

**Centro de Investigación Científica y de Educación
Superior de Ensenada, Baja California**



**Maestría en Ciencias
en Electrónica y Telecomunicaciones
con orientación en Telecomunicaciones**

**Predicción del comportamiento de SLA en redes 5G utilizando
inteligencia artificial**

Tesis

Para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Rubersy Ramos García

Ensenada, Baja California, México
2021

Tesis defendida por
Rubersy Ramos García

y aprobada por el siguiente Comité

Dr. Jorge Enrique Preciado Velasco
Codirector de tesis

Dr. José Eleno Lozano Rizk
Codirector de tesis

Miembros del comité

Dr. Raúl Rivera Rodríguez

Dr. Miguel Ángel Alonso Arévalo

Dr. Jorge Torres Rodríguez



Dra. María del Carmen Maya Sánchez
Coordinadora del Posgrado en Electrónica y Telecomunicaciones

Dr. Pedro Negrete Reganon
Director de Estudios de Posgrado

Rubersy Ramos García © 2021

Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis.

Resumen de la tesis que presenta **Rubersy Ramos García** como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones.

Predicción del comportamiento de SLA en redes 5G utilizando inteligencia artificial

Resumen aprobado por:

Dr. Jorge Enrique Preciado Velasco
Codirector de tesis

Dr. José Eleno Lozano Rizk
Codirector de tesis

El objetivo de la tecnología inalámbrica 5G es ser una plataforma unificada capaz de soportar tres servicios genéricos, heterogéneos y con características de tráfico diferentes que exigen requisitos estrictos y específicos de la red móvil. Estos servicios son: comunicación masiva tipo máquina; comunicación ultra confiable y de baja latencia; y banda ancha móvil mejorada. La tecnología 5G permitirá a los usuarios nuevas aplicaciones incluyendo: realidad aumentada, realidad virtual, internet de las cosas, vehículos autónomos y muchas más. Para cumplir con los requisitos divergentes de los diferentes tipos de servicios, las redes 5G deben admitir la segmentación, y para ello se utilizan tecnologías emergentes como la Virtualización de Funciones de Red y las Redes Definidas por Software, las cuales son consideradas componentes clave de las infraestructuras 5G. Las diversas características y la complejidad que definen estas redes requieren de flexibilidad y dinamismo, haciéndose imprescindible que sean capaces de determinar autónomamente la configuración óptima del sistema con la mínima intervención humana. En esta tesis de maestría se propone un modelo para la predicción de posibles violaciones de los SLA en redes 5G. Como parte del modelo propuesto se diseñan e implementan dos arquitecturas de redes neuronales artificiales: una red recurrente con arquitectura codificador-decodificador, y una red neuronal mixta compuesta por la unión de una red convolucional y una red recurrente. Dentro de los resultados está la predicción del *throughput* de la red 5G para una aplicación del segmento eMBB, considerando los indicadores de desempeño y los factores que impactan de manera más significativa al *throughput* en 5G; y la creación de un sistema de predicción de la calidad del servicio que garantice el cumplimiento de los SLA.

Palabras clave: 5G, aprendizaje automático, *throughput*, SLA.

Abstract of the thesis presented by **Rubersy Ramos García** as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications.

Predicting the behavior of SLAs in 5G networks using artificial intelligence

Abstract approved by:

Dr. Jorge Enrique Preciado Velasco
Codirector de tesis

Dr. José Eleno Lozano Rizk
Codirector de tesis

The objective of 5G wireless technology is to be a unified platform capable of supporting three generic, heterogeneous services with different traffic characteristics that demand strict and specific requirements of the mobile network. These services are massive Machine-Type Communication; ultra-reliable Low Latency Communication; and enhanced Mobile Broadband. 5G technology will allow users new applications including augmented reality, virtual reality, internet of things, autonomous vehicles and many more. To meet the divergent requirements of different types of services, 5G networks must support segmentation, using emerging technologies such as Network Functions Virtualization and Software Defined Networks, which are considered key components of 5G infrastructures. The various characteristics and complexity that define these networks require flexibility and dynamism, making it essential that they be capable of autonomously determining the optimal configuration of the system with minimal human intervention. This master's thesis proposes a model for predicting possible SLA violations in 5G networks. As part of the proposed model, two artificial neural network architectures are designed and implemented: a recurrent network with an encoder-decoder architecture, and a mixed neural network composed of the union of a convolutional network and a recurrent network. Among the results is the prediction of the throughput of the 5G network for an application in the eMBB segment, considering the performance indicators and the factors that most significantly impact throughput in 5G, and the creation of a service quality prediction system that guarantees compliance with SLAs.

Keywords: 5G, machine learning, throughput, SLA

Dedicatoria

A mi amado hijo Gabriel, mi príncipe enano, mi tesoro, mi reyecillo...

A mi dulce e inquieta abuela Lutgarda, de quién no pude despedirme por estar lejos. Nunca te olvido, siempre te llevo conmigo...

Agradecimientos

A Dios, por darme salud y fuerzas para seguir adelante con este proyecto en momentos tan difíciles de pandemia mundial.

A mi director de tesis, el Dr. Jorge Preciado, y a mi codirector, el Dr. José Lozano, por su constante y acertada orientación, su apoyo incondicional y su confianza.

A los miembros del comité de tesis, al Dr. Raúl Rivera, por su generosa orientación, acertadas correcciones y retroalimentación sobre el trabajo de investigación. A los doctores Miguel Alonso y Jorge Torres por sus oportunas correcciones, recomendaciones y apoyo.

A los profesores del CICESE, por todos los conocimientos transmitidos durante estos dos años, así como por su disponibilidad y ayuda prestada en todo momento.

A todos los estudiantes y personal del CICESE, en especial a los miembros del departamento de Electrónica y Telecomunicaciones, por su apoyo y amistad.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por brindarme el apoyo económico para realizar mis estudios de maestría. No. CVU: 1014621.

A México y a su pueblo, por acogerme durante estos dos años y brindarme esta gran oportunidad.

Tabla de contenido

Resumen en español	ii
Resumen en inglés	iii
Dedicatoria	iv
Agradecimientos	v
Lista de figuras.....	ix
Lista de tablas.....	xiii
Glosario de términos	xiv
Capítulo 1. Introducción.....	1
1.1 Planteamiento del problema	4
1.2 Justificación	4
1.3 Hipótesis.....	5
1.4 Objetivos	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5
1.5 Estructura de la tesis.....	6
Capítulo 2. Marco de referencia del proyecto	7
2.1 Arquitectura de la segmentación de red en 5G	7
2.2 Aspectos operacionales de la segmentación de red.....	11
2.3 Acuerdos de nivel de servicio en redes 5G	14
2.3.1 Aseguramiento de los SLA y la QoS.....	15
2.4 Requerimientos de desempeño.....	18
2.5 Análisis de los factores de impacto sobre el <i>throughput</i> en 5G	20
2.5.1 Impacto de la geolocalización y de los factores geométricos.....	22
2.5.2 Impacto del medio ambiente	24
2.5.3 Impacto de la movilidad y el <i>handoff</i>	24
2.6 Aplicación del aprendizaje automático a la gestión de redes 5G	29
2.6.1 Aprendizaje supervisado	32
2.6.2 Aprendizaje no supervisado	33
2.6.3 Aprendizaje por refuerzo	33
2.7 Redes neuronales artificiales	34

2.7.1 Redes neuronales prealimentadas.....	38
2.7.2 Redes neuronales recurrentes	38
2.7.3 Redes neuronales convolucionales	41
2.8 Conclusiones	44
Capítulo 3. Propuesta del modelo de predicción	45
3.1 Requisitos para la predicción del <i>throughput</i>	45
3.1.1 Selección del <i>dataset</i>	45
3.1.2 Selección de los escenarios	47
3.1.3 Selección de los KPI	48
3.1.4 Selección, agrupamiento y combinación de los factores de impacto.....	52
3.1.5 Selección de la aplicación y del patrón de movilidad	53
3.2 Predicción del <i>throughput</i> utilizando redes neuronales.....	55
3.2.1 Propuesta de solución utilizando una RNN.....	56
3.2.2 Propuesta de solución utilizando una red neuronal mixta CNN-RNN	69
3.3 Premisas para la predicción de violaciones de SLA en el segmento eMBB a partir de la predicción del <i>throughput</i>	73
3.3.1 Premisas para la predicción	73
3.3.2 Función de costo por error de predicción.....	75
3.4 Combinación de la predicción del <i>throughput</i> y del VHO para prevenir violaciones de los SLA en escenarios de alta movilidad.....	77
3.5 Diagrama general del modelo de predicción de violaciones de SLA	79
3.6 Conclusiones	80
Capítulo 4. Validación del modelo propuesto	82
4.1 Marco de la simulación	82
4.2 Resultados de las simulaciones para la predicción del <i>throughput</i>	84
4.3 Resultados de las simulaciones para la predicción del <i>handoff</i> vertical.....	87
4.4 Predicción de las violaciones de los SLA	89
4.5 Conclusiones	96
Capítulo 5. Conclusiones y Recomendaciones	97
5.1 Contribuciones al conocimiento	98
5.2 Limitaciones de la investigación.....	99
5.3 Trabajo futuro	99

