecting individual user performance, which are computed from radio connection traces.

First, a novel strategy for steering traffic in multi-tier LTE networks to improve the overall system QoE has been proposed. For this purpose, RSRQ-based inter-frequency HOs are first enabled, and inter-frequency HOMs are tuned later per adjacency after each ROP with a novel QoE-based data-driven MLB algorithm. In each adjacency, the tuning process is carried out by a proportional heuristic controller driven by a novel indicator assessing the average impact of HOs for all users in cells of an adjacency.

Performance assessment has been carried out with the above-mentioned simulator. For this purpose, a realistic two-tier scenario has been first implemented. The considered services are VoIP, video streaming, file download and web browsing. Experiments emulating different mobility scenarios have demonstrated that the proposed traffic steering strategy outperforms classical MLB techniques based on balancing PRB utilization among cells. Improvement is due to the offloading traffic from coverage layers to capacity layers, so that users make the most of the large bandwidth available at capacity layers.

Then, the analysis has been extended to NS scenarios. A novel slice-aware MLB algorithm to increase SLA compliance in sliced RANs has been proposed. For this purpose, the algorithm adjusts slice-specific intra-frequency HOMs per adjacency. The tuning process is driven by a novel indicator reflecting the imbalance of SLA compliance per slice in neighbor cells. In each adjacency and slice, an independent controller increments (or decrements) the value of HO margins based on that indicator. To avoid ineffective actions and network instabilities, the algorithm operates only in a subset of relevant adjacencies, clustered into groups of adjacencies with disjoint cells. Then, parameter tuning is performed in a different adjacency group every 5 seconds.

Performance assessment has been carried out by simulating the activity of a realistic NS scenario with three slices serving eMBB and uRLLC traffic from users with different mobility patterns. For this purpose, traffic models for live video streaming, haptic communications and autonomous driving services have been implemented. SLA per slice has been defined as target session throughput and latency-reliability commitment. Results have shown the poor performance of legacy (i.e., slice-unaware) MLB in NS scenarios, specially for those slices with low radio resource allocation. The proposed algorithm has outperformed other slice-aware traffic steering strategies not driven by SLA or not performing adjacency clustering, converging in only 15 min. Improvement is obtained by balancing SLA compliance across the network.

#### 7.1.4 Discussion on model implementation

When introducing the data-driven approach, the trade-off between a centralized and distributed SON architecture becomes extremely important. All models and algorithms developed within this thesis are conceived to be used in centralized solutions running in the OSS, where data from all cells and slices is gathered. Such a centralized MANO approach eases the detection of network-wide issues. Moreover, orchestrating the be-

havior of radio network equipment across an entire network is more robust against instabilities caused by the concurrent operation of several SON functions with conflicting objectives. Nonetheless, all solutions proposed here can be implemented in a distributed MANO if an interface among base stations exists. For instance, the traffic classification scheme can be applied per cell. Likewise, the driver indicator of the proposed inter-frequency MLB algorithm can be easily computed if base stations exchange data. Finally, those proposals requiring supervised training could be trained with federated learning [255].

## 7.2 Future work

Several work lines arise from this thesis that could be explored in the future. The first interesting research direction is reproducing all the experiments conducted here in other scenarios (preferably in live networks). The aim is to check the capacity of proposed algorithms and methodologies to be generalized to RANs with different characteristics (e.g., topology or RRM algorithms). This work line entails some difficulties, such as the reluctance of operators to a) tune radio parameters network-wide to test MLB algorithms and b) share network data required for the remaining analyses (if available, which is not always the case, especially for traffic forecasting experiments requiring data collected for several years). Moreover, NS feature is not enabled yet in any commercial 5G systems. Thus, by now, NS experiments can only be reproduced via simulation.

Some brief ideas and guidelines on other open research lines for each problem tackled in this thesis are presented next.

### a) Encrypted traffic classification in cellular RANs

The validation of the proposed encrypted traffic classification scheme has been performed over a LTE network. However, the method can easily be extended to other RATs and is especially suitable for 5G networks, where highly differing services co-exist and thus the development of service-oriented NFs is key to warrant customer satisfaction.

In NR, similarly to LTE, traffic classification must start by dividing connections per QoS identifier (i.e., 5QI). Then, the proposed method must be applied over connections with 5QIs comprising different service types. Two aspects must be considered to adapt the method to NR: a)  $\eta_{UL}$  threshold set to split connections with high volume into two

groups must be recomputed considering headers of the NR stack protocol in the user plane (e.g., the new service data adaptation protocol layer), and b) 5G numerology must be taken into account when computing burst-level indicators. For instance, for numerology schemes  $\mu$  where TTIs last less than 1 ms (i.e.,  $\mu > 0$ ),  $T_{DL}^{active}(k) = N_{DL}^{burst}(k) \overline{N_{burst-DL}^{activeTTI}}(k) T_{TTI}$  in (3.5), where  $T_{TTI}$  is TTI duration expressed in ms. By considering these aspects, the classification method should perform well for connections with 5QIs from 6 to 9, analog to those QCI in LTE. It must also be checked if the same traffic descriptors suffice to classify connections with other 5QIs comprising a mix of services (e.g., 5QI 3, with a mix of real-time gaming and V2X traffic, among other services) or if new features must be computed.

### b) Throughput estimation in cellular RANs

This thesis has tackled the estimation of DL radio throughput indicators in the DL, which may suffice for legacy mobile networks. However, for effective redimensioning in NR, the analysis must be extended to the UL, which strongly influences some new 5G services (e.g., live video upload or sensor networks). Moreover, uRLLC and mMTC services should be included to assess the impact on throughput modeling, more importantly, model other performance metrics (e.g., latency-reliability for uRLLC). Note that emulating uRLLC traffic accurately implies analyzing network activity with a 1-ms time resolution. Likewise, mMTC services are characterized by an extremely large number of devices connected simultaneously to the network. These requirements would increase simulation time by more than 10 times, making it unfeasible to run the set of simulations required to get a significant amount of data. Thus, such an analysis must be performed over real data from commercial 5G networks when available. Then, more complex models based on DNN may be tested, which would have overfitted with the size of the datasets simulated here. Additionally, we will consider in the future the use of transfer learning [256] to leverage pretrained models derived for slices with different RRM algorithms (e.g., slices managed by virtual MNOs with different packet schedulers) or for different networks. Likewise, the use of multitask ANNs [257] will be explored to jointly estimate the performance of multiple slices in a cell and its neighbors.

### c) Long-term traffic forecasting

Results in this thesis prove the need for further research on long-term traffic prediction. Apart from enhancing data preprocessing as commented in section 5.5, forecasting models can be extended to account for events that drastically change traffic patterns in a cell, e.g., new neighbor site, equipment upgrades, social events, etc. This approach has been explored for cell hourly traffic time series [258]. Other potentially beneficial approaches are using graph ANNs to simultaneously consider the evolution of traffic in a cell and its neighbors, or creating multi-variable models to predict the evolution of several KPIs in a cell simultaneously. All these extensions are crucial for NGNs, where the coexistence of several services, slice (de-)activation, multi-connectivity and other features will lead to extremely complex traffic patterns, and hence a deep knowledge of the radio and social environment cells will be key to achieve accurate traffic forecasts. Nonetheless, note that all these approaches imply using complex ANNs with thousands of internal parameters, thus requiring large datasets to be trained. As a consequence, it is strongly recommended that operators store data with finer time resolution (e.g., daily busy-hour measurements) in the long term to make the most of SL models for traffic forecasting.

### d) Traffic steering in cellular networks

QoE-based and SLA-based traffic steering strategies presented in this thesis can be extended to handle inter-RAT mobility in legacy cellular networks and sliced NGNs, respectively. The most challenging task is finding the most suitable HO scheme (i.e., HO triggering event and report measurement). Once identified, applying the proposed MLB algorithms should be straightforward. Moreover, the idea of driving parameter tuning with indicators derived from connection traces, reflecting individual user performance, could be extended to create algorithms for optimizing parameters driving carrier aggregation or multi-connectivity schemes in NGNs. For the QoE-driven model, no matter the use case, the first step is developing models to compute QoE experienced for new services.

Another promising research line is designing QoE/SLA-driven MLB algorithms relying on DRL. Solutions proposed here improve a certain FoM that is maximized by reaching an equilibrium point (i.e., a value of 0 for the imbalance indicator). However, the optimal HO set up from a QoE/SLA perspective may be different for some adjacencies, which can be learned by a DRL agent. Moreover, unlike fixed heuristic controllers,

models in DRL agents can be retrained to adapt to changes in the network affecting slice performance (e.g., activation of a new slice or change in capacity broker...), leading to a faster (and maybe better) convergence. In NS scenarios, a promising option is using a collaborative multi-agent approach, as done in [177] for the capacity broker, capturing slice peculiarities and inter-slice performance relationships. The use of federated learning to train these models will avoid: a) overload in the backhaul due to data exchange with the OSS, and b) privacy issues preventing operators from accessing slice-level data [259].

### 7.3 List of contributions

The following publications have arisen from this thesis:

#### Journal publications

- [I] C. Gijón, M. Toril, S. Luna, M.L. Marí, “A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks”, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414-9424, Oct. 2019.
- [II] C. Gijón, M. Toril, M. Solera, S. Luna, L. Jiménez, “Encrypted traffic classification based on unsupervised learning in cellular radio access networks”, *IEEE Access*, vol. 8, pp. 167252-167263, Sep. 2020.
- [III] C. Gijón, M. Toril, S. Luna, J.L. Bejarano, M.L. Marí, “Estimating pole capacity from radio network performance statistics by supervised learning”, *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2090-2101, Dec. 2020.
- [IV] C. Gijón, M. Toril, S. Luna, M.L. Marí, J.M. Ruiz, “Long-term data traffic forecasting for network dimensioning in LTE with short time series”, *Electronics*, vol. 10, p. 1151, May 2021.
- [V] C. Gijón, M. Toril, S. Luna, “Data-driven estimation of throughput performance in sliced radio access networks via supervised learning”, *IEEE Transactions on Network and Service Management*, accepted in Sep. 2022.
- [VI] C. Gijón, T. Mahmoodi, S. Luna, M. Toril, “Data-driven Slice-Aware Traffic Steering for Service Level Agreement Compliance in Sliced Radio Access Networks”, submitted for publication.

### Contributions to conferences and scientific workshops

- [VII] **C. Gijón**, M. Toril, S. Luna, M. L. Marí, “A data-driven user steering algorithm for optimizing user experience in multi-tier LTE networks”, *9th MC and scientific meeting of COST CA15104 (IRACON)*, Dublin (Ireland), Jan. 2019.
- [VIII] **C. Gijón**, M. Toril, S. Luna, M.L. Marí, “Mejora de la calidad de experiencia en redes LTE multi-portadora”, *XXXIV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2019)*, Sevilla (Spain), Sep. 2019.
- [IX] **C. Gijón**, M. Toril, S. Luna, J. L. Bejarano, M. L. Marí, “Estimación de la capacidad en redes LTE mediante aprendizaje supervisado”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (Spain), Sep. 2020.
- [X] **C. Gijón**, M. Toril, S. Luna, “Modelling performance in sliced radio access networks with supervised learning”, *1th scientific meeting of COST CA20120 (INTERACT)*, Bologna (Italy), Feb. 2022.
- [XI] **C. Gijón**, M. Toril, S. Luna, “Modelado de rendimiento de segmento en redes de acceso radio mediante aprendizaje supervisado”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.

[II] presents the encrypted traffic classification strategy described in chapter [3](#). The analysis of the estimation of radio throughput indicators explained in chapter [4](#) is described in [III, V, IX–XI]. [III, IX] focus on legacy (i.e., non-sliced) networks. LTE technology is first considered in [IX], and HSDPA is then added in [III]. Likewise, [V, X, XI] cover throughput estimation in RAN sliced networks at cell level [V, X, XI] and slice level [X, XI]. [IV] describes experiments with long-term cell traffic forecasting presented in chapter [5](#). Finally, the traffic steering strategy for multi-tier networks introduced in chapter [6](#) is proposed in [I, VII, VIII], whereas the MLB algorithm for sliced RANs proposed in that chapter is described in [VI].

This thesis has been funded by the Spanish Ministry of Education, Culture and Sports (grant FPU17/04286). All the publications above have been developed in the framework of several research projects:

- TEC2015-69982-R project (*Planning and optimization methods for B4G networks*), funded by the Spanish Ministry of Economy and Competitiveness [I,



VII, VIII].

- ICT-760809 project (*ONE5G: E2E-aware Optimizations and advancements for the Network Edge of 5G New Radio*), funded by Horizon 2020 [I, VII, VIII].
- RTI2018-099148-B-I00 project (*Automatic planning methods for virtualized 5G networks*), funded by the Spanish Ministry of Science, Innovation and Universities [II, III, V, IX–XI].
- UMA18-FEDERJA-256 project (*QoE prediction in 5G networks*), funded by Junta de Andalucía [IV].
- 8.06/5.59.5705 -2 IDEA contract with Ericsson Spain (*Development of use cases for designing, optimizing and dimensioning mobile networks*), partially funded by Junta de Andalucía (Agencia IDEA, Consejería de Ciencia, Innovación y Empresa) [IV, V, X, XI].

Likewise, some contributions of this thesis have been part of European COST actions CA15104 (Inclusive Radio Communications, IRACON) [VII] and CA20120 (Intelligence-Enabling Radio Communications for Seamless Inclusive Interactions, INTERACT) [X].

The author of this thesis has also co-authored the following publications related to the topics covered here during the time of completing her doctoral studies:

### Related journal publications

- [XII] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “A QoE-driven traffic steering algorithm for LTE networks”, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11271-11282, Nov. 2019.
- [XIII] P. Sánchez, S. Luna, M. Toril, **C. Gijón**, J.L. Bejarano, “A data-driven scheduler performance model for QoE assessment in a LTE radio network planning tool”, *Computer Networks*, vol. 173, p. 107228, May 2020.
- [XIV] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “A self-tuning algorithm for optimal QoE-driven traffic steering in LTE”, *IEEE Access*, vol. 8, pp. 156707-156717, Aug. 2020.
- [XV] A. García, **C. Gijón**, M. Toril, S. Luna, “Data-driven construction of user utility functions from radio connection traces in LTE”, *Electronics*, vol. 10, p. 829, Mar. 2021.
- [XVI] J. L. Bejarano, M. Toril, M. Fernández, **C. Gijón**, S. Luna, “A deep-learning

model for estimating the impact of social events on traffic demand on a cell basis”, *IEEE Access*, vol. 9, pp. 71673-71686, May 2021.

- [XVII] L. Jiménez, M. Solera, M. Toril, **C. Gijón**, P. Casas, “Content matters: clustering web pages for qoe analysis with webCLUST”, *IEEE Access*, vol. 9, pp. 123873-123888, Aug. 2021.
- [XVIII] M.L. Marí, S. Mwanje, S. Luna, M. Toril, H. Sanneck, **C. Gijón**, “A service-centric Q-learning algorithm for mobility robustness optimization in LTE”, *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3541-3555, Sep. 2021.
- [XIX] J. Sánchez, M. Toril, V. Wille, **C. Gijón**, M. Fernández, “On the improvement of cellular coverage maps by filtering MDT measurements”, *IEEE Transactions on Mobile Computing*, Jan. 2022.

#### Related contributions to conferences and scientific workshops

- [XX] M. L. Marí, S. Luna, M. Toril, **C. Gijón**, “A QoE-driven traffic steering algorithm for LTE networks”, *9th MC and scientific meeting of COST CA15104 (IRACON)*, Dublin (Ireland), Jan. 2019.
- [XXI] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “Optimización de la calidad de experiencia en redes LTE mediante el reparto de tráfico”, *XXXIV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2019)*, Sevilla (Spain), Sep. 2019.
- [XXII] A. J. García, **C. Gijón**, M. Toril, S. Luna, “Data-driven construction of user utility functions from connection traces in LTE”, *12th MC Meeting and Final Workshop of COST CA15104 (IRACON)*, Louvain-la-Neuve (Belgium), Jan. 2020.
- [XXIII] M. L. Marí-Altozano, S. Mwanje, S. Luna, M. Toril, **C. Gijón**, “Una visión basada en QoE para algoritmo MRO en redes LTE”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (Spain), Sep. 2020.
- [XXIV] M. Toril, **C. Gijón**, S. Luna, M. Fernández, “Asignación de unidades de banda base en redes de acceso radio centralizadas por teoría de grafos”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (Spain), Sep. 2020.
- [XXV] J. M. Sánchez, M. Toril, **C. Gijón**, S. Luna, M. Fernández, “Análisis de capa-

- cidad de redes de acceso radio centralizadas en escenarios heterogéneos”, *XXXVI Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2021)*, Vigo (Spain), Sep. 2021.
- [XXVI] J. L. Bejarano, M. Toril, M. Fernández, **C. Gijón**, S. Luna, “Evaluación de modelos de deep-learning para series temporales de tráfico horario en redes celulares”, *XXXVI Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2021)*, Vigo (Spain), Sep. 2021.
- [XXVII] M. I. Quesada, S. Luna, M. Toril, **C. Gijón**, A. Durán, “Comparación de estrategias de entrenamiento de modelos de predicción de tráfico mensual en redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.
- [XXVIII] C. Cerezo, S. Luna, A. Durán, M. Toril, **C. Gijón**, “Gestión de valores anómalos en series temporales de redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.
- [XXIX] J. L. Bejarano, M. Toril, **C. Gijón**, S. Luna, A. Durán, “Obtención de intervalos de confianza en redes neuronales para predicción en redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.
- [XXX] N. González, M. Solera, F. Ruiz, **C. Gijón**, M. Toril, “Modelo de evaluación de calidad de experiencia para servicios de vídeo inmersivo por LTE basado en drones”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.
- [XXXI] J. M. Sánchez, M. Toril, **C. Gijón**, J. L. Bejarano, S. Luna, “Filtrado de trazas MDT de alta movilidad mediante aprendizaje supervisado”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (Spain), Sep. 2022.
- [XXXII] N. González, M. Solera, F. Ruiz, **C. Gijón**, M. Toril, “A quality of experience evaluation methodology for first-person-view drone control in cellular networks”, in *16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2022)*, Montreal (Canada), Oct. 2022.



