



Universidad
Politécnica
de Cartagena



Universidad
Politécnica
de Cartagena

Campus
de Excelencia
Internacional

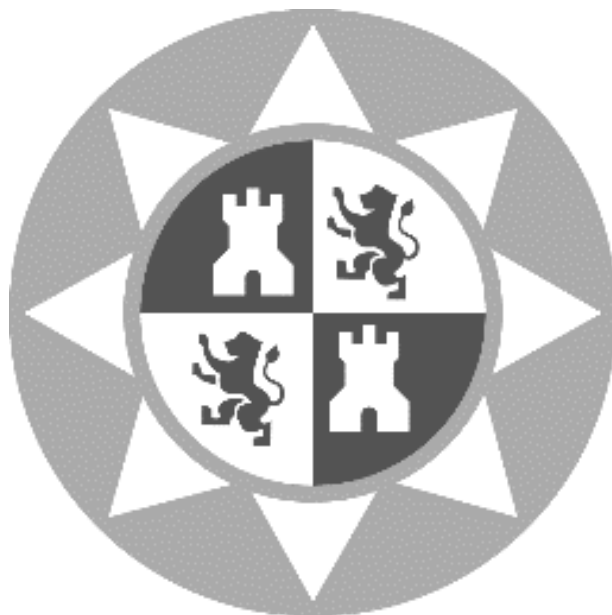
*Contribución al desarrollo de técnicas
avanzadas para la evaluación de prestaciones
en la Internet de las Cosas*

*Tecnologías de la Información y las
Telecomunicaciones*

Autor: José Manuel Martínez Caro

Directora: María Dolores Cano Baños

Cartagena (2020)



Universidad Politécnica de Cartagena
Departamento de Tecnologías de la Información y las comunicaciones

**Contribución al Desarrollo de técnicas avanzadas
para la evaluación de prestaciones en la Internet
de las Cosas**

Tesis doctoral

José M. Martínez Caro

2020

Directora: María Dolores Cano Baños

**CONFORMIDAD DE SOLICITUD DE AUTORIZACIÓN DE DEPÓSITO DE
TESIS DOCTORAL POR EL/LA DIRECTOR/A DE LA TESIS**

D^a. María Dolores Baños Cano Directora de la Tesis doctoral “Contribución al desarrollo de técnicas avanzadas para la evaluación de prestaciones en la Internet de las Cosas”.

INFORMA:

Que la referida Tesis Doctoral, ha sido realizada por D. José Manuel Martínez Caro, dentro del Programa de Doctorado Tecnologías de la información y las comunicaciones, dando mi conformidad para que sea presentada ante el Comité de Dirección de la Escuela Internacional de Doctorado para ser autorizado su depósito.

- Informe positivo sobre el plan de investigación y documento de actividades del doctorando/a emitido por el Director/ Tutor (RAPI).

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

- Ciencias
 Ciencias Sociales y Jurídicas
 Ingeniería y Arquitectura

En Cartagena, a 11 de Septiembre de 2020

LA DIRECTORA DE LA TESIS

MARIA DOLORES
CANO BAÑOS -

NIF: [REDACTED]

Digitally signed by
MARIA DOLORES CANO
BAÑOS - NIF: [REDACTED]

Date: 2020.09.11
10:31:30 +02'00'

Fdo.: María Dolores Cano Baños

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO

CONFORMIDAD DE DEPÓSITO DE TESIS DOCTORAL
POR LA COMISIÓN ACADÉMICA DEL PROGRAMA

D Jorge Larrey Ruiz, Presidente de la Comisión Académica del Programa del Programa de Doctorado Tecnologías de la información y las comunicaciones.

INFORMA:

Que la Tesis Doctoral titulada, “Contribución al desarrollo de técnicas avanzadas para la evaluación de prestaciones en la Internet de las Cosas”, ha sido realizada, dentro del mencionado Programa de Doctorado, por D. José Manuel Martínez Caro, bajo la dirección y supervisión de la Dra. María Dolores Cano Baños.

En reunión de la Comisión Académica, visto que en la misma se acreditan los indicios de calidad correspondientes y la autorización de la Directora de la misma, se acordó dar la conformidad, con la finalidad de que sea autorizado su depósito por el Comité de Dirección de la Escuela Internacional de Doctorado.

- Evaluación positiva del plan de investigación y documento de actividades por el Presidente de la Comisión Académica del programa (RAPI).

La Rama de conocimiento por la que esta tesis ha sido desarrollada es:

- Ciencias
 Ciencias Sociales y Jurídicas
 Ingeniería y Arquitectura

En Cartagena, a 11 de Septiembre de 2020

EL PRESIDENTE DE LA COMISIÓN ACADÉMICA

JORGE|
LARREY|
RUIZ

Firmado
digitalmente por
JORGE|LARREY|RUIZ
Fecha: 2020.09.11
11:04:08 +02'00'

Fdo: Jorge Larrey Ruiz

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO

Agradecimientos

Una tesis doctoral la definiría como la suma de un proceso de especialización profesional junto a un proceso de evolución personal basado en una serie de personas, etapas y experiencias. Gracias a estas variables, nos permiten ser lo que hoy somos y forjar las bases de lo que queremos ser en un futuro.

En primer lugar, debo agradecer a María Dolores Cano por permitirme conocer el mundo de la investigación. Si he llegado aquí es en gran medida gracias a ella, ya que el camino no ha sido nada fácil en estos años aportando ideas y posibles soluciones, no solo al campo profesional, sino también al personal. Espero que este sea el inicio de futuras colaboraciones allá donde estemos.

Agradecer también al Laboratório de Arquitetura e Redes de Computadores (LARC) de la Universidade São Paulo, especialmente a la profesora Graça Bressan que me recibió con los brazos abiertos y me hizo sentir como uno más del equipo de trabajo. También destacar la gran familia cubana formada por José Carlos, Armandito, Patricia, Fernando, Loubrys, Sandra, Mónica y Víctor. Nos volveremos a ver pronto. La adaptación en São Paulo no hubiese sido lo mismo sin ti José Carlos.

Pese a que por motivos del COVID-19 no se pudo realizar la estancia, agradecer también la cooperación con la Military Technical Academy de Bucarest por brindarme la posibilidad de cooperar con ellos, especialmente a Petrica Ciotirnae y Cristian Molder.

A aquellos que por diversas circunstancias ya no están tan cerca.

A mi familia, tías y primos. De forma especial a mis abuelos y abuelas que siempre han hecho lo imposible y han tenido mucha paciencia conmigo. Especialmente a mi “Macarena o Macaria”, mi abuela Isabel, de la que me encanta escuchar sus batallitas de joven y pasar el tiempo viendo “retratos”.

A mis amigos y amigas de Totana, siempre disponibles para cualquier plan que se pueda imaginar. Fuentes de desconexión y risas para poder retomar con las mayores energías en cualquier momento.

A Ana Belén, que llegó sin hacer mucho ruido aportando la energía y contagiando la disciplina que necesitaba para poder finalizar esta etapa.

Y, por último, a mi núcleo familiar: Papá, Gemma y Choni. Habéis aguantado mi carácter y dado todo el cariño del mundo, siendo esta última la clave del éxito. Especialmente a mi padre Pedro, siempre paciente sabiendo que el momento llegaría... y llegó. Simplemente me gustaría seguir creciendo a tu lado para convertirme en la persona que tu eres y la bondad que desprendes. Te quiero mucho padre.

*La constancia es la virtud por la cual todas
las otras virtudes dan fruto.
(Arturo Graf)*

Abstract

The new technology trends aim at technology aggregation, simplifying their use and control, greater integration with the user, and an exponential increase in the number of connected devices. Everything is encompassed under the Internet of Things (IoT) concept on a wide range of applications, such as Industry 4.0 or Smart-Cities, where the relationship with the user is closer. The current trend seeks to provide these devices with cognitive capabilities to learn and act between the physical and social world with minimal human interaction. Traditionally, Quality of Service (QoS) has been used as an objective evaluation metric. The present doctoral thesis proposes a holistic model capable of offering a measurement of the services provided in IoT from metrics based on the cost-benefit domains. The benefit domain is composed by three components, which are Quality of Data (QoD), Quality of Information (QoI), and Quality of user Experience (QoE). The cost domain is made up solely of the Quality Cost (QC) component. These quality components can measure, through the use of different metrics, the performance of a service in different layers of the architecture, being essential for optimization in devices with limited resources. In this context, Low-Power Wide Area Network (LPWAN) technologies such as Long-Range (LoRa) and Long-Range Wide Area Network (LoRaWAN) allow communications over long distances with minimum resource consumption. At the same time, it is a versatile technology since it can be embedded in static or mobile devices such as Unmanned Aerial Vehicles (UAVs). For this reason, LoRa/LoRaWAN and UAVs will be used as case studies. Finally, Artificial Intelligence (AI) techniques have become an extremely useful tool in different environments,

including that of performance evaluation, and above all, for its predictive capacity. For this reason, they will also be a subject of study in this doctoral thesis.

Resumen

Las nuevas tendencias tecnológicas apuntan hacia la agregación de tecnologías simplificando su uso y control, una mayor integración con el usuario, así como un aumento exponencial del número de dispositivos conectados. Todo se engloba bajo el concepto de la Internet de las Cosas (*Internet of Things*, IoT) en torno a un gran abanico de aplicaciones como la Industria 4.0 o las *Smart Cities*, donde el vínculo con el usuario es más estrecho. La tendencia actual pretende dotar estos dispositivos de capacidades cognitivas permitiendo el aprendizaje y la actuación entre el mundo físico y el social con la mínima interacción del ser humano. Tradicionalmente se ha utilizado la Calidad de Servicio (*Quality of Service*, QoS) como métrica de evaluación objetiva. La presente tesis doctoral propone un modelo holístico capaz de ofrecer una medición de los servicios prestados en IoT a partir de métricas basadas en los dominios coste-beneficio. El dominio beneficio está compuesto por tres componentes, que son la Calidad de los Datos (*Quality of Data*, QoD), la Calidad de la Información (*Quality of Information*, QoI) y la Calidad de Experiencia del usuario (*Quality of user Experience*, QoE). El dominio coste queda constituido únicamente por el componente Coste de Calidad (*Quality Cost*, QC). Estas componentes de calidad son capaces de medir, a través del uso de diferentes métricas, el rendimiento de un servicio en diferentes capas de la arquitectura, siendo esenciales en dispositivos con recursos limitados para su optimización. En este contexto, las tecnologías de redes de área amplia de bajo consumo (*Low-Power Wide Area Network*, LPWAN) como *Long-Range* (LoRa) y *Long-Range Wide Area Network* (LoRaWAN) permiten comunicaciones a grandes distancias con mínimo consumo de recursos. A su vez, es una tecnología muy versátil ya que puede ser embebida en dispositivos estáticos o móviles como los

Vehículos Aéreos No Tripulados (*Unmanned Aerial Vehicles*, UAVs). Por este motivo, se usará LoRa/LoRaWAN y los UAV como caso de estudio. Por último, las técnicas de Inteligencia Artificial (*Artificial Intelligence*, AI) se han convertido en una herramienta extremadamente útil en diferentes entornos, incluido el de la evaluación de prestaciones, y sobre todo, por su capacidad predictiva. Por este motivo, serán también tema de estudio en esta tesis doctoral.

Contenidos

Agradecimientos	v
Abstract	viii
Resumen	x
Contenidos	xii
Acrónimos y Abreviaturas	xv
Lista de Tablas	xviii
Lista de Figuras	xix
Introducción	1
1.1. Motivación	1
1.2. Objetivos de la tesis doctoral	5
1.3. Organización de la tesis doctoral	6
1.4. Contribuciones de la tesis doctoral	8
Marco teórico	13
2.1. Introducción	13
2.2. Quality of Service	15
2.3. Quality of Experience	18
2.4. Internet of Things.....	21
2.4.1. Capacidades cognitivas	24
2.5. Low-Power Wide Area Network	26
2.5.1. Long-Range.....	29
2.5.2. Long-Range Wide Area Network	31
2.6. Unmanned Aerial Vehicles	34
2.7. Machine Learning	40
2.7.1. Supervised Learning	41
2.7.2. Unsupervised Learning	42
2.7.3. Reinforcement Learning.....	43
2.8. Conclusión	44
Un Enfoque Innovador para la Evaluación de Rendimiento en IoT	46

3.1.	Introducción	46
3.2.	Estado de la Técnica en Modelos de Prestaciones para IoT	48
3.3.	Nuevo Modelo de Evaluación de Prestaciones	52
3.3.1.	Quality of Data.....	53
3.3.2.	Quality of Information	56
3.3.3.	Quality of user Experience.....	61
3.3.4.	Quality Cost	63
3.4.	Conclusión	65
3.5.	Contribuciones Relacionadas a este Capítulo	65
Caso de Estudio: Evaluación de prestaciones en IoT con LoRa/ LoRaWAN empleando el modelo QoX		66
4.1.	Introducción	66
4.2.	Test-bed.....	68
4.3.	Dispositivos Estáticos	72
4.3.1.	Despliegues	72
4.3.2.	Resultados en escenario estático	76
4.4.	Dispositivos Móviles	87
4.4.1.	Despliegues	88
4.4.2.	Resultados en escenario con movilidad	91
4.5.	Conclusiones	109
4.6.	Contribuciones Relacionadas a este Capítulo	110
Aplicación de Técnicas de Machine Learning en la Evaluación de Prestaciones		111
5.1.	Introducción	111
5.2.	Clasificación y Predicción de Eventos de <i>Stalling</i>	112
5.2.1.	Estado de la Técnica	115
5.2.2.	Test-bed.....	118
5.2.3.	Preprocesado de los datos	121
5.2.4.	Métricas de aprendizaje	124
5.2.5.	Modelo de clasificación basado en <i>threshold</i>	126
5.2.6.	Modelo de predicción de eventos de stalling	131

5.3. Análisis Predictivo en la Evaluación de Rendimiento con el Modelo de Componentes de Calidad QoX.	138
5.3.1. Test-bed.....	140
5.3.2. Preprocesado de datos.....	141
5.3.3. Modelo de predicción y sistema de notificación.....	143
5.3.4. Resultados	144
5.3.5. Estudio comparativo	173
5.4. Conclusión	182
5.5. Contribuciones Relacionadas a este Capítulo	183
Conclusiones y trabajo futuro.....	184
Referencias.....	187

Acrónimos y Abreviaturas

A	<i>Accuracy</i>
AI	<i>Artificial Intelligence</i>
BW	<i>Bandwidth</i>
C	<i>Completeness</i>
CF	<i>Carrier Frequency</i>
CR	<i>Coding Rate</i>
CS	<i>Cognitive Stimulation</i>
CSS	<i>Chirp Spread Spectrum</i>
CIoT	<i>Cognitive Internet of Things</i>
D	<i>Detail</i>
EC	<i>Energy Consumption</i>
FANET	<i>Flying Ad-hoc NETWORK</i>
FP	<i>Flight Plan</i>
FS	<i>Feature Selection</i>
FSPL	<i>Free-Space Path Loss</i>
GM	<i>Gauss-Markov</i>
GW_{Av}	<i>Gateway Availability</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol TV</i>
ISM	<i>Industrial, Science and Medical</i>
ITM	<i>Irregular Terrain Model</i>
ITS	<i>Intelligent Transport System</i>
ITU	<i>International Telecommunication Union</i>

KPI	<i>Key Performance Indicator</i>
LoRa	<i>Long-Range</i>
LoRaWAN	<i>Long-Range Wide Area Network</i>
LPWAN	<i>Low Power Wide Area Network</i>
MANET	<i>Mobile Ad-hoc NETWORK</i>
ML	<i>Machine Learning</i>
MOS	<i>Mean Opinion Score</i>
NRA	<i>National Regulatory Agency</i>
P	<i>Precision</i>
PCA	<i>Principal Component Analysis</i>
PDR	<i>Packet Delivery Rate</i>
Q	<i>Quantity</i>
QoD	<i>Quality of Data</i>
QC	<i>Quality Cost</i>
QoE	<i>Quality of user Experience</i>
QoI	<i>Quality of Information</i>
QoS	<i>Quality of Service</i>
QoSD	<i>Quality of Service Delivered</i>
QoSE	<i>Quality of Service Experienced</i>
QoSR	<i>Quality of Service Required</i>
R	<i>Recall</i>
RD	<i>Random Direction</i>
RFE	<i>Recursive Feature Elimination</i>
RFID	<i>Radio Frequency Identification</i>
RPGM	<i>Reference Point Group Mobility</i>

RW	<i>Random Walk</i>
RWP	<i>Random Waypoint</i>
SF	<i>Spreading Factor</i>
SLA	<i>Service Level Agreement</i>
SRCM	<i>Semi-Random Circular Movement</i>
ST	<i>Smooth Turn</i>
T	<i>Timeliness</i>
Th	<i>Truthfulness</i>
TCP	<i>Transmission Control Protocol</i>
TIC	<i>Tecnologías de la Información y las Comunicaciones</i>
ToA	<i>Time on Air</i>
TP	<i>Transmission Power</i>
UAV	<i>Unmanned Aerial Vehicle</i>
UDP	<i>User Datagram Protocol</i>
U2G	<i>UAV to Ground</i>
U2U	<i>UAV to UAV</i>
V	<i>Validity</i>
VANET	<i>Vehicular Ad-hoc NETWORK</i>
VoIP	<i>Voice over Internet Protocol</i>
WBAN	<i>Wireless Body Area Network</i>
WSN	<i>Wireless Sensor Network</i>

Lista de Tablas

Tabla 1: Equivalencia MOS.....	21
Tabla 2: Comparación de Tecnologías LPWAN [46].....	28
Tabla 3: Comparación de características MANET, VANET y FANET.....	35
Tabla 4: Métricas de evaluación de rendimiento propuestas por Wu et al. [42]. Tabla I en [88].	49
Tabla 5: Métricas de evaluación de rendimiento propuestas por Floris et al. [84]. Tabla II en [88].	50
Tabla 6: Métricas de evaluación de rendimiento propuestas por Li et al. [89]. Tabla III en [88].	51
Tabla 7: Métricas de evaluación de rendimiento propuestas por Ikeda et al. [90]. Tabla IV en [88].	51
Tabla 8: Parámetros de formulación en el modelo Okumura-Hata.	69
Tabla 9: Parámetros de simulación para dispositivos estáticos.	73
Tabla 10: Parámetros de simulación para dispositivos móviles.	91
Tabla 11: Características del entorno emulado.	119
Tabla 12: Características extraídas por cada paquete en el dataset.	122
Tabla 13: Detalles del servidor utilizado para simulación.	136
Tabla 14: Ejemplo del desempeño del modelo de predicción para una captura.	138
Tabla 15: Parámetros de entrada en el modelo planteado por [135].....	176

Lista de Figuras

Fig. 1: Interfaces y acuerdos SLA entre usuarios y proveedores de servicios.....	17
Fig. 2: Dispositivos en redes LoRa: (a) dispositivos LoRa, (b) gateway LoRa, y (c) Servidor. ..	32
Fig. 3: Ejemplo de redes FANET con enlaces U2U y U2G.	36
Fig. 4: Clasificación de los modelos de movilidad [79].	40
Fig. 5: Aprendizaje Supervisado: (a) Clasificación, (b) Regresión.	42
Fig. 6: Aprendizaje no Supervisado.....	43
Fig. 7: Aprendizaje por Refuerzo (Reinforcement Learning).....	44
Fig. 8: Alcance de las diferentes métricas de calidad.	53
Fig. 9: División de celdas en el área de cobertura.	60
Fig. 10: Red LoRa con dispositivos IoT/LoRa y Servidor pasando por Gateway y red.....	71
Fig. 11: Ejemplo de despliegue de un entorno rural.	75
Fig. 12: Ejemplo de despliegue de un entorno suburbano.	75
Fig. 13: Ejemplo de despliegue de un entorno urbano.....	76
Fig. 14: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoS, QoI, QoE y QC propuestas sobre un entorno urbano y nodos fijos en el caso 1.....	78
Fig. 15: Métricas de QoI y QoE detalladas sobre un entorno urbano.....	78
Fig. 16: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoS, QoI, QoE y QC propuestas sobre un entorno suburbano y nodos fijos en el caso 1.	80
Fig. 17: Métricas de QoI y QoE detalladas sobre un entorno suburbano.	80
Fig. 18: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoS, QoI, QoE y QC propuestas sobre un entorno rural y nodos fijos en el caso 1.....	81
Fig. 19: Métricas de QoI y QoE detalladas sobre un entorno rural.	82
Fig. 20: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoS, QoI, QoE y QC propuestas sobre un entorno urbano y nodos fijos en el caso 2.....	83
Fig. 21: Métricas de QoI y QoE detalladas sobre un entorno urbano.....	84

Fig. 22: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno suburbano y nodos fijos en el caso 2.	85
Fig. 23: Métricas de QoI y QoE detalladas sobre un entorno suburbano.	85
Fig. 24: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno rural y nodos fijos en el caso 2.	86
Fig. 25: Métricas de QoI y QoE detalladas sobre un entorno rural.	87
Fig. 26: Despliegue IoT que implementa tecnología de red LoRa en UAVs.	89
Fig. 27: Parámetros específicos de los modelos de movilidad en la simulación.	90
Fig. 28: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RPMG y (b) RW, donde se obtienen los mejores resultados para esta componente QoD.	93
Fig. 29: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RWP y (b) RD, donde se obtienen los peores resultados para esta componente QoD.	94
Fig. 30: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles para el resto de modelos; (a) GM; (b) Pathway; (c) SRCM; y (d) ST.	96
Fig. 31: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoI propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) ST, (b) SRCM, (c) RWP, (d) RW, (e) RPGM, (f) RD, (g) Pathway y (h) GM.	101
Fig. 32: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoE propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RD y (b) RPGM, que obtienen el mejor y el peor resultado para esta componente, respectivamente.	103
Fig. 33: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoE propuesta sobre un entorno suburbano y nodos móviles para el resto de modelos: (a) GM; (b) Pathway; (c) RW; (d) RWP; (e) SRCM; y (f) ST.	106
Fig. 34: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QC propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) GM y (b) RPGM, que obtienen el mejor y el peor resultado para esta componente, respectivamente.	108
Fig. 35: Escenario de adquisición de datos.	119

Fig. 36: Selección de características más relevantes para el proceso de clasificación de eventos de stalling.	123
Fig. 37: Matriz de confusión.	125
Fig. 38: Ejemplo de ventana deslizante para el proceso de clasificación de eventos de stalling.	128
Fig. 39: Comparación de los resultados de experimentación vs. resultados del modelo de clasificación.	131
Fig. 40: Comparativa de prestaciones para diferentes tamaños de ventana deslizante (t) en el modelo de clasificación.	132
Fig. 41: Modelo de predicción ABi-LSTM.	134
Fig. 42: Resultados del método 10-Fold Cross Validation para el modelo ABi-LSTM de predicción de ocurrencia de eventos de stalling.	138
Fig. 43: Ventana deslizante con t=5 para la predicción del valor de la componente de calidad QoX.	143
Fig. 44: Función de pérdidas del modelo en un escenario real ideal; (a) QoD, (b) QoI, (c) QoD y (d) QC.	146
Fig. 45: Función de pérdidas del modelo en un escenario real con pérdidas; (a) QoD, (b) QoI, (c) QoD y (d) QC.	147
Fig. 46: Resultados de estimación QoD en entorno urbano ideal.	151
Fig. 47: Resultados del sistema de notificación QoD en entorno urbano ideal.	151
Fig. 48: Resultados de estimación QoD en entorno suburbano ideal.	152
Fig. 49: Resultados del sistema de notificación QoD en entorno suburbano ideal.	152
Fig. 50: Resultados de estimación QoD en entorno rural ideal.	153
Fig. 51: Resultados del sistema de notificación QoD en entorno rural ideal.	153
Fig. 52: Resultados de estimación QoI en entorno urbano ideal.	154
Fig. 53: Resultados del sistema de notificación QoI en entorno urbano ideal.	154
Fig. 54: Resultados de estimación QoI en entorno suburbano ideal.	155
Fig. 55: Resultados del sistema de notificación QoI en entorno suburbano ideal.	155
Fig. 56: Resultados de estimación QoI en entorno rural ideal.	156
Fig. 57: Resultados del sistema de notificación QoI en entorno rural ideal.	156
Fig. 58: Resultados de estimación QoE en entorno urbano ideal.	157
Fig. 59: Resultados del sistema de notificación QoE en entorno urbano ideal.	157
Fig. 60: Resultados de estimación QoE en entorno suburbano ideal.	158
Fig. 61: Resultados del sistema de notificación QoE en entorno suburbano ideal.	158

Fig. 62: Resultados de estimación QoE en entorno rural ideal.	159
Fig. 63: Resultados del sistema de notificación QoE en entorno rural ideal.	159
Fig. 64: Resultados de estimación QC en entorno urbano ideal.	160
Fig. 65: Resultados del sistema de notificación QC en entorno urbano ideal.	160
Fig. 66: Resultados de estimación QC en entorno suburbano ideal.	161
Fig. 67: Resultados del sistema de notificación QC en entorno suburbano ideal.	161
Fig. 68: Resultados de estimación QC en entorno rural ideal.	162
Fig. 69: Resultados del sistema de notificación QC en entorno rural ideal.	162
Fig. 70: Resultados de estimación QoD en entorno urbano con pérdidas.	163
Fig. 71: Resultados del sistema de notificación QoD en entorno urbano con pérdidas.	163
Fig. 72: Resultados de estimación QoD en entorno suburbano con pérdidas.	164
Fig. 73: Resultados del sistema de notificación QoD en entorno suburbano con pérdidas.	164
Fig. 74: Resultados de estimación QoD en entorno rural con pérdidas.	165
Fig. 75: Resultados del sistema de notificación QoD en entorno rural con pérdidas.	165
Fig. 76: Resultados de estimación QoI en entorno urbano con pérdidas.	166
Fig. 77: Resultados del sistema de notificación QoI en entorno urbano con pérdidas.	166
Fig. 78: Resultados de estimación QoI en entorno suburbano con pérdidas.	167
Fig. 79: Resultados del sistema de notificación QoI en entorno suburbano con pérdidas.	167
Fig. 80: Resultados de estimación QoI en entorno rural con pérdidas.	168
Fig. 81: Resultados del sistema de notificación QoI en entorno rural con pérdidas.	168
Fig. 82: Resultados de estimación QoE en entorno urbano con pérdidas.	169
Fig. 83: Resultados del sistema de notificación QoE en entorno urbano con pérdidas.	169
Fig. 84: Resultados de estimación QoE en entorno suburbano con pérdidas.	170
Fig. 85: Resultados del sistema de notificación QoE en entorno suburbano con pérdidas.	170
Fig. 86: Resultados de estimación QoE en entorno rural con pérdidas.	171
Fig. 87: Resultados del sistema de notificación QoE en entorno rural con pérdidas.	171
Fig. 88: Resultados de estimación QC en entorno urbano con pérdidas.	172
Fig. 89: Resultados del sistema de notificación QC en entorno urbano con pérdidas.	173
Fig. 90: Resultados de estimación QC en entorno suburbano con pérdidas.	174
Fig. 91: Resultados del sistema de notificación QC en entorno suburbano con pérdidas.	174
Fig. 92: Resultados de estimación QC en entorno rural con pérdidas.	175

Fig. 93: Resultados del sistema de notificación QC en entorno rural con pérdidas.	175
Fig. 94: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno urbano ideal.	179
Fig. 95: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno urbano ideal.....	179
Fig. 96: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno suburbano ideal.....	180
Fig. 97: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno suburbano ideal.	180
Fig. 98: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno rural ideal.....	181
Fig. 99: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno rural ideal.....	181
Fig. 100: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno urbano con pérdidas.....	182
Fig. 101: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno urbano con pérdidas.....	182
Fig. 102: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno suburbano con pérdidas.....	183
Fig. 103: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno suburbano con pérdidas.....	183
Fig. 104: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno rural con pérdidas.....	184
Fig. 105: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno rural con pérdidas.....	184

Introducción

1.1. Motivación

Internet ha cambiado y continuará cambiando el estilo de vida de las personas y sus relaciones mediante la interacción del mundo real y el virtual. El nuevo paradigma llamado Internet de las Cosas (*Internet of Things*, IoT) pretende integrar diferentes tecnologías y soluciones en comunicaciones que interactúen con las personas de forma constante para conseguir un objetivo.

IoT suma a los dispositivos capacidades cognitivas, aportándoles la capacidad de comunicarse en cualquier momento con objetos comunes que los rodean, convirtiéndolos así en objetos inteligentes. Los objetos inteligentes capaces de entender y aprender tanto del mundo social como físico se agrupan bajo el término Cognitive IoT (CIoT) [1]. Estos dispositivos tienen la capacidad de tomar datos de los sensores para alcanzar un objetivo. Muchos de estos objetos tienen recursos limitados, por lo que alcanzar un uso eficiente será un aspecto clave en las aplicaciones y servicios de IoT en diferentes ámbitos como la Industria 4.0, las Ciudades Inteligentes (*Smart Cities*), la Logística 4.0, los Sistemas Inteligentes de Transporte (*Intelligent Transport Systems*, ITS), la Salud (*intelligent Health*, iHealth), los hogares inteligentes (*Smart Homes*), la Red Eléctrica Inteligente (*Smart Grid*), etc. Estas aplicaciones y servicios pretenden mejorar nuestra calidad de vida

en cualquier entorno en el que nos encontremos (por ejemplo, trabajo, hogar, calle, centros médicos, centros deportivos, etc.).

Con los años, IoT conseguirá la interconexión global de objetos a través del uso de protocolos de comunicación que enviarán los datos capturados bajo la infraestructura de red existente. Además de algunas características de diseño básicas como el uso de interfaces inalámbricas o memoria flash, existen necesidades específicas para cada aplicación, como pueden ser la autonomía de los dispositivos, el reconocimiento del contexto que le rodea, el comportamiento proactivo, las comunicaciones colaborativas, etc. [2].

Dentro de la arquitectura de la IoT, podemos distinguir tres capas: (i) los periféricos, que obtienen datos del entorno (por ejemplo, temperatura, localización, movimiento...); (ii) el *middleware*, que permite las comunicaciones entre dispositivos independientemente del hardware equipado y de las capas más bajas de software, y (iii) la aplicación, que permite al usuario tomar el control del sistema, haciendo uso de todo su potencial. La mejora de las capacidades cognitivas, la eficiencia en las comunicaciones y el aumento del número de dispositivos conectados en la IoT no solo la dota de un mejor rendimiento, sino que por las características del mismo permite aprovechar el *feedback* del proceso para aprender y mejorar. Se espera que en la próxima década se alcance una cifra de cientos de miles de millones de dispositivos inteligentes conectados a la red, lo que supone un reto importante a la infraestructura existente de comunicaciones [1].

Sin embargo, dado el entorno dinámico de creación de servicios en la IoT y las restricciones que podrían darse, la evaluación de rendimiento en la IoT debe resolverse teniendo en cuenta los recursos disponibles para un servicio, así como el impacto de las capacidades cognitivas aplicando las métricas de calidad apropiadas. Los aspectos anteriores no son fáciles debido a su propia idiosincrasia. Tecnologías como las redes de área amplia de baja consumo (*Low-Power Wide Area Network*, LPWAN) son apropiadas para la IoT porque comparten aspectos como el bajo consumo energético, baja potencia, alto rendimiento, amplio rango de cobertura y rápido despliegue. Además, y de forma progresiva, se espera que las tecnologías emergentes sustituyan a tecnologías actualmente asentadas como pueden ser WiFi, 2G, 3G, 4G, etc.

Basado en el trabajo realizado por Ray *et al.* [3], el punto de partida de esta tesis doctoral es la evaluación de rendimiento basada en dos dimensiones: coste y beneficio. La hipótesis que se plantea como base de este trabajo doctoral es la separación (y representación) de esas dimensiones en varias componentes que reflejen la gran variedad de factores que pueden afectar a las prestaciones de un servicio en la IoT. Así, se propondrá que la dimensión beneficio calcule el rendimiento de las aplicaciones o servicios a través de su representación en tres componentes: (i) Calidad de los Datos (*Quality of Data*, QoD), que evaluará el *raw data* (dato en crudo) adquirido por los sensores en su entorno; (ii) Calidad de la Información (*Quality of Information*, QoI), que medirá el rendimiento del proceso transformando el *raw data* en información apropiada para la toma de decisiones; y (iii) Calidad de Experiencia de usuario (*Quality of user Experience*, QoE), que será la evaluación de una aplicación o servicio a través de la percepción que se espera que tenga

un usuario de ella. Respecto a la dimensión de coste, solo se propondrá la inclusión de la componente Coste de Calidad (*Quality Cost*, QC), que calculará el coste de cada mejora implementada para cada aspecto del sistema; por ejemplo, a través de la medición del nivel de utilización de los dispositivos, del consumo energética o de almacenamiento, de requisitos de computación, etc.

Es importante destacar también que una de las áreas de mayor auge en la comunidad científica es el estudio y la aplicación de técnicas de Aprendizaje Máquina (*Machine Learning*, ML), dentro del ámbito de Inteligencia Artificial (*Artificial Intelligence*, AI). Estas técnicas proporcionan a los sistemas la capacidad de aprender a partir de un conjunto de datos de forma automática. Este aprendizaje se basa en modelos estadísticos resueltos mediante computación tratando de minimizar errores y costes. El proceso de aprendizaje comienza con la observación de los datos para buscar patrones, aparentemente no visibles, y tomar mejores decisiones en el futuro. Así, el objetivo es permitir el aprendizaje de forma autónoma sin asistencia humana [4].

En esta tesis doctoral se propondrán dos modelos para predecir el valor de las métricas de calidad en función del histórico de datos. Para ello se empleará una técnica ML basada en memoria a largo y corto plazo y series temporales. En un primer estudio, se buscará maximizar el rendimiento de un servicio de *video streaming* mediante la predicción de eventos de *stalling* futuros que degradan la calidad del servicio ofrecido. En una segunda contribución, a partir de los datos obtenidos por simulación del modelo de evaluación de prestaciones, se predecirá de forma independiente el valor de cada componente de calidad

para poder anticiparse a pérdidas de la misma y actuar de forma previa. Para poder ubicar este trabajo en la literatura existente, se hará uso de un sistema de notificación que permitirá comparar esta contribución con otras existentes.

1.2. Objetivos de la tesis doctoral

El objetivo principal de la tesis es contribuir en el avance de las técnicas de evaluación de prestaciones en el entorno de la IoT. Para ello, las contribuciones de la tesis serán las siguientes: (i) proponer un nuevo modelo con visión holística basado en métricas de coste-beneficio para la evaluación de rendimiento en la IoT con el fin de mejorar la calidad de los servicios; (ii) validación de los enfoques desarrollados mediante simulaciones; y (iii) aplicación de técnicas de AI en el contexto de la evaluación de prestaciones en la IoT.

Para alcanzar el objetivo planteado, el desarrollo de la tesis doctoral ha pasado por las siguientes etapas:

1. Conocer estado del arte de IoT y de los modelos de evaluación de prestaciones.
 - a. Conocer los modelos de QoE existentes, especialmente desarrollados para IoT.
 - b. Conocer métricas alternativas de evaluación para IoT, tanto objetivas como subjetivas.
2. Proponer un nuevo modelo de evaluación de rendimiento basado en métricas sobre las que satisfacer las prestaciones en IoT.

3. Análisis del modelo propuesto mediante simulación en un entorno estático y otro móvil.
4. Estudio y propuesta de aplicación de técnicas AI en la evaluación de rendimiento, para estimar el comportamiento futuro de las métricas de calidad en IoT y poder realizar acciones correctivas.

1.3. Organización de la tesis doctoral

En esta sección se resumen los contenidos de cada uno de los capítulos de esta tesis:

Capítulo 2: “Estado del arte” presenta una revisión de los trabajos científicos relacionados con esta tesis doctoral. Los sistemas IoT, desde su nivel físico hasta el nivel de aplicación, pueden estar dotados de capacidades cognitivas que permiten trabajar de forma autónoma sin intervención humana a partir de un conjunto de datos mediante la aplicación de técnicas de ML. Bajo los sistemas existentes, las métricas de evaluación de prestaciones, como QoS o QoE, son de gran utilidad en este contexto para la medición del rendimiento. Otro hándicap en dispositivos IoT es la seguridad de las comunicaciones, suponiendo un reto importante al tratarse de dispositivos con recursos limitados. Estudios previos abordan la aplicación de entornos multidisciplinares con el uso de múltiples tecnologías englobadas bajo IoT como es el caso de drones que emplean tecnología LoRa para sus comunicaciones inalámbricas.

Capítulo 3: “Un Enfoque Innovador para la Evaluación de Rendimiento en IoT” describe un novedoso modelo de evaluación de prestaciones basado en componentes y

métricas que mide el rendimiento de una aplicación o servicio ante la carencia de un estándar de estas características en IoT. Por la naturaleza de los parámetros considerados, estas métricas toman mediciones objetivas y subjetivas que pueden adaptarse y mejorar en función de las necesidades de cada aplicación o servicio. Las métricas propuestas se dividen en dos dominios: beneficio, que incluye QoD, QoI y QoE, y coste, que comprende la componente QC.

Capítulo 4: “Caso de Estudio: LoRa/LoRaWAN” verifica el modelo de prestaciones planteado en el capítulo anterior bajo diferentes escenarios, entornos y condiciones. Para ello se emplean diferentes herramientas de simulación, como OMNeT++ INET y FLoRa. La aplicación bajo estudio es una red de estaciones de medición de la calidad del aire en la que se utilizan dispositivos Long-Range (LoRa) bajo dos enfoques: nodos estáticos y nodos con movilidad. Para el primer caso, se presentan dos escenarios, uno ideal y otro con pérdidas en la red troncal, evaluados en base a tres entornos (urbano, suburbano y rural). En el segundo caso, los nodos con movilidad (representados como UAVs) presentan 6 escenarios bajo un entorno suburbano donde varía el número de dispositivos, los modelos de movilidad y la velocidad de estos.

Capítulo 5: “Aplicación de Técnicas de Machine Learning en la Evaluación de Prestaciones” plantea dos propuestas que incorporan el uso de técnicas AI en la evaluación de prestaciones. La primera propuesta aborda la problemática de los eventos de *stalling* en la distribución de contenido multimedia, siendo este un factor clave en la degradación de la QoE. Recordemos que la QoE será una de las componentes de calidad en la evaluación

de prestaciones según el modelo propuesto en el Capítulo 3. El método propuesto permite clasificar paquetes como pertenecientes a un evento de *stalling* y predecir si los siguientes se etiquetarán también como eventos de *stalling*, lo que puede repercutir enormemente en la QoE observada. La segunda propuesta definirá un modelo para predecir el valor futuro de las diferentes componentes de calidad (QoD, QoI, QoE y QC), anticipándose a la caída de rendimiento de éstas.

Capítulo 6: “Conclusión” cierra la tesis destacando los principales hallazgos extraídos en esta contribución científica.

1.4. Contribuciones de la tesis doctoral

Las principales contribuciones de esta tesis pueden resumirse en:

Capítulo 2: “Marco teórico”.

- Revisión de trabajos científicos en la literatura relacionada, donde se concluye que actualmente no existe un modelo de evaluación de prestaciones para IoT.
- Adicionalmente, la revisión de estos trabajos permite la búsqueda e identificación de métricas de rendimiento a diferentes niveles bajo el paradigma de IoT.
- Por último, se presentan conceptos y tecnologías que se han empleado en el desarrollo de esta tesis doctoral, como por ejemplo la tecnología LoRa y técnicas de ML.

Capítulo 3: “Un Enfoque Innovador para la Evaluación de Rendimiento en IoT”.

- Ante la carencia de un modelo de evaluación de prestaciones para IoT, se propone un método nuevo de medición de rendimiento con el objetivo de ser más eficientes en la toma de decisiones, en función del estado del sistema.
- La propuesta incluye la medición de calidad en diferentes niveles, denominados componentes de rendimiento. En particular, se introducen la QoD, QoI, QoE y QC. Cada una de estas componentes, con sus correspondientes métricas, analiza el rendimiento en IoT en distintos puntos del proceso, siendo este un factor clave en los sistemas de evaluación de prestaciones.

J.M. Martínez-Caro, M.-D. Cano, “A novel holistic approach for performance evaluation in Internet of Things”, *Int. J. Commun. Syst.* (2020), pp. 1–24. <https://doi.org/10.1002/dac.4454>.

Factor de Impacto JCR: 1,319 (Q3), Rank: 198/266 in Engineering, Electrical & Electronic.

J.M. Martínez-Caro, M.-D. Cano, “Definición de métricas de calidad en IoT basadas en coste-beneficio”, In *I Jornadas Doctorales UPCT. Salón de Grados de la Escuela Técnica Superior de Ingeniería Industrial, Universidad Politécnica de Cartagena. Lunes 16 de abril de 2018*. Comunicación oral.

Capítulo 4: “Caso de Estudio: LoRa/LoRaWAN”

- El modelo de evaluación de prestaciones planteado en el capítulo anterior pasa a ser examinado bajo un entorno de simulación desarrollado a partir de herramientas *Open Source* como OMNeT++ [5], INET [6] y FLoRa Framework

[7], con mejoras adicionales para permitir una evaluación en escenarios diversos con LoRa.

- La robustez del modelo planteado se evalúa bajo dos escenarios bien diferenciados: (i) con nodos estáticos en la topología y (ii) con nodos móviles mediante el uso de UAVs con varios patrones de movilidad.

J.M. Martínez-Caro, M.-D. Cano, “Improving the FLoRa Simulation Framework for the Performance Evaluation of IoT Scenarios”. In *Proceedings of SENSORCOMM'19 (2019), Niza (Francia)*, pp. 1-7. ISSN: 2308-4405. ISBN: 978-1-61208-744-3.

J.M. Martínez-Caro, M.-D. Cano, “Simulación y análisis de rendimiento en dispositivos LoRa sobre drones”. In *Proceedings of Jornadas de Ingeniería Telemática JITEL'19 (2019), Zaragoza (España)*, pp. 1–8. Comunicación oral. ISBN: 978-84-09-21112-8.

J.M. Martínez-Caro, M.-D. Cano, “IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study”, *Wireless Communications and Mobile Computing (2019)*, pp. 1-12. <https://doi.org/10.1155/2019/4307925>
Factor de Impacto JCR: 1,819 (Q3), Rank: 153/266 in Engineering, Electrical & Electronic.

J.M. Martínez-Caro, M.-D. Cano, “A novel holistic approach for performance evaluation in Internet of Things”, *Int. J. Commun. Syst. (2020)*, pp. 1–24, <https://doi.org/10.1002/dac.4454>
Factor de Impacto: 1,319 (Q3), Rank: 198/266 in Engineering, Electrical & Electronic.

Capítulo 5: “Aplicación de Técnicas de Machine Learning en la Evaluación de Prestaciones”

- A partir de un *dataset* obtenido en un contexto de transmisión de tráfico multimedia, se busca minimizar la aparición de eventos de *stalling*, debido al impacto negativo que este tipo de eventos tienen sobre la QoE. Mediante la observación de los datos de las cabeceras de los mensajes intercambiados a nivel de red y transporte, y con el uso de un modelo de clasificación basado en *Long Short-Term Memory* (LSTM) bidireccional y ventanas deslizantes, será posible con nuestra propuesta clasificar y predecir eventos de *stalling* en la transmisión de video.
- A partir de los *dataset* generados en las simulaciones del capítulo 4, se entrena un modelo de regresión basado en LSTM bidireccional y series temporales capaces de evaluar cada componente de calidad del modelo propuesto en el capítulo 3 de forma independiente. El modelo planteado mejora los resultados en un sistema de notificación respecto otros trabajos de la literatura.

J.M. Martinez-Caro, M.-D. Cano, “On the identification and prediction of stalling events to improve QoE in video streaming”, *Computer Communications* (2020), pp. 1–10. En proceso de revisión.

Factor de Impacto: 2,816 (Q2), Rank: 106/266 in Engineering, Electrical & Electronic.

J.M. Martinez-Caro, M.-D. Cano, “Improving the IoT performance based on a regression model and QoX metrics”, *Computer Communications* (2020), pp. 1–6, Factor de Impacto: 2.816 (Q2), Rank: 106/266 in Engineering, Electrical & Electronic. En proceso de revisión.

H.F. Bermudez, J.M. Martinez-Caro, R. Sanchez-Iborra, J.L. Arciniegas, M.-D. Cano, “Live video-streaming evaluation using the ITU-T P.1203 QoE model in LTE networks”, *Comput. Networks* (2019), vol. 165, pp. 1–12, <https://doi.org/10.1016/j.comnet.2019.106967>
Factor de Impacto: 3.111 (Q2), Rank: 85/266 in Engineering, Electrical & Electronic.

2.1. Introducción

Como se ha demostrado durante el periodo de confinamiento por COVID-19, Internet es una herramienta fundamental en nuestro día a día y lo será aún más en los próximos años [8–10]. El aumento considerable que ha experimentado en cuanto al número de dispositivos conectados a la red, así como el incremento en el número de aplicaciones y servicios ofrecidos a los usuarios, dando lugar a una red cada vez más heterogénea [11], son muestra de ello.

Tradicionalmente los dispositivos conectados a Internet han adoptado la arquitectura TCP/IP. No obstante, esta arquitectura y los protocolos de comunicación que se suelen emplear en ella no se ajustan perfectamente a todo tipo de dispositivos, especialmente aquellos con recursos de software y hardware. Generalmente, estos dispositivos son de dimensiones reducidas, operan cerca del usuario y se agrupan bajo el nuevo paradigma de la IoT. El objetivo principal es hacer la vida de los usuarios más sencilla dando lugar a nuevas aplicaciones o servicios en ámbitos muy amplios como la seguridad en el hogar, domótica, monitorización de salud, etc. [12,13].

Bajo la IoT se buscan alternativas a las soluciones tradicionales para conseguir comunicaciones fiables utilizando la menor cantidad de medios. Véase LPWAN, que consigue excelentes prestaciones optimizando los recursos. Entre las tecnologías

desarrolladas hasta el momento destaca LoRa por su bajo precio, tasa de datos versus consumo energético, gran alcance, alto rendimiento y rápida implementación [14]. Sobre su capa física, LoRaWAN define la arquitectura, capas y protocolos óptimos para la conservación de los recursos.

En redes de comunicaciones clásicas, tradicionalmente se han utilizado métricas estandarizadas para la evaluación objetiva de las condiciones de la red mediante la QoS. Posteriormente QoS evolucionó a una métrica más compleja como QoE, que cuenta con una componente subjetiva como el *background* del usuario y el contexto en el que se encuentra. Aunque si existen estándares de medición de QoE, no los encontramos dentro del marco IoT [15].

En estas innovadoras redes de comunicaciones que nos ofrece el concepto IoT, los dispositivos operan muy próximos al usuario aumentando la interacción con éste favoreciendo la adquisición de capacidades cognitivas para la adquisición y entrenamiento de los datos mediante el uso de ML.

En las siguientes subsecciones se explicarán estas técnicas para ofrecer al lector una base del estado de la técnica en cada una de ellas, y siempre que sea posible, dentro del marco de trabajo de la IoT.

2.2. Quality of Service

Según la Unión Internacional de Telecomunicaciones (*International Telecommunication Union*, ITU) se define el término QoS como “*Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service*” bajo la recomendación Rec. E.800 [16]. Tradicionalmente, el término QoS ha sido abordado desde la perspectiva del cliente para detectar la degradación de los servicios. Es importante tener en cuenta que no todos los servicios son iguales ni tienen los mismos requisitos. Hay que destacar que dos servicios similares pueden ser resueltos de modo muy diverso dependiendo de varios factores como pueden ser sesiones de conexión, tipos de dispositivos, interacción hombre-máquina, etc. Así, la QoS basa sus mediciones en parámetros que aportan un valor cuantificable al rendimiento alcanzado dentro de un rango [17]. Los actores principales de esta comunicación son dos: por un lado, tenemos al cliente que demanda un servicio; por otro, el proveedor de servicio le hace llegar al cliente el servicio demandado. Desde una perspectiva más amplia, entran en juego varios factores en este procedimiento como [18]: (i) *QoS Required by users* (QoSR), que especifica el nivel de calidad que el cliente final demanda al proveedor de servicios; (ii) *QoS Offered by service provider* (QoSO), que representa el nivel de calidad que el proveedor de servicios ofrece al usuario derivando en un Acuerdo de Nivel de Servicio (*Service Level Agreement*, SLA); (iii) *QoS Delivered by service provider* (QoSD), que es el valor real de la calidad conseguida en la comunicación; and (iv) *QoS Experienced by user* (QoSE), conocida usualmente como QoE, que expresa

el nivel de calidad en grado de satisfacción que el cliente cree que ha experimentado, generalmente de forma subjetiva.

La recomendación E.860 de la ITU-T [19], define SLA como un acuerdo formal entre dos o más entidades con el objetivo de evaluar las características del servicio, las responsabilidades y las prioridades de cada entidad. Un SLA puede incluir declaraciones sobre rendimiento, tarifas, facturación, prestación de servicios y compensaciones. Este acuerdo comercial se monitoriza desde la correspondiente Agencia Nacional Reguladora (*National Regulatory Agency*, NRA). En el caso de existir diferentes proveedores para un servicio, entra en juego el concepto “*one stop responsibility*”, donde el proveedor principal es el responsable de garantizar las prestaciones pactadas en el SLA de todos los servicios en el cliente. Tanto el proveedor de servicios como el cliente son entidades que prestan o consumen servicios respectivamente. El servicio prestado se intercambia bajo un punto de interacción, que pueden constituir la frontera lógica de dos entidades (Fig. 1) [20]. *QoS Agreement* es la parte del SLA que incluye el acuerdo del servicio a prestar para garantizar los niveles de QoS y conseguir la satisfacción del cliente [18,21]. Tanto en [18,20,21], la QoS depende de la precisión de los servicios proporcionados que son evaluados en base a tres criterios (Rec. I.350): velocidad, precisión y fiabilidad.

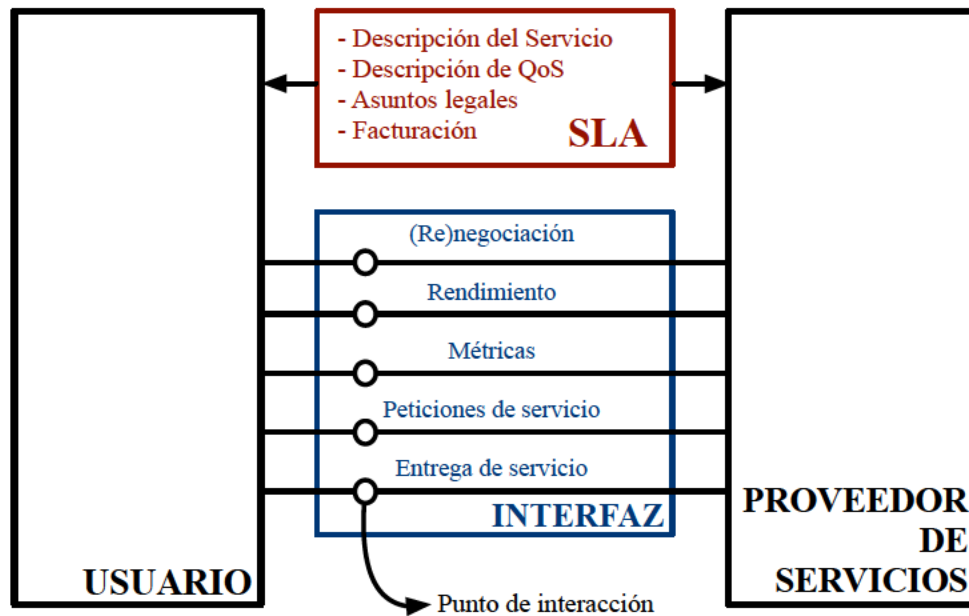


Fig. 1: Interfaces y acuerdos SLA entre usuarios y proveedores de servicios.

Los parámetros definen las mediciones que deben realizarse, y los objetivos establecen los valores que deben alcanzar dichos parámetros para que la calidad se considere satisfactoria. El término “*rate*” establece la frecuencia con la que se desarrollan las acciones, “*ratio*” define la proporción de acciones que tienen éxito, y “*time*” determina la duración media de las acciones que tienen éxito [17].

Respecto a la medición de parámetros QoS, podemos encontrar: (i) medición pasiva, donde los sistemas de gestión envían paquetes “*prueba*” tomando medidas como el *delay*, *jitter*, o el ratio de pérdidas en el *path*; y (ii) medición activa, donde a lo largo de la red se ubican sondas que proporcionan información precisa del estado de los dispositivos y de la propia red. La medición activa puede no ser conveniente en grandes redes por lo altos

costes acarreados, pero permite elaborar informes de evaluación bajo diferentes situaciones [22].

Tradicionalmente, la QoS se ha centrado en los parámetros orientados a servicios de datos con independencia de la red subyacente y bajo el modelo *best effort* en la entrega de paquetes. Los *Key Performance Indicators* (KPI) más importantes para la medición de la QoS son: (i) *Bandwidth* (BW), como la cantidad máxima de bits que un enlace puede transmitir por unidad de tiempo (generalmente segundos), mencionando que a menudo también se cuantifica el consumo en bits que tiene una aplicación o servicio; (ii) *delay* es el retardo experimentado por un paquete desde que enviado por la red hasta que se recibe en el destinatario donde se suma el tiempo de propagación por la red y de encolado en *buffers*; (iii) *jitter* mide la variabilidad del retardo; (iv) *Loss Rate* analiza el número de mensajes perdidos respecto al total transmitidos; y (v) *Error Rate* el número de mensajes erróneos en recepción respecto al total transmitidos [23].

2.3. Quality of Experience

De acuerdo a la recomendación ITU-T P.10/G.100 [24], la QoE se define como la aceptación general de una aplicación o servicio, según el criterio subjetivo del usuario final. Dentro de los factores que influyen dentro de QoE se incluyen el tipo y las características de la aplicación o servicio, las expectativas del cliente, los antecedentes culturales del usuario, cuestiones socioeconómicas, perfiles psicológicos, estado emocional del usuario, etc. [24]. En la evaluación de dicha métrica deben constar los factores de influencia del

servicio o aplicación, siendo una medición más exhaustiva cuanto mayor sea el número de factores conocidos. La aceptabilidad general puede verse influida por las expectativas de los usuarios y el contexto (estado de ánimo, entorno de hogar o trabajo, etc.) [24].

Dependiendo de la arquitectura propuesta, la evaluación y explotación de datos en QoE pueden ser enfocados desde diferentes perspectivas. Los autores en [25], subdividen la QoE en tres grandes grupos: (i) factores de red, que recogen el rendimiento extremo a extremo de la comunicación descrito por la QoS; (ii) factores humanos, que tienen en cuenta aspectos psicológicos y sociológicos; y (iii) factores de contexto, que están influenciados por el entorno espacio-temporal donde se presta el servicio, dispositivos disponibles, contenido, aspectos financieros, etc. Además, esta contribución propone un *framework* conceptual formado por tres módulos que cooperan entre sí para maximizar la experiencia del usuario, denominados *QoE-controller*, *QoE-monitor* y *QoE-manager*.

Fiedler *et al.* [26] centran su propuesta en una relación cualitativa entre QoS y QoE sobre diferentes escenarios. Para valores óptimos de QoS, pequeñas perturbaciones de esta no ocasionan alteración en la QoE. Cuando los valores de QoS se alteran superando cierto umbral x_1 , el valor de la QoE decrece de forma exponencial. Cuando el valor de QoS rebasa un segundo umbral x_2 , puede hacer el valor de QoE inaceptable. Por último, los autores llevan a cabo demostraciones de su propuesta sobre entornos de voz y navegación web.

Tu [27] particulariza el uso de QoE en aplicaciones de *Smart Cities* basándola en la satisfacción del usuario mediante un modelo de cuatro capas, denominadas *Object*, *Network*, *Offline* y *Online*, y cinco métricas, denominadas *Efficiency*, *Usability*,

Usefulness, Interactability y Personalization, con las que conocer el rendimiento general del sistema. Al igual que Liotou *et al.*, Tu también propone un marco de trabajo para la gestión de la QoS/QoE y la mejora del rendimiento de los sistemas mediante el uso de técnicas AI.

Chen *et al.* [28] proponen la evaluación de la calidad de reproducción de *video streaming* mediante la relación de QoE y QoS en diferentes entornos. Los valores de QoE se obtienen mediante test subjetivos a los usuarios participantes, los cuales presentan grandes limitaciones. Finalmente se propone un modelo donde, a partir de los datos existentes y técnicas de AI, se predice el valor de QoE basándose en los parámetros de QoS de forma indirecta. Finalmente, los autores proponen la optimización de la transmisión de video por la red basada en métricas QoE minimizando costes.

Mitra *et al.* [29] llevan a cabo su estudio sobre aplicaciones de Voz sobre IP (*Voice over IP, VoIP*) mediante la simulación de muy diversas condiciones de red. En dichas simulaciones proponen, desarrollan y validan un modelo de predicción de QoE teniendo en cuenta atributos muy variados mediante la utilización de redes Bayesianas. Estos atributos quedan organizados en cuatro grandes grupos, que son los siguientes: usuario y entorno, dispositivos y herramientas, datos de aplicación y, por último, datos de red.

Liu *et al.* [30] examinan el problema de la optimización holística en IoT sobre una arquitectura *Edge-Computing* basada en tres planos (plano de dispositivos, planos de servidores y plano en la nube - *cloud*). Los autores consideran algunos factores importantes dentro de la IoT, como la heterogeneidad en las redes, consumo energético, tiempo de

ejecución, fiabilidad del sistema y la QoE. Para cuantificar el valor de QoE, se utiliza la escala *Mean Opinion Score* (MOS) sugerida por la ITU, que permite utilizar un método cuantitativo para describir el valor subjetivo del usuario [24]. La valoración subjetiva de un mismo servicio puede ser diferente de un usuario a otro debido a los factores anteriormente descritos. La variable MOS se emplea para cuantificar la QoE de los servicios o aplicaciones puntuando de 1 a 5 la calidad experimentada en el servicio dado por el cliente, siendo 1 la peor calidad y 5 la mejor calidad experimentada por el usuario (véase Tabla 1) [24].

Tabla 1: Equivalencia MOS.

Mean Opinion Score (MOS)	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

2.4. Internet of Things

El término IoT [2][31] fue acuñado por Kevin Ashton tras el desarrollo de un sistema de identificación de dispositivos llamado Identificación por Radio Frecuencia (*Radio Frequency IDentification*, RFID), formado por lectores y etiquetas con un identificador único. Estas etiquetas están compuestas por un pequeño microchip y antena, distinguiéndose entre etiquetas pasivas o activas. En primer lugar, se desarrollaron sistemas

RFID pasivos donde la energía enviada por el lector es utilizada por la etiqueta para reenviar información. Más tarde, se crearon otras versiones más complejas con etiquetas activas que incorporaban baterías y componentes adicionales permitiendo un mayor rango y coste. Esta versión de etiquetas activas fue el origen de lo que hoy en día se conoce como IoT [32].

Por tanto, IoT es un nuevo paradigma donde confluyen un gran número de dispositivos electrónicos de forma coordinada y colaborativa en temas de comunicación, sensorización y actuación bajo un mismo marco de trabajo. También pueden verse como dispositivos simples que permiten la interconexión de humanos, máquinas y elementos comunes (cosas – “*things*”). Estos elementos comunes que usamos en nuestro día a día (véase electrodomésticos, mobiliario, iluminación, etc.) se mimetizan con el entorno que los rodea pudiendo ser controlados y monitorizados de forma remota [32,33]. Estos dispositivos pueden disponer de capacidades cognitivas que les atribuyen el poder de aprender, pensar y entender tanto el mundo social y virtual que les rodea por ellos mismos, imitando los procesos cognitivos de los humanos en el diseño IoT. Mark Weiser, el precursor de la computación ubicua, definió el entorno inteligente como “el mundo físico enriquecido de forma transparente con sensores, actuadores, pantallas y elementos computacionales incrustados en los objetos cotidianos de nuestras vidas y conectados a una red de comunicaciones de forma continua” [33,34]. Generalmente, estos dispositivos son limitados en términos de computación, capacidad, memoria o batería. Este hecho hace que las tecnologías que se engloban bajo IoT estén expuestas a amenazas muy diversas, poniendo en peligro la gran cantidad de información que manejan [12,35]. Entre las muchas

aplicaciones que puede tener, IoT pretende mejorar la calidad de vida de las personas haciendo frente a los retos globales (véase la contaminación, el bienestar de las personas o la productividad en sectores como la ganadería o la agricultura, entre otros). Para ello, es necesario poder sensorizar el entorno, obtener datos, compartirlos, procesarlos y actuar en función de la información adquirida, pero también se requiere una gran expansión de dispositivos para poder obtener variedad en los datos.

Un despliegue IoT requiere una infraestructura integrada con diferentes tecnologías, aplicaciones y servicios. Diversas redes, tecnologías de comunicación y acceso, así como dispositivos, se integran dentro de un mismo entorno e interactúan intercambiando datos entre ellos, generando información, conocimiento y valor. Es importante saber qué se conecta y cómo es conectado dentro de estas redes, teniendo en cuenta que la IoT no es una simple combinación tecnológica de diversas conectividades que satisfacen las necesidades de cada caso particular [3]. La conectividad en IoT se ha visto favorecida en gran medida por el desarrollo de tecnologías como WiFi y 4G. La computación es una parte fundamental de IoT que requiere conocer los recursos disponibles y apunta a un comportamiento autónomo e inteligente consciente del contexto que la rodea. La integración con plataformas *cloud* puede solventar la desventaja de recursos de computación en los dispositivos [36]. Así, IoT no es una tecnología joven sino más bien madura que está apunto de transformar Internet para convertirla en una Internet del futuro con total integración [37].

En relación con la conexión entre humanos, equipos de computación y elementos de uso común, se intercambian una gran cantidad de datos con información sensible relevante a las personas. Estos datos pueden aportar información de la zona en la que se mueve, constantes vitales, o acciones rutinarias que podrían ser de gran valor para un atacante. Por ello es importante aplicar técnicas de seguridad efectivas que permitan conexiones seguras entre dispositivos, así como cifrado de los datos para que solo el emisor y el receptor del mensaje sean capaces de ver el contenido disminuyendo la vulnerabilidad de los individuos [38].

2.4.1. Capacidades cognitivas

El término cognición humana puede entenderse como el conocimiento alcanzado mediante el ejercicio de las facultades mentales bajo el marco de la percepción, pensamiento, memoria, imaginación y voluntad. La cognición implica, por tanto, procesos mentales, marcos de referencia, estructuras neuronales, percepción, evaluación, interpretación, y comunicación. El fin de la ciencia cognitiva es la expresión del conocimiento humano mediante la explicación de factores causales, conscientes o inconscientes, simples o complejos, etc.

Dentro de las relaciones sujeto-objeto, la cognición es el reflejo de la información adquirida, interpretada y utilizada por el ser humano. En este aspecto, podemos ampliar el concepto de existencia-motivación, destacando cómo se entiende el entorno, cómo se conforma el mundo conceptual sobre nuestro entorno, cómo se adquiere consciencia, y qué

se percibe y se aprende en la memoria en función de los acontecimientos. Dentro de dicho mundo conceptual, la mente es un mecanismo de computación que se asemeja al de los ordenadores regido por reglas lógicas y una secuencia de pasos o etapas [39].

El ser humano posee capacidades cognitivas con las que comprender y adaptarse al entorno que le rodea, mejorando mediante Estimulación Cognitiva (*Cognitive Stimulation*, CS) en base a práctica y experiencia. La CS se define como “un conjunto de técnicas y estrategias para optimizar la eficacia del funcionamiento de las distintas capacidades cognitivas”, no únicamente centradas en la parte cognitiva, sino que además incluyen otros factores como la esfera conductual, social, familiar, etc. [40,41].

A pesar de la naturaleza inherente de este concepto en el contexto de la neurociencia a nivel humano, algunos autores lo extraen para utilizarlo a nivel técnico, partiendo de una base matemática aplicable a cualquier campo de la ingeniería. En lo que respecta al sector de las Tecnologías de la Información y las Comunicaciones (TIC) puede utilizarse a nivel de comunicaciones, computación o conectividad, siendo IoT uno de los campos de trabajo con mayor futuro en este aspecto. Para ello, algunos autores definen la integración de tecnologías bajo el paradigma IoT y capacidades cognitivas como *Cognitive IoT* (CIoT) [42]. Si el objetivo de la tecnología IoT es facilitar la vida a los usuarios, el uso de estos dispositivos sin la parte inteligente o cognitiva podría hacerlos inservibles para la misión que han sido encomendados. Por ende, la capacidad cognitiva puede verse como un campo integrador que permite entender y aprender del mundo físico y social con la mínima interacción humana. Entre las ventajas de integración de capacidades cognitivas en IoT

destacan una mejor experiencia de usuario, aumento del conocimiento mediante la experiencia, el ahorro de tiempo, esfuerzos, recursos, o una mejor prestación de servicios entre otros [43].

2.5. Low-Power Wide Area Network

Una de las tecnologías de comunicación más apropiadas en la implementación de IoT es LPWAN ya que asegura transmisiones de datos a grandes distancias, gran escalabilidad, bajo consumo energético y bajo coste en el proceso de despliegue de la red. Como ejemplo, LoRa permite alcanzar distancias en torno a 30 km en zonas rurales de forma inalámbrica y más de 10 años en la vida útil de sus baterías a un bajo coste [44].

La mayoría de las aplicaciones se implementan bajo una topología de estrella donde los terminales repartidos en el área de trabajo envían los datos capturados a un nodo central, también llamado *gateway*. Este montaje permite simplificar la red LPWAN estableciendo una conexión directa entre la estación de medición y el *gateway*. Una vez recogidos los datos en el *gateway*, se reenvían por la red de Internet si fuese necesario. La desventaja de esta tecnología es el bajo *bit-rate* en la transmisión de datos ya que los dispositivos existentes son limitados en recursos y están sujetos a las restricciones en el uso del canal de comunicación [45]. En concreto, la mayoría de las plataformas utilizadas actualmente utilizan la banda de frecuencia ISM (*Industrial, Science, and Medical*) que depende la región en la que nos encontremos (868 MHz en Europa, 915 MHz en US y 433 MHz globalmente). En algunas implementaciones de este tipo de redes se da un tráfico

asimétrico donde la mayor parte es ascendente, siendo origen el nodo final y destino el *gateway* [46].

Las características de la tecnología LPWAN se ajustan a entornos tolerantes al retardo donde no se requieran altos *data rates*, con bajo consumo energético implementado en dispositivos de bajo coste. Es decir, escenarios donde se envíen pequeñas cantidades de datos de forma infrecuente con un coste de implementación bajo. Otras tecnologías de comunicación como WiFi y Bluetooth no son aceptadas bajo el término LPWAN debido a su alto consumo energético, especialmente en aplicaciones de alta demanda, así como tareas de transmisión y recepción [46].

Todos los posibles casos de uso en IoT no están gestionados por tecnologías LPWAN, pero sí que ambas se ajustan perfectamente al diseño de sistemas integrados con otras redes. Tal es el caso que plantea [13], que dota de acceso a internet a “*wearables*” mediante el uso de *Wireless Body Area Network* (WBAN) usando tecnologías de acceso LPWAN. Este tipo de aplicaciones pueden ser de gran utilidad en el campo de la medicina para monitorización de individuos. En entornos industriales, LPWAN también está en auge proporcionando comunicaciones a grandes distancias y bajo consumo energético, anteriormente provistas por redes móviles (3G, 4G, etc.) y Redes de Sensores Inalámbricas (*Wireless Sensor Networks*, WSN) respectivamente [46]. Debido a la constante monitorización necesaria de los terrenos, plantaciones, riegos, etc., el sector agrícola es otro gran candidato en la explotación de esta tecnología. Aun así, tampoco debemos olvidar otros muchos campos de aplicación como son las *Smart Cities*, las *Smart Grids*,

aplicaciones *Smart Home*, seguimiento en diferentes entornos (paquetería, animales en su medio natural o infraestructuras), etc. [1].

Algunas de las plataformas LPWAN de más relevancia son Sigfox [47], LoRa/LoRaWAN [48], Weightless [49], Ingenu RPMA [50], Telensa [51], Dash7 [52], etc. (véase Tabla 2) [1,45,53]. Dadas las muchas posibilidades que nos ofrece LPWAN, vamos a particularizar en este trabajo doctoral en la tecnología LoRa/LoRaWAN como caso de estudio.

Tabla 2: Comparación de Tecnologías LPWAN [46].