Literatura citada	101
Anexos	113
Anexo A. Detalles de los modelos de ML propuestos.....	113
Anexo B. Análisis combinado a partir de la predicción del <i>throughput</i> y del VHO.....	118

Lista de figuras

Figura 1. Escenarios de uso previstos para las redes 5G (ITU-R, 2015).	1
Figura 2. Segmentación de una red de comunicación 5G (Kurtz et al., 2018).....	3
Figura 3. Asignación entre dispositivos, segmento de acceso y segmento CN en una red 5G segmentada (5gAmericas, 2016).....	4
Figura 4. Arquitectura NFV extendida con componentes para brindar la solución de segmentación de la red 5G (Kurtz et al., 2018).	8
Figura 5. Separación del plano de control y del plano de usuario en el CN de una red 5G (5G Americas, 2016).	9
Figura 6. Compartición de recursos de radio entre segmentos de red utilizando FDM y TDM (5G Americas, 2016).	11
Figura 7. Gestión de los segmentos de red (5G Americas, 2016).	12
Figura 8. Mapeo entre la instancia del servicio y la instancia del segmento de red (5G Americas, 2016).	13
Figura 9. Estructura de un SLA (García Rodríguez, 2007).	15
Figura 10. Ciclo de vida de los SLA en 5G (Kapassa et al., 2019).	16
Figura 11. Posicionamiento del Gestor de SLA en un sistema 5G con <i>network slicing</i> (Papageorgiou et al., 2020).	17
Figura 12. Modos de despliegue de redes 5G: modo no autónomo (izquierda) y modo autónomo (derecha) (Narayanan, Ramadan, Carpenter, et al., 2020).....	20
Figura 13. Ángulo de posición entre el UE y el panel 5G (Narayanan, Ramadan, Mehta, et al., 2020).....	23
Figura 14. Ángulo de movilidad entre el UE y el panel 5G (Narayanan, Ramadan, Mehta, et al., 2020). ..	23
Figura 15. Tipos de <i>handoff</i> en redes heterogéneas (Zenalden et al., 2017).	26
Figura 16. Continuidad del servicio de voz entre redes 5G, 4G y 3G (Samsung, 2021).	28
Figura 17. Arquitecturas 5G NSA / SA y potenciales caminos de migración para el inter-funcionamiento entre 4G y 5G (G. Liu et al., 2020).	28
Figura 18. Diagrama de Venn de la relación entre inteligencia artificial, aprendizaje automático y el aprendizaje profundo, así como su aplicación en redes 5G y futuras (Morocho-Cayamcela et al., 2019).	31
Figura 19. Comparación de la programación tradicional con los tres tipos de aprendizaje automático: (a) programación tradicional, (b) aprendizaje supervisado, (c) aprendizaje no supervisado y (d) aprendizaje por refuerzo (Morocho-Cayamcela et al., 2019).	32

Figura 20. Modelo del perceptrón (Alvarez, 2018).....	35
Figura 21. Ejemplo de arquitectura MLP totalmente conectada con 1 capa oculta y 5 neuronas ocultas (Trinh, 2020).....	35
Figura 22. Funciones de activación <i>step</i> , <i>sigmoid</i> , <i>tanh</i> , ReLU y sus derivadas(Géron, 2019).	37
Figura 23. Arquitectura RNN expandida (Olah, 2015).	39
Figura 24. Unidad LSTM estándar (Trinh, 2020).	40
Figura 25. Arquitectura de una CNN (MathWorks, 2021).....	42
Figura 26. Convolución 2D utilizando un kernel 3x3 (Roman, 2020).....	43
Figura 27. Capacidades claves para los servicios 5G genéricos (ITU-R, 2015).....	49
Figura 28. Posibles combinaciones de CU,DU y RU en la arquitectura NG-RAN propuesta por la ITU-T (5G Americas, 2020).....	51
Figura 29. Diagrama de alto nivel de la solución de ML utilizando una RNN para la predicción del throughput (Kousias et al., 2020).....	56
Figura 30. División del dataset en datos de entrenamiento, validación y prueba (elaboración propia)....	59
Figura 31. Ejemplo de una ventana para realizar una predicción en el futuro (t=47) a partir de un historial de 24 unidades de tiempo (TensorFlow Team, 2019).....	60
Figura 32. Representación de las funciones MAE (izquierda) y MSE (derecha) y diferencias en los valores del gradiente para cada punto (Grover, 2018).....	62
Figura 33. Representación gráfica del método holdout cross-validation (Cochrane, 2018).	64
Figura 34. Representación gráfica del método nested cross-validation (Cochrane, 2018).....	64
Figura 35. RNN con celdas LSTM en una arquitectura <i>Encoder-Decoder</i> para la predicción del <i>throughput</i> (elaboración propia).	68
Figura 36. Diagrama de alto nivel de la solución de ML utilizando una red mixta CNN-RNN para la predicción del throughput (Kousias et al., 2020).	70
Figura 37. Red CNN-RNN con celdas Conv1D, LSTM y BiLSTM para la predicción del <i>throughput</i> (elaboración propia).	72
Figura 38. Representación gráfica de la función de costo asociada a los errores de predicción (Bega et al., 2019).	76
Figura 39. Diagrama general del modelo de predicción de violaciones de SLA (elaboración propia).....	80
Figura 40. Arquitectura de <i>TensorFlow</i> (TensorFlow Team, 2017).....	82
Figura 41. Representación gráfica de tres tensores (Analytics Vidhya, 2017).....	83

Figura 42. Resultados obtenidos en la predicción del throughput (métrica MAE) (elaboración propia)...	85
Figura 43. Resultados obtenidos en la predicción del throughput (métrica RMSE) (elaboración propia).	86
Figura 44. Resultados obtenidos en la predicción del VHO (métrica MAE) (elaboración propia).....	88
Figura 45. Resultados obtenidos en la predicción del VHO (métrica RMSE) (elaboración propia).....	88
Figura 46. Predicción del throughput (elaboración propia).....	89
Figura 47. Aplicación de umbrales máximos al throughput. Escenarios: Zona Suburbana, Zona de Downtown y Zona de interiores (elaboración propia).....	90
Figura 48. Aplicación de umbrales de decisión al throughput para el escenario Zona Suburbana (elaboración propia).....	91
Figura 49. Consola del sistema de predicción de violaciones de SLA. En este caso se muestra el pronóstico de las violaciones de los SLA (fila roja) para el escenario Zona Suburbana (elaboración propia).	93
Figura 50. Aplicación de umbrales de decisión al throughput para el escenario Zona de Downtown (elaboración propia).....	94
Figura 51. Consola del sistema de predicción de violaciones de SLA. En este caso se muestra el pronóstico de las violaciones de los SLA (fila roja) para el escenario Zona de Downtown (elaboración propia).	94
Figura 52. Predicción del handoff vertical (elaboración propia).....	95
Figura 53. Aplicación de los umbrales al handoff vertical (elaboración propia).	95
Figura 54. Resumen del modelo RNN con arquitectura Encoder-Decoder de una capa (RNN-E1D1) (elaboración propia).....	113
Figura 55. Detalles de las capas de la RNN con arquitectura Encoder-Decoder de una capa (RNN-E1D1) (elaboración propia).....	113
Figura 56. Resumen del modelo RNN con arquitectura Encoder-Decoder de dos capas (RNN-E2D2) (elaboración propia).....	114
Figura 57. Detalles de las capas de la RNN con arquitectura Encoder-Decoder de dos capas (RNN-E2D2) (elaboración propia).....	114
Figura 58. Resumen del modelo RNN con arquitectura Encoder-Decoder de tres capas (RNN-E3D3) (elaboración propia).....	115
Figura 59. Detalles de las capas de la RNN con arquitectura Encoder-Decoder de tres capas (RNN-E3D3) (elaboración propia).....	115
Figura 60. Resumen del modelo de red mixta CNN-RNN (elaboración propia).	116
Figura 61. Detalles de las capas de la red mixta CNN-RNN (elaboración propia).....	117

Figura 62. Predicción del throughput y del VHO para el análisis combinado. Ejemplo del aumento brusco del throughput debido a un VHO 4G→5G (combinación de A con C). Ejemplo de fluctuación del throughput sin ocurrencia de VHO (combinación de B con D) (elaboración propia). 118