UNIVERSIDAD  
DE MÁLAGA

# Appendix A

## Simulation tool

This appendix describes the simulation tool used in this thesis. It consists of a dynamic system-level simulator that emulates the activity of the DL in a LTE-Advanced RAN with NS feature. The tool is implemented in Matlab due to its efficiency in operating with large matrices. In this appendix, the general structure of the simulator is first described. Then, the most relevant procedures in physical, link and network layers are explained. Finally, NS implementation is presented.

### A.1 General structure

This section introduces the simulation tool. For clarity, its workflow is first outlined. Next, the different scenarios considered within this thesis are described. Then, UE model is detailed. Finally, the implemented QoE models are presented.

#### A.1.1 Work flow

Fig. [A.1](#) illustrates the simulator workflow. A simulation starts by loading a presaved configuration file that acts as an interface allowing the user to run simulations without a deep knowledge of the internal code. A wide range of parameters can be tuned regarding simulation time (e.g., duration, time resolution...), scenario (e.g., site location, number of cells per site...), base station parameters (i.e., carrier, bandwidth, transmit power...), traffic (e.g., UE density and spatial distribution, service mix per cell and slice, UE speed...), network procedures (e.g., HO parameters, packet scheduling algorithm, target BLER per service...) and NS set-up (e.g., number of slices and service

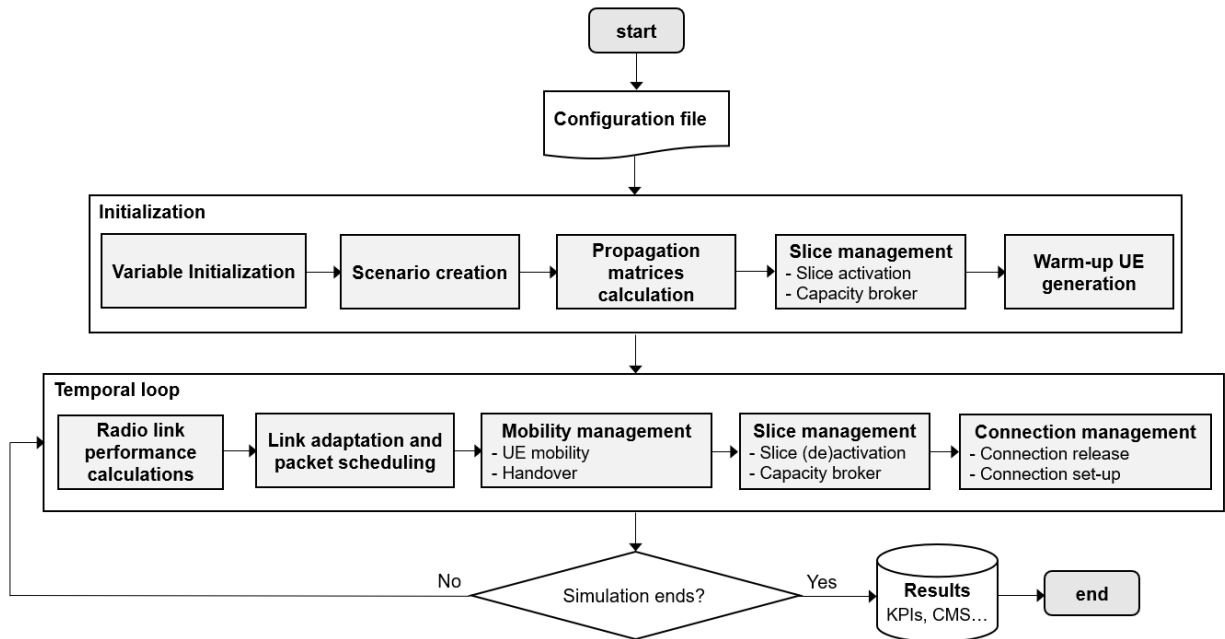


Figure A.1: Simulator workflow.

time, capacity broker...). Table [A.1](#) summarizes the simulation set-up considered in this thesis, detailed in subsequent sections of this appendix.

Once the configuration file is loaded, the initialization stage begins, where: a) simulation variables are initialized, b) the scenario is generated, c) radio resources are split into active slices, and d) several UEs (a.k.a. *warm-up* UEs) are created following pre-established traffic conditions per slice. These UEs emulate users already connected to the network before the simulation starts and aim to ensure that traffic conditions are stable, thus guaranteeing the reliability of results from the very beginning of the simulated time. Then, for computational efficiency, matrices containing radio link performance at each point of the scenario are computed and saved. These matrices consider path loss, antenna gains, slow fading and fast fading effects. During the simulation, they are used for cell allocation and to compute the SINR experienced by UEs in the scenario.

Next, a temporal loop is executed. The number of iterations depends on the configured simulation time and step resolution (e.g., 360,000 iterations to emulate 1 h of network activity with a resolution of 10 ms). In each iteration, radio link performance computations (i.e., RSRP, RSRQ and SINR per UE) are first performed. Then, RRM procedures are executed. At link layer, packet scheduling is carried out with link adaptation based on CQI. At network layer, inter-frequency and intra-frequency HO mechanisms are implemented. Regarding NS management, active slices whose service

Table A.1: Main simulation parameters.

Parameter	Description
Time resolution	10 ms
Transmission mode	Frequency division duplexing
5G numerology ( $\mu$ )	0
Propagation model	Path loss: Hata, COST-231 [260] Slow fading: log-normal $\sigma_{SF} = 8$ dB, $d_c = 50$ m Fast fading: ETU model [261]
Base station model	Tri-sectorized antennas, MIMO 2x2, transmit power from real base stations ([47.8-49] dBm), no beamforming
Packet scheduler	Classical exponential/proportional fair [262]
Link adaptation	CQI-based, MCS selected to guarantee a target BLER defined per service
Traffic model	Non-uniform spatial UE distribution and traffic mix Services: VoIP, progressive video streaming, file download, web browsing, live video streaming, haptic communications, autonomous driving
UE mobility	Constant speed at 0 km/s (static), 3 km/s (pedestrian) or 50 km/h (car) and constant random direction
HO set-up	Event A3: $HOM(i, j)=3$ dB and $TTT(i, j)=256$ ms Event A5 (RSRP-driven): $thd(i)=-115$ dBm, $thd(j)=-108$ dBm
NS implementation	Slicing at packet scheduling level [116], adaptive capacity broker, slice admission control based on PRB availability

time expires at that iteration are released. Likewise, new slice requests are attended by an admission control mechanism and, if accepted, new slices are conformed and activated. Slice activation and release imply a redistribution of spectrum among slices. Finally, finished connections (i.e., UE served by a recently-released slice, ended connection or dropped call) are released and new connections from previously existing or new slices are created.

When the temporal loop finishes, results (i.e., KPIs, KQIs, PMs...) are saved for further analysis.

### A.1.2 Simulation scenarios

The simulation tool allows emulating ideal scenarios with regular cell area or realistic scenarios with base station location and parameters from a live network. The latter option is selected in this thesis. Specifically, the two scenarios illustrated in Fig. A.2.a) and b) are considered for different contributions. Network A consists of 108 irregular

cells placed in urban and suburban areas covering  $11 \times 23$  km<sup>2</sup>. All cells work at 2.1 GHz. This scenario was already available in the simulation tool from previous works [33] [73]. Network B comprises 48 macro cells located in a dense urban area working at two different carriers: 736 MHz and 2100 MHz. Cells are distributed in 8 sites, each including two co-located sets (one per carrier) of tri-sectorized antennas. Thus, half of the cells work at each carrier. This scenario has been included in the simulator as part of this thesis to validate the MLB algorithm for multi-tier networks described in chapter 6.

As specified in LTE and NR standards for the considered frequency bands, the simulator operates in frequency-division duplexing mode [263]. Different bandwidths in [263] are set in different simulations with a subcarrier spacing of 15 kHz (i.e., 5G numerology with  $\mu = 0$ ).

### A.1.3 UE model

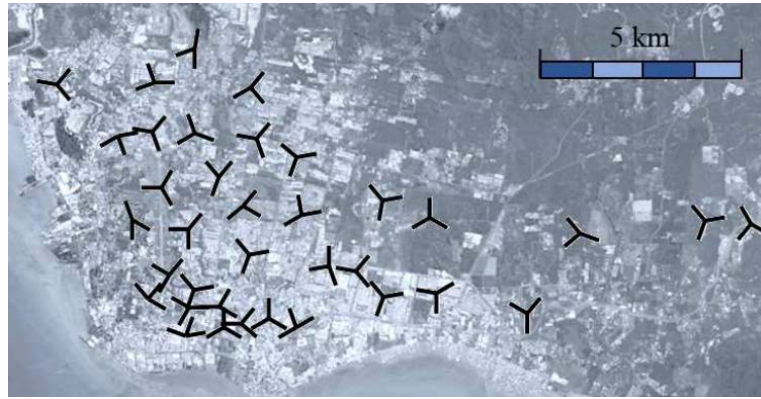
The simulated UEs can demand seven different services, namely VoIP (VoIP), progressive video streaming (VIDEO), file download via FTP protocol (FTP), web browsing via HTTP protocol (WEB), live video streaming (LIVE VIDEO), haptic communications (HAPTIC) and autonomous driving (DRIVING). Table A.2 breaks down the main service parameters regarding traffic model. It should be pointed out that the set of considered services comprises applications that are mainly delay-sensitive (e.g., VoIP), throughput sensitive (e.g., FTP), or both (e.g., HAPTIC), which will coexist in NGNs. Haptic and live video services can be demanded simultaneously by the same UE.

Both spatial UE distribution and service mix are configurable and can be either uniform or vary per cell and slice. During simulations, in a cell  $c$ , new UEs appear following a Poisson process with arrival rate  $\lambda(c)$ . UEs can be static (indoor users), pedestrians walking at 3 km/h or car travelers at 50 km/h. Those UEs in motion follow a straight trajectory with a random direction chosen at the beginning of the connection. The ratio of UEs at each speed can be set per service.

### A.1.4 QoE model

QoE is measured for VoIP, VIDEO, FTP and WEB sessions. For this purpose, utility functions are used to map objective QoS measurements into a MOS value, ranging from 1 (bad) to 5 (excellent). Likewise, context information is used to differentiate





(a) Network A (one-tier).



(b) Network B (two-tier).

Figure A.2: Scenarios implemented in the simulation tool.

indoor and outdoor users, so that indoor users are more demanding in terms of QoS. Thus, two utility functions are defined per service, as in [13].

The utility functions used for VoIP service are [266]

$$QoE^{(VoIP_{outdoor})} = 1 + 0.035R + 7 \cdot 10^{-6}(R - 60)(100 - R), \quad (A.1)$$

$$QoE^{(VoIP_{indoor})} = 1 + 0.035 \frac{R}{1.5} + 7 \cdot 10^{-6} \frac{R}{1.5} \left( \frac{R}{1.5} - 60 \right) \left( 100 - \frac{R}{1.5} \right), \quad (A.2)$$

where  $R$  is a parameter related to packet delay, ranging from 0 to 93. It is assumed that  $QoE^{(VoIP_{indoor})} = QoE^{(VoIP_{outdoor})} = 1$  if a VoIP connection is dropped.

Table A.2: Service model parameters.

Service	Description
VoIP	Coding rate: 16 kbps (a packet of 40 B every 20 ms) Call duration: exponential (avg. 60 s) Call dropped after 1 s without resources
VIDEO	Packet arrival process and file size from H.264/MPEG-4 AVC real trace with 720p resolution Call duration: uniform [30, 540] s Chunk size: 20 s of video content (initial burst), 5 s of video content (rest) Connection dropped when stalling lasts for twice the video duration
FTP	File size: log-normal (avg. [15, 85] MB)
WEB	No. of pages per session: log-normal (avg. 4) Page size: shifted log-normal (avg. 9.5 MB, min. 1.5 MB) Reading time: exponential (avg. 30 s)
LIVE VIDEO	Packet arrival process and file size from H.264/MPEG-4 AVC real trace with 720p resolution Call duration: uniform [30, 300] s Chunk size: 5 s of video content (initial burst), 2 s of video content (rest)
HAPTIC	Multi-point haptic traffic model in [264] Three components: globe, position tracker and actuators Packet size per component: fixed (min. 72 B, max. 442 B) Inter-packet time per component: Gaussian (avg. [10.87, 12.95] ms, std. dev. [1.98, 2.49] ms) Call duration: uniform [300, 600] s
DRIVING	Packet size=201 B Inter-packet arrival time of 100 ms (derived from lane merge use case data [265]) Call duration: uniform [300, 600] s

For VIDEO service, the utility functions are [267]

$$QoE^{(VIDEO_{outdoor})} = 4.23 - 0.0672T_{init} - 0.742F_{reb} - 0.106T_{reb}, \quad (A.3)$$

$$QoE^{(VIDEO_{indoor})} = 4.23 - 0.0672(1.5T_{init}) - 0.742(1.5F_{reb}) - 0.106(1.5T_{reb}), \quad (A.4)$$

where  $T_{init}$  is the initial buffering time in seconds,  $F_{reb}$  is the average stalling (a.k.a. re-buffering) frequency in seconds<sup>-1</sup> and  $T_{reb}$  is the average stalling duration in seconds.

For both indoor and outdoor users, the QoE value for a video connection is upper limited to 4.23, showing that some users do not score their experience as excellent even with the best possible link conditions. Again, a value of 1 is set if a connection is dropped.

The QoE of FTP users is computed as [268]

$$QoE^{(FTP_{outdoor})} = \max(1, \min(5, 6.5TH - 0.54)), \quad (A.5)$$

$$QoE^{(FTP_{indoor})} = \max(1, \min(5, 6.5\frac{TH}{1.5} - 0.54)), \quad (A.6)$$

where  $TH$  is the average session throughput in Mbps.

Finally, the utility functions used for WEB users are [268]

$$QoE^{(WEB_{outdoor})} = 5 - \frac{578}{1 + \left(\frac{TH+541.1}{45.98}\right)^2}, \quad (A.7)$$

$$QoE^{(WEB_{indoor})} = 5 - \frac{578}{1 + \left(\frac{\frac{TH}{1.5}+541.1}{45.98}\right)^2}, \quad (A.8)$$

where  $TH$  is the average session throughput in kbps.

## A.2 Physical layer

This section outlines the main aspects of the physical layer implementation of the simulation tool, namely propagation, noise and interference models.

### A.2.1 Propagation model

In this subsection, path loss, slow fading and fast fading models used in the simulator are first presented, and the process performed to compute radio link performance metrics per UE are outlined later.

### a) Path loss

Hata model and its extension COST 231 are used to compute path loss at 736 MHz ( $PL_{Hata}^{736}$ ) and 2100 MHz ( $PL_{COST231}^{2100}$ ), respectively [260]. The resulting path loss expressions considering an urban zone and a fixed UE height of 1.5 m are

$$PL_{Hata}^{736} \text{ [dB]} = 144.55 - 13.82 \log h_{BS} + (44.9 - 6.55 \log h_{BS}) \log d, \quad (\text{A.9a})$$

$$PL_{COST231}^{2100} \text{ [dB]} = 158.92 - 13.82 \log h_{BS} + (44.9 - 6.55 \log h_{BS}) \log d, \quad (\text{A.9b})$$

where  $d$  is the distance between base station and UE in km and  $h_{BS}$  denotes base station height in m. For the scenarios considered here,  $h_{BS}$  is extracted from the corresponding live networks, within the range [10, 64] m.

The resulting path loss is then refined by considering base station azimuth and a typical radiation pattern for tri-sectorized antennas.

### b) Slow fading

The effect of large buildings or geographical structures obstructing the line of sight on the radio signal is modeled as a log-normal distribution characterized by a standard deviation  $\sigma_{SF}$  whose value depends on the environment. For urban macro-cell scenarios as those considered in this thesis,  $\sigma_{SF}$  is typically set to 8 dB [269].

### c) Fast fading

The Extended Typical Urban (ETU) model is used for multi-path fast fading, whose power profiles are broken down in Table A.3 [261]. ETU model can be seen as a time-variant filter characterized by finite time response  $h(\tau, t)$ . A narrow-band Rayleigh model is first applied over each multi-path component and then a bi-dimensional Doppler filter is used as indicated in [1]. Next, Fourier transform is applied over the delay  $\tau$  to obtain the transfer function  $H(f, t)$ . Finally, to get UE position (and not time) as an independent variable (i.e.,  $H(f, d)$ ), the relation  $d = v \cdot t$  is applied. In this process, cell bandwidth is sampled with a 45 kHz resolution.

Table A.3: ETU model [1].

Delay ( $\tau$ ) [ $\mu\text{s}$ ]	Relative power [dB]
0	-1.0
50	-1.0
120	-1.0
200	0.0
230	0.0
500	0.0
1600	-3.0
2300	-5.0
5000	-7.0

#### d) Radio link performance calculations per UE

Radio link performance indicators (e.g., RSRP, SINR...) per UE are updated at every iteration to capture UE mobility and traffic fluctuations across time. For computational efficiency, calculations are based on three precomputed 3D matrices containing path loss plus antenna gains, slow fading and fast fading components across the scenario for each base station, respectively. For this purpose, the covered area is divided into a grid of points. For path loss and slow fading matrices, grid points have a resolution of  $50 \times 50$  m. In contrast, to capture every multi-path component, grid points in the fast fading matrix cover  $15 \times 15$  cm. In the latter case, for computational efficiency, a propagation matrix of  $48 \times 48$  m is created and repeated across both spatial dimensions until the whole scenario is covered.