	LoRaWAN	Sigfox	Weightless			Ingenu	Telensa	Dash7
			-W	-N	-P			
Banda de frecuencia	433/868/915 MHz	868/915 MHz	Claros banda TV	Sub-GHz	Sub-GHz	2.4 GHz	Sub-GHz	Sub-GHz
Max. Data rate	50 kbps	100 bps	10 Mbps	100 bps	100 kbps	19 kbps /MHz	346 Mbps	-
Alcance (urbano)	5 km	10 km	5 km	5 km	2 km	15 km	1 km	3 km
Tamaño de paquete	Max. 256 Bytes	12 Bytes	Min. 10 Bytes	Max. 20 Bytes	Min. 10 Bytes	Max. 10 kBytes	Max. 65 kBytes	-
Downlink	Si	Si	Si	No	Si	Si	Si	Si
Topología	Estrella	Estrella	Estrella	Estrella	Estrella	Estrella /Tree	Estrella /Tree	Estrella
Roaming	Si	Si	Si	Si	Si	Si	Si	Si
Seguridad	Completa	Parcial	Completa	Completa	Completa	Completa	En desarrollo	-
Protocolo propietario	Parcial	Propietaria	Estándar	Estándar	Estándar	Propietaria	Estándar	Propietaria

2.5.1. Long-Range

La tecnología LoRa es apropiada para el paradigma planteado por IoT ya que comparte conceptos clave como el bajo coste, bajo consumo energético, alto rendimiento, gran área de cobertura, bajo *bit-rate* y rápido despliegue con un gran número de dispositivos conectados. La gran aceptación de LoRa la ha convertido en una solución destacada dentro de LPWAN y del mercado de IoT.

Al hablar de LoRa, hablamos del nivel físico de los dispositivos que utilizan modulación *Chirp Spread Spectrum* (CSS) en la banda ISM. Dicha técnica de modulación es propietaria de Semtech [54] y permite extender una señal estrecha sobre un ancho de banda de canal más amplio ganando resistencia frente a interferencias y ruidos del canal como el efecto Doppler y multi-camino [1,55,56]. A este nivel existen diferentes parámetros que caracterizan la comunicación inalámbrica de estos dispositivos. En primer lugar, el parámetro *Spreading Factor* (SF) influye en el rango de operación y el *data rate* asociado. Esta variable ajusta su valor de 7 a 12. Cuanto menor es el valor de SF, mayor es la distancia alcanzada, y menor será el *data rate* obtenido. Los valores de SF son ortogonales, lo que añade un tercer valor de diversidad permitiendo al *gateway* recibir múltiples tramas en el mismo instante temporal con la misma frecuencia y diferente valores de SF [56][57]. Otro parámetro, la Potencia de Transmisión (*Transmission Power*, TP) influye en el alcance y especifica la potencia de transmisión utilizada por el dispositivo variando de -4dBm a 20dBm. Cuanto menor sea la TP más difícil será para la señal alcanzar una distancia mayor. La frecuencia se centra en pasos de 61 Hz bajo el parámetro

Frecuencia Portadora (*Carrier Frequency*, CF) en la banda ISM. El *Coding Rate* (CR) proporciona protección frente a interferencias (4/5, 4/6, 4/7 y 4/8), donde mayores valores aportan una mayor protección e incrementan el *Time-on-Air* (ToA) entre dispositivo y *gateway* [57] dependiendo del número de símbolos y el periodo de los mismos [58]. Por último, *Bandwidth* (BW) es el ancho de la frecuencia utilizada. Un BW alto proporciona una mayor velocidad en la transmisión de datos, pero reduce la sensibilidad en recepción.

De entre todos los anteriores, los principales parámetros que definen el modo de funcionamiento de un dispositivo LoRa son SF, BW y TP. Mediante la utilización de la técnica de modulación CSS y con un ajuste óptimo de los parámetros de configuración, LoRa permite decodificar las señales recibidas con 19,5dB por debajo del nivel de ruido [57] y alcanzar una distancia de 30 km en zonas rurales con línea directa de visión [1,13,48,59–61]. La optimización de parámetros es importante dependiendo de la topología y distribución de dispositivos utilizada, con el objetivo de disminuir el consumo energético y maximizar el área de cobertura. Pese a que la mayoría de contribuciones implementan topología en estrella para optimizar los recursos disponibles, LoRa también puede establecer una topología en malla [62].

En la mayoría de las contribuciones la topología utilizada es estrella, enviando los datos obtenidos a un *gateway* mediante una conexión unidireccional o bidireccional. En la topología puede existir más de un *gateway*. Cuando hay más de un *gateway* en el rango de cobertura puede que todos reciban la trama y si el servidor recibe más de un paquete con el mismo número de secuencia, puede eliminar las duplicadas [63].

Un aspecto importante es la limitación del ciclo de trabajo (*duty cycle*) donde los dispositivos no pueden utilizar el canal de comunicación más del 1% del tiempo, lo que repercute en su *data rate*. El *data rate* resultante podría ser inadecuado para muchas aplicaciones, como aquellas que trabajan con recursos multimedia [64].

Otros trabajos previos [65] evalúan la transmisión de la señal en exteriores acorde a tres modelos de pérdidas como: (i) *Free-Space Path Loss* (FSPL), donde la atenuación de la señal depende de la distancia que separa a emisor y receptor y la frecuencia de trabajo; (ii) coeficiente de regresión basado en dos ajustes diferentes, uno en la ciudad de Dortmund (Alemania) y otro en la ciudad de Oulu (Finlandia); y (iii) un *Irregular Terrain Model* (ITM) usado en la distribución de señal de televisión aplicable en el rango de 20 MHz hasta 20 GHz. Pese a que no existe el modelo que se ajuste perfectamente al funcionamiento real de estos sistemas, el modelo FSPL se ajusta mejor al comportamiento real de los sistemas bajo estudio en el medio, aunque el modelo ITM se ajusta muy bien en entornos montañosos. Estudios recientes [66] evalúan el rendimiento de grandes despliegues LoRa en interiores a diferentes alturas en base a las características de los dispositivos, rango de cobertura, consumo energético, fiabilidad, etc., destacando la importancia de la correcta selección de parámetros.

2.5.2. Long-Range Wide Area Network

LoRaWAN es un estándar abierto que define la arquitectura, las capas y los protocolos existentes sobre la capa física LoRa. Es uno de los responsables en el aceleramiento del

despliegue de servicios IoT debido a su robustez en las comunicaciones a grandes distancias siendo respetuosa con la vida de la batería. La desventaja viene dada por los bajos *data rates*, la alta latencia (crítica en ciertos servicios) y escalabilidad [63,67]. En los despliegues LoRa se pueden diferenciar tres tipos de participantes: (a) dispositivos LoRa o nodos, (b) *gateways* y (c) Servidor (ver Fig. 2) [55].

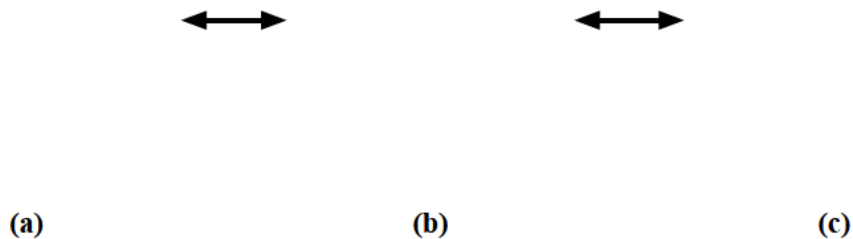


Fig. 2: Dispositivos en redes LoRa: (a) dispositivos LoRa, (b) gateway LoRa, y (c) Servidor.

Los dispositivos LoRa son los encargados de, usando los sensores que llevan integrados en el sistema, captar datos de su entorno y enviarlos al *gateway* por su interfaz inalámbrica. Estos dispositivos se clasifican en tres clases según la gestión que hacen de la ventana de recepción – respetando siempre la restricción del *duty cycle* (1%). Los dispositivos de clase A utilizan ALOHA puro, de forma que envía un mensaje y abre una ventana de recepción pasado un offset de tiempo para recibir ACKs o mensajes *downlink*. Esta clase es la que presenta un menor consumo energético [56]. La clase B extiende la clase A añadiendo la apertura de la ventana de recepción programada de forma periódica. Por último, la clase C mantiene abierta la ventana de recepción todo el tiempo excepto

cuando el dispositivo está transmitiendo [45,48]. Las tres clases pueden coexistir en la misma red y los dispositivos pueden cambiar de una clase a otra, recayendo esta tarea en la capa de aplicación y no en el *gateway* [56]. El *gateway* puede diversificar las recepciones en tiempo, frecuencia y SF, disminuyendo así la probabilidad de pérdidas. A su vez, es capaz de encapsular una trama LoRa en una trama Ethernet para enviarla a la red de *backbone* bajo la arquitectura de comunicación TCP/IP. Según Semtech, el *gateway* es capaz de albergar una red conectada de unos 10^4 dispositivos [68]. Por último, el servidor recibe la información reenviada por el *gateway*. En el caso de existir servidores de aplicación, el servidor reenvía la trama recibida al servidor de aplicación pertinente. En caso contrario, el mismo servidor procesa los datos obteniendo información que posteriormente se podrá emplear para la toma de decisiones [67].

Dependiendo del valor de SF en uso, el *data rate* de LoRaWAN puede variar desde 0,3 kbps hasta 50 kbps [69]. En los casos de bajo *throughput*, el rendimiento está limitado por las colisiones; mientras que en el caso de *throughput* alto, la limitación del *duty cycle* impiden a los dispositivos aumentar la tasa de transmisión de paquetes y estabiliza el rendimiento. Para despliegues con pocos dispositivos LoRa, la restricción del *duty cycle* limita el rendimiento máximo. El *payload* de la trama puede variar de 51 hasta 222 bytes dependiendo del valor de SF [56].

2.6. Unmanned Aerial Vehicles

Los UAVs, también conocidos como drones, han sido adoptados por muchos sectores como la agricultura, monitorización, vigilancia, telecomunicaciones, etc. [70,71]. Estos vehículos pueden clasificarse en función del tipo de vuelo (autónomo o controlado de forma remota), el tamaño, el tipo de hélices (fijas o giratorias), o las capacidades de comunicación. Los UAVs de hélices fijas tienen una mayor autonomía de vuelo, mayores velocidades y diseños más aerodinámicos. Por el contrario, los UAVs de hélices giratorias permiten los despegues y aterrizajes en vertical, mayor estabilidad y la capacidad de flotar sobre puntos estáticos. Respecto a las capacidades de comunicación diferenciamos UAVs se conectan con una estación base; o sistemas multi-UAV, empleados en aplicaciones civiles y militares que consisten en enjambres o formaciones de pequeños UAVs bajo el término Flying Ad hoc Network (FANET). Estos últimos presentan algunas ventajas como menores retardos de comunicación, reducción de costes, alta escalabilidad y fiabilidad [72,73].

FANET es un subtipo del concepto Vehicular Ad hoc Networks (VANET), que a su vez es un subtipo de Mobile Ad hoc Networks (MANET). Como se puede intuir, FANET es específico para dispositivos de comunicación voladores que se conectan con tecnología inalámbrica, como los UAVs. Las diferencias entre FANET y VANET son principalmente: (i) el medio en el que se encuentran los dispositivos; (ii) el rango de movilidad, mayor en redes FANET ya que no dependen de infraestructuras terrestres; y (iii) las restricciones energéticas. Otra particularidad de la redes FANET es la habilidad de poder desplazarse

libremente por todo el espacio disponible en las tres dimensiones del espacio, mientras que las redes MANET apenas tienen movilidad y cambios de velocidad, y las redes VANET presentan movilidad media según la circulación de los vehículos en dos dimensiones [74]. Estas diferencias tienen un impacto directo en los modelos de movilidad que estos dispositivos describen. Otros factores que influyen en gran medida en las redes FANET son la movilidad de los vehículos, velocidad de los dispositivos, cambios en la topología, o limitaciones energéticas de sus dispositivos (véase Tabla 3).

Tabla 3: Comparación de características MANET, VANET y FANET.

Características	MANET	VANET	FANET
Movilidad de los dispositivos	Muy poca (2D)	Poca (2D)	Hélices fijas: Media (3D) Hélices giratorias: Alta (3D)
Velocidad de los dispositivos	Baja (~6km/h)	Media – Alta (~20 – 100km/h)	Hélices fijas: Alta (~100km/h) Hélices giratorias: Media (~50km/h)
Cambios en la topología	Muy pocos	Normal	Alta
Limitaciones energéticas	Media	Pocas	Hélices fijas: Media (hasta 5h) Hélices giratorias: Alta (~15-30 min)

Los modelos de estimación de pérdidas son diferentes para FANET, ya que estos dispositivos se encuentran a una altura considerable y generalmente tanto emisor como receptor están en línea de visión directa al no darse reflexiones debido a los objetos ambientales o difracción debida a los edificios. En función de la posición de los UAVs en las redes FANET, podemos distinguir dos tipos de enlaces que también deberán tenerse en

cuenta en los modelos de propagación: (i) enlace que conecta dos UAVs (UAV-to-UAV, U2U) y (ii) enlaces que conectan un UAV con un nodo terrestre (UAV-to-Ground, U2G) donde pueden darse difracciones y reflexiones (ver Fig. 3) [75].

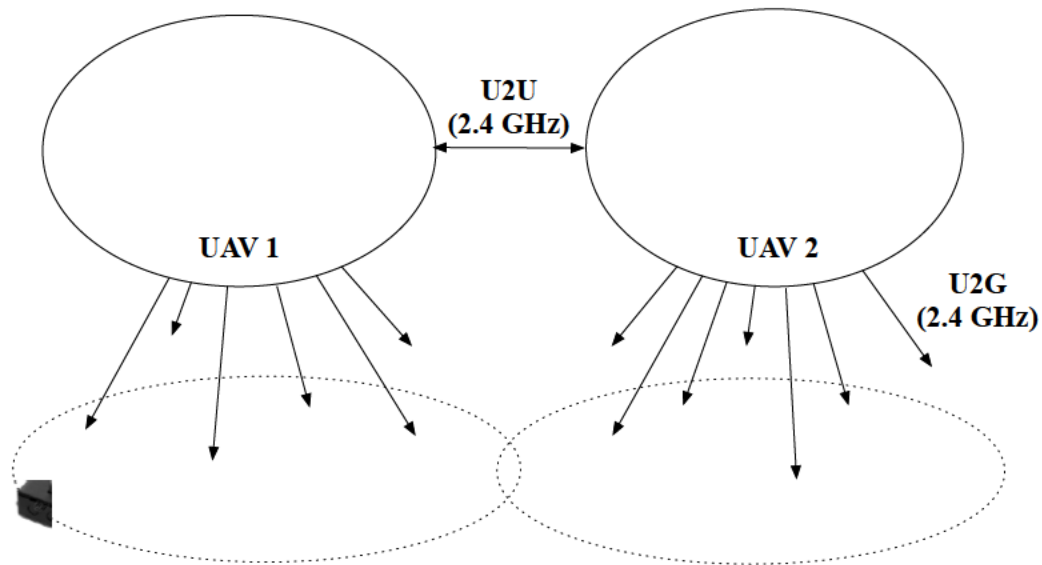


Fig. 3: Ejemplo de redes FANET con enlaces U2U y U2G.

Las redes FANET definen el movimiento de los dispositivos en el aire en base a unos patrones de movilidad especificando cambios en la posición, velocidad y aceleración en un escenario [76]. Algunos modelos han sido adaptados de redes MANET o VANET, mientras que otros modelos han sido específicamente diseñados para redes FANET. Los modelos de movilidad adaptados a redes FANET se agrupan bajo cinco grupos: modelos aleatorios, temporales, grupales, geográficos e híbridos [77].

- Modelos aleatorios: son los más comunes, donde cada UAV selecciona la dirección, la velocidad y el tiempo independientemente del resto de UAVs. Existen diferentes tipos:

- *Random Walk (RW)*, el UAV selecciona la orientación y la velocidad durante un intervalo de tiempo y antes de agotarse el *timer* selecciona de forma aleatoria una nueva orientación y velocidad para el nuevo periodo.
- *Random Waypoint (RWP)*, cada dispositivo selecciona de forma aleatoria el objetivo en el área de cobertura y la velocidad de viaje. Cuando el UAV llega al destino, espera por un tiempo aleatorio y antes de que se agote selecciona aleatoriamente un nuevo objetivo y una nueva velocidad.
- *Random Direction (RD)*, el dron selecciona la dirección y la velocidad hasta el límite del área de cobertura. Una vez alcanza el límite, selecciona una nueva dirección a la que dirigirse hasta chocar con un nuevo límite, repitiendo cíclicamente este proceso.
- Modelos temporales: están basados en ecuaciones matemáticas que dependen del estado actual del nodo en un tiempo dado evitando cambios bruscos en velocidad y dirección.
 - *Gauss-Markov (GM)*, pretende que el dispositivo se mueva suavemente dentro de los límites del área de cobertura. Ha sido utilizado en gran medida para evaluación de prestaciones.
 - *Smooth Random*, reproduce el comportamiento que siguen los vehículos terrestres en la tierra mediante el proceso *Stop-Turn-Go*.
- Modelos grupales dependientes del espacio: el comportamiento de un UAV depende del espacio disponible en el área de cobertura y del comportamiento del resto de UAVs más cercanos.

- *Reference Point Group Mobility* (RPGM) sigue el modelo maestro-esclavo donde los miembros de un grupo siguen al líder de dicho grupo [78].
- *Spatial Correlated Mobility*, donde el comportamiento de un UAV depende de las acciones del resto de dispositivos.
- Modelos geográficos: incluyen modelos con restricciones en las trayectorias como obstáculos en el camino.
 - *Pathway Mobility*, restringe la movilidad del dispositivo en función de un mapa o gráfico.
 - *Obstacle Mobility*, describe la trayectoria del dispositivo evitando los obstáculos existentes en la ruta programada.
- Modelos híbridos: comparten al menos dos características de los modelos anteriormente descritos.
 - *Free-way Mobility*, basado en la situación actual (modelo temporal) y en la posición de nodos cercanos (modelos grupales).
 - *Disaster-Area Mobility*, donde se dan características de muchos modelos de movilidad debido a la urgencia de la situación.

Por el contrario, existen otros modelos de movilidad específicamente creados para redes FANET no extensibles a MANET o VANET debido a las grandes áreas que se requerirían en tierra. Estos modelos son:

- *Semi-Random Circular Movement (SRCM)*, donde cada UAV se mueve respecto a un centro fijo a todos los UAVs con un radio, velocidad y ángulo inicial diferente para cada uno de ellos.
- *Three-Way Random Mobility*, que tiene en cuenta tres estados en la cadena de Markov: ir recto, girar a la derecha y girar a la izquierda. Cuando llega al borde del área de cobertura retorna hacia el interior.
- *Pheromone Repel Mobility*, divide el área de cobertura en pequeñas sub-áreas tratando de cubrirlas todas. Cuando llega al límite retorna al interior con el mismo ángulo.
- *Smooth Turn Mobility (ST)*, captura la movilidad en el espacio libre haciendo trayectorias suaves con un gran radio como los aviones. Este modelo tiene versiones tanto para 2D como 3D, donde pueden fijar la altura del dispositivo constante o variable a lo largo del vuelo.
- *Flight-Plan Mobility (FP)*, en el vuelo de los dispositivos se conoce tanto el punto inicial y el final de antemano. Este modelo no es válido para UAVs autónomos.
- *Multi-Tier Mobility*, son redes de dispositivos inalámbricos voladores que utilizan diferentes tipos de UAVs volando a diferentes alturas.

Como resumen, la Fig. 4 recoge una clasificación general de todos los modelos de movilidad descritos anteriormente.

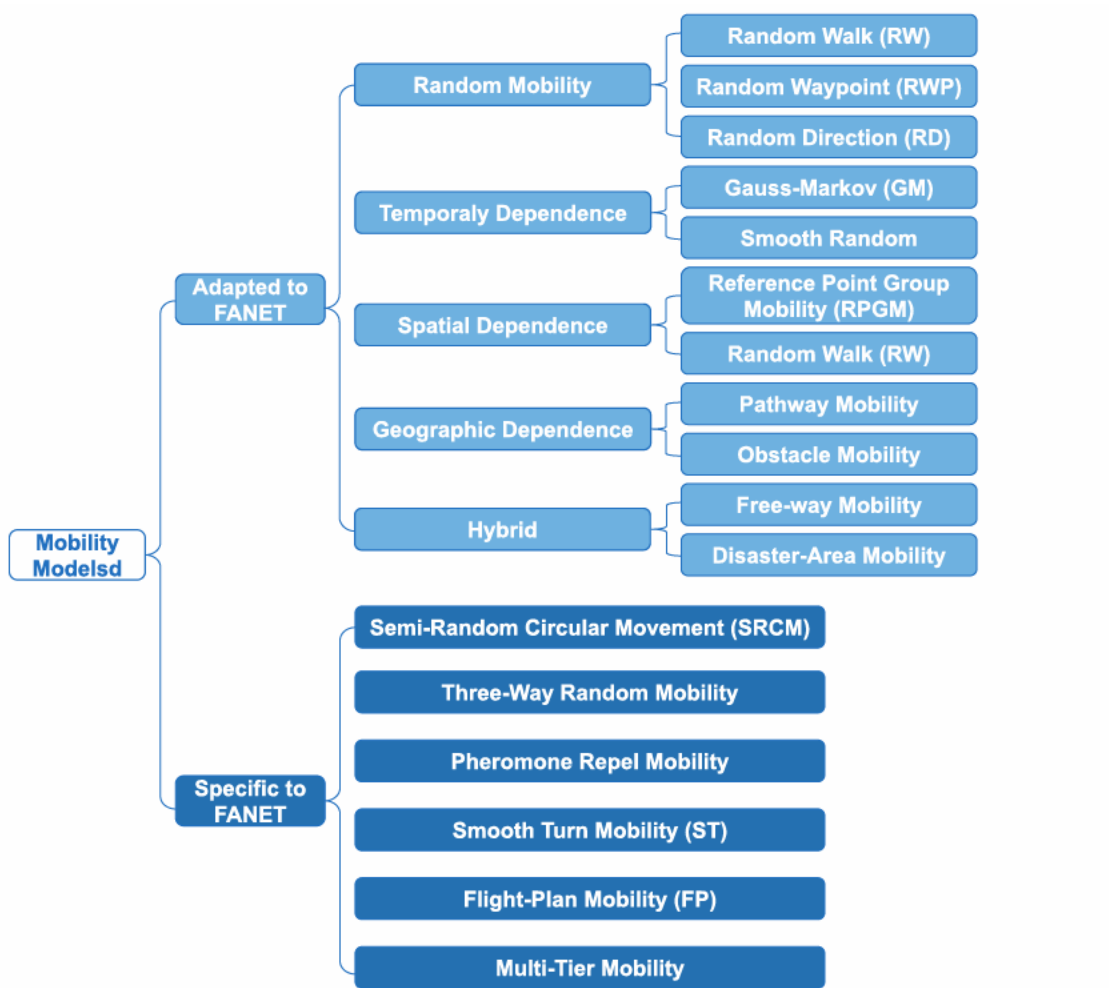


Fig. 4: Clasificación de los modelos de movilidad [79].

2.7. Machine Learning

ML es una rama específica de la AI, definida por Arthur Samuel en 1959 como el estudio que permite a los ordenadores aprender de los datos obtenidos [4]. De una forma breve, los algoritmos establecen una función de coste que debe ir minimizando para encontrar los pesos óptimos que tienden a una predicción más precisa. A mayor volumen de datos de entrenamiento, mayor fiabilidad de la predicción. Existen tres tipos de

problemas de aprendizaje dentro de ML: aprendizaje supervisado (*Supervised Learning*), aprendizaje no supervisado (*Unsupervised Learning*) y aprendizaje por refuerzo (*Reinforcement Learning*) [4].

Cuando hablamos de problemas de aprendizaje es importante analizar qué características (o *features*) son las más importantes para no añadir variables adicionales que aporten ruido al sistema de aprendizaje. Este proceso es conocido como *preprocessing* y pueden darse diferentes implementaciones. Algunas técnicas utilizan métodos de correlación para conocer la semejanza de las variables como *Feature Selection* (FS) o *Recursive Feature Elimination* (RFE), que elimina aquellas con mayor correlación asumiendo que son redundantes. Otras técnicas reducen el número de características por aproximación como *Principal Component Analysis* (PCA) [80,81].

2.7.1. Supervised Learning

En los algoritmos de aprendizaje supervisado los datos recogidos están etiquetados en la fase de entrenamiento. Estos datos son utilizados para predecir de forma precisa a partir del mapeo entre variables y etiquetas. Este proceso se realimenta constantemente para identificar entrada de datos con características similares. Esta técnica se utiliza para, a partir de un conjunto de datos de entrada, llevar a cabo una estimación entre un conjunto de clases diferentes englobadas bajo problemas de clasificación (ver Fig. 5a) y para predecir un resultado futuro mediante los problemas de regresión (ver Fig. 5b). Algunos casos de aplicación de esta técnica son la clasificación entre paciente sano o paciente enfermo,

correo SPAM o no SPAM, predicción de valoraciones de usuarios en escala MOS, estimación de número de clientes por temporada, etc. Existe una gran variedad de este tipo de algoritmos ya que son los más utilizados. Algunos ejemplos son las *Artificial Neural Networks*, Técnicas estadísticas Bayesianas, *Support Vector Machine*, Modelos de Markov, *Decision Trees* (C4.5, ID3, CART, *Random Forest*), *K-Nearest Neighbor* y Clasificadores de conjunto (*Bagging*, *Boosting*), entre otros [4,82].

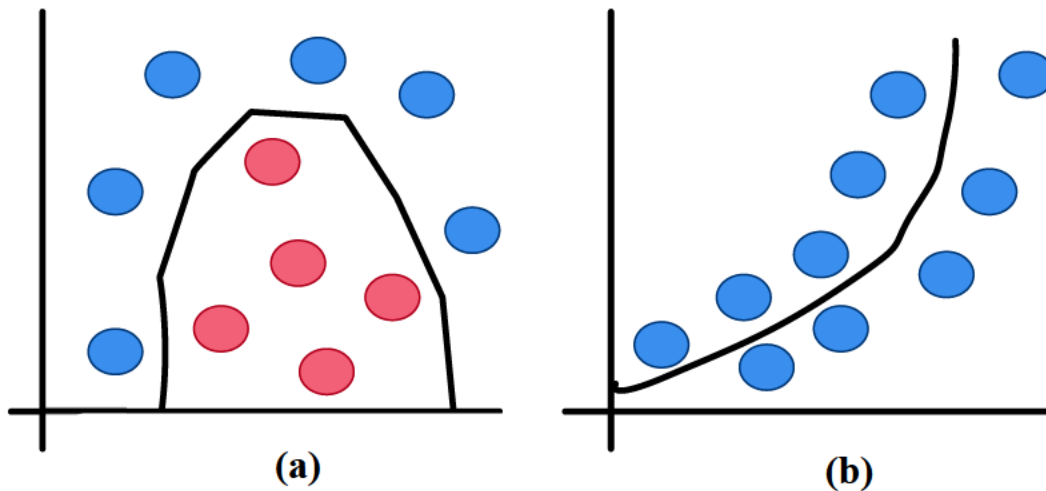


Fig. 5: Aprendizaje Supervisado: (a) Clasificación, (b) Regresión.

2.7.2. Unsupervised Learning

Los conjuntos de datos no etiquetados se gestionan bajo los algoritmos de aprendizaje no supervisado. Esta técnica se utiliza para encontrar los patrones, estructuras o conocimientos ocultos de los datos agrupando las entradas de datos con características similares en diferentes grupos desarrollando técnicas de *clustering* (ver Fig. 6). Los tipos

de algoritmos no supervisados más comunes son *K-Means*, *Hierarchical clustering* o detección de patrones o anomalías [4,83].

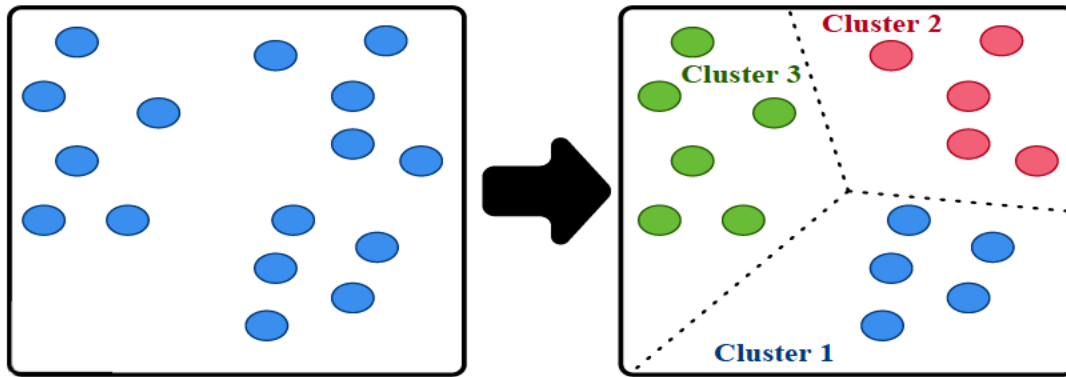


Fig. 6: Aprendizaje no Supervisado.

2.7.3. Reinforcement Learning

Las técnicas de *Reinforcement Learning* interactúan con el entorno para conseguir un objetivo marcado, del mismo modo que los humanos aprenden y llevan a cabo sus tareas cotidianamente. En las técnicas de *Reinforcement Learning* existen uno o varios agentes que interactúan con el entorno maximizando la recompensa acumulada. En cada instante, el agente selecciona la mejor opción basado en una política dentro de un entorno donde cambia su estado, recibiendo una recompensa por su acción que puede obtenerse de forma inmediata o retrasada (ver Fig. 7). De este modo, el agente aprende una política óptima por ensayo y error [4,82].

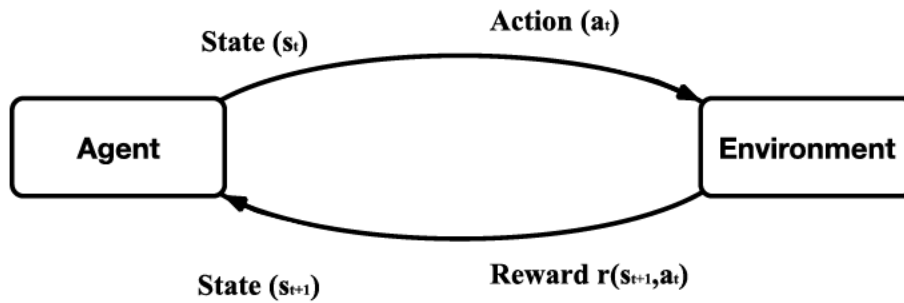


Fig. 7: Aprendizaje por Refuerzo (Reinforcement Learning).

2.8. Conclusión

En este capítulo se introducen una serie de términos de gran relevancia en este contexto de trabajo respecto a métricas de calidad, IoT, LPWAN y ML para una mejor comprensión de los siguientes capítulos. Primero, se lleva a cabo una extensa definición por parte de la ITU-T de los términos de calidad como QoS y QoE, donde QoS es un valor cuantificable mediante métricas objetivas y QoE es un valor más complejo donde entran en juego métricas subjetivas.

A continuación, se presenta el término IoT como un nuevo paradigma que pretende conectar todo lo que nos rodea, introduciendo su contexto histórico cuyo precursor son las etiquetas RFID. Entre los retos más importantes de IoT está la evaluación de prestaciones, la seguridad y la heterogeneidad de dispositivos y tecnologías, que puede abordarse de forma eficiente mediante las capacidades cognitivas presentes en los dispositivos conectados. Dentro de IoT destacan las tecnologías LPWAN por su gran rango de cobertura (hasta 30 km en áreas rurales), gran escalabilidad, bajo consumo energético, bajo coste y bajo *bit-rate*. En esta sección se incluye una tabla comparativa de las diferentes plataformas

disponibles en el mercado que permite escoger la mejor plataforma en función de las necesidades. En nuestro caso, vamos a particularizar sobre dispositivos LoRa que cumplen con las especificaciones de LPWAN.

Por último, se introducen dos términos que son de gran tendencia como los vehículos aéreos UAVs o drones y las técnicas de ML. Los UAVs se pueden comunicar entre sí y con la estación base en tierra creando redes FANET con características propias definiendo modelos de movilidad que determinan el comportamiento de estos vehículos. Por último, las técnicas de ML trabajan con un histórico de datos previo donde se conseguirá mayor precisión conforme mayor sea el volumen de datos. En función de la estructura del histórico de datos, se pueden llevar a cabo diferentes tareas bajo tres grandes problemas de aprendizaje, como aprendizaje supervisado, no supervisado o por refuerzo.

Un Enfoque Innovador para la Evaluación de Rendimiento en IoT

3.1. Introducción

La evaluación de prestaciones en las redes de telecomunicaciones se viene desarrollando desde sus orígenes midiendo el nivel de calidad de una aplicación, servicio o protocolo en la red. Para ello, se han definido una serie de métricas para cuantificar el nivel de calidad a partir de conceptos bien conocidos como QoS y, recientemente, QoE. En QoS, las métricas que definen el nivel de calidad son principalmente *delay*, *jitter*, *throughput* y *packet loss*. En QoE las métricas empleadas son más variadas, evaluando la calidad en función de la perspectiva del usuario teniendo en cuenta por ejemplo los codecs de video empleados, los tamaños de las colas de reproducción (*buffers*), el grado de disponibilidad, etc.

Mientras que existe un claro consenso en el uso de QoS/QoE para la evaluación de prestaciones en los servicios multimedia que hacen uso de la infraestructura de telecomunicación, tales como *video streaming*, VoIP, o *Internet Protocol Television* (IPTV), la evaluación de prestaciones en IoT es todavía un desafío. Es en este punto donde se plantean una serie de cuestiones como la optimización de diseño e implementación, la provisión de recursos en un servicio o aplicación IoT, la evaluación de los beneficios de

las aplicaciones o servicios para satisfacer los niveles de calidad, qué suponen las capacidades cognitivas para una compañía o para el usuario final y cómo medir la inteligencia adquirida por el servicio o aplicación. Las respuestas a estas cuestiones requieren una adecuada evaluación de prestaciones en IoT, lo que no es tarea fácil. Estos dispositivos son limitados en recursos energéticos, tolerables a fallos y experimentan mayores retardos que en redes tradicionales. Aunque los modelos de evaluación de prestaciones para redes tradicionales han sido ampliamente estudiados, estamos todavía en los orígenes para el caso de la IoT.

En este capítulo, se presenta un nuevo modelo de evaluación de prestaciones especialmente diseñado bajo un entorno IoT basado dos dominios: coste y beneficio. En el dominio de beneficio se recogen tres componentes: (i) *Quality of Data* (QoD), que evalúa el dato en bruto de cada medición sin ser previamente procesado; (ii) *Quality of Information* (QoI), que valora lo relevante que son los conocimientos adquiridos una vez se han procesado los datos recibidos; y (iii) *Quality of user Experience* (QoE), que tiene incluye tanto métricas de medición de QoS como otros parámetros que pueden tener un efecto sobre la percepción del servicio por parte de los usuarios. Por último, el dominio de coste queda representado por la componente *Quality Cost* (QC), donde se pretende estimar las consecuencias a nivel de recursos disponibles o consumidos.

3.2. Estado de la Técnica en Modelos de Prestaciones para IoT

En las redes IoT la heterogeneidad es muy común, donde diferentes sensores, dispositivos y sistemas están conectados entre si gracias al uso de las redes de comunicaciones. Es evidente que la definición y propuesta de métricas para la evaluación de prestaciones contribuirá en un mejor entendimiento de los sistemas IoT a gran escala, cerrando la brecha teórico-práctica existente.

En la actualidad no existe un modelo de referencia para la evaluación de QoE en aplicaciones IoT, principalmente debido a las diferentes arquitecturas y aplicaciones IoT que ofrece el mercado [84]. Uno de los primeros trabajos en integrar una visión del rendimiento en IoT fue Wu *et al.* [42], donde se clasifican las métricas de rendimiento en dos dimensiones: beneficio y coste (ver Tabla 4). Por un lado, la dimensión de beneficio requiere métricas que evalúen la calidad del dato (QoC), de la información obtenida (QoI) y de la experiencia del usuario (QoE). La primera toma la calidad de los datos adquiridos por los dispositivos a nivel de exactitud, veracidad, integridad o actualidad. La segunda está ligada a la toma de decisiones en función de la información obtenida tras procesar los datos. Por último, la experiencia contempla a nivel subjetivo la aceptación de una aplicación o servicio por el usuario según la ITU [24]. Estas métricas ofrecen una visión más extensa que QoS contemplando los tipos de dispositivos, la capacidad de comunicación de los mismos, la disponibilidad de recursos, etc. La dimensión de coste evalúa la eficiencia en la utilización de recursos del dispositivo como un elemento negativo

(QC). En este caso se valora la carga computacional, consumo energético, eficiencia de almacenamiento, etc. Estas dos dimensiones simula un modelo de análisis coste-beneficio empleados en teoría de decisión existentes en la literatura entre QoS y eficiencia energética [85–87].

Tabla 4: Métricas de evaluación de rendimiento propuestas por Wu et al. [42]. Tabla I en [88].

Capa o Nivel	Métrica	Descripción	Beneficio	Coste
<i>Quality of Data</i> (QoD)	<i>Data accuracy</i>	Precisión	X	
	<i>Data truthfulness</i>	Grado de fiabilidad	X	
	<i>Data completeness</i>	Ratio de datos adquiridos sobre requeridos	X	
	<i>Data up-to-dateness</i>	Validez en la toma de decisión	X	
<i>Quality of Information</i> (QoI)	<i>Quantity (Q)</i>	Cuánta información válida se ha obtenido para la toma de decisión	X	
	<i>Precision (P)</i>	Proporción de información relevante respecto a toda la información capturada por sensores/redes/servicios	X	
	<i>Recall (R)</i>	Proporción de información relevante respecto a toda la información capturada sin sensores/redes/servicios	X	
	<i>Accuracy (A)</i>	Grado de precisión según los requisitos del tomador de decisiones	X	
	<i>Detail (D)</i>	Grado de completación de cara al tomador de decisiones	X	
	<i>Timeliness (T)</i>	Relación entre la información obtenida y el momento de uso para la toma de decisión (inversamente proporcional al retardo)	X	
	<i>Validity (V)</i>	Veracidad de la información obtenida	X	
<i>Quality of user Experience</i> (QoE)	<i>Access</i>	Conexión	X	
	<i>Communication</i>	Mediciones de la calidad de la comunicación, QoS	X	
	<i>Resources</i>	Disponibilidad de recursos, por ejemplo para servicios con altas demandas de computación	X	
	<i>Application</i>	Variables que afectan directamente al usuario (p.e., facilidad de uso)	X	
<i>Quality Cost</i> (QC)	<i>Device utilization efficiency</i>	Grado eficiencia en la utilización de los recursos		X
	<i>Computation efficiency</i>	Grado de eficiencia computacional (recursos de cómputo en la CIoT)		X
	<i>Energy efficiency</i>	Grado de eficiencia energética de los dispositivos en toda su perspectiva: cómputo, acceso a red, comunicación, etc.		X
	<i>Storage efficiency</i>	Grado de habilidad para el almacenamiento físico (y su gestión) de la información		X

Floris *et al.* [84] propone dividir la métrica QoE en cinco dominios asociados a niveles de arquitectura: físico, dispositivos, red, combinación (virtualización e integración con la cloud), aplicación y contexto. El valor de QoE se toma como combinación de calidad en cada dominio independiente. El modelo propuesto es sencillo y está limitado en el número de variables utilizadas. Posteriormente evalúa el modelo experimentalmente con el uso de test subjetivos (ver Tabla 5).

Tabla 5: Métricas de evaluación de rendimiento propuestas por Floris et al. [84]. Tabla II en [88].

Dominio o nivel	Descripción
<i>Physical devices</i>	Medición de la calidad que proporcionan los elementos físicos de la IoE
<i>Network layer</i>	Prestaciones de la infraestructura de red sobre la que opera la IoE
<i>Combination layer</i>	Medición de la interacción con el <i>cloud</i> . Objetos virtuales y su combinación interaccionando con objetos físicos
<i>Application layer</i>	Calidad en términos de control, interactividad, presentación y usabilidad
<i>Context layer</i>	Efecto del contexto de uso, p.e., tipo de dispositivo, tipo de consumidor, precio, ubicación geográfica,...

Li et al. [89] proponen un modelo para medir la QoE en IoT en base a tres niveles: usuario, servicio y contexto (ver Tabla 6). El trabajo planteado por los autores es muy limitado a nivel de aplicabilidad, donde en un análisis posterior, solo emplean métricas como el tiempo de respuesta, tasa de transmisión, estabilidad y precisión.

Finalmente Ikeda et al. [90] modelan la métrica QoE mediante dos tipos de métricas: física y metafísica; y cuatro capas: dispositivos, red, computing e interfaz de usuario (véase Tabla 7). En la capa física se escogen un conjunto limitado de métricas como resolución de las cámaras (dispositivos), ancho de banda (red), prestaciones del servidor en la nube (computing) o diseño de la aplicación (interfaz de usuario). Las métricas metafísicas capturan la capacidad cognitiva de IoT asociándola a obtener información del contexto o

disponer de un servicio flexible. Posteriormente los autores relacionan métricas físicas y metafísicas entre sí y la relación entre métricas metafísicas y la QoE, aunque existe trabajo futuro por desarrollar. Una vez definidas las capas de cada métrica (ver Tabla 7), se modela la relación entre las capas de ambas métricas para calcular el valor de QoE en la aplicación IoT.

Tabla 6: Métricas de evaluación de rendimiento propuestas por Li et al. [89]. Tabla III en [88].

Dominio o nivel	Descripción
<i>User</i>	<i>Métricas: background, experience, expectation, physiological and mental state</i>
<i>Service</i>	<i>Métricas: Transmission layer (p.e., métricas de red), application layer (p.e., conversión, resolución de video, etc.), service layer (prioridades)</i>
<i>Context</i>	<i>Physical context (condiciones hardware and software) y external context (entorno social, cultural, etc.)</i>

Tabla 7: Métricas de evaluación de rendimiento propuestas por Ikeda et al. [90]. Tabla IV en [88].

Métrica	Capa	Métrica	Atributo
Física	Interfaz de usuario	Metafísica	<i>Accuracy</i>
	Computación		<i>Quantity</i>
	Red		<i>Timeliness</i>
	Dispositivo		<i>Recognizability</i>
			<i>Comfortability</i>

3.3. Nuevo Modelo de Evaluación de Prestaciones

Los trabajos de investigación actuales están aún lejos de una visión holística de las prestaciones en dispositivos IoT. Entre las dificultades actuales que frenan el desarrollo de estos trabajos se encuentra la heterogeneidad de redes, ya que estas albergan un amplio volumen de dispositivos con un gran rango de arquitecturas diferentes.

La definición, caracterización y evaluación de métricas para la evaluación de prestaciones contribuirán para un mejor entendimiento de los sistemas IoT a gran escala, cerrando el actual espacio teórico-práctico y permitiendo una planificación de los servicios y aplicaciones desplegados a nivel de sistema. Actualmente no existe ningún modelo de referencia para la evaluación de QoE en aplicaciones de IoT debido principalmente a la falta de una arquitectura IoT de referencia y la gran variedad de requisitos de las aplicaciones o servicios. Este crecimiento acelerado de aplicaciones y servicios viene dado por el incremento de los dominios de uso de la IoT [42,84].

Es importante definir los indicadores o parámetros adecuados y específicos para cada sistema o aplicación, conocidos también como KPIs. Algunos de ellos podrían ser la precisión de los datos, la cantidad de información recibida, el retardo en la comunicación, la opinión subjetiva del usuario o el consumo energético de los dispositivos, entre otros. A su vez, estos KPIs pueden ser organizados en dos dominios: coste y beneficio.

Siguiendo esta idea, se propone en esta tesis doctoral un modelo original y alternativo para cuantificar las prestaciones de las aplicaciones y servicios en la IoT usando cuatro

componentes de calidad, denominados QoD, QoI, QoE (en la dimensión beneficio) y QC (en la dimensión coste). Cada componente tiene su propio modelo y se centra en un nivel diferente, como veremos en las siguientes secciones. El alcance de cada una de las componentes anteriores queda ilustrado en la Fig. 8.

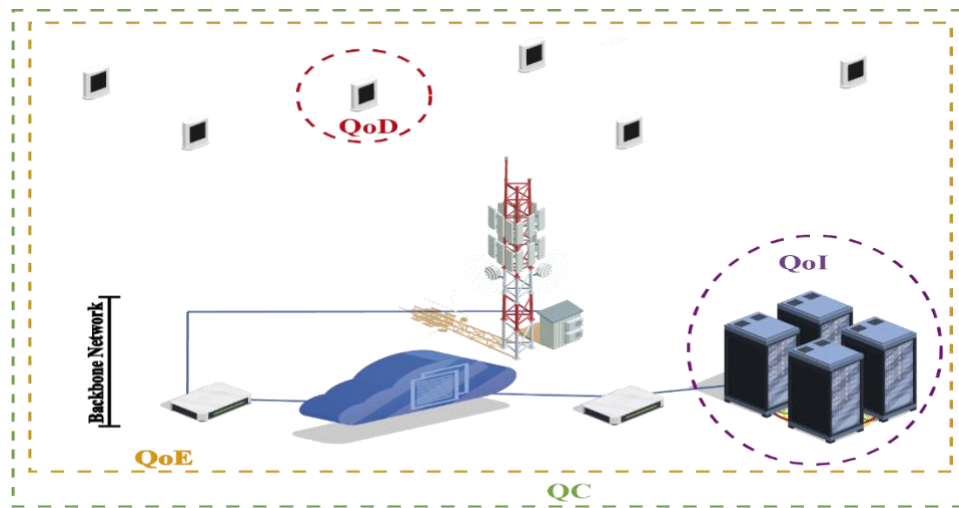


Fig. 8: Alcance de las diferentes métricas de calidad.

A la hora de representar estos niveles de calidad se usará el concepto de periodo de evaluación (T_{eval}). De esta forma, podremos observar el comportamiento global de los dispositivos y el sistema de forma precisa. El valor de T_{eval} puede variar adaptándose a los requisitos de cada aplicación o servicio particular.

3.3.1. Quality of Data

Esta primera componente, QoD, permite medir la calidad de los datos en bruto (*raw data*) capturados por los sensores, distribuidos en la correspondiente zona sensorizada.

Estos datos capturados son muy importantes para evaluar el comportamiento general de los sistemas por el hecho de que los datos erróneos pueden generar información confusa, y por ende, tomar decisiones imprecisas o llevar a cabo acciones innecesarias.

Estos sensores incorporados en los sistemas de medición toman datos del entorno que los rodea. Asumiendo que cada nodo IoT tiene un total de n sensores s , una muestra (S_i) del sistema estará compuesta por el conjunto de todas las mediciones de cada sensor. Esto es $S_i = \{s_1, s_2, \dots, s_n\}$. Los valores obtenidos en el proceso de medición de estos parámetros determinarán el valor final de la QoD. Este modelo de QoD posee aspectos particulares que no comparte con otras contribuciones previas. Principalmente, el valor de esta componente de calidad dependerá de tres KPIs, que serán *Completeness* (C), *Precision* (P) y *Truthfulness* (Tr).

Completeness (C) evalúa la integridad del sistema de sensores y se calcula en función del número de sensores que toman medidas en un dispositivo (s_m) respecto al total de sensores incluidos en los sistemas (s_n). Este parámetro recoge el caso en el que un dispositivo con un número de sensores n no recoja datos de todos los sensores implementados en el sistema, sino de un grupo de ellos (por ejemplo, debido a diferentes frecuencias de muestreo), por lo que dicha muestra S será incompleta. Respecto a una muestra que recoge datos de todos los sensores ($s_m = s_n$), recolecta la cantidad máxima de datos por muestra y obtiene el valor $C_i = 1$. En caso contrario, si una muestra no recoge ningún valor ($s_m = 0$), entonces $C_i = 0$. En cualquier otro caso, donde el número de muestras es mayor que 0 y menor que n ($0 < s_m < n$), el valor de C_i viene dado por (1) como el ratio

de muestras tomadas respecto al total de sensores existente. Al agotarse el T_{eval} , se calcula el valor de C en función del número total de muestras (S) obtenidas durante dicho intervalo de tiempo como muestra (2).

$$C_i = \frac{S_m}{S_n} \quad (1)$$

$$C = \frac{1}{S} \sum_{i=0}^S C_i \quad (2)$$

Precision (P) evalúa la exactitud de las muestras del sensor s_i . Examinando la ficha técnica de los sensores, podemos obtener la precisión del mismo. Por lo general, la medición de los sensores va desde décimas, centésimas o milésimas, hasta incluso decenas, centenas o millares según el sistema de medición, y la escalabilidad de los datos de la variable a medir. El valor de P se calcula para la muestra tomada por cada sensor como el ratio entre la lectura actual y la precisión teórica establecida en la ficha técnica (3). Como ejemplo, cuando la precisión teórica es 0,01 y el dispositivo mide 0,93, el valor $P_i=1$ porque el sensor toma datos hasta el máximo rango de magnitud (centésimas en este caso). Por el contrario, si con la misma precisión teórica el sensor sólo toma valores enteros, significa que el sensor podría tomar valores con mayor exactitud, exactamente de dos órdenes de magnitud mayor (de unidades a centésimas). Por cada orden de magnitud de diferencia, se penaliza con $-0,1$ el valor de P_i . Así, el valor de P se calcula cuando finaliza un T_{eval} como la media de todos los P_i en las S muestras obtenidas durante dicho intervalo de tiempo (3).

$$P = \frac{1}{S} \sum_{i=0}^S P_i \quad (3)$$

Truthfulness (Tr) es un valor binario que penaliza y toma valor 0 si el dato es falso o se encuentra fuera del rango de medición definido en la ficha técnica. En caso contrario, el sensor opera correctamente entre los límites establecidos y toma valor 1. Para conocer el valor de Tr de una muestra, se multiplican los valores de veracidad (T_i) de cada uno de los sensores (n) que conforman los valores en la muestra (4).

$$Tr = \frac{1}{n} \prod_{i=0}^n T_i \quad (4)$$

Por último, el valor de la componente QoD se calcula como se indica en (5) transcurrido un T_{eval} . Como C, P, y Tr son valores comprendidos entre 0 y 1, el resultado de QoD también estará normalizado en el mismo rango [0,1]. Por lo tanto, el mejor resultado que puede darse en la evaluación del *raw data* es 1.

$$QoD = C \cdot P \cdot Tr \quad (5)$$

3.3.2. Quality of Information

Esta componente mide tanto la calidad del dato recibido una vez ha sido procesado y como su relevancia. Esta componente se calcula en el servidor, elemento del sistema que asumimos que lleva a cabo la toma de decisiones para adaptar las operaciones de la red en

función del rendimiento de la misma. En este caso se continúa con el modelo propuesto por contribuciones anteriores [42,91].

Particularizando en nuestra propuesta, la métrica QoI se calcula empleando 7 KPIs como se puede observar en (6). Estas métricas son *Quality* (Q), *Precision* (P), *Recall* (R), *Accuracy* (A), *Detail* (D), *Timeliness* (T) y *Validity* (V). Todos estos parámetros o bien están normalizados en el rango [0, 1] o son variables binarias. Finalmente, el valor de QoI queda comprendido en el mismo rango, donde los valores más cercanos a 1 equivalen a un mayor nivel de calidad.

$$QoI = Q \cdot P \cdot R \cdot A \cdot D \cdot T \cdot V \quad (6)$$

A este nivel, la cantidad de datos recibidos en el servidor es muy significativa porque con un mayor volumen de datos podemos conseguir una mayor eficiencia. En caso contrario, al recibirse poca cantidad de información las decisiones pueden ser inconsistentes. Teniendo en cuenta el hecho anterior, el parámetro *Quantity* (Q) indica si la cantidad de datos recibida en un periodo de evaluación es mejor o peor que otros periodos de evaluación previos. Concretamente, se calcula como la relación entre el número de muestras recibidas (S_{RX}) durante el periodo de evaluación (T_{eval}) más reciente, respecto al número de muestras recibidas en el mejor periodo de evaluación hasta la fecha (S_{max}) (7). Para el primer periodo de evaluación (T_{eval_0}), el parámetro Q siempre va a ser 1, porque no existen registros previos. Para T_{eval_1} , Q será 1 siempre y cuando se reciba un número de muestras mayor o igual que en T_{eval_0} , actualizando el valor de S_{max} al número de

muestras recibido en T_{eval_1} . En caso contrario, el parámetro Q se calcula según la ecuación (7). De esta manera, podemos conocer el comportamiento del sistema en términos de cantidad de datos recibidos en el servidor.

$$Q = \frac{S_{RX}}{S_{max}} \quad (7)$$

Aunque pueda resultar confuso, *Precision* (P) es otra métrica usada en QoI que también se utiliza en QoD pero con diferente alcance y significado. Mientras que en QoD se usa para determinar la precisión de las mediciones del sensor, en QoI se usa para evaluar la precisión de la información obtenida para una población de dispositivos dada. En un sistema con una población de N dispositivos se podría, por diferentes razones (diseño, fallo, limitación de *duty cycle*, errores de configuración, etc.), no recibir mensajes de todos ellos en un T_{eval} . Por tanto la pregunta es, ¿cuál es la cantidad de información mínima necesaria desde diferentes dispositivos (aunque no sean todos) para que el sistema sea preciso? Para resolver esta pregunta hemos extrapolado el concepto de margen de error usado en estadística, donde se toma una muestra de la población y se generalizan los resultados para abarcar a toda la población existente. El margen de error capta esta situación disminuyendo el error conforme aumentan el número de muestras, por lo que el sistema será más preciso cuantas más fuentes envíen información al servidor. Si el servidor recibe muestras de todos los dispositivos existentes en la población en un T_{eval} , entonces $P = 1$. De forma similar, si en un T_{eval} el servidor no recibe muestras de ningún dispositivo, entonces $P = 0$. En caso de que no se corresponda con cualquiera de los anteriores, el servidor recibiría muestras de

un conjunto n de la población, donde $0 < n < N$, y P se calcularía según (8). Para el resto de parámetros de la fórmula planteada, Z representa la *Z-Score* con un 95% de intervalo de confianza tomando valor $Z=1,98$. σ es la desviación estándar, N es la población de dispositivos o el número total de dispositivos desplegados en el sistema, y n es el número de dispositivos que emiten de forma favorable muestras al servidor ($n < N$).

$$P = 1 - Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}} \quad (8)$$

Para calcular la métrica *Recall* (R), nos desviamos de la idea de contribuciones anteriores para adaptarla a escenarios donde entran en juego nuevos enfoques como IoT. Mientras Wu *et al.* [42] definen *Recall* como la fracción de información relevante obtenida a la respuesta de una consulta, en este trabajo se pretende utilizar esta métrica para tener una visión de qué porción de área o zona está siendo monitorizada. Para ello, el área sensorizada se divide en pequeñas celdas y el valor de *Recall* será mayor conforme más celdas entren en juego. En el caso de que se reciban muestras de todas las celdas sensorizadas existentes en el área, entonces $R = 1$. Si no se recibe ninguna muestra de ninguna celda, entonces $R = 0$. En cualquier otro caso, R se calcula utilizando (9), donde r_{jk} es un valor binario que indica si se han recibido muestras de la celda jk ($r_{jk} = 1$) o no ($r_{jk} = 0$). El objetivo por ende es tratar de cubrir la máxima área posible. La Fig. 9 representa un ejemplo de cómo se distribuirían las celdas en una cuadrícula, siendo J y K el número de filas y de columnas respectivamente.

$$R = \frac{1}{J \cdot K} \sum_{j=1}^J \sum_{k=1}^K r_{jk} \quad (9)$$

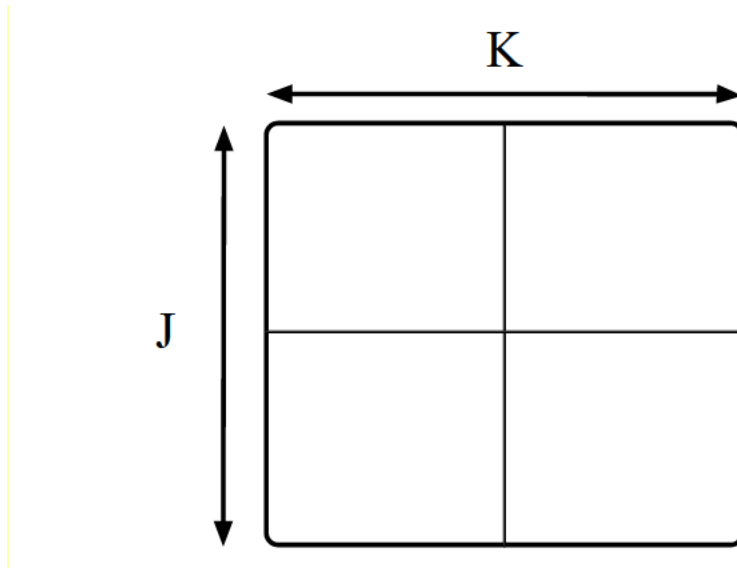


Fig. 9: División de celdas en el área de cobertura.

La métrica *Accuracy* (A) podría tener diferentes interpretaciones. En nuestra propuesta se define como la rigurosidad de la información medida respecto la información real. Este concepto puede variar bastante entre servicios, ya que por ejemplo en envíos de contenido multimedia puede ser utilizado para comparar la calidad recibida en el destinatario respecto a la calidad enviada por el origen.

La métrica *Detail* (D) determina si en un T_{eval} la información recibida en el servidor para cada uno de los parámetros de medición es suficiente, puesto que se deben recibir una cantidad de muestras mínimas de cada parámetro para llevar a cabo una toma de decisiones precisa. Este es un dato binario donde en un T_{eval} , $D = 1$ si el número de muestras por cada

sensor es mayor que el mínimo establecido (*threshold*), y $D = 0$ si el número de muestras no supera dicho *threshold*.

La siguiente métrica, *Validity* (V), es una variable binaria que se empleará para confirmar que en un T_{eval} se han recibido muestras de un número mínimo de dispositivos diferentes. Ese valor mínimo se adapta también de la estadística y queda representado como se indica en (10), con un intervalo de confianza del 95% ($Z=1,98$), σ como desviación estándar, N siendo la población total de dispositivos y permitiendo un error máximo del 5%. Así, si el servidor recibe datos de un número de dispositivos $n > n_{min}$, la métrica $V = 1$, y en caso contrario, $V = 0$.

$$n_{min} = \left(\frac{Z^2 \sigma^2 N}{Z^2 \sigma^2 + e^2 (N - 1)} \right) \quad (10)$$
$$if \ n > n_{min}: V = 1; else: V = 0$$

Por último, *Timeliness* (T) será la métrica empleada para penalizar los sistemas si la información se obtiene de forma tardía, penalizando en función de la demora de tiempo. Se puede considerar un $T = 1$ cuando la información está disponible antes del plazo de entrega o bien, cuando el sistema no es sensible al retraso.

3.3.3. Quality of user Experience

Al igual que QoD y QoI, la componente *Quality of user Experience* (QoE) es fundamental para desarrolladores y operadores. Desde su primera aparición para evaluar los servicios multimedia, su popularidad se ha visto incrementada en múltiples sectores de

diferente índole. Partiendo de la QoS y utilizando KPIs tradicionales como *bandwidth*, *delay*, *jitter* y *packet loss*, la QoE puede ser obtenida a partir de un gran conjunto de parámetros dada su propia idiosincrasia. Tanto los parámetros objetivos heredados de QoS como los emergentes parámetros subjetivos basado en la opinión de los usuarios, forman parte de la métrica que evalúa la experiencia percibida a la hora de utilizar una aplicación o servicio.

En el modelo planteado por esta contribución, la componente QoE queda definida en (11) como producto de los parámetros *delay*, *jitter*, *Packet Delivery Rate* (PDR), *throughput* y *Gateway Availability* (GW_{Av}). El parámetro *delay* es el retardo medio extremo a extremo entre los dispositivos finales y el servidor. El parámetro *jitter* es la fluctuación del retardo medio entre los paquetes que atraviesan la red. PDR es el porcentaje de paquetes que atraviesan la red de forma satisfactoria hasta el servidor. El *throughput* extremo a extremo indica la tasa media de bps recibidos en el servidor durante un T_{eval} . Y el GW_{Av} representa el porcentaje de tiempo que un *gateway* tiene la interfaz Ethernet activa y disponible para reenviar tráfico por la red tradicional hasta el servidor en un periodo T_{eval} (12); donde M es el número total de paquetes enviados en un T_{eval} y t_m es el tiempo que utiliza la interfaz Ethernet el paquete m que es enviado por el *gateway* hacia el servidor dentro de un T_{eval} .

Todos estos parámetros que definen la componente QoE están normalizados en el rango $[0, 1]$, donde el 1 es el valor óptimo (mínimo *delay* y *jitter*; máximo *throughput*, PDR y GW_{Av}). Además, destacar que tanto PDR y *throughput* están normalizados respecto

al máximo valor de esta métrica obtenida en T_{eval} anteriores; mientras que *delay*, *jitter* y GW_{Av} están normalizadas respecto al mínimo retardo, variación de retardo y uso de la interfaz Ethernet del *Gateway*, respectivamente. En el caso de que en un T_{eval} alguna de estas métricas mejore su mejor valor obtenido hasta el momento (por ejemplo, se alcanza por primera vez una PDR mejor que las obtenidas en tiempos de evaluación previos), el parámetro adquirirá valor 1 y será utilizado como referencia en los siguientes periodos de evaluación.

$$QoE = jitter \cdot delay \cdot PDR \cdot Thr_{bps} \cdot GW_{Av} \quad (11)$$

$$GW_{Av_i} = 1 - \frac{1}{T_{eval}} \sum_{m=1}^M t_{m_i} \quad (12)$$

3.3.4. Quality Cost

Respecto a los trabajos previos, la componente *Quality Cost* (QC) aporta una característica diferenciadora en la evaluación del rendimiento. Como en cualquier desarrollo, cada ganancia de rendimiento puede tener su contrapartida negativa, y tratamos de reflejar este hecho mediante esta componente. La idea fundamental es, teniendo en cuenta las situaciones en las que los dispositivos utilizan un mayor número de recursos (energía, memoria, seguridad, etc.) el QC debe disminuir, de forma que el objetivo sea dotar de las mejores prestaciones con los mínimos recursos posibles.

Proponemos modelar QC como el producto de dos métricas (13). Por un lado, el consumo de energía o *Energy Consumption* (EC) necesario para realizar una acción o un conjunto de ellas por parte de todos los dispositivos bajo evaluación; y por otro lado que se respete la limitación de *duty cycle* (1%) en el caso de existir este tipo de limitaciones temporales.

En general, los transceptores de numerosos dispositivos empleados en la IoT tienen cuatro estados posibles: transmisión, recepción, *idle* (o inactividad) y apagado. Para calcular el EC tendremos en cuenta el tiempo consumido en cada estado del transceptor y el consumo asociado a cada estado (14). Como en parámetros anteriores, $EC = 1$ si el consumo en el T_{eval} actual es mejor que cualquiera de los anteriores. En caso contrario, el valor de EC queda normalizado en el rango $[0, 1]$ respecto al mejor EC en T_{eval} anteriores. Sobre la limitación de *duty cycle* que algunos dispositivos presentan, se comprueba que los dispositivos no utilicen el canal inalámbrico para el envío de datos más de lo que cada estándar define (por ejemplo, 1%). En este sentido, la métrica *Interface Use* (IU) es binaria, y tomará el valor $IU = 1$ si los dispositivos cumplen la restricción temporal y $IU = 0$ en caso contrario. Esta componente QC se podría ampliar con otras métricas según el servicio o aplicación bajo análisis.

$$QC = EC \cdot IU \quad (13)$$

$$EC_{not\ normalized} = \sum_{\forall\ transceiver\ states, nodes} c_{state} t_{state} \quad (14)$$

3.4. Conclusión

La naturaleza de IoT conduce a la necesidad de un nuevo enfoque para evaluar el rendimiento de los servicios y aplicaciones en este paradigma capturando sus peculiaridades. Esto se traduce en nuevos modelos de rendimiento que definen la calidad, el nivel de conocimiento, el nivel de inteligencia, el consumo, etc. Hemos abordado este reto proponiendo un método original de evaluación del rendimiento cuya medición se compone de cuatro componentes de calidad, en general denominadas QoX y concretamente QoD, QoI, QoE y QC. Cada componente se refiere a una dimensión diferente, desde los datos en bruto hasta la eficiencia, incluyendo las clásicas métricas de rendimiento de la red. Es más sencillo, visual y eficiente desde el punto de vista operacional disociar el efecto de los diferentes parámetros y métricas empleadas cuando se evalúa el rendimiento de un servicio.

3.5. Contribuciones Relacionadas a este Capítulo

J.M. Martínez-Caro, M.-D. Cano, “A novel holistic approach for performance evaluation in Internet of Things”, *Int. J. Commun. Syst.* (2020), pp. 1–24. <https://doi.org/10.1002/dac.4454>
Factor de Impacto JCR: 1,319 (Q3), Rank: 198/266 in Engineering, Electrical & Electronic.

J.M. Martínez-Caro, M.-D. Cano, “Definición de métricas de calidad en IoT basadas en coste-beneficio”, *In I Jornadas Doctorales UPCT. Salón de Grados de la Escuela Técnica Superior de Ingeniería Industrial, Universidad Politécnica de Cartagena. Lunes 16 de abril de 2018.* Comunicación oral.

Caso de Estudio: Evaluación de prestaciones en IoT con LoRa/ LoRaWAN empleando el modelo QoX

4.1. Introducción

Debido a que no existe actualmente ningún estándar para la evaluación de prestaciones de una aplicación o sistema en el paradigma de la IoT, proponemos en este capítulo el uso del modelo de evaluación de prestaciones presentado en el capítulo anterior y que tiene en cuenta diversas componentes de calidad (QoD, QoI, QoE y QC) a las que en conjunto denominaremos QoX. Pese a que algunas tecnologías IoT se encuentran todavía en fase de investigación, es importante resolver cuanto antes la definición del método de evaluación más apropiado ya que se prevé un despliegue masivo a corto y medio plazo [92]. Establecer un modelo de prestaciones puede orientar, tanto al desarrollador como al usuario, sobre el funcionamiento de una aplicación o servicio, y a partir de ahí buscar los métodos para mejorarlo. Hay que tener siempre presente que no todas las aplicaciones o servicios tendrán las mismas necesidades, y por ello el modelo de evaluación de prestaciones debe ser flexible a cada escenario.

En un ecosistema denso y de gran escala como IoT, entran en juego diferentes tecnologías como LoRa y LoRaWAN para dar cabida a diferentes soluciones sociales y técnicas. IoT está cambiando Internet hacia una red mucho más heterogénea en la que se simplifica la transmisión y recepción de datos, además de provocar un aumento exponencial de dispositivos conectados [11]. Hemos considerado por tanto que un buen caso de uso sobre el que evaluar la bondad del modelo de evaluación QoX propuesto es el de un sistema de monitorización de la calidad del aire. Este sistema se implementará utilizando tecnología LoRa/LoRaWAN y se considerarán varios escenarios que serán evaluados mediante simulación por ordenador. En concreto, se hará uso y se ampliará la funcionalidad de diferentes herramientas software de tipo *Open Source* que permiten ajustar la simulación conforme a las necesidades del investigador y que se describirán en las siguientes secciones. El sistema de monitorización de calidad del aire será implementado de dos formas diferentes. En primer lugar, se estudiará la QoX de este servicio en un escenario en el que los dispositivos LoRa se fijan de forma estática en el área monitorizada. Se llevará a cabo un estudio bajo tres posibles entornos geográficos (urbano, suburbano y rural) y con dos escenarios diferentes en cuanto a comportamiento de la red. En segundo lugar, se estudiará la QoX de este servicio de monitorización en un escenario con movilidad, donde los dispositivos LoRa IoT de medición de calidad del aire no estarán fijos sino embebidos en drones. Se estudiará el impacto del número de drones/dispositivos LoRa y la velocidad a la que estos se desplazan sobre el área monitorizada usando el modelo de evaluación de rendimiento propuesto en el capítulo anterior.

4.2. Test-bed

La simulación es la solución óptima para evaluar el comportamiento del modelo de prestaciones que analizará el rendimiento de los dispositivos LoRa bajo diferentes singularidades. También será útil como punto de partida para contrastar los resultados con los obtenidos en futuras líneas mediante experimentación.

Para llevar a cabo la simulación indicada, se utiliza OMNeT++ [5], INET [6] y FLoRa [7]. En primer lugar, OMNeT++ es una biblioteca y un marco de simulación programado bajo el lenguaje de programación C++. Está basado en componentes, destacando por ser extensible y modulable, y donde junto con INET permite construir simuladores de redes cableadas, inalámbricas y móviles. Por último, FLoRa es una herramienta de simulación de código abierto basada en los componentes de OMNeT++ e INET. Esta herramienta proporciona un entorno de simulación completo para la tecnología LoRa que debe importarse dentro de OMNeT++. Una vez importado, se aplican los ajustes apropiados al escenario haciendo uso de los múltiples módulos existentes. En este caso de estudio, hemos incorporado nuevas características al entorno FLoRa para desarrollar diferentes tareas como leer los *datasets*, adaptar la topología o los parámetros LoRa, división del área en celdas, etc.

En la arquitectura definida por LoRaWAN, modificaremos la implementación por defecto dividiendo el nivel de aplicación en dos sub-módulos denominados Read y SimpleLoRaApp. El primero leerá los datos de un *dataset*, manteniendo las lecturas fallidas como operaciones naturales del sensor. El segundo enviará los datos recibidos del módulo

Read a la capa inferior teniendo en cuenta los parámetros de LoRa y se calculará el rendimiento generado por cada aplicación (estimado en pkts/s y bits/s). Cuando la trama LoRa está lista, será encapsulada por la interfaz LoRa en un mensaje RadioFrame y enviada por la interfaz inalámbrica al *gateway*.