Figura 63. Predicción del throughput y del VHO para el análisis combinado. Ejemplo del aumento y disminución brusca del throughput debido a un VHO del tipo ping pong (combinación de A con C). Ejemplo de fluctuación del throughput sin ocurrencia de VHO (combinación de B con D) (elaboración propia)..... 118

Lista de tablas

Tabla 1. Escenarios e indicadores de desempeño de varios servicios 5G (elaboración propia).....	19
Tabla 2. Valores de los KPI para los escenarios <i>Urban Macro</i> y <i>Dense Urban</i> (elaboración propia).....	20
Tabla 3. Bandas de frecuencias de operación de las redes 5G (elaboración propia).	21
Tabla 4. Análisis del desempeño del handoff en diferentes escenarios (G. Liu et al., 2020).	29
Tabla 5. Algoritmos de aprendizaje supervisado y su aplicación en 5G (elaboración propia).	33
Tabla 6. Algoritmos de aprendizaje no supervisado y su aplicación en 5G (elaboración propia).	33
Tabla 7. Algoritmos de aprendizaje por refuerzo y su aplicación en 5G (elaboración propia).....	34
Tabla 8. Fuente de los datos utilizados en las investigaciones previas (elaboración propia).....	46
Tabla 9. <i>Throughput</i> promedio y rango de variación para diferentes patrones de movilidad para las aplicaciones <i>file download</i> y <i>streaming</i> (Raca, Leahy, et al., 2020).....	54
Tabla 10. Bibliotecas y entornos de desarrollo de ML utilizados (elaboración propia).....	57
Tabla 11. Valores de los hiperparámetros del modelo RNN (elaboración propia).....	66
Tabla 12. Otros parámetros del modelo RNN (elaboración propia).....	67
Tabla 13. Valores de los hiperparámetros del modelo mixto CNN-RNN (elaboración propia).	70
Tabla 14. Otros parámetros del modelo mixto CNN-RNN (elaboración propia).	71
Tabla 15. Combinación de la predicción del <i>throughput</i> y del <i>handoff vertical</i> para prevenir violaciones de SLA en escenarios de alta movilidad (elaboración propia).	79
Tabla 16. Resultados obtenidos en la predicción del <i>throughput</i> utilizando varios modelos de redes neuronales (métrica MAE) (elaboración propia).....	84
Tabla 17. Resultados obtenidos en la predicción del <i>throughput</i> utilizando varios modelos de redes neuronales (métrica RMSE) (elaboración propia).....	85
Tabla 18. Resultados obtenidos en la predicción del VHO utilizando varios modelos de redes neuronales (métrica MAE) (elaboración propia).....	87
Tabla 19. Resultados obtenidos en la predicción del VHO utilizando varios modelos de redes neuronales (métrica RMSE) (elaboración propia).	87

Glosario de términos

#

3G	<i>3rd Generation</i>	3ra Generación
3GPP	<i>Third Generation Partnership</i>	Proyecto de Asociación de 3ra Generación
4G	<i>4th Generation</i>	4ta Generación
5G	<i>5th Generation</i>	5ta Generación
5GC	<i>5G Core Network</i>	Núcleo de la red 5G
5G mmWave	<i>5G millimeter wave</i>	Red 5G que utiliza la tecnología de radio de onda milimétrica
5G NR	<i>5G New Radio</i>	5G Nuevo Radio
5G NSA	<i>5G Non-Standalone</i>	5G modo no autónomo
5G SA	<i>5G Standalone</i>	5G modo autónomo

A

AI	<i>Artificial Intelligence</i>	Inteligencia Artificial
ANN	<i>Artificial Neural Networks</i>	Redes Neuronales Artificiales
API	<i>Application Programming Interface</i>	Interfaz de Programación de Aplicaciones
AR	<i>Augmented Reality</i>	Realidad Aumentada

B

<i>Backhaul</i>		Red de retorno (enlace intermedio entre la red de acceso y la red de núcleo)
<i>Best effort</i>		Mecanismo de mejor esfuerzo
<i>Big Data</i>		Datos masivos
BiLSTM	<i>Bilateral LSTM</i>	LSTM bilateral
BOA	<i>Bayesian Optimization</i>	Optimización bayesiana
<i>Broadcast</i>		Difusión amplia
BS	<i>Batch Size</i>	Tamaño del lote
<i>Buffer</i>		Espacio de memoria en el que se almacenan datos de manera temporal

C

CN	<i>Core Network</i>	Red de núcleo
CNN	<i>Convolutional Neural Networks</i>	Redes neuronales convolucionales
Conv1D	<i>Convolutional 1D Cell</i>	Celda convolucional de una dimensión
Conv2D	<i>Convolutional 2D Cell</i>	Celda convolucional de dos dimensiones
Conv3D	<i>Convolutional 3D Cell</i>	Celda convolucional de tres dimensiones
CQI	<i>Channel Quality Indicator</i>	Indicador de calidad de canal

D

D2D	<i>Device-to-Device communication</i>	Comunicaciones dispositivo a dispositivo
<i>Dataset</i>		Conjunto de datos
<i>Deep Learning</i>		Aprendizaje profundo
DL	<i>Down Link</i>	Enlace Descendente

E

E1D1	<i>One-layer Encoder-Decoder</i>	Arquitectura codificador-decodificador de una capa
E2D2	<i>Two-layer Encoder-Decoder</i>	Arquitectura codificador-decodificador de dos capas
E3D3	<i>Three-layer Encoder-Decoder</i>	Arquitectura codificador-decodificador de tres capas
eMBB eNodeB	<i>Enhanced Mobile Broadband Evolved Node B</i>	Banda Ancha Móvil Mejorada Nodo B evolucionado (radio base de la red LTE)
EPC Epochs	<i>Evolved Packet Core</i>	Núcleo de la red LTE Hiperparámetro que define el número de veces que el algoritmo de ML funcionará en todo el conjunto de datos de entrenamiento.
EPS	<i>Evolved Packet System</i>	Sistema de paquetes evolucionados (red LTE). Se divide en dos partes: E-UTRAN y EPC.
EPS fallback ETSI	<i>Evolved Packet System fallback European Telecommunications Standards Institute</i>	Respaldo de EPS Instituto Europeo de Normas de Telecomunicaciones
E-UTRAN	<i>Evolved UMTS Terrestrial Radio Access Network</i>	Red de Acceso Radio Terrestre UMTS evolucionada

F

FDM	<i>Frequency-division multiplexing</i>	Multicanalización por división de frecuencia
FFN	<i>Feed Forward Networks</i>	Redes neuronales prealimentadas

G

gNodeB	<i>5G Base Station</i>	Estación base en 5G
GPU	<i>Graphics processing unit</i>	Unidad de procesamiento gráfico
GRU	<i>Gated recurrent units</i>	Celda recurrente con compuerta
GS	<i>Grid Search</i>	Búsqueda de cuadrícula

H

Handoff		Proceso de transferencia de una llamada o sesión de datos en curso desde un canal conectado a la red central a otro canal.
HetNets	<i>Heterogeneous network</i>	Red heterogénea
HHO	<i>Horizontal handoff</i>	Handoff horizontal
HL	<i>Hidden layers</i>	Capas ocultas

I

IEEE	<i>Institute of Electrical and Electronics Engineers</i>	Instituto de Ingeniería Eléctrica y Electrónica
IMT-2020	<i>International Mobile Telecommunications 5th Generation</i>	Telecomunicaciones Móviles Internacionales de 5ta Generación
IoT	<i>Internet of Things</i>	Internet de las Cosas
ITU	<i>International Telecommunications Union</i>	Unión Internacional de Telecomunicaciones

J

Jitter Fluctuación del retardo (Variabilidad del tiempo de ejecución de los paquetes)

K

KNN *k-Nearest Neighbors* k-Vecinos más Cercanos
 KPI *Key Performance Indicators* Indicadores Clave de Desempeño
 KQI *Key Quality Indicators* Indicadores Clave de Calidad

L

LoS *Line-of-sight* Línea de visión
 LR *Learning rate* Tasa de aprendizaje
 LSTM *Long Short Term Memory* Memoria de largo y corto plazo (Tipo particular de red recurrente)
 LTE *Long-Term Evolution* Evolución a largo plazo
 LTE Advanced *Long-Term Evolution Advanced* Evolución a largo plazo avanzada
 LTE eNB *LTE Base Station (eNodeB)* Estación base en LTE