During simulation, radio link performance metrics per UE are computed by combining the transmit power with all the above-mentioned matrices. The value of these matrices for the exact UE location are estimated by linearly interpolating values corresponding to the two nearest positions.

### A.2.2 Noise model

Noise level per PRB is computed as

$$N_{PRB} [\text{dBm}] = NSD [\text{dBm/Hz}] + NF_{UE} [\text{dB}] + 10 \log BW_{PRB} [\text{Hz}], \quad (\text{A.10})$$

where  $NSD$  is the noise power spectral density,  $NF_{UE}$  is the noise figure of UEs and

$BW_{PRB}$  is PRB bandwidth. In the simulation tool,  $NSD=-174$  dBm/Hz,  $NF_{UE}=9$  dB and  $BW_{PRB}=180$  kHz (i.e.,  $\mu=0$ ). As a result,  $N_{PRB}=-112.44$  dBm.

### A.2.3 Interference model

A total frequency reuse scheme is considered. Thus, the interference level received by a UE  $u$  in a given PRB  $p$  at iteration  $t$ ,  $I(u, p, t)$ , can be computed as the sum of the signal received from all cells transmitting data simultaneously in that PRB, i.e.,

$$I(u, p, t) = \sum_{j \neq i} P_{rx}(u, j, t) \cdot \overline{PRB_{util}}(j, t - 1), \quad (\text{A.11})$$

where  $P_{rx}(u, j, t)$  is the power level received by UE  $u$  from cell  $j$  at time  $t$  considering path loss and slow fading, and  $\overline{PRB_{util}}(j, t - 1)$  is the average PRB utilization of cell  $j$  in the last simulation iteration  $t - 1$ .

## A.3 Link layer

This section presents the main link level functionalities in the simulation tool, namely link adaptation, packet scheduling and retransmission scheme.

### A.3.1 Link adaptation

Link adaptation is in charge of selecting the most appropriate MCS to transmit data to a particular UE through the PDSCH channel according to link conditions. In the simulator, such a process is performed per UE and PRB in each iteration relying on CQI information sent by UEs in the UL.

The set of 16 MCSs in the 4-bit CQI table in [189] are considered. The effective SINR per PRB,  $SINR_{eff}$ , is first computed as detailed in [270]. Then, for computational efficiency, a link abstraction model built with the link-level Vienna simulator described in [271] is used to map  $SINR_{eff}$  to BLER on a certain MCS [272]. For each UE, the MCS allowing to transmit the highest number of bits while guaranteeing the target BLER for the demanded service is selected.

### A.3.2 Packet scheduling

Packet scheduling consists on allocating PRBs for the communication with UEs at every TTI. The simulator includes several packet schedulers, such as round robin, best channel, Proportional Fair (PF) or classical EXPonential/Proportional Fair (EXP/PF) [273]. The latter scheme is used in all the simulations within this thesis.

PF is a channel-aware packet scheduler that provides a trade-off between throughput performance and fairness. Classical EXP/PF is an extension of PF conceived to simultaneously handle real-time (RT) and Non-Real-Time (NRT) connections. It enhances the priority of RT flows, guaranteeing a bounded delay to RT packets while still maximizing system throughput and ensuring proportional fairness between UEs. For this purpose, the UE priority metric is separately calculated for RT and NRT users. For the former UEs, priority grows exponentially with delay, so that they have higher priority than NRT UEs when packet delay is reaching the dateline. This algorithm performs better than PF in high-load scenarios at the expense of higher computational complexity [262].

### A.3.3 Retransmission scheme

HARQ is a mechanism combining retransmissions and error correction. In the simulator, the maximum number of attempts to transmit a given block of data (i.e., set of bytes transmitted in a PRB) is defined per service, ranging from 1 (for uRLLC services with very stringent latency constraints) to 4. Since link adaptation guarantees a predefined target BLER per service, the probability of failure when transmitting a scheduled block of data is that target BLER. Such a process is modeled with a random variable. To emulate HARQ round-trip time, if the transmission fails and the maximum number of attempts has not been reached, the eNB tries to retransmit it in the next simulation iteration (i.e., after 10 ms).

## A.4 Network layer

This section outlines the network layer procedures in the simulator most relevant for this thesis, namely admission control and handover.

### A.4.1 Admission control

A UE  $u$  is accepted in the network if there is some cell  $c$  from which the received RSRP,  $RSRP_u(c)$ , exceeds a threshold  $RSRP_{min}(c)$  predefined on a cell basis, i.e.,

$$RSRP_u(c) \geq RSRP_{min}(c) . \quad (\text{A.12})$$

If several cells fulfill this condition, the UE camps in the cell with the best radio channel conditions (i.e., highest  $RSRP_u(c)$ ). Once the UE changes from idle to connected RRC mode, the condition in (A.12) is assessed again for both inter-frequency and intra-frequency neighbors of the camping cell. If all candidate serving cells work in the same carrier, the serving cell is that with the highest  $RSRP_u(c)$ . Otherwise, the serving cell is randomly chosen among the best candidates at all carriers. In such a process, the higher the cell bandwidth compared to other candidates, the higher the probability of being selected as serving cell.

### A.4.2 Handover scheme

Three different well-established HO schemes are used to handle inter-frequency and intra-frequency HO in the simulator:

1. *Power-BudGeT (PBGT) HO*: this type of HO is triggered when HO event A3 (i.e., neighbor cell is better than serving cell by a threshold) based on RSRP is fulfilled during a certain TTT (i.e., condition in (6.2)). The aim is to guarantee that UEs are always connected to the cell from which they receive the best signal. In SON, PBGT HOs are often used for load balancing among cells working at the same carrier.
2. *Quality-BudGeT (QBGT) HO*: this type of HO is analog to PBGT, HO but using RSRQ instead of RSRP (i.e., triggering condition in (6.5)). The aim is to guarantee that UEs are always connected to the cell with the best quality even if received signal is not the highest. This strategy can be beneficial for both intra-frequency and inter-frequency traffic steering.
3. *Level HO*: this type of HO is triggered by event A5 (i.e., serving cell is worse than a threshold and neighbor cell is better than another threshold) based on RSRP. This strategy is often used for inter-frequency traffic mobility.



For computational efficiency, in the simulator, HO events are assessed every 50 ms.

## A.5 Network slicing implementation

NS feature has been implemented in the simulation tool as part of this thesis. If enabled, different scenarios can be emulated with a configurable number of slices serving a single service (e.g., OTT service provider) or a set of services (e.g., eMBB slice). These slices can either remain active during the whole simulation or be activated/deactivated at a predefined time instant. This section outlines how the main procedures in slice life cycle management have been implemented.

As in [176], the SLA is defined in terms of capacity requirements for the expected traffic in a given area. In the preparation phase, a tenant applying for a slice  $i$  provides the infrastructure owner an individual slice template including: a) the slice operation area, b) the expected service spatial distribution and traffic intensity in peak periods, c) the required average session throughput per service  $s$  offered in the slice,  $TH_{APP}(i, s)$ , defined at application layer, and d) the required reliability in the slice for a target E2E latency.

In the planning phase, the capacity conformance NF uses an analytical model to determine the spectrum allocation required per cell to fulfill the SLA. The number of PRBs required by slice  $i$  in cell  $c$ ,  $N_{PRB}(c, i)$ , is the aggregation of PRBs required to schedule UEs demanding all services  $s$  offered in the slice,  $N_{serv}(i)$ ,

$$N_{PRB}(c, i) = \sum_{s=1}^{N_{serv}(i)} N_{PRB}(c, i, s). \quad (\text{A.13})$$

For a given service  $s$ ,  $N_{PRB}(c, i, s)$  can be estimated as

$$\widehat{N}_{PRB}(c, i, s) = N_{UE}(c, i, s) \widehat{N}_{PRB\_UE}(c, i, s), \quad (\text{A.14})$$

where  $N_{UE}(c, i, s)$  is the expected average number of simultaneous RRC connected UEs from slice  $i$  demanding service  $s$  in cell  $c$  and  $\widehat{N}_{PRB\_UE}(c, i, s)$  is the estimated number of PRBs required by each individual UE in cell  $c$  to fulfill service requirements in the SLA. The former term can readily be computed from traffic information in the SLA,

whereas the latter term can be estimated as

$$\widehat{N_{PRB_{UE}}}(c, i, s) = \frac{TH_{PHY}(i, s)}{\overline{TH_{PRB}}(c)}, \quad (\text{A.15})$$

where  $TH_{PHY}(i, s)$  is the throughput required at the physical layer to achieve the target  $TH_{APP}(i, s)$  and  $\overline{TH_{PRB}}(c)$  is the DL throughput per PRB experienced by a UE with average SINR in the service area of cell  $c$ .

For VoIP, HAPTIC and DRIVING services (i.e., small data chunks), headers are considered to compute  $TH_{PHY}(i, s)$ . In contrast, for the remaining considered services, header size is negligible compared to data chunk size, and it is therefore assumed that  $TH_{PHY}(i, s) \approx TH_{APP}(i, s)$ . To avoid underestimating required capacity in cells with low traffic, a minimum value of  $N_{PRB}(c, i) = 3$  is set for every cell and slice. If there are enough available resources in all cells of the slice operation area, the request is accepted and the slice is conformed and activated. Otherwise, the request is rejected.

In the operation phase, as in [177], the capacity broker NF periodically adjusts  $N_{PRB}(c, i)$  per cell and active slice by reassigning underutilized PRBs to slices whose capacity requirements have been underestimated. This process is repeated every 5 min until a steady state is reached. To make the most of spectrum capacity, the minimum slice chunk is reduced to 1 PRB.

The above-described spectrum sharing scheme is well aligned with previous proposals in the literature [116]. Intra-cell traffic isolation is ensured, since the packet scheduling function of a slice can only use PRBs assigned to that slice, thus preventing a high-load period in a slice from affecting other slices. However, since PRB assignment may differ in adjacent cells, inter-cell traffic isolation is not guaranteed. Nonetheless, spectrum splitting is performed by minimizing the probability of assigning a certain PRB  $p$  to different slices in neighbor cells to reduce inter-cell inter-slice interference.

# Appendix B

## Summary (Spanish)

Este apéndice presenta un resumen en español del trabajo realizado en esta tesis. En primer lugar, se describen los antecedentes que han motivado su realización. Durante el discurso, se expone el estado actual de la investigación y la tecnología, justificando la necesidad del estudio. A continuación se plantean los objetivos de la investigación y la metodología de trabajo seguida. Después, se resumen los resultados obtenidos en cada uno de los temas tratados. Por último, se identifican las principales contribuciones originales y se adjunta la lista de publicaciones asociadas a este trabajo.

### B.1 Antecedentes y motivación

Las redes de comunicaciones móviles han experimentado grandes cambios en los últimos años. En primer lugar, el crecimiento exponencial del tráfico de datos asociado a usuarios en movilidad ha obligado a los operadores a aumentar la capacidad de la red. Para ello, en la red de acceso radio (*Radio Access Networks*, RAN), las redes clásicas con una capa de macroceldas se están transformando en redes multi-portadora (es decir, con varias bandas de operación) y heterogéneas (p.ej., combinando celdas pequeñas y macroceldas) [2]. Además, los distintos requisitos de rendimiento de los servicios en movilidad y las altas expectativas de los usuarios han llevado a un cambio en los procedimientos de gestión de la red, que ya no se centran en el rendimiento de la red sino en la calidad de la experiencia del cliente (*Quality of Experience*, QoE) [3]. En paralelo, la llegada de la tecnología 5G está ampliando el modelo de negocio de los operadores, que proporcionarán servicios de banda ancha móvil mejorada, comunicaciones ultrafiabiles de baja latencia y comunicaciones masivas de tipo máquina a industrias

verticales [4]. El 3GPP lanzó las especificaciones del estándar *New Radio* (NR) en la versión 15. Para alcanzar los ambiciosos objetivos de rendimiento de 5G, en la RAN se introducen nuevos rangos de frecuencia (p.ej., bandas milimétricas) y funcionalidades (p.ej., multiconectividad o antenas con esquemas masivos de entrada múltiple y salida múltiple).

Como resultado de estos cambios, el tamaño y la complejidad de las redes móviles se han incrementado drásticamente, siendo evidente la necesidad de contar con herramientas de gestión automática con mínima intervención humana para garantizar un funcionamiento eficiente de la red. En la literatura, se han propuesto numerosas herramientas de configuración, optimización y curación automática para la RAN de las redes 2G (p.ej., [6] [7] [8]), 3G (p.ej., [9] [10] [11]) y 4G (p.ej., [12] [13] [14]). Las soluciones clásicas para estas redes auto-organizadas (*Self-Organizing Networks*, SON) se basan en modelos analíticos y/o controladores heurísticos derivados manualmente por expertos. Sin embargo, se prevé que este enfoque funcione mal en las redes celulares de nueva generación por varias razones. En primer lugar, la coexistencia de servicios con requisitos de calidad de servicio muy diferentes (p.ej., eficiencia energética, latencia extremo a extremo, caudal o *throughput*...) requiere soluciones de gestión automática orientadas al servicio. En segundo lugar, es probable que las herramientas basadas en controladores preconfigurados no aprovechen al máximo las capacidades de todas las redes 5G, con diferentes combinaciones de servicios (p.ej., ciudad inteligente frente a industria 4.0), topologías (p.ej., redes heterogéneas frente a redes de macroceldas) y configuración (p.ej., multiconectividad activada o no). Por último, las nuevas funcionalidades 5G como la virtualización de la red o la segmentación de la red extremo a extremo deben considerarse a la hora de gestionar la red [5].

Una funcionalidad que destaca especialmente por su alto impacto en el funcionamiento y rendimiento de las redes es la segmentación de red (*Network slicing*, NS), que permite la operación simultánea de varias redes lógicas independientes, diseñadas para un propósito específico, sobre una infraestructura física compartida [15]. Desde el punto de vista de la gestión de red, con NS surgen nuevas funciones de red (*Network Functions*, NFs) (p.ej., repartidores de capacidad) cuyos parámetros pueden configurarse y optimizarse automáticamente. Además, a la hora de diseñar cualquier solución de autogestión consciente de la existencia de segmentos, hay que considerar aspectos como: a) la división de los recursos de red entre los segmentos, b) la activación, desactivación o redimensionado de segmentos, que altera dicha división de recursos, c) la posibilidad de adaptar o incluso omitir ciertas NFs por segmento y d) las cuestiones de

privacidad, que pueden impedir que el sistema de gestión central de la red (*MANager and Orchestrator*, MANO) acceda a la información a nivel de segmento, gestionada por el arrendatario [16]. Además, se debe tener en cuenta que una solución de autogestión específica puede funcionar de forma diferente en distintos escenarios de NS (p.ej., redes con segmentos multiservicio frente a aquellas con segmentos uniservicio).

Para sortear las limitaciones de las soluciones SON clásicas en las redes celulares actuales, con los últimos avances en el análisis masivo de datos y la inteligencia artificial, es posible aprovechar los datos (p.ej., alarmas, trazas de conexión...) recopilados en el sistema de soporte a las operaciones (*Operations Support System*, OSS) para desarrollar herramientas de gestión 100% automáticas basadas en datos en un paradigma de gestión de redes y servicios sin intervención humana (*Zero-touch Service and Network Management*, ZSM) [17]. Las soluciones basadas en el uso intensivo de datos más avanzadas para las redes ZSM emplean técnicas de aprendizaje automático, capaces de captar las peculiaridades de cada red (p.ej., tipo de escenario, topología, algoritmos de gestión de recursos radio, combinación de servicios, configuración de NS...) [18] [19]. La combinación de NS y ZSM, que da lugar a redes lógicas gestionadas sin intervención humana, ha sido reconocida como el método más eficiente para aprovechar al máximo los activos de la red y garantizar la satisfacción del cliente en las redes de próxima generación [20].

Existe un gran número de casos de uso SON que pueden ser mejorados con el uso intensivo de datos de la red. Por ello, el alcance de esta tesis se ha limitado a dos casos de uso de auto-configuración y auto-optimización muy extendidos: a) el redimensionado de la RAN y b) el balance de carga por movilidad (*Mobility Load Balancing*, MLB).

El redimensionado de la RAN es una tarea fundamental para evitar cuellos de botella de capacidad causados por cambios en los patrones de tráfico, así como para evitar actualizaciones innecesarias de los recursos de la red. Para detectar posibles problemas con antelación, las herramientas de planificación radio proactivas comparan previsiones futuras de tráfico en la hora cargada con estimaciones de la capacidad de la red. La predicción de tráfico a partir de datos históricos se ha abordado tradicionalmente como un problema de análisis de series temporales (*Time Series Analytics*, TSA). Este enfoque ha mostrado buenos resultados para la predicción de tráfico en redes celulares de conmutación de paquetes para distintas escalas geográficas (p.ej., red [190], provincia [191], celda [192]) y temporales (p.ej., minutos [194], horas [192], días [190], meses [191]). Trabajos posteriores exploran el uso de aprendizaje supervisado (*Supervised Learning*, SL) para predecir el tráfico a corto plazo (es decir, en una escala

temporal de segundos o minutos) y a medio plazo (es decir, en una escala temporal de días) en las redes celulares de conmutación de paquetes, cuyos perfiles de tráfico son mucho más complejos [21]. Sin embargo, algunas acciones de replanificación (p.ej., el despliegue de nuevas celdas) pueden conllevar hasta varios meses. Está por comprobar si los algoritmos de SL también mejoran el rendimiento de los métodos clásicos de TSA para predecir el tráfico de celda a largo plazo a partir de series temporales mensuales, que son cortas (es decir, con pocas muestras) y ruidosas.