Cuando el *gateway* recibe el mensaje RadioFrame en la capa física, determina si la recepción ha sido exitosa, y de ser así, desencapsula la trama LoRa de RadioFrame. La recepción es satisfactoria si y solo si la señal recibida en recepción es superior a la sensibilidad del dispositivo que varía según el parámetro SF. La señal recibida dependerá de FSPL según el modelo Okumura-Hata, distinguiendo tres entornos: urbano (16), suburbano (17) y rural (18). Cabe destacar que los parámetros que estiman las pérdidas por espacio libre están recogidos en la Tabla 8. Si la trama LoRa se recibe correctamente, el *gateway* vuelve a encapsular la trama LoRa en una trama EthernetIIFrame y la reenvía a través de la red troncal (o *backbone network*) hasta el dispositivo Servidor.

Tabla 8: Parámetros de formulación en el modelo Okumura-Hata.

Parámetro	Descripción
f	Frecuencia de trabajo en LoRa (banda ISM).
h_m	Altura del dispositivo final.
h_b	Altura del <i>gateway</i> .
d_m	Distancia entre dispositivo emisor y el <i>gateway</i> .
$a(h_m)$	Factor de corrección.

$$a(h_m) = 3.2(\log_{10}(11.75 \cdot h_m))^2 - 4.97 \quad (15)$$

$$L_{urban} = 69.55 + 26.16\log_{10}f - 13.82\log_{10}h_b - a(h_m) \\ + (44.9 - 6.55\log_{10}h_b)\log_{10}d_m \quad (16)$$

$$L_{sub-urban} = L_{urban} - 2\left(\log_{10}\left(\frac{f}{28}\right)\right)^2 - 5.4 \quad (17)$$

$$L_{rural} = L_{urban} - 4.78(\log_{10}(f))^2 + 18.33\log_{10}(f) - 40.94 \quad (18)$$

Por último, será el Servidor el destinatario final de todos los mensajes generados por los dispositivos LoRa. Su misión es recoger dichos datos y procesarlos para obtener información. En este dispositivo hemos creado dos módulos adicionales con el objetivo de mejorar la capa de aplicación: (i) el módulo `CommunicationParameters` que calcula las métricas de la componente QoE extremo a extremo de la comunicación (desde su origen en los dispositivos LoRa hasta su llegada al Servidor); y (ii) el módulo `Processing`, que, a su vez, tiene dos tareas bien diferenciadas. Por un lado, recibe, almacena y calcula tanto las prestaciones de los módulos intermedios de la comunicación mediante las métricas de QoS, como los valores de QoD calculados en los dispositivos IoT/LoRa a partir de los datos en crudo. Por otro lado, calcula el promedio de los valores de QoD, QoI, QoE y QC pasado el periodo de evaluación, conocido como T_{eval} y fijado acorde a la aplicación dada.

Para este caso en concreto se ha establecido un T_{eval} de 500s. El tiempo total de simulación será de 360000s, equivalente a 100 horas de funcionamiento del sistema de monitorización. Así, durante este periodo, se simula una red de estaciones que miden la calidad del aire. Estas estaciones envían pequeñas cantidades de datos no prioritarios utilizando la tecnología LoRa y la arquitectura LoRaWAN para la comunicación entre dispositivos IoT/LoRa y Servidor pasando por el *Gateway* y la red (véase Fig. 10). Los parámetros leídos en relación con la calidad del aire son: NO, NO₂, NO_x, PM10, PM2.5, velocidad del viento, temperatura, dirección del viento y humedad, donde NO son diferentes óxidos de nitrógeno y PM hace referencia a “*Particulate Matter*” (materia particulada) de diferentes tamaños. El envío de tramas por parte de los dispositivos LoRa seguirá una distribución aleatoria exponencial de media 100s. El conjunto de datos corresponde a mediciones reales (no simuladas) obtenidas de [93].

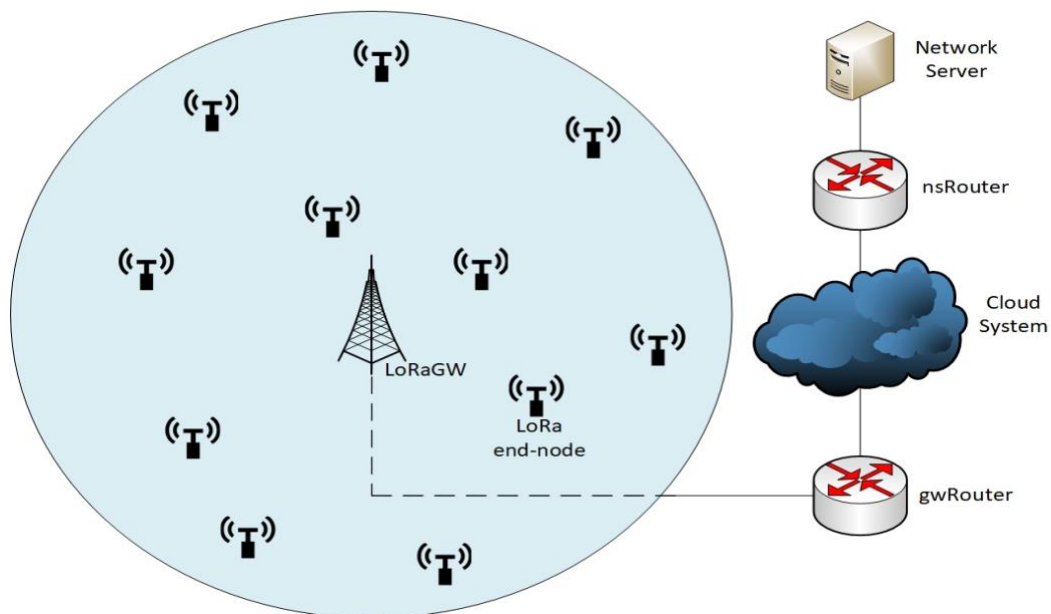


Fig. 10: Red LoRa con dispositivos IoT/LoRa y Servidor pasando por Gateway y red.

Para cada escenario se ha realizado una serie de 10 simulaciones y se han computado las medias para cada componente con el objetivo de normalizar la componente aleatoria durante la generación de paquetes en cada dispositivo LoRa.

4.3. Dispositivos Estáticos

En esta sección se detallan las características de los despliegues LoRa emplazados en un lugar fijo cuyos dispositivos no presentan ningún patrón de movilidad. La aplicabilidad de estos sensores en entornos no móviles es extensa, pudiéndose utilizar en labores de agricultura, control medioambiental, hidrología, geología, infraestructuras, etc.

4.3.1. Despliegues

En la evaluación del rendimiento del escenario estático consideramos dos posibles situaciones. Primero, un escenario estático con retrasos y fallos sólo en la parte de comunicación inalámbrica LoRa, cuyas pérdidas se deben a las características inherentes del canal inalámbrico presentadas por el modelo Okumura-Hata. Segundo, un escenario estático con retrasos y fallos tanto en la red LoRa como en la red troncal. En este segundo caso, se define una probabilidad de pérdidas del 10% en la red cableada y un retardo aleatorio adicional entre 0 y 100 ms.

Para cada una de estas situaciones se miden las métricas de las distintas componentes de calidad (QoD, QoI, QoE y QC). Cada situación del escenario estático tiene sus propias características de implementación (véase Tabla 9).

Tabla 9: Parámetros de simulación para dispositivos estáticos.

Parámetro	Valor
<i>Bandwidth (BW)</i>	125000 Hz
<i>Coding-Rate (CR)</i>	4/5
<i>Tiempo entre paquetes</i>	Distribución aleatoria exponencial de media 100 s
<i>Número de celdas</i>	5x5
<i>Altura de las estaciones</i>	5 m
<i>Altura del gateway</i>	49 m
<i>Entornos</i>	Rural, suburbano o urbano
<i>d_{max}, d_{min} rural</i>	21029.57 m, 558.6 m
<i>d_{max}, d_{min} suburbano</i>	12148.93 m, 999.71 m
<i>d_{max}, d_{min} urbano</i>	6824.47 m, 457.09 m

Las zonas rurales se consideran aquellas fuera de las ciudades en las que apenas hay edificios. Estos entornos son los mejores para analizar la propagación de la señal ya que el FSPL es mínimo. En las topologías rurales, los dispositivos LoRa se situarían como muestra la Fig. 11, con una distancia media de 11,52 km y una distancia máxima de 21,030 km entre dispositivo LoRa y gateway. Un entorno suburbano se refiere a ciudades pequeñas o barrios cercanos a ciudades con una menor densidad de población. Un ejemplo de este despliegue se muestra en la Fig. 12, donde la distancia media entre dispositivos LoRa y gateway es de 6,57 km y la máxima de 12,149 km. Por último, el entorno urbano es el que presenta una mayor densidad de población y graves pérdidas debido a la alta concentración de edificios y la consiguiente atenuación de la señal. A causa de los factores anteriores, el

área de cobertura es menor, por lo que los dispositivos conectados se encuentran más cerca entre sí. Como se ilustra en la Fig. 13, la distancia media y máxima en este entorno entre dispositivos LoRa y el gateway es de 3,95 km y 6,824 km, respectivamente. Las distribuciones geográficas mostradas en las Fig. 11, Fig. 12 y Fig. 13 representan cómo se podrían ubicar estos dispositivos en el área disponible sobre los diferentes entornos, donde la marca anaranjada representa al gateway responsable de recopilar la información. Cabe destacar que dichas imágenes no se corresponden con el emplazamiento exacto de los dispositivos LoRa ya que este caso de estudio se trata de forma exclusiva mediante simulaciones por ordenador.

La asignación de parámetros LoRa (SF y TP) se lleva a cabo en función de la distancia de separación entre el dispositivo LoRa y el *gateway*. A mayores distancias de separación, valores de SF menores y de TP mayores. A menores distancias de separación, valores de SF mayores y de TP menores. Los paquetes en un dispositivo LoRa se generan siguiendo una distribución aleatoria exponencial de media 100s y se utilizará una malla en el área monitorizada de 5x5 celdas.

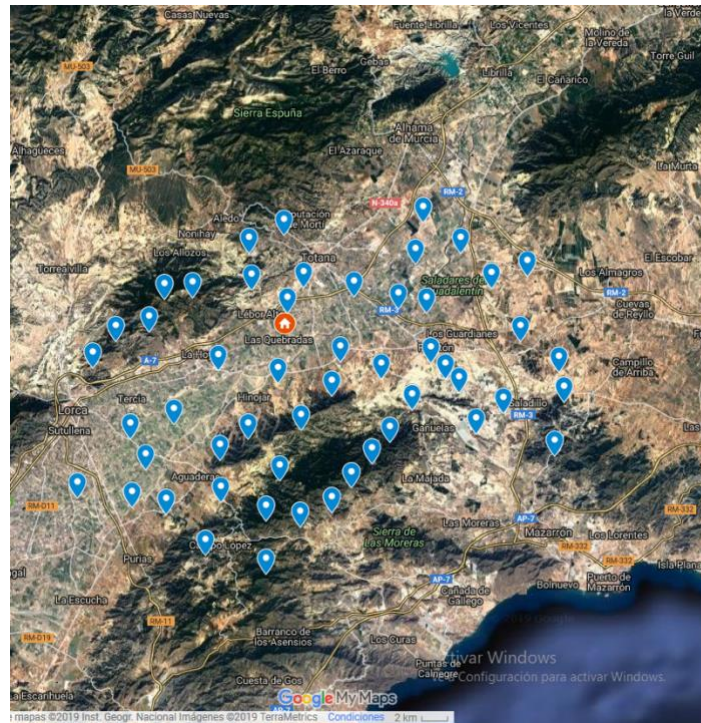


Fig. 11: Ejemplo de despliegue de un entorno rural.

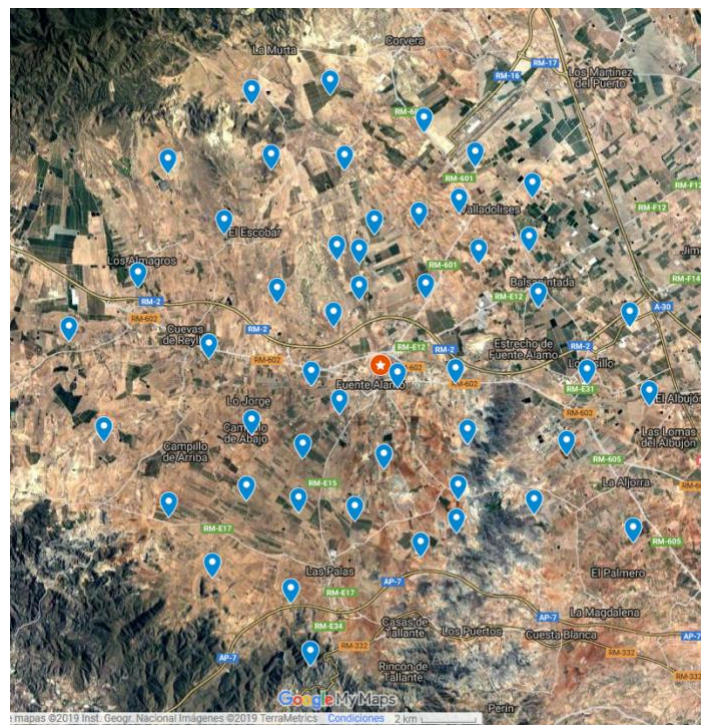


Fig. 12: Ejemplo de despliegue de un entorno suburbano.

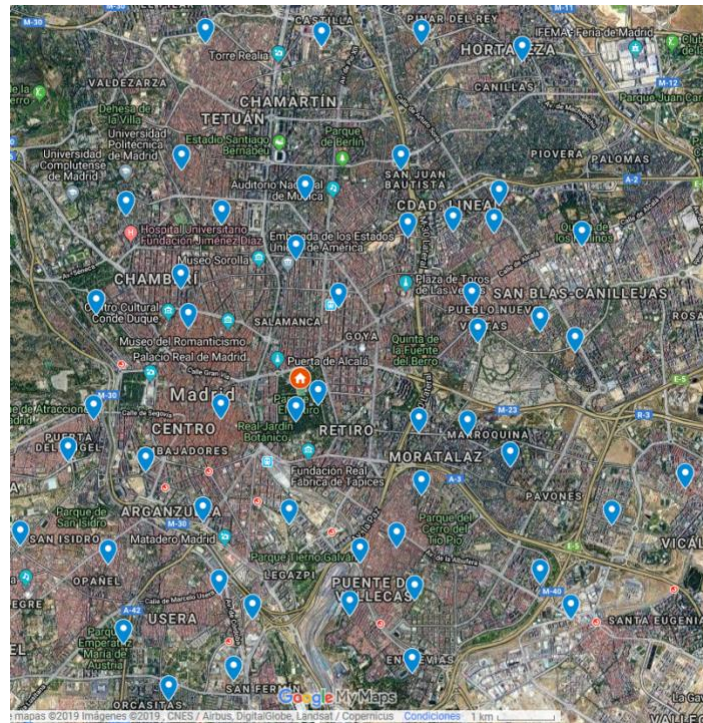


Fig. 13: Ejemplo de despliegue de un entorno urbano.

4.3.2. Resultados en escenario estático

Caso 1: Condiciones ideales en la red troncal

El entorno urbano queda representado por la Fig. 14 mediante sus componentes QoD, QoI, QoE y QC. Los valores alcanzados en QoD muestran que el 60% de los dispositivos finales funcionan correctamente, mientras que el 40% presenta algún tipo de fallo en términos de integridad y precisión en sus sensores debido principalmente a las características inherentes del *dataset* seleccionado. Las dos métricas que recogen peor resultado son QoI y QoE. Respecto a QoI, los parámetros que mayor impacto tienen en la degradación de prestaciones son *Recall* y *Quantity* como se observa en la Fig. 15, véase

(6). Dentro de la cuadrícula virtual dispuesta en el área monitorizada, hay varias celdas desde las que no se recibe información en el servidor, por lo que disminuye el valor de *Recall*. El ratio de paquetes recibidos en el servidor es bajo, lo que disminuye el parámetro *Quantity*. La componente con peores resultados es QoE, donde la Fig. 15 muestra los valores de los KPIs que degradan esta componente, véase también (11). En esta figura, se observa que GW_{Av} , *jitter* y *delay* obtienen valores con un gran rendimiento, cerca del 90% de los valores óptimos. Sin embargo, PDR y Thr_{bps} tienen valores muy bajos, lo que hace disminuir el valor de esta componente de calidad debido principalmente a las notables pérdidas de paquetes en este entorno (urbano). Dentro de la componente QC, véase (13), el parámetro IU mantiene su máximo valor durante toda la simulación ya que todos los dispositivos de la red LoRa respetan la restricción de *duty cycle* (1%). Dado que IU es 1 y que QC se mantiene constante en 0,8 durante toda la simulación, se puede visualizar que el consumo de energía se mantiene cercano al 80% del óptimo para este entorno. Este aspecto es invariante respecto a los entornos de simulación representados, ya que el dispositivo envía el paquete al destinatario acarreado un consumo energético se reciba o no dicho paquete en el servidor.

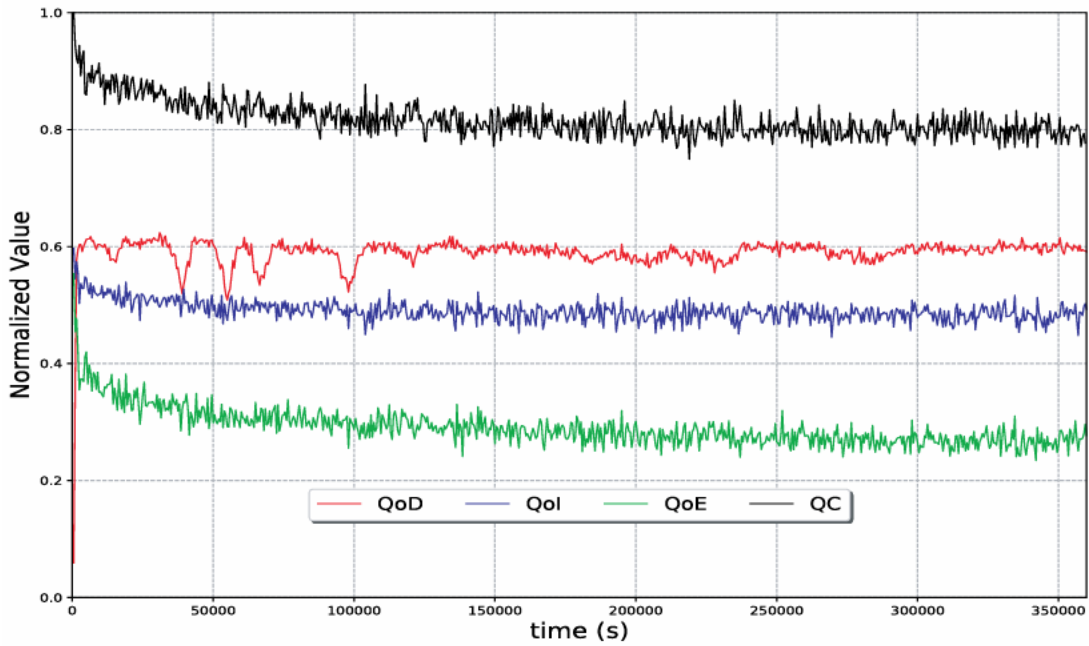


Fig. 14: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno urbano y nodos fijos en el caso 1.

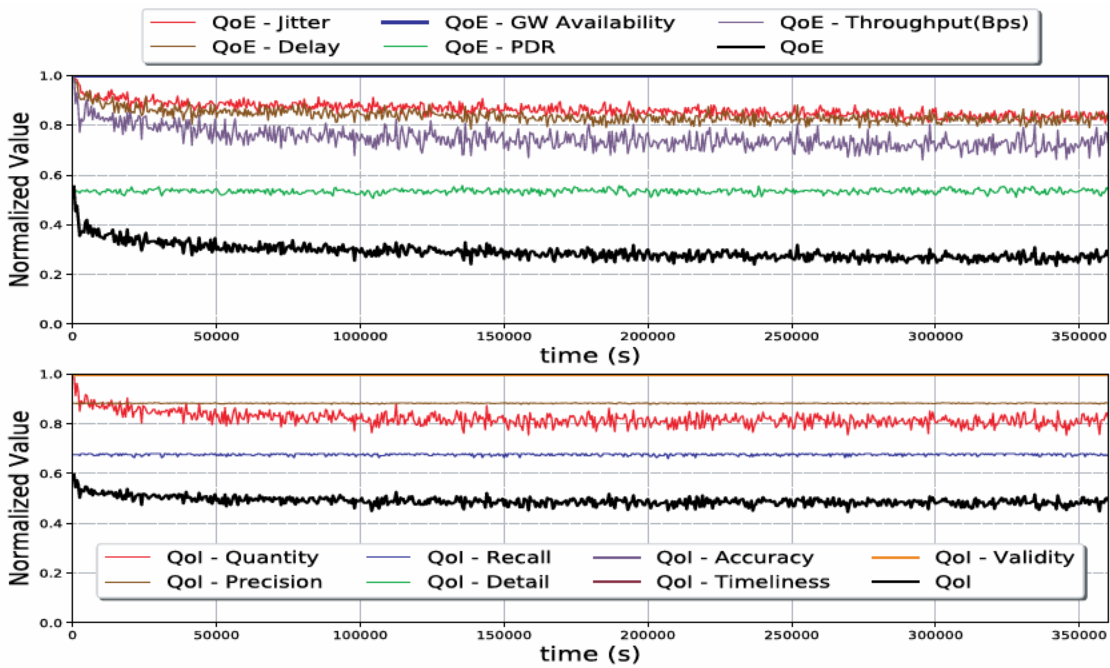


Fig. 15: Métricas de QoI y QoE detalladas sobre un entorno urbano

En relación al entorno suburbano, muestra unas mejores prestaciones si se compara con el entorno urbano, como queda reflejado en la Fig. 16. En lo que respecta a QoD, aumenta el número de paquetes recibidos en el servidor, donde algunos de los mensajes procedente de las nuevas estaciones obtienen un valor de *Completeness* bajo. Esto hace que la componente QoD disminuya ligeramente. Las peores componentes de QoX en el entorno anterior han mejorado significativamente aquí, un 69% para el caso de QoI y un 89% para el caso de QoE, en comparación con la Fig. 16. Como adelantábamos en QoD, se incrementa el número de paquetes recibidos en el servidor (véase Fig. 17) procedentes de toda el área monitorizada, mejorando notablemente los parámetros *Recall* y *Quantity* (damnificados previamente en el entorno urbano). Dicho aumento de paquetes recibidos maximiza también la variable PDR de QoE. De esta forma, mejoran de forma significativa los valores de QoI y QoE.

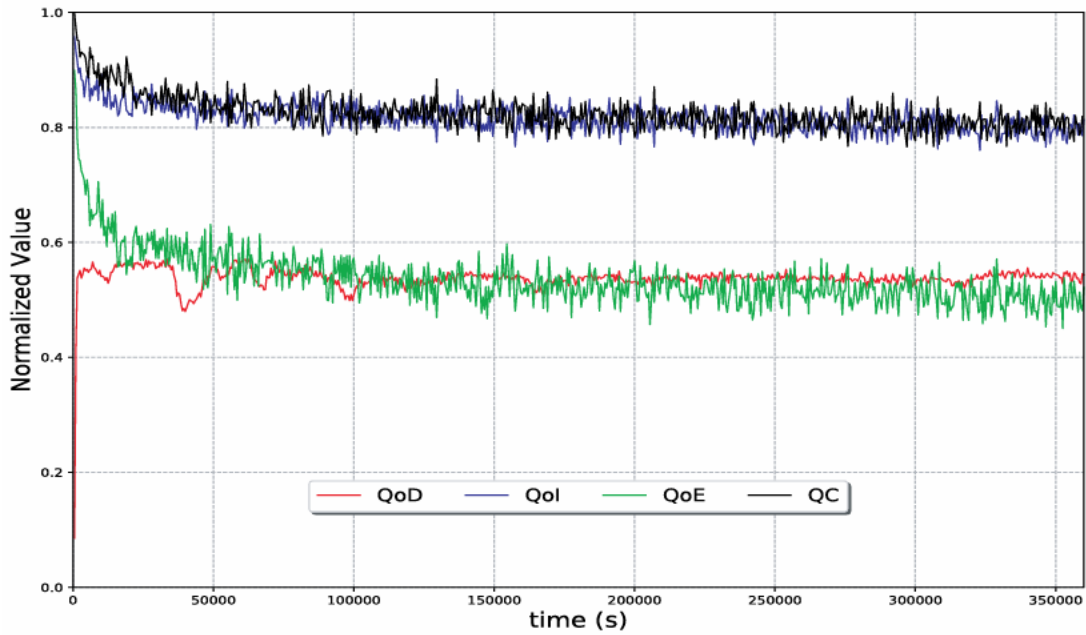


Fig. 16: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno suburbano y nodos fijos en el caso 1.

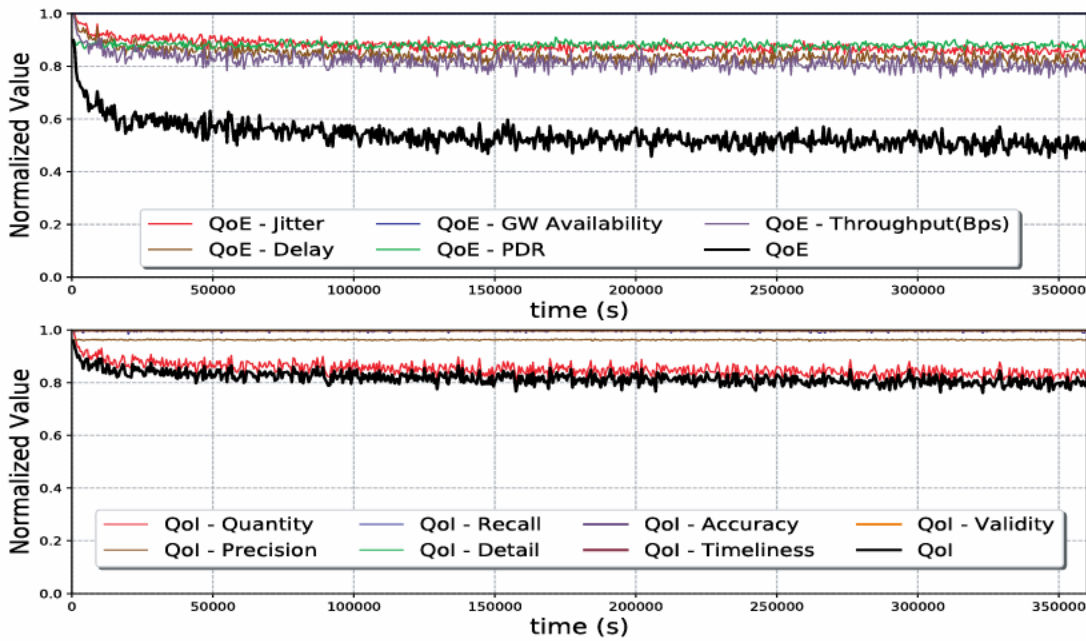


Fig. 17: Métricas de QoI y QoE detalladas sobre un entorno suburbano.

De manera genérica, el entorno que obtiene el mejor rendimiento es el entorno rural (véase Fig. 18). Las métricas de QoD y QC se mantienen similares a las recogidas en el entorno suburbano. El entorno rural registra una pérdida de paquetes mínima respecto a otros entornos, incrementando considerablemente los parámetros *Recall*, *Quantity* y PDR, como se observa en la Fig. 19. Por lo tanto, el hecho de minimizar la pérdida de paquetes repercute directamente sobre la QoI y la QoE. En lo que respecta a QoI, se establece como la componente con mejor rendimiento, mejorando un 75% y un 4% respecto a los entornos urbanos y suburbanos, respectivamente. En relación a la QoE, mejora un 110% respecto al entorno urbano y un 12% respecto al entorno suburbano.

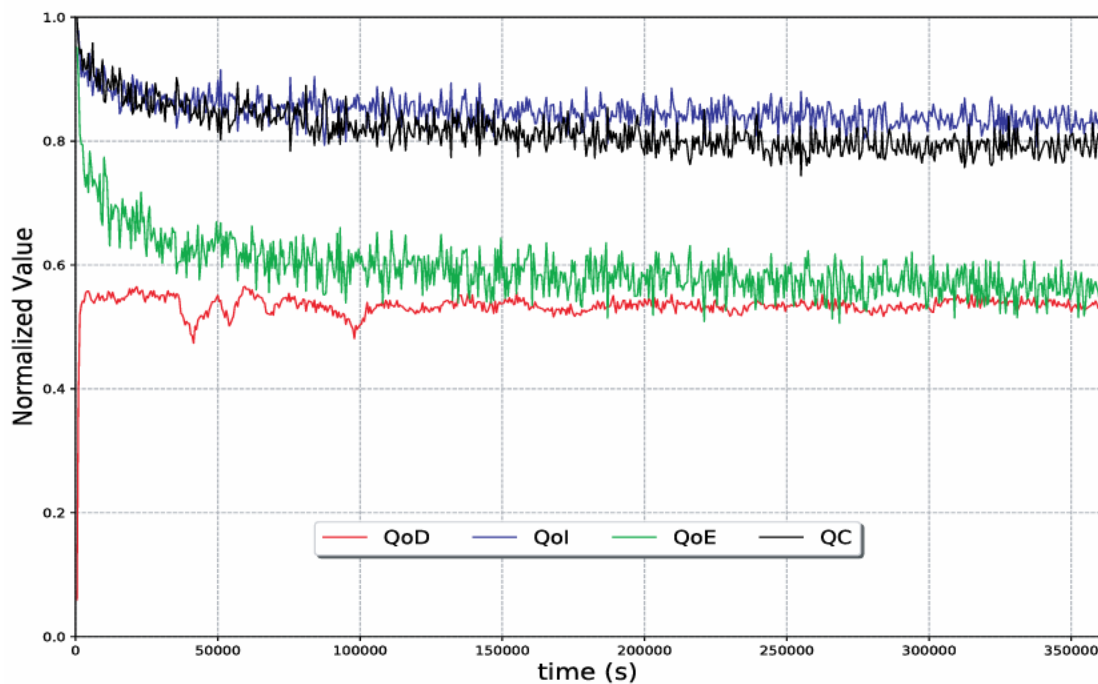


Fig. 18: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno rural y nodos fijos en el caso 1.

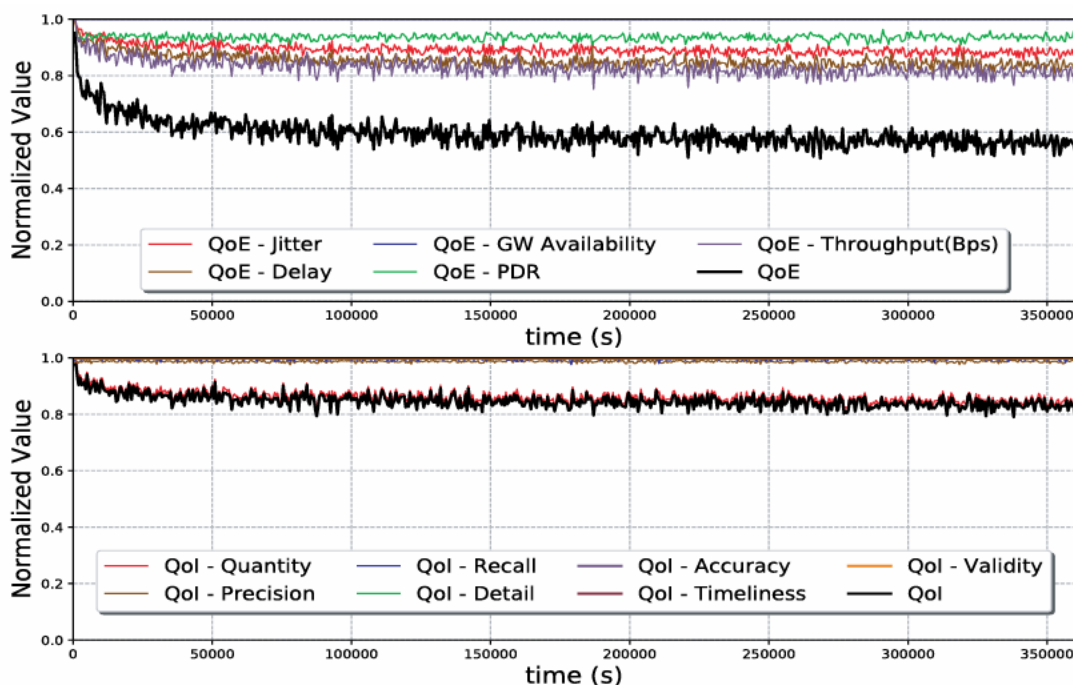


Fig. 19: Métricas de QoI y QoE detalladas sobre un entorno rural.

Caso 2: Variaciones en los parámetros de la red troncal

En este segundo caso se añade un retardo aleatorio [0, 100 ms] y una probabilidad de pérdidas del 10% en lo que denominamos red troncal (*backbone network*). Cuantas más pérdidas y mayores retrasos, se obtendrá un peor rendimiento como se observa en las Fig. 20, Fig. 22, y Fig. 24 en comparación con los resultados del escenario anterior. Al igual que el escenario anterior, se estudian el entorno urbano, el suburbano y el rural.

En entorno urbano (véase Fig. 20), QoD es invariante porque los datos utilizados en la simulación son exactamente los mismos que para el caso 1 donde no había pérdidas ni retardos añadidos en el *backbone*. Del mismo modo, el valor de QoI prácticamente no cambia respecto al caso 1, como muestra la Fig. 21. En lo relativo a QoE sobre un entorno

urbano, la mayoría de los parámetros mantienen su rendimiento respecto al caso ideal, excepto las métricas *delay* y PDR. El retardo aleatorio añadido provoca un aumento del retardo medio en torno a 60 ms. Como el aumento del *delay* es significativo, se observa a simple vista una leve pérdida de rendimiento. También se observa una disminución de rendimiento del 10% en la métrica PDR (coincidiendo con la probabilidad de pérdidas añadidas a la red). Los dispositivos LoRa siguen respetando el ciclo de trabajo y consumiendo los mismos recursos energéticos, lleguen o no los mensajes al destinatario. Es por ello por lo que el valor de QC se mantiene constante comparando este caso (Fig. 20) con el caso anterior (Fig. 14).

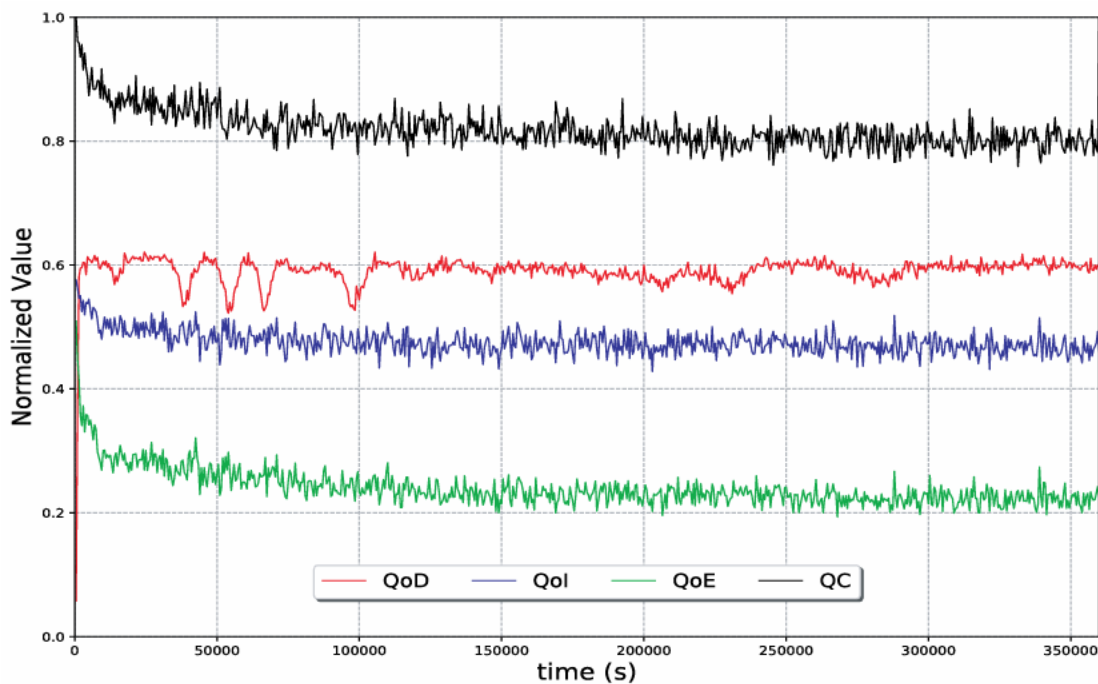


Fig. 20: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno urbano y nodos fijos en el caso 2.

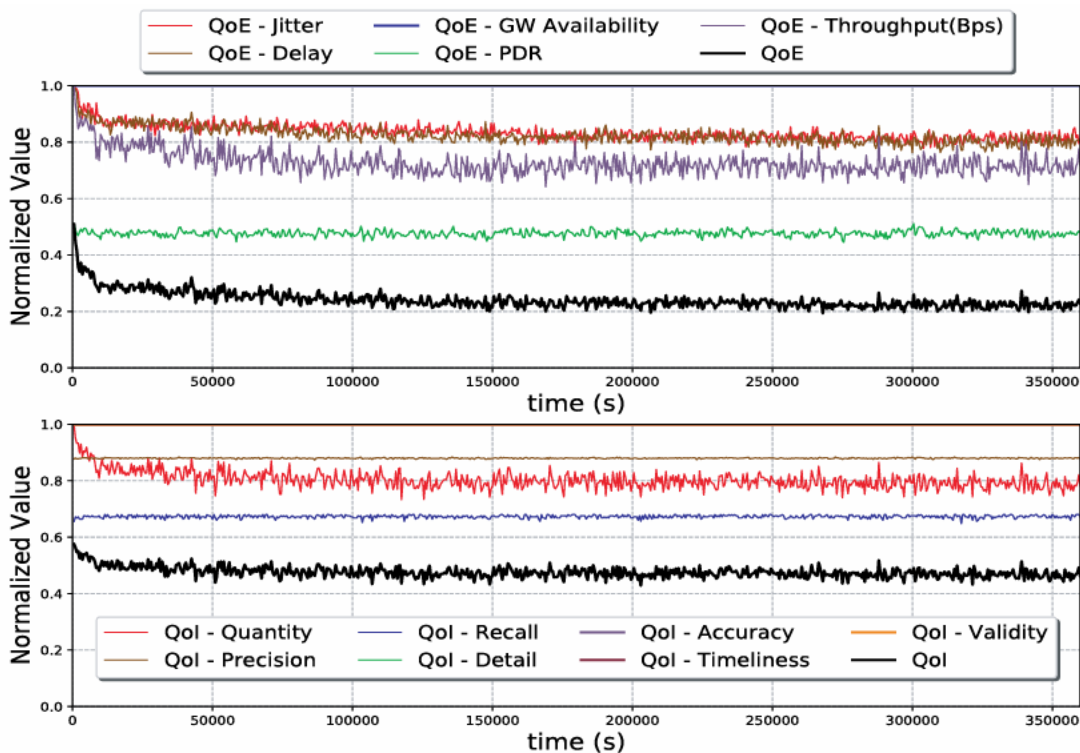


Fig. 21: Métricas de QoI y QoE detalladas sobre un entorno urbano.

Para el entorno suburbano el número de tramas recibidas aumenta significativamente (véase Fig. 22), como también ocurría en el caso anterior. Las componentes QoD y QC no cambian respecto al escenario ideal ya que no se ven afectadas por las condiciones de la red. Por el contrario, QoI y QoE sí presentan variaciones en sus resultados (véase Fig. 23). En lo que respecta a QoI, mejora respecto al entorno urbano principalmente bajo las métricas *Recall* y *Precision*, pero obtiene un rendimiento del 10% menor respecto al primer caso planteado. Por último, en QoE mejoran la mayoría de sus parámetros, pero destacamos el impacto producido sobre el PDR, donde se duplica el rendimiento respecto al entorno urbano. Respecto al escenario ideal, se observa una disminución de rendimiento algo

superior al 10%, que coincide de forma aproximada con la probabilidad de pérdidas añadida a la red troncal.

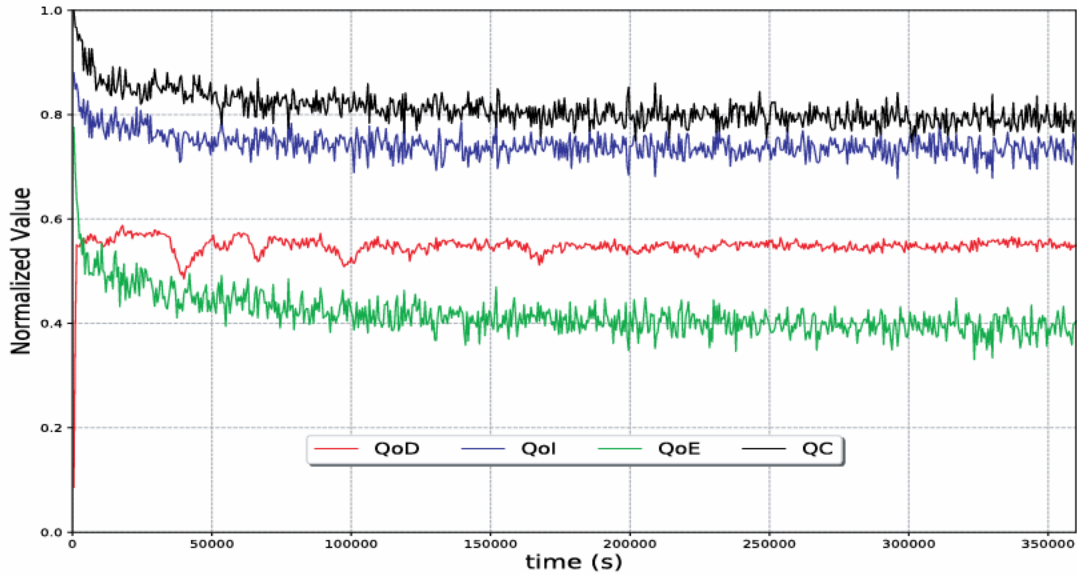


Fig. 22: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno suburbano y nodos fijos en el caso 2.

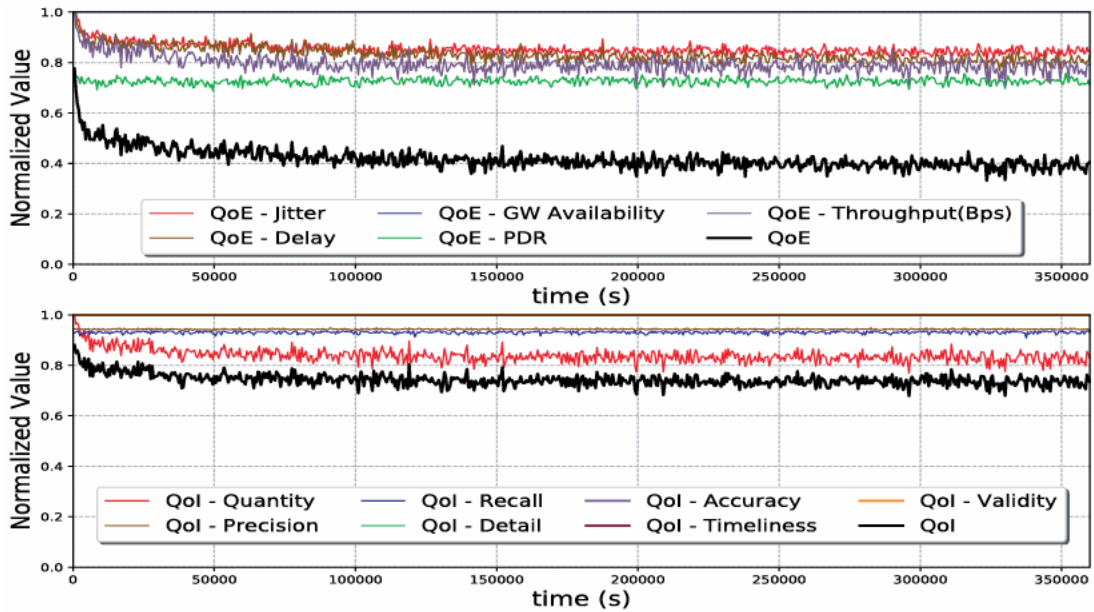


Fig. 23: Métricas de QoI y QoE detalladas sobre un entorno suburbano.

En el entorno rural, se observa un mejor comportamiento respecto al resto de entornos (véase Fig. 24). Al igual que los casos anteriores, los valores de QoD y QC son prácticamente constantes debido a su independencia de la red, mientras que QoI y QoE sí varían respecto a entornos y casos (véase Fig. 25). Por su parte, tanto la QoI como la QoE obtienen los mejores resultados comparados con el resto de entornos para este escenario, pero desciende su rendimiento respecto al caso ideal alrededor de un 7% y 15% respectivamente.

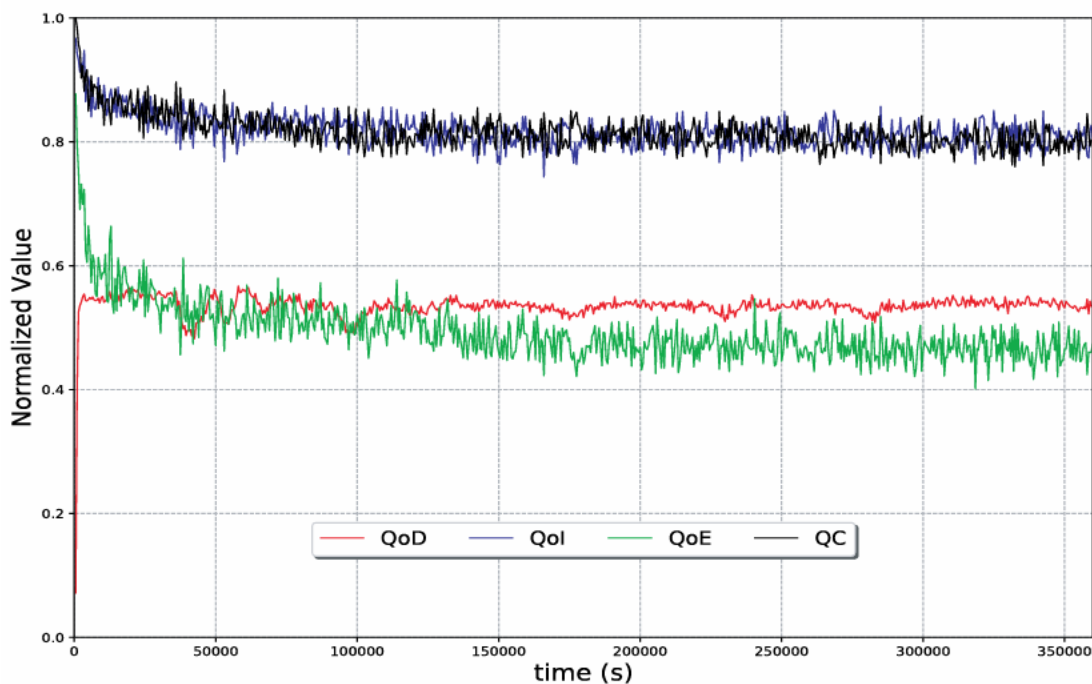


Fig. 24: Evaluación de prestaciones del sistema de monitorización de calidad del aire según las componentes de calidad QoD, QoI, QoE y QC propuestas sobre un entorno rural y nodos fijos en el caso 2.

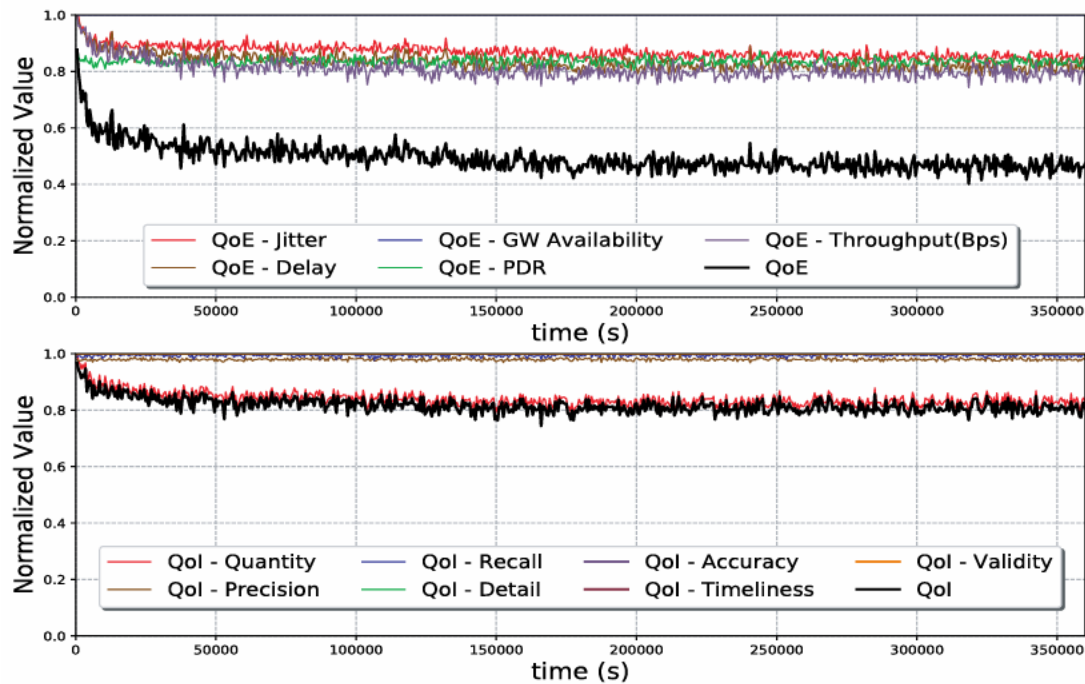


Fig. 25: Métricas de QoI y QoE detalladas sobre un entorno rural.

4.4. Dispositivos Móviles

Los dispositivos LoRa destacan por tener un tamaño reducido y gran capacidad de comunicación pudiendo ser totalmente compatibles con vehículos aéreos no tripulados como los UAVs. De este modo, los UAVs se convierten en vehículos que permiten desplegar servicios basados en IoT. Aunque la madurez de UAVs y LoRa en la IoT es cada vez mayor, hay algunos retos aún por resolver. Previamente, se han descrito los principales modelos de movilidad, tanto específicos como adaptados a FANET. Sin embargo, sigue estando pendiente aún la elección del modelo de movilidad mas eficiente, así como un modelo de evaluación de prestaciones que se ajuste al escenario IoT.

4.4.1. Despliegues

El objetivo es evaluar el rendimiento del sistema implementado con UAVs y tecnología LoRa mediante el modelo de evaluación de prestaciones definido en el capítulo 3 basado en QoD, QoI, QoE y QC. Cada UAV mediante sensores mide la calidad del aire e incorpora dispositivos LoRa para enviar los datos capturados al *gateway*. En este escenario, se prueban hasta ocho modelos de movilidad diferentes como RW, RWP, RD, GM, RPGM, *Pathway*, SRCM, y ST. Para cada modelo de movilidad se estudiará el rendimiento variando tanto el número de UAVs en la red (5, 10, 15 y 20) como la velocidad de vuelo de los mismos (10, 25 y 50 km/h) en un área descrita en Fig. 26. De esta forma, se pretende encontrar el modelo de movilidad que se adapta mejor al servicio bajo estudio.

Se diferencian principalmente dos tipos de configuración. Por un lado, se deben ajustar los parámetros de los modelos de movilidad para variar el comportamiento de los UAVs como se detalla en Fig. 27. Por otro lado, se deben ajustar los parámetros del simulador (véase Tabla 10) definiendo el valor de los parámetros LoRa, la tasa de generación de paquetes, el número de celdas y el entorno en el que se encuentra este escenario. En este caso concreto, los parámetros SF y TP de LoRa son inicializados de forma aleatoria utilizando una distribución aleatoria uniforme, donde SF varía entre 7 y 12 y TP entre 2 y 14 dBm. Por el contrario, BW y CR son fijados a 125 kHz y a 4/5 respectivamente. Los paquetes en un dispositivo LoRa se generan siguiendo una distribución aleatoria exponencial de media 100s y se utilizará en el área disponible una malla de 5x5 celdas. El

entorno seleccionado es suburbano donde la distancia máxima entre dispositivo LoRa y *gateway* es de 12149m y la media 6,57 km.

Destacar que tal y como se observa en la Fig. 26, se disponen 4 *gateways* en la topología. Los dispositivos LoRa envían su mensaje con una dirección *broadcast*, donde todos los *gateways* pueden recibirlo. Un mensaje se recibe en el *gateway* favorablemente si y solo si la potencia de llegada es superior a la sensibilidad del dispositivo. Dicha potencia de llegada se estima en un entorno de simulación mediante el modelo Okumura-Hata. En caso de recibir múltiples copias de un mensaje en el servidor se descartan aquellas recibidas a posteriori.

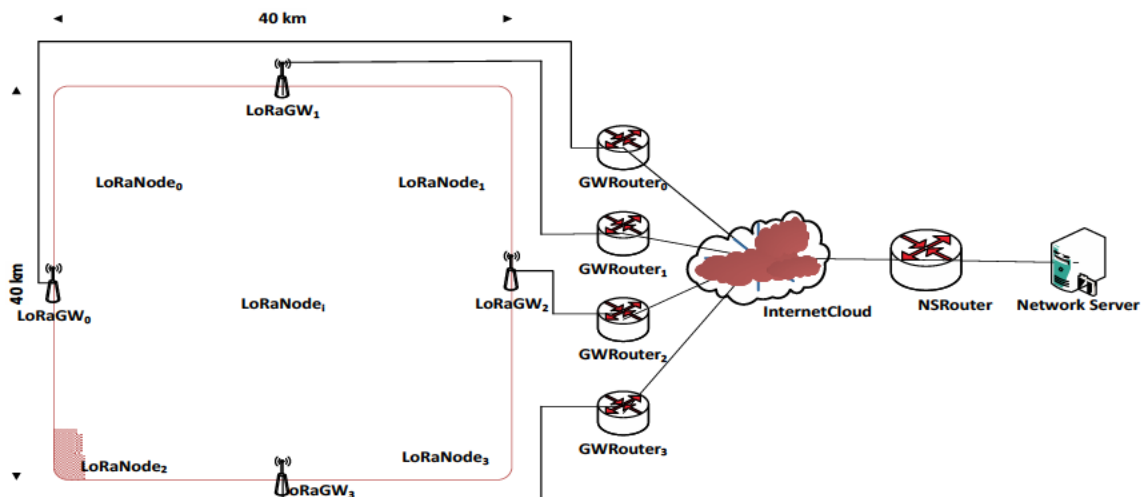


Fig. 26: Despliegue IoT que implementa tecnología de red LoRa en UAVs.

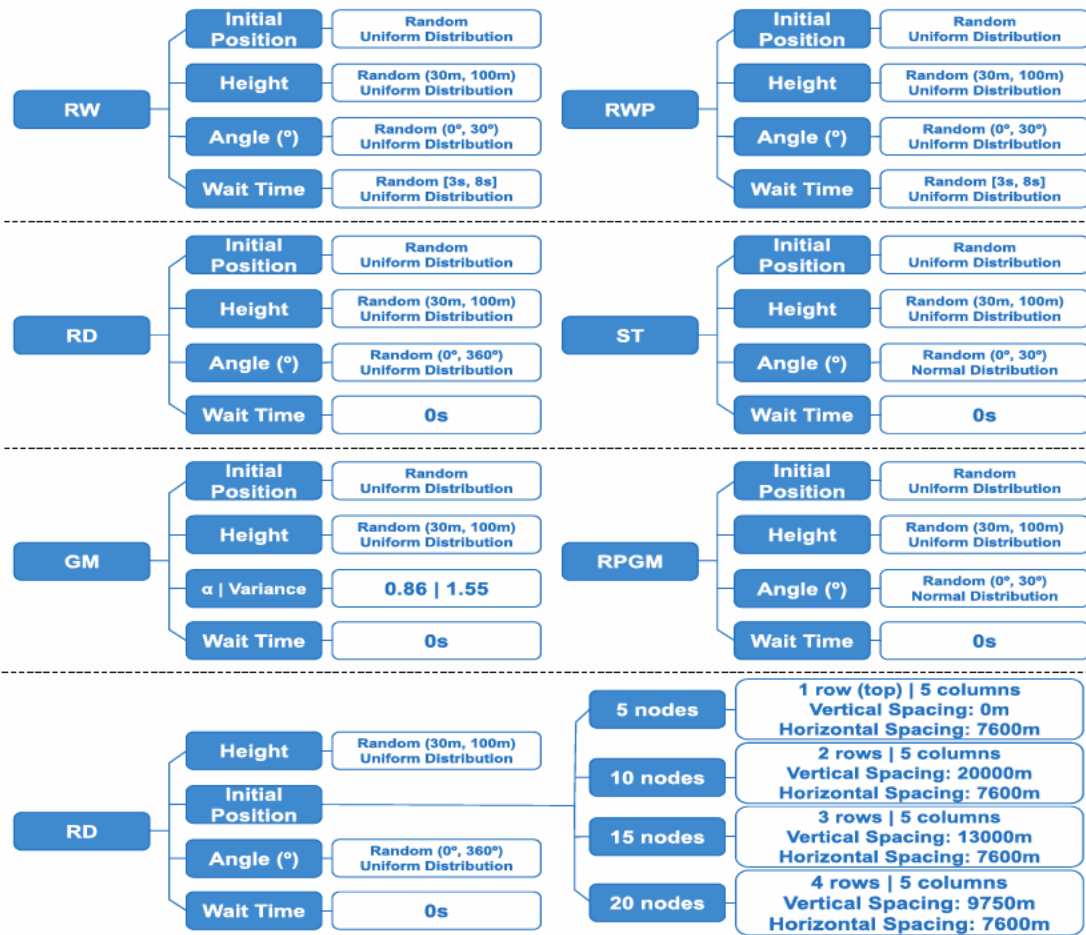


Fig. 27: Parámetros específicos de los modelos de movilidad en la simulación.

Tabla 10: Parámetros de simulación para dispositivos móviles.

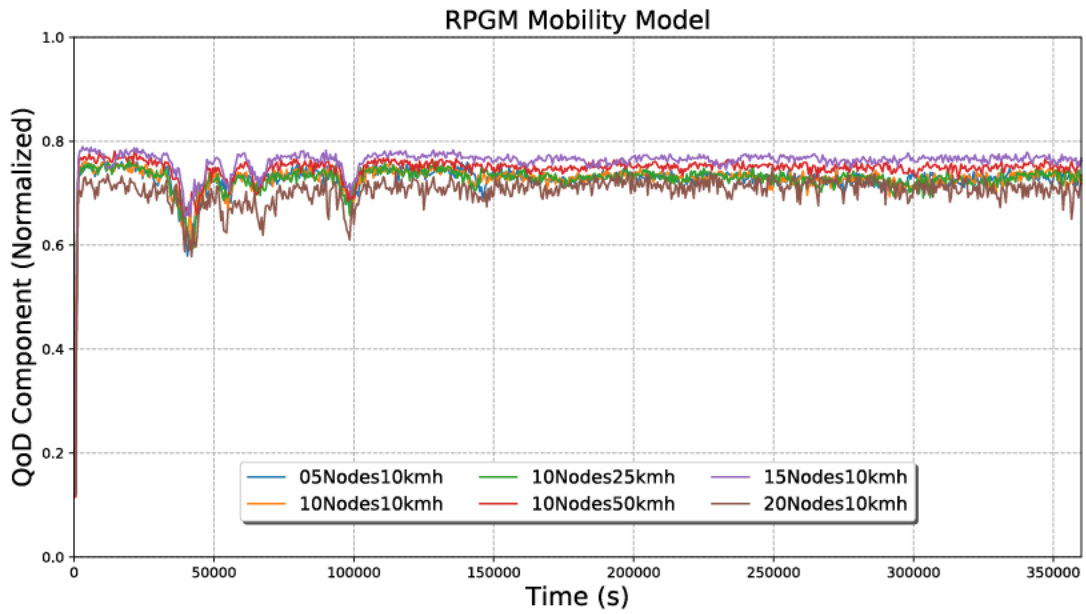
Parámetro	Valor
<i>Spreading Factor (SF)</i>	Distribución aleatoria uniforme en el rango [7, 12]
<i>Transmission Power (TP)</i>	Distribución aleatoria uniforme en el rango [2, 14 dBm]
<i>Bandwidth (BW)</i>	125000 Hz
<i>Coding-Rate (CR)</i>	4/5
<i>Tiempo entre paquetes</i>	Distribución aleatoria exponencial de media 100 s
<i>Número de celdas</i>	5x5
<i>Altura de las estaciones</i>	Depende del modelo de movilidad
<i>Altura del gateway</i>	49 m
<i>Entorno</i>	Suburbano
<i>d_{max}, d_{min} suburbano</i>	12148.93 m, 999.71 m

4.4.2. Resultados en escenario con movilidad

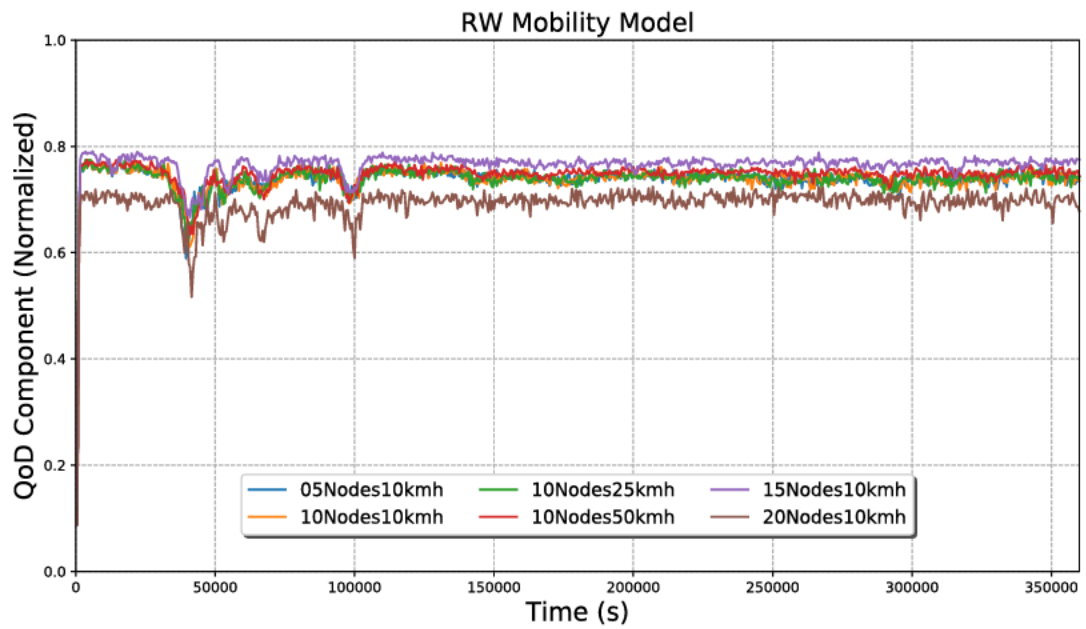
Tras simular los ocho modelos de movilidad variando tanto el número de dispositivos como la velocidad de movimiento, se obtienen resultados acordes a las métricas utilizadas (QoD, QoI, QoE y QC) haciéndonos una idea del rendimiento de los dispositivos LoRa sobre UAVs. Tal y como se observa en las siguientes gráficas, en todos los modelos de movilidad se puede observar una mejora de las métricas con el aumento de dispositivos en la red excepto para RPGM ya que penaliza notablemente que todos los nodos vayan agrupados ocupando una pequeña área de espacio. En este caso concreto, en un T_{eval} sólo se reciben paquetes de una o dos celdas. En el escenario en el que mantenemos constante el número de dispositivos y variamos la velocidad de los mismos, se produce una ligera

mejora de rendimiento a medida que aumenta la velocidad de desplazamiento. Para el caso particular que comentábamos anteriormente de RPGM, una mayor velocidad de los nodos minimizará la pérdida de rendimiento al recibir paquetes de más celdas en un T_{eval} .

A nivel particular, se analiza el proceso de adquisición de datos a partir de los *raw data* tomados por los sensores incorporados en los UAVs. La métrica QoD penaliza por las variables de contaminación que no se toman en una estación. Respecto a la variación del número de nodos, QoD mejora cuanto mayor es el número de UAVs que recorren el área de cobertura. En el escenario con 5 nodos los datos en bruto son más débiles, ya que algunas estaciones como $nodo_0$ no incorporan todos los sensores que permiten analizar la calidad del aire. A medida que aumenta el número de UAVs, este “pobre” UAV ($nodo_0$) pasa desapercibido ya que el resto de estaciones recogen un gran volumen de datos, aumentando así el valor de QoD. El valor de QoD es prácticamente constante conforme varía la velocidad de los UAVs para un número fijo de estaciones, aunque sí presenta cierta varianza entre los diferentes escenarios debido a la generación aleatoria de los paquetes en cada UAV. Por último, el escenario con 20 UAVs es ligeramente mejor en términos de QoD para todos los modelos de movilidad. De entre los modelos probados, el mejor rendimiento lo consiguen RPGM y RW (véase Fig. 28) y peor, el RWP y RD (véase Fig. 29). El resto de modelos se representan en la Fig. 30.

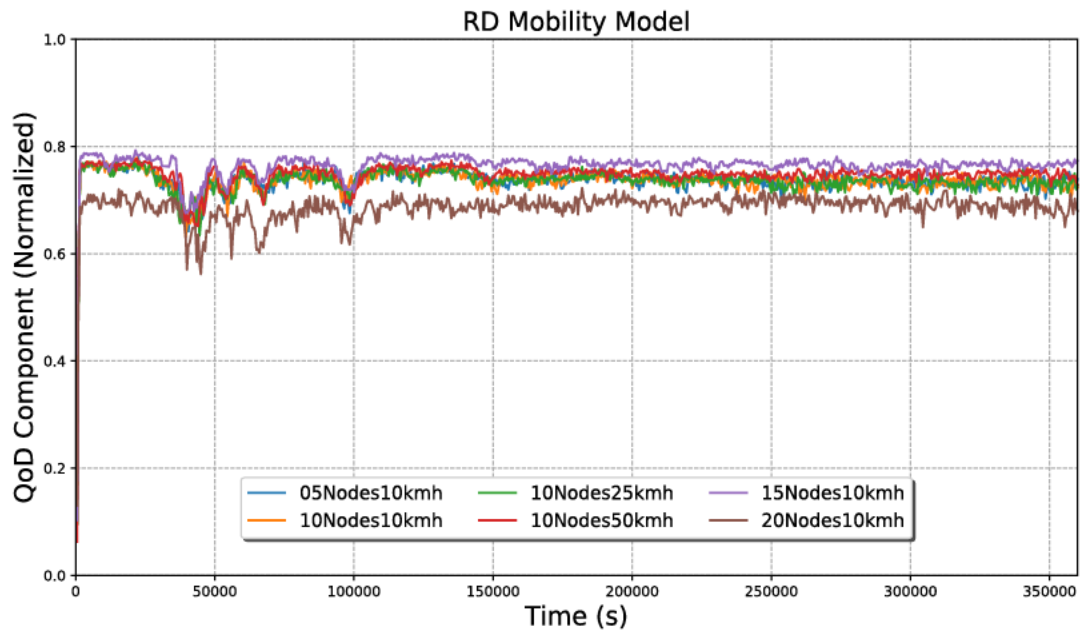


(a)

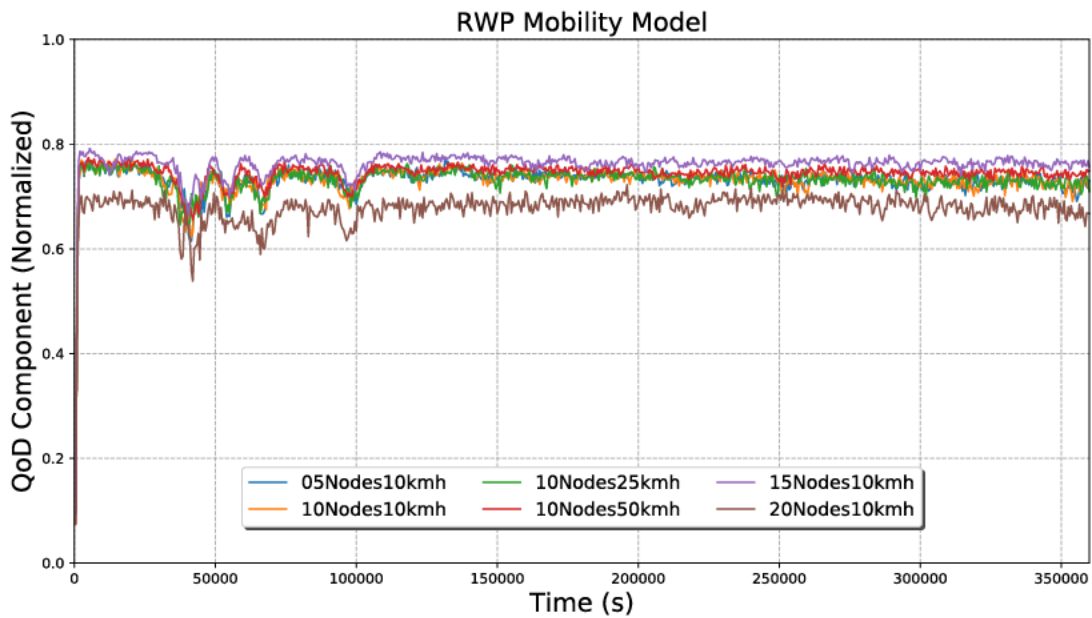


(b)

Fig. 28: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RPGM y (b) RW, donde se obtienen los mejores resultados para esta componente QoD.