M

MAE *Mean Absolute Error* Error absoluto medio
 MC *Memory cell* Celdas de memoria
 MIMO *Multiple-input and multiple-output* Múltiples entradas y múltiples salidas
 ML *Machine Learning* Aprendizaje automático o aprendizaje de máquinas
 mMTC *Massive Machine Type Communication* Comunicación Masiva Tipo Máquina
 MLP *Multilayer perceptron* Perceptrón multicapa
 MS *Manual Search* Búsqueda manual
 MSE *Mean Square Error* Error cuadrático medio
 MTBF *Mean time between failures* Tiempo medio entre fallos
 MTTR *Mean time to repair* Tiempo medio de reparación

N

Network slicing Segmentación de red
 NGMN *Next Generation Mobile Networks* Redes Móviles de Próxima Generación
 NF *Network Function* Función de Red
 NFV *Network Function Virtualization* Virtualización de Funciones de Red
 NFVO *NFV orchestrator* Orquestador de NFV
 NG-RAN *New Generation RAN* RAN de próxima generación
 NLoS *Non-line-of-sight* Sin Línea de Visión
 NMO *Network Management and Orchestration plane* Plano de orquestación y gestión de red
 NPL *Natural language processing* Procesamiento de lenguaje natural
 NR *New Radio technology* Tecnología Nuevo Radio
 NR gNB *New Radio gNodeB* Estación base en 5G
 NS *Network Slice* Segmento de red (red virtual)

O*Open Source*

Código abierto. Es un término que denota que un producto incluye permiso para usar su código fuente

Q

QoS

Quality of Service

Calidad de Servicio

QoE

Quality of Experience

Calidad de Experiencia

R

RAN

Radio Access Network

Red de Acceso de Radio

RAT

Radio Access Technologies

Tecnología de Acceso por Radio

ReLU

Rectified Linear Unit

Unidad lineal rectificada

RMSE

Root Mean Square Error

Raíz del error cuadrático medio

RNN

Recurrent Neural Networks

Red neuronal recurrente

RS

Random Search

Búsqueda aleatoria

RSRP

Reference Signals Received Power

Potencia recibida de la señal de referencia

RSRQ

Reference Signal Received Quality

Calidad de la señal de referencia recibida

RSSI

Received Signal Strength Indicator

Indicador de intensidad de señal recibida

S

SBA

Service Based Architecture

Arquitectura Basada en Servicios

SDN

Software Defined Networks

Redes Definidas por Software

SLA

Service Level Agreement

Acuerdo de Nivel de Servicio

SLO

Service Level Objectives

Objetivo de Nivel de Servicio

SLS

Service Level Specifications

Especificaciones del nivel de servicio

SNR

Signal-to-noise ratio

Relación señal/ruido

SRVCC

Single Radio Voice Call Continuity

Continuidad de llamadas de voz de radio única

Streaming

Tecnología que permite transmitir archivos de audio y video en un flujo continuo a través de una conexión a Internet alámbrica o inalámbrica

SVM

Support vector machines

Máquinas de vectores de soporte

T

TDM

Time-division multiplexing

Multicanalización por división de tiempo

Throughput

Rendimiento de la red. Es la tasa de entrega exitosa de mensajes a través de un canal de comunicación

TPU

Tensor Processing Unit

Unidad de procesamiento tensorial

U

UE

User Equipment

Equipo de Usuario

UMTS

Universal Mobile

Sistema universal de telecomunicaciones móviles

*Uptime**Telecommunications System*

Tiempo de actividad. Es una medida de la confiabilidad del sistema, expresada como el porcentaje de tiempo que ha estado funcionando

urLLC	<i>Ultra-reliable and Low Latency Communication</i>	Comunicaciones UltraConfiables y de Baja Latencia
UTRAN	<i>UMTS Terrestrial Radio Access Network</i>	Red de Acceso Radio Terrestre UMTS
V		
VHO	<i>Vertical handoff</i>	<i>Handoff</i> vertical
VNF	<i>Virtual network functions</i>	Funciones de red virtual
VoLTE	<i>Voice over LTE</i>	Voz sobre LTE
VoNR	<i>Voice over NR</i>	Voz sobre NR
VR	<i>Virtual Reality</i>	Realidad Virtual

Capítulo 1. Introducción

El objetivo de la tecnología inalámbrica 5G propuesta en el estándar IMT-2020 (Telecomunicaciones Móviles Internacionales de 5ta. generación) es que sea una red enfocada a la prestación de servicios; que sea una plataforma unificada, capaz de soportar tres servicios genéricos, heterogéneos y con características de tráfico diferentes que exigen requisitos estrictos y específicos de la red móvil. Estos servicios clasificados por la Unión Internacional de Telecomunicaciones (ITU, por sus siglas en inglés) son: comunicación masiva tipo máquina (mMTC); comunicación ultra confiable y de baja latencia (urLLC); y banda ancha móvil mejorada (eMBB) (Popovski et al., 2018), (ITU-R, 2015).

La tecnología 5G permitirá a los usuarios nuevas aplicaciones incluyendo: realidad aumentada (AR), realidad virtual (VR), internet de las cosas (IoT), vehículos autónomos y muchas más; todo esto gracias a que proporcionaría avances significativos en cuanto a los servicios y la gestión de red comparado con las infraestructuras móviles tradicionales anteriores, con la influencia de esquemas de inteligencia artificial como análisis de datos, reconocimiento de patrones y predicciones (Barona et al., 2017). En la figura 1 se ilustran los distintos escenarios previstos para las redes 5G según (ITU-R, 2015).



Figura 1. Escenarios de uso previstos para las redes 5G (ITU-R, 2015).

Las redes 4G presentan un diseño basado en una sola arquitectura móvil conocida como "one size fit all", donde todos los servicios son tratados de manera similar, esto conlleva a un aprovechamiento ineficiente cuando los requisitos para diferentes servicios son generalmente similares; ya que la red no sabe diferenciar que trato brindar a cada usuario (Sama et al., 2016). Por el contrario, 5G se auxilia de

tecnologías emergentes como Virtualización de Funciones de Red (NFV) y Redes Definidas por Software (SDN) que permiten mejorar la eficiencia y flexibilidad de la red, introduciendo una Arquitectura Basada en Servicios (SBA) y una característica vital introducida en 5G, *network slicing* o segmentación de la red (Vyakaranam and Krishna, 2018).

En el escenario de *network slicing*, se necesita crear instancias compuestas por un grupo de funciones de red particularizadas para cada servicio, algunos de ellos con características muy similares del caso de uso en función, definidos por la ITU. Con *network slicing*, una infraestructura de red física se particiona en múltiples redes virtuales facilitando la oportunidad del operador de tener una configuración de red dinámica, se establece un acuerdo de nivel de servicio (SLA) específico para la aplicación y/o servicio que se implemente en dicha red virtual (*network slice*, NS) (5G Americas, 2016). Cada una de esas redes virtuales tiene armonizada (apareada) su infraestructura con sus respectivos parámetros específicos de desempeño KPI (latencia, ancho de banda, *throughput*, etc.) y de calidad (KQI) (Song et al., 2018).

Como se ilustra en la figura 2, en 5G se propone que los recursos de red estén virtualizados, para lo cual se crean segmentos o cortes de red que están aislados de forma lógica, aun cuando se comparte la misma infraestructura física. En el caso del segmento de red eMBB, éste está dirigido a los servicios que demanden gran ancho de banda y altas velocidades de datos (hasta 10 Gbps, y en ciertos casos y circunstancias podría alcanzarse una velocidad máxima de 20 Gbit/s), como video de ultra alta resolución, banda ancha inalámbrica móvil y fija, realidad aumentada y realidad virtual, (ITU, 2018).

El segundo segmento de red, mMTC, está dedicado a la comunicación masiva entre máquinas. Este tipo de comunicaciones ha aumentado significativamente con la aparición de dispositivos IoT e Industria 4.0. En estos casos los requerimientos de ancho de banda son bajos, pero se requiere de una red dedicada (segmento) para cumplir con los objetivos de rendimiento de una gran cantidad de dispositivos distribuidos físicamente en grandes áreas. El tercer segmento genérico incluido en 5G aborda aplicaciones de misión crítica y sensibles a la latencia, denominados servicios uRLLC.

En este tipo de servicios se debe garantizar 1 ms de latencia en todo momento, permitiendo lazos de control rápidos y, por lo tanto, alta capacidad de reacción. Como ejemplos se pueden citar: mecanismos de protección para la electricidad, autos autónomos, cirugías robóticas a distancia, entre otros (Kurtz et al., 2018), (Popovski et al., 2018).

Para cumplir con los requisitos divergentes de los diferentes tipos de servicio, las redes deben admitir la segmentación. Para poder llevar a cabo la segmentación de red se utilizan las tecnologías NFV y SDN, las cuales son consideradas componentes clave de las infraestructuras 5G (Kurtz et al., 2018), (Popovski et al., 2018).