La estimación de rendimiento consiste en predecir el desempeño de la red en un determinado instante a partir de otra información (conocida o predicha) del estado de la red en ese instante. En lo que respecta al aprendizaje automático, en la literatura se ha estimado el throughput de las celdas en redes HSDPA y LTE aplicando una simple regresión lineal múltiple [22] [23] [24] o complejas redes neuronales profundas [25] sobre información disponible en el OSS. Sin embargo, está por comprobar el potencial de otros modelos no lineales menos propensos al sobreajuste que las redes neuronales profundas para estimar este indicador de rendimiento en las mismas tecnologías radio. Asimismo, en el proceso de dimensionado deben considerarse también otras métricas de rendimiento con mayor impacto en la QoE, como el throughput de usuario. Además, la correlación entre indicadores de celda puede cambiar en las redes 5G segmentadas, requiriéndose un análisis separado para este tipo de redes. En dicho análisis, se debe abordar también la estimación de rendimiento a nivel de segmento, útil para NFs como los repartidores de capacidad [116].

Algunas acciones de redimensionado no pueden aplicarse inmediatamente. Mientras tanto, una forma rentable de aliviar los cuellos de botella de capacidad es repartir el tráfico entre celdas adyacentes. El balance de carga es un caso de uso de optimización automática muy extendido que redistribuye a los usuarios entre las celdas de la red para hacer frente a la distribución irregular de la demanda de tráfico. Dado que esta funcionalidad garantiza que cada usuario sea atendido constantemente por la celda más conveniente, tiene un alto impacto en la QoE. El reparto de tráfico puede realizarse ajustando los parámetros de las antenas, como la potencia de transmisión [26] [27]. Sin embargo, este enfoque aumenta el coste de operación y puede ocasionar huecos de cobertura. Como alternativa, la mayoría de los trabajos abordan el balance de carga a través de la optimización de las NFs asociadas a la movilidad (también conocido como MLB), impulsadas por parámetros lógicos (p.ej., temporizadores, márgenes de potencia...) que pueden ajustarse de forma inmediata sin coste alguno. Algunos autores optan por optimizar los parámetros de reelección de celda [28] [29]. Sin embargo,

la opción preferida es ajustar los márgenes de traspaso, ya que este procedimiento tiene un mayor impacto en el rendimiento de la red. Los primeros algoritmos propuestos buscan equilibrar el rendimiento de la red empleando distintos tipos de controladores (p.ej., proporcionales [30] [32], lógica difusa [31] [86]) y escenarios (p.ej., macroceldas [30], celdas pequeñas [32], femtoceldas [86], redes multi-tecnología [238]...). Contribuciones posteriores proponen algoritmos de reparto de tráfico con criterios de QoE para redes de macroceldas LTE [33] [34]. Para aprovechar al máximo la capacidad en las redes celulares multi-capa actuales, este enfoque debe extenderse para el reparto de tráfico entre capas. Adicionalmente, es necesario diseñar técnicas de reparto de tráfico para RAN segmentadas que permitan garantizar los requisitos de rendimiento recogidos en los acuerdos de nivel de servicio (*Service Level Agreement*, SLA) en términos de throughput, latencia extremo a extremo o fiabilidad [35].

Cabe destacar que las soluciones SON orientadas al servicio, que manejan las conexiones de forma personalizada, asumen un conocimiento previo del servicio demandado por cada usuario. Además, conocer la mezcla de servicios en cada celda puede mejorar los modelos de rendimiento usados en las herramientas de planificación radio. A la hora de clasificar conexiones por servicio, deben tenerse en cuenta los siguientes aspectos: a) el cifrado del tráfico, que impide el uso de técnicas de inspección profunda de paquetes [36], b) la reticencia de los operadores a instalar costosas sondas para capturar los flujos de tráfico en el núcleo de la red, y c) la escasez de datos etiquetados, que dificulta el uso de clasificadores basados en SL como los propuestos en [93] [134]. Como alternativa, la clasificación puede basarse en aprendizaje no supervisado (*UnSupervised Learning*, USL) sobre descriptores de tráfico derivados de trazas de conexión radio.

## B.2 Objetivos

El objetivo principal de esta tesis es desarrollar soluciones automáticas para los casos de uso de auto-configuración y auto-optimización en la RAN mencionados anteriormente. Como aspecto diferenciador, se plantea el uso intensivo de datos disponibles en herramientas comerciales de gestión de red. En concreto, esta tesis persigue 3 objetivos:

- O1.** Diseñar un sistema de clasificación de tráfico encriptado por servicio en la interfaz radio, que permita un tratamiento personalizado por conexión en los algoritmos MLB y que proporcione información de la mezcla de servicios para los modelos de rendimiento usados en las herramientas de planificación radio.

- O2.** Explorar el uso del aprendizaje supervisado sobre los datos recogidos en el OSS para mejorar el rendimiento de las herramientas de planificación radio. En este ámbito, se abordan tres tareas:
- O2.1.** Predicción de tráfico mensual de celda en la hora cargada a largo plazo (es decir, con un horizonte temporal del orden de meses) a partir de series temporales con pocas muestras y ruidosas.
  - O2.2.** Estimación de métricas de throughput a nivel radio que reflejen la capacidad de celda y usuario en redes LTE y HSDPA.
  - O2.3.** Estimación de métricas de throughput a nivel radio que reflejen el rendimiento de celda y segmento en RANs segmentadas.
- O3.** Desarrollar algoritmos de MLB basados en el uso intensivo de datos orientados al servicio para escenarios en los que este enfoque no se ha considerado en la literatura. En este ámbito, se han cubierto dos casos de uso:
- O3.1.** Reparto de tráfico entre portadoras con criterios de QoE para redes LTE multi-portadora.
  - O3.2.** Reparto de tráfico con criterios de SLA para redes 5G segmentadas.

Un aspecto distintivo de esta tesis es la consideración de aspectos prácticos que a menudo no se tienen en cuenta en los trabajos de investigación. Todos los modelos y algoritmos propuestos son soluciones centralizadas concebidas para explotar las trazas de conexión y contadores que se recopilan en el OSS de las redes celulares actuales. Además, los algoritmos de SL se han seleccionado teniendo en cuenta la posibilidad de no disponer de un amplio juego de datos o la reticencia de los operadores a aumentar la carga computacional de sus herramientas de gestión de red, lo que desaconseja el uso de modelos complejos basados en redes neuronales profundas. Por último, la evaluación del rendimiento se ha realizado con datos de redes comerciales o, en su defecto, en entornos realistas de simulación.

## B.3 Metodología de trabajo

A continuación se describen los pasos seguidos para la consecución de los objetivos definidos, desglosando las peculiaridades de cada objetivo específico:



- a) *Definición del problema y revisión de la literatura.* En primer lugar, se identifica el conjunto de casos de uso de SON a abordar. A continuación, se revisa el estado de la investigación y la tecnología en los campos del ámbito de esta tesis. Los principales temas estudiados son: a) las redes auto-organizadas, para identificar las limitaciones de las soluciones actuales de redimensionado y MLB, b) el aprendizaje automático y el análisis masivo de datos, para conocer las técnicas de preprocesado de datos y los algoritmos que se utilizarán en las herramientas desarrolladas; y c) la tecnología 5G, especialmente la funcionalidad de NS, para conocer el funcionamiento de una RAN segmentada, entender su impacto en los casos de uso abordados y garantizar que la posterior implementación de esta funcionalidad en una herramienta de simulación esté alineada con la literatura.
- b) *Formulación del problema y propuesta.* Una vez detectadas las lagunas de investigación, se formulan los problemas a tratar y se proponen nuevas soluciones basadas en el uso intensivo de datos. Para O1 y O2, se propone una metodología para el propósito correspondiente (es decir, clasificación de tráfico, predicción de tráfico o estimación de throughput). Para O3, las contribuciones son nuevos algoritmos de MLB para los escenarios considerados.
- c) *Actualización de herramienta de simulación.* Se actualiza un simulador dinámico de nivel de sistema LTE programado en Matlab para evaluar las soluciones propuestas en O2.3 y O3. Los cambios realizados más importantes son: a) la implementación de un nuevo escenario realista con dos capas de macroceldas para validar el algoritmo diseñado en O3.1, b) la inclusión de la funcionalidad de NS para crear juegos de datos necesarios para la consecución de O2.3 y para validar el algoritmo propuesto en O3.2, y c) la inclusión de nuevos servicios 5G con distintos requisitos de tasa de error de bloque (*Block Error Rate*, BLER) y de calidad de servicio para enriquecer la diversidad de usuarios en las pruebas realizadas en O3.2. Estas actualizaciones se han validado comprobando la coherencia de los resultados en simulaciones largas (una hora de actividad de la red). La herramienta de simulación resultante se describe en detalle en el apéndice [A](#).
- d) *Recogida, preprocesado y análisis de datos.* Los juegos de datos utilizados para O1, O2.1 y O2.2 proceden de redes celulares comerciales. El operador se encarga de recopilar y descargar los datos del OSS. Tras ello, los datos en bruto se exportan a un formato legible mediante herramientas propias proporcionadas por el fabricante y se inspeccionan superficialmente (p.ej., para comprobar los nombres y el significado de los campos disponibles). A continuación, los datos se prepro-

cesan (p.ej., decodificar y sincronizar los eventos de las trazas utilizadas para O1, crear series temporales para O2.1, calcular indicadores útiles a partir de los datos brutos en O1, O2.2 y O2.3...). Para O2.3, a falta de juegos de datos públicos de redes comerciales 5G con NS, los datos se generan mediante simulación, por lo que no es necesario el preprocesado. Una vez creado el juego de datos, se lleva a cabo un análisis estadístico preliminar (p.ej., comprobación de la distribución estadística de cada indicador, análisis de correlación...) y se eliminan los muestras no válidas o atípicas.

- e) *Evaluación del desempeño.* La validación de las soluciones propuestas se realiza con datos de redes reales o, en su defecto, con una herramienta de simulación calibrada con datos de configuración y rendimiento de la red emulada. En todos los casos, las soluciones propuestas se comparan con otras de la literatura consideradas como referencia. Los experimentos relacionados con cada objetivo se ejecutan en distintos entornos, como Matlab (O1, O3.1 y O3.2), SPSS Modeler (O2.1) y Python (O2.2 y O2.3), mientras que el análisis de los resultados se realiza siempre en Matlab. El uso de diferentes plataformas para las tareas de modelado permite identificar las ventajas e inconvenientes de cada herramienta y ofrecer recomendaciones al operador.

## B.4 Desarrollo de la investigación

### B.4.1 Clasificación de tráfico encriptado en redes celulares

La clasificación de conexiones por servicio se ha realizado a partir de las trazas de conexión recopiladas en la interfaz radio. En la formulación del problema, primero se ha presentado un conjunto de descriptores de tráfico clásicos que se pueden calcular a partir de la información contenida en dichas trazas. Tras ello, se ha analizado el tráfico capturado en un terminal móvil conectado a una red LTE comercial cuando demanda aplicaciones de mensajería instantánea, navegación en distintas páginas web, descarga de vídeo progresivo y descarga de ficheros. Este experimento preliminar ha permitido estimar el valor máximo del ratio de volumen de datos transmitido en el UL ( $\eta_{UL}$ ) para los servicios tipo *full buffer*. También se ha señalado la dependencia de otros descriptores de tráfico de las condiciones de la red (p.ej., esquema de control de flujo).

A continuación, se ha propuesto un nuevo esquema de clasificación de tráfico. El método se basa en USL, concretamente en agrupamiento jerárquico aglomerativo. Por

lo tanto, no requiere datos etiquetados, raramente disponibles en las redes móviles comerciales. Nótese que la demanda de servicios en dichas redes no es uniforme. Para mejorar el desempeño del algoritmo de agrupamiento ante juegos de datos desequilibrados, primero se han creado 4 bloques de conexiones basados en el conocimiento previo del valor de ciertos descriptores de tráfico clásicos (p.ej.,  $\eta_{UL}$ ) para distintos servicios. A continuación se ha realizado el agrupamiento de las conexiones de cada bloque para obtener una clasificación más detallada. Para evitar la influencia de las condiciones de la red en la clasificación, el algoritmo de agrupamiento recibe como entrada un conjunto de descriptores de tráfico novedosos que caracterizan las conexiones a nivel de ráfaga. Dichos descriptores pueden construirse a partir de la información contenida en las trazas mediante un modelo analítico.

La validación se ha realizado sobre un juego de trazas recopiladas una red LTE comercial que incluye 162.965 conexiones con identificador de calidad de servicio (*Quality-of-service Class Identifier*, QCI) del 6 al 9, correspondientes a servicios multimedia y basados en el protocolo de control de transmisión (*Transmission Control Protocol*, TCP) [121]. En primer lugar, se ha comprobado el desempeño del algoritmo de agrupamiento aplicado sobre todas las conexiones simultáneamente. La incoherencia de los resultados obtenidos ha mostrado la necesidad de dividir las conexiones en bloques previamente. El esquema de clasificación propuesto ha dividido las conexiones en 8 grupos, que se han asociado con servicios de notificación emergente, mensajería instantánea, intercambio de archivos, audio y vídeo, servicios de tipo *full buffer*, navegación web y redes sociales. El porcentaje de volumen en el enlace descendente (*DownLink*, DL) agregado por categoría es consistente con la cuota de tráfico reportada para las redes comerciales el año en que se recogieron los datos según [157], confirmando el potencial de los descriptores de tráfico a nivel de ráfaga para agrupar las conexiones por tipo de servicio.

## B.4.2 Estimación de rendimiento en redes de acceso radio celulares

La estimación de indicadores de throughput radio a partir de la información disponible en el OSS se ha formulado como un problema de regresión. Para su resolución, se ha propuesto una metodología basada en el uso de SL. Se ha comparado el desempeño de algoritmos clásicos de SL basados en regresión lineal, distancia, vectores, árboles de decisión y redes neuronales artificiales. Para reducir la complejidad de los modelos, se

ha llevado a cabo un proceso de selección de atributos.

### a) Estimación de rendimiento en redes no segmentadas

En primer lugar se han considerado las redes HSDPA y LTE. En cada tecnología de acceso radio (*Radio Access Technology, RAT*), se ha definido un conjunto específico de predictores candidatos para estimar el throughput agregado de celda y el throughput medio de usuario por celda en el DL (en adelante, throughput de celda/usuario en el DL, respectivamente) en escenarios de congestión. Dichos predictores se construyen a partir de indicadores de configuración (*Configuration Management, CM*) y contadores de rendimiento (*Performance Management, PM*) agregados a nivel de celda. Se han comparado seis algoritmos de SL: regresión lineal múltiple, bosque aleatorio, regresión de vectores de soporte,  $k$ -vecinos más cercanos y perceptrones multicapa superficial y profundo. Para cada algoritmo, se ha realizado selección de atributos con dos métodos basados en correlación lineal y en envoltura, respectivamente.

La evaluación del desempeño se ha llevado a cabo sobre juegos de datos recopilados en una red HSDPA comercial y en una red LTE comercial, respectivamente. Los resultados han demostrado que los métodos de selección de atributos basados en envoltura superan a los enfoques basados en correlación lineal a la hora de encontrar el subconjunto óptimo de predictores (y, por tanto, la información que debe recopilarse en el OSS). En ambas RATs, los algoritmos de SL no lineales han mostrado mejor desempeño de la regresión lineal múltiple, especialmente para estimar el throughput de usuario. Aún más importante, el rendimiento de las redes neuronales profundas, complejas y propensas al sobreajuste, ha sido mejorado por otros algoritmos más sencillos. Los mejores algoritmos han sido el perceptrón multicapa superficial para HSDPA y  $k$ -vecinos más cercanos para LTE, con un error porcentual absoluto medio inferior al 10% cuando se estiman ambos indicadores con modelos entrenados con menos de 2.000 muestras y, como máximo, 5 predictores.

### b) Estimación de rendimiento en redes segmentadas

A continuación, el análisis se ha extendido a los nuevos escenarios 5G con NS. Ante la inexistencia de juegos de datos de redes comerciales con NS, la evaluación se ha realizado sobre datos simulados. Para ello, se ha implementado la funcionalidad de NS en la herramienta de simulación descrita en el apéndice [A](#), que emula la actividad de una red celular LTE-*Advanced* en un entorno realista. Se han considerado los servicios

de llamada, descarga de vídeo progresivo, navegación web y descarga de ficheros. Se ha ejecutado un mismo conjunto de diez simulaciones en tres escenarios: un escenario sin NS y dos escenarios con NS con segmentos uniservicio (NS\_SS) y multiservicio (NS\_MS), respectivamente. Esto ha permitido construir tres juegos de datos a nivel de celda (uno por escenario) y dos juegos de datos a nivel de celda-y-segmento (uno por escenario con NS). Los juegos de datos de celda se han usado para estimar el throughput agregado de celda en el DL, mientras que los juegos de datos por celda-y-segmento se han empleado para estimar el throughput agregado por celda-y-segmento en el DL (en adelante, throughput de segmento en el DL). Para ello, se ha definido un conjunto de predictores calculados a partir de trazas radio y CM/PM agregados por celda y por celda-y-segmento. A la vista de los resultados obtenidos en el análisis anterior, se han descartado el algoritmo de regresión lineal y la selección de atributos basada en correlación, y se han incluido los algoritmos basados en árboles de decisión con potenciación adaptativa (*Adaptive Boosting*, AdaBoost) y potenciación extrema del gradiente (*eXtreme Gradient Boosting*, XGBoost), respectivamente.

Un análisis preliminar de los tres juegos de datos a nivel de celda ha demostrado el impacto de habilitar el NS en las relaciones entre indicadores de red y el throughput de celda, así como la existencia de ciertas diferencias entre los escenarios NS\_SS y NS\_MS, lo que justifica la necesidad del estudio y la conveniencia de realizar un análisis separado para cada escenario de NS.