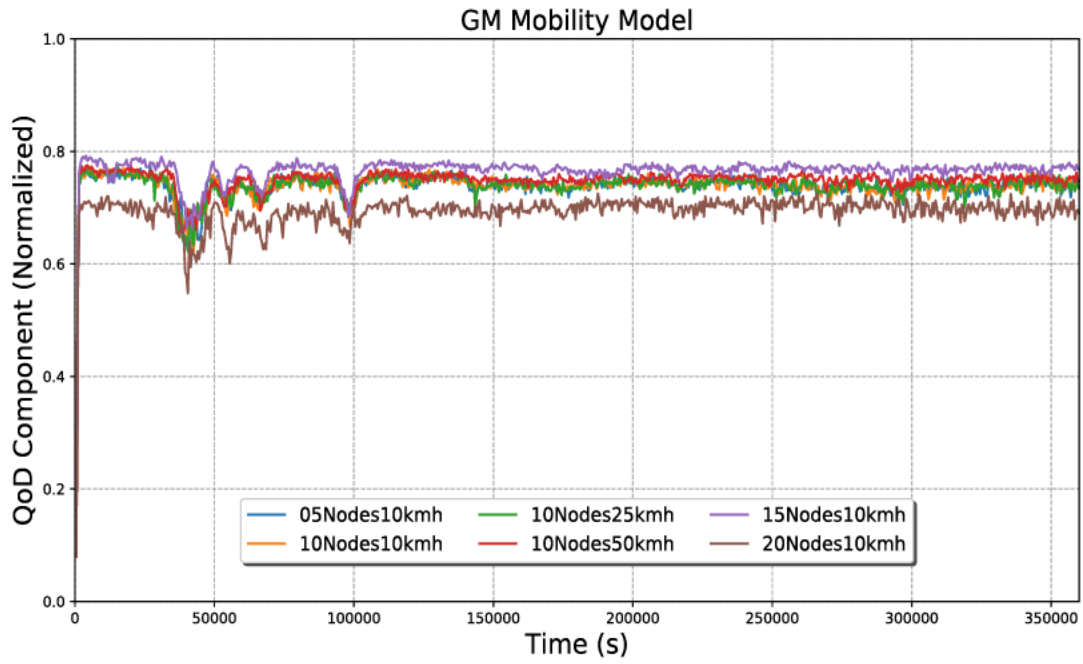


(a)

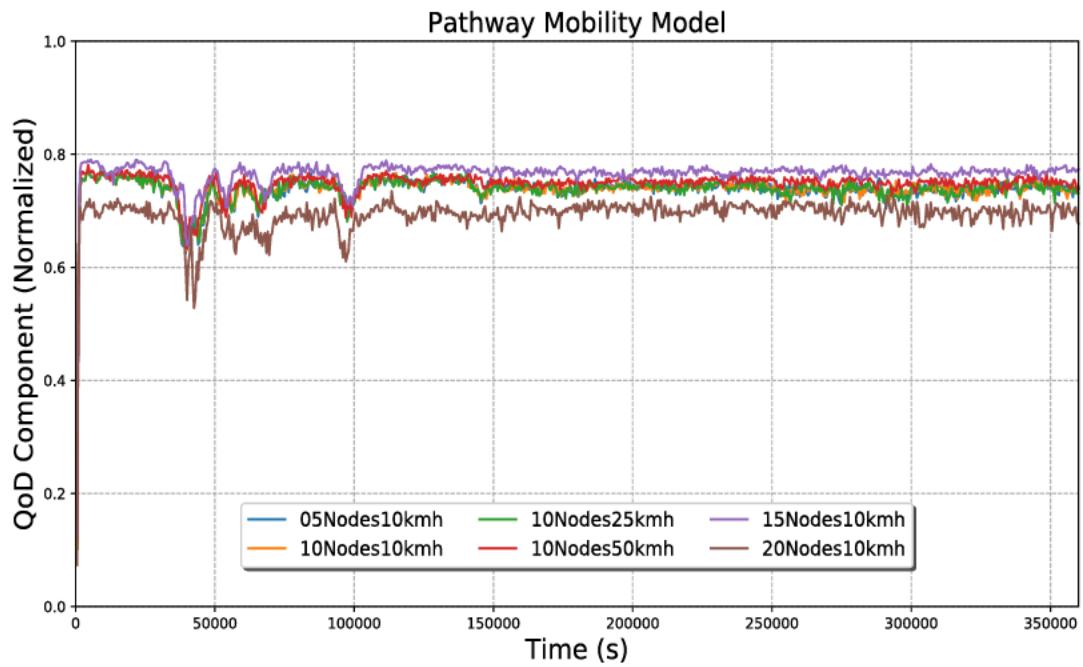


(b)

Fig. 29: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RWP y (b) RD, donde se obtienen los peores resultados para esta componente QoD.



(a)



(b)

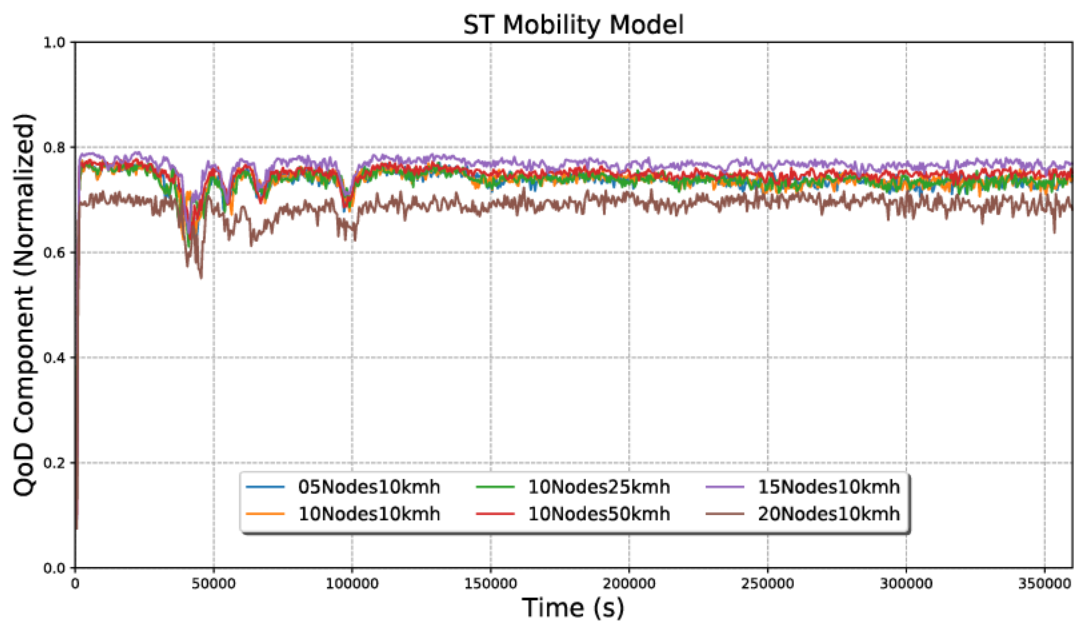
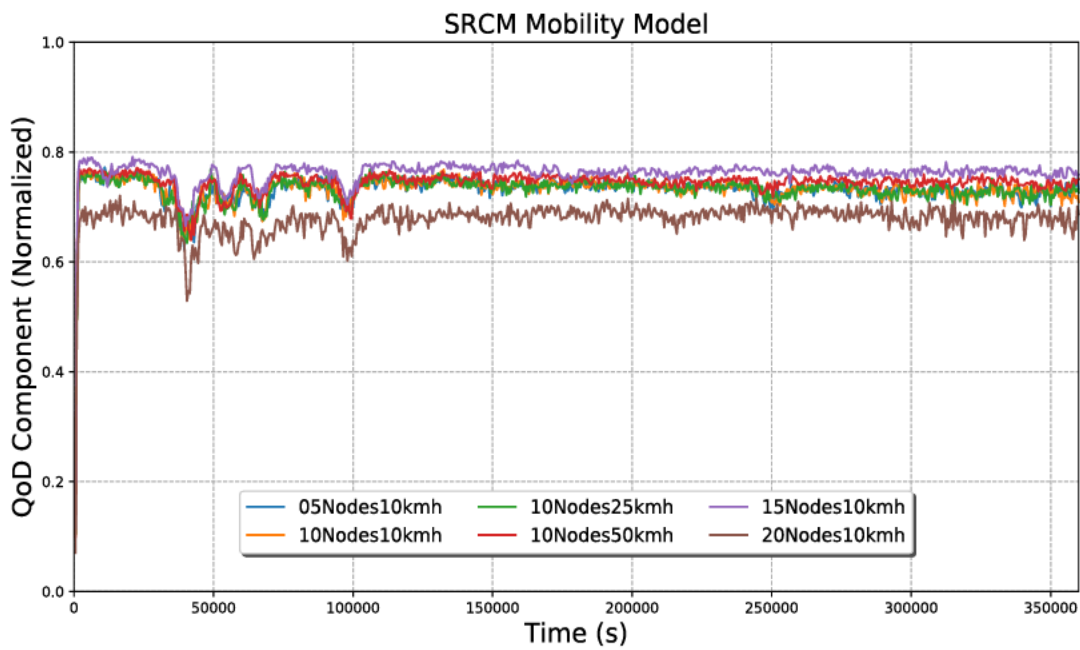
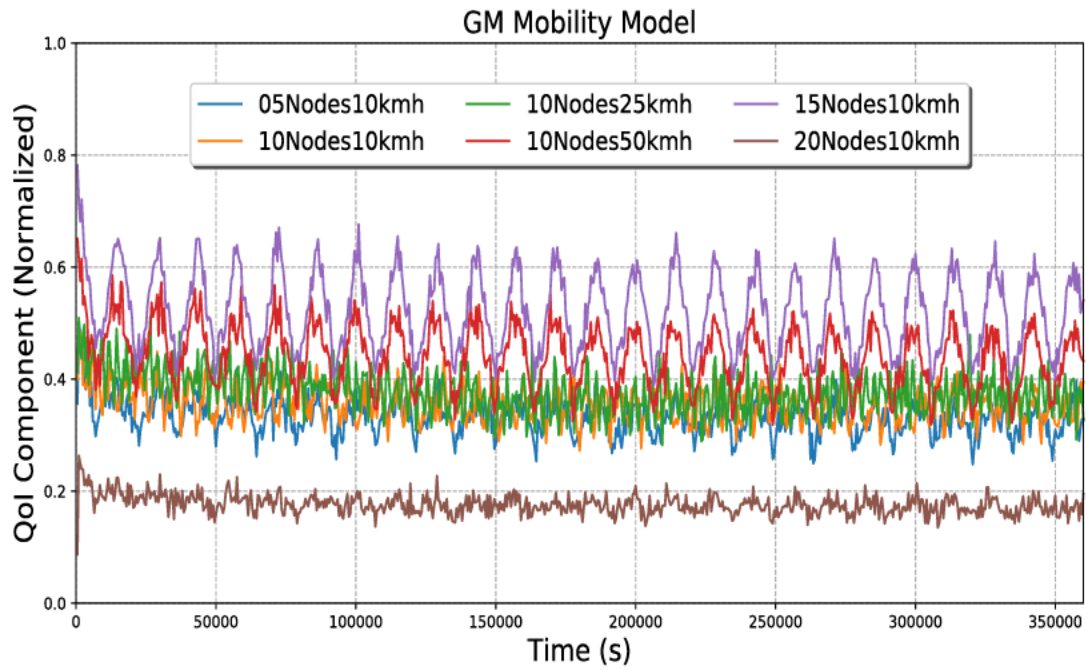
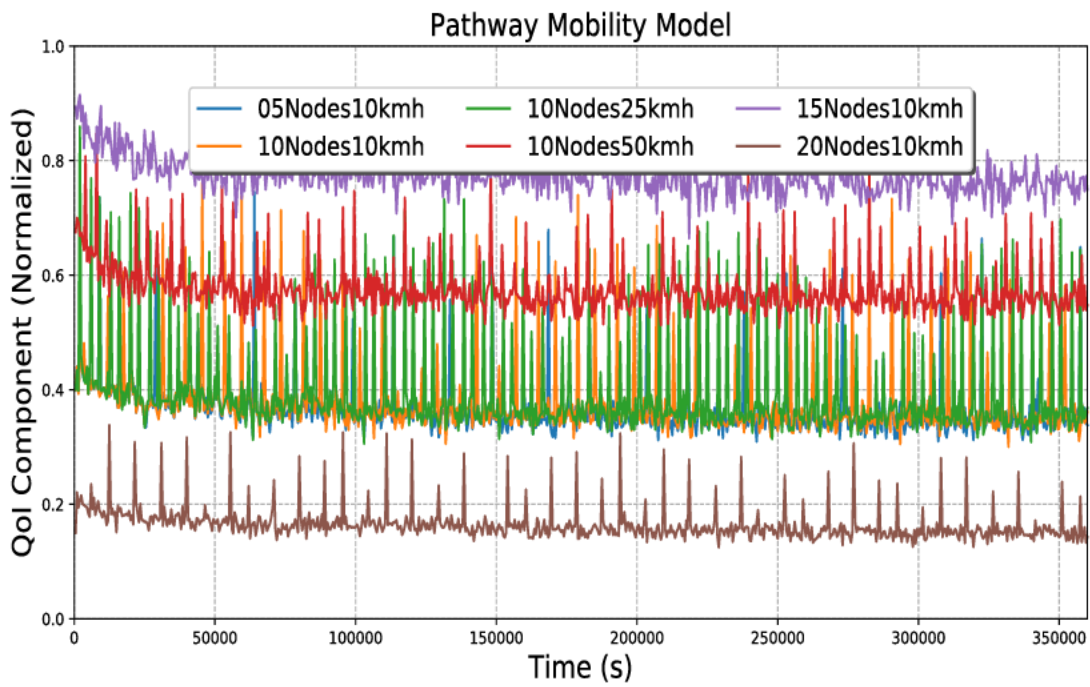


Fig. 30: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoD propuesta sobre un entorno suburbano y nodos móviles para el resto de modelos; (a) GM; (b) Pathway; (c) SRCM; y (d) ST.

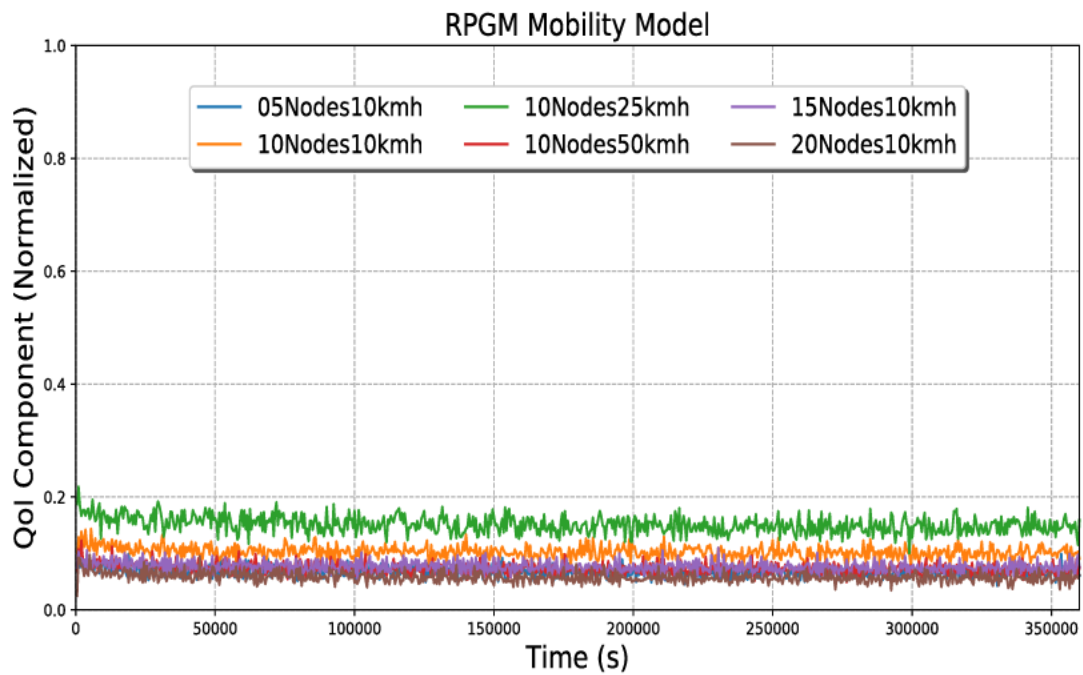
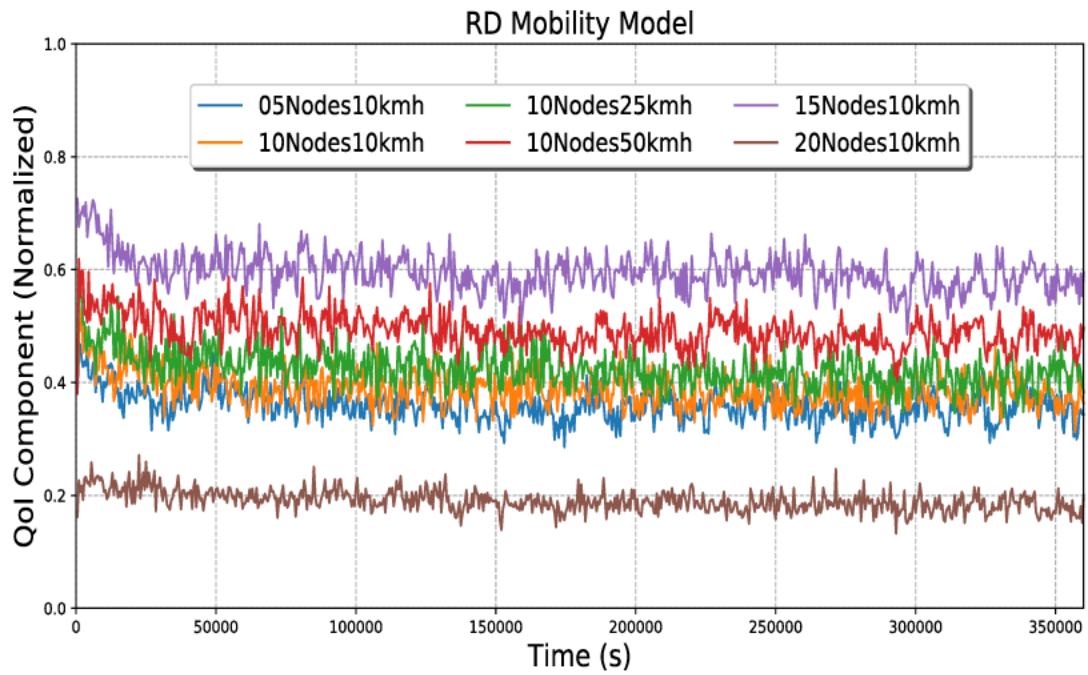
Después del proceso de adquisición de datos, el componente QoI mide la calidad de la información obtenida. La QoI se reduce a siete métricas como se muestra en (6), lo que la hace muy volátil. En otras palabras, la movilidad tiene un mayor impacto en QoI que en QoD. En este caso particular, cuanto más rápido se muevan los dispositivos por el área sensorizada, mejor será el rendimiento ya que el Servidor recibirá paquetes de todas las celdas en las que se divide el área de cobertura. La afirmación anterior supone que el Servidor puede tomar decisiones más precisas cuanto mayor es la velocidad de desplazamiento de los dispositivos, ya que le permite recorrer un mayor número de celdas. De lo contrario, recibiría paquetes de una zona exclusiva y tomaría decisiones en función de una parte del área sensorizada disponible. Por ello, *Recall* es el parámetro con mayor impacto sobre QoI para este caso. El modelo de movilidad que toma el mejor rendimiento para QoI es *Pathway Mobility* debido a la disposición ordenada de los UAVs en el área monitorizada (véase Fig. 31).

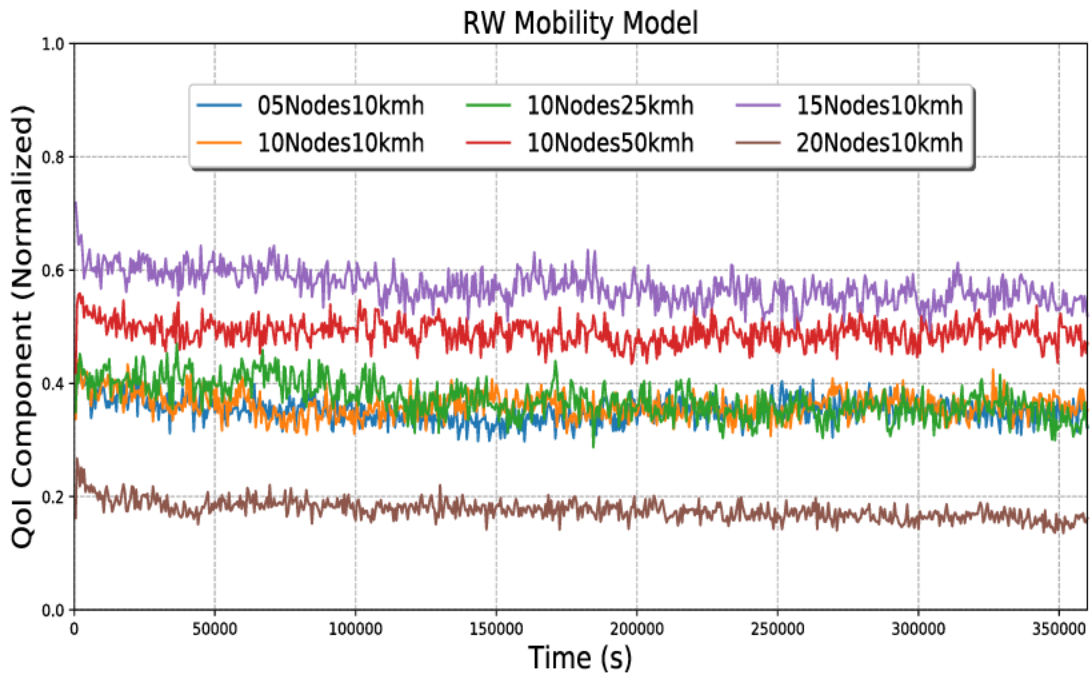


(a)

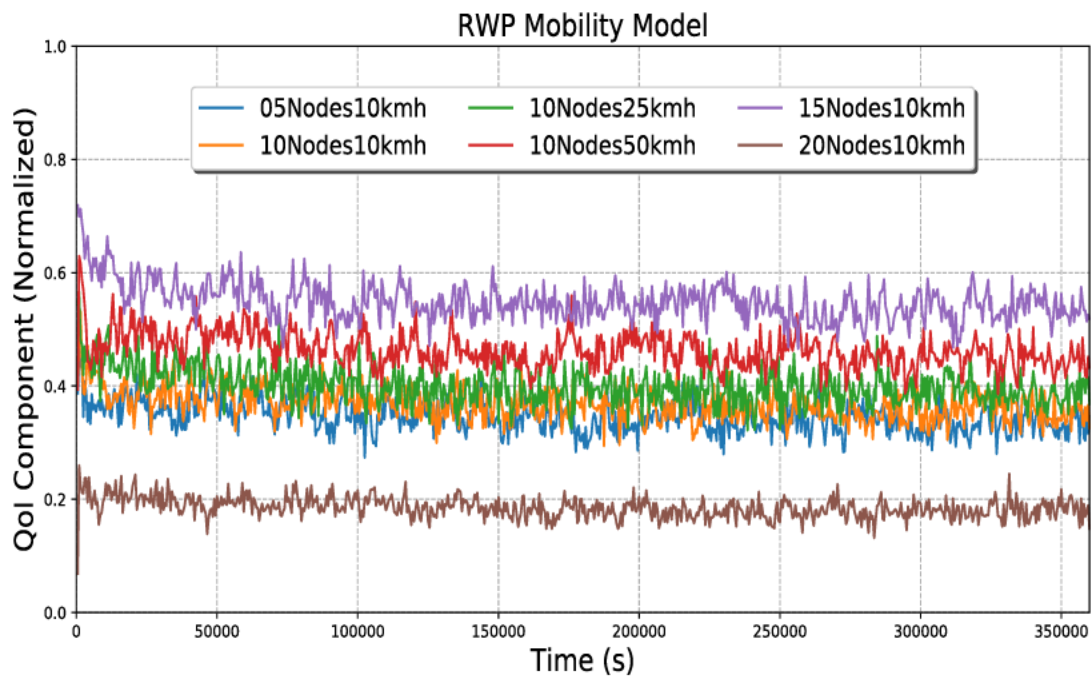


(b)

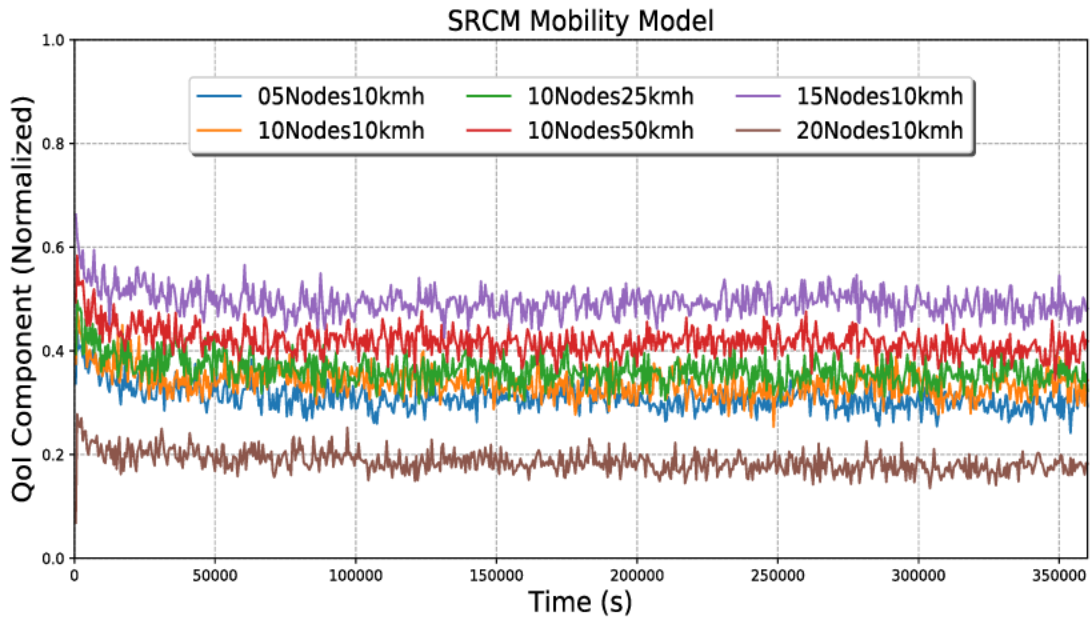




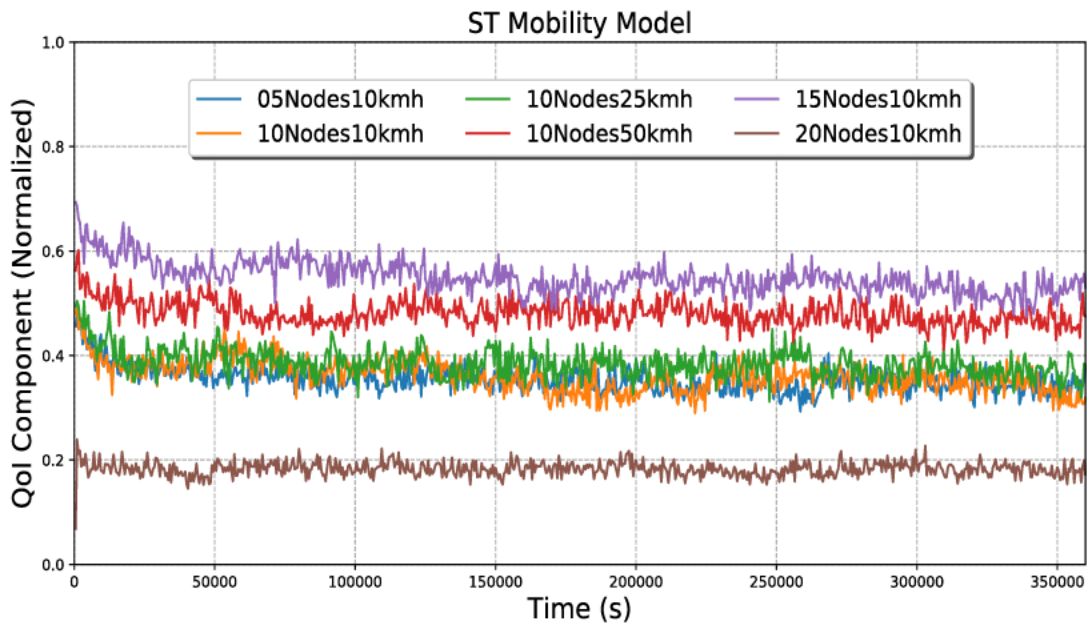
(e)



(f)



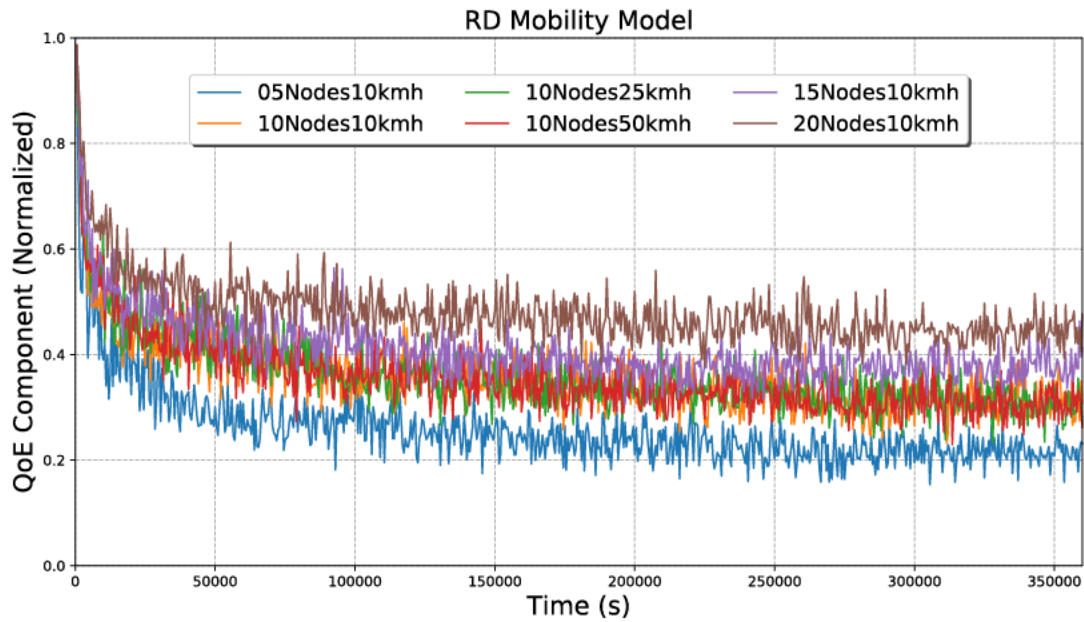
(g)



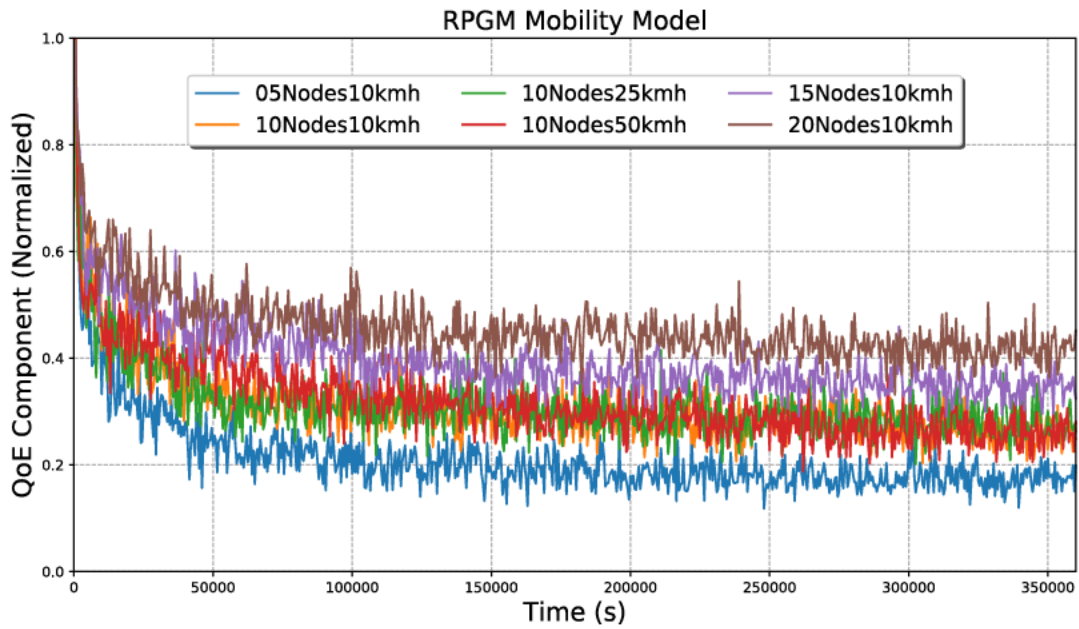
(h)

Fig. 31: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoI propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) ST, (b) SRCM, (c) RWP, (d) RW, (e) RPGM, (f) RD, (g) Pathway y (h) GM.

El uso de cuatro *gateways* garantiza que no hay áreas ocultas o sin cobertura de LoRa en el área disponible. Por tanto el rendimiento en términos de QoE es muy similar para todos los modelos de movilidad. La única métrica que tiene una mayor variación es GW_{Av} (12). Esta métrica estudia el tiempo de ocupación de la interfaz Ethernet en un *gateway* que reenvía tramas al Servidor para determinar si existe un *bottleneck* en este dispositivo en determinados escenarios de congestión, aunque este en particular no es el caso. Todos los modelos de movilidad tienen resultados muy buenos y similares. Destaca ligeramente RD por su comportamiento en el borde del área de cobertura para cambiar de dirección (véase Fig. 32), lo que permite una mayor proporción de paquetes entregados que otros modelos de movilidad. La velocidad de los UAVs no es un factor a tener en cuenta en el rendimiento de la métrica QoE. Finalmente, Fig. 33 muestra los resultados para el resto de modelos de movilidad.

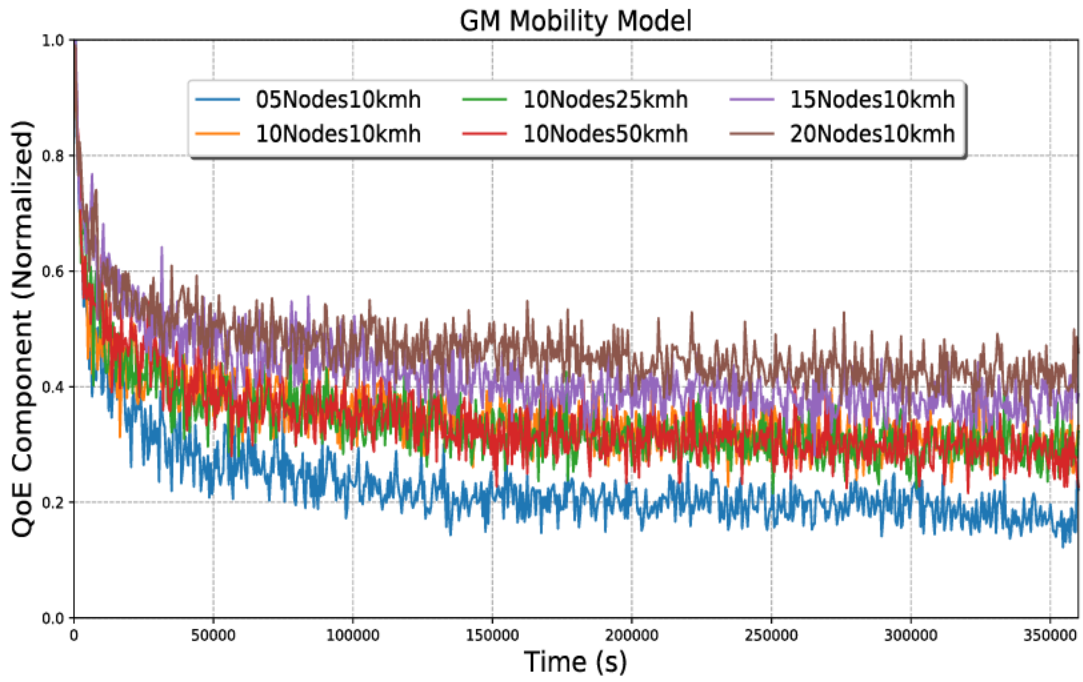


(a)

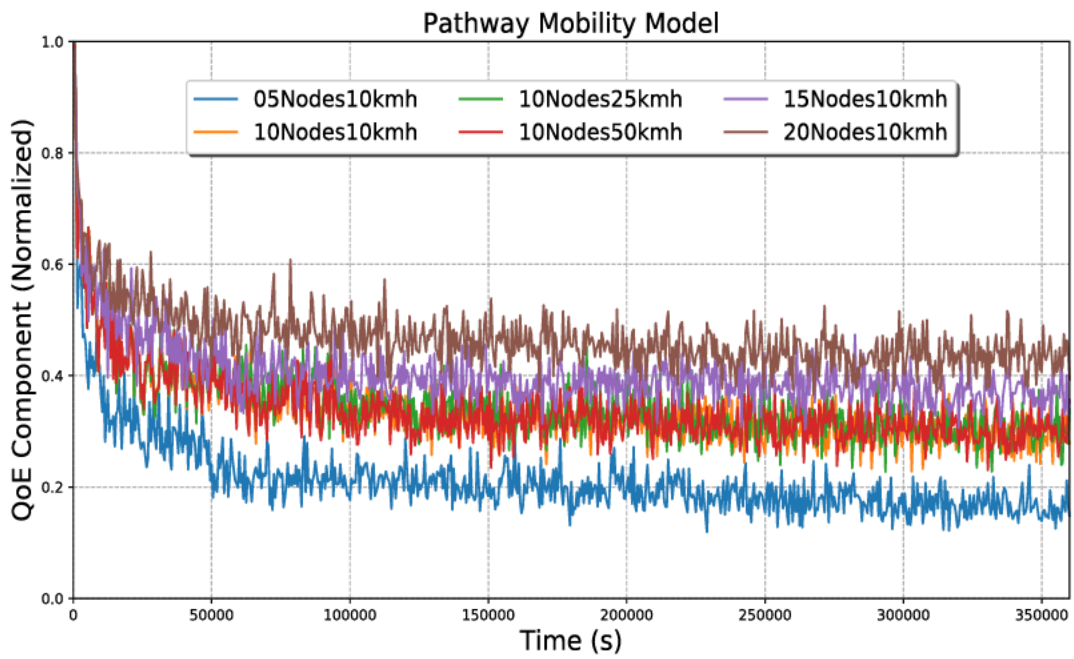


(b)

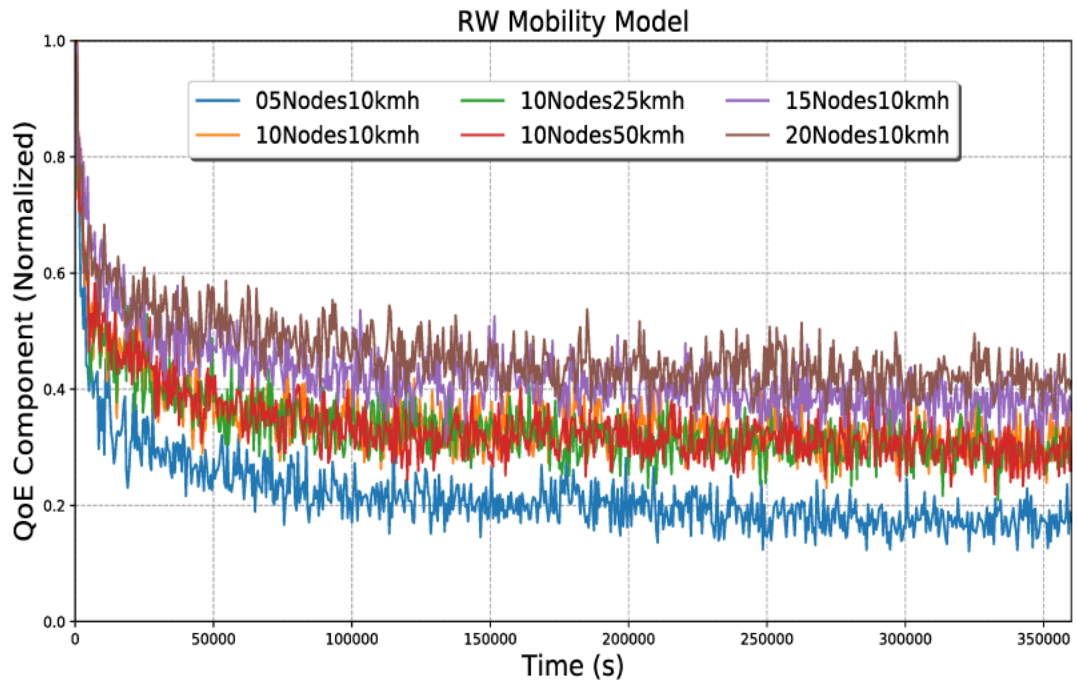
Fig. 32: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoE propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) RD y (b) RPGM, que obtienen el mejor y el peor resultado para esta componente, respectivamente.



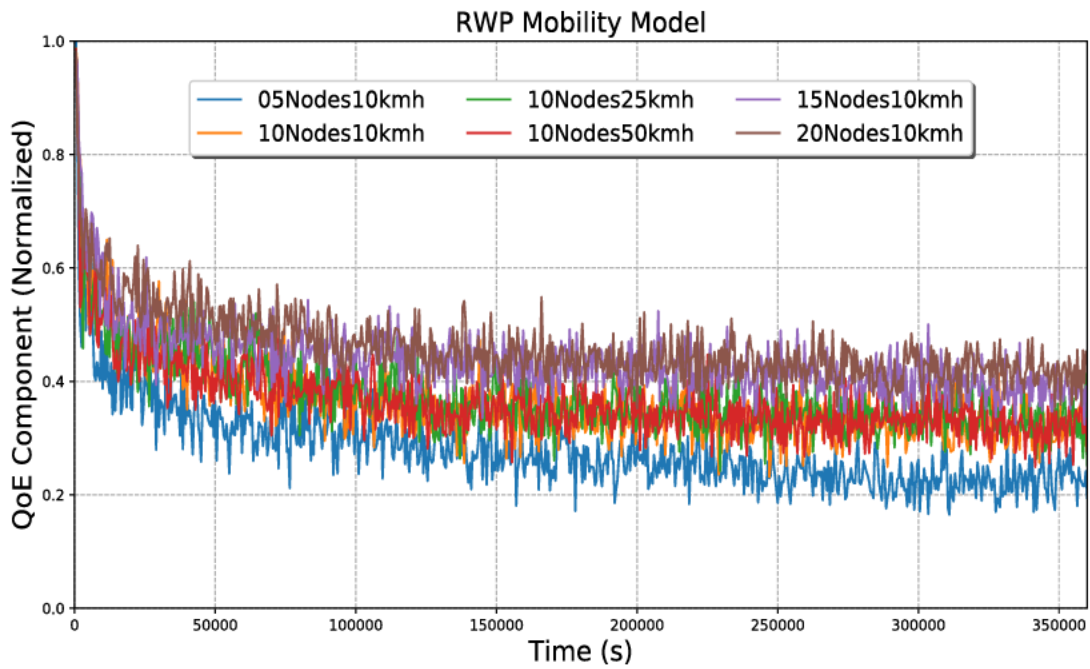
(a)



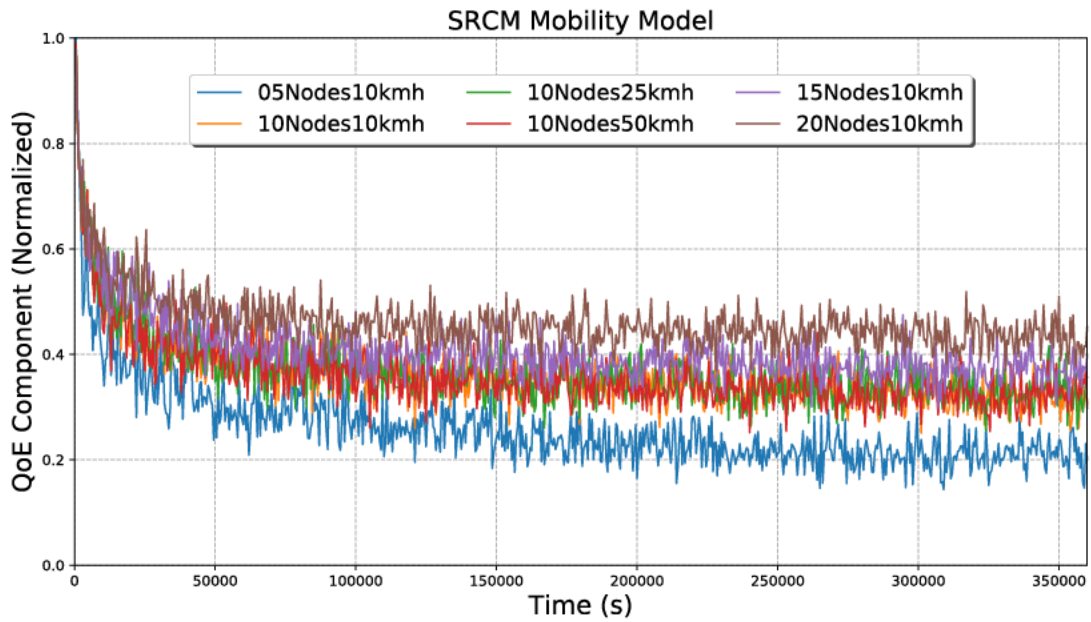
(b)



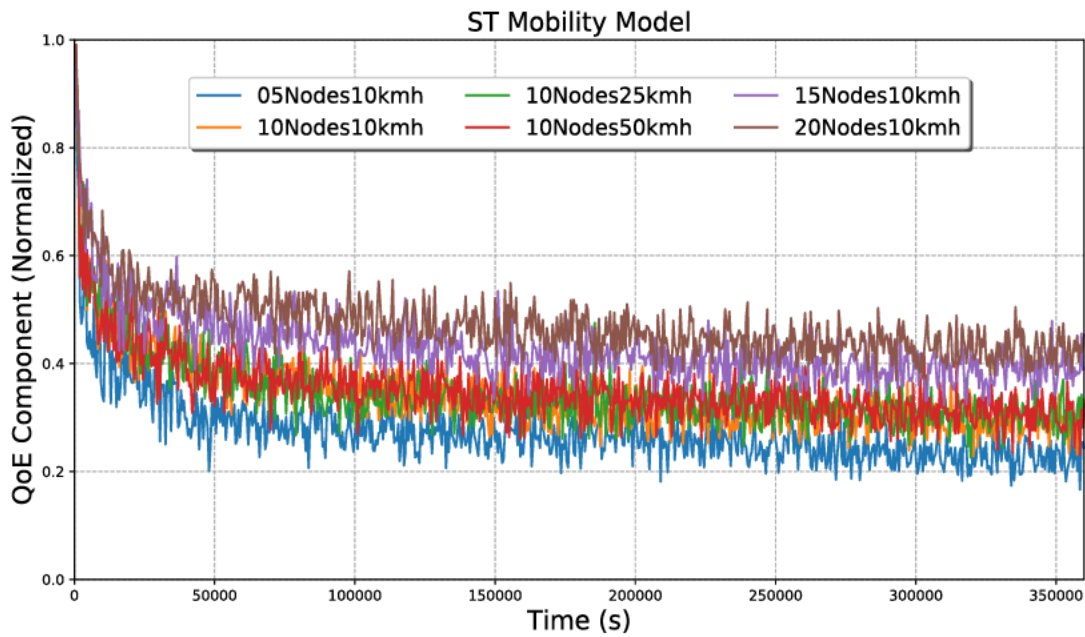
(c)



(d)



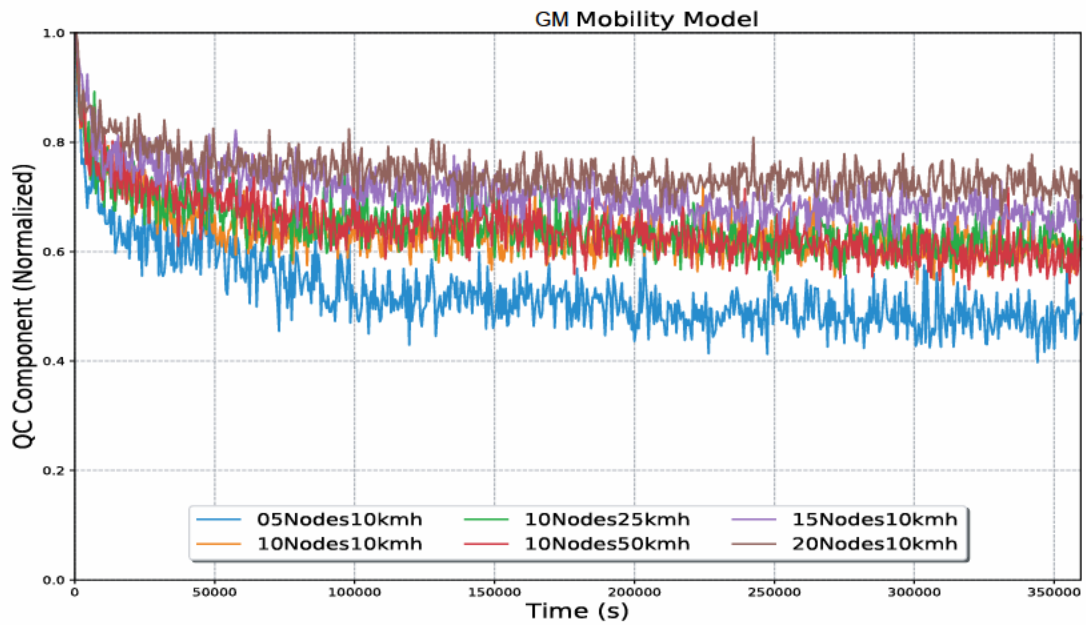
(e)



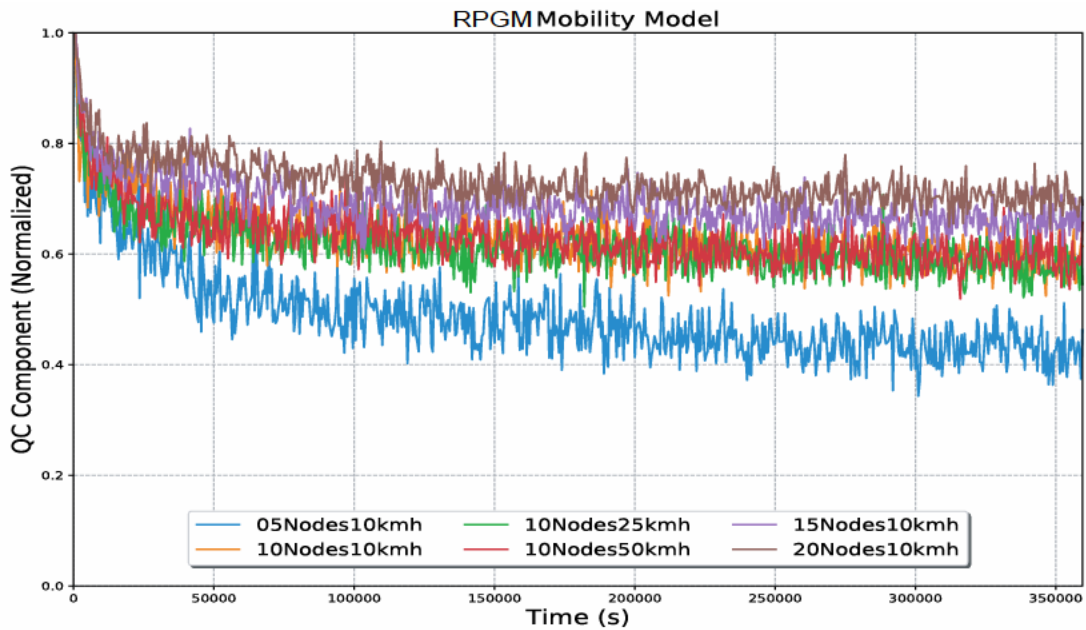
(f)

Fig. 33: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QoE propuesta sobre un entorno suburbano y nodos móviles para el resto de modelos: (a) GM; (b) Pathway; (c) RW; (d) RWP; (e) SRCM; y (f) ST.

Por último, abordaremos el componente de coste QC, donde tenemos en cuenta el consumo energético y la restricción del ciclo de trabajo (1%). El consumo energético depende principalmente del estado del transceptor, siendo mayor para el estado de transmisión. El hecho de emplazarse un menor número de dispositivos en la red y generarse los paquetes de forma aleatoria, hace muy oscilante esta métrica pudiendo variar mucho de un T_{eval} a otro. El mejor resultado se obtiene para el modelo GM y el peor para RPGM (véase Fig. 34), aunque las diferencias son mínimas.



(a)



(b)

Fig. 34: Evaluación de prestaciones del sistema de monitorización de calidad del aire según la componente de calidad QC propuesta sobre un entorno suburbano y nodos móviles con los modelos de movilidad (a) GM y (b) RPGM, que obtienen el mejor y el peor resultado para esta componente, respectivamente.

4.5. Conclusiones

Existe una tendencia actual en la que se pretende diseñar sistemas multifacéticos que impulsen los futuros servicios inteligentes. Por un lado, las tecnologías emergentes dotan de gran flexibilidad a las aplicaciones o servicios que se desarrollen (como UAVs y LoRa). Hoy en día no existe un estándar para evaluar el rendimiento de estas tecnologías (o combinación de ellas). En esta sección se pretende analizar el rendimiento de la tecnología LoRa en comunicaciones tanto de forma aislada como combinada con otras tecnologías, como UAVs, sobre diferentes escenarios y entornos (rural, suburbano y urbano). Para ello se utiliza el modelo descrito en el capítulo 3, que aborda diferentes dimensiones del funcionamiento del servicio. En el caso de experimentarse movimiento en los dispositivos, se contrastan los resultados utilizando ocho modelos de movilidad.

A nivel de componentes de calidad, observamos que QoS y QoC son prácticamente invariantes ya que sus parámetros no dependen de factores de la comunicación. Por el contrario, QoI y QoE sí experimentan variaciones al variar el número de dispositivos, las condiciones de la red, o la velocidad de movimiento de los dispositivos. Más en detalle, QoI logra un mejor rendimiento para aquellos escenarios donde los dispositivos se mueven de forma ordenada, mientras que QoE depende mayormente de las condiciones de la red y favorecerá aquellos escenarios en los que se minimice el retardo, el *jitter* o la probabilidad de pérdidas.

4.6. Contribuciones Relacionadas a este Capítulo

J.M. Martínez-Caro, M.-D. Cano, “Improving the FLoRa Simulation Framework for the Performance Evaluation of IoT Scenarios”. In *Proceedings of SENSORCOMM'19* (2019), Niza (Francia), pp. 1-7. ISSN: 2308-4405. ISBN: 978-1-61208-744-3.

J.M. Martínez-Caro, M.-D. Cano, “Simulación y análisis de rendimiento en dispositivos LoRa sobre drones”. In *Proceedings of Jornadas de Ingeniería Telemática JITEL'19* (2019), Zaragoza (España), pp. 1–8. Comunicación oral. ISBN: 978-84-09-21112-8.

J.M. Martínez-Caro, M.-D. Cano, “IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study”, *Wireless Communications and Mobile Computing* (2019), pp. 1-12. <https://doi.org/10.1155/2019/4307925>
Factor de Impacto JCR: 1,819 (Q3), Rank: 153/266 in Engineering, Electrical & Electronic.

J.M. Martínez-Caro, M.-D. Cano, “A novel holistic approach for performance evaluation in Internet of Things”, *Int. J. Commun. Syst.* (2020), pp. 1–24, <https://doi.org/10.1002/dac.4454>
Factor de Impacto: 1,319 (Q3), Rank: 198/266 in Engineering, Electrical & Electronic.

Aplicación de Técnicas de Machine Learning en la Evaluación de Prestaciones

5.1. Introducción

Desde hace años, los humanos soñamos con el uso de dispositivos dotados de capacidades cognitivas capaces de facilitar nuestro día a día. Este sueño se hará realidad con el uso de técnicas englobadas bajo el término Aprendizaje Máquina (*Machine Learning*, ML). Este concepto se basa en el desarrollo de algoritmos que aplican una base matemática y estadística para dotar a los dispositivos de capacidades cognitivas que simulen la inteligencia humana. Además del área técnica, entran en juego otras como la lingüística, la neurociencia o la psicología [94].

La Inteligencia Artificial (*Artificial Intelligence*, AI) pretende dotar a los dispositivos de ciertas capacidades como el análisis, la deducción, el procesamiento de datos, la manipulación de objetos o las capacidades sociales entre otras. Un ejemplo práctico de AI son todos aquellos asistentes tecnológicos que encontramos en los dispositivos personales como Siri (Apple) o Alexa (Amazon).

Dentro de AI, ML permite a partir de datos existentes generar un patrón o modelo para aplicar a los nuevos datos generados. Un caso práctico es el filtrado de correo basura en

cualquier bandeja de entrada a partir del histórico de observaciones previas [95]. Dentro de ML, la técnica Aprendizaje Profundo (*Deep Learning*, DL) se basa en redes neuronales profundas para encontrar los patrones o modelos, y son de gran interés al evidenciar buenos resultados en gran variedad de problemas como *Speech* o *Image Recognition*. Estas técnicas, aún por explotar, tendrán una gran aceptación en sectores como la medicina, los vehículos autónomos, la agricultura, la economía, la educación, etc.

En este capítulo se presentan dos propuestas para la inclusión de ML en la evaluación de prestaciones en la IoT.. La primera, aborda la problemática de los eventos de *stalling* en la reproducción de contenido multimedia, en caso de realizarse este tipo de transmisión sobre escenarios IoT.. La propuesta permite estimar cuándo se darán las interrupciones en función de los datos obtenidos a nivel de red y transporte de la arquitectura de comunicación. La segunda propuesta, se aplica sobre el modelo de evaluación de prestaciones de componentes QoX presentado en el Capítulo 3, con la finalidad de estimar el valor futuro de las métricas mediante series temporales y notificar con antelación las posibles caídas de prestaciones. Para esta segunda propuesta se empleará la implementación con dispositivos LoRa sobre diferentes escenarios y entornos.

5.2. Clasificación y Predicción de Eventos de *Stalling*

Pese a que a día de hoy aún existe cierta duda sobre la capacidad de los dispositivos IoT para soportar servicios más complejos como *video streaming*, es muy probable que lo

sean en un futuro no muy lejano, donde dispositivos de dimensiones muy reducidas puedan dar este servicio. Actualmente dispositivos como Raspberry Pi pueden desplegar un centro multimedia con equipos de pequeño tamaño, pero esta implementación aún debe mejorar considerablemente.

La popularidad de los servicios de *video streaming* ha crecido de modo exponencial en los últimos años, recaudando hasta 26 millones de dólares en 2020 y esperando llegar a la cifra de 1,3 miles de millones de clientes activos en 2024 [96]. Es por ello que el sector del entretenimiento digital está creando constantemente nuevos contenidos para su público, pero también nuevos despliegues tecnológicos para poder ofrecer a sus clientes precios más baratos y mejor calidad de los mismos [97].

En la evaluación de estos servicios siempre se han utilizado dos componentes principalmente: QoE y QoS. En este entorno, QoS continúa utilizando las métricas ya conocidas como *delay*, *jitter*, etc., mientras que QoE proporciona un abanico más amplio de métricas que infieren la percepción del usuario sobre la calidad recibida. Es por tanto QoE una métrica adoptada por la comunidad científica como la mejor para evaluar el rendimiento de los servicios de *video streaming*, ya que considera la calidad del video, el tiempo de carga, los eventos de *stalling*, etc. [98]. Se ha demostrado, que una de las métricas con mayor relevancia sobre la QoE son los eventos de *stalling*, ya que durante una reproducción multimedia cuantos más eventos de *stalling* se produzcan, menor será la calidad de la propia reproducción [98–103]. Estos eventos de *stalling* son causados por el agotamiento de los buffers de recepción, por lo que la reproducción del video se detiene

inesperadamente durante un intervalo de tiempo. Este intervalo puede oscilar de segundos a minutos. Por lo tanto, la identificación, la evaluación y la predicción de eventos de *stalling* en tiempo real es una herramienta muy valiosa en la mejora este servicio. En la actualidad, existen modelos objetivos de estimación de QoE en los servicios de *video streaming* como ITU-T Rec. P1023 [100,104]. En este modelo, uno de los parámetros de entrada es el número de eventos de *stalling* dados durante la reproducción, el cual se conoce una vez reproducido el video. Por tanto, la detección y predicción de este tipo de eventos podría dar lugar a acciones correctivas que mejorarán el rendimiento de la reproducción, y por tanto la QoE.

En este contexto, el objetivo que se pretende alcanzar con esta investigación es doble. Por un lado, centraremos nuestros esfuerzos en definir un método capaz de clasificar los paquetes de un flujo de *video streaming* como pertenecientes a un evento de *stalling* o no. Por otro lado, presentaremos un modelo temporal para predecir si el paquete futuro se clasificará como evento de *stalling* o no utilizando una red neuronal recursiva (*Recursive Neural Network*, RNN) y *Long Short Term Memory* (LSTM). El *dataset* que se ha empleado para este estudio, se obtuvo sobre un escenario de servicio de *video streaming* sobre una red *Long Term Evolution* (LTE). El escenario fue emulado y validado en [100,105]. Para la transmisión y recepción del contenido multimedia se emplearon dispositivos reales, mientras que para la red LTE se emuló utilizando NS3. El servicio de video se proporcionaba sobre *Dynamic Adaptive Streaming over HTTP* (DASH).

5.2.1. Estado de la Técnica

En la literatura científica se han propuesto numerosos trabajos para mejorar la calidad de los servicios de *video streaming*. En esta sección se presenta una visión general de la evaluación de QoS y QoE, el impacto de los eventos de *stalling* y como estos eventos pueden ser estimados o pronosticados.

DASH o MPEG-DASH [106] es un estándar para transmisión de video que emplea el protocolo TCP a nivel de transporte y un marco HTTP para entregar el contenido a nivel de aplicación [107] sin ser afectados por los *firewalls*. Una de las claves es que el dispositivo cliente es quien elige el *data rate* que mejor se adapta a sus condiciones. El servidor divide el video en segmentos más pequeños (generalmente de 2 a 10 segundos) y los codifica a diferentes niveles de compresión, por ejemplo, proporcionando diferentes *bit rates*. Todos estos segmentos estarán disponibles en el cliente. Diferentes *bit rates* y niveles de calidad entre segmentos permiten cumplir con el ancho de banda disponible, maximizando el uso dinámico de la red [102].

Los protocolos multimedia son sensibles a fluctuaciones afectando severamente a la reproducción de video, mitigando su efecto con el uso de *buffers* de recepción. Cuando un usuario pulsa el botón de reproducir, el cliente recibe el primer segmento en la cola de su *buffer* de recepción antes de comenzar la reproducción. Este proceso se conoce como *prebuffering*. Para la gestión de los *buffers* se utiliza el método *First-In First-Out* (FIFO), donde la entrada más antigua es procesada primero. Será necesario la retransmisión de un segmento cuando el número de secuencia no coincida con el esperado en el sistema. Esta

retransmisión no implica la parada de la retransmisión, ya que el cliente puede seguir procesando datos almacenados en su *buffer*. Un evento de *stalling* ocurre cuando el reproductor consume más segmentos que recibe, procediendo a vaciar el *buffer*, y finalizará cuando se reciban nuevos segmentos en el *buffer*.

Está demostrado que los eventos de *stalling* son uno de los factores más importantes a la hora de degradar la QoE en una reproducción multimedia, considerando tanto el número de apariciones como la duración de las mismas [25,98–103]. En un trabajo previo [100], validamos el modelo objetivo para la estimación de QoE en servicios de *video streaming* propuesto por la ITU-T Rec. P-1203 [108] bajo un escenario LTE mediante experimentación y emulación. Casey y Muntean [102] proponen una solución llamada *DASH-based Quality oriented Video Delivery* (DQVD) para reducir los eventos de *stalling* durante la reproducción del video bajo una arquitectura heterogénea mediante la selección dinámica de la interfaz inalámbrica. Este trabajo activa un segundo canal de conexión mediante dispositivos móviles para descargar el contenido necesario y mejorar temporalmente el rendimiento. La solución monitoriza y observa un incremento del rendimiento global del *buffer*. Cuando las condiciones mejoran en el canal principal, minimiza el uso del canal secundario hasta desconectarlo.

Desde una perspectiva diferente, Tao *et al.* [98] presentan una nueva técnica para procesar los datos recopilados del usuarios durante la reproducción de contenidos multimedia sobre redes móviles. Los autores seleccionan las mejores características y emplean técnicas de DL basadas en 7 capas para predecir valores de QoE basados en

parámetros como calidad visual, carga, *stalling* y puntuación general. De modo similar, Seufert *et al.* [99] concluyen que los eventos de *stalling* causados por *rebuffering* son el mayor degradante de QoE y focalizan su trabajo en predecir eventos de *stalling* en tiempo real. Los autores proponen un modelo de ML capaz de predecir la aparición de eventos de *stalling* a partir de métricas de QoE obtenidas en YouTube. Para entrenar el modelo hacen uso de la información de 4714 sesiones usando HTTP Adaptive Streaming (HAS).

Empleando también HAS, Bampis *et al.* [109] presentan un modelo testado *offline* de predicción de QoE basado en *Support Vector Machine* (SVM) y tres características: la densidad de eventos de *stalling*, tiempo desde el último evento de *stalling* y la calidad de reproducción de video. Wassermann *et al.* [110] estudian el comportamiento de QoE bajo LTE tomando datos de la capa de red y el contexto (tamaño de pantalla, orientación, modo de reproducción, volumen, etc.) durante la reproducción de videos de YouTube. Los autores usan el modelo ITU-T P.1203 para estimar el valor MOS y aplican diferentes algoritmos de ML como *Random Forest*, *Decision Tree*, SVM o *Naïve Bayes* para predecir el valor de QoE a partir de una selección de 30 características. Krishnamoorthi *et al.* [111] utilizan HAS como protocolo multimedia accediendo a las características del video y al estado del *buffer*. Este trabajo utiliza dos métodos: un clasificador basado en *threshold* y un modelo de ML basado en DT y SVM. Finalmente es Mazhar *et al.* [112] quien propone un algoritmo de ML supervisado para predecir el valor de QoE utilizando C4.5 DT y *Adaboost* para reducir los errores de clasificación. Se obtiene un gran número de parámetros de la ventana temporal de recepción y de la información de las cabeceras a nivel de red y transporte.

Nuestra hipótesis inicial es que la información contenida en las capas de red o de transporte (o una combinación de ambas) podría ser suficientemente buena para detectar la existencia de eventos de *stalling* o para predecir su ocurrencia (o ambas). De esta forma, evitaríamos los efectos del uso de técnicas de cifrado empleadas en capas superiores, cada vez más frecuente, o puntos de monitorización de la red (en caso de que se utilice información de la capa de transporte de extremo a extremo).

5.2.2. Test-bed

En esta sección se describe el escenario que se utilizó para obtener el *dataset*. El escenario planteado consta de tres dispositivos: un servidor de video real, una red emulada LTE y un cliente que reproduce el contenido multimedia (véase Fig. 35). El servidor de video emplea la herramienta *Wowza Streaming Engine* para distribuir el contenido multimedia. La red LTE es emulada mediante NS3 y el cliente consume el contenido multimedia haciendo uso del navegador web utilizando DASH. Para todos los test se emite una secuencia H.264 de “Big Buck Bunny” con una duración total de 180 segundos y una resolución de 720p. La red emulada emite de forma artificial retardos de 0 a 150 ms en pasos de 25 ms. Para cada uno de los retardos, los test se repiten 10 veces para evitar excepciones inesperadas recopilando un total de 70 capturas mediante Wireshark [113]. La Tabla 11 muestra las características del escenario emulado. Las capturas son tomadas en ambos extremos de la comunicación, pero se utilizan los datos del lado del cliente para llevar a cabo las tareas de clasificación y predicción que se describen en este capítulo.