La arquitectura de segmentación de red contiene segmentos de acceso (acceso radio y acceso fijo), segmentos de la red de núcleo (CN) y la función de selección que conecta estos segmentos parciales en un segmento de red completo compuesto tanto de la red de acceso como de la red de núcleo. La función de selección enruta las comunicaciones a un segmento CN apropiado para proporcionar un servicio específico (5G Americas, 2016).

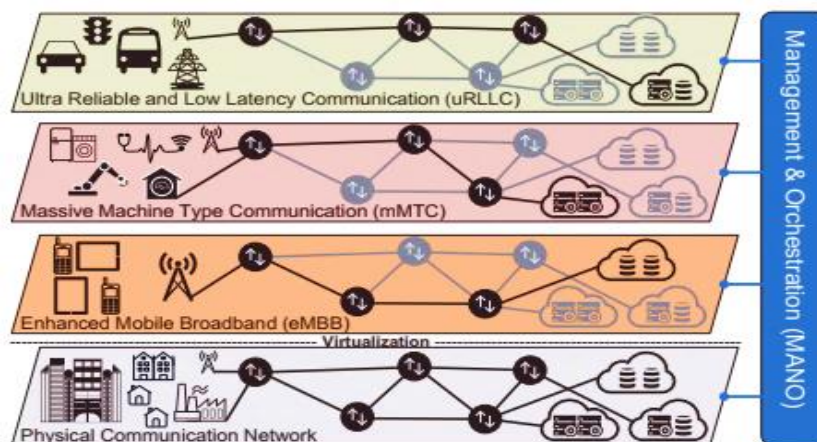


Figura 2. Segmentación de una red de comunicación 5G (Kurtz et al., 2018).

Los criterios para definir los segmentos de acceso y los segmentos CN incluyen la necesidad de cumplir con los diferentes requisitos de los servicios (o aplicaciones). Además, se debe cumplir con los diferentes requisitos de la comunicación. Cada segmento CN se construye a partir de un conjunto de funciones de red (NF). Un factor importante es que algunas NF se pueden usar en varios segmentos, mientras que otras NF se adaptan a un segmento específico. La asignación entre dispositivos, segmentos de acceso y segmentos CN puede ser 1:1:1 o 1:M:N, tal y como se puede observar en la figura 3 (5G Americas, 2016).

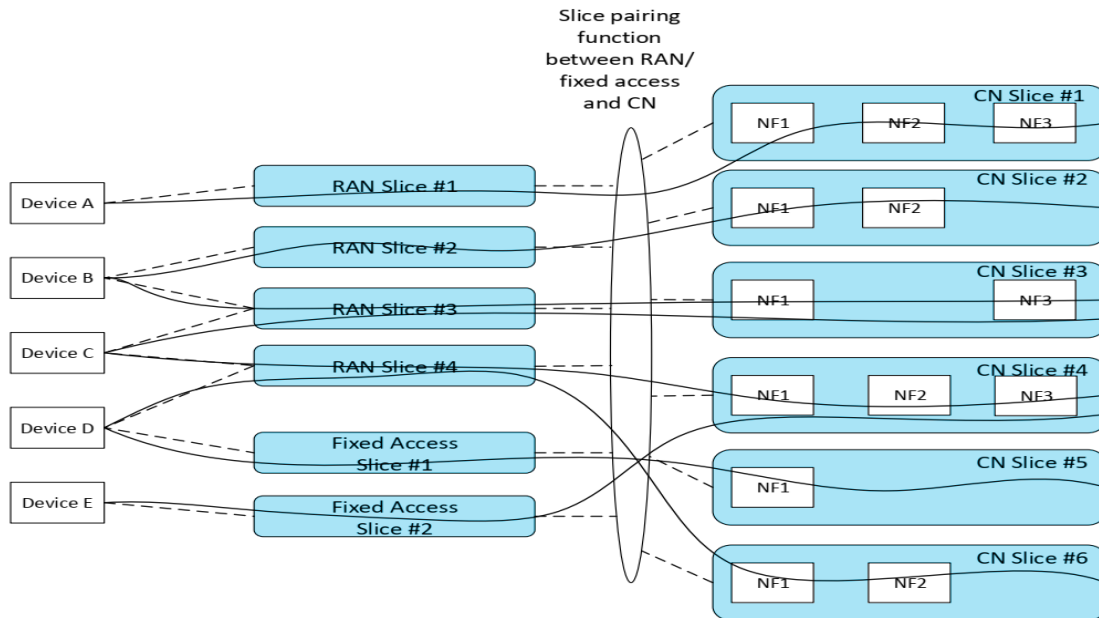


Figura 3. Asignación entre dispositivos, segmento de acceso y segmento CN en una red 5G segmentada (5gAmericas, 2016).

1.1 Planteamiento del problema

Es imperativo, para establecer un SLA, considerar y cumplir los indicadores claves que llevan a prestar un servicio de calidad. En ese sentido, tanto los indicadores claves de desempeño (KPI) como los de calidad (KQI) conforman la base de la calidad del servicio (QoS) para el establecimiento de los SLA entre el usuario y el proveedor. Resulta insoslayable e importante realizar un monitoreo del cumplimiento del SLA (parámetros de QoS) para de ser necesario realizar ajustes de configuración en forma automática (Gramaglia et al., 2017).

De acuerdo con lo anterior, el problema a resolver en esta investigación es el siguiente:

- ✓ ¿Cuál sería el modelo que permita la predicción del comportamiento de los indicadores clave de QoS en la creación de instancias de *network slicing* y el cumplimiento de su correspondiente SLA?

1.2 Justificación

Las diversas características y la complejidad que definen las redes 5G requieren de flexibilidad y dinamismo en sí, haciéndose imprescindible que sean manejadas de un modo automatizado y sean

capaces de determinar autónomamente la configuración óptima del sistema con la mínima intervención humana (Van Der Meer et al., 2018). El avance de técnicas como el aprendizaje automático o aprendizaje de máquinas (*Machine Learning*, ML por sus siglas en inglés), hace posible que éstas se conviertan en una de las herramientas imprescindibles como soporte a la gestión de las redes 5G (Ben Yahia et al., 2017), (Jiang et al., 2017), (Mullins et al., 2017). Estas herramientas permiten automatizar la gestión para cada uno de los escenarios propuestos en IMT-2020, se mejora el desempeño a través de la optimización de la QoS brindada (Gramaglia et al., 2017), (5G Americas, 2016).

1.3 Hipótesis

En el proceso de creación de instancias de *network slicing* es posible, utilizar técnicas de aprendizaje automático, para predecir el comportamiento de los indicadores clave de QoS y el cumplimiento de los acuerdos de servicios.

1.4 Objetivos

1.4.1 Objetivo general

Proponer un modelo de predicción de violaciones de SLA en una red 5G con *network slicing*, utilizando como apoyo el aprendizaje automático.

1.4.2 Objetivos específicos

- ✓ Determinar los indicadores clave de QoS de los servicios de red 5G.

- ✓ Seleccionar el conjunto de datos (*dataset*) que se utilizará para el análisis de los datos de los indicadores de la red.

- ✓ Definir las violaciones de los SLA de los servicios que soporta la red, teniendo en cuenta los datos disponibles en el *dataset*.
- ✓ Seleccionar los modelos de predicción que se utilizarán.
- ✓ Crear sistema de monitorización de la QoS para la predicción de violaciones de SLA.

1.5 Estructura de la tesis

Esta sección detalla cómo se estructura el presente trabajo de tesis. A continuación, se describen brevemente los aspectos que se tratan en cada uno de los capítulos.

En el Capítulo 2, se presenta el estado del arte sobre la segmentación de las redes 5G, la gestión de los SLA y la aplicación del ML a la gestión en 5G. Se detallan los fundamentos de esta investigación, se analizan trabajos relacionados y se presentan conceptos vinculados al tema tratado tales como: aseguramiento de los SLA y QoS, factores de impacto sobre el *throughput* y redes neuronales.

En el Capítulo 3, se realiza la propuesta del modelo de predicción de violaciones de los SLA. Se diseña la solución y se describen cada uno de los bloques funcionales de la misma, destacando dentro de ellos las dos arquitecturas de redes neuronales que se proponen para el pronóstico del *throughput* y el *handoff* vertical.

En el Capítulo 4, se valida el modelo propuesto y se realiza la discusión de los resultados y las consideraciones fundamentales sobre estos en la presente investigación.

En el Capítulo 5, se presentan las conclusiones del trabajo de tesis, las contribuciones al conocimiento y las limitaciones. Finalmente, se comenta el trabajo futuro asociado a la investigación.

En este trabajo de tesis, las figuras se dejan en formato original para cuidar su entendimiento y aplicación, y así evitar una posible pérdida de información en la traducción.