Los resultados han mostrado que, con una adecuada selección de atributos, todos los algoritmos evaluados alcanzan un rendimiento similar y aceptable (es decir, con error inferior al 10%) para estimar el throughput de celda en los escenarios NS\_SS y NS\_MS. En ambos casos, el algoritmo que muestra mejor relación entre precisión y complejidad es el perceptrón multicapa superficial, con error absoluto normalizado medio inferior al 2% para modelos basados en cuatro predictores relacionados con el ancho de banda, la utilización de recursos radio y la eficiencia espectral. Estos indicadores pueden calcularse a partir de PMs/CMs agregados a nivel de celda. En cambio, sólo los modelos basados en árboles de decisión y perceptrones multicapa han obtenido una precisión aceptable al estimar el throughput de segmento. Además, el desempeño ha sido peor en el escenario con segmentos multiservicio. Los mejores modelos han sido bosque aleatorio para el escenario NS\_SS y el perceptrón multicapa superficial para el escenario NS\_MS. Es destacable que los cinco atributos de entrada a estos modelos no solo incluyen predictores calculados a partir de PMs/CMs a nivel de celda, sino también predictores calculados a partir de PMs/CMs a nivel de segmento e

información sobre la mezcla de servicios por segmento derivada de trazas de conexión.

Tras este análisis, se puede concluir que las técnicas de SL no profundas pueden estimar métricas de throughput radio en diferentes RAT y escenarios con un rendimiento adecuado. Además, el hecho de que los modelos a nivel de segmento requieran información sobre la mezcla de servicios confirma la utilidad del método de clasificación de tráfico desarrollado en esta tesis.

### B.4.3 Predicción de tráfico de celda a largo plazo

La tarea de predecir el tráfico de celda a largo plazo (es decir, con varios meses de antelación) se ha abordado como un problema de series temporales. Como parte de la formulación del problema, se ha presentado un análisis de la autocorrelación del tráfico de celda a nivel horario y a nivel de la hora cargada del mes empleando datos de redes LTE comerciales. Dicho análisis ha puesto de manifiesto las dificultades que plantea la predicción de tráfico a largo plazo, basada en series temporales con pocas muestras y ruidosas.

A continuación, se ha llevado a cabo un estudio comparativo del desempeño de algoritmos clásicos de SL, no probados hasta ahora para este fin, frente a los enfoques clásicos de TSA. En concreto, se han comparado seis algoritmos: media móvil autorregresiva integrada con estacionalidad, Holt-Winters aditivo, regresión de vectores de soporte, bosque aleatorio, red neuronal basada en perceptrón multicapa superficial y red neuronal basada unidades con memoria (*Long Short-Term Memory*, LSTM). Un aspecto clave al definir la metodología de predicción ha sido la construcción de los modelos de SL, dado que la corta longitud de las series temporales mensuales impide entrenar un modelo específico por celda.

El análisis ha constado de tres experimentos realizados sobre un juego de datos único por su escasez, que incluye mediciones de tráfico recogidas durante dos años y medio en una red LTE real con 7160 celdas que cubre un país completo. El primer experimento ha demostrado que los algoritmos de SL superan a los TSA en precisión y cantidad de datos históricos necesaria. El segundo experimento ha confirmado que dichas conclusiones se pueden extrapolar a todos los meses del año, y que aumentar el horizonte de predicción de 3 a 6 meses disminuye considerablemente la precisión. El tercer experimento ha demostrado que es conveniente desarrollar modelos específicos para celdas de alto tráfico, donde la precisión de la predicción es crítica. Inesperadamente, los modelos de bosque aleatorio y perceptrón multicapa han mostrado los

mejores resultados (error porcentual absoluto medio del 11% en celdas cargadas), con un rendimiento similar a la red neuronal basada en unidades LSTM diseñadas para modelar dependencias temporales. Estos resultados confirman la limitada predictibilidad del tráfico de la hora cargada del mes en comparación con las series de tráfico horarias o diarias. Ninguno de los algoritmos considerados ha mostrado resultados altamente precisos, especialmente para los meses de verano, con vacaciones en el país donde opera la red.

#### B.4.4 Reparto de tráfico en redes celulares

Por último, se ha abordado la tarea de diseñar algoritmos de MLB basados en el uso intensivo de datos que tienen en cuenta aspectos propios de cada servicio. Se han considerado dos casos de uso: MLB entre portadoras con criterios de QoE, y MLB en redes con NS considerando aspectos de SLA. En ambos casos, el reparto de tráfico se ha formulado como un problema de control. Además, a diferencia de las soluciones de reparto de tráfico tradicionales, basadas en indicadores calculados a partir de contadores agregados por celda, los algoritmos propuestos se rigen por nuevos indicadores calculados a partir de trazas de conexión, que reflejan el rendimiento de los usuarios individuales.

##### a) Reparto de tráfico basado en QoE en redes LTE multi-portadora

En primer lugar, se ha propuesto una estrategia de reparto de tráfico en redes LTE multi-portadora que busca mejorar la QoE global del sistema. Para ello, primero se activa un mecanismo de traspaso entre portadoras basado en la calidad de señal recibida (*Reference Signal Received Quality*, RSRQ), y posteriormente se ajustan los márgenes de traspaso por adyacencia con un algoritmo novedoso de MLB con criterios de QoE. El ajuste de parámetros se realiza con un controlador heurístico de tipo proporcional guiado por un nuevo indicador que evalúa el impacto medio de los traspasos para todos los usuarios servidos por las celdas de una adyacencia. Una prueba de concepto realizada en una red piloto LTE (no incluida en este documento por brevedad) ha confirmado que dicho indicador puede construirse procesando trazas de conexión radio [252].

La evaluación del rendimiento se ha llevado a cabo con el simulador mencionado anteriormente, en el que se ha implementado un escenario realista con dos capas de macro celdas. Se han considerado servicios de voz, descarga progresiva de vídeo, descarga de ficheros y navegación web. Experimentos realizados emulando diferentes escenarios de

movilidad han demostrado que la estrategia de reparto de tráfico propuesta supera significativamente a las técnicas clásicas de MLB basadas en equilibrar el porcentaje de utilización de recursos radio entre celdas, mejorando la QoE global de la red en un 19% tras 10 iteraciones. La mejora se logra traspasando el exceso de tráfico en la capa de cobertura hacia la capa de capacidad, de modo que los usuarios aprovechan al máximo el gran ancho de banda disponible en esta última.

### **b) Reparto de tráfico basado en SLA en redes de acceso radio segmentadas**

A continuación, el análisis se ha extendido a escenarios 5G con segmentación. Se ha propuesto un nuevo algoritmo de reparto de tráfico que busca aumentar el cumplimiento de los SLAs. Para ello, el algoritmo ajusta los márgenes de traspaso intra-frecuencia en un esquema de movilidad con parámetros definidos por segmento. El proceso de ajuste se basa en un indicador novedoso que refleja el desequilibrio en el cumplimiento del SLA por segmento entre celdas vecinas. En cada adyacencia y segmento, un controlador independiente basado en proporcionalidad incrementa/decrementa el valor de los márgenes de traspaso en función de dicho indicador. Para evitar acciones ineficaces e inestabilidades en la red, el algoritmo opera solo en las adyacencias más relevantes de la red, que se dividen en grupos compuestos por adyacencias con distintas celdas. Cada 5 segundos, se ajustan los parámetros de las adyacencias de un grupo distinto.

La evaluación del rendimiento se ha llevado a cabo simulando la actividad de una red con NS realista, con tres segmentos que sirven tráfico eMBB y uRLLC de usuarios con diferentes patrones de movilidad. Para ello, se han implementado modelos de tráfico para los servicios de transmisión de vídeo en directo, comunicaciones hápticas y conducción autónoma. El SLA por segmento se ha definido en términos de throughput de sesión y cumplimiento de requisitos de latencia. Los resultados han mostrado el bajo rendimiento del esquema MLB tradicional (es decir, que no considera la existencia de segmentos) en escenarios de NS, especialmente para aquellos segmentos con baja asignación de recursos radio. El algoritmo propuesto ha mostrando mejor desempeño que otras estrategias de MLB por segmento, que a) no se rigen por indicadores de SLA, o b) que no realizan agrupación de adyacencias, convergiendo en solo 15 minutos. La mejora se obtiene equilibrando el cumplimiento del SLA a lo largo del escenario.



## B.5 Conclusiones

A continuación se exponen las principales contribuciones de esta tesis a nivel científico, desglosadas por problema abordado.

### 1) *Clasificación de tráfico encriptado en redes celulares*

En este ámbito, las principales contribuciones son:

- a) Se ha desarrollado un modelo analítico para construir un nuevo conjunto de descriptores de tráfico a nivel de ráfaga a partir de información contenida en trazas de conexión radio.
- b) Por primera vez, se ha propuesto un esquema de clasificación de tráfico por servicio basado en trazas radio, que puede utilizarse sin sondas de red ni datos etiquetados, identificar nuevos tipos de aplicaciones que surjan en la red y adaptarse fácilmente a diferentes RATs. El método primero crea grandes bloques de conexiones a partir de conocimiento experto, y después aplica agrupamiento jerárquico aglomerativo sobre las conexiones de cada bloque empleando los descriptores de tráfico a nivel de ráfaga. El criterio para la división de conexiones en bloques se ha derivado de un análisis del tráfico de un terminal conectado a una red LTE, que ha permitido estimar el valor de algunos descriptores de tráfico para distintos servicios.
- c) La validación del método propuesto sobre un juego de trazas de una red LTE comercial ha confirmado el potencial de los descriptores de tráfico de nivel de ráfaga para agrupar las conexiones por servicio, así como la importancia de añadir conocimiento experto a las técnicas de USL para realizar la clasificación, debido a la demanda desigual de servicios en las redes móviles actuales.

### 2) *Estimación de rendimiento en redes de acceso radio celulares*

Esta tesis ha presentado el primer estudio comparativo del desempeño de distintos algoritmos clásicos de SL para estimar indicadores de throughput radio a partir de información recopilada en el OSS. Para ello, se ha adaptado una metodología clásica de SL a dicha aplicación (definición de atributos de entrada y salida, cifras de mérito...). Se han comparado algoritmos basados en regresión lineal, vectores, distancia, árboles de decisión y redes neuronales. Las principales contribuciones son:

- a) Se ha evaluado por primera vez el rendimiento de algoritmos clásicos de SL no profundos para estimar el throughput de celda y usuario en el DL en la hora

cargada en una red LTE comercial a partir de información del OSS. Hasta ahora, solo se había considerado el uso de regresión lineal múltiple [23] [24] y redes neuronales profundas [25].

- b) El análisis se ha extendido a una red HSDPA comercial, donde la literatura solo cubre la estimación del throughput de celda en el DL con regresión lineal múltiple [22].
- c) Se ha presentado el primer análisis comparativo del desempeño de algoritmos de SL para estimar el throughput de celda y segmento en el DL en redes con NS a partir de información recopilada en el OSS. Se han considerado dos escenarios diferentes de NS con segmentos uniservicio y multiservicio, respectivamente. Como no existen redes comerciales operativas con NS, para generar los juegos de datos se ha implementado la funcionalidad de NS en una herramienta de simulación existente. Dicha herramienta es ahora un recurso muy valioso para el desarrollo de soluciones SON en redes 5G segmentadas.
- d) En todos los casos, se ha identificado el conjunto mínimo de indicadores de rendimiento de la red que deben almacenarse en el OSS para la estimación.
- e) Se ha presentado un análisis del impacto de la activación de la funcionalidad de NS en la correlación entre los indicadores de red y el throughput de celda en el DL, que justifica la inclusión de este indicador en el estudio realizado en c).
- f) Los resultados han demostrado que es posible estimar las métricas de throughput definidas en todas las redes consideradas con un rendimiento aceptable (error inferior al 10%) a través del uso de SL no profundo sobre información recopilada en el OSS. Para ello, debe emplearse un método de selección de atributos de envoltura. El mejor modelo (es decir, combinación de algoritmo de SL y conjunto predictores) puede variar en cada red concreta y para cada métrica a estimar. Además, se ha comprobado que es conveniente proporcionar información sobre la mezcla de servicios a los modelos de rendimiento a nivel segmento. En segmentos multiservicio, esta información puede obtenerse procesando trazas de conexión radio.

### 3) *Predicción de tráfico de celda a largo plazo*

En este ámbito, las principales aportaciones son:

- a) Se ha presentado un análisis de autocorrelación que ha evidenciado las diferencias a nivel de predictibilidad de las series temporales de tráfico en la hora cargada del

mes frente a otras series con mayor resolución temporal, justificando la necesidad de estudiar estos problemas de forma separada.

- b) Se ha realizado el primer estudio comparativo del rendimiento de algoritmos clásicos de SL frente a técnicas clásicas de TSA para la predicción de tráfico de celda a largo plazo (es decir, en un horizonte temporal de meses) en una red LTE comercial. Como parte de este estudio, se ha analizado el impacto de los principales parámetros de diseño, como son la ventana de recopilación de datos, el horizonte de predicción y el número de modelos a crear.
- c) Los resultados han demostrado que los algoritmos de SL mejoran el rendimiento de las técnicas clásicas de TSA y han confirmado la limitada predictibilidad del tráfico de la hora cargada del mes en comparación con las series de tráfico horarias o diarias, evidenciando la necesidad de seguir investigando en esta dirección. De hecho, una extensión de este análisis realizada con posterioridad ha revelado que la gestión de los valores atípicos por serie, combinada con el suavizado o la descomposición aditiva de las series temporales, puede mejorar las predicciones de forma significativa. No obstante, se recomienda a los operadores almacenar medidas de tráfico con una mayor resolución temporal a largo plazo para explotar las capacidades de los algoritmos de SL.

#### 4) *Reparto de tráfico basado en QoE en redes LTE multi-portadora*

Las principales contribuciones en esta línea de investigación son:

- a) Se ha desarrollado un nuevo indicador que evalúa el impacto de un evento (en este trabajo, un traspaso entre frecuencias) en la QoE de las celdas de cada adyacencia. Este indicador, derivado a partir de trazas radio, puede adaptarse para optimizar los parámetros lógicos que controlan cualquier mecanismos de movilidad que implique eventos como la agregación de portadoras o la multiconectividad.
- b) Se ha abordado por primera vez el problema del reparto de tráfico entre celdas vecinas que trabajan a distintas frecuencias con criterios de QoE. La estrategia de tráfico propuesta se basa en ajustar los márgenes que controlan el proceso de traspaso de usuarios desde las capas de capacidad a las capas de cobertura, que se configura previamente para dispararse por el evento A3 basado en RSRQ. El ajuste se realiza mediante un controlador heurístico basado en el indicador de cambio de QoE por adyacencia, que busca minimizar el efecto de estos traspasos en la QoE global.

- c) Se ha presentado una comparación exhaustiva del algoritmo propuesto los métodos de reparto de tráfico de la literatura en un simulador dinámico LTE que emula una red realista, considerándose varios escenarios de movilidad.

5) *Reparto de tráfico basado en SLA en redes de acceso radio segmentadas*

Las principales aportaciones en este ámbito son:

- a) Se ha desarrollado un nuevo indicador que evalúa el desequilibrio del cumplimiento de SLA por segmento en celdas vecinas. Este indicador, derivado de trazas radio, puede calcularse para SLAs que incluyan cualquier conjunto de métricas de rendimiento.
- b) Se ha abordado por primera vez el problema del reparto de tráfico entre celdas vecinas en escenarios con NS con criterios de SLA. La estrategia de tráfico propuesta se basa en ajustar los márgenes que controlan el proceso de traspaso intra-frecuencia en un esquema de movilidad consciente de la existencia de segmentos. El ajuste se realiza mediante un controlador heurístico basado en el indicador de desequilibrio de SLA por adyacencia y segmento, que busca homogeneizar el cumplimiento de SLA en la red.
- c) Se ha presentado una comparación exhaustiva del rendimiento del algoritmo propuesto frente a otros métodos de reparto de tráfico de la literatura, tanto en su versión original como adaptados a la existencia de segmentos, a través de la simulación de una red con NS realista con segmentos que ofrecen servicios eMBB y uRLLC.

Aun siendo concebidas para ser ejecutadas de manera centralizada, todas las soluciones propuestas aquí pueden implementarse en una arquitectura SON distribuida siempre que exista una interfaz entre las estaciones base. Por ejemplo, el esquema de clasificación del tráfico puede aplicarse por celda. Del mismo modo, el indicador de cambio de QoE por adyacencia que dirige el algoritmo MLB inter-frecuencia propuesto puede calcularse fácilmente si las estaciones base pueden intercambiar datos. Por último, los modelos que requieren aprendizaje supervisado pueden entrenarse con aprendizaje federado [255]. En los escenarios de NS, el uso de aprendizaje federado también podría ser útil para resolver posibles problemas de privacidad que impidan a los operadores acceder a la información del rendimiento de los segmentos [255].

## B.6 Lista de contribuciones

En el marco de esta tesis se han elaborado las publicaciones que se enumeran a continuación:

### Publicaciones en revistas

- [I] **C. Gijón**, M. Toril, S. Luna, M.L. Marí, “A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks”, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414-9424, oct. 2019.
- [II] **C. Gijón**, M. Toril, M. Solera, S. Luna, L. Jiménez, “Encrypted traffic classification based on unsupervised learning in cellular radio access networks”, *IEEE Access*, vol. 8, pp. 167252-167263, sep. 2020.
- [III] **C. Gijón**, M. Toril, S. Luna, J.L. Bejarano, M.L. Marí, “Estimating pole capacity from radio network performance statistics by supervised learning”, *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2090-2101, dic. 2020.
- [IV] **C. Gijón**, M. Toril, S. Luna, M.L. Marí, J.M. Ruiz, “Long-term data traffic forecasting for network dimensioning in LTE with short time series”, *Electronics*, vol. 10, p. 1151, may. 2021.
- [V] **C. Gijón**, M. Toril, S. Luna, “Data-driven estimation of throughput performance in sliced radio access networks via supervised learning”, *IEEE Transactions on Network and Service Management*, publicación aceptada en sep. 2022.
- [VI] **C. Gijón**, T. Mahmoodi, S. Luna, M. Toril, “Data-driven Slice-Aware Traffic Steering for Service Level Agreement Compliance in Sliced Radio Access Networks”, en proceso de revisión.