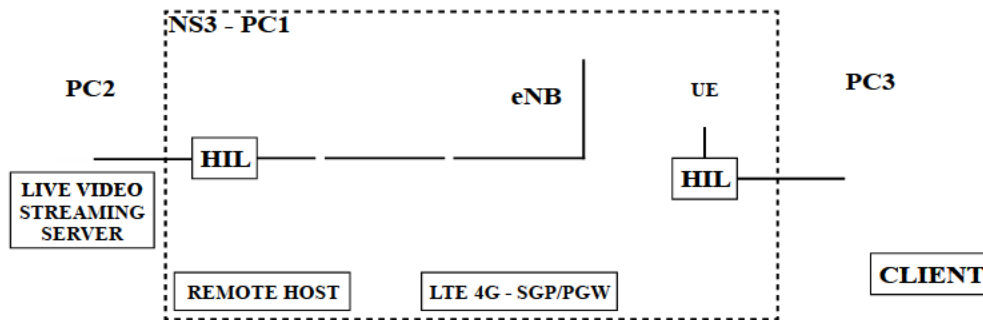


Fig. 35: Escenario de adquisición de datos.

Tabla 11: Características del entorno emulado.

Características	Descripción
<i>Número de dispositivos</i>	3
<i>Dispositivos</i>	<ol style="list-style-type: none"> 1. PC2 – Un servidor de <i>video streaming</i> en directo (Wowza Streaming Engine). 2. PC1 – Ejecutando una red emulada LTE bajo NS3. 3. PC3 – Cliente de video (Navegador web y Wireshark).
<i>Codecs</i>	H.264/MPEG-4 AVC
<i>Codificación de audio</i>	ACC/estéreo
<i>Resolución</i>	720p (1280 x 720)
<i>Bitrate de codificación (Kbps)</i>	2628
<i>Protocolo de transporte de video</i>	DASH
<i>Modelo de propagación</i>	Nakagami (m=5)
<i>Número de simulaciones</i>	10 por cada retardo (<i>delay</i>)
<i>Tipo de antena</i>	Coseno, altura 1,5 m
<i>Ruido eNB/UE</i>	2 dB/ 7 dB
<i>Operación UE</i>	26 dBm, altura 2,5 m, distancia 50 m

Es importante remarcar que el *dataset* del que disponemos es un *dataset* no balanceado. Suponemos que el rendimiento de la red es favorable y que generalmente no se producen

caídas de rendimiento. Bajo este supuesto, los datos capturados de la red serán generalmente no balanceados ya que los eventos de *stalling* ocurrirán ocasionalmente, siendo una clase minoritaria (no es el estado natural del sistema). Sin embargo, la clase minoritaria sería la más importante desde la perspectiva de la QoE, ya que está vinculada a un comportamiento que debe evitarse o limitarse.

A pesar de trabajar con un *dataset* no balanceado, nuestro objetivo es utilizarlo para proponer mecanismos eficientes capaces de clasificar un paquete como perteneciente (o no) a un evento de *stalling*. Mientras que la mayoría de algoritmos regulares de aprendizaje se generalizan en *dataset* equilibrados, un *dataset* no balanceado necesitan ser abordados desde una perspectiva diferente [109,110], de lo contrario, la clase minoritaria sería más débil que la clase mayoritaria.

Sun *et al.* [114] presentaron una revisión de los problemas de clasificación no balanceados en diferentes aplicaciones e informaron de diferentes soluciones. También, los trabajos realizados por He *et al.* [115], Krawczyk [116], y Vluymans [117] explican cómo gestionar los datos no balanceados y su asimetría, centrándose en la eficiencia computacional y los métodos adaptativos. Existe un acuerdo unánime sobre la necesidad de preprocesamiento en estos casos, donde la mayoría de técnicas se pueden encontrar dentro de *Scikit-Learn* [118]. *Scikit-Learn* es una herramienta de código abierto que proporciona análisis predictivo haciendo uso de bibliotecas adicionales como *NumPy*, *SciPy* y *matplotlib*. El primer paso del preprocesamiento es reducir el número de variables respecto al *dataset* original, pudiendo ignorar alguna de ellas. Alguna de las técnicas más

conocidas son *Recursive Feature Elimination* (RFE), *Principal Feature Analysis* (PFA), *Independent Component Analysis* (ICA) y *Fisher Linear Discriminate Analysis* (LDA) [114,115]. Todos estos métodos siguen un proceso similar, se crea un subconjunto de variables y una de ellas se excluye del grupo cada iteración. Por el contrario, *Principal Component Analysis* (PCA) y *Forward Selection Component Analysis* (FSCA) reducen el número de variables por aproximación, pudiendo reconstruir la mayor parte de la información original [119].

5.2.3. Preprocesado de los datos

En las capturas analizadas del *dataset*, el volumen de datos de las diferentes capas de la arquitectura de comunicación es enorme, pero para esta contribución nos centramos en la capa de red y la de transporte. La Tabla 12 muestra las características extraídas de cada paquete que circula por la red. Dentro de este conjunto de características, se determinarán aquellas más relevantes y para ello se utiliza el algoritmo *Principal Feature Analysis* (PFA). PFA permite calcular la matriz de correlación entre las características del dataset y a partir de ellas, los vectores V_i que representan la proyección i -ésima del vector de características X para un subconjunto de características. De modo que cuanto mayor es la correlación entre las características, mayor es el valor absoluto del vector V_i . El algoritmo PFA asocia en un mismo grupo las características con alta correlación, seleccionando sólo una de ellas que representará de forma óptima el conjunto de características incluidas en el mismo.

Tabla 12: Características extraídas por cada paquete en el dataset.

Características	Descripción
<i>tcp_seq_num</i>	Número de secuencia TCP
<i>tcp_win_siz</i>	Tamaño de ventana TCP
<i>tcp_fast_ret</i>	Trama con retransmisión rápida
<i>Tcp_ack_dup</i>	Trama con ACK duplicado
<i>Tcp_prev_NC</i>	Trama anterior perdida
<i>Tcp_pkt_ooo</i>	Trama TCP fuera de orden
<i>Pkt_len</i>	Longitud de la cabecera IP
<i>Ip_len</i>	Longitud del paquete IP
<i>Tcp_heh_len</i>	Longitud de la cabecera TCP
<i>Tcp_flag_ack</i>	<i>Flag</i> TCP ACK
<i>Tcp_flag_urg</i>	<i>Flag</i> TCP URG
<i>Tcp_flag_psh</i>	<i>Flag</i> TCP PUSH
<i>Tcp_flag_rst</i>	<i>Flag</i> TCP RESET
<i>Tcp_flag_syn</i>	<i>Flag</i> TCP SYN
<i>Tcp_flag_fin</i>	<i>Flag</i> TCP FIN

Tras procesar el *dataset* y emplear técnicas de *cross-validation*, detectamos que la característica más relevante en la aparición de eventos de *stalling* es el *flag* TCP SYN (véase la Fig. 36). Aunque existen otras características también relevantes, como el número de secuencia o el tamaño de la ventana TCP, decidimos centrar en una única característica para simplificar el proceso de clasificación.

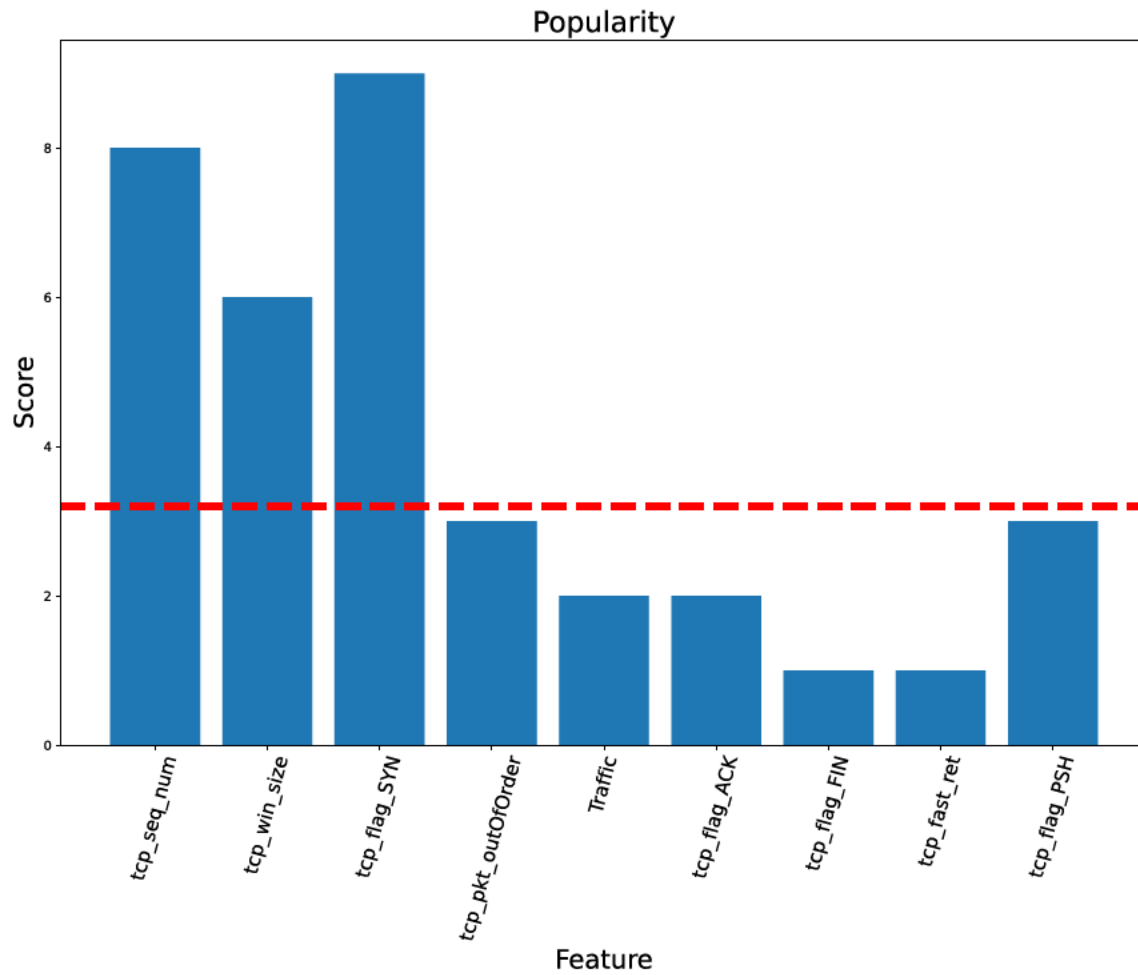


Fig. 36: Selección de características más relevantes para el proceso de clasificación de eventos de stalling.

Así, planteamos el uso de una ventana deslizante de tamaño t para analizar esta característica y tomar decisiones de clasificación acorde a los paquetes recibidos recientemente. El tamaño de esta ventana deslizante podría ser variable, por lo que los resultados de diferentes tamaños han sido comparados para estudiar su rendimiento. De forma experimental veremos que se obtiene un mejor rendimiento para ventanas deslizantes de tamaño mayor de 512 paquetes, llegando a la conclusión de que con ventanas

deslizantes más pequeñas se produce un efecto de *overfitting*, pues la cantidad de datos es inconsistente de cara a la toma de decisiones final.

5.2.4. Métricas de aprendizaje

En estas circunstancias cabe destacar que trabajamos con un escenario binario, en el que además el número de paquetes correctamente recibidos supera al número de paquetes erróneos. Para los problemas de clasificación binaria emplearemos la matriz de confusión basada en cuatro componentes (véase Fig. 37): *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) y *False Negative* (FN). En TP, tanto el valor estimado como el real coinciden tomando valor positivo, mientras que en TN el valor estimado y el real son ambos negativos. Por el contrario, en los FP y FN los valores estimados y los reales no coinciden. Los FP suceden cuando el valor estimado es positivo y el valor real es negativo, mientras que los FN toma valor estimado negativo y valor real positivo.

	Predicted Positive	Predicted Negative
Real Positive	True Positive (TP)	False Negative (FN)
Real Negative	False Positive (FP)	True Negative (TN)

Fig. 37: Matriz de confusión.

Tradicionalmente, el proceso de aprendizaje artificial ha tomado la métrica de *Accuracy* para ajustar y mejorar el rendimiento del modelo. Esta métrica se define en este ámbito como el número de estimaciones realizadas correctamente por el modelo sobre el total, aunque en ocasiones puede resultar poco precisa. Como alternativa, se presentan métricas adicionales como *Precision*, *Recall*, F1 y *Area Under Curve* (AUC). *Precision* evalúa el número de muestras etiquetadas sin errores como la relación entre muestras positivas correctamente estimadas respecto al total de muestras que se han estimado como positivas (19). *Recall* es el número de muestras positivas determinadas entre todas las muestras positivas reales (20). F1 es el resultado de una media armónica, que busca un equilibrio entre *Precision* y *Recall* en forma de una única métrica dotando de robustez al clasificador (21). Por último, AUC evalúa el modelo de acuerdo a la distinción entre clases (22); a mayor valor, mayor separación entre clases (22). El rango de todas las métricas anteriormente descritas va desde 0 hasta 1 [0, 1] [120].

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

$$F1 = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \quad (21)$$

$$AUC = \frac{TP(TN + FP) + TN(TP + FN)}{2(TP + FN + TN + FP)} \quad (22)$$

5.2.5. Modelo de clasificación basado en *threshold*

La primera solución presentada trata de analizar el comportamiento del *flag* TCP SYN en sesiones multimedia, al ser la característica identificada como la más relevante en el análisis previo realizado. Para ello, como hemos indicado anteriormente, haremos uso de una ventana deslizante de tamaño fijo, lo que permitirá facilitar el proceso.

La ventana deslizante tendrá un tamaño t , siendo t mucho menor que el número de paquetes totales T de la captura a analizar ($t \ll T$). Se define también un vector L de tamaño N , donde N (23) representa el número de ventanas deslizantes posibles para una captura de longitud T y un tamaño de ventana deslizante t . Cada posición del vector L indica el número de paquetes en la ventana deslizante con el *flag* TCP SYN activo. Una vez se ha obtenido todo el vector L , se calculará el valor *threshold* th como la suma del valor medio y la desviación típica del vector. Cuando una posición del vector L supera el valor umbral th , el paquete se marcará como candidato a ser un evento de *stalling*.

$$N = T - t + 1 \quad (23)$$

Lo veremos con un ejemplo. Supongamos una ventana deslizante de tamaño igual a tres ($t=3$) y un conjunto de datos de longitud igual a 6 ($T=6$). Aplicando (23), el vector L tendría 4 elementos ($N=4$). Es decir, hay 4 ventanas consecutivas que contienen diferentes cuadros a través del conjunto de datos (véase Fig. 38). La primera ventana deslizante w_1 cubre desde $i=1$ hasta $i=3$ (ambos incluidos). Si la suma de *flag* TCP SYN en esas 3 posiciones es 3, entonces $L[0]=3$. Para la segunda iteración, la ventana deslizante w_2 cubre

desde $i=2$ hasta $i=4$ (ambos incluidos). Si la suma de *flags* TCP SYN en estas tres posiciones es 2, entonces $L[1]=2$, y así sucesivamente hasta la última ventana deslizante w_4 (por ejemplo, $L=[3, 2, 2, 1]$). Una vez calculado el vector L , se obtiene el valor medio (24) y la desviación típica (25) de todos sus elementos. Definimos el valor umbral th como se muestra en (26). En nuestro ejemplo, th sería igual a 2,71 (el valor medio es igual a 2 y la desviación típica es igual a 0,71). Para clasificar un paquete i como perteneciente a un evento de *stalling* (o no), observamos las posiciones en el vector L afectadas por el paquete i . Por ejemplo, el primer paquete ($i=1$) aparece sólo en la primera ventana w_1 , por lo que sólo contemplamos $L[0]$. Si $L[0] > th$, entonces el paquete i (en este caso $i=0$) pertenece a un evento de *stalling*. En el ejemplo, $3 > 2,71$, por lo que se asume que el primer paquete pertenece a un evento de *stalling*. El segundo paquete ($i=2$) aparece en las dos primeras ventanas deslizantes w_1 y w_2 , por lo que necesitamos calcular el valor promedio de $L[0]$ y $L[1]$; en este caso, un valor 2,5. Por tanto, si $avg(L[0], L[1]) > th$ este paquete sí pertenece a un evento de *stalling*; en caso contrario no.

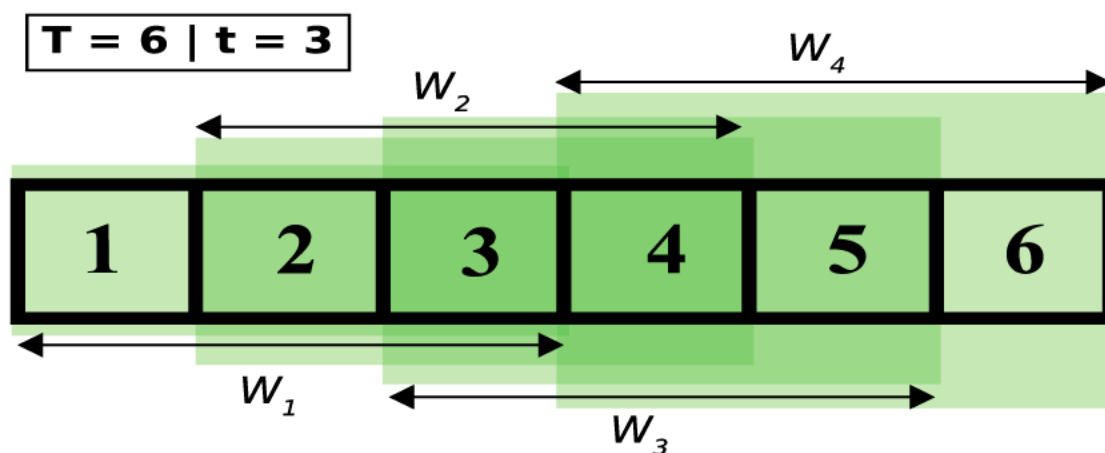


Fig. 38: Ejemplo de ventana deslizante para el proceso de clasificación de eventos de *stalling*.

Teniendo en cuenta que cualquier paquete aparecerá en t ventanas deslizantes, excepto los primeros y los últimos paquetes, el paquete i (donde $t \leq i < T-t+1$) pertenece a las ventanas deslizantes desde w_{i-t+1} hasta w_i . Por consiguiente, para clasificar el paquete i , se evaluarán las posiciones desde $L[i-t]$ a $L[i-1]$ (27).

$$avg(L) = \sum_{n=1}^N \frac{L[n]}{N} \quad (24)$$

$$std(L) = \sqrt{\frac{1}{N} \sum_{n=1}^N (L[n] - avg(L))^2} \quad (25)$$

$$th = avg(L) + std(L) \quad (26)$$

$$Frame\ i \begin{cases} \text{if } i < t & \text{eval. from } L[0] \text{ to } L[i-1] \\ \text{if } t \leq i \leq T-t+2 & \text{eval. from } L[i-t] \text{ to } L[i-1] \\ \text{if } i > T-t+1 & \text{eval. from } L[i-t+1] \text{ to } L[t-1] \end{cases} \quad (27)$$

Algorithm 1 *Modelo de clasificación basado en threshold*

```
1:  procedure DATASETSPLITTING (data(ex, cap))
2:    experiment = 150
3:    while experiment ≥ 0 do
4:      capture = 10
5:      while capture ≥ 0 do
6:        if capture > 2 then
7:          train = data(experiment, capture)
8:        else
9:          test = data(experiment, capture)
10:       end if
11:       capture = capture - 1
12:     end while
13:     experiment = experiment - 25
14:  end while
15:  end procedure
16:  procedure GETTHRESHOLDVALUE(train, t)
17:    dataX = []
18:    T = len(train)
19:    while i ≤ T - t + 1 do
20:      a = train[i : (i + t)]
21:      dataX = a.append(a)
22:      i = i + 1
23:    end while
24:    dataTrain = sum(dataX, 1)
25:    th = mean(dataTrain) + std(dataTrain)
26:    return th
27:  end procedure
28:  procedure SLIDINGWINDOWS(test, t)
29:    dataY = []
30:    T = len(test)
31:    while i ≤ T - t + 1 do
32:      a = test[i : (i + t)]
33:      dataY = a.append(a)
34:      i = i + 1
35:    end while
36:    dataTest = sum(dataY, 1)
37:    return dataTest
38:  end procedure
39:  procedure CHECKINGTHRESHOLD(th, a)
40:    if a.mean() ≥ th then
41:      st_event.append(1)
42:    else
43:      st_event.append(0)
44:    end if
45:  end procedure
46:  procedure STALLINGDETECTION(dataTest, th, t)
47:    st event = []
48:    T = len(dataTest)
49:    while i ≤ T + t do
50:      if i < t then
51:        a = dataTest[0 : i]
52:        CheckingThreshold(th, a)
53:      else if i > T then
54:        a = dataTest[i - t : ]
55:        CheckingThreshold(th, a)
56:      else
57:        a = dataTest[i - t : i]
58:        CheckingThreshold(th, a)
59:      end if
60:      i = i + 1
61:    end while
62:  end procedure
```

Para evaluar este método de clasificación de paquetes como pertenecientes o no a un evento de *stalling*, vamos a comparar la clasificación obtenida tras aplicar el modelo con los eventos de *stalling* registrados mediante experimentación en [100]. En primer lugar, observamos que los eventos de *stalling* obtenidos mediante experimentación tienen un leve retraso respecto a los estimados mediante el modelo propuesto (véase Fig. 39). Se aprecia que el evento de *stalling* se detecta con antelación en el modelo planteado comparado con el tiempo en el que el usuario final lo experimenta. Esto puede considerarse una ventaja, ya que podría permitir al sistema tomar medidas y mejorar la opinión del usuario.

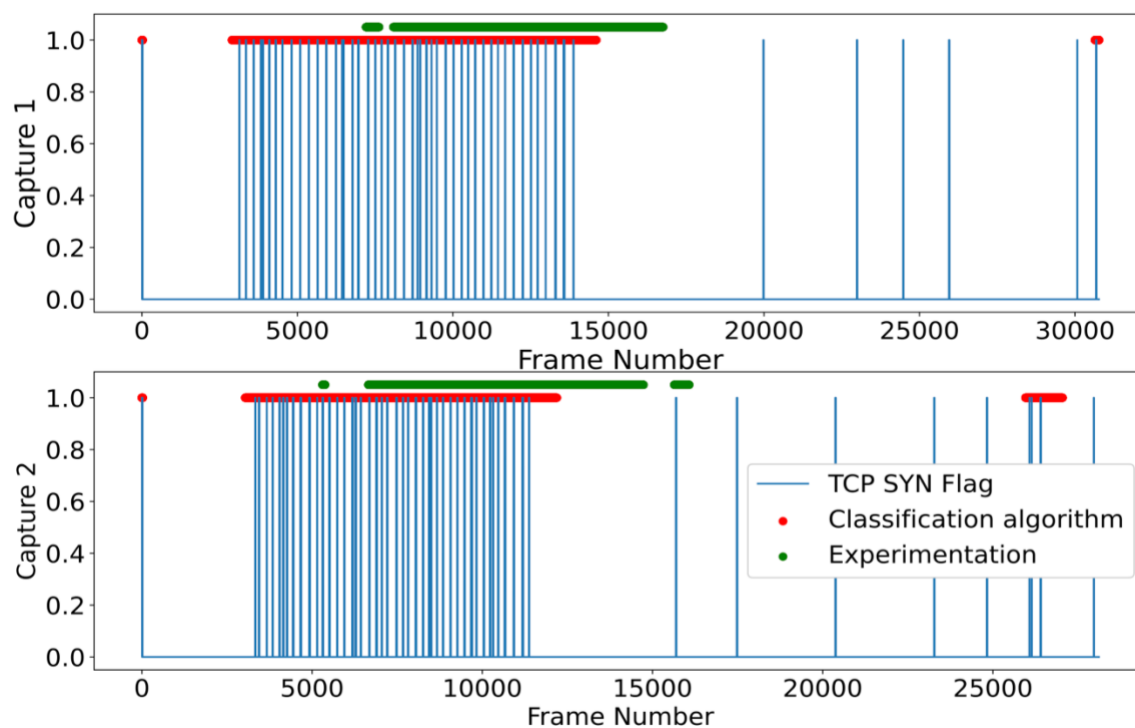


Fig. 39: Comparación de los resultados de experimentación vs. resultados del modelo de clasificación.

Como se mencionó previamente, también se han comparado los resultados obtenidos utilizando diferentes tamaños de ventanas deslizantes. Para esta comparación se ha usado el valor AUC como métrica. Así, podemos observar que el modelo trabaja mejor con valores de t mayores, reduciendo la cantidad de FP, ya que de lo contrario, se produce *overfitting* por insuficiencia de datos (véase la Fig. 40).

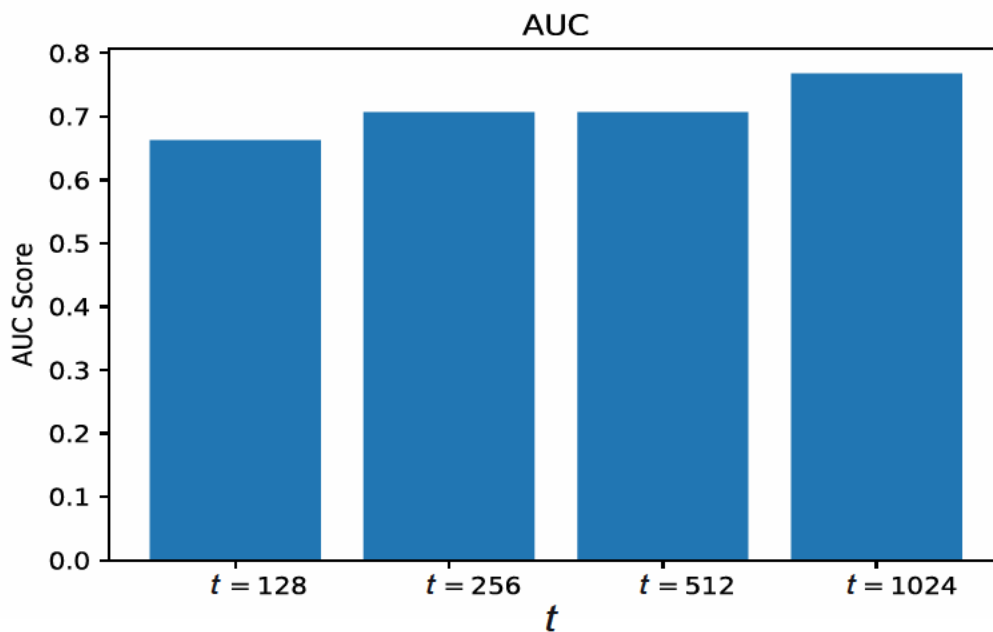


Fig. 40: Comparativa de prestaciones para diferentes tamaños de ventana deslizante (t) en el modelo de clasificación.

5.2.6. Modelo de predicción de eventos de stalling

En esta sección se describe y evalúa el modelo propuesto para predicción de eventos de *stalling*. El objetivo del modelo propuesto es predecir el comportamiento de paquetes futuros.

Uno de los métodos más populares para predicción reside en el uso de Redes Neuronales Artificiales (artificial Neural Network, NN). A pesar de los avances en NN, su uso con conjunto de datos no balanceados aún no está completamente consolidado. En la encuesta de Johnson y Khoshgoftaar [121], se describieron los detalles de la aplicación de diferentes casos, destacando sus puntos fuertes y débiles en el tratamiento de datos no balanceados en aspectos relacionados con la complejidad y la arquitectura de datos, la evaluación del rendimiento o la facilidad de uso. Particularmente, focalizaremos en RNN que utilizan LSTM. LSTM fue introducido por Hochreiter y Schmidhuber [122] para superar y explorar los problemas de gradientes ganando popularidad en el análisis predictivo gracias a su capacidad para predecir series temporales de datos [123–125].

En consecuencia, esta propuesta se basa en RNN ya que el estado de la memoria influye en el proceso de toma de decisiones. En particular, utilizará el LSTM. Del mismo modo que en el modelo de clasificación, utilizaremos una ventana deslizante de tamaño t para la toma de decisiones. Del mismo modo que en el modelo de clasificación, utilizaremos una ventana deslizante de tamaño t para tomar decisiones. En este caso, tendremos en cuenta los t paquetes más recientes para predecir si el siguiente paquete será etiquetado como evento de *stalling* o no. El tamaño de la ventana deslizante seleccionada coincide con la empleada en el caso anterior, donde t adquiere el valor 1024, evitando así el efecto de *overfitting* producido con tamaños de ventana menores.

La versión de LSTM que vamos a utilizar se denomina *Attention enhanced Bidirectional Long Short-Term Memory* (ABi-LSTM), con bloques de memoria a largo plazo y

propagación *back and forward*. Para gestionar el estado y obtener la predicción del siguiente paquete, estos bloques de memoria usan *gates*, tanto de entrada como de salida. Por último, se utiliza *forget gate* para actualizar la memoria del modelo. Es importante remarcar que el modelo que proponemos está basado en la clasificación de secuencias utilizando dos LSTM como secuencia de entrada como muestra la Fig. 41. La primera añade una copia de los datos de la ventana deslizante tal cual y la otra una copia invertida, lo que permite un aprendizaje más rápido y profundo [124,126,127].

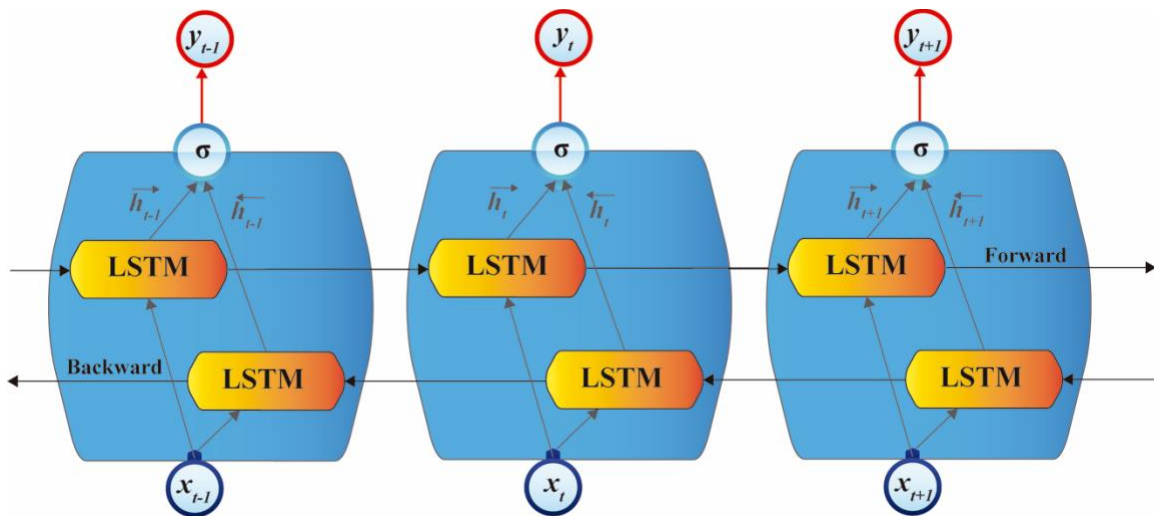


Fig. 41: Modelo de predicción ABi-LSTM.

El proceso de entrenamiento actualiza los pesos w de los coeficientes del modelo iteración tras iteración, utilizando *Stochastic Gradient Descent* (SGD) (28), optimizando los pesos para minimizar la función de pérdidas. Dentro de este proceso de optimización entran en juego los parámetros *learning rate* (η) y el factor de regularización, que permiten regular la velocidad de aprendizaje y gestionar el efecto del *overfitting* [120]. Para este modelo, se ha utilizado un factor de regularización igual a 0,3 con la técnica Dropout. Esta

técnica es la más potente en DL, permitiendo que el modelo no se adapte a un conjunto específico de variables.

$$w := w + \Delta w; \quad \Delta w = -\eta \nabla J(w) \quad (28)$$

El modelo propuesto tiene una capa de entrada, una capa de salida y una capa oculta, siendo las conexiones entre las diferentes capas bidireccionales. Cada capa tiene diferentes unidades de conexión: I unidades en la capa de entrada, H unidades en la capa oculta y K unidades en la capa de salida. La complejidad computacional de un modelo LSTM bidireccional viene dada por $O(2n)$ donde n es el número de pesos [122]. Para este caso, n puede ser expresado como muestra (29), con $I=1024$, $H=1024$ y $K=1$. Para cada ciclo, el tiempo medio de entrenamiento es de 2650 segundos y 60 segundos para evaluación utilizando un servidor con los recursos *hardware* descritos en Tabla 13.

$$n = 4IH + 4H^2 + 3H + HK \quad (29)$$

Tabla 13: Detalles del servidor utilizado para simulación.

Características	Descripción
<i>Descripción del equipo</i>	Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
<i>CPU(s)</i>	60
<i>Thread(s) per core</i>	2
<i>Core(s) per socket</i>	10

Algorithm 2 *ABi-LSTM prediction model*

```

1: procedure DEFINITIONANDTRAINING
   (trainX[k] trainY[k], testX[k], testY[k])
2:   model = Sequential();
3:   Bidirectional(LSTM(activation = 'relu'));
4:   Dropout(0.3);
5:   Dense(activation = 'sigmoid' );
6:   model.compile(loss =binary_crossentropy,
   metrics = [accuracy,F1,Recall,Precision],
   opt = SGD(lr = 0.0001))
7:   model.fit(trainX[k],trainY[k])
8:   return model.evaluate(testX[k],testY [k])
9: end procedure
10: procedure CV ABI-LSTM(data, K)
11:   Partition randomly the data in K subsets
   (DT1, ..., DTk)
12:   for k=1 to K:
13:     Construct the kth training set with K - 1 subsets
14:     Construct the kth testing set with 1 subset, different to
   training set.
15:     result[k] = DEFINITIONANDTRAINING
   (trainX[k], trainY[k], testX[k], testY[k])
16:   end for
17:   Represent results for Loss Function, Accuracy, Precision, Recall
   and F1 Score for each fold.

```

Los resultados de este modelo se muestran haciendo uso de la técnica *k-Fold Cross Validation*, un método estadístico que estima la capacidad del modelo en diferentes subconjuntos de datos dentro del conjunto de datos. Este método tiene un único parámetro *k* que indica el número de subconjuntos en los que se divide el conjunto de datos. En cada grupo se escogen *k-1* subconjuntos para entrenamiento y uno para *testing*. Se guardan los

resultados para cada subconjunto y se comparan los resultados a posteriori. Los resultados obtenidos con *k-Fold Cross Validation* suelen ofrecer una estimación menos sesgada u optimista de la capacidad del modelo. Para este estudio, *k* tomará el valor 10, por lo que el método de validación cruzada se llevará a cabo en 10 ocasiones.

Tras el proceso de entrenamiento y validación, el modelo propuesto ABi-LSTM es capaz de predecir el estado del siguiente paquete (*stalling* o no) con unas prestaciones excelentes. No obstante, el proceso de aprendizaje es lento para el conjunto de datos recogido, con un valor de η de 0,0001. Para casi todos los subconjuntos probados se consiguen unos resultados notables en términos de *Precision*, *Recall* y F1, como muestra la Fig. 42. Por ejemplo, en uno de los conjuntos de prueba, el modelo sólo falla al clasificar 2936 paquetes de un total de 27091, lo que se traduce en una tasa de error del 10,83%. La Tabla 14 incluye los resultados de rendimiento de las métricas para este ejemplo en concreto.

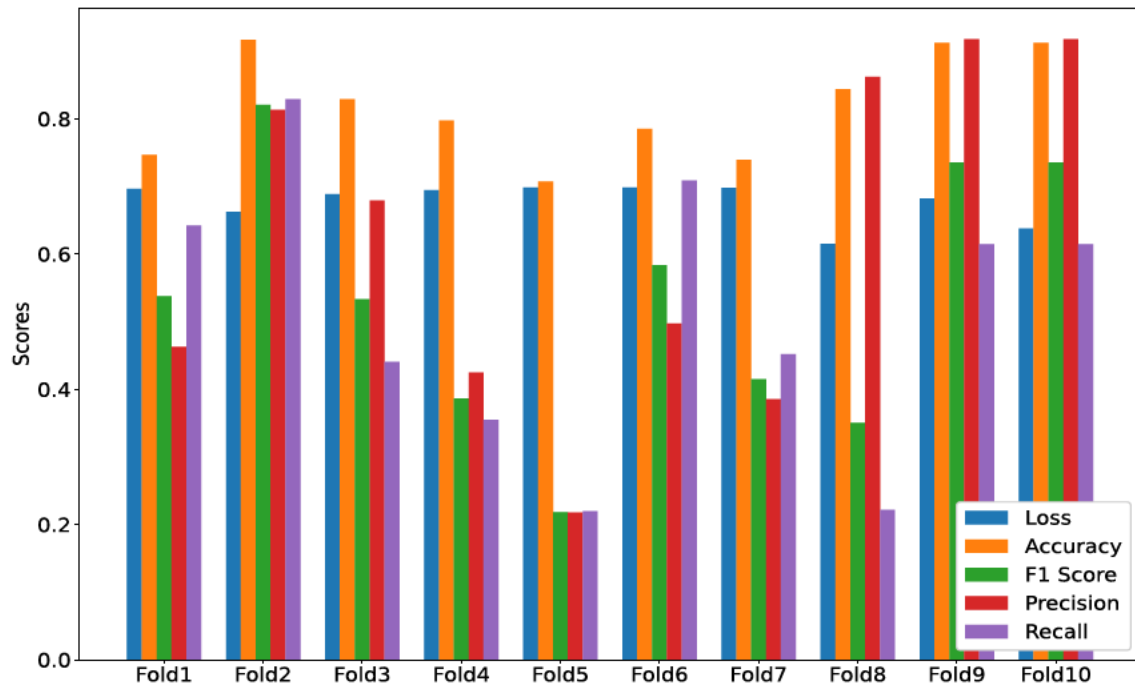


Fig. 42: Resultados del método 10-Fold Cross Validation para el modelo ABi-LSTM de predicción de ocurrencia de eventos de stalling.

Tabla 14: Ejemplo del desempeño del modelo de predicción para una captura.

Métrica	Valor
<i>TP</i>	6467
<i>FP</i>	821
<i>FN</i>	2115
<i>TN</i>	17688
<i>Precision</i>	$\frac{TP}{TP + FN} = \frac{17688}{17688 + 821} = 0.956$
<i>Recall</i>	$\frac{TP}{TP + FP} = \frac{17688}{17688 + 2115} = 0.893$
<i>F1</i>	$\frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} = \frac{2 \cdot (0.956 \cdot 0.893)}{0.956 + 0.893} = 0.923$

5.3. Análisis Predictivo en la Evaluación de Rendimiento con el Modelo de Componentes de Calidad QoX

Como hemos visto anteriormente, es complejo y, a la vez, importante definir un modelo para la evaluación de prestaciones en entornos IoT versátiles, donde varía el número y tipo de dispositivos, su ubicación, los recursos hardware asignados, los protocolos o parámetros de comunicación, el número de capas de la arquitectura, etc. Al tratarse de un reto innovador, existen algunas contribuciones científicas en la literatura especializada que ya han comenzado a visualizar el potencial del uso de técnicas AI en la evaluación de prestaciones.

Por ejemplo, el escenario planteado por Suryanegara *et al.* [128] es similar al caso de estudio que se ha utilizado en esta tesis doctoral. Su objetivo es medir el nivel de QoE en un entorno de servicios IoT a partir de unos parámetros de calidad específicos. Aloqaily *et al.* realizan en [129] un análisis completo de la QoE en un entorno inalámbrico, empleando una estimación de QoE muy sencilla. En otra contribución, los autores [130] emplean capacidades cognitivas para la optimización de recursos para las radiocomunicaciones en IoT. En la misma línea, Minovski *et al.* [131] abordan la temática de capacidades cognitivas sobre IoT utilizando métricas objetivas y comparten con nuestro modelo de evaluación el uso de las componentes QoD y QoE. Aunque los autores hacen referencia al uso de AI y ML, no se concreta en la aplicación de ninguna técnica.

Mientras que el ámbito de 5G podemos encontrar numerosas propuestas que aplican conceptos y técnicas de AI, como [132,133] donde se propone optimizar la precisión de estas redes minimizando costes mediante técnicas DL, no son tantos los trabajos científicos que abordan el uso de estas técnicas para la evaluación de prestaciones. Deebak *et al.* [134] plantean un *framework* con el que monitorizar y minimizar los costes de computación y comunicación. Este caso de estudio sigue nuestro enfoque de utilizar una componente tipo QC, estimando los autores un valor óptimo de esta para asignar eficientemente los recursos disponibles en cada momento. Casas *et al.* [135] utilizan un entorno de redes móviles donde claramente se distinguen dos fases: (i) generar un nuevo *dataset* y (ii) estimar el valor de las métricas de calidad utilizando técnicas de ML basadas en árboles de decisión. Este caso de estudio obtiene un buen rendimiento, utilizando modelos más complejos basados en DL y series temporales sobre un entorno IoT. También en redes inalámbricas, Sui *et al.* [136] presentan un modelo basado en LSTM para detectar y predecir anomalías en tiempo real en este tipo de redes. Estas anomalías son el origen de caídas de rendimiento en cualquier dispositivo o red.

De forma similar, López-Martin *et al.* [137] aplican un modelo de DL basado en CNN y RNN para predecir valores actuales y futuros de QoE en series temporales. También con foco en la QoE, Ibarrola *et al.* [138] hacen hincapié en la correcta selección de los KPIs más relevantes para estimar correctamente el valor de QoE en un escenario WiFi mediante algoritmos de ML supervisados y no supervisados. Li *et al.* [89] proponen un modelo en un escenario IoT que utiliza regresión multilinear para estimar valores de QoE y Asma Ben

Letaifa [123] proponen un método para estimar el valor de QoE basado en la escala MOS pero limitado a escenarios web.

En comparación con estos trabajos previos, nuestra contribución en este sentido será la incorporación de un sistema que permita el análisis predictivo de todas las componentes de calidad del modelo de evaluación de prestaciones para los servicios sobre IoT descrito en el capítulo 3.

5.3.1. Test-bed

Partimos de los datos generados mediante simulación para demostrar la validez del modelo holístico para medir el nivel de rendimiento en un entorno IoT. En concreto, se hace uso de 53 dispositivos LoRa ubicados en el área monitorizada, simulando un sistema de monitorización de calidad del aire donde cada nodo sería una estación de medición, y que trabaja de continuadamente durante 360.000 segundos.

Los datos de calidad de aire incluidos en la simulación forman parte de un *dataset* real [93] y recoge los mismos parámetros descritos con anterioridad (NO, NO₂, NO_x, PM₁₀, PM_{2.5}, velocidad del viento, temperatura, velocidad del viento y humedad). La generación de paquetes por parte de los dispositivos LoRa sigue una distribución exponencial, respetando las limitaciones de la tecnología (*duty cycle*). Las simulaciones se desarrollan bajo dos escenarios: (i) uno ideal desde la perspectiva del *backbone*, donde sólo habrá pérdidas o retardos en la comunicación inalámbrica, y (ii) otro en el que se añaden pérdidas con probabilidad del 10% y un retardo aleatorio entre 0 y 100 ms en el *backbone*. Para cada

escenario se evalúan tres entornos diferentes, (urbano, suburbano y rural). Sobre cada escenario y entorno se obtienen los valores de las componentes de calidad descritas en el capítulo 3, esto es, QoD, QoI, QoE y QC.

El modelo planteado está pensado para predecir valores futuros de cada una de las componentes mediante la utilización de técnicas de DL y series temporales. Para ello, la técnica LSTM Bidireccional ofrece un buen rendimiento en este cometido combinado con el uso de ventanas deslizantes. Es importante optimizar el tamaño de la misma ya que dependiendo de este, hará que nuestra estimación sea más estable o más sensible a cambios inmediatos. Para este caso en concreto, el tamaño de la ventana deslizante es pequeño ya que necesitamos basar nuestra decisión en función de los últimos valores de las métricas registrados.

5.3.2. Preprocesado de datos

En primer lugar, se importan los resultados de la simulación en un software de procesamiento de datos como Python. En los resultados se incluyen datos de 10 simulaciones independientes con el objetivo de normalizar la componente aleatoria que se introduce en cada ejecución.

Para evitar el efecto *overfitting* en nuestro modelo, los datos recopilados se dividen en datos de entrenamiento y de test, incorporando un 75% y 25% de los datos respectivamente. Con los primeros, el modelo aprende y utiliza los segundos para comprobar el acierto en sus precisiones.

El último paso en esta fase es la utilización de ventanas deslizantes de tamaño t , donde t es mucho menor que el número total de datos recopilados. La técnica empleada para la implementación de ventanas deslizantes está descrita en la sección 5.2.5. En este caso concreto, la ventana deslizante tiene un tamaño igual a 5 ($t=5$) como se muestra en la Fig. 43. Por lo tanto, se requiere un histórico de cinco datos para poder estimar el valor del sexto. Basándonos en la imagen anterior, necesitamos saber los valores de A, B, C, D y E para poder estimar F, y así sucesivamente, hasta el último dato recibido. Cuanto menor es el tamaño de ventana, más volátil es el valor resultante de la predicción y mejor se puede adaptar a los cambios repentinos en nuestra red.

A	B	C	D	E	F
B	C	D	E	F	G
C	D	E	F	G	H
D	E	F	G	H	I
...
U	V	W	X	Y	Z

Fig. 43: Ventana deslizante con $t=5$ para la predicción del valor de la componente de calidad QoX.

5.3.3. Modelo de predicción y sistema de notificación

Al contrario que en casos de estudios anteriormente descritos, en este se plantea un modelo de estimación basado en regresión lineal mediante la técnica LSTM bidireccional y ventanas deslizantes. Este modelo utiliza descenso de gradiente estocástico como optimizador y la técnica de error cuadrático medio como función de pérdidas. Para cada componente a estimar se entrena un modelo diferente, por lo que la estimación de las componentes es independiente una de otra. Al final de cada fase de entrenamiento se obtiene la gráfica que representa la función de pérdidas del modelo planteado.

Una vez el modelo está entrenado, se utilizan los datos reservados para la fase de entrenamiento. El objetivo es medir el rendimiento alcanzado por el modelo propuesto para cada componente, comparando los resultados de la predicción respecto a los reales. La comparativa será con el uso de gráficas y las métricas de rendimiento en estimaciones regresivas utilizando R^2 score. Esta métrica se calcula según (30) y varía en el rango [0,1], donde 1 es el valor óptimo de dicha métrica. La fórmula minimiza según el valor obtenido al dividir el error cuadrático medio (*Mean Squared Error*, MSE) del modelo planteado, respecto a la MSE de la línea base, que es la MSE que obtendría el modelo mas simple posible (31).

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)} \quad (30)$$

$$MSE(baseline) = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (31)$$

Por último, se plantea un sistema de notificación de las caídas de rendimiento experimentadas en el sistema. En este sentido, el sistema notificará la caída de rendimiento cuando se produzca una caída de rendimiento del 50% respecto al T_{eval} anterior y esta caída se sostenga en el tiempo (varios T_{evals} consecutivos).

En este caso de estudio, utilizamos la matriz de confusión para evaluar los resultados obtenidos en el sistema de notificación y las métricas *Precision*, *Recall* y *F1* vistas en sección 5.2.4.

5.3.4. Resultados

Al finalizar la fase de entrenamiento se obtiene la función de pérdidas. Las funciones de pérdidas para un mismo entorno y métrica son poco variables, aunque sí diferentes entre escenarios. Fig. 44(a), Fig. 44(b), Fig. 44(c) y Fig. 44(d) representan la función de pérdidas en el escenario ideal descrito anteriormente para QoD, QoI, QoE y QC, respectivamente. Del mismo modo, Fig. 45(a), Fig. 45(b), Fig. 45(c) y Fig. 45(d) representan la función de pérdidas pero en el escenario con pérdidas planteado para QoD, QoI, QoE y QC.

Para cada medición dentro de la fase de *testing*, nuestro modelo estimará un valor y lo comparará con el real. En ellas, se observan caídas de prestaciones notables a lo largo de

los datos registrados. Las caídas de rendimiento de menor duración también consigue detectarlas disminuyendo temporalmente el valor predicho.

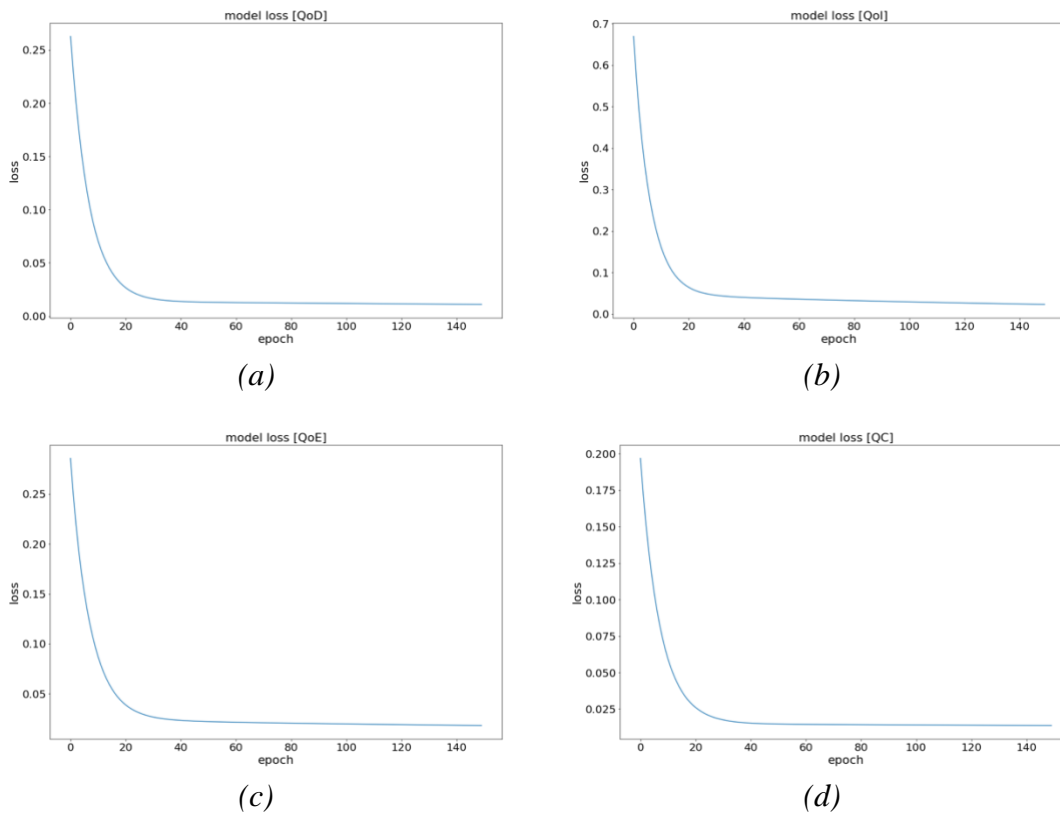


Fig. 44: Función de pérdidas del modelo en un escenario real ideal; (a) QoD, (b) QoI, (c) QoE y (d) QC.

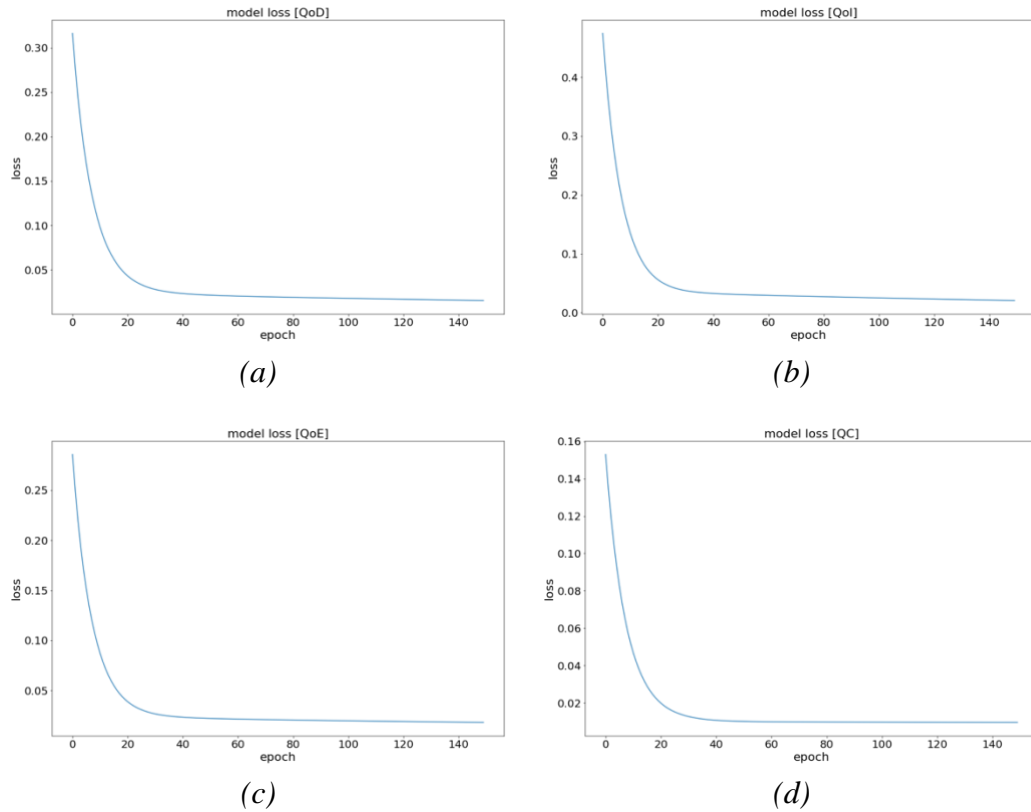


Fig. 45: Función de pérdidas del modelo en un escenario real con pérdidas; (a) QoD, (b) QoI, (c) QoE y (d) QC.

Respecto a la notificación por caída de prestaciones, el sistema compara cuando notificaría nuestro sistema basándose en los datos resultantes de la predicción respecto a cuando notificaría el sistema basándose en los datos reales de *testing*.

Sobre un escenario ideal, Fig. 46 y Fig. 47 representan los resultados del modelo de estimación de la componente QoD y los resultados del sistema de notificación, respectivamente, en un entorno urbano. Fig. 48 y Fig. 49 muestran los resultados en el mismo escenario sobre un entorno suburbano y Fig. 50 y Fig. 51 lo hacen para un entorno rural. Fig. 52 muestra los resultados del modelo de estimación para la componente QoI, y

Fig. 53 lo hace para el sistema de notificación sobre un entorno urbano, mientras que Fig. 54 y Fig. 55 lo hacen sobre un entorno suburbano y Fig. 56 y Fig. 57 hace lo propio en un entorno rural. Sobre la componente QoE también se representan los resultados del modelo de estimación y del sistema de notificación. Fig. 58 y Fig. 59 presentan resultados sobre un entorno urbano; Fig. 60 y Fig. 61 sobre un entorno suburbano; y Fig. 62 y Fig. 63 hace lo propio sobre un entorno rural. Finalmente se presentan los resultados para la componente QC, donde Fig. 64 y Fig. 65 muestran los resultados sobre un entorno urbano, Fig. 66 y Fig. 67 sobre un entorno suburbano, y Fig. 68 y Fig. 69 sobre un entorno rural. En todas estas gráficas, se observa que las caídas de rendimiento en muy pocos T_{evals} consecutivos son difíciles de predecir.

Por el contrario, teniendo en cuenta un escenario en el que se contemplan pérdidas con una probabilidad del 10% y un retardo aleatorio entre 0 y 100 ms, el modelo planteado obtiene un rendimiento similar que en el escenario ideal con resultados para las métricas QoX algo menores por la naturaleza de los propios datos. Para cada componente, sobre cada entorno y escenario, se representan dos gráficas: (i) los resultados del modelo de predicción del sistema y (ii) los resultados del sistema de notificación. En lo que respecta a QoD sobre un entorno urbano, los resultados se recogen en Fig. 70 y Fig. 71. Fig. 72 y Fig. 73 muestran los resultados para un entorno suburbano, y Fig. 74 y Fig. 75 hacen lo propio para un entorno rural. Para evaluar los resultados de QoI, Fig. 76 y Fig. 77 lo hacen en un entorno urbano; Fig. 78 y Fig. 79 en un entorno suburbano; y Fig. 80 y Fig. 81 en un entorno rural. En relación a QoE, Fig. 82 y Fig. 83 recogen los resultados en un entorno rural; Fig. 84 y Fig. 85 en un entorno suburbano, y Fig. 86 y Fig. 87 en un entorno rural.

Por último, Fig. 88 y Fig. 89 recogen los resultados de QC en un entorno urbano; Fig. 90 y Fig. 91 en un entorno suburbano, y Fig. 92 y Fig. 93 en un entorno rural.

Partiendo de un escenario ideal o con pérdidas nuestro modelo de regresión permite predecir con exactitud valores futuros de las diferentes métricas bajo estudio, siendo el modelo para la métrica QoI la que mejores resultados ofrece. Por el contrario, la métrica que peores resultados ofrece es QC, especialmente en un entorno con pérdidas.

Escenario ideal – QoD – Entorno Urbano.

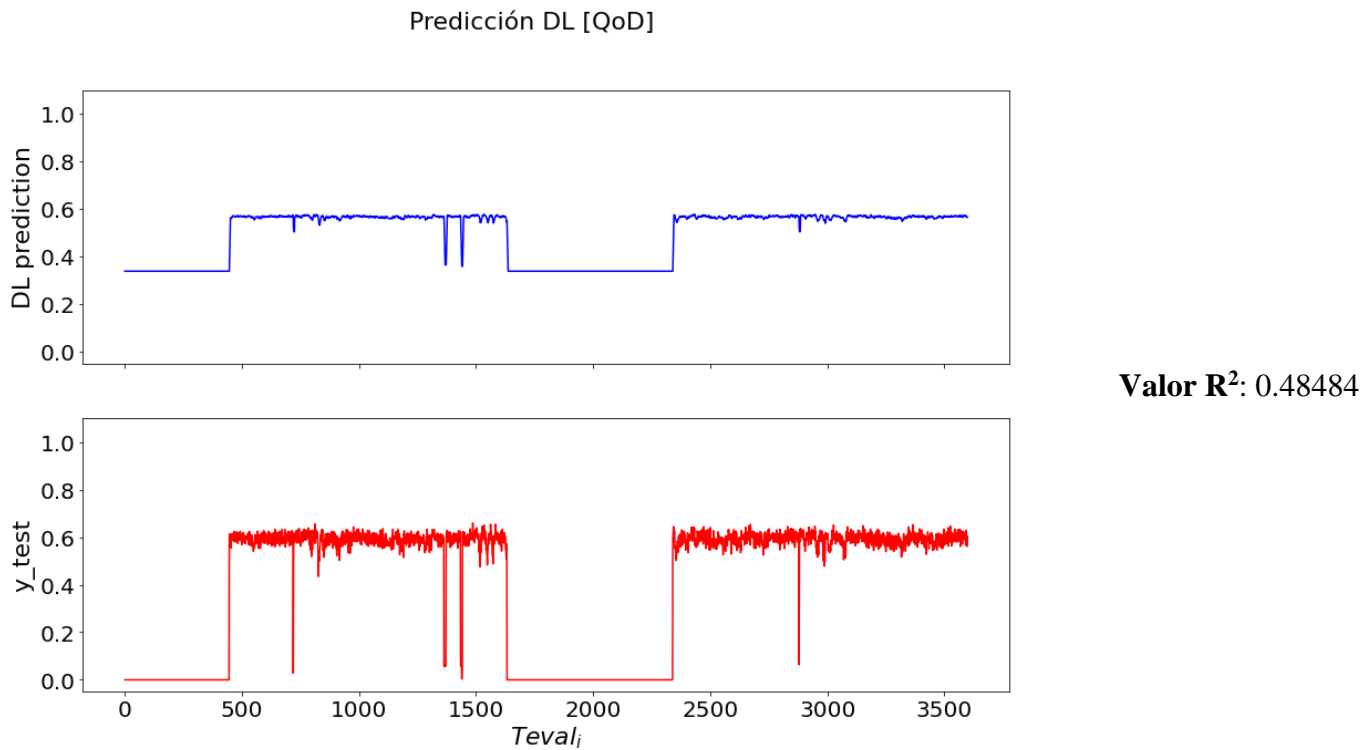


Fig. 46: Resultados de estimación QoD en entorno urbano ideal.

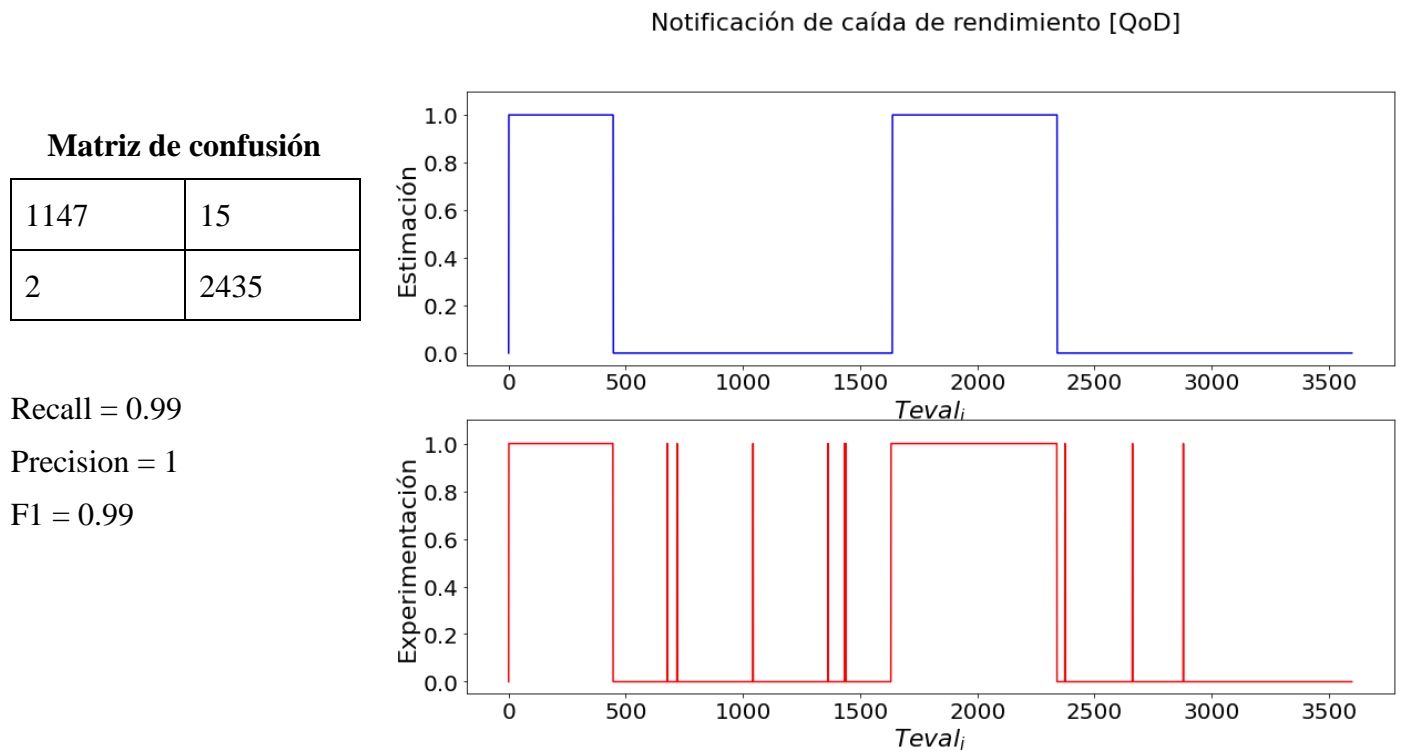
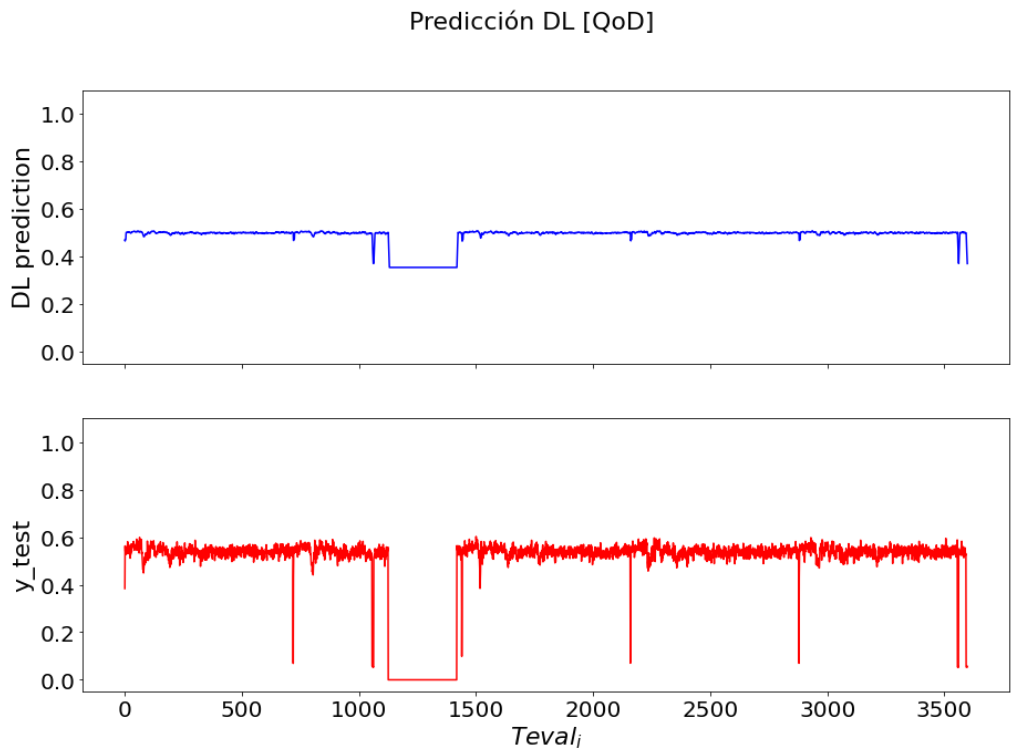


Fig. 47: Resultados del sistema de notificación QoD en entorno urbano ideal.

Escenario ideal – QoD – Entorno Suburbano.



Valor R^2 : 0.5253

Fig. 48: Resultados de estimación QoD en entorno suburbano ideal.

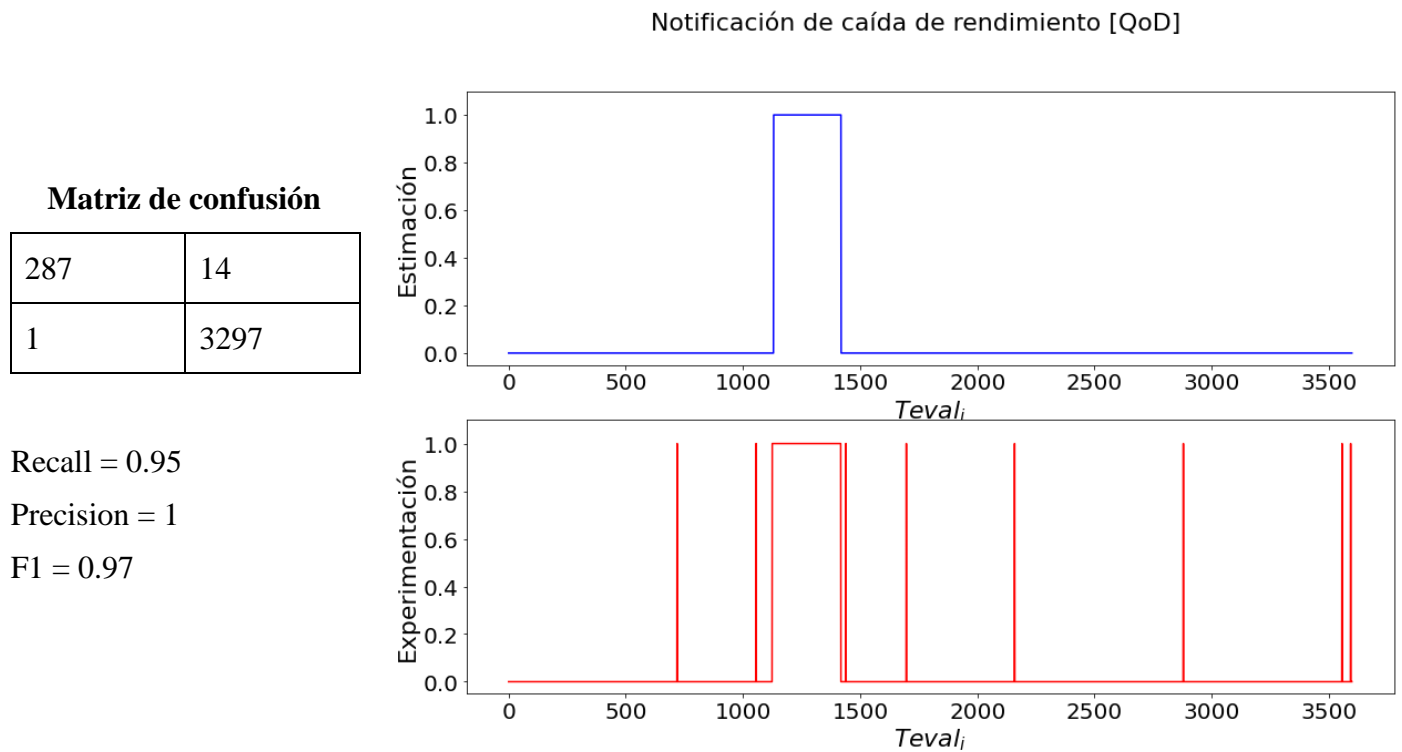


Fig. 49: Resultados del sistema de notificación QoD en entorno suburbano ideal.