Capítulo 2. Marco de referencia del proyecto

Las redes 5G presentan la capacidad de cumplir con una amplia gama de requisitos al tiempo que deben soportar una diversidad de servicios, aplicaciones, usuarios y modelos de negocios sin precedentes. Estas redes deben admitir ancho de banda de bajo costo en un extremo del espectro, al tiempo que proporcionan conexiones de IoT de baja potencia y velocidad. Asimismo, tendrán que garantizar conexiones de baja latencia, alta velocidad y ultra confiables en el otro extremo. Lo anterior conlleva a que el éxito de este paradigma de red inalámbrica multifuncional radicaré en cómo se diseñan las redes 5G de extremo a extremo (Kurtz et al., 2018), (Popovski et al., 2018), (5G Americas, 2016).

La arquitectura de las redes 5G debe ser altamente adaptable para cumplir con las expectativas de rendimiento de los presentes y futuros servicios de telecomunicaciones. La segmentación de las redes 5G es una característica vital para garantizar lo anterior, ya que permite a los operadores dividir la red en múltiples redes virtuales que se ejecutan en una infraestructura de red común, que incluye la red de acceso por radio (RAN) y la red de núcleo, para soportar diferentes tipos de servicios. Este enfoque ofrece varios beneficios al mejorar la capacidad de los operadores para implementar solo las funciones necesarias para admitir casos de uso específicos (Kurtz et al., 2018), (Popovski et al., 2018), (5G Americas, 2016).

2.1 Arquitectura de la segmentación de red en 5G

Tal y como se explicó en el capítulo anterior (ver figura 2), en 5G se propone que los recursos de red estén virtualizados, para lo cual se crean segmentos de red que están aislados de forma lógica, aún cuando se comparte la misma infraestructura física. Para cumplir con los requisitos divergentes de los diferentes tipos de servicio, las redes deben admitir la segmentación de red, para lo cual se utilizan las tecnologías NFV y SDN. En la figura 4 se muestra una arquitectura genérica de NFV definida por la ETSI, y ampliada con módulos para brindar la solución de segmentación en redes 5G (Kurtz et al., 2018), (Popovski et al., 2018).

En el capítulo anterior se explicó que la arquitectura de segmentación de red contiene segmentos de acceso, segmentos CN y la función de selección que conecta estos segmentos parciales en un segmento de red completo (ver figura 3). El emparejamiento entre segmentos de acceso y segmentos CN, puede ser estático o puede tenerse una configuración semidinámica para lograr garantizar la función de red requerida y las necesidades de comunicación. Un segmento de red debe estar disponible el tiempo

previsto por el servicio que lo demande, además debe proporcionar soporte completo de la función de red a los dispositivos conectados a dicho segmento de red (5G Americas, 2016).

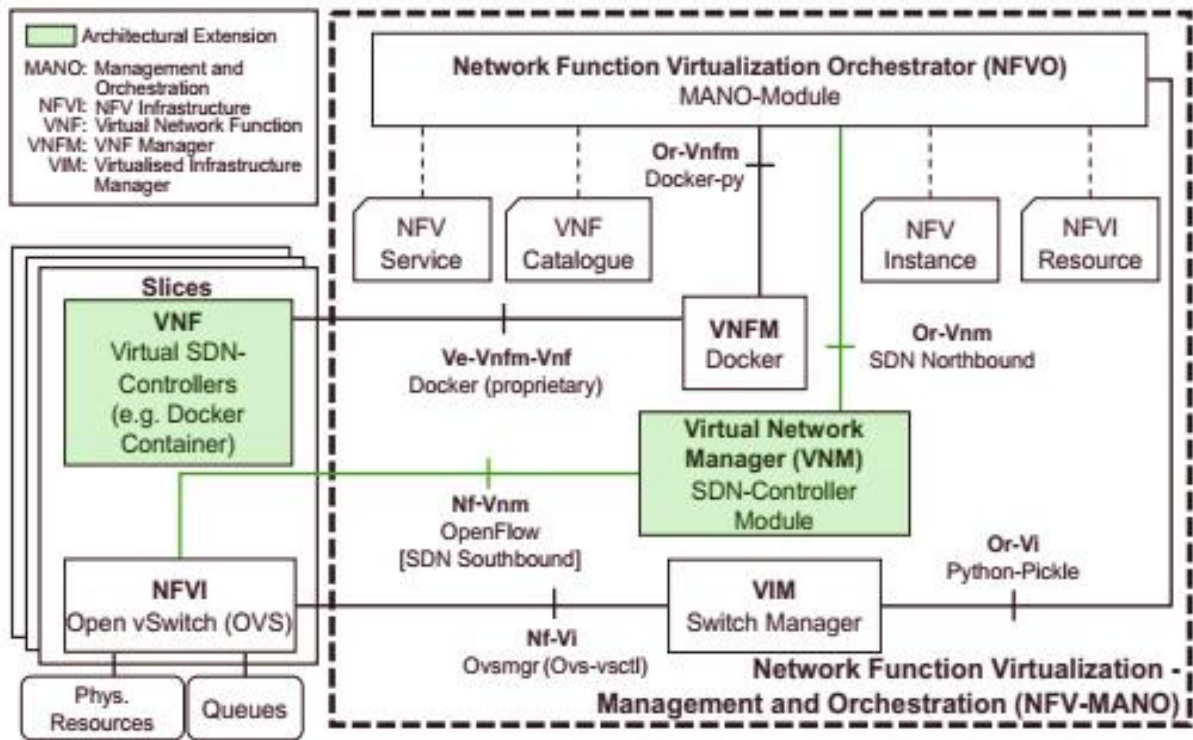


Figura 4. Arquitectura NFV extendida con componentes para brindar la solución de segmentación de la red 5G (Kurtz et al., 2018).

La separación de las funciones del plano de control y del plano de usuario es uno de los principios más significativos de la arquitectura de la red CN de 5G. Esta separación permite las siguientes acciones (5G Americas, 2016):

- ✓ Escalar los recursos de control y del plano de usuario de forma independiente.
- ✓ La distribución del plano del usuario a sitios más cercanos al dispositivo del usuario.
- ✓ La selección de las funciones del plano de usuario necesarias para diferentes segmentos.
- ✓ La descomposición del plano de usuario en funciones más pequeñas.
- ✓ El soporte para la migración a implementaciones basadas en la nube.

Tal y como se muestra en la figura 5, el plano de control puede ser independiente en muchos aspectos del plano del usuario, como es el caso del despliegue físico y las especificaciones de transporte de los niveles L2 y L3. La funcionalidad típica del plano de control incluye capacidades tales como el mantenimiento de información de ubicación, negociación de políticas y autenticación de sesión (5G Americas, 2016).

En el caso de la RAN, la segmentación debe tener en cuenta los siguientes aspectos (Popovski et al., 2018), (5G Americas, 2016):

- ✓ El tipo de acceso de radio (RAT) que admite los servicios de red proporcionados por el segmento.
- ✓ La configuración de los recursos RAN para interactuar adecuadamente con el segmento de red y soportarlo.

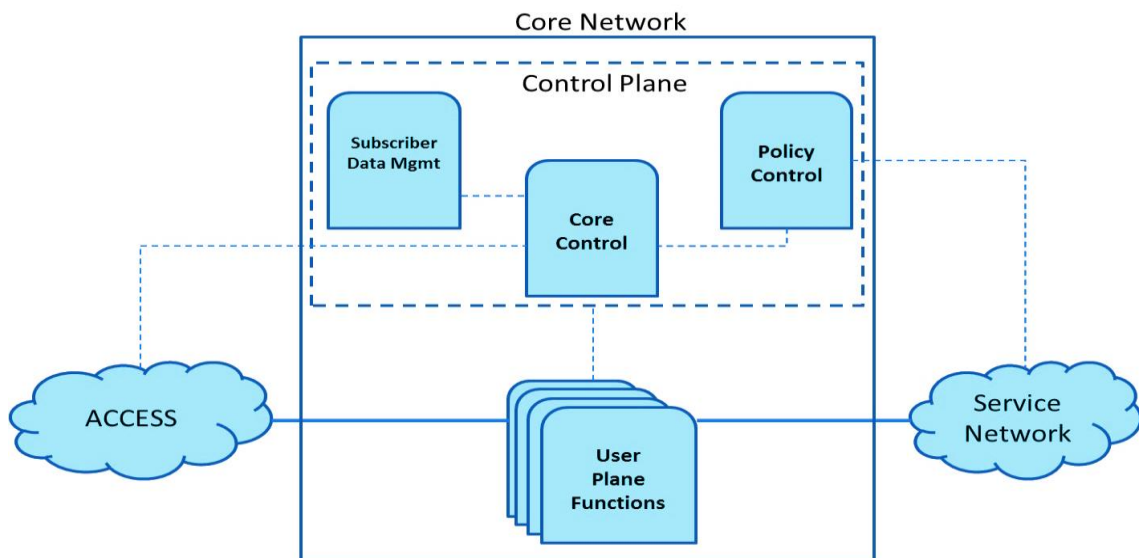


Figura 5. Separación del plano de control y del plano de usuario en el CN de una red 5G (5G Americas, 2016).