### Contribuciones a conferencias y reuniones científicas

- [VII] **C. Gijón**, M. Toril, S. Luna, M. L. Marí, “A data-driven user steering algorithm for optimizing user experience in multi-tier LTE networks”, *9th MC and scientific meeting of COST CA15104 (IRACON)*, Dublín (Irlanda), ene. 2019.
- [VIII] **C. Gijón**, M. Toril, S. Luna, M.L. Marí, “Mejora de la calidad de experiencia en redes LTE multi-portadora”, *XXXIV Simposio Nacional de la Unión Científica*

*Internacional de Radio (URSI 2019)*, Sevilla (España), sep. 2019.

- [IX] **C. Gijón**, M. Toril, S. Luna, J. L. Bejarano, M. L. Marí, “Estimación de la capacidad en redes LTE mediante aprendizaje supervisado”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (España), sep. 2020.
- [X] **C. Gijón**, M. Toril, S. Luna, “Modelling performance in sliced radio access networks with supervised learning”, *1th scientific meeting of COST CA20120 (INTERACT)*, Bolonia (Italia), feb. 2022.
- [XI] **C. Gijón**, M. Toril, S. Luna, “Modelado de rendimiento de segmento en redes de acceso radio mediante aprendizaje supervisado”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.

En [II] se presenta la estrategia de clasificación de tráfico encriptado descrita en el capítulo 3. El análisis sobre la estimación de los indicadores de rendimiento explicado en el capítulo 4 se describe en [III, V, IX–XI]. [III, IX] se centran en las redes no segmentadas. La tecnología LTE se considera primero en [IX], y [III] añade el caso de HSDPA. [V, X, XI] cubren la estimación de throughput en redes con NS a nivel de segmento [X, XI] y de celda [V, X, XI]. En [IV] se describen los experimentos de predicción de tráfico de celda a largo plazo presentados en el capítulo 5. Por último, los algoritmos de reparto de tráfico presentados en el capítulo 6 para redes multi-portadora y redes segmentadas se proponen en [I, VII, VIII] y [VI], respectivamente.

Esta tesis ha sido financiada por el Ministerio de Educación, Cultura y Deporte (ayuda FPU17/04286). Todas las publicaciones anteriores se han desarrollado en el marco de varios proyectos de investigación:

- Proyecto TEC2015-69982-R (*Métodos de planificación y optimización de redes B4G*), financiado por el Ministerio de Economía y Competitividad [I, VII, VIII].
- Proyecto ICT-760809 (*ONE5G: E2E-aware Optimizations and advancements for the Network Edge of 5G New Radio*), financiado por Horizonte 2020 [I, VII, VIII].
- Proyecto RTI2018-099148-B-I00 (*Métodos de planificación automática para redes 5G virtualizadas*), financiado por el Ministerio de Ciencia, Innovación y Universidades [II, III, V, IX–XI].
- Proyecto UMA18-FEDERJA-256 (*Predicción de QoE en redes 5G*), financiado

por la Junta de Andalucía [IV].

- 8.06/5.59.5705-2 Contrato IDEA con Ericsson España (*Desarrollo de casos de uso para el diseño, optimización y dimensionamiento de redes móviles*), financiado parcialmente por la Junta de Andalucía (Agencia IDEA, Consejería de Ciencia, Innovación y Empresa) [IV, V, X, XI].

Asimismo, las contribuciones de esta tesis han formado parte de las acciones europeas COST CA15104 (*Inclusive Radio Communications, IRACON*) [VII] y CA20120 (*Intelligence-Enabling Radio Communications for Seamless Inclusive Interactions, INTE-RACT*) [X].

La autora de esta tesis también ha sido coautora las siguientes publicaciones relacionadas con los temas aquí tratados.

### Publicaciones de revista relacionadas

- [XII] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “A QoE-driven traffic steering algorithm for LTE networks”, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11271-11282, nov. 2019.
- [XIII] P. Sánchez, S. Luna, M. Toril, **C. Gijón**, J.L. Bejarano, “A data-driven scheduler performance model for QoE assessment in a LTE radio network planning tool”, *Computer Networks*, vol. 173, p. 107228, may. 2020.
- [XIV] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “A self-tuning algorithm for optimal QoE-driven traffic steering in LTE”, *IEEE Access*, vol. 8, pp. 156707-156717, ago. 2020.
- [XV] A. García, **C. Gijón**, M. Toril, S. Luna, “Data-driven construction of user utility functions from radio connection traces in LTE”, *Electronics*, vol. 10, p. 829, mar. 2021.
- [XVI] J. L. Bejarano, M. Toril, M. Fernández, **C. Gijón**, S. Luna, “A deep-learning model for estimating the impact of social events on traffic demand on a cell basis”, *IEEE Access*, vol. 9, pp. 71673-71686, may. 2021.
- [XVII] L. Jiménez, M. Solera, M. Toril, **C. Gijón**, P. Casas, “Content matters: clustering web pages for qoe analysis with webCLUST”, *IEEE Access*, vol. 9, pp. 123873-123888, ago. 2021.
- [XVIII] M.L. Marí, S. Mwanje, S. Luna, M. Toril, H. Sanneck, **C. Gijón**, “A service-

centric Q-learning algorithm for mobility robustness optimization in LTE”, *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3541-3555, sep. 2021.

- [XIX] J. Sánchez, M. Toril, V. Wille, **C. Gijón**, M. Fernández, “On the improvement of cellular coverage maps by filtering MDT measurements”, *IEEE Transactions on Mobile Computing*, ene. 2022.

### Contribuciones a congresos y reuniones científicas relacionadas

- [XX] M. L. Marí, S. Luna, M. Toril, **C. Gijón**, “A QoE-driven traffic steering algorithm for LTE networks”, *9th MC and scientific meeting of COST CA15104 (IRACON)*, Dublín (Irlanda), ene. 2019.
- [XXI] M.L. Marí, S. Luna, M. Toril, **C. Gijón**, “Optimización de la calidad de experiencia en redes LTE mediante el reparto de tráfico”, *XXXIV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2019)*, Sevilla (España), sep. 2019.
- [XXII] A. J. García, **C. Gijón**, M. Toril, S. Luna, “Data-driven construction of user utility functions from connection traces in LTE”, *12th MC Meeting and Final Workshop of COST CA15104 (IRACON)*, Lovaina la Nueva (Bélgica), ene. 2020.
- [XXIII] M. L. Marí-Altozano, S. Mwanje, S. Luna, M. Toril, **C. Gijón**, “Una visión basada en QoE para algoritmo MRO en redes LTE”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (España), sep. 2020.
- [XXIV] M. Toril, **C. Gijón**, S. Luna, M. Fernández, “Asignación de unidades de banda base en redes de acceso radio centralizadas por teoría de grafos”, *XXXV Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2020)*, Málaga (España), sep. 2020.
- [XXV] J. M. Sánchez, M. Toril, **C. Gijón**, S. Luna, M. Fernández, “Análisis de capacidad de redes de acceso radio centralizadas en escenarios heterogéneos”, *XXXVI Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2021)*, Vigo (España), sep. 2021.
- [XXVI] J. L. Bejarano, M. Toril, M. Fernández, **C. Gijón**, S. Luna, “Evaluación de modelos de deep-learning para series temporales de tráfico horario en redes celulares”, *XXXVI Simposio Nacional de la Unión Científica Internacional de Radio*



- (*URSI 2021*), Vigo (España), sep. 2021.
- [XXVII] M. I. Quesada, S. Luna, M. Toril, **C. Gijón**, A. Durán, “Comparación de estrategias de entrenamiento de modelos de predicción de tráfico mensual en redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.
- [XXVIII] C. Cerezo, S. Luna, A. Durán, M. Toril, **C. Gijón**, “Gestión de valores anómalos en series temporales de redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.
- [XXIX] J. L. Bejarano, M. Toril, **C. Gijón**, S. Luna, A. Durán, “Obtención de intervalos de confianza en redes neuronales para predicción en redes celulares”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.
- [XXX] N. González, M. Solera, F. Ruiz, **C. Gijón**, M. Toril, “Modelo de evaluación de calidad de experiencia para servicios de vídeo inmersivo por LTE basado en drones”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.
- [XXXI] J. M. Sánchez, M. Toril, **C. Gijón**, J. L. Bejarano, S. Luna, “Filtrado de trazas MDT de alta movilidad mediante aprendizaje supervisado”, *XXXVII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2022)*, Málaga (España), sep. 2022.
- [XXXII] N. González, M. Solera, F. Ruiz, **C. Gijón**, M. Toril, “A quality of experience evaluation methodology for first-person-view drone control in cellular networks”, in *16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2022)*, Montreal (Canadá), oct. 2022.



UNIVERSIDAD  
DE MÁLAGA

# Bibliography

- [1] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception,” in *TS 36.101*, version 15.2.0, 2018.
- [2] L.-R. Hu and S. S. Rappaport, “Personal communication systems using multiple hierarchical cellular overlays,” *IEEE Journal on Aected Areas in Communications*, vol. 13, no. 2, pp. 406–415, 1995.
- [3] L. Pierucci, “The quality of experience perspective toward 5G technology,” *IEEE Wireless Communications*, vol. 22, no. 4, pp. 10–16, 2015.
- [4] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, “Toward 6G networks: Use cases and technologies,” *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.
- [5] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.
- [6] M. Toril, R. Ferrer, S. Pedraza, V. Wille, and J. J. Escobar, “Optimization of half-rate codec assignment in GERAN,” *Wireless Personal Communications*, vol. 34, no. 3, pp. 321–331, 2005.
- [7] M. Toril and V. Wille, “Optimization of handover parameters for traffic sharing in GERAN,” *Wireless Personal Communications*, vol. 47, no. 3, pp. 315–336, 2008.
- [8] R. Barco, V. Wille, L. Díez, and M. Toril, “Learning of model parameters for fault diagnosis in wireless networks,” *Wireless Networks*, vol. 16, no. 1, pp. 255–271, 2010.

- [9] P. A. García, A. Á. González, A. A. Alonso, B. C. Martínez, J. M. A. Pérez, and A. S. Esguevillas, “Automatic umts system resource dimensioning based on service traffic analysis,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 323, 2012.
- [10] H. Mfula, T. Isotalo, and J. K. Nurminen, “Self-optimization of power parameters in WCDMA networks,” in *2015 IEEE International Conference on High Performance Computing & Simulation (HPCS)*, 2015, pp. 80–87.
- [11] R. M. Khanafer, B. Solana, J. Triola, R. Barco, L. Moltsen, Z. Altman, and P. Lazaro, “Automated diagnosis for umts networks using bayesian network approach,” *IEEE Transactions on vehicular technology*, vol. 57, no. 4, pp. 2451–2461, 2008.
- [12] P. A. Sánchez, S. Luna-Ramírez, M. Toril, C. Gijón, and J. L. Bejarano-Luque, “A data-driven scheduler performance model for QoE assessment in a LTE radio network planning tool,” *Computer Networks*, vol. 173, p. 107186, 2020.
- [13] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramírez, and R. G. Garaluz, “Self-tuning of service priority parameters for optimizing quality of experience in LTE,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3534–3544, 2018.
- [14] D. Palacios, S. Fortes, I. de-la Bandera, and R. Barco, “Self-healing framework for next-generation networks through dimensionality reduction,” *IEEE Communications Magazine*, vol. 56, no. 7, pp. 170–176, 2018.
- [15] NGMN Alliance, “Description of network slicing concept,” in *NGMN 5G P1: Requirements Architecture Work Stream End-to-End Architecture*, 2016.
- [16] S. Zhang, “An overview of network slicing for 5g,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.
- [17] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, “A survey on Zero-touch network and Service Management (ZSM) for 5G and beyond networks,” *Journal of Network and Computer Applications*, p. 103362, 2022.
- [18] J. Moysen and L. Giupponi, “From 4G to 5G: Self-organized network management meets machine learning,” *Computer Communications*, vol. 129, pp. 248–268, 2018.

- [19] H. Fourati, R. Maaloul, L. Chaari, and M. Jmaiel, “Comprehensive survey on self-organizing cellular network approaches applied to 5G networks,” *Computer Networks*, vol. 199, p. 108435, 2021.
- [20] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, “Machine learning-based zero-touch network and service management: A survey,” *Digital Communications and Networks*, 2021.
- [21] W. Jiang, “Cellular traffic prediction with machine learning: A survey,” *Expert Systems with Applications*, p. 117163, 2022.
- [22] V. Wille, M. Toril, and S. Luna-Ramirez, “Estimating pole capacity in a live HSDPA network,” *IEEE Communications Letters*, vol. 17, no. 6, pp. 1260–1263, 2013.
- [23] J. A. Fernández-Segovia, S. Luna-Ramírez, M. Toril, and J. J. Sánchez-Sánchez, “Estimating cell capacity from network measurements in a multi-service LTE system,” *IEEE Communications Letters*, vol. 19, no. 3, pp. 431–434, 2015.
- [24] D. Parracho, D. Duarte, I. Pinto, and P. Vieira, “An improved capacity model based on radio measurements for a 4G and beyond wireless network,” in *21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2018, pp. 314–318.
- [25] T. ur Rehman, M. A. I. Baig, and A. Ahmad, “LTE downlink throughput modeling using neural networks,” in *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017, pp. 265–270.
- [26] K. Lee, S. Kim, S. Lee, and J. Ma, “Load balancing with transmission power control in femtocell networks,” in *13th International Conference on Advanced Communication Technology (ICACT2011)*. IEEE, 2011, pp. 519–522.
- [27] S. Musleh, M. Ismail, and R. Nordin, “Load balancing models based on reinforcement learning for self-optimized macro-femto LTE-advanced heterogeneous network,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 1, pp. 47–54, 2017.
- [28] I. Siomina and D. Yuan, “Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization,” in *IEEE International Conference on Communications (ICC)*, 2012, pp. 1357–1361.

- [29] M. Malmirchegini, M. Shukair, P. Rached, S. Sawhney, M. Ambriss, K. R. Chaudhuri, and S. Sarkar, “Layer management through idle-mode parameter optimization in multi-carrier LTE networks,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2016, pp. 1–6.
- [30] R. Kwan, R. Arnott, R. Paterson, R. Trivisonno, and M. Kubota, “On mobility load balancing for LTE systems,” in *IEEE 72nd Vehicular Technology Conference Fall (VTC-2010-Fall)*, 2010, pp. 1–5.
- [31] P. Muñoz, R. Barco, D. Laselva, and P. Mogensen, “Mobility-based strategies for traffic steering in heterogeneous networks,” *IEEE Communications Magazine*, vol. 51, no. 5, pp. 54–62, 2013.
- [32] M. M. Hasan, S. Kwon, and J.-H. Na, “Adaptive Mobility Load Balancing Algorithm for LTE Small-Cell Networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2205–2217, 2018.
- [33] M. L. Marí-Altozano, S. Luna-Ramírez, M. Toril, and C. Gijón, “A QoE-driven traffic steering algorithm for LTE networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 271–11 282, 2019.
- [34] M. L. M. Altozano, M. Toril, S. Luna-Ramírez, and C. Gijón, “A self-tuning algorithm for optimal QoE-driven traffic steering in LTE,” *IEEE Access*, vol. 8, pp. 156 707–156 717, 2020.
- [35] ITU-R, “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,” in *Recommendation M.2083*, 2015.
- [36] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, “A survey of payload-based traffic classification approaches,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.
- [37] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, *The elements of statistical learning: data mining, inference and prediction, Second edition*. Springer series in statistics, 2001.
- [38] I. Muhammad and Z. Yan, “Supervised Machine Learning Approaches: a Survey,” *ICTACT Journal on Soft Computing*, vol. 5, no. 3, 2015.
- [39] M. Awad and R. Khanna, “Support vector regression,” in *Efficient learning machines*. Springer, 2015, pp. 67–80.

- [40] L. Rokach, “Ensemble methods in supervised learning,” in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 959–979.
- [41] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [42] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [43] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM. ACM, 2016, pp. 785–794.
- [44] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [45] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- [46] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [47] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” in *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, 2020, pp. 310–316.
- [48] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory, 1957.
- [49] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: LSTM cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [50] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

- [52] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [53] L. Prechelt, “Early stopping – but when?” in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.
- [54] L. Rokach and O. Maimon, “Clustering methods,” in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.
- [55] W. H. Day and H. Edelsbrunner, “Efficient algorithms for agglomerative hierarchical clustering methods,” *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.
- [56] T. Velmurugan and T. Santhanam, “A survey of partition based clustering algorithms in data mining: An experimental approach,” *Information Technology Journal*, vol. 10, no. 3, pp. 478–484, 2011.
- [57] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [58] D. A. Reynolds, “Gaussian mixture models,” *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.
- [59] C.-H. Cheng, A. W. Fu, and Y. Zhang, “Entropy-based subspace clustering for mining numerical data,” in *Proc. of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 84–93.
- [60] T. Kohonen, *Self-organization and associative memory*. Springer Science & Business Media, 2012, vol. 8.
- [61] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.
- [62] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3 (March), pp. 1157–1182, 2003.
- [63] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *2014 science and information conference*, 2014, pp. 372–378.