Escenario ideal – QoD – Entorno Rural.

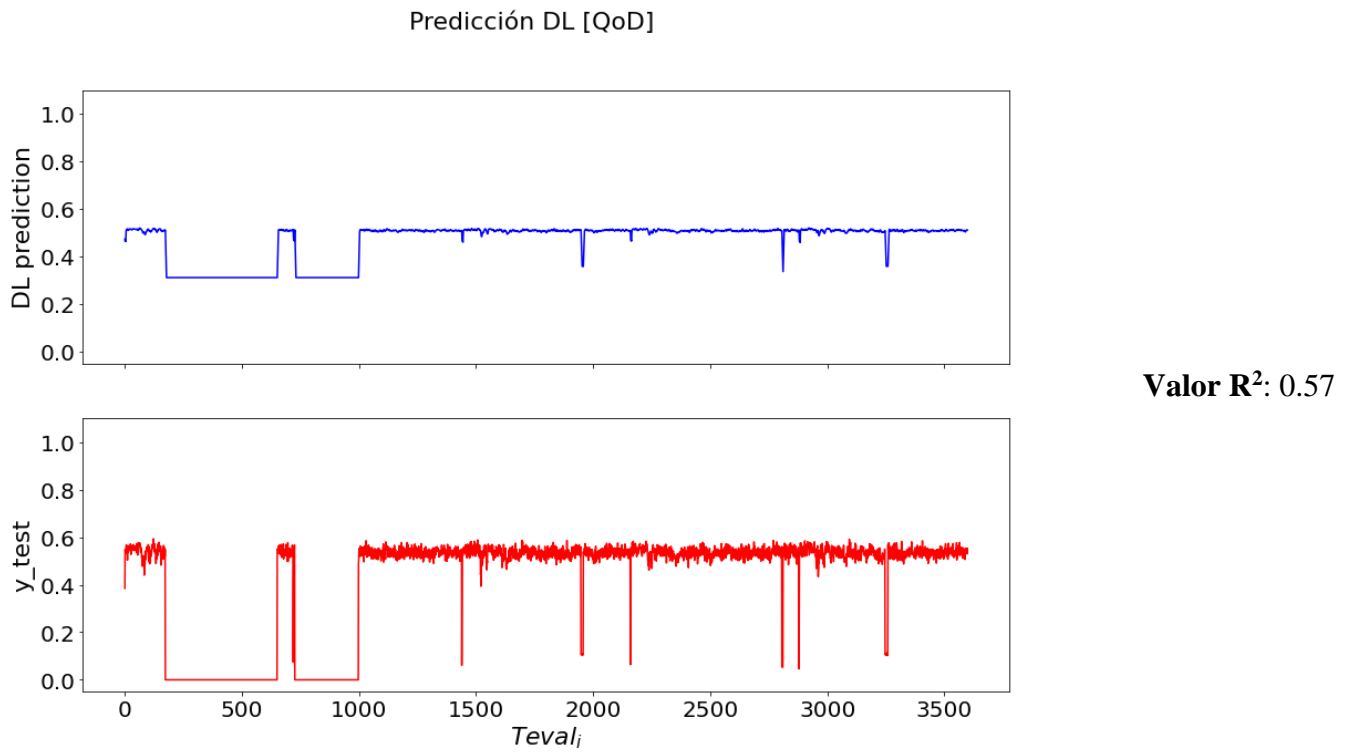


Fig. 50: Resultados de estimación QoD en entorno rural ideal.

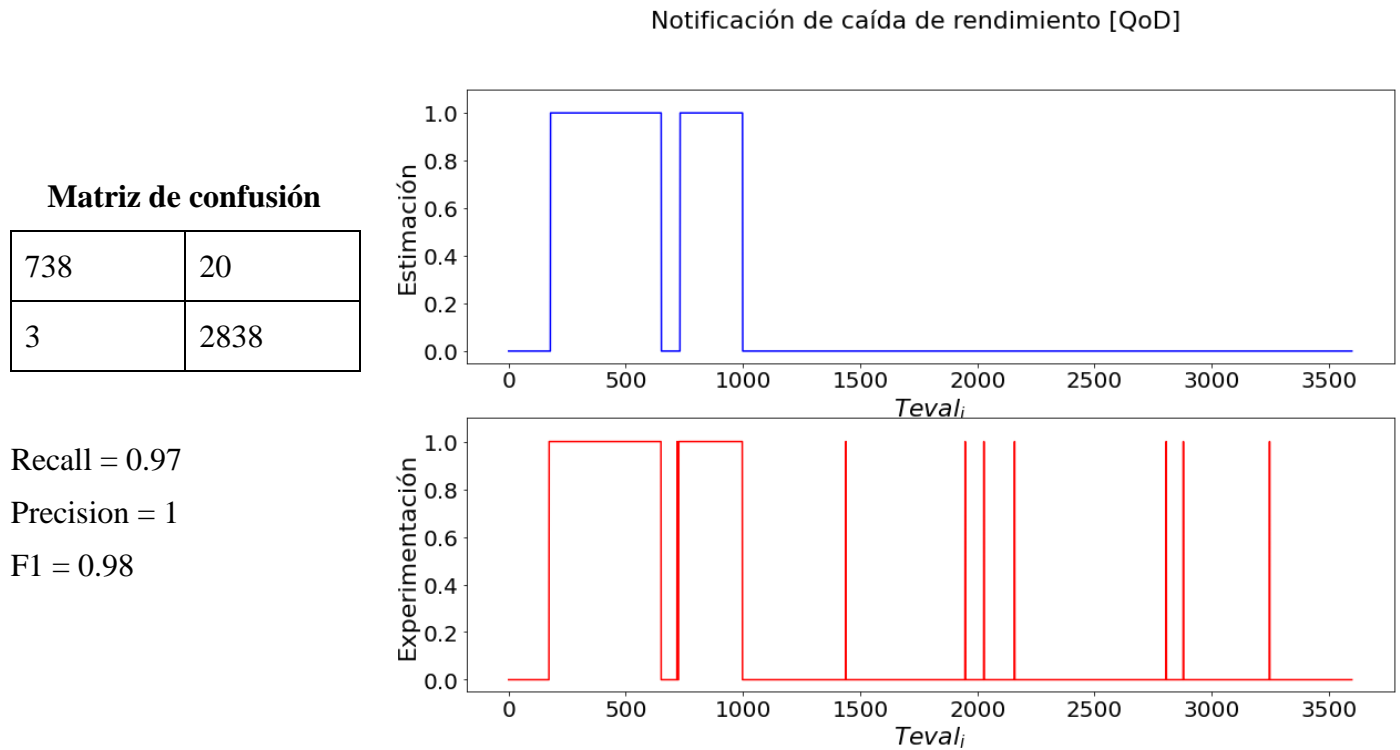


Fig. 51: Resultados del sistema de notificación QoD en entorno rural ideal.

Escenario ideal – QoI – Entorno Urbano.

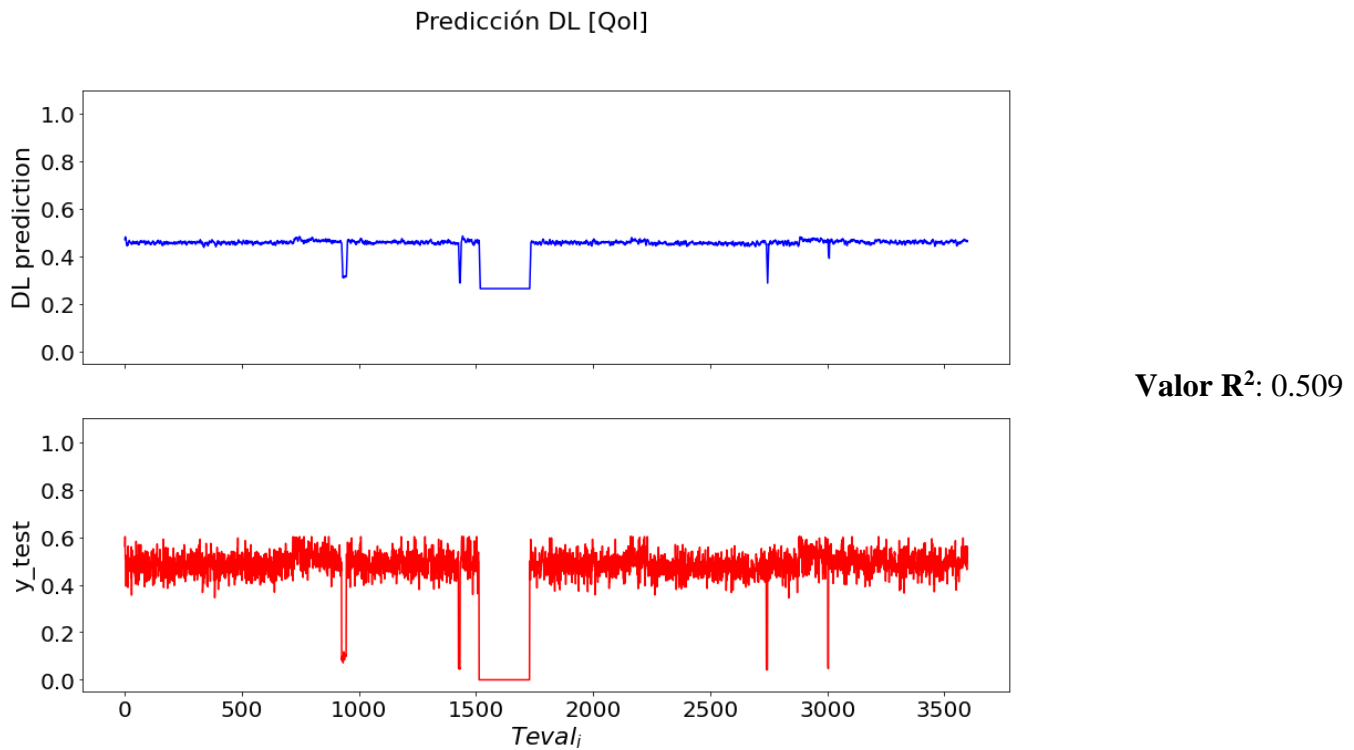


Fig. 52: Resultados de estimación QoI en entorno urbano ideal.

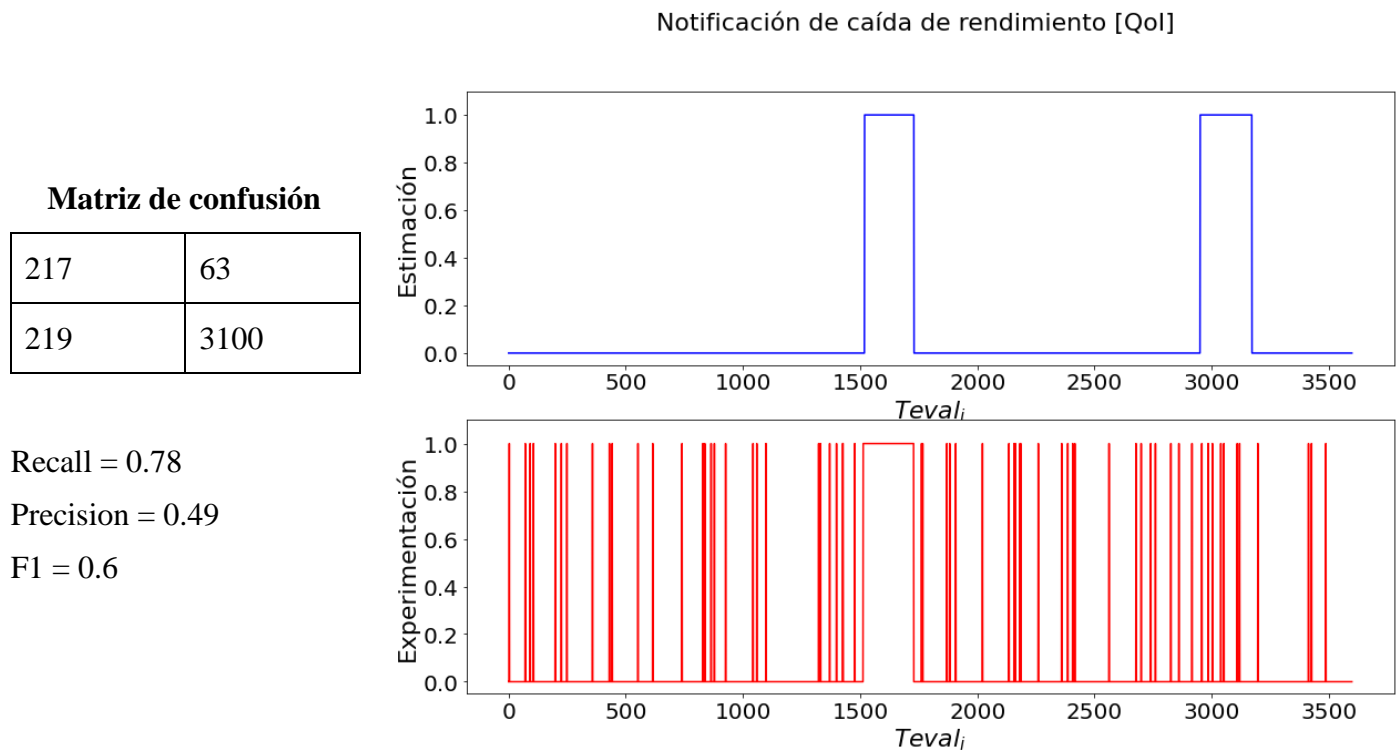


Fig. 53: Resultados del sistema de notificación QoI en entorno urbano ideal.

Escenario ideal – QoI – Entorno Suburbano.

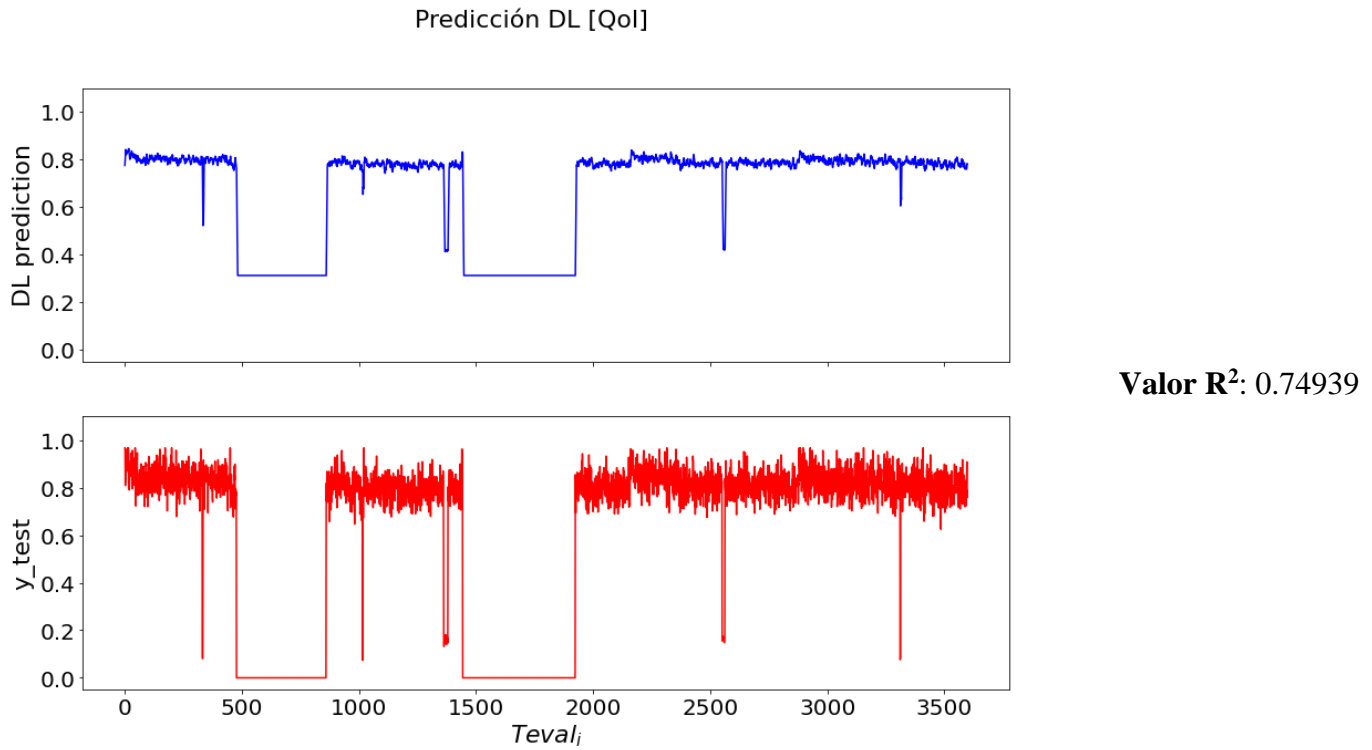


Fig. 54: Resultados de estimación QoI en entorno suburbano ideal.

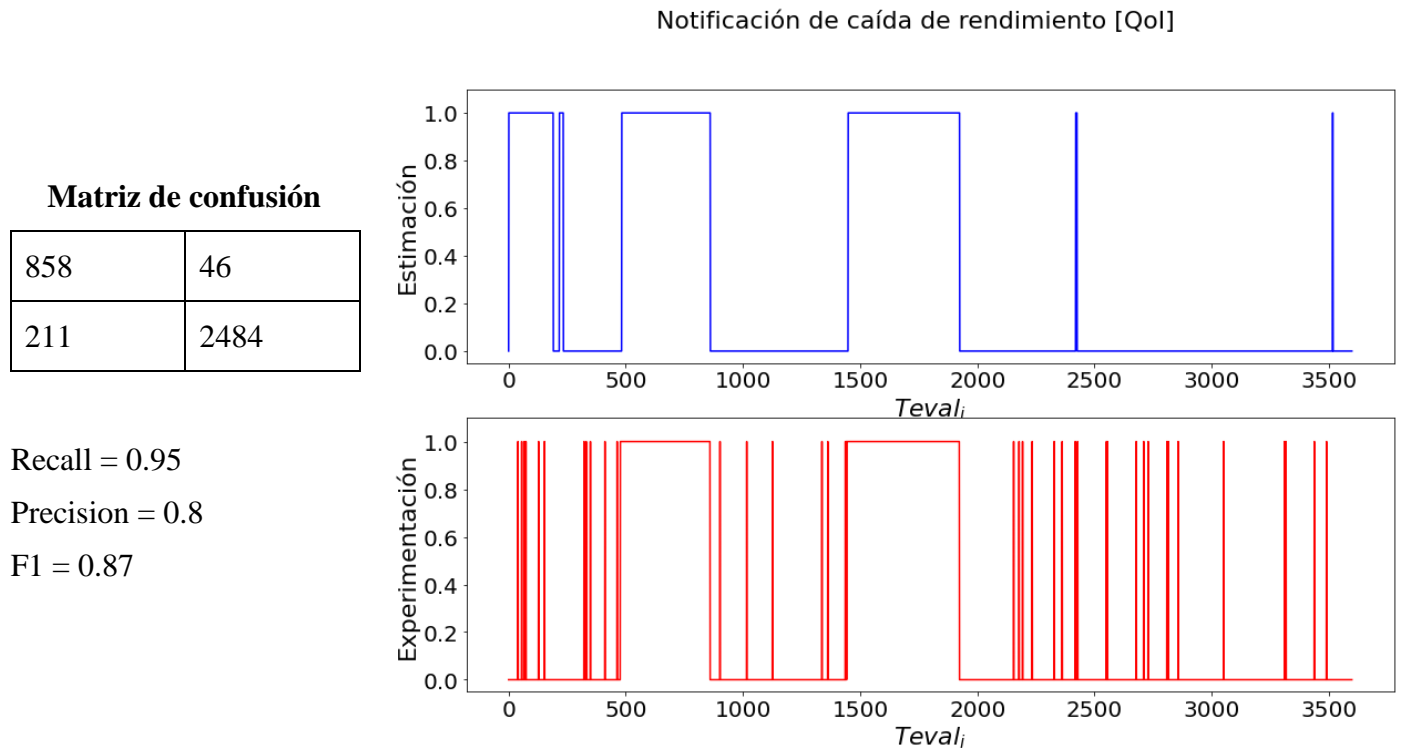


Fig. 55: Resultados del sistema de notificación QoI en entorno suburbano ideal.

Escenario ideal – QoI – Entorno Rural.

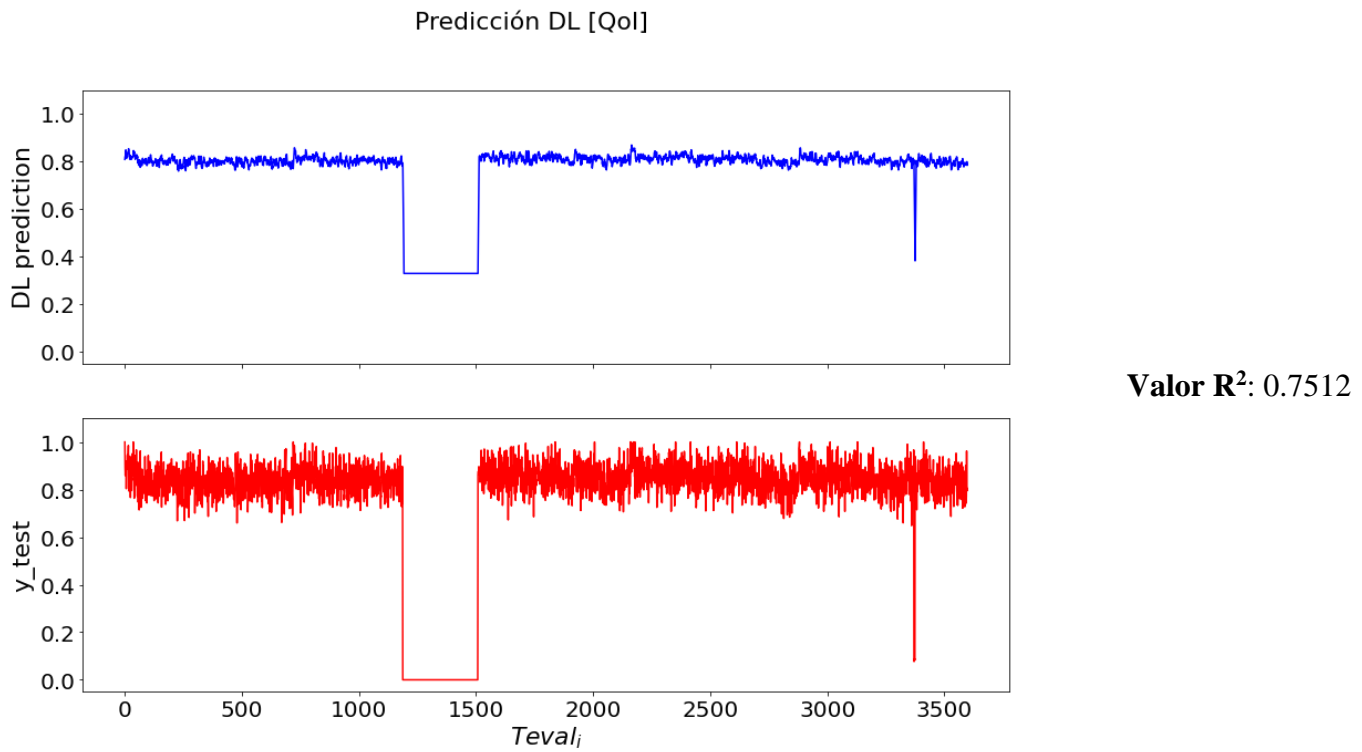


Fig. 56: Resultados de estimación QoI en entorno rural ideal.

Notificación de caída de rendimiento [QoI]

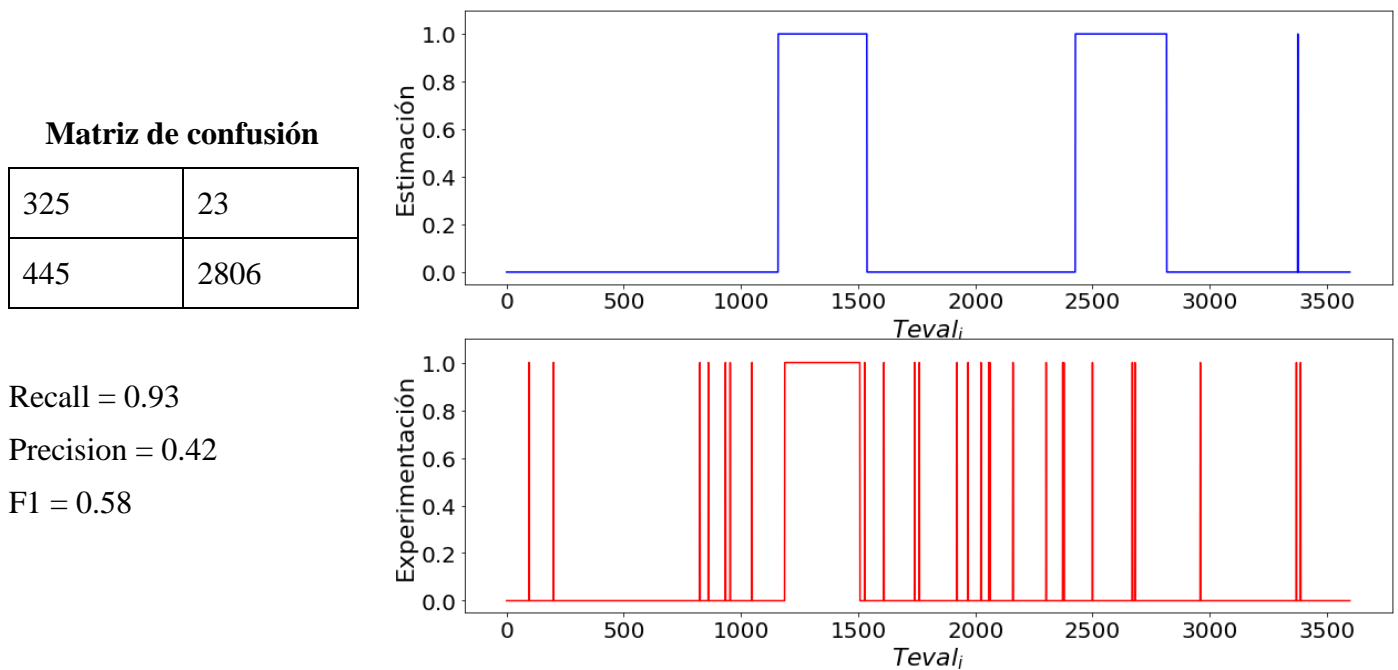


Fig. 57: Resultados del sistema de notificación QoI en entorno rural ideal.

Escenario ideal – QoE – Entorno Urbano.

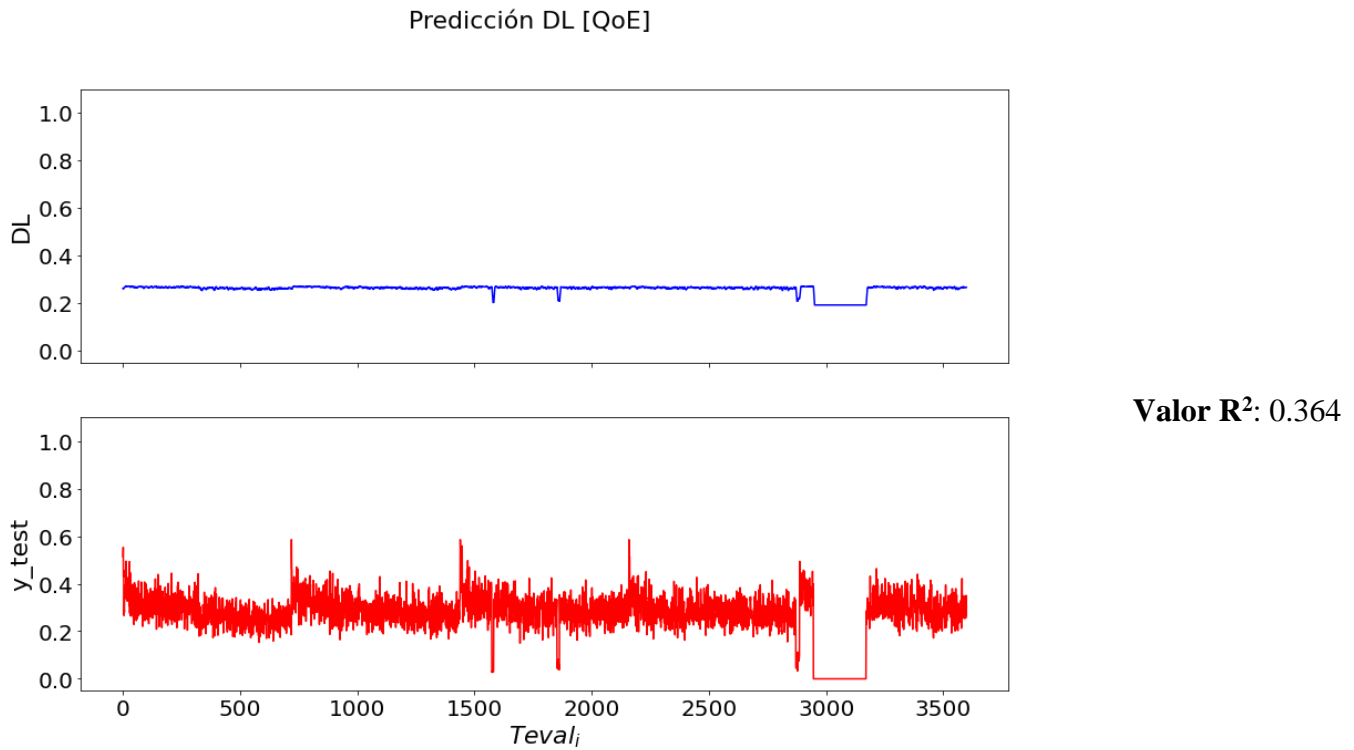


Fig. 58: Resultados de estimación QoE en entorno urbano ideal.

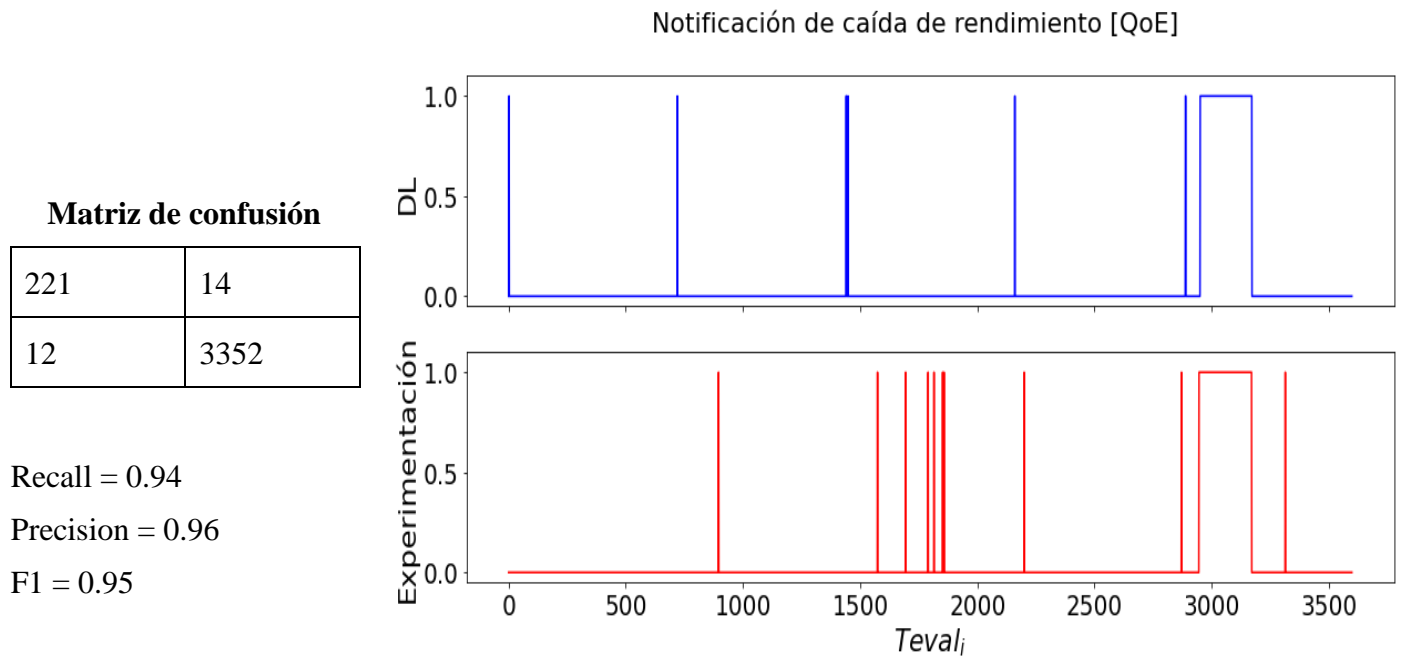


Fig. 59: Resultados del sistema de notificación QoE en entorno urbano ideal.

Escenario ideal – QoE – Entorno Suburbano.

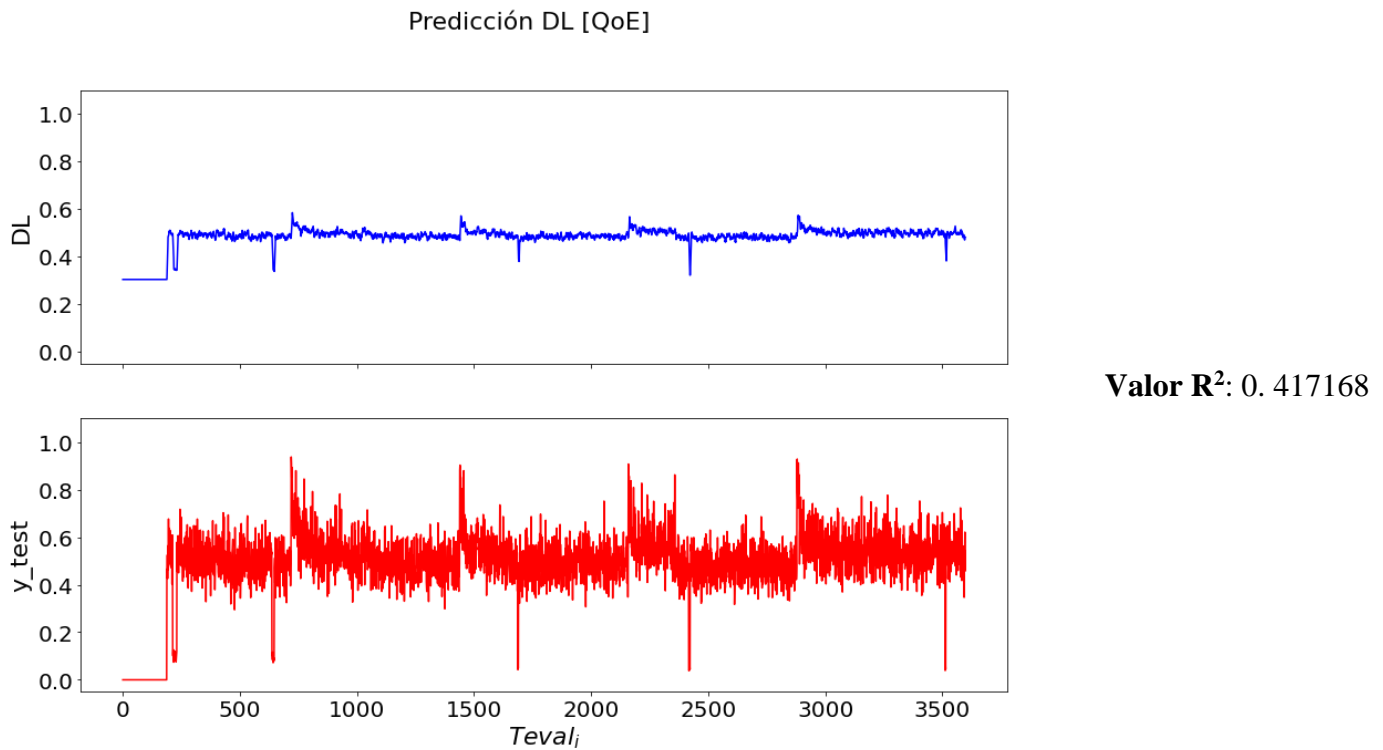


Fig. 60: Resultados de estimación QoE en entorno suburbano ideal.

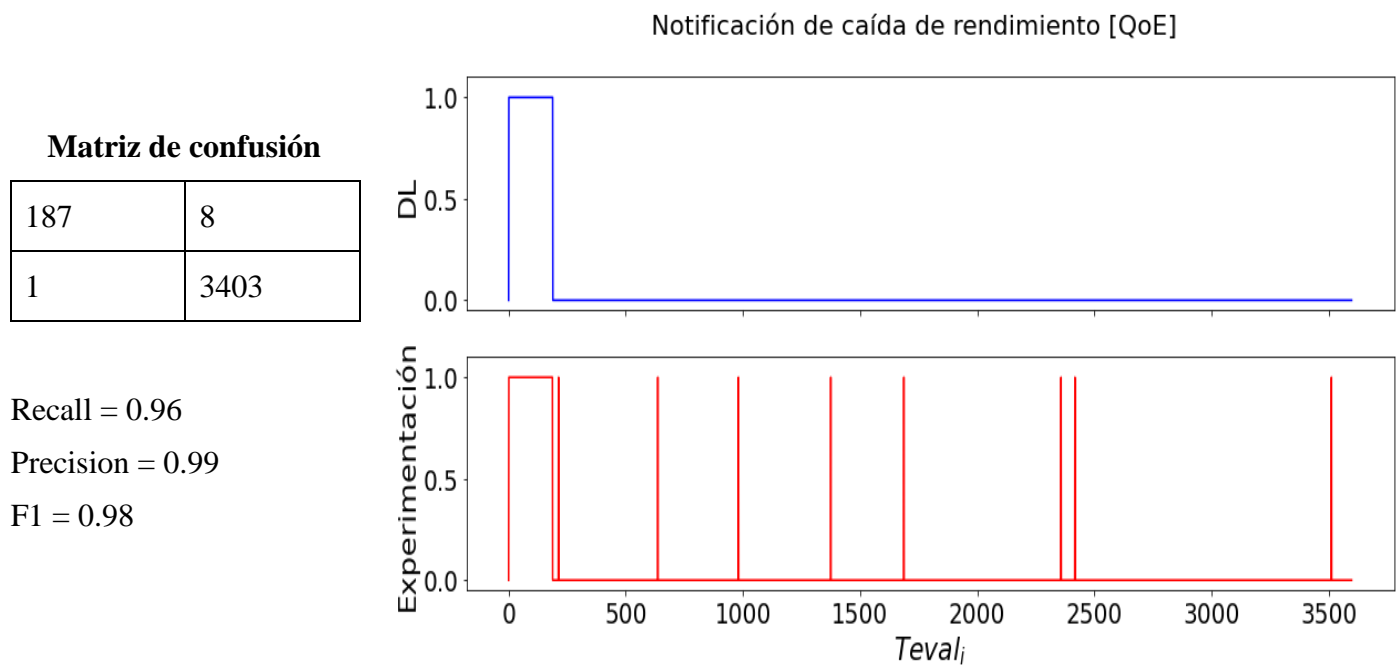


Fig. 61: Resultados del sistema de notificación QoE en entorno suburbano ideal.

Escenario ideal – QoE – Entorno Rural.

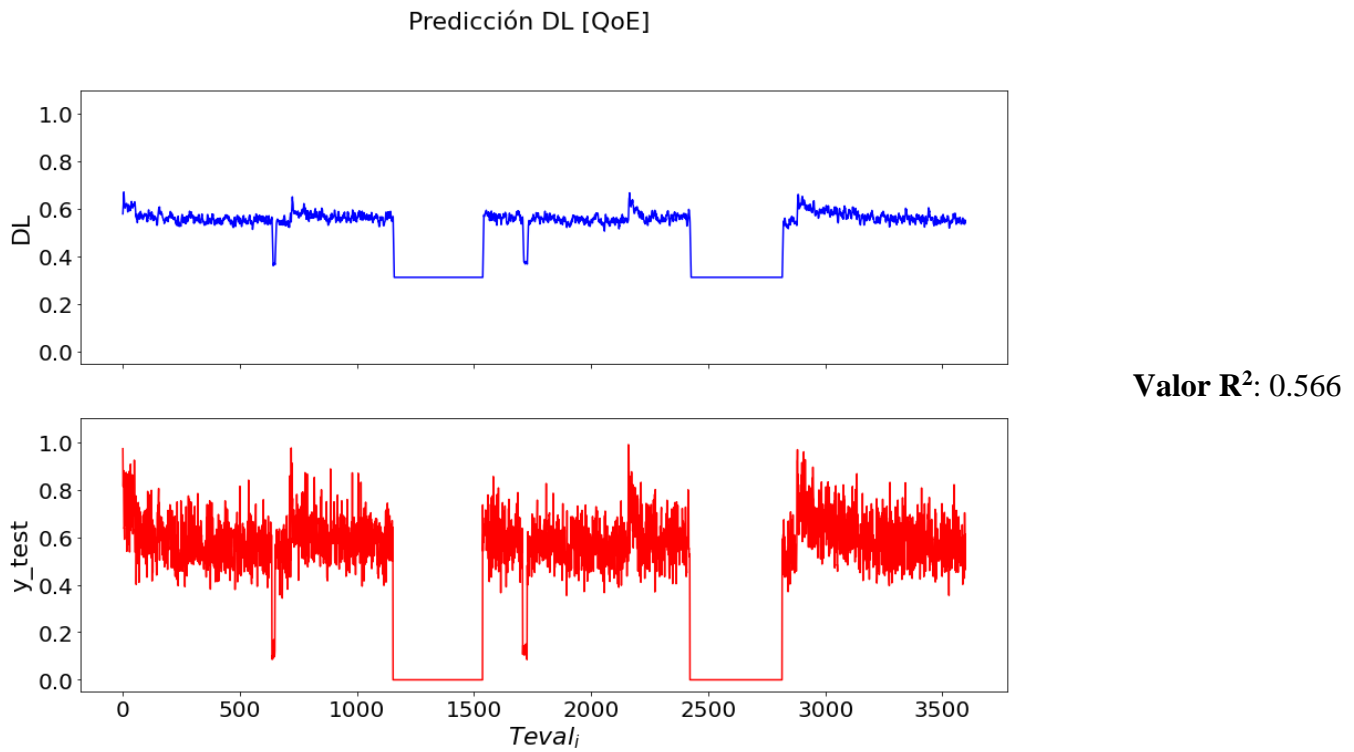


Fig. 62: Resultados de estimación QoE en entorno rural ideal.

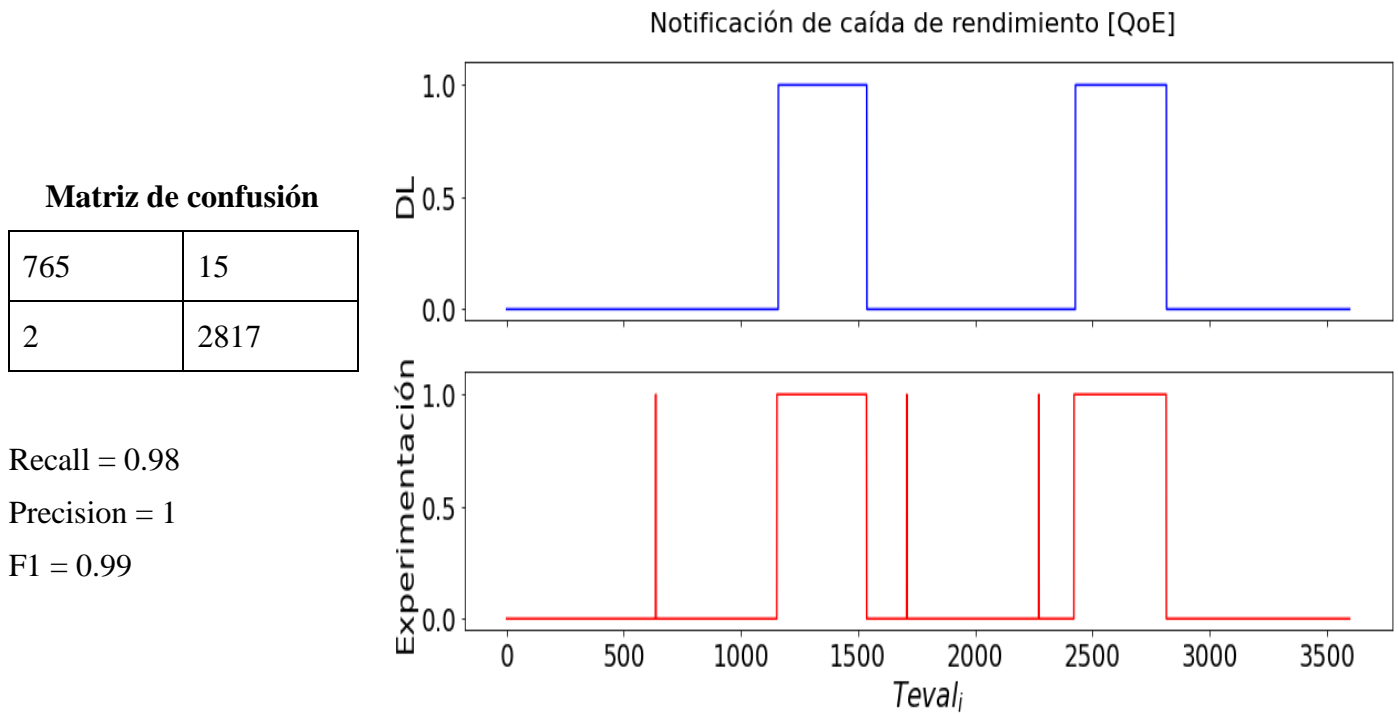
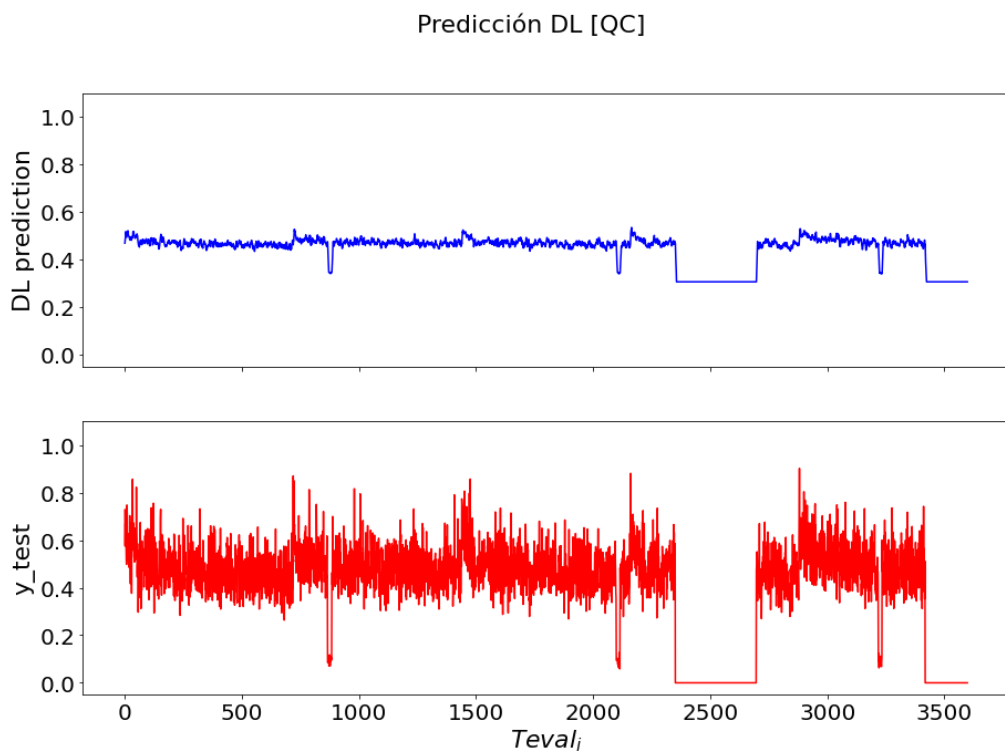


Fig. 63: Resultados del sistema de notificación QoE en entorno rural ideal.

Escenario ideal – QC – Entorno Urbano.



Valor R^2 : 0.3809

Fig. 64: Resultados de estimación QC en entorno urbano ideal.

Matriz de confusión

515	21
1	3062

Recall = 0.96
 Precision = 1
 F1 = 0.98

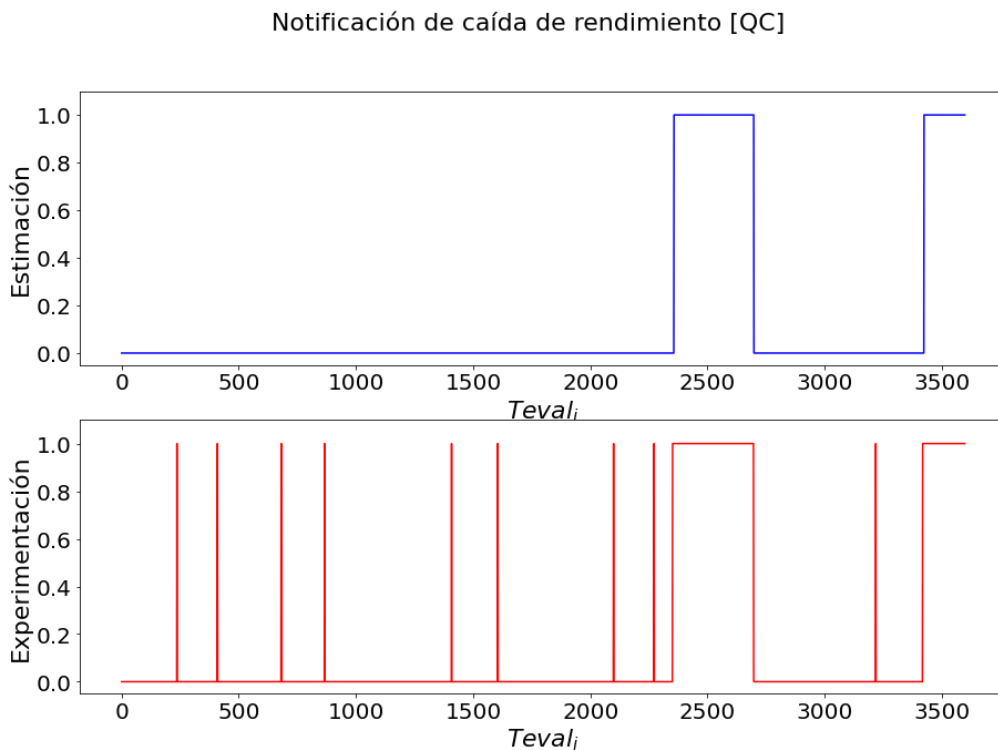


Fig. 65: Resultados del sistema de notificación QC en entorno urbano ideal.

Escenario ideal – QC – Entorno Suburbano.

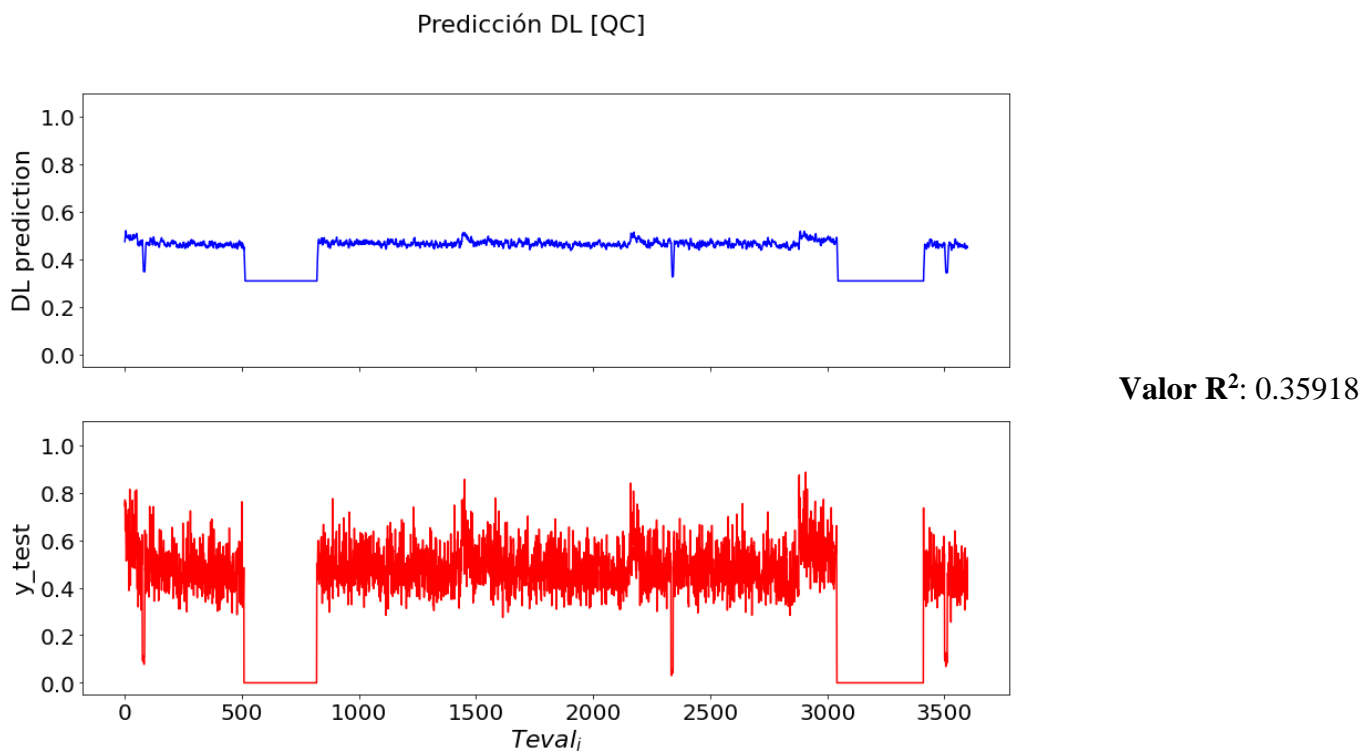


Fig. 66: Resultados de estimación QC en entorno suburbano ideal.

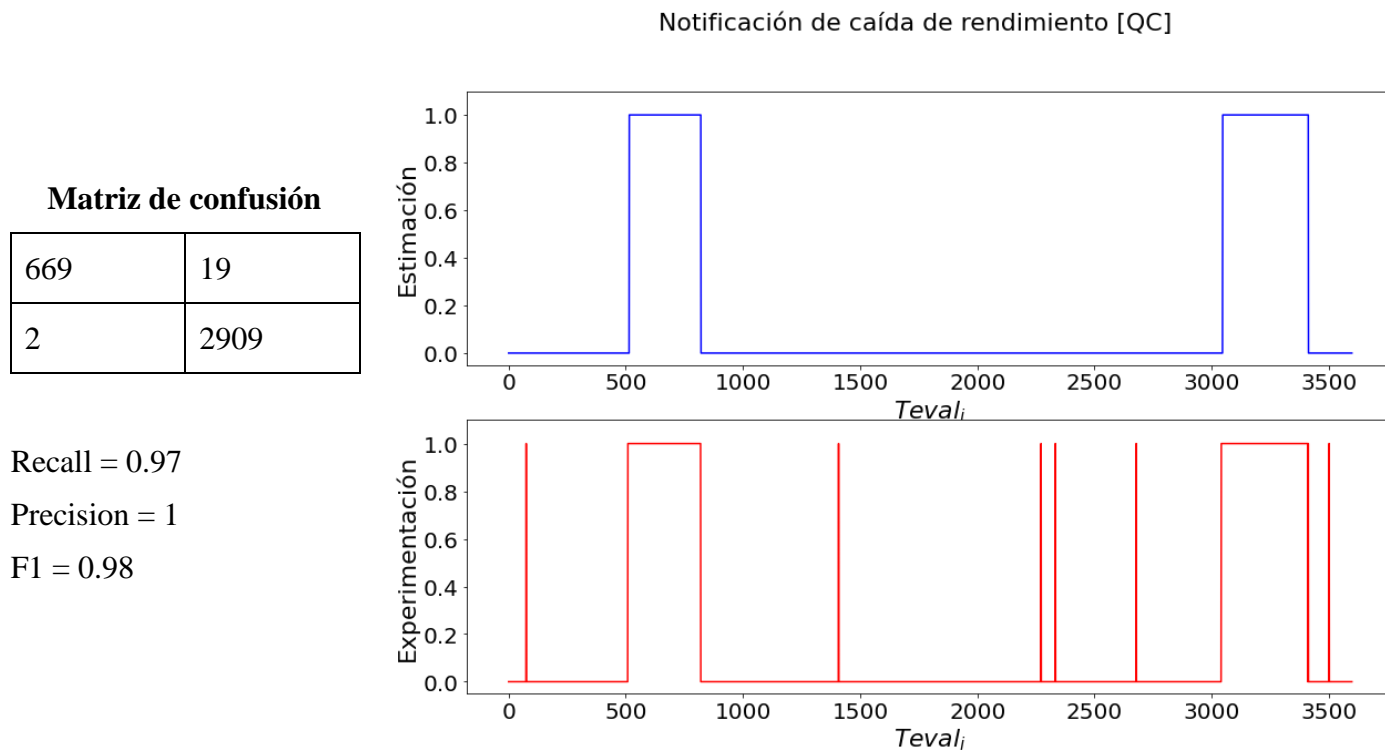
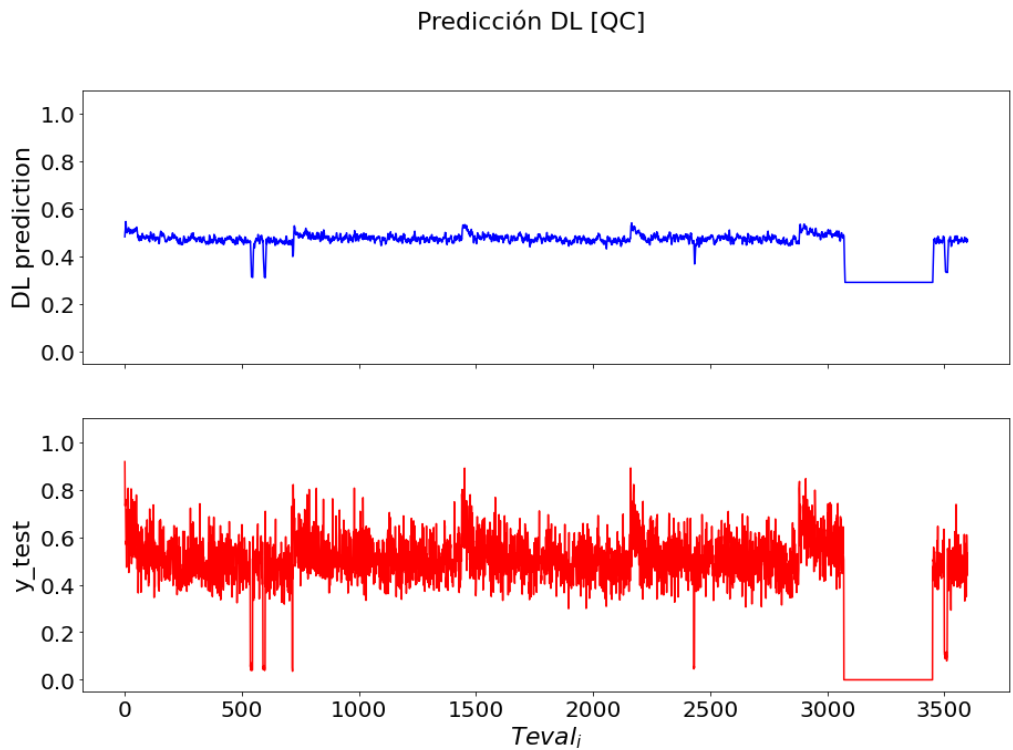


Fig. 67: Resultados del sistema de notificación QC en entorno suburbano ideal.

Escenario ideal – QC – Entorno Rural.



Valor R²: 0.47238

Fig. 68: Resultados de estimación QC en entorno rural ideal.

Matriz de confusión

373	12
1	3214

Recall = 0.97

Precision = 1

F1 = 0.98

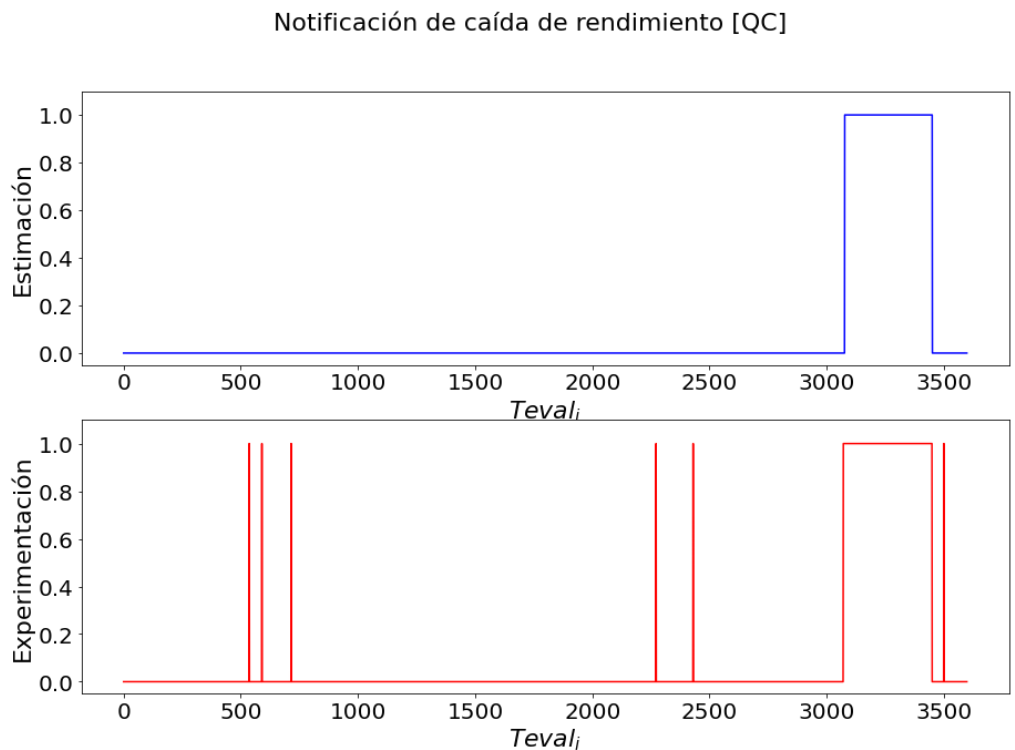


Fig. 69: Resultados del sistema de notificación QC en entorno rural ideal.

Escenario con pérdidas – QoD – Entorno Urbano.

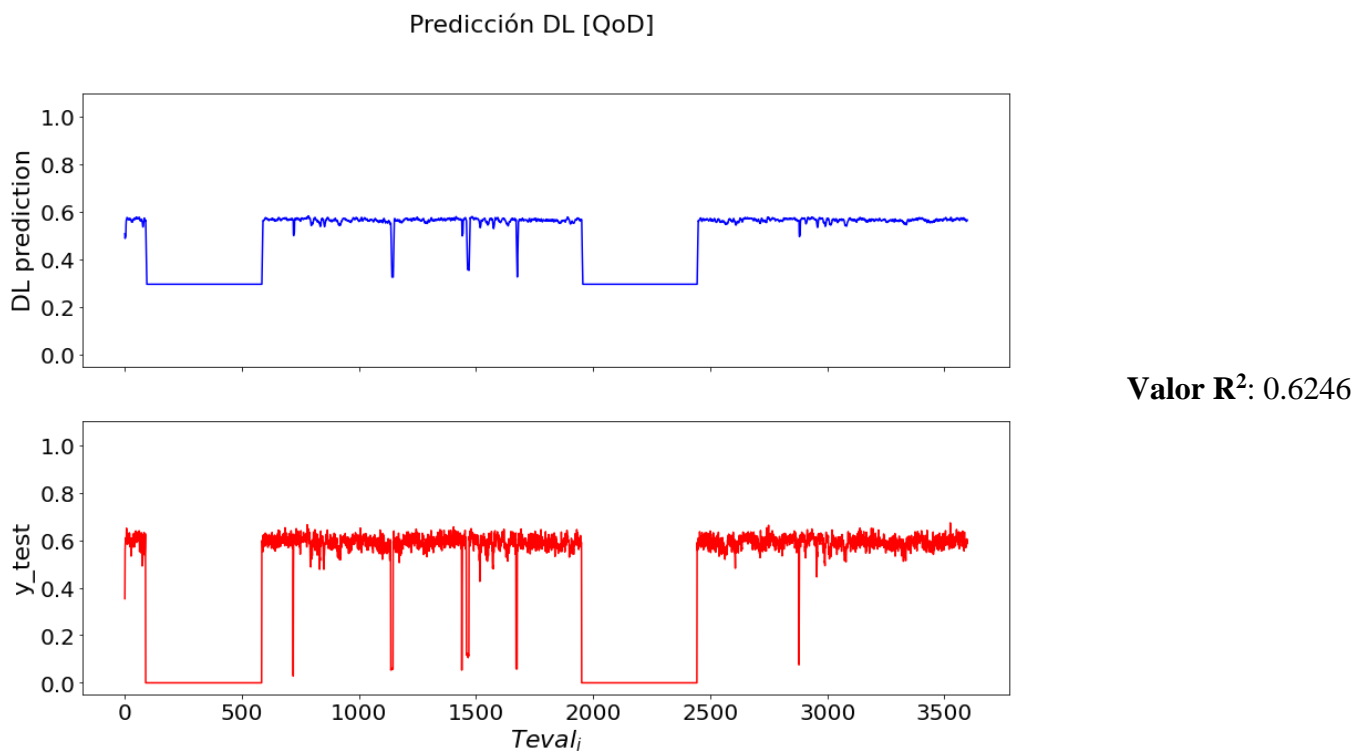


Fig. 70: Resultados de estimación QoD en entorno urbano con pérdidas.