Las redes 5G deben cumplir objetivos KPI multidimensionales. Algunos de los objetivos no se pueden cumplir al mismo tiempo, por ejemplo: la baja latencia y la fiabilidad a menudo conllevan el costo de la eficiencia espectral (5G Americas, 2016).

La segmentación en la RAN es una asignación (mapeo) de identificadores de segmentos a un conjunto de reglas de configuración. En este caso no se separan las funciones del plano de control y del plano de usuario en cada segmento, ya que existen reglas de configuración de la RAN asociadas con cada segmento,

de tal manera que se garanticen los requisitos de los servicios de red soportados por el segmento (Popovski et al., 2018), (5G Americas, 2016).

Algunos requisitos de diseño y operación en la segmentación de la RAN son (Popovski et al., 2018), (5G Americas, 2016):

- ✓ Cada segmento de red es soportado en la RAN aplicando un conjunto de reglas de configuración a las funciones de los planos de control y de usuario de la RAN.
- ✓ Algunas funciones de red son comunes a varios segmentos (por ejemplo, gestión de la movilidad).
- ✓ Las funciones de control comunes coordinan el uso de recursos RAN entre los segmentos. Lo anterior permite mejorar el rendimiento y la eficiencia de todo el sistema 5G.

Los recursos de radio entre los segmentos se pueden compartir mediante programación o contención. Si se realiza mediante la programación, cada uno de los segmentos envía solicitudes de recursos a un planificador central, como el planificador en una estación base o un controlador RAN central. Luego, el programador asigna los recursos radio a los segmentos de red en función de factores como la cantidad de recursos solicitados por el segmento, la prioridad del trabajo que realiza el segmento y la carga de tráfico general. En un sistema basado en contención, cada uno de los segmentos adquiere recursos de radio de manera autónoma siguiendo algunas reglas predefinidas, lo que constituye una asignación de recursos estáticos, ya que cada segmento está preconfigurado para operar con recursos dedicados en todo su tiempo de operación. Se puede concluir que compartir los recursos de manera estática garantiza la asignación de recursos a cada segmento, mientras que si se comparten dinámicamente se logra la optimización general del uso de los recursos de radio (Popovski et al., 2018), (5G Americas, 2016).

En la figura 6, se muestra un ejemplo en el que se asignan recursos de radio dedicados para cada segmento. Los segmentos comparten los recursos de radio utilizando la multicanalización por división de frecuencia (FDM) y la multicanalización por división de tiempo (TDM) (5G Americas, 2016).

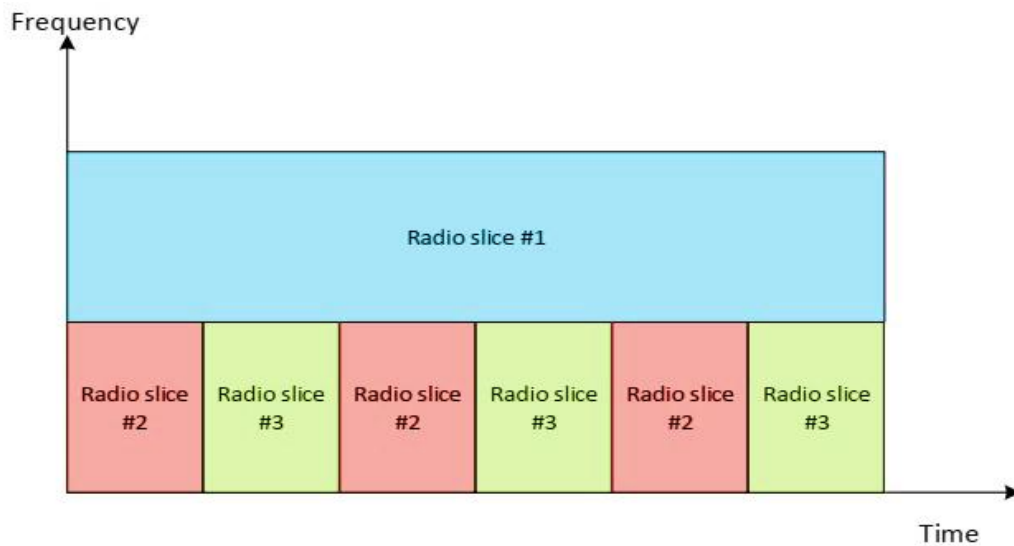


Figura 6. Compartición de recursos de radio entre segmentos de red utilizando FDM y TDM (5G Americas, 2016).

2.2 Aspectos operacionales de la segmentación de red

La Alianza para la Nueva Generación de Redes Móviles (*Next Generation Mobile Network*, NGMN por sus siglas en inglés) define la arquitectura de segmentación de red compuesta de tres capas (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016), (Gutz et al., 2012):

- ✓ **Capa de instancia de servicio:** los servicios de usuario final o los servicios empresariales son compatibles con la red. Dichos servicios pueden ser proporcionados por el operador de red o por un tercero.
- ✓ **Capa de instancia de segmento de red:** la instancia del segmento de red proporciona las características de red requeridas por una instancia de servicio.
- ✓ **Capa de recursos:** las funciones de red física y virtual utilizadas para implementar una instancia de segmento.

NGMN también define un anteproyecto de segmento de red, que es una descripción completa de la estructura, configuración y los flujos de trabajo sobre cómo crear y controlar la instancia de segmento de red durante su ciclo de vida (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016), (Gutz et al., 2012).

La figura 7, ilustra el plano de orquestación y gestión de red (NMO), que proporciona la gestión y las funciones de orquestación para las tres capas que componen la arquitectura de red segmentada (5G Americas, 2016).

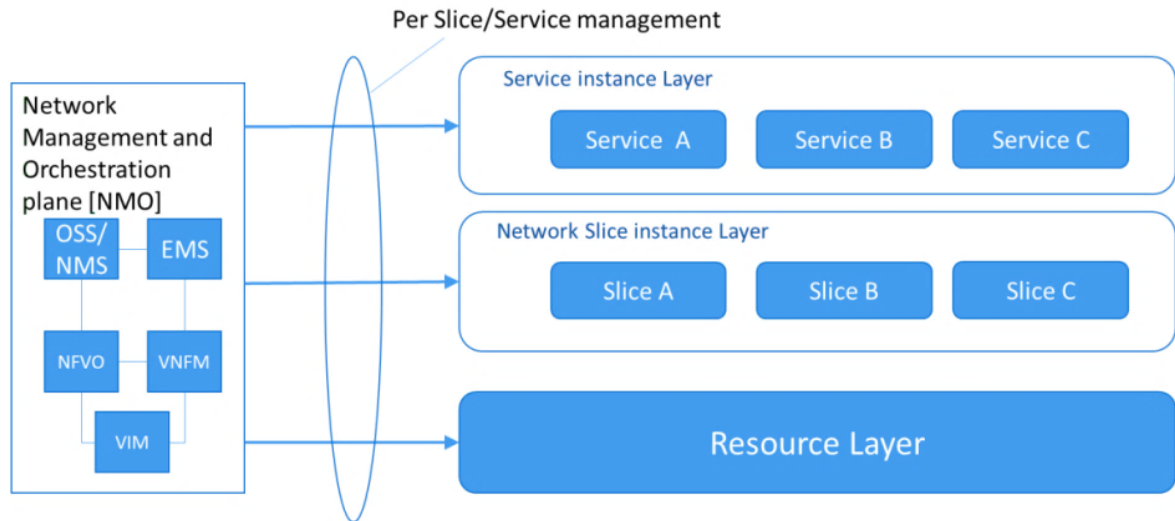


Figura 7. Gestión de los segmentos de red (5G Americas, 2016).

La gestión del segmento de red debe tener en cuenta las funciones de mapeo relevantes utilizadas para construir un segmento. También debe tener en cuenta las reglas que utiliza el proveedor de servicios para definir el estado inicial de este mapeo y las decisiones del ciclo de vida basadas en la dinámica de red y de uso (3GPP, 2018), (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016).

Tal y como se muestra en la figura 8, los servicios se pueden mapear en diferentes segmentos, de acuerdo con las políticas del operador y las estrategias comerciales. Cada segmento tiene un determinado conjunto de atributos (por ejemplo: baja latencia de extremo a extremo, gran ancho de banda, etc.), lo cual puede hacer que sea más apropiado para cierto tipo de servicios (3GPP, 2018), (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016).