- [64] T. A. Kumbhare and S. V. Chobe, “An overview of association rule mining algorithms,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014.
- [65] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [66] J. Ramiro and K. Hamied, *Self-organizing networks: self-planning, self-optimization and self-healing for GSM, UMTS and LTE*. John Wiley & Sons, 2011.
- [67] J. L. Bejarano-Luque, M. Toril, M. Fernández-Navarro, A. J. García, and S. Luna-Ramírez, “A context-aware data-driven algorithm for small cell site selection in cellular networks,” *IEEE Access*, vol. 8, pp. 105 335–105 350, 2020.
- [68] V. Buenestado, M. Toril, S. Luna-Ramírez, and J. M. Ruiz-Aviles, “Self-planning of base station transmit power for coverage and capacity optimization in LTE,” *Mobile Information Systems*, vol. 2017, 2017.
- [69] P. A. S. Ordóñez, S. Luna-Ramírez, and M. Toril, “A Computationally Efficient Method for QoE-Driven Self-Planning of Antenna Tilts in a LTE Network,” *IEEE Access*, vol. 8, pp. 197 005–197 016, 2020.
- [70] R. Acedo-Hernández, M. Toril, S. Luna-Ramírez, and C. Ubeda, “A PCI planning algorithm for jointly reducing reference signal collisions in LTE uplink and downlink,” *Computer Networks*, vol. 119, pp. 112–123, 2017.
- [71] J. Á. Fernández-Segovia, S. Luna-Ramírez, M. Toril, A. B. Vallejo-Mora, and C. Úbeda, “A computationally efficient method for self-planning uplink power control parameters in LTE,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1–13, 2015.
- [72] V. Buenestado, M. Toril, S. Luna-Ramírez, J. M. Ruiz-Avilés, and A. Mendo, “Self-tuning of remote electrical tilts based on call traces for coverage and capacity optimization in LTE,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4315–4326, 2016.
- [73] M. L. Mari-Altozano, S. S. Mwanje, S. Luna-Ramírez, M. Toril, H. Sanneck, and C. Gijón, “A service-centric Q-learning algorithm for mobility robustness optimization in LTE,” *IEEE Transactions on Network and Service Management*, 2021.

- [74] B. Soret, A. De Domenico, S. Bazzi, N. H. Mahmood, and K. I. Pedersen, “Interference coordination for 5G new radio,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 131–137, 2017.
- [75] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramírez, and J. M. Ruiz Aviles, “Self-tuning of scheduling parameters for balancing the quality of experience among services in lte,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–12, 2016.
- [76] J. Ferragut and J. Mangues-Bafalluy, “A self-organized tracking area list mechanism for large-scale networks of femtocells,” in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 5129–5134.
- [77] C. Gijón, M. Toril, S. Luna-Ramírez, and M. L. Marí-Altozano, “A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414–9424, 2019.
- [78] M. Alias, N. Saxena, and A. Roy, “Efficient cell outage detection in 5G HetNets using hidden Markov model,” *IEEE Communications Letters*, vol. 20, no. 3, pp. 562–565, 2016.
- [79] M. Selim, A. E. Kamal, K. Elsayed, H. M. Abdel-Atty, and M. Alnuem, “Fronthaul cell outage compensation for 5G networks,” *IEEE Communications Magazine*, vol. 54, no. 8, pp. 169–175, 2016.
- [80] H. Fattah and C. Leung, “An overview of scheduling algorithms in wireless multimedia networks,” *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, 2002.
- [81] E. Onur, H. Deliç, C. Ersoy, and M. U. Çağlayan, “Measurement-based re-planning of cell capacities in GSM networks,” *Computer Networks*, vol. 39, no. 6, pp. 749–767, 2002.
- [82] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, “A mobile network planning tool based on data analytics,” *Mobile Information Systems*, vol. 2017, 2017.
- [83] Net2Plan, “The open source network planner,” Available in: <http://www.net2plan.com/index.php>, online. Accessed: Sep 12, 2019.

- [84] 3GPP, “Evolved Universal Terrestrial Radio Access (EUTRA); Radio Resource Control; Protocol specification,” in *TS 36.331*, version 15.3.0, 2018.
- [85] —, “New Radio (NR); Radio Resource Control (RRC) protocol specification,” in *TS 38.331*, version 17.1.0, 2022.
- [86] J. M. Ruiz-Avilés, S. Luna-Ramírez, M. Toril, and F. Ruiz, “Traffic steering by self-tuning controllers in enterprise LTE femtocells,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 337, 2012.
- [87] J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramírez, V. Buenestado, and M. Regueira, “Analysis of limitations of mobility load balancing in a live LTE system,” *IEEE wireless communications letters*, vol. 4, no. 4, pp. 417–420, 2015.
- [88] T. Taleb, R. L. Aguiar, I. Grida Ben Yahia, B. Chatras, G. Christensen, U. Chunduri, A. Clemm, X. Costa, L. Dong, J. Elmirghani *et al.*, “White paper on 6G networking,” *6G research visions*, 2020.
- [89] M. U. Khan, A. García-Armada, and J. Escudero-Garzás, “Service-Based Network Dimensioning for 5G Networks Assisted by Real Data,” *IEEE Access*, vol. 8, pp. 129 193–129 212, 2020.
- [90] K. Attiah, K. Banawan, A. Gaber, A. Elezabi, K. Seddik, Y. Gadallah, and K. Abdullah, “Load balancing in cellular networks: A reinforcement learning approach,” in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–6.
- [91] G. Alshuhli, K. Banawan, K. Attiah, A. Elezabi, K. Seddik, A. Gaber, M. Zaki, and Y. Gadallah, “Mobility load management in cellular networks: A deep reinforcement learning approach,” *IEEE Transactions on Mobile Computing*, 2021.
- [92] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapé, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.
- [93] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapé, “MIMETIC: Mobile encrypted traffic classification using multimodal deep learning,” *Computer Networks*, vol. 165, p. 106944, 2019.

- [94] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, “Big data empowered self organized networks,” in *European Wireless 2014; 20th European Wireless Conference*. VDE, 2014, pp. 1–8.
- [95] 3GPP, “Configuration Management (CM); Concept and high-level requirements.” in *TS 32.600*, version 17.0.0, 2022.
- [96] —, “Performance Management (PM); Concept and requirements.” in *TS 32.421*, version 17.0.0, 2022.
- [97] A. J. García, M. Toril, P. Oliver, S. Luna-Ramírez, and M. Ortiz, “Automatic alarm prioritization by data mining for fault management in cellular networks,” *Expert Systems with Applications*, vol. 158, p. 113526, 2020.
- [98] 3GPP, “Subscriber and Equipment Trace: Trace Data Definition and Management,” in *TS 32.423*, version 15.0.0, 2018.
- [99] —, “Subscriber and Equipment Trace: Trace Concepts and Requirements,” in *TS 32.421*, version 15.0.0, 2018.
- [100] I. de-la Bandera, M. Toril, S. Luna-Ramírez, V. Buenestado, and J. M. Ruiz-Avilés, “Complex Event Processing for Self-Optimizing Cellular Networks,” *Sensors (MDPI)*, vol. 20, no. 7, p. 1937, 2020.
- [101] W. A. Hapsari, A. Umesh, M. Iwamura, M. Tomala, B. Gyula, and B. Sebire, “Minimization of drive tests solution in 3GPP,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 28–36, 2012.
- [102] M. Toril, R. Acedo-Hernández, A. Sánchez, S. Luna-Ramírez, and C. Úbeda, “Estimating Spectral Efficiency Curves from Connection Traces in a Live LTE Network,” *Mobile Information Systems*, vol. 2017, 2017.
- [103] A. Sánchez, R. Acedo-Hernández, M. Toril, S. Luna-Ramírez, and C. Úbeda, “A trace data-based approach for an accurate estimation of precise utilization maps in LTE,” *Mobile Information Systems*, vol. 2017, 2017.
- [104] V. Buenestado, J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramírez, and A. Mendo, “Analysis of throughput performance statistics for benchmarking lte networks,” *IEEE Communications letters*, vol. 18, no. 9, pp. 1607–1610, 2014.

- [105] A. Durán, M. Toril, F. Ruiz, and A. Mendo, “Self-optimization algorithm for outer loop link adaptation in LTE,” *IEEE Communications Letters*, vol. 19, no. 11, pp. 2005–2008, 2015.
- [106] A. Gomez-Andrades, R. Barco, I. Serrano, P. Delgado, P. Caro-Oliver, and P. Munoz, “Automatic root cause analysis based on traces for LTE self-organizing networks,” *IEEE Wireless Communications*, vol. 23, no. 3, pp. 20–28, 2016.
- [107] Ericsson and A. D. Little, “Network slicing: A go-to-market guide to capture the high revenue potential,” *White paper*, 2021.
- [108] 3GPP, “Management and orchestration; Concepts, use cases and requirements,” in *TS 28.530*, version 17.1.0, 2021.
- [109] F. Javed, K. Antevski, J. Manges-Bafalluy, L. Giupponi, and C. J. Bernardos, “Distributed Ledger Technologies For Network Slicing: A Survey,” *IEEE Access*, vol. 10, pp. 19 412–19 442, 2022.
- [110] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, “5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges,” *Computer Networks*, vol. 167, p. 106984, 2020.
- [111] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, “A comprehensive survey on the E2E 5G network slicing model,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 49–62, 2020.
- [112] M. O. Ojijo and O. E. Falowo, “A survey on slice admission control strategies and optimization schemes in 5G network,” *IEEE Access*, vol. 8, pp. 14 977–14 990, 2020.
- [113] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, “Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models,” *IEEE Network*, vol. 33, no. 6, pp. 172–179, 2019.
- [114] R. Martínez, L. Vettori, J. Baranda, J. Manges-Bafalluy, E. Zeydan, and B. Bakhshi, “Resource Abstractions in NFV Management and Orchestration: Experimental Evaluation,” *IEEE Transactions on Network and Service Management*, 2022.

- [115] C. Casetti, C. F. Chiasserini, S. Marcato, C. Puligheddu, J. Mangues-Bafalluy, J. Baranda, J. Brenes, F. Bocchi, G. Landi, and B. Bakhshi, “ML-Driven Provisioning and Management of Vertical Services in Automated Cellular Networks,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2017–2033, 2022.
- [116] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, “On radio access network slicing from a radio resource management perspective,” *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.
- [117] B. Han and H. D. Schotten, “Machine learning for network slicing resource management: a comprehensive survey,” *arXiv preprint arXiv:2001.07974*, 2020.
- [118] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
- [119] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, “5G RAN slicing for verticals: Enablers and challenges,” *IEEE Communications Magazine*, vol. 57, no. 1, pp. 28–34, 2019.
- [120] J. Pérez-Romero, O. Sallent, R. Ferrús, and R. Agustí, “Profit-based radio access network slicing for multi-tenant 5G networks,” in *2019 European Conference on Networks and Communications (EuCNC)*. IEEE, 2019, pp. 603–608.
- [121] 3GPP, “Policy and charging control architecture,” in *TS 23.203*, version 15.4.0, 2018.
- [122] —, “System architecture for the 5G system; stage 2,” in *TS 23.203*, version 17.5.0, 2022.
- [123] “Internet assigned numbers authority (IANA),” Available in: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>, online. Accessed: Jul 2, 2020.
- [124] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, “Transport layer identification of P2P traffic,” in *Proc. of the 4th ACM SIGCOMM conference on Internet measurement*, 2004, pp. 121–134.

- [125] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, “Independent comparison of popular DPI tools for traffic classification,” *Computer Networks*, vol. 76, pp. 75–89, 2015.
- [126] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, “Traffic classification on the fly,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.
- [127] Y. Liu, J. Chen, P. Chang, and X. Yun, “A novel algorithm for encrypted traffic classification based on sliding window of flow’s first N packets,” in *2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*. IEEE, 2017, pp. 463–470.
- [128] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, “Who do you sync you are? Smartphone fingerprinting via application behaviour,” in *Proc. of the sixth ACM conference on Security and privacy in wireless and mobile networks*, 2013, pp. 7–12.
- [129] I.-C. Hsieh, L.-P. Tung, and B.-S. P. Lin, “On the classification of mobile broadband applications,” in *IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2016, pp. 128–134.
- [130] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, “Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic,” in *2016 IEEE European Symposium on Security and Privacy (EuroSecP)*, 2016, pp. 439–454.
- [131] —, “Robust smartphone app identification via encrypted network traffic analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 63–78, 2017.
- [132] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapé, “Multi-classification approaches for classifying mobile app traffic,” *Journal of Network and Computer Applications*, vol. 103, pp. 131–145, 2018.
- [133] P. Wang, X. Chen, F. Ye, and Z. Sun, “A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning,” *IEEE Access*, vol. 7, pp. 54 024–54 033, 2019.

- [134] D. Li, Y. Zhu, and W. Lin, “Traffic identification of mobile apps based on variational autoencoder network,” in *13th International Conference on Computational Intelligence and Security (CIS)*, 2017, pp. 287–291.
- [135] A. Nakao and P. Du, “Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing,” *IEICE Transactions on Communications*, pp. 1536–1543, 2018.
- [136] B. Saltaformaggio, H. Choi, K. Johnson, Y. Kwon, Q. Zhang, X. Zhang, D. Xu, and J. Qian, “Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic,” in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, 2016.
- [137] J. Erman, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms,” in *Proc. of the 2006 SIGCOMM workshop on Mining network data*. ACM, 2006, pp. 281–286.
- [138] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, “An in-depth study of LTE: effect of network protocol and application behavior on performance,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 363–374, 2013.
- [139] K. R. Fall and W. R. Stevens, *TCP/IP illustrated, volume 1: The protocols*. Addison-Wesley, 1994.
- [140] S. Sesia, M. Baker, and I. Toufik, *LTE - the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.
- [141] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, “The QUIC transport protocol: Design and internet-scale deployment,” in *Proc. of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.
- [142] M. Kubat, S. Matwin *et al.*, “Addressing the curse of imbalanced training sets: one-sided selection,” in *Proc. of the 14th International Conference of Machine Learning (ICML)*, 1997, pp. 179–186.
- [143] Amazon’s Alexa, “The top 500 sites on the web,” Available in: <https://www.alexa.com/topsites>, online. Accessed: Jun 30, 2020.



- [144] “WebPageTest tool,” Available in: <https://www.webpagetest.org/>, online. Accessed: Jun 30, 2020.
- [145] A. Schwind, F. Wamser, T. Gensler, P. Tran-Gia, M. Seufert, and P. Casas, “Streaming characteristics of Spotify sessions,” in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018, pp. 1–6.
- [146] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Analysis and modelling of Youtube traffic,” *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012.
- [147] S. E. Coull and K. P. Dyer, “Traffic analysis of encrypted messaging services: Apple imessage and beyond,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 5–11, 2014.
- [148] U. Acer, A. Mashhadi, C. Forlivesi, and F. Kawsar, “Energy efficient scheduling for mobile push notifications,” in *Proc. of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 2015, pp. 100–109.
- [149] H.-P. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, pp. 1–58, 2009.
- [150] M. Steinbach, L. Ertöz, and V. Kumar, “The challenges of clustering high dimensional data,” pp. 273–309, 2004.
- [151] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [152] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [153] T. Caliński and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
- [154] Google, “HTTPS encryption on the web,” Available in: <https://transparencyreport.google.com/https/overview>, online. Accessed: Jun 10, 2020.

- [155] L. Hubert, H.-F. Köhn, and D. Steinley, “Cluster analysis: a toolbox for MATLAB,” *The SAGE handbook of quantitative methods in psychology*, pp. 444–512, 2009.
- [156] J. H. Ward Jr, “Hierarchical grouping to optimize an objective function,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
- [157] Ericsson, “Mobile traffic by application type in 2016,” Available in: <https://www.ericsson.com/TET/trafficView/loadBasicEditor.ericsson>, online. Accessed: Jul 10, 2020.
- [158] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, 2013.
- [159] Y. Fu, H. Xiong, X. Lu, J. Yang, and C. Chen, “Service usage classification with encrypted internet traffic in mobile messaging apps,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2851–2864, 2016.
- [160] Youtube, “Live encoder settings, bitrates, and resolutions,” Available in: <https://support.google.com/youtube/answer/2853702?hl=en>, online. Accessed: Jun 15, 2020.
- [161] L. R. Jiménez, M. Solera, M. Toril, C. Gijón, and P. Casas, “Content matters: Clustering web pages for QoE analysis with WebCLUST,” *IEEE Access*, vol. 9, pp. 123 873–123 888, 2021.
- [162] B. Sas, E. Bernal-Mor, K. Spaey, V. Pla, C. Blondia, and J. Martinez-Bauset, “Modelling the time-varying cell capacity in LTE networks,” *Telecommunication Systems*, vol. 55, no. 2, pp. 299–313, 2014.
- [163] G. Aceto, F. Palumbo, V. Persico, and A. Pescapé, “Available bandwidth vs. achievable throughput measurements in 4G mobile networks,” in *14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 125–133.
- [164] A. S. Khatouni, M. Mellia, M. A. Marsan, S. Alfredsson, J. Karlsson, A. Brunstrom, O. Alay, A. Lutu, C. Midoglu, and V. Mancuso, “Speedtest-like measurements in 3G/4G networks: The MONROE experience,” in *29th International Teletraffic Congress (ITC 29)*, vol. 1, 2017, pp. 169–177.