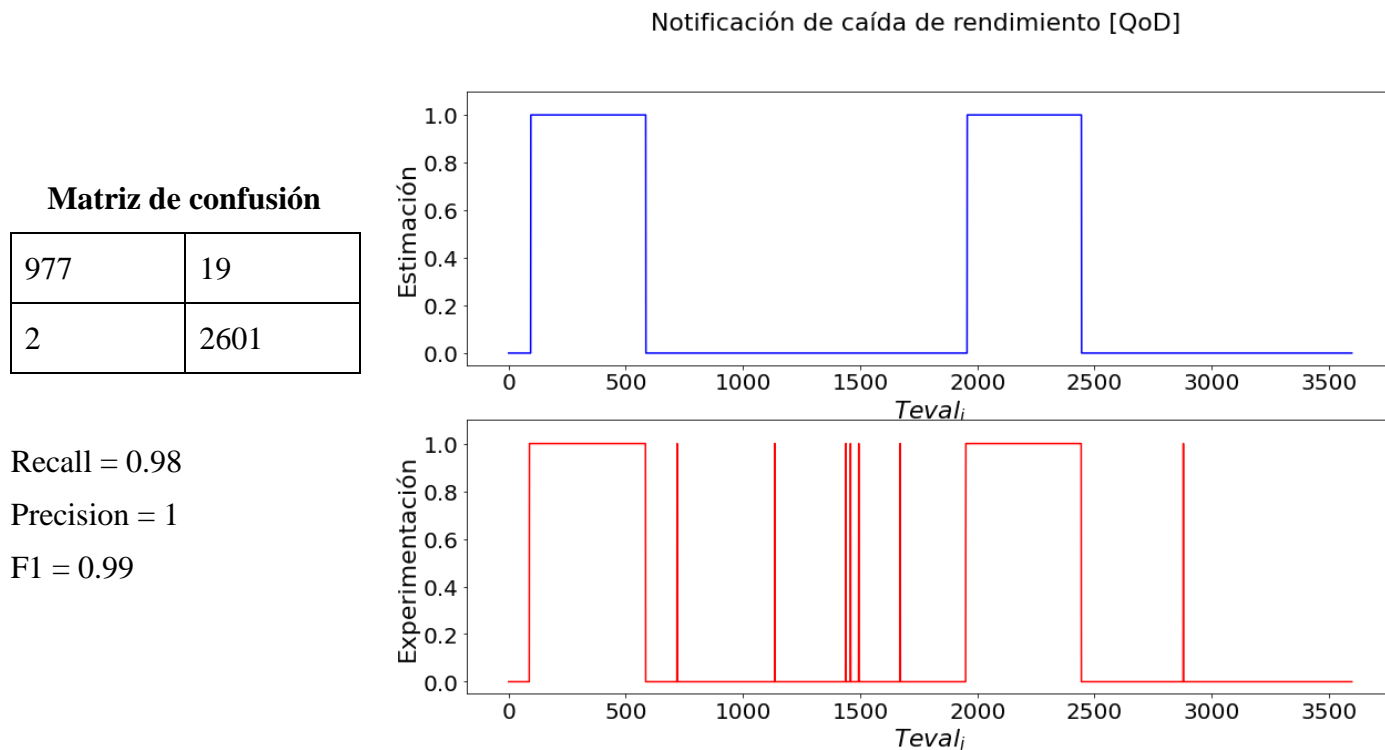
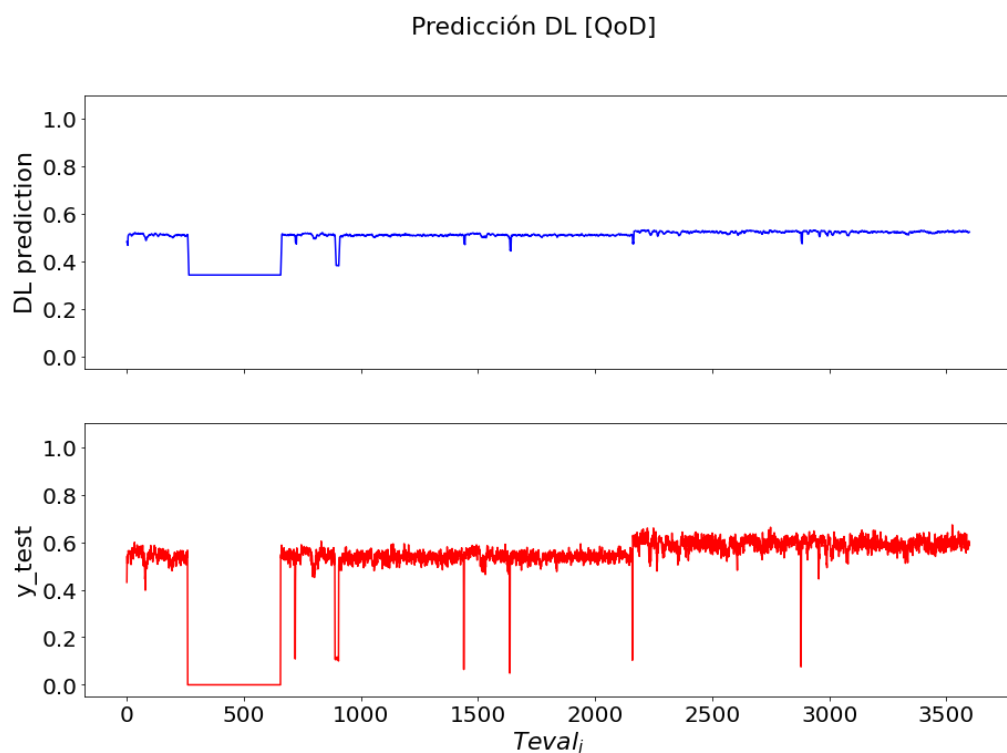


Fig. 71: Resultados del sistema de notificación QoD en entorno urbano con pérdidas.

Escenario con pérdidas – QoD – Entorno Suburbano.



Valor R²: 0.60528

Fig. 72: Resultados de estimación QoD en entorno suburbano con pérdidas.

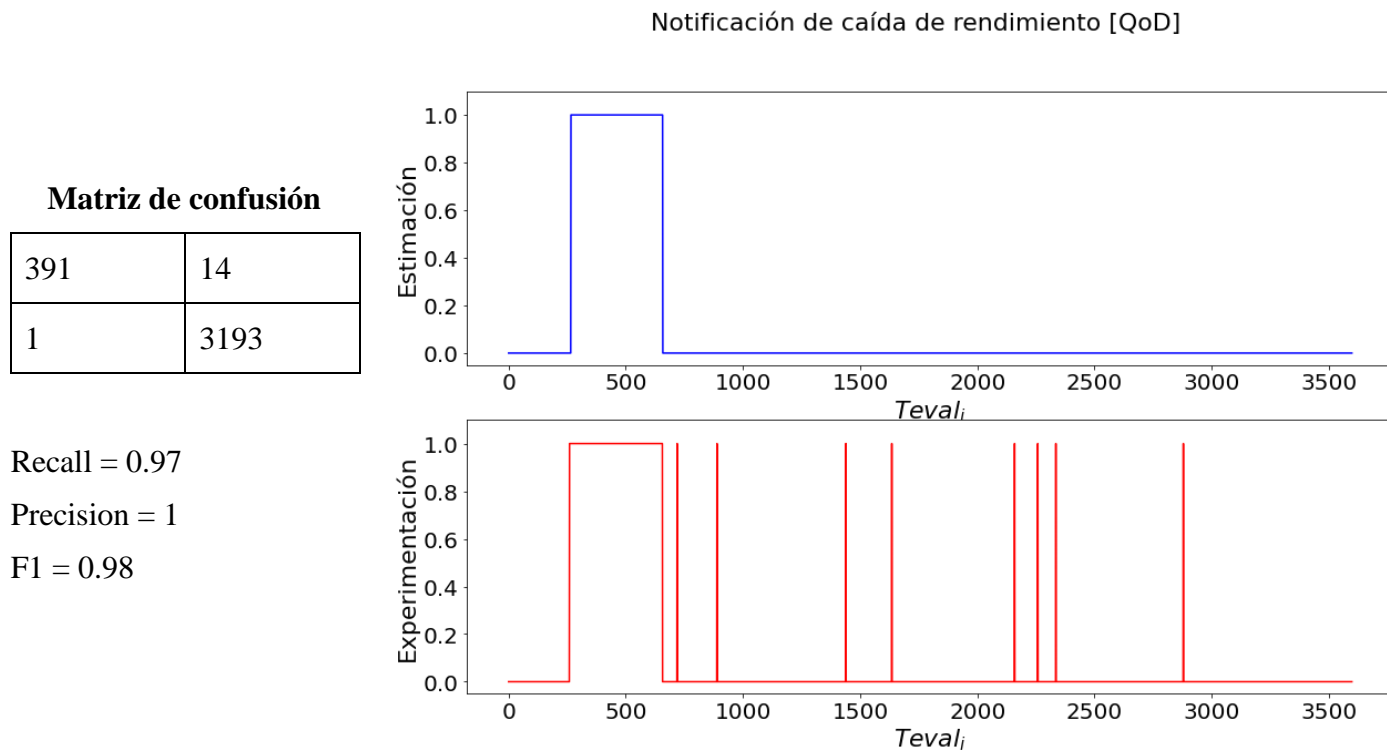


Fig. 73: Resultados del sistema de notificación QoD en entorno suburbano con pérdidas.

Escenario con pérdidas – QoD – Entorno Rural.

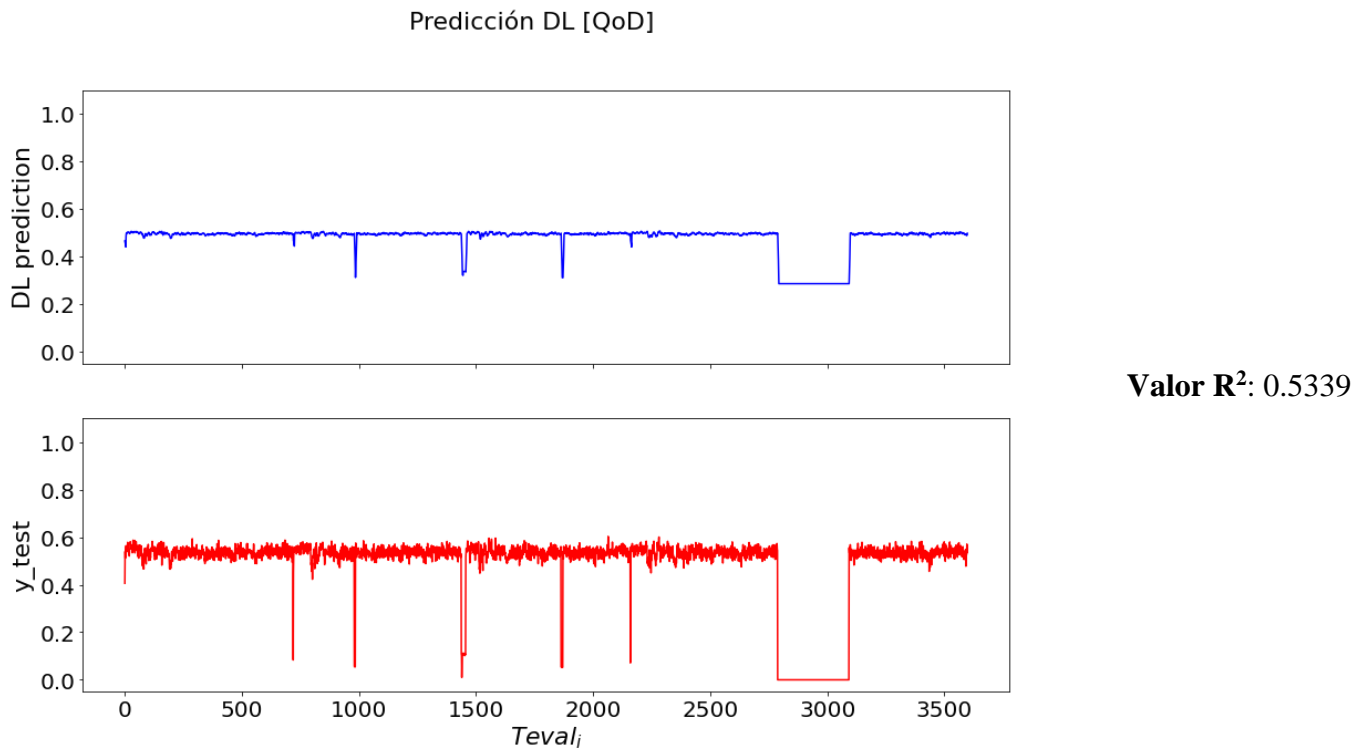


Fig. 74: Resultados de estimación QoD en entorno rural con pérdidas.

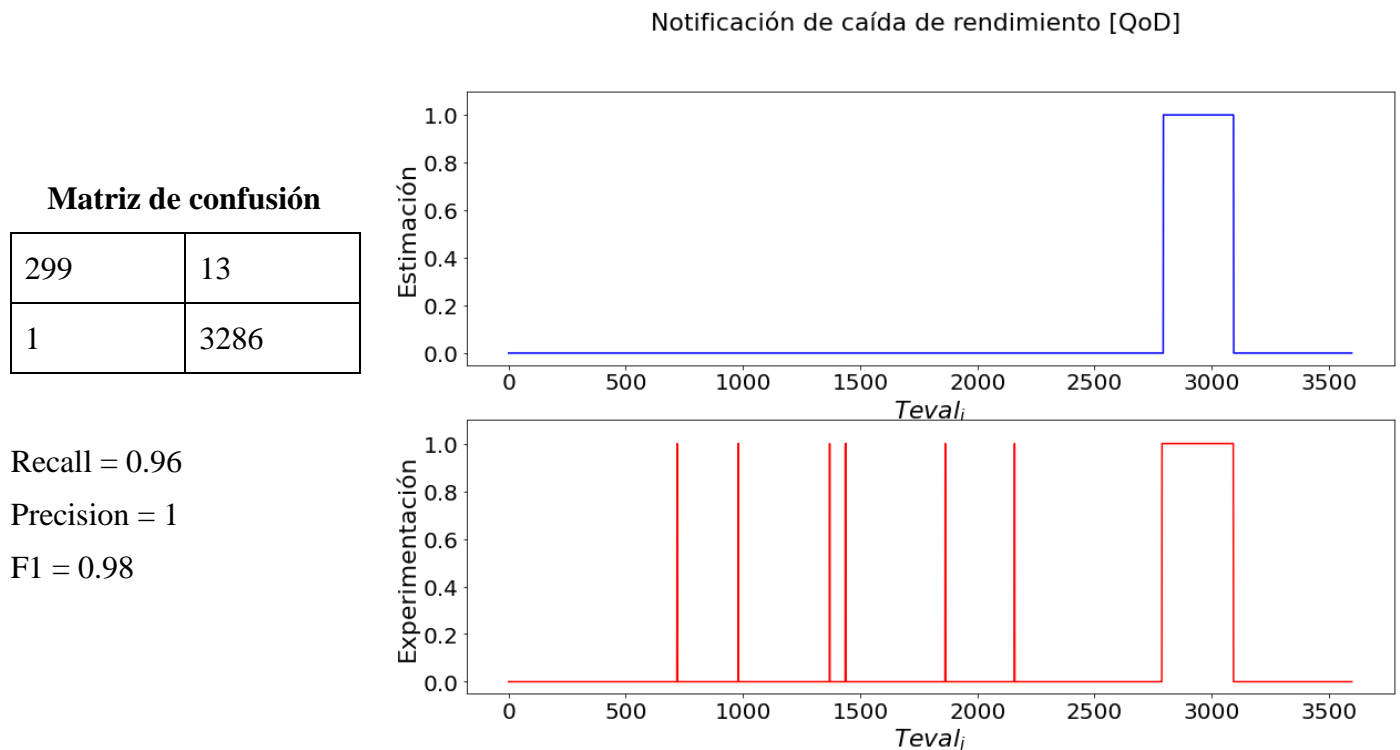
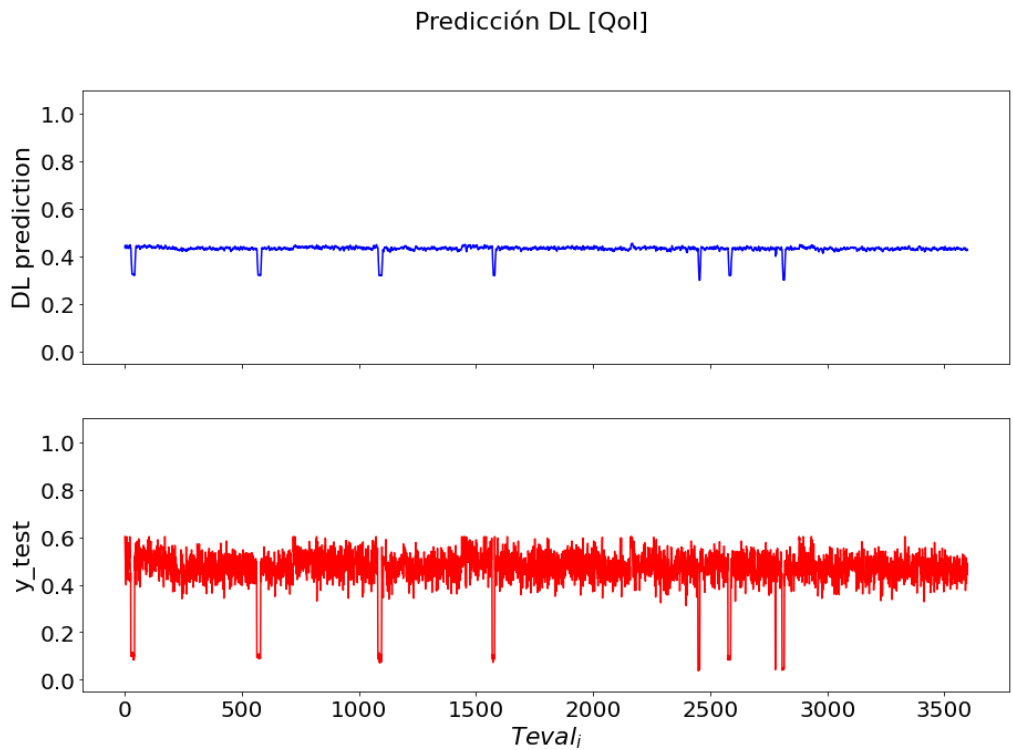


Fig. 75: Resultados del sistema de notificación QoD en entorno rural con pérdidas.

Escenario con pérdidas – QoI – Entorno Urbano.



Valor R^2 : 0.1093

Fig. 76: Resultados de estimación QoI en entorno urbano con pérdidas.

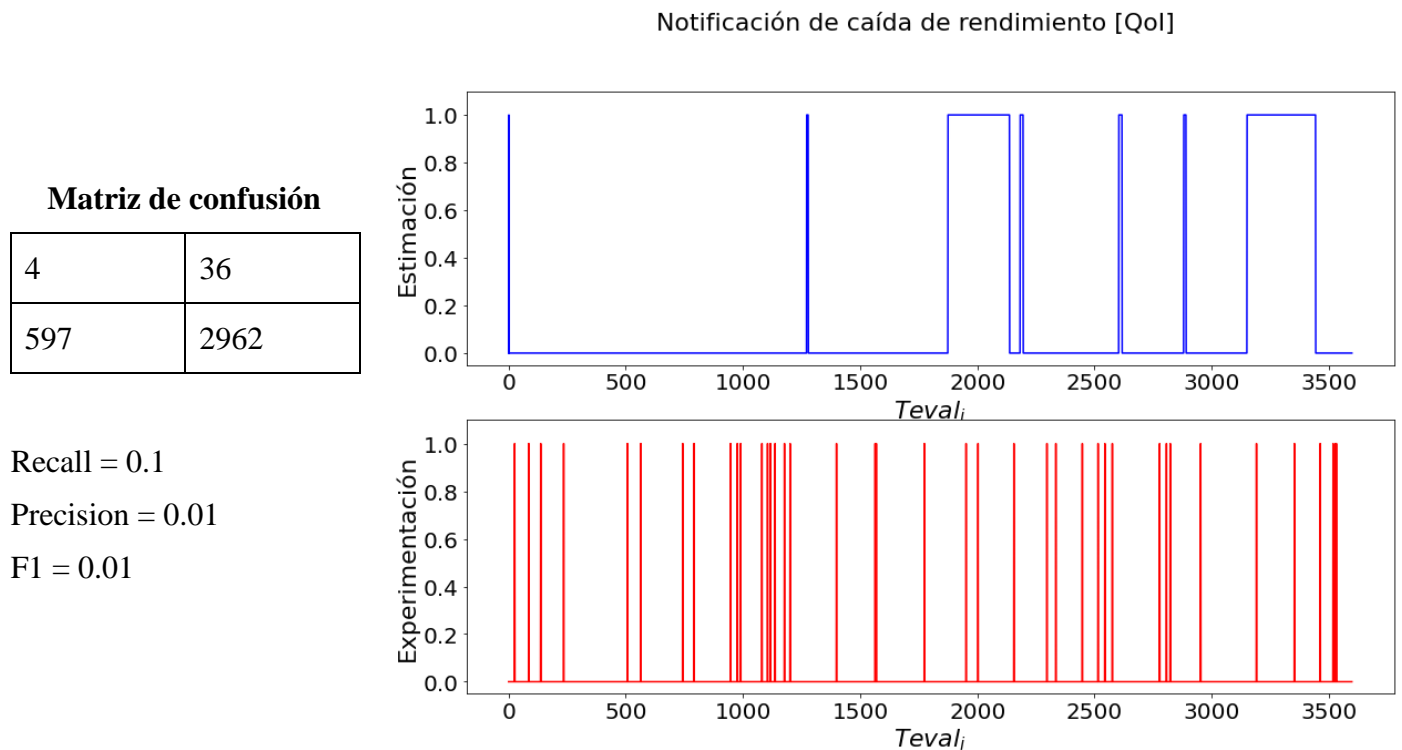


Fig. 77: Resultados del sistema de notificación QoI en entorno urbano con pérdidas.

Escenario con pérdidas – QoI – Entorno Suburbano.

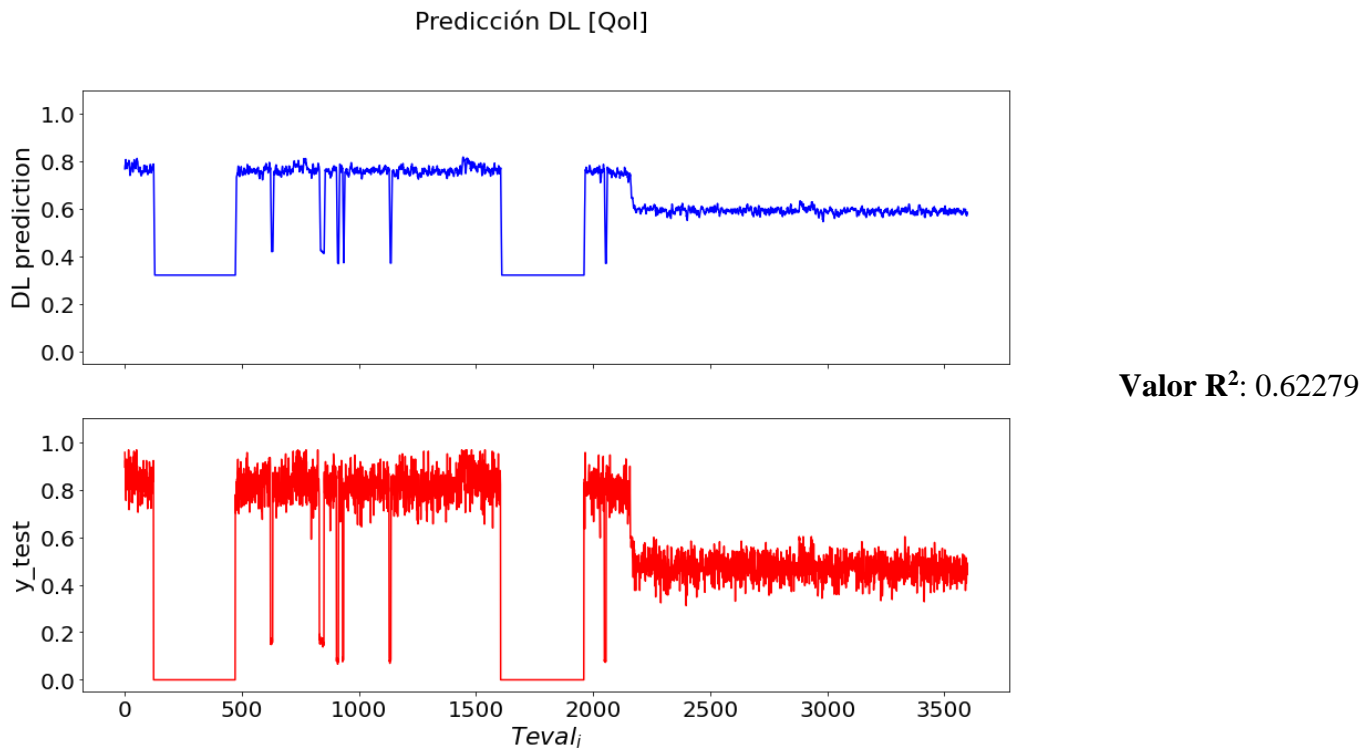


Fig. 78: Resultados de estimación QoI en entorno suburbano con pérdidas.

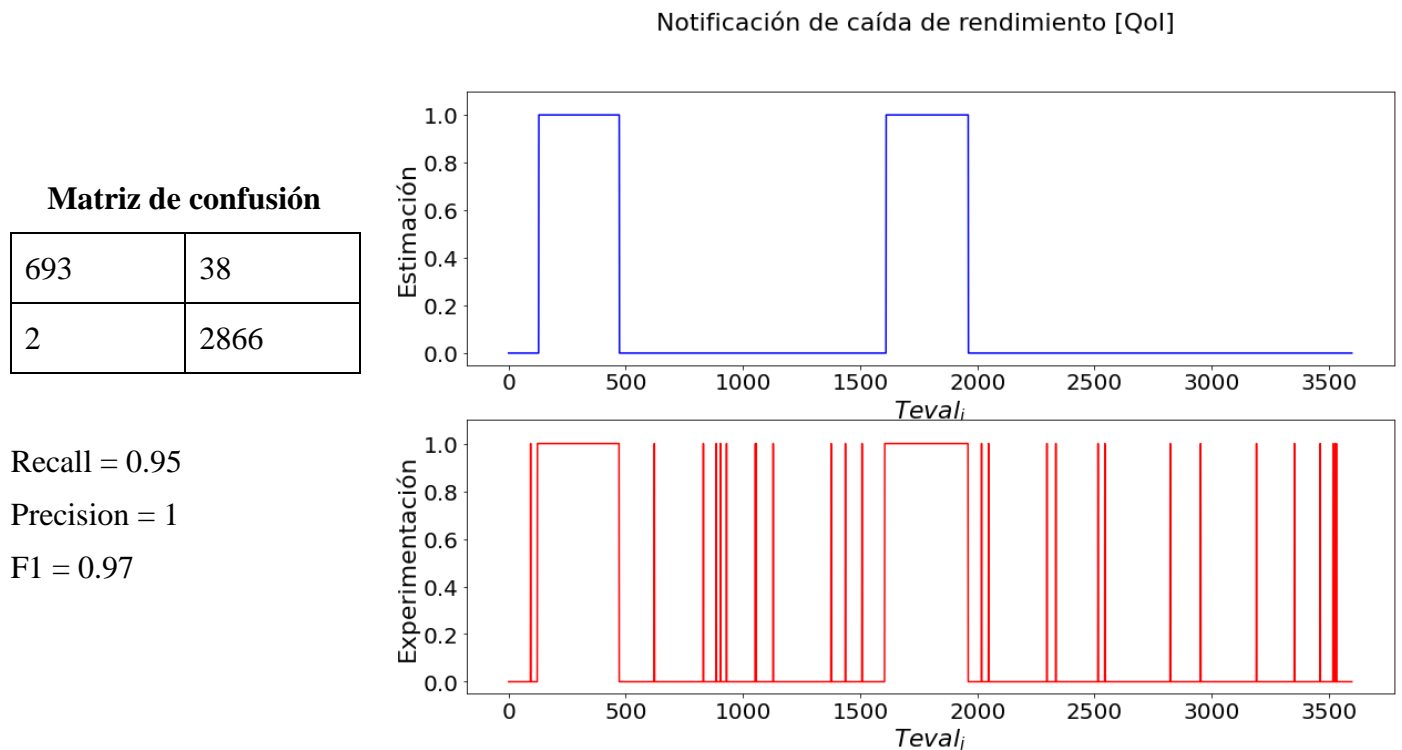


Fig. 79: Resultados del sistema de notificación QoI en entorno suburbano con pérdidas.

Escenario con pérdidas – QoI – Entorno Rural.

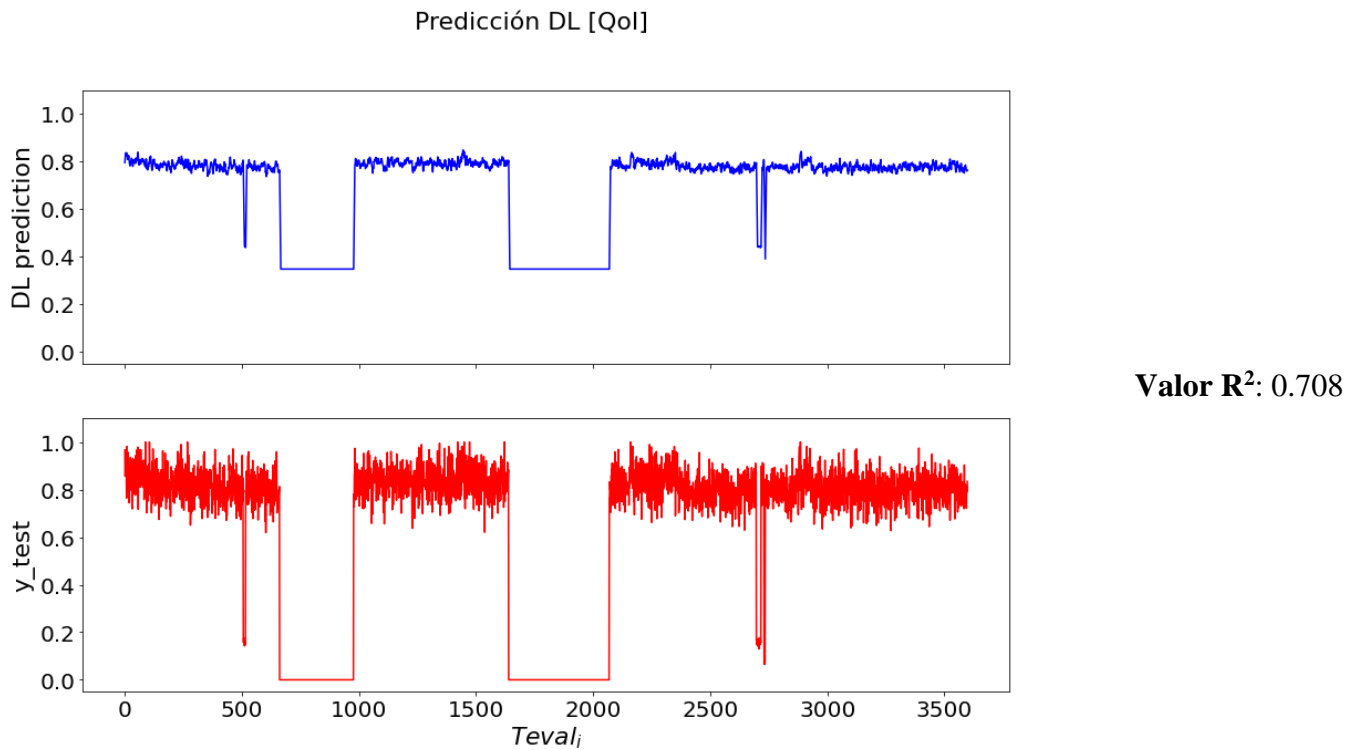


Fig. 80: Resultados de estimación QoI en entorno rural con pérdidas.

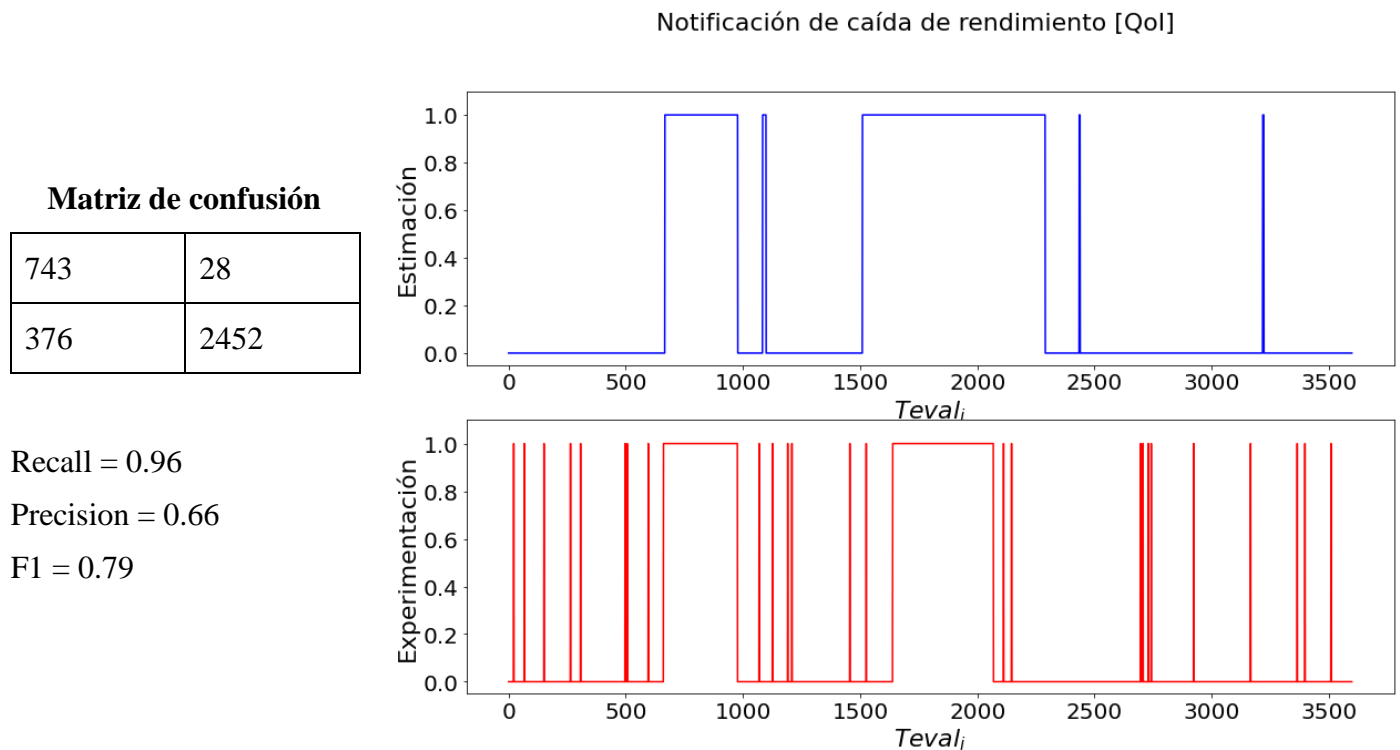


Fig. 81: Resultados del sistema de notificación QoI en entorno rural con pérdidas.

Escenario con pérdidas – QoE – Entorno Urbano.

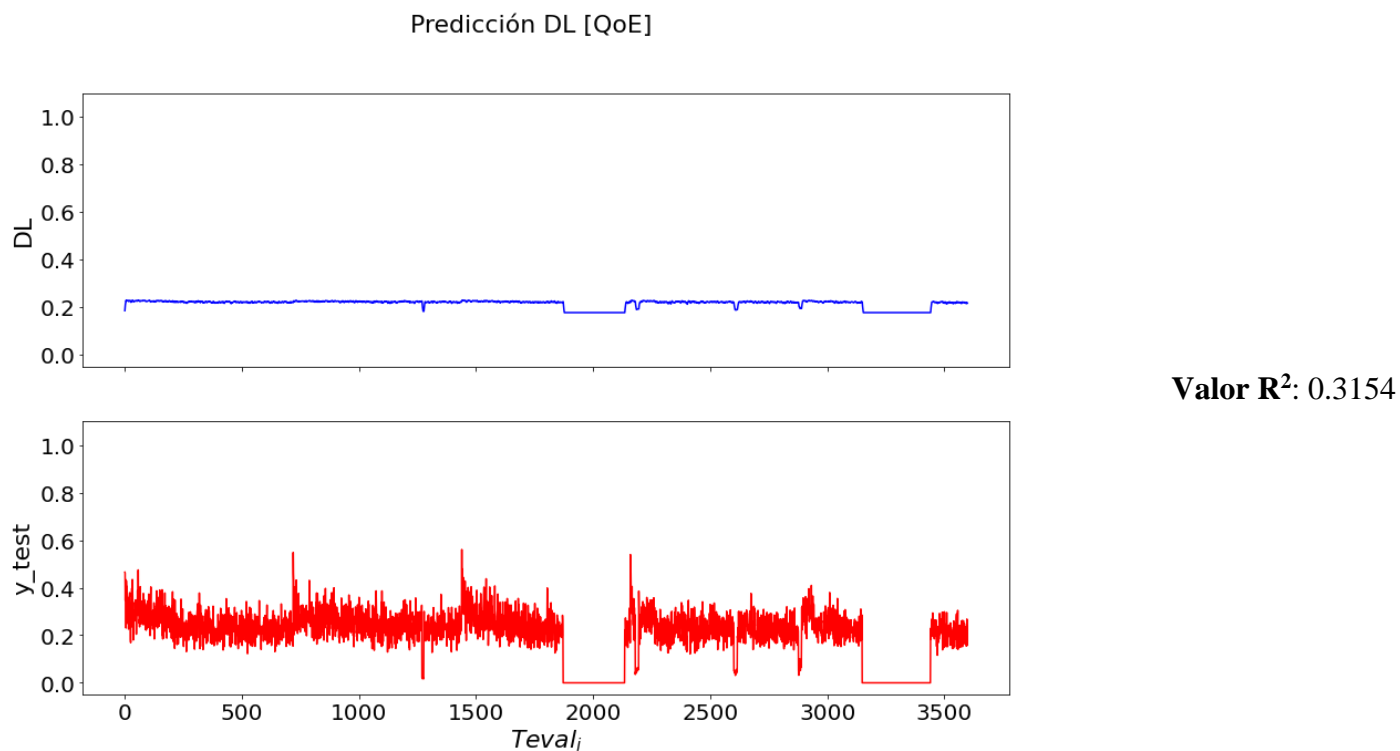


Fig. 82: Resultados de estimación QoE en entorno urbano con pérdidas.

Matriz de confusión

550	32
63	2954

Recall = 0.95

Precision = 0.9

F1 = 0.92

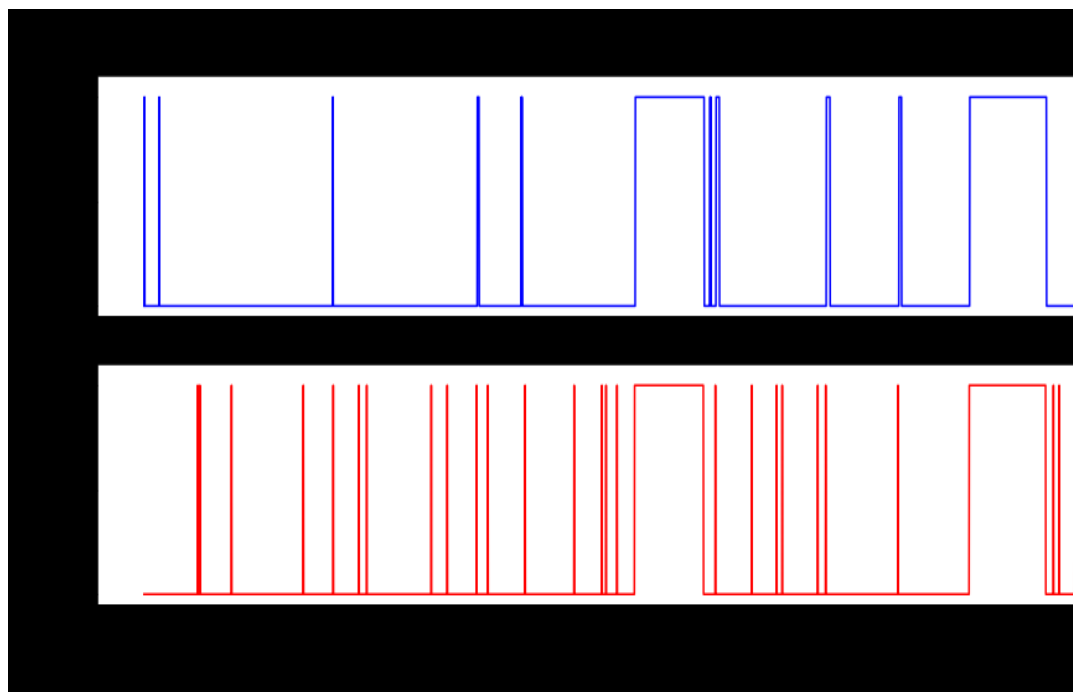


Fig. 83: Resultados del sistema de notificación QoE en entorno urbano con pérdidas.

Escenario con pérdidas – QoE – Entorno Suburbano.

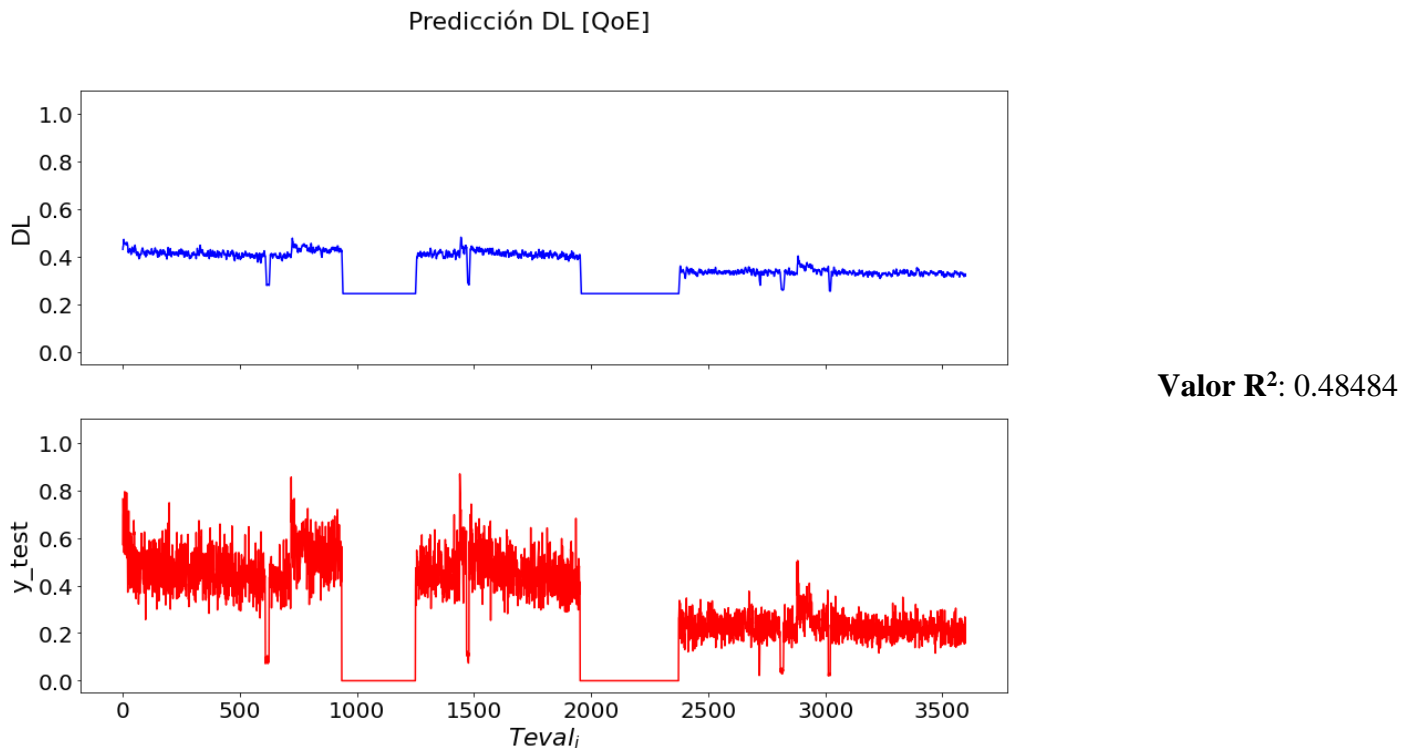


Fig. 84: Resultados de estimación QoE en entorno suburbano con pérdidas.

Matriz de confusión

726	25
20	2828

Recall = 0.97

Precision = 0.97

F1 = 0.97

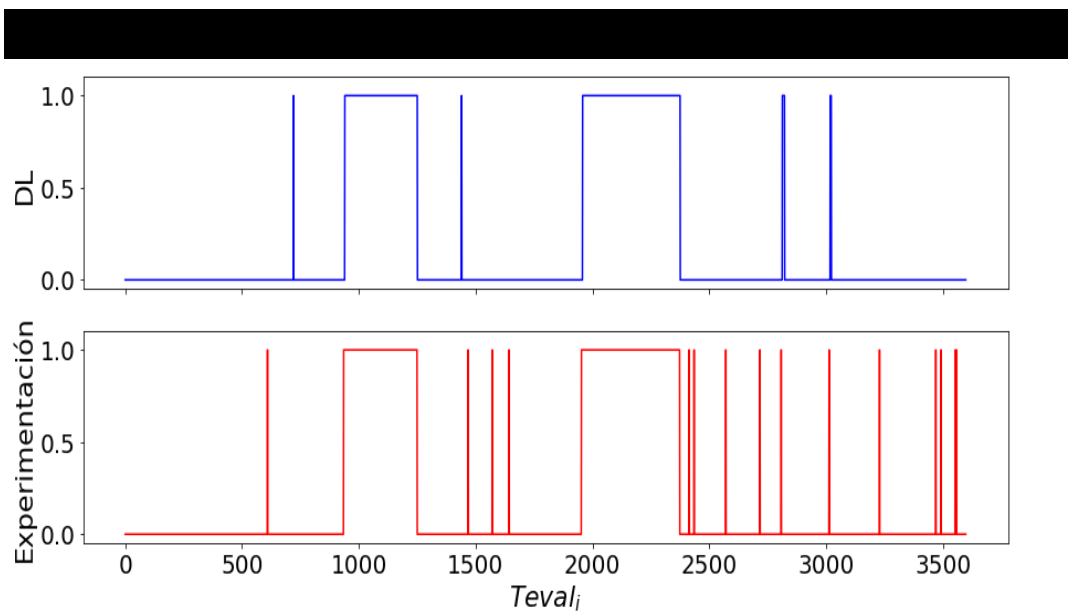


Fig. 85: Resultados del sistema de notificación QoE en entorno suburbano con pérdidas.

Escenario con pérdidas – QoE – Entorno Rural.

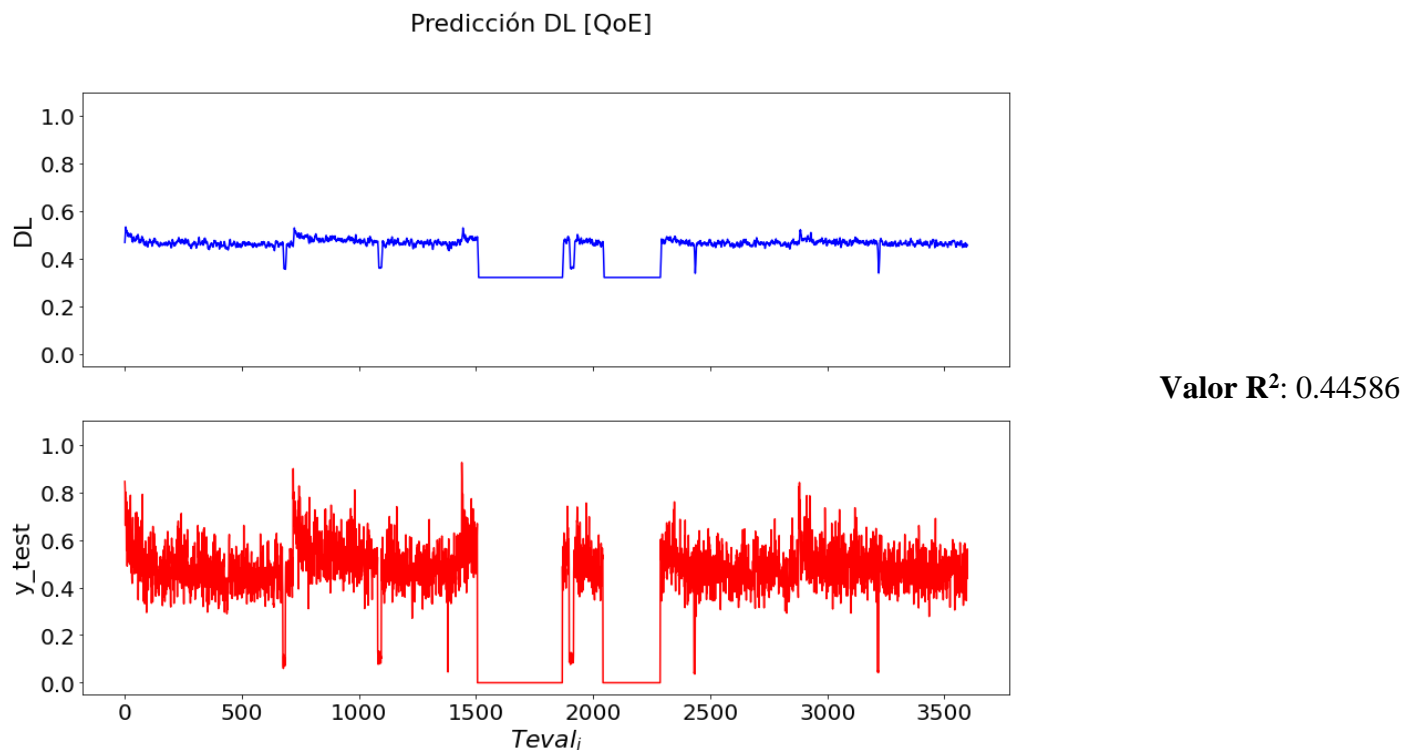


Fig. 86: Resultados de estimación QoE en entorno rural con pérdidas.

Matriz de confusión

596	18
2	2983

Recall = 0.97

Precision = 1

F1 = 0.98

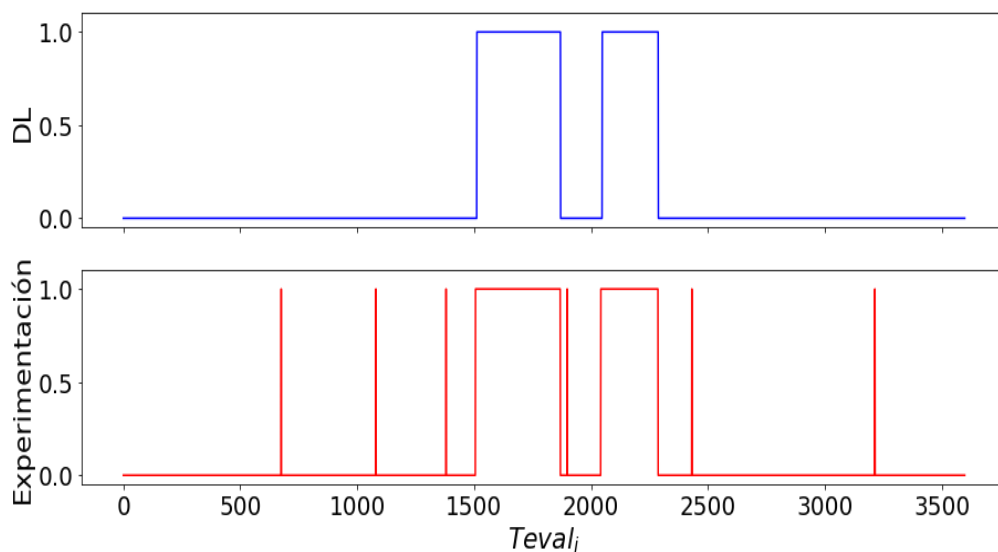
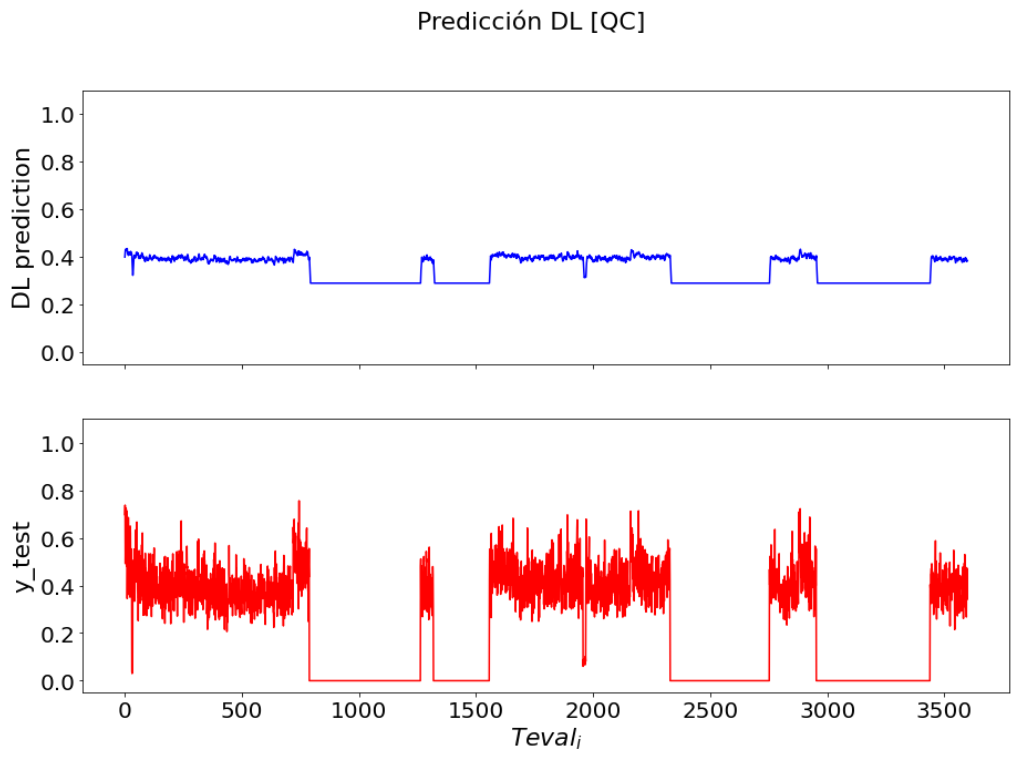


Fig. 87: Resultados del sistema de notificación QoE en entorno rural con pérdidas.

Escenario con pérdidas – QC – Entorno Urbano.



Valor R²: 0.1332

Fig. 88: Resultados de estimación QC en entorno urbano con pérdidas.

Matriz de confusión

1600	29
4	1966

Recall = 0.98

Precision = 1

F1 = 0.99

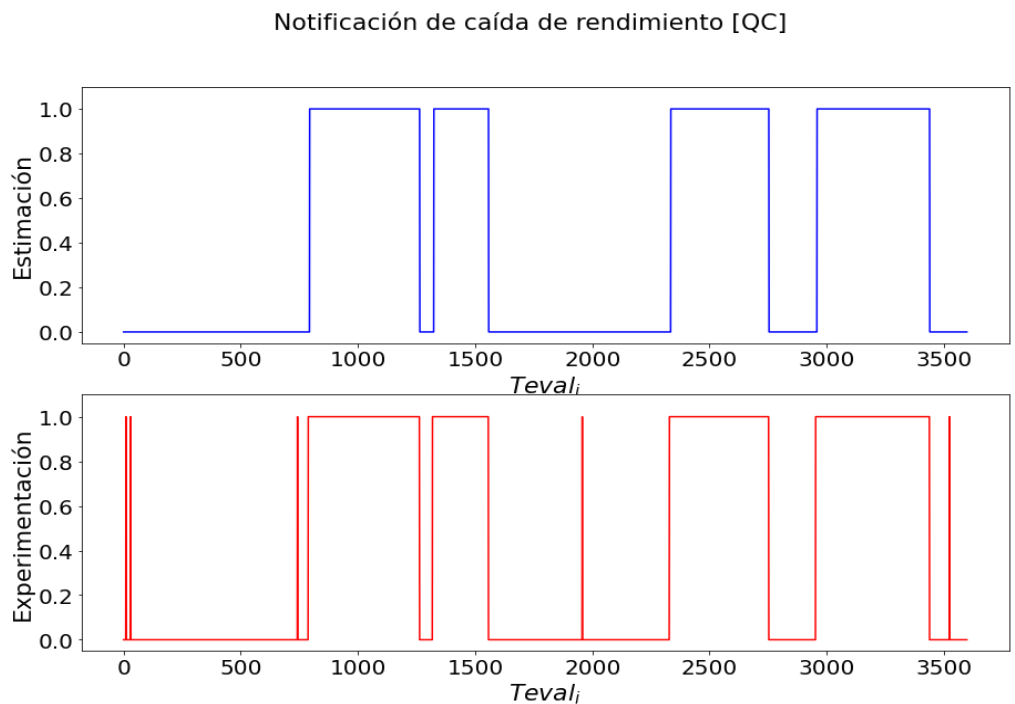


Fig. 89: Resultados del sistema de notificación QC en entorno urbano con pérdidas.

Escenario con pérdidas – QC – Entorno Suburbano.

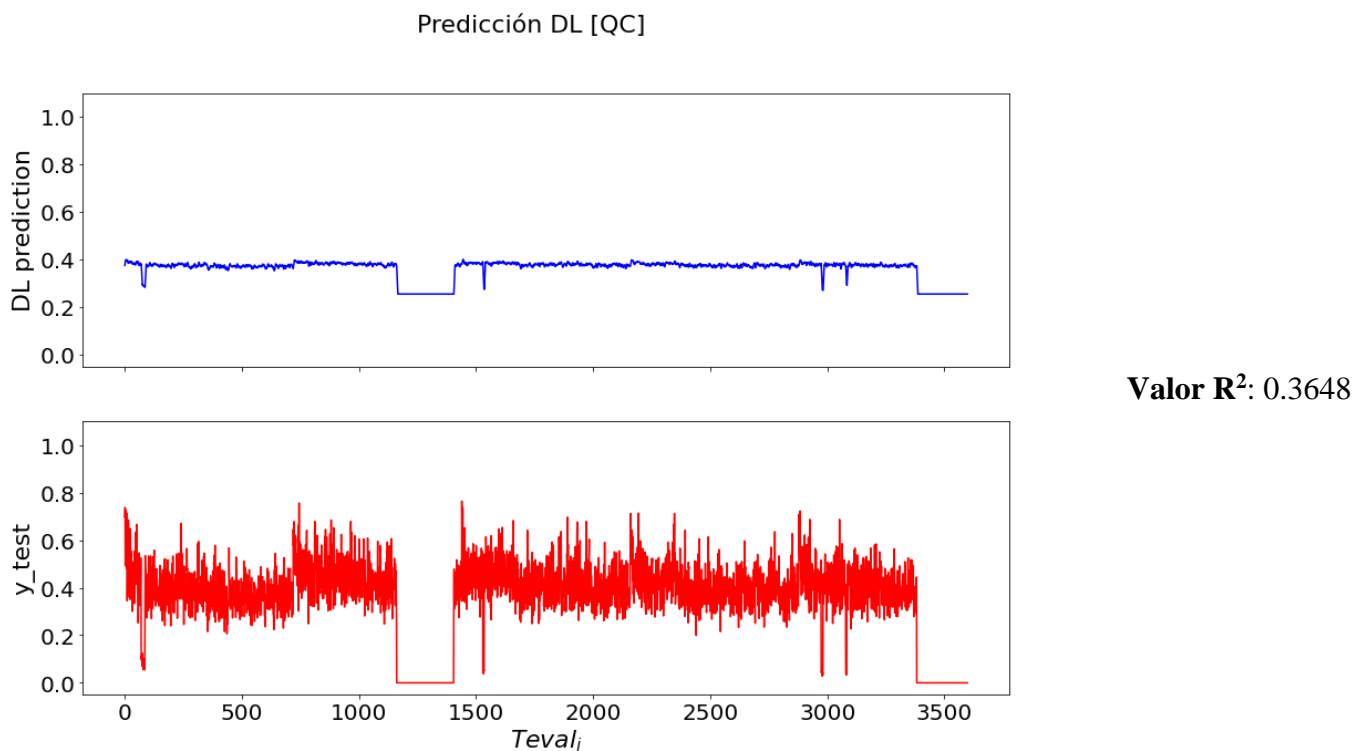


Fig. 90: Resultados de estimación QC en entorno suburbano con pérdidas.

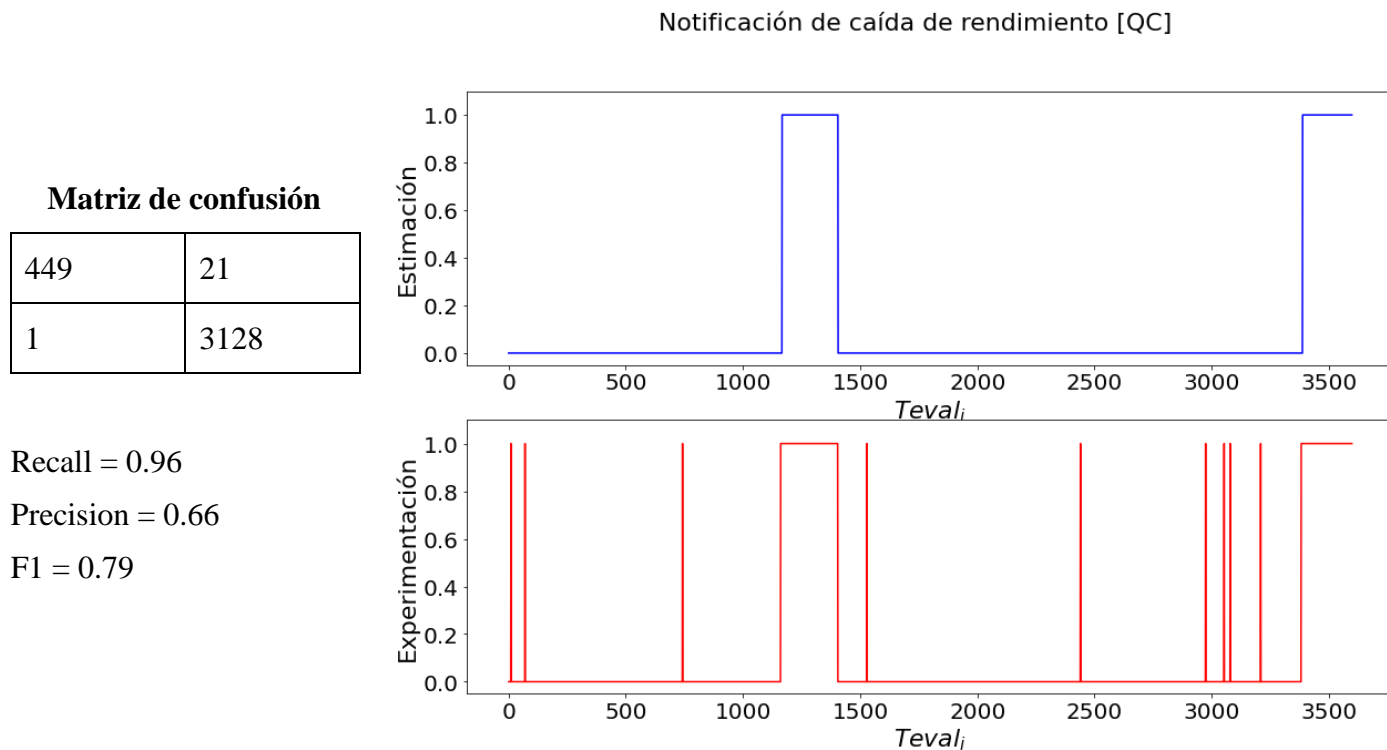
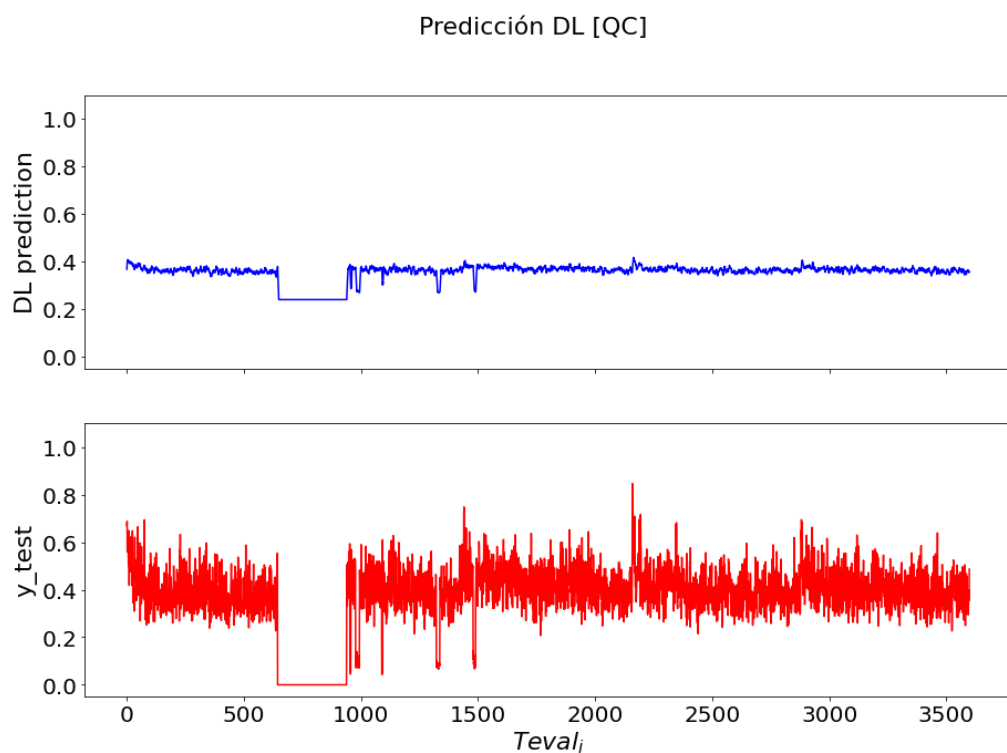


Fig. 91: Resultados del sistema de notificación QC en entorno suburbano con pérdidas.

Escenario con pérdidas – QC – Entorno Rural.



Valor R^2 : 0.35085

Fig. 92: Resultados de estimación QC en entorno rural con pérdidas.

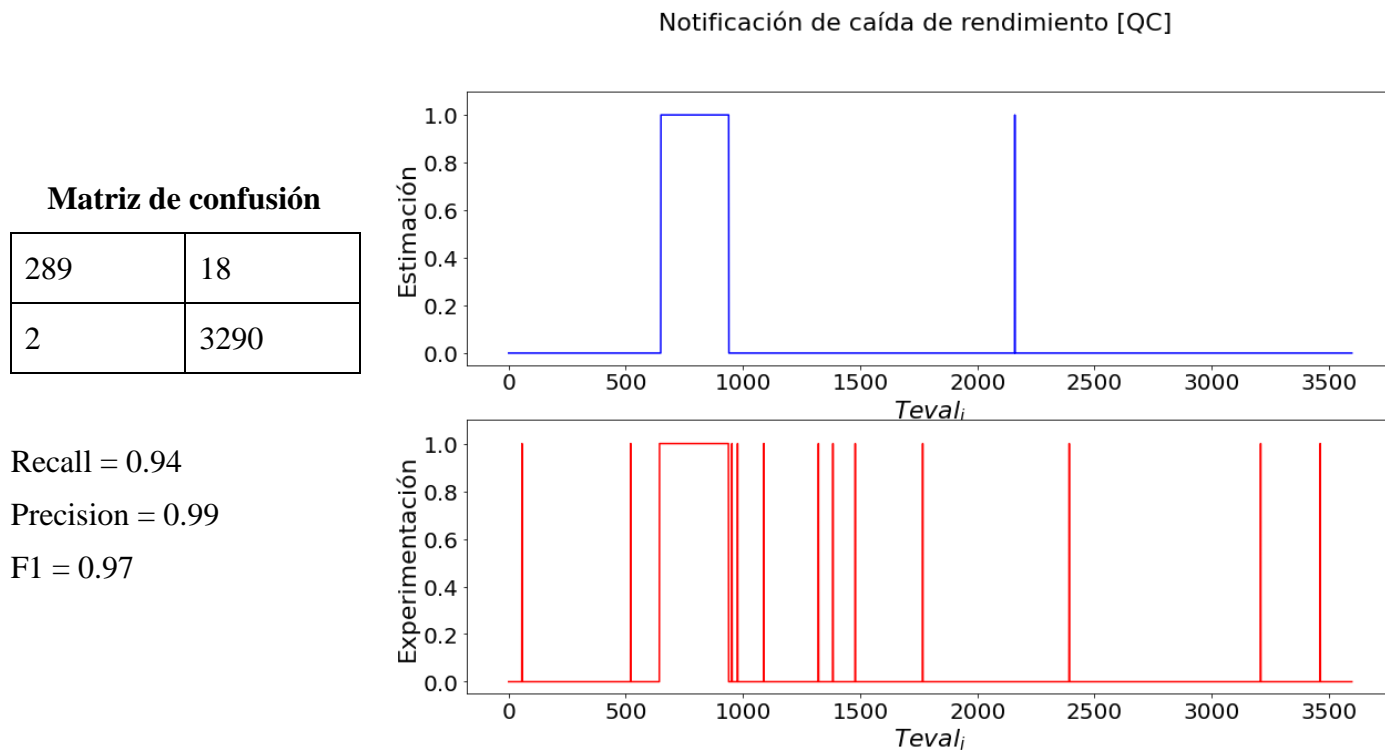


Fig. 93: Resultados del sistema de notificación QC en entorno rural con pérdidas.

5.3.5. Estudio comparativo

Como vimos en la literatura existen algunos artículos que abordan temáticas similares a este caso de estudio. Con el objetivo de poder realizar una comparativa con otros trabajos, nos centraremos en el caso de la monitorización, evaluación y estimación de QoE. En particular, compararemos nuestra propuesta con el trabajo presentado en [135]. En ese artículo, los autores afirman que es necesario obtener datos de todos los niveles de la pila de protocolos para disponer de una estimación precisa. En lo que respecta a la estimación, comparan diferentes modelos de ML para predecir el valor de QoE pero es el modelo de clasificación C4.5 basado en árboles de decisión con el que obtienen el mejor rendimiento. Este modelo calculará el valor de MOS en función de una serie de parámetros descritos (ver Tabla 15).

Tabla 15: Parámetros de entrada en el modelo planteado por [135].

Variables de entrada	Variable de salida
GW_{Av}	MOS
Jitter	
Delay	
TH_{bps}	
Quantity	
PDR	
Recall	

A priori, ambos modelos no serían comparables ya que abordan la problemática mediante dos soluciones distintas, uno con un algoritmo de clasificación y otro con un algoritmo de regresión. Para poder compararlo hacemos uso de los resultados del sistema de notificaciones a partir de las estimaciones de cada modelo respecto al valor real mediante la matriz de confusión y las métricas de *Recall*, *Precision* y F1. Esta comparativa solamente se realiza para la métrica QoE ya que es la única que los autores utilizan.

En un escenario ideal y bajo un entorno urbano, Fig. 94 muestra el resultado del modelo de estimación en escala normalizada [0,1] junto al resultado obtenido por el modelo de estimación propuesto y el valor real del conjunto de *testing*. Fig. 95 muestra los resultados del sistema de notificación para cada uno de los modelos de estimación comparados. En un entorno rural, Fig. 96 y Fig. 97 comparan los resultados de los modelos de estimación y sistemas de notificación respectivamente. Por último, Fig. 98 y Fig. 99 hacen lo propio sobre un entorno rural.

En este caso, también se compara el rendimiento de ambos modelos sobre un entorno con pérdidas y retardos. Los resultados del modelo de estimación quedan representados por Fig. 100 sobre un entorno urbano, mientras que Fig. 101 recoge los resultados del sistema de notificación. Fig. 102 y Fig. 103 hacen lo propio para un entorno suburbano, mientras que Fig. 104 y Fig. 105 con el entorno rural.

Al comparar los resultados se aprecia principalmente un menor número de FP en nuestro modelo. A partir de estas mejoras en la matriz de confusión (ver Fig. 37) se

proyecta una mejora en las métricas utilizadas, siendo esta mejora del 11,4% en *Precision* y del 7,3% en *Recall*.

Finalmente se ofrece una tabla comparativa para cada caso donde se muestran los resultados de las métricas de rendimiento del sistema de notificación del modelo planteado frente a la contribución de la literatura.

Escenario ideal – QoE – Entorno Urbano.

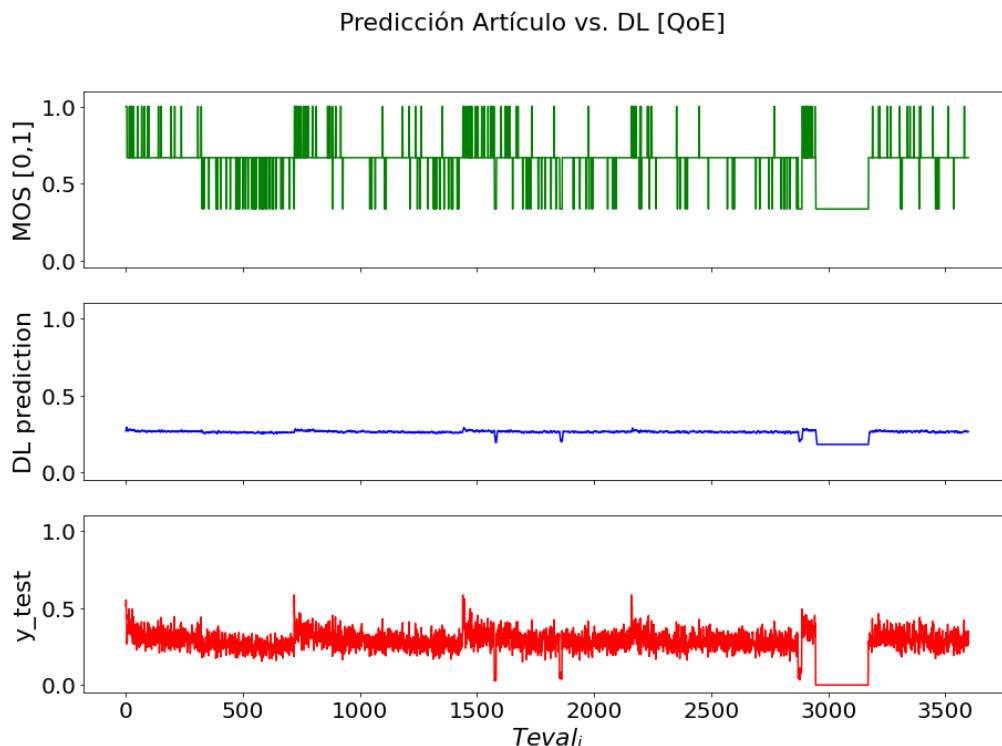


Fig. 94: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno urbano ideal.

Matriz de confusión
[135]

221	14
30	3334

Recall: 0.94

Precision: 0.88

F1: 0.91

Matriz de confusión
caso de estudio

221	14
12	3352

Recall: 0.94

Precision: 0.95

F1: 0.94

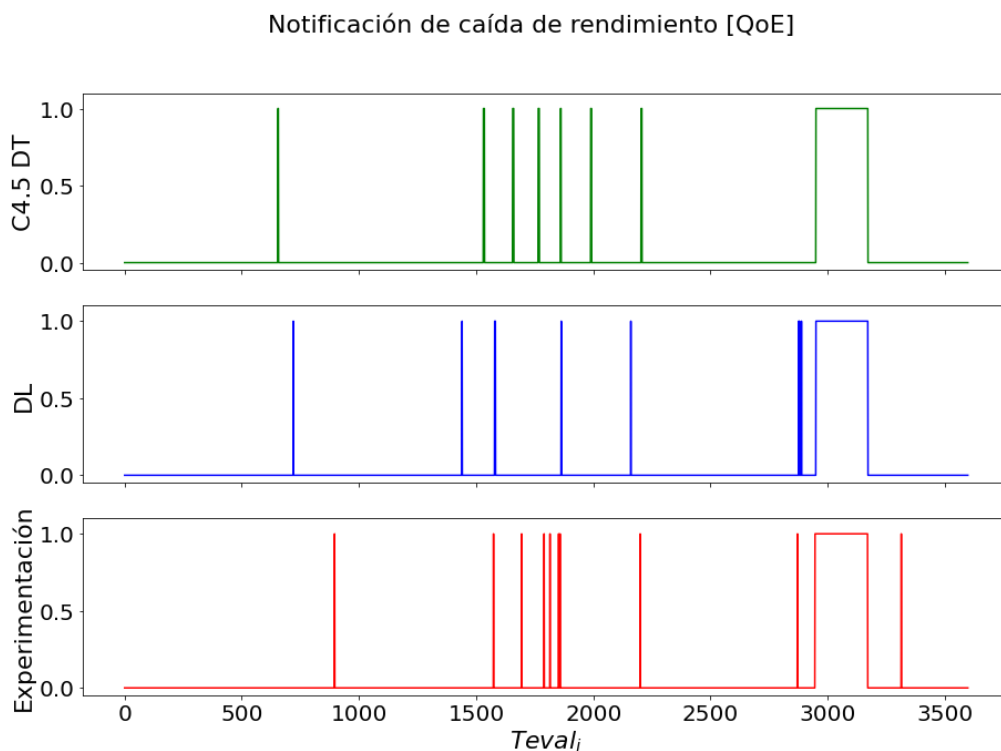


Fig. 95: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno urbano ideal.

Escenario ideal – QoE – Entorno Suburbano.

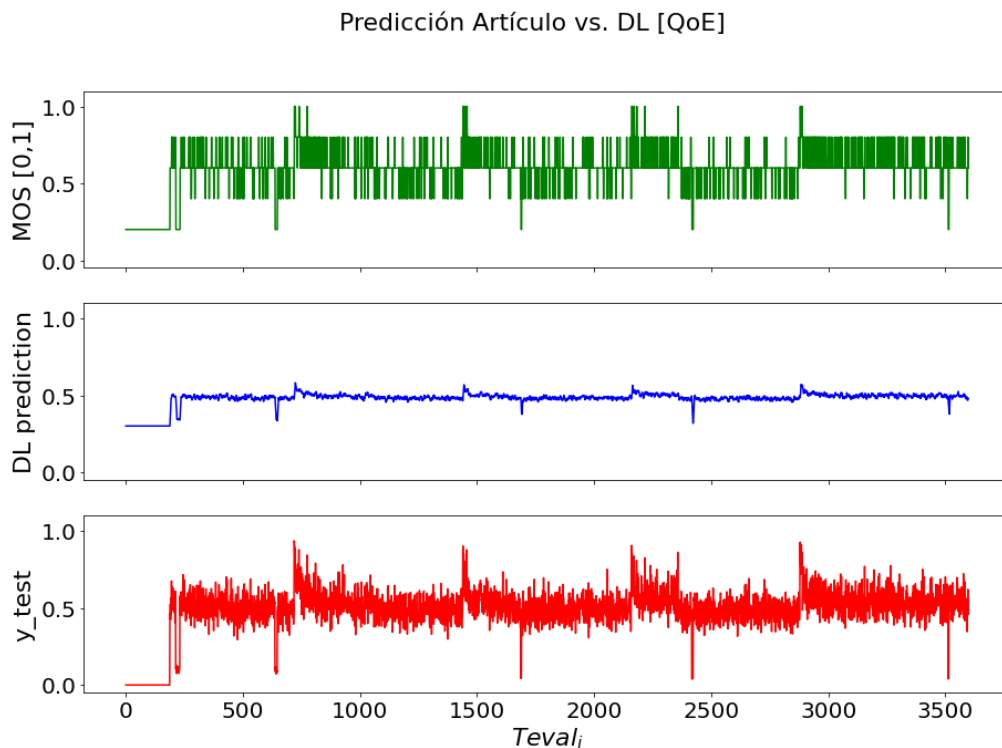


Fig. 96: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno suburbano ideal.

Matriz de confusión
[135]

187	8
41	3363

Recall: 0.96

Precision: 0.82

F1: 0.88

Matriz de confusión
caso de estudio

187	8
1	3403

Recall: 0.96

Precision: 0.99

F1: 0.98

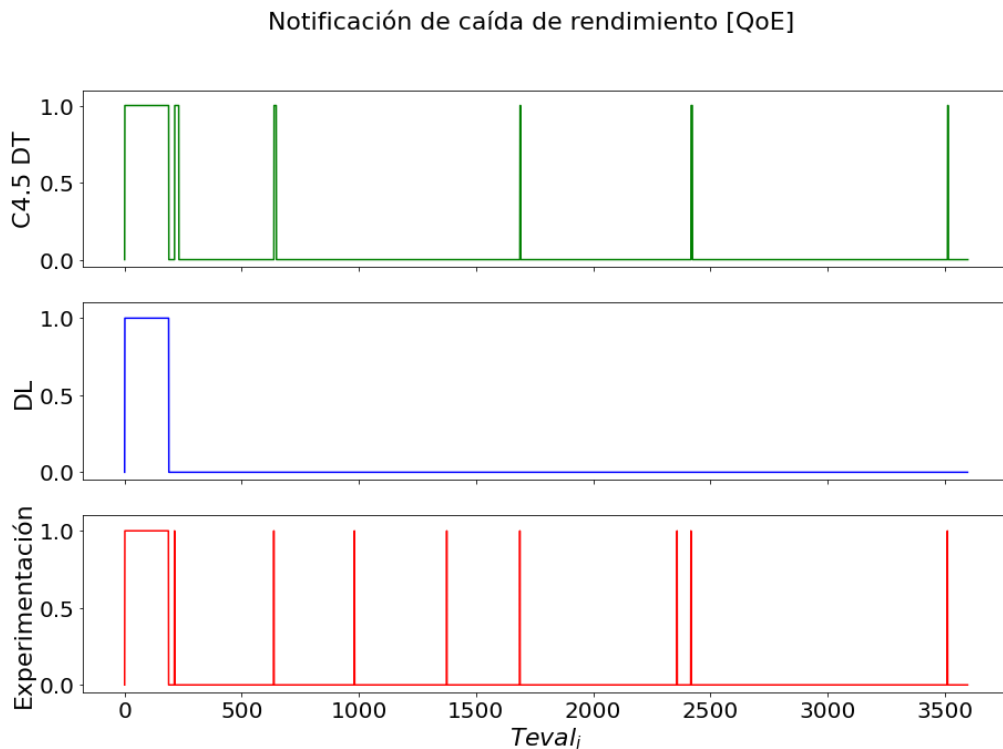


Fig. 97: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno suburbano ideal.

Escenario ideal – QoE – Entorno Rural.

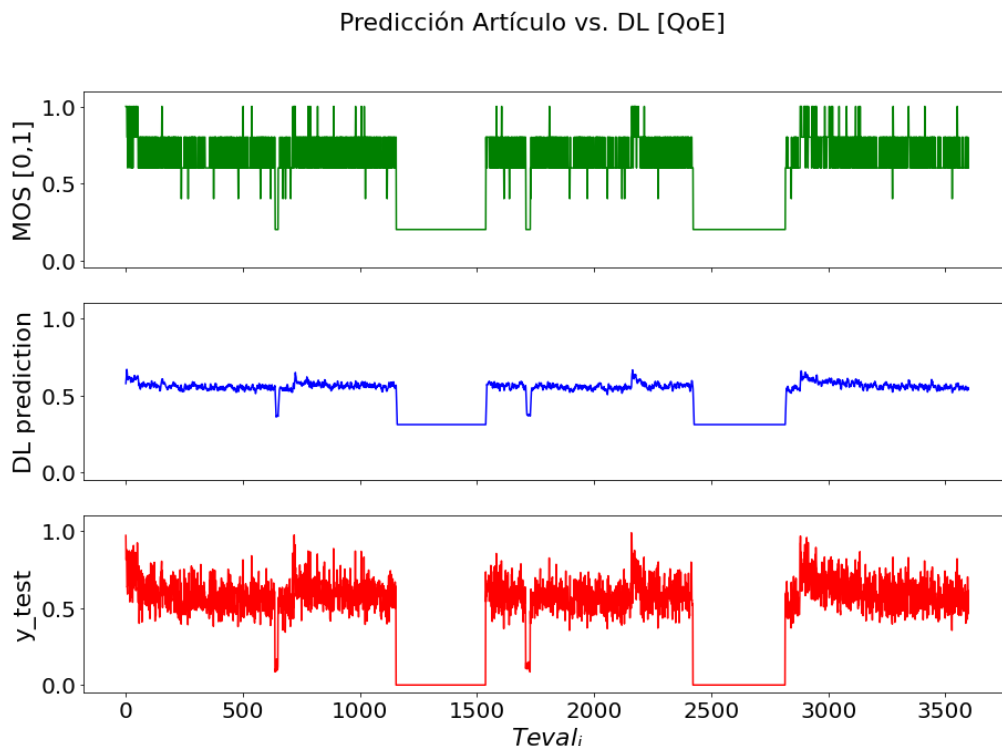


Fig. 98: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno rural ideal.

Matriz de confusión
[135]

747	33
34	2785

Recall: 0.96

Precision: 0.96

F1: 0.96

Matriz de confusión
caso de estudio

765	15
2	2817

Recall: 0.98

Precision: 1

F1: 0.99

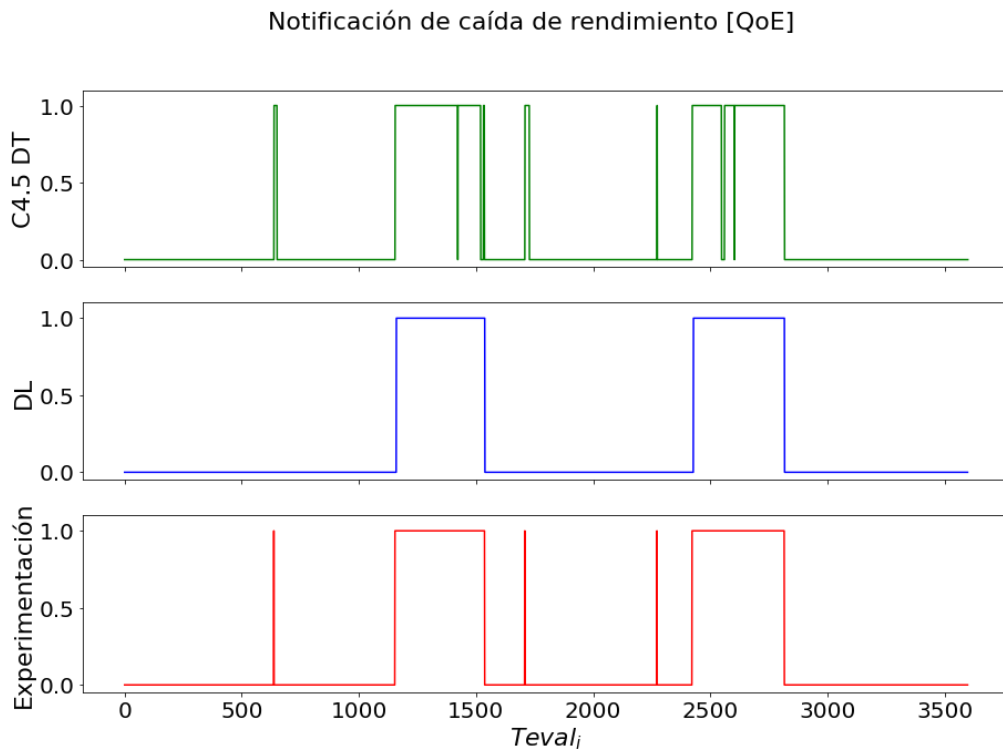


Fig. 99: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno rural ideal.

Escenario con pérdidas – QoE – Entorno Urbano.

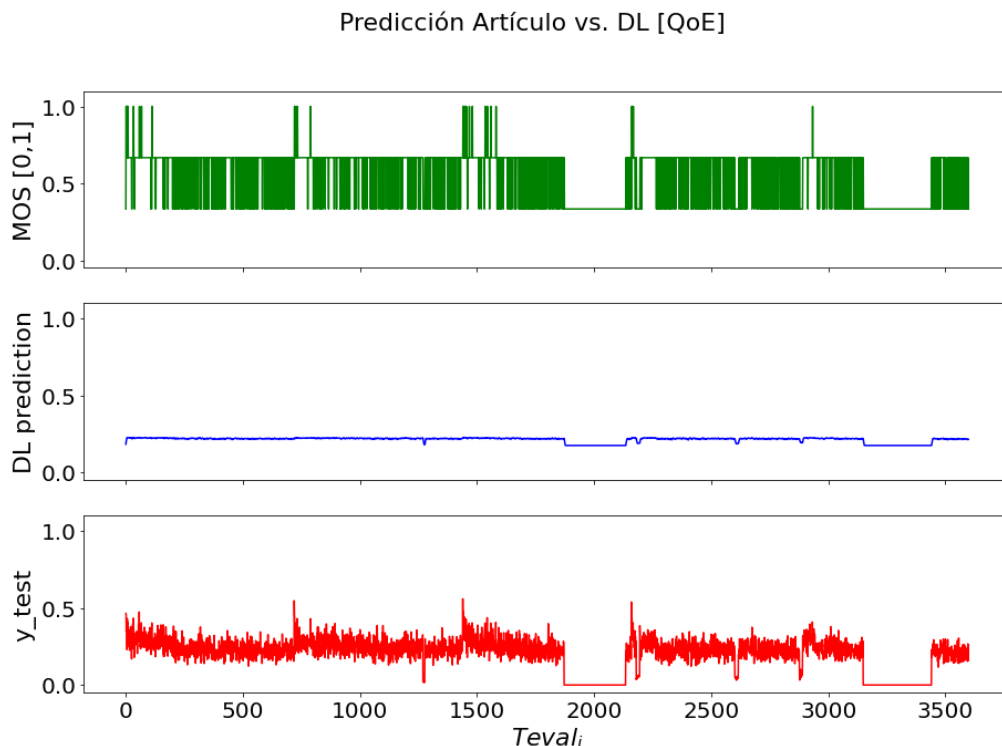


Fig. 100: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno urbano con pérdidas.

Matriz de confusión
[135]

541	41
79	2938

Recall: 0.93

Precision: 0.87

F1: 0.90

Matriz de confusión
caso de estudio

550	32
63	2954

Recall: 0.95

Precision: 0.90

F1: 0.92

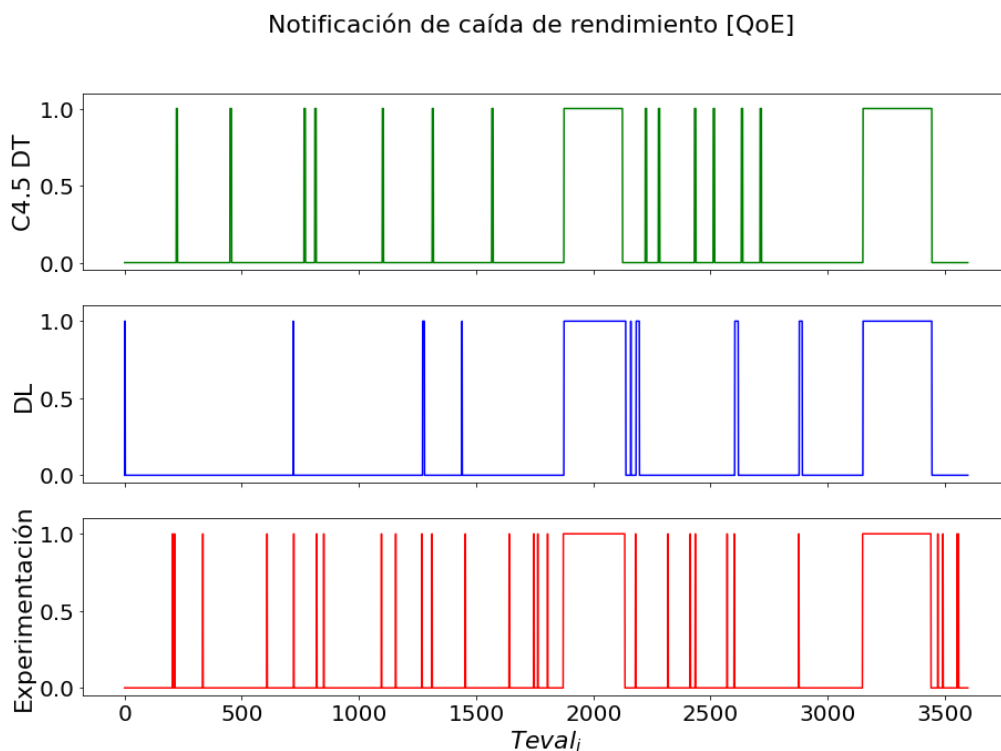


Fig. 101: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno urbano con pérdidas.

Escenario con pérdidas – QoE – Entorno Suburbano.

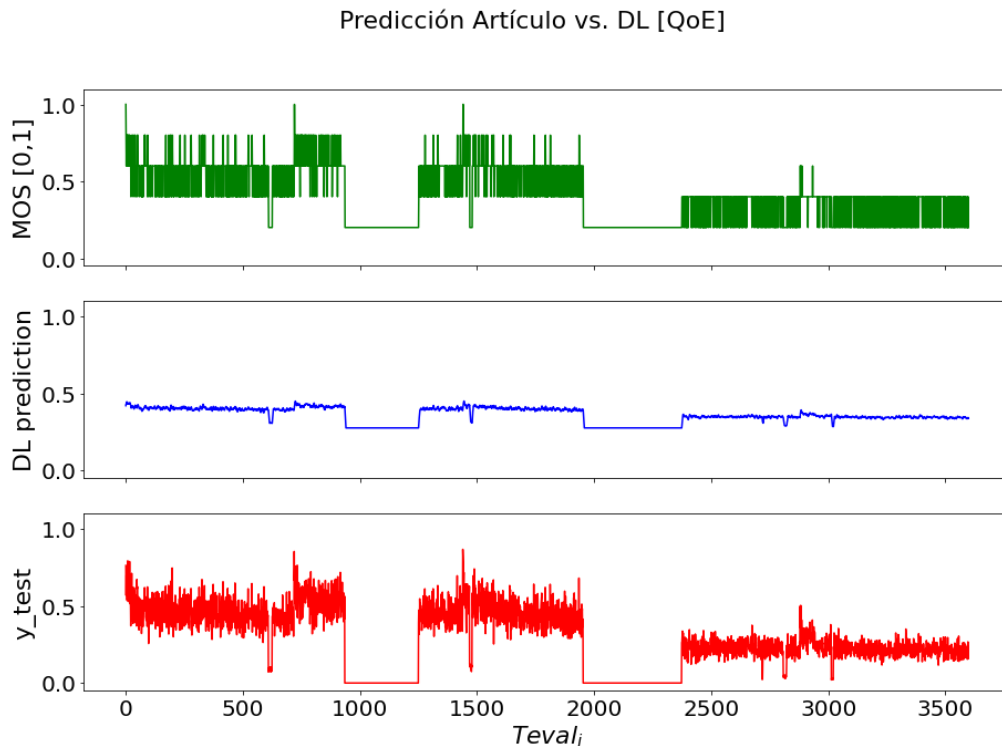


Fig. 102: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno suburbano con pérdidas.

Matriz de confusión
[135]

549	202
247	2601

Recall: 0.73

Precision: 0.69

F1: 0.71

Matriz de confusión
caso de estudio

726	25
20	2828

Recall: 0.97

Precision: 0.97

F1: 0.97

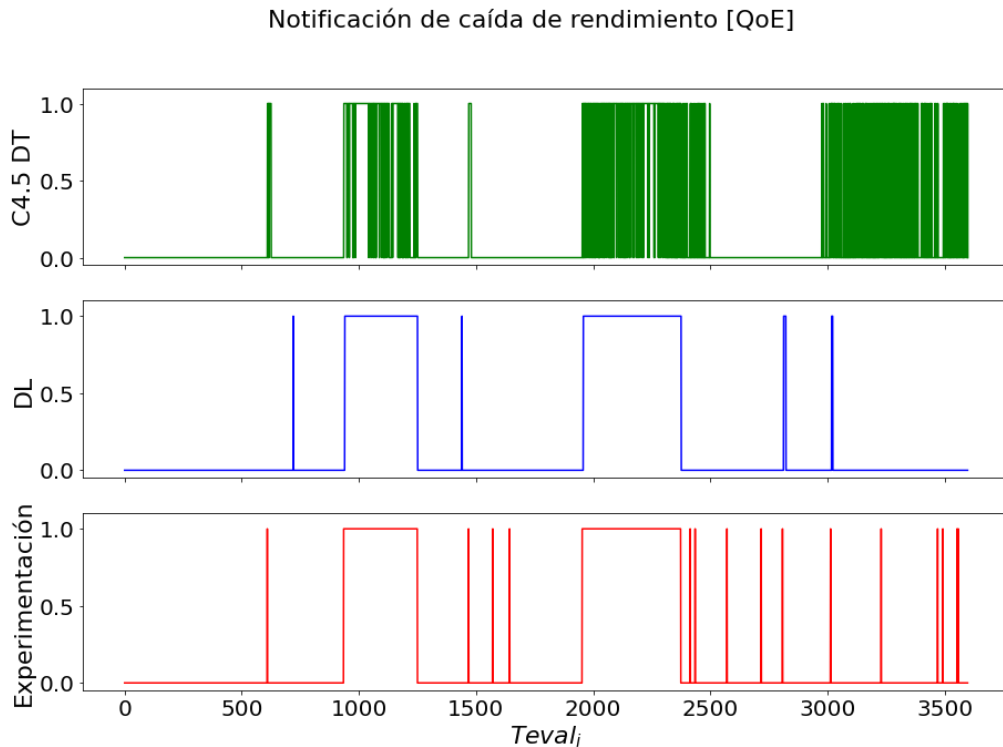


Fig. 103: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno suburbano con pérdidas.

Escenario con pérdidas – QoE – Entorno Rural.

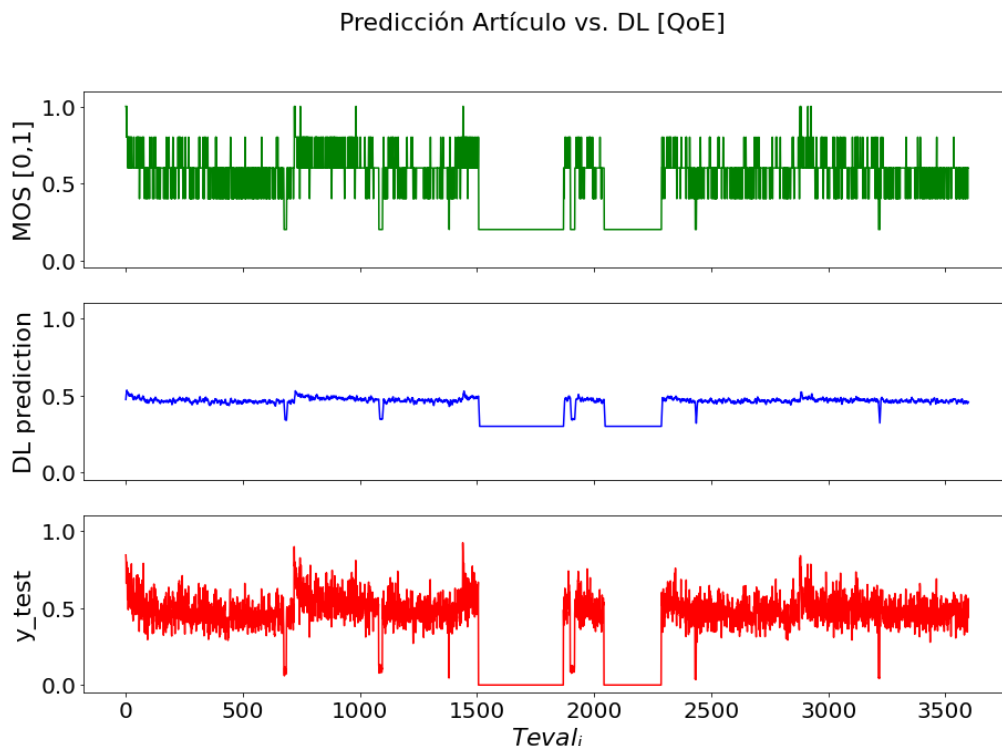


Fig. 104: Comparativa de la estimación de los modelos respecto a los valores de entrenamiento para QoE en entorno rural con pérdidas.

Matriz de confusión
[135]

533	81
46	2939

Recall: 0.87

Precision: 0.92

F1: 0.89

Matriz de confusión
caso de estudio

596	18
2	2983

Recall: 0.97

Precision: 1

F1: 0.98

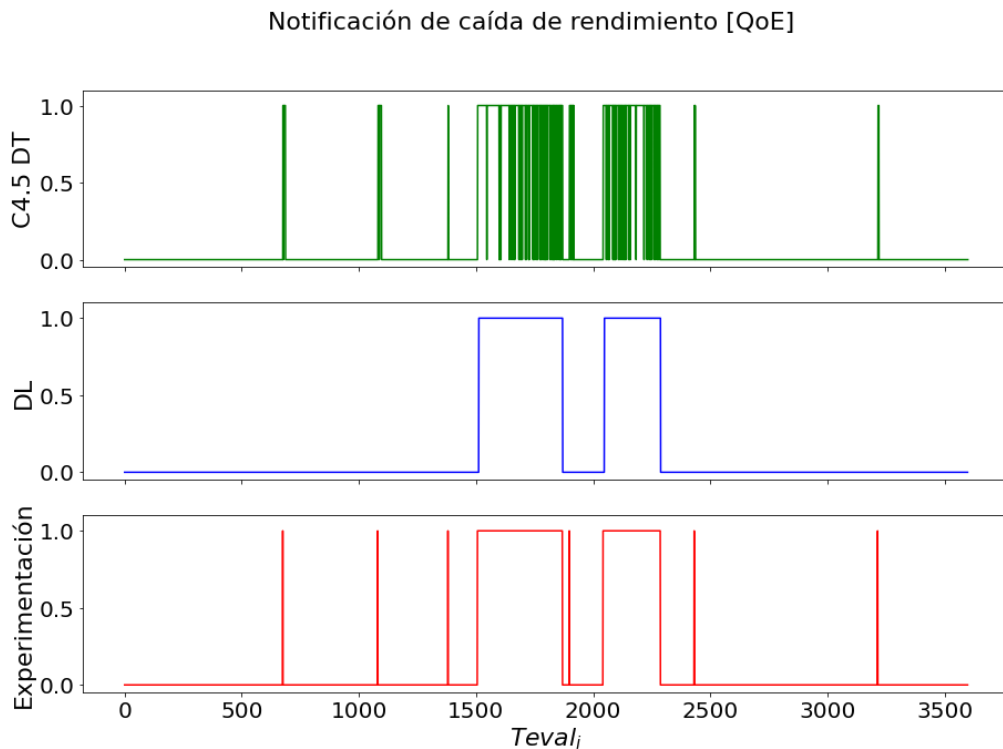


Fig. 105: Comparativa del resultado del sistema de notificación para caídas de QoE en entorno rural con pérdidas.

5.4. Conclusión

Hoy en día, la inteligencia artificial está en auge y su aplicación en nuestra vida cotidiana va en aumento. Estas técnicas nos permiten tomar decisiones de forma más precisa según un histórico de datos previos existente. En este capítulo, hemos trabajado con técnicas y modelos que permiten predecir o estimar datos futuros con una baja probabilidad de error mediante el uso de varios modelos.

Por un lado, tenemos la clasificación y predicción de eventos de *stalling* en reproducción de contenido multimedia. La disminución de estos sucesos supone un factor clave para plataformas multimedia tales como Netflix, YouTube, Amazon Prime, etc., ya que supone una mejora notable del rendimiento. En la sección 5.2. se propone un algoritmo de clasificación y un modelo de predicción con el que se podrían detectar y medir la duración de los eventos de *stalling*, empleando únicamente información de la capa de red y transporte. Los modelos han sido entrenados y testeados mediante un conjunto de datos provenientes de un flujo de video que emplea el protocolo DASH. Los resultados son muy prometedores, alcanzando valores cerca del máximo en *Precision*, *Recall* y F1. Como trabajo futuro se propone el uso de flujos simultáneos y de otros protocolos de *streaming*.

Por otro lado, se ha propuesto la inclusión de análisis predictivo en las componentes de calidad QoX del modelo de evaluación para IoT presentado previamente. Se trata de un modelo de regresión desarrollado a partir de una técnica de DL como LSTM bidireccional. A este modelo se le añade la implementación de ventanas deslizantes para hacer uso de

series temporales en la toma de decisiones. Como resultado se obtienen buenos rendimientos en la estimación del valor de la métrica. Por último en este caso de estudio, se comparan los resultados obtenidos con una contribución de la literatura que utiliza un modelo de clasificación. Como ambos modelos no son comparables a priori, se utiliza un sistema de notificación cuando la componente observada cae por debajo del 50% respecto al último registro y mantiene este valor constante pasados varios T_{evals} . Comparando resultados numéricos con la propuesta de la literatura, la implementación mostrada en este caso de estudio mejora el rendimiento bajo todos los entornos y escenarios.

5.5. Contribuciones Relacionadas a este Capítulo

H.F. Bermudez, J.M. Martinez-Caro, R. Sanchez-Iborra, J.L. Arciniegas, M.-D. Cano, “Live video-streaming evaluation using the ITU-T P.1203 QoE model in LTE networks”, *Comput. Networks* (2019), vol. 165, pp. 1–12, <https://doi.org/10.1016/j.comnet.2019.106967>

Factor de Impacto: 3.111 (Q2), Rank: 85/266 in Engineering, Electrical & Electronic.

J.M. Martinez-Caro, M.-D. Cano, “On the identification and prediction of stalling events to improve QoE in video streaming”, *Computer Communications* (2020), pp. 1–10. En proceso de revision de pares.

Factor de Impacto: 2,816 (Q2), Rank: 106/266 in Engineering, Electrical & Electronic.

J.M. Martinez-Caro, M.D. Cano, “Predictive analysis for quality monitoring in IoT services”, *Measurements* (2020). En proceso de revisión.

Factor de Impacto JCR: 3,364 (Q1). Rank: 22/91 (Engineering, Multidisciplinary); 13/64 (Instruments & Instrumentation).

Conclusiones y trabajo futuro

En el paradigma IoT, sigue siendo todo un desafío la evaluación de prestaciones. Para ello, una adecuada definición de métricas de rendimiento es fundamental, siendo flexible al entorno a tratar.

La presente tesis doctoral aborda este reto proponiendo un nuevo modelo con visión holística basado en cuatro componentes de calidad, para poder monitorizar y satisfacer las prestaciones en IoT bajo los dominios de coste-beneficio. El dominio beneficio está formado por las componentes QoD, QoI y QoE, mientras que el dominio coste únicamente por QC. La componente QoD mide la calidad en el origen de la cadena, sobre el dato recién capturado por los sensores. Por su parte, la componente QoI evalúa la calidad de la información obtenida tras el procesamiento de los datos por parte del servidor. La componente QoE intenta reflejar tanto el comportamiento de la red, en términos tradicionales de QoS, como otros parámetros, que serían personalizables según el servicio a evaluar. Por último, la componente QC intenta captar la compensación de ganancia de rendimiento en función del consumo de recursos en los dispositivos.

El modelo propuesto se ha llevado a examen bajo diferentes entornos haciendo uso de un *test-bed* común. El escenario escogido como caso de estudio es un sistema de monitorización de la calidad del aire, implementado con tecnología LoRa/LoRaWAN. Estas pruebas se realizaron siguiendo una metodología de simulación, empleando las herramientas OMNeT++, INET y FLoRa. Para diferentes entornos con nodos estáticos, se

produce una variación mínima de las métricas de QoD y QC ya que no se ven afectadas por la caída de mensajes en la red. El resto de componentes sí mejoran con el aumento de mensajes recibidos correctamente en el servidor, tanto en escenarios ideales como con pérdidas, diferenciándose claramente los resultados según sea un entorno urbano, suburbano o rural. En el caso de nodos móviles, asumiendo que los nodos de medición continúan usando tecnología LoRa/LoRaWAN pero que en este caso están embebidos en drones, se reafirman en gran medida los resultados del caso de nodos estáticos y, de forma adicional, se ha estudiado el comportamiento del sistema de monitorización según diferentes patrones de movilidad, número de nodos móviles y velocidad de movimiento en un entorno suburbano. En este caso, se obtiene un buen rendimiento en todos los patrones de movilidad, excepto cuando los dispositivos se desplazan en grupo, ya que solo se obtiene información de una zona reducida del área monitorizada.

Por último, es de gran importancia en este contexto de la IoT y de la evaluación de prestaciones de los servicios subyacentes, poder adelantarse a las caídas de rendimiento para evitar incidentes. Con este objetivo, se han propuesto dos modelos diferentes para la predicción del desempeño en función del histórico de datos más reciente. Por un lado, se ha presentado un modelo de clasificación y predicción de eventos de *stalling*, considerados atenuantes de la calidad en servicios de distribución y reproducción multimedia. Los resultados obtenidos son una tasa de error media del 10,83%, así como una *Precision* y F1 como mínimo de 0,956 y 0,923, respectivamente. Por otro, se ha propuesto un modelo para predecir el valor de cada una de las componentes de calidad (QoD, QoI, QoE y QC) de forma autónoma para poder notificar cuando se produzca una caída considerable de

calidad. Al comparar los resultados obtenidos con otros de la literatura, en particular un trabajo basado en C4.5 DT, se alcanzan mejoras significativas para cada escenario y entorno; en media, se obtiene una mejora en las métricas de clasificación del 9,1% en F1, 10,3% en *Recall* y 8% en *Precision*.

IoT ha llegado para quedarse con el objetivo de facilitar las tareas diarias e industriales, permitiendo una amplia variedad de soluciones. Aspectos como la eficiencia energética y la experiencia del usuario se hacen imprescindibles bajo este paradigma. De ahí la importancia del uso de técnicas de optimización que permitan ahorrar recursos y adaptar el servicio a las condiciones y necesidades del usuario, donde las contribuciones de esta tesis podrán servir como base.

Referencias

- [1] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 855–873, 2017, doi:10.1109/COMST.2017.2652320, ISSN:1553877X.
- [2] C. W. Thompson, “Smart Devices and Soft Controllers,” *IEEE Internet Comput.*, vol. 9, no. 1, pp. 82–85, 2005, doi:10.1109/MIC.2005.22.
- [3] S. Ray, Y. Jin, and A. Raychowdhury, “The Changing Computing Paradigm with Internet of Things: A Tutorial Introduction,” *IEEE Des. Test*, vol. 33, no. 2, pp. 76–96, 2016, doi:10.1109/MDAT.2016.2526612, ISSN:21682356.
- [4] N. Farah, M. Avishek, F. Muhammad, A. Rahman, M. Rafni, and D. Md., “Application of Machine Learning Approaches in Intrusion Detection System: A Survey,” *Int. J. Adv. Res. Artif. Intell.*, vol. 4, no. 3, pp. 9–18, 2015, doi:10.14569/ijarai.2015.040302, ISSN:21654050.
- [5] “OMNeT++ Simulator.” [Online]. Available: <https://omnetpp.org/>. [Accessed: 05-Feb-2019].
- [6] “INET Framework.” 2019. [Online]. Available: <https://inet.omnetpp.org/>. [Accessed: 05-Feb-2019].
- [7] “FLoRa Framework,” 2019. [Online]. Available: <https://flora.aalto.fi/>. [Accessed: 05-Feb-2019].
- [8] “The importance of accessibility in COVID-19 times,” *Nature Photonics*. [Online]. Available: <https://www.inclusite.com/en/news/the-importance-of-accessibility-in-covid-19-times/>. [Accessed: 02-Apr-2020].
- [9] J.-M. Martínez-Caro, A.-J. Aledo-Hernández, A. Guillen-Perez, R. Sanchez-Iborra, and M.-D. Cano, “A Comparative Study of Web Content Management Systems,” *Information*, vol. 9, no. 2, p. 27, 2018, doi:10.3390/info9020027, ISSN:2078-2489.
- [10] A. J. Aledo-Hernández, A. Guillen-Pérez, J.-M. Martínez-Caro, R. Sánchez-Iborra, and M.-D. Cano, “Sistemas de gestión de contenidos web: uso y estudio comparativo de su seguridad,” in *Jitel 2017*, 2017, p. 7, doi:10.4995/JITEL2017.2017.6558, ISBN:9788490485958.
- [11] “• IoT: number of connected devices worldwide 2012-2025 | Statista.” [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. [Accessed: 02-Apr-2020].
- [12] N. Shahid and S. Aneja, “Internet of Things: Vision, application areas and research

- challenges,” *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, vol. 10, no. 7, pp. 583–587, 2017, doi:10.1109/I-SMAC.2017.8058246, ISBN: 9781509032433, ISSN:1570-8705.
- [13] F. Wu, T. Wu, and M. R. Yuce, “An internet-of-things (IoT) network system for connected safety and health monitoring applications,” *Sensors (Switzerland)*, vol. 19, no. 1, 2019, doi:10.3390/s19010021, ISSN:14248220.
- [14] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of Lora: Long range & low power networks for the internet of things,” *Sensors (Switzerland)*, vol. 16, no. 9, 2016, doi:10.3390/s16091466, ISSN:14248220.
- [15] T. Janevski and M. Jankovic, “Quality of Service Regulation Manual,” 2017.
- [16] ITU-T, “Recommendation E.800: Definitions of terms related to Quality of Service,” 2008.
- [17] T. Janevski and M. Jankovic, “Definition of Quality of Service (QoS),” in *Quality of Service Regulation Manual*, 2017.
- [18] ITU-T, “Recommendation G. 1000 - Communications Quality of Service: Framework and Definitions,” 2001.
- [19] ITU-T, “Recommendation E.860 - Framework of a Service Level Agreement,” 2002.
- [20] T. Janevski and M. Jankovic, “Service Level Agreement (SLA),” in *Quality of Service Regulation Manual*, 2017.
- [21] T. Janevski and M. Jankovic, “QoS Agreement,” in *Quality of Service Regulation Manual*, 2017.
- [22] T. Janevski and M. Jankovic, “Activities in QoS regulation,” in *Quality of Service Regulation Manual*, 2017.
- [23] T. Janevski and M. Jankovic, “Key Performance Indicators (KPI),” in *Quality of Service Regulation Manual*, 2017.
- [24] ITU-T, “Recommendation P.10/G.100 - Communications Quality of user Experience: Framework and Definitions,” 2016.
- [25] E. Liotou, D. Tsolkas, N. Passas, and L. Merakos, “Quality of experience management in mobile cellular networks: Key issues and design challenges,” *IEEE Commun. Mag.*, vol. 53, no. 7, pp. 145–153, 2015, doi:10.1109/MCOM.2015.7158278, ISSN:01636804.
- [26] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *IEEE Netw.*, vol. 24, no. 2, pp. 36–41, 2010, doi:10.1109/MNET.2010.5430142, ISBN: 0890-8044 VO - 24, ISSN:08908044.

- [27] W. Tu, “Data-Driven QoS and QoE Management in Smart Cities: A Tutorial Study,” *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 126–133, 2019, doi:10.1109/MCOM.2018.1700870, ISSN:15581896.
- [28] Y. Chen, K. Wu, and Q. Zhang, “From QoS to QoE: A tutorial on video quality assessment,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2015, doi:10.1109/COMST.2014.2363139, ISBN: 1553-877X VO - 17, ISSN:1553877X.
- [29] K. Mitra, A. Zaslavsky, and C. Åhlund, “Context-Aware QoE Modelling , Measurement , and Prediction in Mobile Computing Systems,” *IEEE Trans. Mob. Comput.*, vol. 14, no. 5, pp. 920–936, 2015, doi:10.1109/TMC.2013.155.
- [30] H. Liu, L. T. Yang, M. Lin, D. Yin, and Y. Guo, “A Tensor-Based Holistic Edge Computing Optimization Framework for Internet of Things,” *Edge Comput. Internet Things*, no. February, pp. 88–95, 2018.
- [31] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi:10.1109/COMST.2015.2444095, ISSN:1553877X.
- [32] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, “A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,” *2014 Int. Conf. Sci. Eng. Manag. Res.*, pp. 1–8, 2014, doi:10.1109/ICSEMR.2014.7043637, ISBN: 978-1-4799-7613-3, ISSN:10952055.
- [33] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi:10.1016/j.future.2013.01.010, ISSN:0167739X.
- [34] M. Weiser, R. Gold, and J. S. Brown, “The origins of ubiquitous computing research at PARC,” *IBM Syst. J.*, vol. 38, no. 4, pp. 693–696, 2010, ISSN:0018-8670.
- [35] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, *et al.*, “Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System Keywords — Internet of things, Artificial Neural Network, Denial of Service, Intrusion detection System and Multi-Level Perceptron,” in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 4–9, ISBN:9781509002849.
- [36] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, 2013, doi:10.1016/j.future.2013.01.010, ISSN:0167739X.
- [37] J. Buckley, Ed., *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems*. Auerbach Publications, New York, 2006.
- [38] S. Subashini, R. Venkateswari, and P. Mathiyalagan, *A Study on LoRaWAN for*

- Wireless Sensor Networks*. Springer Singapore, 2018, doi:10.1515/9783110470383ISBN:9789811315138.
- [39] J. A. Rodríguez, “Cognición y ciencia cognitiva,” 2007.
- [40] S. Villalba and R. E. Tortajada, “Estimulación cognitiva: una revisión neuropsicológica,” pp. 73–93, 2014.
- [41] Y. Ginarte-Arias, “Rehabilitación cognitiva. Aspectos teóricos y metodológicos,” *Rev. Neurol.*, vol. 35, no. 9, pp. 870–876, 2002, doi:10.33588/rn.3509.2002418, ISSN:02100010.
- [42] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, *et al.*, “Cognitive internet of things: A new paradigm beyond connection,” *IEEE Internet Things J.*, vol. 1, no. 2, pp. 129–143, 2014, doi:10.1109/JIOT.2014.2311513, ISBN: 2327-4662 VO - 1, ISSN:23274662.
- [43] F. M. Al-Turjman, “Information-centric sensor networks for cognitive IoT: an overview,” *Ann. des Telecommun. Telecommun.*, vol. 72, no. 1–2, pp. 3–18, 2017, doi:10.1007/s12243-016-0533-8, ISSN:19589395.
- [44] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT,” *2018 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2018*, pp. 197–202, 2018, doi:10.1109/PERCOMW.2018.8480255, ISBN: 9781538632277.
- [45] R. Sanchez-Iborra, J. Sanchez-Gomez, and A. Skarmeta, “Evolving IoT networks by the confluence of MEC and LP-WAN paradigms,” *Futur. Gener. Comput. Syst.*, vol. 88, pp. 199–208, 2018, doi:10.1016/j.future.2018.05.057, ISSN:0167739X.
- [46] R. Sanchez-Iborra and M. D. M.-D. Cano, “State of the art in LP-WAN solutions for industrial IoT services,” *Sensors (Switzerland)*, vol. 16, no. 5, 2016, doi:10.3390/s16050708, ISBN: 978-3-03842-370-6, ISSN:14248220.
- [47] “Sigfox,” 2019. [Online]. Available: <https://www.sigfox.com/en>. [Accessed: 09-Sep-2019].
- [48] LoRa Alliance, “LoRaWAN - What is it?. A technical overview of LoRa and LoRaWAN,” no. November, pp. 1–20, 2015.
- [49] “Weightless.,” 2019. [Online]. Available: <http://www.weightless.org/>.
- [50] Ingenu, “RPMA Technology,” 2019. [Online]. Available: <https://www.ingenu.com/technology/rpma/>.
- [51] “Telensa.,” 2019. [Online]. Available: <https://www.telensa.com/technology>.
- [52] “DASH7 Alliance – An open specification.” [Online]. Available: <https://dash7-alliance.org/>. [Accessed: 04-Oct-2020].
- [53] C. Goursaud and J. M. Gorce, “Dedicated networks for IoT: PHY / MAC state of

- the art and challenges,” *EAI Endorsed Trans. Internet Things*, vol. 1, no. 1, p. 150597, 2015, doi:10.4108/eai.26-10-2015.150597.
- [54] “Semtech.” [Online]. Available: www.semtech.com. [Accessed: 26-Jun-2019].
- [55] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of LoRa networks in a smart city scenario,” *IEEE Int. Conf. Commun.*, 2017, doi:10.1109/ICC.2017.7996384, ISBN: 9781467389990, ISSN:15503607.
- [56] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the Limits of LoRaWAN,” *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017, doi:10.1109/MCOM.2017.1600613, ISSN:01636804.
- [57] M. Bor and U. Roedig, “LoRa transmission parameter selection,” *Proc. - 2017 13th Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2017*, vol. 2018-Janua, pp. 27–34, 2018, doi:10.1109/DCOSS.2017.10, ISBN: 9781538639917, ISSN:2325-2944.
- [58] A. Lavric and V. Popa, “A LoRaWAN: Long Range Wide Area Networks study,” in *2017 11th International Conference on Electromechanical and Power Systems, SIELMEN 2017 - Proceedings*, 2017, vol. 2017-Janua, pp. 417–420, doi:10.1109/SIELMEN.2017.8123360, ISBN:9781538618462.
- [59] J. Petäjäljärvi, K. Mikhaylov, A. Roivainen, T. Hänninen, and M. Pettissalo, “On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology,” in *14th International Conference on ITS Telecommunications, ITST 2015*, 2016, pp. 55–59, doi:10.1109/ITST.2015.7377400, ISBN:9781467393829.
- [60] H. H. R. Sherazi, G. Piro, L. A. Grieco, and G. Boggia, “When Renewable Energy Meets LoRa: A Feasibility Analysis on Cable-Less Deployments,” *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5097–5108, 2018, doi:10.1109/JIOT.2018.2839359, ISSN:23274662.
- [61] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “Do LoRa Low-Power Wide-Area Networks Scale?,” *Proc. 19th ACM Int. Conf. Model. Anal. Simul. Wirel. Mob. Syst. - MSWiM '16*, pp. 59–67, 2016, doi:10.1145/2988287.2989163, ISBN: 9781450345026, ISSN:00404020.
- [62] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, “Evaluating LoRa energy efficiency for adaptive networks: From star to mesh topologies,” *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, vol. 2017-Octob, 2017, doi:10.1109/WiMOB.2017.8115793, ISBN: 9781538638392, ISSN:21619654.
- [63] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, “Improving reliability and scalability of LoRaWANs through lightweight scheduling,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1830–1842, 2018, doi:10.1109/JIOT.2018.2815150, ISSN:23274662.
- [64] M. Slabicki, G. Premsankar, and M. Di Francesco, “Adaptive configuration of lora networks for dense IoT deployments,” *IEEE/IFIP Netw. Oper. Manag. Symp. Cogn.*

- Manag. a Cyber World, NOMS 2018*, pp. 1–9, 2018, doi:10.1109/NOMS.2018.8406255, ISBN: 9781538634165.
- [65] H. Linka, M. Rademacher, O. G. Aliu, K. Jonas, B. Rhein-sieg, and S. Augustin, “Path Loss Models for Low-Power Wide-Area Networks : Experimental Results using LoRa,” in *VDE ITG-Fachbericht Mobilkommunikation, Osnabrück*, 2018, no. May, pp. 10–14.
- [66] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, “Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 298–310, 2020, doi:10.1109/JIOT.2019.2946900, ISSN:23274662.
- [67] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Low overhead scheduling of LoRa transmissions for improved scalability,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3097–3109, 2019, doi:10.1109/JIOT.2018.2878942, ISSN:23274662.
- [68] SEMTECH, “SEMTECH SX1272/73 Datasheet,” no. January. 2019.
- [69] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M. D. Cano, and A. F. Skarmeta, “Performance evaluation of lora considering scenario conditions,” *Sensors (Switzerland)*, vol. 18, no. 3, 2018, doi:10.3390/s18030772, ISSN:14248220.
- [70] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016, doi:10.1109/COMST.2016.2560343, ISSN:1553877X.
- [71] J. Sánchez-García, J. M. García-Campos, M. Arzamendia, D. G. Reina, S. L. Toral, and D. Gregor, “A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications,” *Computer Communications*. 2018.
- [72] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013, doi:10.1016/j.adhoc.2012.12.004, ISSN:15708705.
- [73] L. Gupta, R. Jain, and G. Vaszkun, “Survey of Important Issues in UAV Communication Networks,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016, doi:10.1109/COMST.2015.2495297, ISSN:1553877X.
- [74] B. Masini, A. Bazzi, and A. Zanella, “A Survey on the Roadmap to Mandate on Board Connectivity and Enable V2V-Based Vehicular Sensor Networks,” *Sensors*, vol. 18, no. 7, p. 2207, 2018, doi:10.3390/s18072207, ISSN:1424-8220.
- [75] A. Guillen-Perez and M. D. Cano, “Flying ad hoc networks: A new domain for network communications,” *Sensors (Switzerland)*, vol. 18, no. 10, 2018, doi:10.3390/s18103571, ISSN:14248220.

- [76] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, “Comparative study of wireless network simulators,” *Proc. - 7th Int. Conf. Networking, ICN 2008*, pp. 517–523, 2008, doi:10.1109/ICN.2008.97, ISBN: 9780769531069.
- [77] J. Xie, Y. Wan, J. H. Kim, S. Fu, and K. Namuduri, “A survey and analysis of mobility models for airborne networks,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 3, pp. 1221–1238, 2014, doi:10.1109/SURV.2013.111313.00138, ISSN:1553877X.
- [78] X. Hong, M. Gerla, G. Pei, C. Chiang, and L. Angeles, “A group mobility model for ad hoc wireless networks,” *Group*.
- [79] J.-M. Martinez-Caro and M.-D. Cano, “IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study,” *Wirel. Commun. Mob. Comput.*, vol. 2019, p. 12, 2019, doi:10.1155/2019/4307925.
- [80] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, “Feature selection using principal feature analysis,” *Proc. ACM Int. Multimed. Conf. Exhib.*, pp. 301–304, 2007, doi:10.1145/1291233.1291297, ISBN: 9781595937025.
- [81] H. Li, K. Ota, and M. Dong, “Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing,” *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, 2018, doi:10.1109/MNET.2018.1700202, ISSN:08908044.
- [82] F. Al-Turjman and I. Baali, “Machine learning for wearable IoT-based applications: A survey,” *Trans. Emerg. Telecommun. Technol.*, vol. e3635, p. 16, 2019, doi:10.1002/ett.3635, ISSN:21613915.
- [83] J. Xie, F. Richard Yu, T. Huang, R. Xie, J. Liu, C. Wang, *et al.*, “A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 1, pp. 393–430, 2019, doi:10.1109/COMST.2018.2866942, ISSN:1553877X.
- [84] A. Floris and L. Atzori, “Quality of Experience in the Multimedia Internet of Things: Definition and practical use-cases,” in *2015 IEEE International Conference on Communication Workshop, ICCW 2015*, 2015, pp. 1747–1752, doi:10.1109/ICCW.2015.7247433, ISBN:9781467363051, ISSN: 14248220.
- [85] Zaheeruddin, D. K. Lobiyal, and S. Prasad, “Ant based Pareto optimal solution for QoS aware energy efficient multicast in wireless networks,” *Appl. Soft Comput. J.*, vol. 55, pp. 72–81, Jun. 2017, doi:10.1016/j.asoc.2017.01.029, ISSN:15684946.
- [86] Q. Han, B. Yang, C. Chen, and X. Guan, “Energy-aware and QoS-aware load balancing for HetNets powered by renewable energy,” *Comput. Networks*, vol. 94, pp. 250–262, Jan. 2016, doi:10.1016/j.comnet.2015.10.025, ISSN:13891286.
- [87] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, “Toward QoI and energy-efficiency in internet-of-things sensory environments,” *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 4, pp. 473–487, Dec. 2014, doi:10.1109/TETC.2014.2364915,

ISSN:21686750.

- [88] M.-D. Cano, “QoE en el contexto de Internet of Everything,” in *JITEL 2017*, 2017, p. 6.
- [89] L. L. Li, M. Rong, and G. Zhang, “An Internet of Things QoE evaluation method based on multiple linear regression analysis,” *2015 10th Int. Conf. Comput. Sci. Educ. ICCSE*, no. Iccse, pp. 925–928, Jul. 2015, doi:10.1109/ICCSE.2015.7250382, ISBN: 9781479966004.
- [90] Y. Ikeda, S. Kouno, A. Shiozu, and K. Noritake, “A framework of scalable QoE modeling for application explosion in the Internet of Things,” in *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, 2017, pp. 425–429, doi:10.1109/WF-IoT.2016.7845393, ISBN:9781509041305.
- [91] J. Mitola, “Cognitive radio architecture evolution,” *Proc. IEEE*, vol. 97, no. 4, pp. 626–641, Apr. 2009, doi:10.1109/JPROC.2009.2013012, ISBN: 0018-9219, ISSN:0018-9219.
- [92] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, “Modelling and simulation of Opportunistic IoT Services with Aggregate Computing,” *Futur. Gener. Comput. Syst.*, vol. 91, pp. 252–262, Feb. 2019, doi:10.1016/j.future.2018.09.005, ISSN:0167739X.
- [93] “Euskadi air quality (2018),” 2018. [Online]. Available: <http://opendata.euskadi.eus/catalogo/-/calidad-aire-en-euskadi-2018/>. [Accessed: 09-Sep-2019].
- [94] “Introducción a la inteligencia artificial.” [Online]. Available: <https://relopezbriega.github.io/blog/2017/06/05/introduccion-a-la-inteligencia-artificial/>. [Accessed: 17-Sep-2020].
- [95] A. NG, “Introduction to Machine Learning - Coursera.” [Online]. Available: <https://es.coursera.org/learn/machine-learning>. [Accessed: 17-Sep-2020].
- [96] Statista The Statistics Portal, “Video Streaming Services,” 2020. .
- [97] The Economist, “Netflix, Disney and the battle to control eyeballs. Who will win the media wars?,” *The Economist*, Nov-2019.
- [98] X. Tao, Y. Duan, M. Xu, Z. Meng, and J. Lu, “Learning QoE of Mobile Video Transmission with Deep Neural Network: A Data-Driven Approach,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1337–1348, 2019, doi:10.1109/JSAC.2019.2904359, ISSN:15580008.
- [99] M. Seufert, P. Casas, N. Wehner, L. Gang, and K. Li, “Features that Matter: Feature Selection for On-line Stalling Prediction in Encrypted Video Streaming,” *INFOCOM 2019 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPs 2019*, pp. 688–695, 2019, doi:10.1109/INFCOMW.2019.8845109, ISBN: 9781728118789.

- [100] H. F. Bermudez, J. M. Martinez-Caro, R. Sanchez-Iborra, J. L. Arciniegas, and M. D. Cano, "Live video-streaming evaluation using the ITU-T P.1203 QoE model in LTE networks," *Comput. Networks*, vol. 165, 2019, doi:10.1016/j.comnet.2019.106967, ISSN:13891286.
- [101] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A Subjective and Objective Study of Stalling Events in Mobile Streaming Videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 183–197, 2019, doi:10.1109/TCSVT.2017.2768542, ISSN:10518215.
- [102] T. Casey and G. M. Muntean, "Reducing stalling events during DASH video playback in heterogeneous multi-network wireless environments," *IEEE Int. Symp. Broadband Multimed. Syst. Broadcast. BMSB*, pp. 1–6, 2017, doi:10.1109/BMSB.2017.7986139, ISBN: 9781509049370, ISSN:21555052.
- [103] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "A survey on parametric QoE estimation for popular services," *J. Netw. Comput. Appl.*, vol. 77, no. February 2016, pp. 1–17, 2017, doi:10.1016/j.jnca.2016.10.016, ISSN:10958592.
- [104] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," *ITU-T Recomm. P.1203*, 2017.
- [105] H. F. Bermudez, R. Sanchez-Iborra, J. L. Arciniegas, W. Y. Campo, and M. D. Cano, "Statistical validation of an LTE emulation tool using live video streaming over reliable transport protocols," *Telecommun. Syst.*, vol. 71, pp. 491–504, 2019, doi:https://doi.org/10.1007/s11235-018-0521-6.
- [106] ISO/IEC, "ISO/IEC 23009-1:2014 'Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats,'" 2014.
- [107] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 401–418, 2016, doi:10.1109/COMST.2015.2401424, ISBN: 1553-877X, ISSN:1553-877X.
- [108] ITU Telecommunication Standardization Sector, "ITU-T Rec P 1203 Models and tools for quality assessment of streamed media," 2017.
- [109] C. G. Bampis and A. C. Bovik, "Feature-based prediction of streaming video QoE: Distortions, stalling and memory," *Signal Process. Image Commun.*, vol. 68, no. August 2017, pp. 218–228, 2018, doi:10.1016/j.image.2018.05.017, ISSN:09235965.
- [110] S. Wassermann, P. Casas, M. Seufert, and F. Wamser, "On the Analysis of YouTube QoE in Cellular Networks through in-Smartphone Measurements," *Proc. 12th IFIP Wirel. Mob. Netw. Conf. WMNC 2019*, pp. 71–78, 2019, doi:10.23919/WMNC.2019.8881828, ISBN: 9783903176188.

- [111] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan, “BUFFEST: Predicting bufer conditions and real-time requirements of HTTP(S) adaptive streaming clients,” *Proc. 8th ACM Multimed. Syst. Conf. MMSys 2017*, pp. 76–87, 2017, doi:10.1145/3083187.3083193, ISBN: 9781450334891.
- [112] M. H. Mazhar and Z. Shafiq, “Real-time Video Quality of Experience Monitoring for HTTPS and QUIC,” *Proc. - IEEE INFOCOM*, vol. 2018-April, pp. 1331–1339, 2018, doi:10.1109/INFOCOM.2018.8486321, ISBN: 9781538641286, ISSN:0743166X.
- [113] “Wireshark.” [Online]. Available: <https://www.wireshark.org/>. [Accessed: 04-Feb-2020].
- [114] Y. Sun, A. K. C. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009, doi:10.1142/S0218001409007326, ISSN:02180014.
- [115] H. He and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, pp. 1263–1285, 2009, doi:10.1109/TKDE.2008.239.
- [116] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016, doi:10.1007/s13748-016-0094-0, ISSN:2192-6352.
- [117] S. Vluymans, *Dealing with Imbalanced and Weakly Labelled Data in Machine Learning using Fuzzy and Rough Set Methods*, vol. 807. Cham: Springer International Publishing, 2019, doi:10.1007/978-3-030-04663-7ISBN:978-3-030-04662-0.
- [118] “API Reference — scikit-learn 0.22.1 documentation.” [Online]. Available: <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing>. [Accessed: 04-Feb-2020].
- [119] L. Puggini and S. McLoone, “Feature Selection for Anomaly Detection Using Optical Emission Spectroscopy,” *IFAC-PapersOnLine*, 2016, doi:10.1016/j.ifacol.2016.07.102, ISSN:24058963.
- [120] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2^o Edition. Packt Publishing Ltd., 2017ISBN:978-1-78712-593-3.
- [121] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *J. Big Data*, vol. 6, no. 1, 2019, doi:10.1186/s40537-019-0192-5, ISSN:2196-1115.
- [122] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi:10.1162/neco.1997.9.8.1735, ISSN:08997667.
- [123] V. García, J. S. Sánchez, and R. A. Mollineda, “On the effectiveness of preprocessing methods when dealing with different levels of class imbalance,”

- Knowledge-Based Syst.*, vol. 25, no. 1, pp. 13–21, 2012, doi:10.1016/j.knosys.2011.06.013, ISSN:09507051.
- [124] Y. Cao, F. Yang, Q. Tang, and X. Lu, “An attention enhanced bidirectional LSTM for early forest fire smoke recognition,” *IEEE Access*, vol. 7, pp. 154732–154742, 2019, doi:10.1109/ACCESS.2019.2946712, ISSN:21693536.
- [125] A. N. Sokolov, S. K. Alabugin, and I. A. Pyatnitsky, “Traffic modeling by recurrent neural networks for intrusion detection in industrial control systems,” *2019 Int. Conf. Ind. Eng. Appl. Manuf. ICIEAM 2019*, pp. 1–5, 2019, doi:10.1109/ICIEAM.2019.8742961, ISBN: 9781538681190.
- [126] S. Dai, L. Li, and Z. Li, “Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction,” *IEEE Access*, vol. 7, pp. 38287–38296, 2019, doi:10.1109/ACCESS.2019.2907000, ISBN: 2017081809, ISSN:21693536.
- [127] M. Wang, Z. Wang, J. Lu, J. Lin, and Z. Wang, “E-LSTM: An Efficient Hardware Architecture for Long Short-Term Memory,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 2, pp. 280–291, 2019, doi:10.1109/JETCAS.2019.2911739, ISBN: 0210143800, ISSN:21563357.
- [128] M. Suryanegara, D. A. Prasetyo, F. Andriyanto, and N. Hayati, “A 5-Step Framework for Measuring the Quality of Experience (QoE) of Internet of Things (IoT) Services,” *IEEE Access*, vol. 7, pp. 175779–175792, 2019, doi:10.1109/ACCESS.2019.2957341, ISSN:21693536.
- [129] M. Aloqaily, H. Bany Salameh, I. Al Ridhawi, K. Batiha, and J. Ben Othman, “A multi-stage resource-constrained spectrum access mechanism for cognitive radio IoT networks: Time-spectrum block utilization,” *Futur. Gener. Comput. Syst.*, vol. 110, pp. 254–266, 2020, doi:10.1016/j.future.2020.04.022, ISSN:0167739X.
- [130] I. Sousa, M. P. Queluz, and A. Rodrigues, “A survey on QoE-oriented wireless resources scheduling,” *J. Netw. Comput. Appl.*, vol. 158, no. December 2019, p. 102594, 2020, doi:10.1016/j.jnca.2020.102594, ISSN:10958592.
- [131] D. Minovski, C. Ahlund, and K. Mitra, “Modeling Quality of IoT Experience in Autonomous Vehicles,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3833–3849, 2020, doi:10.1109/JIOT.2020.2975418, ISSN:23274662.
- [132] S. Schwarzmann, C. C. Marquezan, R. Trivisonno, S. Nakajima, and T. Zinner, “Accuracy vs. Cost Trade-off for Machine Learning Based QoE Estimation in 5G Networks,” *IEEE Int. Conf. Commun.*, vol. 2020-June, pp. 1–6, 2020, doi:10.1109/ICC40277.2020.9148685, ISBN: 9781728150895, ISSN:15503607.
- [133] B. Shubyn, N. Lutsiv, O. Syrotynskyi, and R. Kolodii, “Deep Learning based Adaptive Handover Optimization for Ultra-Dense 5G Mobile Networks,” *Proc. - 15th Int. Conf. Adv. Trends Radioelectron. Telecommun. Comput. Eng. TCSET 2020*, pp. 869–872, 2020, doi:10.1109/TCSET49122.2020.235560, ISBN: 9781728155661.

- [134] B. D. Deebak, F. Al-Turjman, M. Aloqaily, and O. Alfandi, “IoT-BSFCAN: A smart context-aware system in IoT-Cloud using mobile-fogging,” *Futur. Gener. Comput. Syst.*, vol. 109, pp. 368–381, 2020, doi:10.1016/j.future.2020.03.050, ISSN:0167739X.
- [135] P. Casas, A. D’Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, *et al.*, “Predicting QoE in cellular networks using machine learning and in-smartphone measurements,” *2017 9th Int. Conf. Qual. Multimed. Exp. QoMEX 2017*, vol. 02152, pp. 3–8, 2017, doi:10.1109/QoMEX.2017.7965687, ISBN: 9781538640241.
- [136] T. Sui, X. Tao, S. Xia, H. Chen, H. Wu, X. Zhang, *et al.*, “A Real-Time Hidden Anomaly Detection of Correlated Data in Wireless Networks,” *IEEE Access*, vol. 8, pp. 60990–60999, 2020, doi:10.1109/ACCESS.2020.2984276, ISSN:21693536.
- [137] M. Lopez-Martin, B. Carro, J. Lloret, S. Egea, and A. Sanchez-Esguevillas, “Deep Learning Model for Multimedia Quality of Experience Prediction Based on Network Flow Packets,” *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 110–117, 2018, doi:10.1109/MCOM.2018.1701156, ISSN:15581896.
- [138] E. Ibarrola, M. Davis, C. Voisin, C. Close, and L. Cristobo, “QoE Enhancement in Next Generation Wireless Ecosystems : A Machine Learning Approach,” *IEEE Commun. Stand. Mag.*, vol. 3, no. September 2019, pp. 63–70, 2020, doi:10.1109/MCOMSTD.001.1900001.