- [165] R. Senapati and H. K. Pati, “VoLTE cell capacity estimation using AMR-WB codec,” in *International Conference on Advances in Computing, Communications and Informatics (ICACCI 2018)*, 2018, pp. 1885–1889.
- [166] D. Parracho, D. Duarte, I. Pinto, and P. Vieira, “An enhanced capacity model based on network measurements for a multi-service 3G system,” in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 203–208.
- [167] M. Assaad and D. Zeglache, “On the capacity of HSDPA,” in *IEEE Global Telecommunications Conference (GLOBECOM'03)*, vol. 1. IEEE, 2003, pp. 60–64.
- [168] O. Østerbø, “Scheduling and capacity estimation in LTE,” in *23rd International Teletraffic Congress (ITC 2011)*, 2011, pp. 63–70.
- [169] C. Dou and Y.-H. Chang, “Class-based downlink capacity estimation of a WCDMA network in a multiservice context,” *Computer Communications*, vol. 28, no. 12, pp. 1443–1455, 2005.
- [170] K. Ivanov, C. Ball, and F. Treml, “GPRS/EDGE performance on reserved and shared packet data channels,” in *IEEE 58th Vehicular Technology Conference (VTC 2003-Fall)*, vol. 2, 2003, pp. 912–916.
- [171] K. I. Pedersen, F. Frederiksen, T. E. Kolding, T. F. Lootsma, and P. E. Mogensen, “Performance of high-speed downlink packet access in coexistence with dedicated channels,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 3, pp. 1262–1271, 2007.
- [172] K. Kousias, Ö. Alay, A. Argyriou, A. Lutu, and M. Riegler, “Estimating downlink throughput from end-user measurements in mobile broadband networks,” in *IEEE 20th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*, 2019, pp. 1–10.
- [173] K. Chang and R. P. Wicaksono, “Estimation of network load and downlink throughput using RF scanner data for LTE networks,” in *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*. IEEE, 2017, pp. 1–8.
- [174] P. J. M. Johansson and Y.-S. Chen, “Location for minimization of drive test in LTE systems,” 2014, US Patent 8,903,420.

- [175] O. Adamuz-Hinojosa, P. Ameigeiras, P. Munoz, and J. M. Lopez-Soler, “Analytical Model for the UE Blocking Probability in an OFDMA Cell providing GBR Slices,” in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–7.
- [176] P. Muñoz, O. Sallent, and J. Pérez-Romero, “Self-dimensioning and planning of small cell capacity in multitenant 5G networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4552–4564, 2018.
- [177] I. Vilà, J. Pérez-Romero, O. Sallent, and A. Umberto, “A novel approach for dynamic capacity sharing in multi-tenant scenarios,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–6.
- [178] S. Matoussi, I. Fajjari, N. Aitsaadi, and R. Langar, “Deep Learning based User Slice Allocation in 5G Radio Access Networks,” in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 286–296.
- [179] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed, W. Ameen, and T. R. Gadekallu, “Optimal 5G network slicing using machine learning and deep learning concepts,” *Computer Standards & Interfaces*, vol. 76, p. 103518, 2021.
- [180] C. Baena, S. Fortes, E. Baena, and R. Barco, “Estimation of video streaming KQIs for radio access negotiation in network slicing scenarios,” *IEEE Communications Letters*, vol. 24, no. 6, pp. 1304–1307, 2020.
- [181] H. Wang, Y. Wu, G. Min, and W. Miao, “A graph neural network-based digital twin for network slicing management,” *IEEE Transactions on Industrial Informatics*, 2020.
- [182] Ericsson, “Ericsson Mobility Report,” *Technical report*, Jun. 2020.
- [183] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, “On the potential of ensemble regression techniques for future mobile network planning,” in *IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 477–483.
- [184] M. Claesen and B. De Moor, “Hyperparameter search in machine learning,” *arXiv preprint arXiv:1502.02127*, 2015.

- [185] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [186] F. Chollet *et al.*, “Keras: Deep learning library for theano and tensorflow,” Available in: <https://keras.io>, online. Accessed: Jun 12, 2020.
- [187] J. Grus, *Data science from scratch: first principles with python*. O’Reilly Media, 2019.
- [188] P. Sedgwick, “Pearson’s correlation coefficient,” *BMJ*, vol. 345, 2012.
- [189] 3GPP, “New Radio (NR); Physical Layer procedures for data,” in *TS 38.214*, version 16.6.0, 2021.
- [190] Y. Shu, M. Yu, O. Yang, J. Liu, and H. Feng, “Wireless traffic modeling and prediction using seasonal ARIMA models,” *IEICE transactions on communications*, vol. 88, no. 10, pp. 3992–3999, 2005.
- [191] Y. Yu, J. Wang, M. Song, and J. Song, “Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient,” in *2010 International Conference on Intelligent System Design and Engineering Application*, vol. 1, 2010, pp. 980–983.
- [192] D. Tikunov and T. Nishimura, “Traffic prediction for mobile network using Holt-Winter’s exponential smoothing,” in *2007 15th International Conference on Software, Telecommunications and Computer Networks*, 2007, pp. 1–5.
- [193] J. Bastos, “Forecasting the capacity of mobile networks,” *Telecommunication Systems (Springer)*, vol. 72, no. 2, pp. 231–242, 2019.
- [194] B. Zhou, D. He, and Z. Sun, “Traffic modeling and prediction using AR-IMA/GARCH model,” in *Modeling and Simulation Tools for Emerging Telecommunication Networks*. Springer, 2006, pp. 101–121.
- [195] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, “The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 234–240, 2014.
- [196] M. D. Jnr, J. D. Gadze, and D. K. Anipa, “Short-term traffic volume prediction in UMTS networks using the Kalman filter algorithm,” *Int. J. Mobile Netw. Commun. Telemat.*, vol. 3, no. 6, pp. 31–40, 2013.

- [197] N. Bui and J. Widmer, “Data-driven evaluation of anticipatory networking in LTE networks,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2252–2265, 2018.
- [198] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, “The learning and prediction of application-level traffic data in cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3899–3912, 2017.
- [199] C.-W. Huang, C.-T. Chiang, and Q. Li, “A study of deep learning networks on mobile traffic forecasting,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.
- [200] Y. Hua, Z. Zhao, Z. Liu, X. Chen, R. Li, and H. Zhang, “Traffic prediction based on random connectivity in deep learning with long short-term memory,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–6.
- [201] H. D. Trinh, L. Giupponi, and P. Dini, “Mobile traffic prediction from raw data using LSTM networks,” in *IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1827–1832.
- [202] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, “DeepTP: An end-to-end neural network for mobile cellular traffic prediction,” *IEEE Network*, vol. 32, no. 6, pp. 108–115, 2018.
- [203] L. Nie, D. Jiang, S. Yu, and H. Song, “Network traffic prediction based on deep belief network in wireless mesh backbone networks,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–5.
- [204] H. Assem, B. Caglayan, T. S. Buda, and D. O’Sullivan, “ST-DenNetFus: A new deep learning approach for network demand prediction,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 222–237.
- [205] C. Zhang and P. Patras, “Long-term mobile traffic forecasting using deep spatio-temporal neural networks,” in *Proc. of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 231–240.
- [206] L. Fang, X. Cheng, H. Wang, and L. Yang, “Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3091–3101, 2018.

- [207] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, “Spatio-temporal analysis and prediction of cellular traffic in metropolis,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2018.
- [208] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, 2017, pp. 1–9.
- [209] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-temporal wireless traffic prediction with recurrent neural network,” *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 554–557, 2018.
- [210] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, “Citywide cellular traffic prediction based on densely connected convolutional neural networks,” *IEEE Communications Letters*, vol. 22, no. 8, pp. 1656–1659, 2018.
- [211] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, “A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1790–1821, 2017.
- [212] A. R. Mishra, *Fundamentals of network planning and optimisation 2G/3G/4G: evolution to 5G*, 2nd ed. John Wiley & Sons, 2018.
- [213] D. Grillo, R. A. Skoog, S. Chia, and K. K. Leung, “Teletraffic engineering for mobile personal communications in ITU-T work: The need to match practice and theory,” *IEEE Personal Communications*, vol. 5, no. 6, pp. 38–58, 1998.
- [214] B. S. Northcote and N. A. Tompson, “Dimensioning Telstra’s WCDMA (3G) network,” in *2010 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD)*, 2010, pp. 81–85.
- [215] E. Jailani, M. Ibrahim, and R. Ab Rahman, “LTE speech traffic estimation for network dimensioning,” in *2012 IEEE Symposium on Wireless Technology and Applications (ISWTA)*, 2012, pp. 315–320.
- [216] M. Toril, S. Luna-Ramírez, and V. Wille, “Automatic replanning of tracking areas in cellular networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2005–2013, 2013.

- [217] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, “Big data driven mobile traffic understanding and forecasting: A time series approach,” *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 796–805, 2016.
- [218] R. J. Hyndman and A. V. Kostenko, “Minimum sample size requirements for seasonal forecasting models,” *Foresight*, vol. 6, no. Spring, pp. 12–15, 2007.
- [219] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [220] P. R. Winters, “Forecasting sales by exponentially weighted moving averages,” *Management science*, vol. 6, no. 3, pp. 324–342, 1960.
- [221] C.-K. Ing, “Multistep prediction in autoregressive processes,” *Econometric theory*, vol. 19, no. 2, pp. 254–279, 2003.
- [222] Z. C. Lipton, “The mythos of model interpretability,” *Queue*, vol. 16, no. 3, pp. 30:31–30:57, 2018.
- [223] IBM, “IBM SPSS modeler 18.1.1 User’s guide,” 2017.
- [224] R. A. Yaffee and M. McGee, *An introduction to time series analysis and forecasting: with applications of SAS® and SPSS®*. Elsevier, 2000.
- [225] B. Devi, K. Rao, S. Setty, and M. Rao, “Disaster prediction system using ibm spss data mining tool,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, pp. 3352–3357, 2013.
- [226] X.-H. Meng, Y.-X. Huang, D.-P. Rao, Q. Zhang, and Q. Liu, “Comparison of three data mining models for predicting diabetes or prediabetes by risk factors,” *The Kaohsiung Journal of Medical Sciences*, vol. 29, no. 2, pp. 93–99, 2013.
- [227] IBM, “IBM SPSS modeler 18.1 Algorithms guide,” 2017.
- [228] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, “Load balancing in downlink LTE self-optimizing networks,” in *IEEE 71st Vehicular Technology Conference (VTC-2010-Spring)*, 2010, pp. 1–5.
- [229] E. Gures, I. Shayea, M. Ergen, M. H. Azmi, and A. A. El-Saleh, “Machine Learning Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey,” *IEEE Access*, 2022.



- [230] C. A. S. Franco and J. R. B. de Marca, "Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach," in *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2015, pp. 1–5.
- [231] P. Muñoz, R. Barco, J. M. Ruiz-Avilés, I. De La Bandera, and A. Aguilar, "Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE femtocells," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1962–1973, 2012.
- [232] P. Muñoz, R. Barco, and I. de la Bandera, "Optimization of load balancing using fuzzy Q-learning for next generation wireless networks," *Expert Systems with Applications*, vol. 40, no. 4, pp. 984–994, 2013.
- [233] S. S. Mwanje and A. Mitschele-Thiel, "A Q-learning strategy for LTE mobility load balancing," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2013, pp. 2154–2158.
- [234] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in HetNets: Old myths and open problems," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 18–25, 2014.
- [235] W.-R. Lai, Y.-B. Lin, and H. C.-H. Rao, "Analysis and modeling of dual-band GSM networks," *Journal of Communications and Networks*, vol. 1, no. 3, pp. 158–165, 1999.
- [236] X. Chen and R. Q. Hu, "Joint uplink and downlink optimal mobile association in a wireless heterogeneous network," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 4131–4137.
- [237] P. Muñoz, D. Laselva, R. Barco, and P. Mogensen, "Adjustment of mobility parameters for traffic steering in multi-RAT multi-layer wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 133, 2013.
- [238] L. Zhang, Y. Liu, M. Zhang, S. Jia, and X. Duan, "A two-layer mobility load balancing in LTE self-organization networks," in *IEEE 13th International Conference on Communication Technology*, 2011, pp. 925–929.

- [239] R. Fang, G. Chuai, and W. Gao, “Improve quality of experience of users by optimizing handover parameters in mobile networks,” in *Proc. of the 4th International Conference on Computer Science and Application Engineering*, 2020, pp. 1–7.
- [240] P. E. Iturria-Rivera and M. Erol-Kantarci, “QoS-Aware Load Balancing in Wireless Networks using Clipped Double Q-Learning,” in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 10–16.
- [241] —, “Competitive Multi-Agent Load Balancing with Adaptive Policies in Wireless Networks,” in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 796–801.
- [242] L. C. Gimenez, I. Z. Kovács, J. Wigard, and K. I. Pedersen, “Throughput-based traffic steering in LTE-Advanced HetNet deployments,” in *IEEE 82nd Vehicular Technology Conference (VTC-2015-Fall)*, 2015, pp. 1–5.
- [243] P. Reichl, S. Egger, R. Schatz, and A. D’Alconzo, “The logarithmic nature of QoE and the role of the Weber-Fechner law in QoE assessment,” in *IEEE International Conference on Communications*, 2010, pp. 1–5.
- [244] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.
- [245] I. Da Silva, G. Mildh, A. Kaloxylou, P. Spapis, E. Buracchini, A. Trogolo, G. Zimmermann, and N. Bayer, “Impact of Network Slicing on 5G Radio Access Networks,” in *2016 European conference on networks and communications (EuCNC)*. IEEE, 2016, pp. 153–157.
- [246] Q. Liao, T. Hu, and D. Wellington, “Knowledge Transfer in Deep Reinforcement Learning for Slice-Aware Mobility Robustness Optimization,” *arXiv preprint arXiv:2203.03227*, 2022.
- [247] 3GPP, “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements,” in *TS 136.214*, version 14.2.0, 2017.
- [248] J. Kurjenniemi, T. Henttonen, and J. Kaikkonen, “Suitability of RSRQ measurement for quality based inter-frequency handover in LTE,” in *IEEE International*

- Symposium on Wireless Communication Systems, ISWCS'08*. IEEE, 2008, pp. 703–707.
- [249] M. Kazmi, O. Sjobergh, W. Muller, J. Wierok, and B. Lindoff, “Evaluation of inter-frequency quality handover criteria in E-UTRAN,” in *IEEE 69th Vehicular Technology Conference (VTC-2009-Spring)*, 2009, pp. 1–5.
- [250] I. Petrut, M. Otesteanu, C. Balint, and G. Budura, “Hetnet handover performance analysis based on RSRP vs. RSRQ triggers,” in *IEEE 38th International Conference on Telecommunications and Signal Processing (TSP)*, 2015, pp. 232–235.
- [251] 3GPP, “Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Mobility enhancements in heterogeneous networks,” in *TS 36.839*, version 11.1.0, 2012.
- [252] C. Gijón, S. Luna-Ramirez, and M. Toril, “Un nuevo criterio basado en calidad de experiencia para el balance de carga en redes LTE,” in *XXXIII Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2018)*. URSI, 2018.
- [253] S. Luna-Ramírez, M. Toril, M. Fernández-Navarro, and V. Wille, “Optimal traffic sharing in GERAN,” *Wireless Personal Communications*, vol. 57, no. 4, pp. 553–574, 2011.
- [254] P. Popovski, J. J. Nielsen, C. Stefanovic, E. De Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park *et al.*, “Wireless access for ultra-reliable low-latency communication: Principles and building blocks,” *IEEE Network*, vol. 32, no. 2, pp. 16–23, 2018.
- [255] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [256] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [257] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv preprint arXiv:1706.05098*, 2017.

- [258] J. L. Bejarano-Luque, M. Toril, M. Fernandez-Navarro, C. Gijon, and S. Luna-Ramirez, “A deep-learning model for estimating the impact of social events on traffic demand on a cell basis,” *IEEE Access*, vol. 9, pp. 71 673–71 686, 2021.
- [259] B. Brik and A. Ksentini, “On predicting service-oriented network slices performances in 5G: A federated learning approach,” in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 164–171.
- [260] Y. Singh, “Comparison of Okumura, Hata and COST-231 models on the basis of path loss and signal strength,” *International journal of computer applications*, vol. 59, no. 11, 2012.
- [261] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception,” in *TS 36.104*, version 15.2.0, 2018.
- [262] J.-H. Rhee, J. M. Holtzman, and D.-K. Kim, “Scheduling of real/non-real time services: adaptive EXP/PF algorithm,” in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003 (VTC-2003-Spring)*, vol. 1. IEEE, 2003, pp. 462–466.
- [263] 3GPP, “New Radio (NR); User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone,” in *TS 38.101-1*, version 17.2.0, 2021.
- [264] M. Abu-Tair and A. Marshall, “An empirical model for multi-contact point haptic network traffic,” in *Proc. of the 2nd International Conference on Immersive Telecommunications*, 2009, pp. 1–6.
- [265] O. Nassef, L. Sequeira, E. Salam, and T. Mahmoodi, “Building a lane merge coordination for connected vehicles using deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2540–2557, 2020.
- [266] ITU-T, “Vocabulary for performance, quality of service and quality of experience,” in *Recommendation P.10/G.100*, 2017.
- [267] R. K. Mok, E. W. Chan, and R. K. Chang, “Measuring the quality of experience of HTTP video streaming,” in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2011, pp. 485–492.
- [268] J. Navarro-Ortiz, J. M. Lopez-Soler, and G. Stea, “Quality of experience based resource sharing in IEEE 802.11 e HCCA,” in *2010 European Wireless Conference (EW)*. IEEE, 2010, pp. 454–461.

- [269] F. Khan, *LTE for 4G mobile broadband: air interface technologies and performance*. Cambridge university press, 2009.
- [270] P. Muñoz, I. de la Bandera, F. Ruiz, S. Luna-Ramírez, R. Barco, M. Toril, P. Lázaro, and J. Rodríguez, “Computationally-efficient design of a dynamic system-level LTE simulator,” *International Journal of Electronics and Telecommunications*, vol. 57, no. 3, pp. 347–358, 2011.
- [271] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, “Versatile mobile communications simulation: The Vienna 5G link level simulator,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–17, 2018.
- [272] K. Brueninghaus, D. Astely, T. Salzer, S. Visuri, A. Alexiou, S. Karger, and G.-A. Seraji, “Link performance models for system level simulations of broadband radio access systems,” in *16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4. IEEE, 2005, pp. 2306–2311.
- [273] S. Monikandan, A. Sivasubramanian, and S. Babu, “A review of MAC scheduling algorithms in LTE system,” *International Journal in Advanced Science, Engineering and Technology*, vol. 3, pp. 1056–1068, 2017.