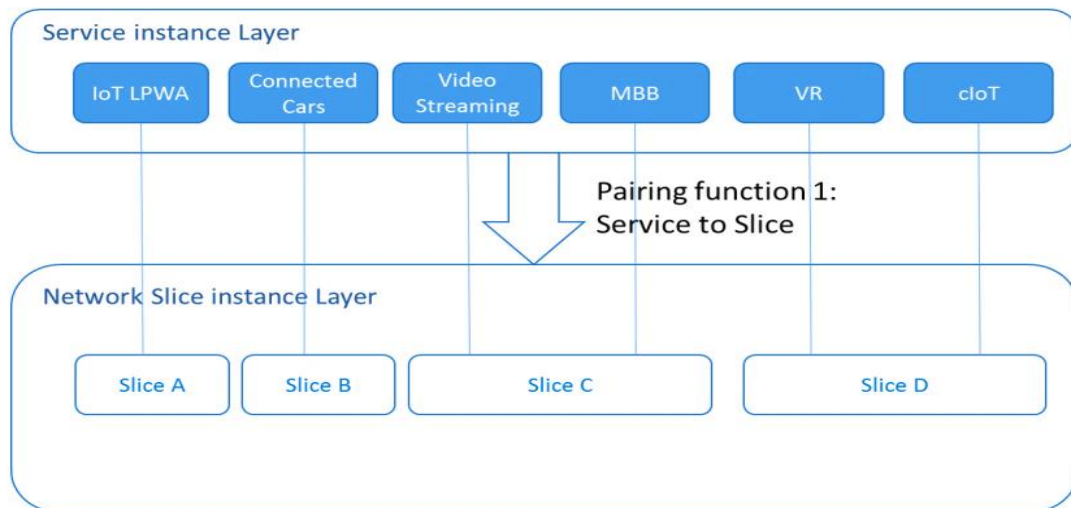


Figura 8. Mapeo entre la instancia del servicio y la instancia del segmento de red (5G Americas, 2016).

Desde la perspectiva operativa, el sistema de gestión debe proporcionar a los operadores la capacidad de mapeo de servicios a segmentos de red, tal que (3GPP, 2018), (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016):

- ✓ El operador tiene que mapear un servicio ofrecido a un segmento de red particular.
- ✓ Un segmento de red proporciona un conjunto de capacidades de red y nivel de rendimiento / SLA que pueden ser adecuado para un conjunto de tipos de servicios. Como resultado, se pueden asignar varios servicios al mismo segmento de red. Por ejemplo, un segmento orientado a eMBB puede admitir transmisión de video, descargas de música y transferencias de archivos grandes. Mientras tanto, otro segmento orientado a baja latencia puede admitir VoIP y mensajería en tiempo real.

Los usuarios con el mismo tipo de servicio pueden estar asociados con diferentes segmentos. Lo anterior es útil cuando se hace necesario el aislamiento de inquilinos, donde cada segmento está dedicado a un grupo de usuarios (por ejemplo: usuarios locales y usuarios itinerantes). Otro ejemplo puede ser la separación geográfica, donde los usuarios son dirigidos a un segmento más cercano para reducir los requisitos de latencia y *backhaul*. Por otra parte, un segmento de red puede estar dedicado a un tipo de servicio particular. Por ejemplo, las comunicaciones de IoT pueden estar asociadas a un segmento de red

cuya baja sobrecarga y características mínimas están diseñadas para proporcionar el mejor rendimiento para dispositivos IoT (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016).

Debido a la naturaleza de un segmento de red, pueden estar involucrados múltiples dominios administrativos. Esto abarca recursos de red tanto cableados como inalámbricos, virtualizados o de otro tipo. Por ejemplo, la itinerancia es un escenario donde una instancia de segmento de red puede atravesar más de un dominio administrativo. Por lo tanto, los servicios, las instancias de servicio de red o las instancias de segmento de red deben operarse en múltiples dominios administrativos. En este contexto, se necesita en la capa de servicio (servicio de cara al cliente) una función de orquestación de segmento de red, responsable para gestionar el segmento de la red. En la capa de recursos (servicio orientado a recursos), la gestión del segmento de red la realiza el orquestador de recursos, compuesto por el NFVO y los configuradores de recursos de aplicaciones, donde las aplicaciones son servicios 3GPP, transporte, y otros (Sohaib et al., 2021), (Napolitano et al., 2018), (5G Americas, 2016), (NGMN Alliance and Hedman (Ed.), 2016).

2.3 Acuerdos de nivel de servicio en redes 5G

Entre los aspectos del sistema 5G que se ven afectados por la adopción de la segmentación de red se encuentran los procedimientos en torno a la gestión y la aplicación de SLA; estos son los contratos entre los proveedores y los clientes que definen la QoS esperada en términos de uno o varios objetivos de nivel de servicio. En el contexto de los sistemas móviles 5G, los operadores de red proporcionan un segmento de red de extremo a extremo como servicio, que tiene asociado un SLA según los requisitos del cliente (Papageorgiou et al., 2020), (3GPP, 2018).

Un SLA entre dos entidades por lo general contiene dos partes: una técnica, y otra no técnica, tal y como se muestra en la figura 9 (García, 2007), (Haddad and Viniotis, 2007).

Los objetivos de nivel de servicio (SLO) constituyen la parte no-técnica de un SLA. Estos dividen a un SLA en objetivos individuales, definiendo métricas para hacer cumplir, y/o para vigilar el SLA, para así determinar si se están cumpliendo o no los objetivos. Ejemplos: *Uptime*; tiempo medio entre fallas (MTBF); tiempo de respuesta; tiempo medio de reparación (MTTR). También dentro de los parámetros no técnicos pueden ser considerados los procedimientos de violación. Estos se deben llevar a cabo en caso de que se

produzcan violaciones en las garantías de las especificaciones del nivel de servicio (SLS) (García, 2007), (Haddad and Viniotis, 2007).

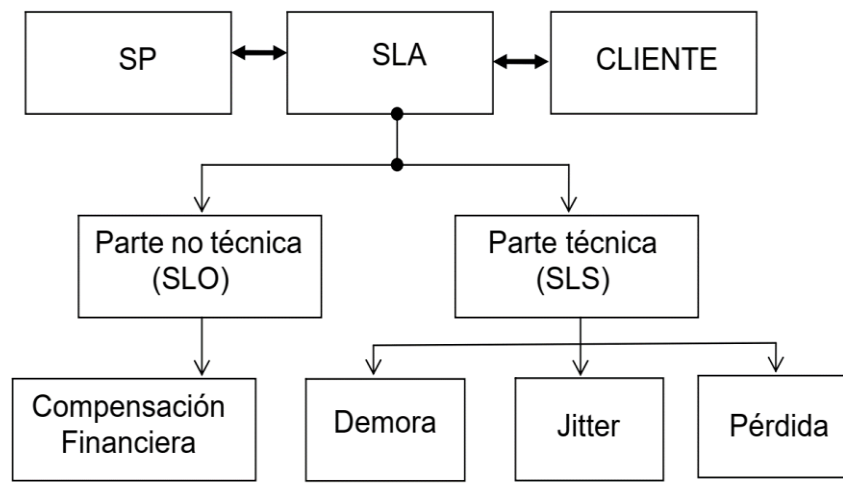


Figura 9. Estructura de un SLA (García Rodríguez, 2007).

Asimismo, las especificaciones del nivel de servicio (SLS) constituyen la parte técnica de un SLA e incluyen métricas de QoS, por ejemplo: latencia, *throughput*, *jitter*, pérdida de paquetes, disponibilidad, entre otros. Los SLS pueden ayudar a conseguir el control de QoS de extremo a extremo. Las métricas usadas toman la forma de atributos cuantitativos o cualitativos (García, 2007), (Haddad and Viniotis, 2007).

2.3.1 Aseguramiento de los SLA y la QoS

Se espera que la tecnología 5G impacte la red móvil y a ecosistemas asociados. Las garantías necesarias para la calidad del servicio se pueden maximizar con las capacidades de las funciones de red virtualizadas (VNF) y los servicios de red. Esto implica la utilización de SLA para garantizar que los servicios de red se proporcionen de manera eficiente y controlada (Kapassa et al., 2018), (Sama et al., 2016).

Sin embargo, la complejidad de la tarea de determinar las políticas de provisión de recursos en entornos multimodales, así como las características y propiedades de varios VNF y servicios, da como resultado SLA personalizados que no consideran todos los aspectos del entorno 5G (Sama et al., 2016), (Kapassa et al., 2018). De lo anterior surge la necesidad de un marco de gestión de SLA para mapear los requisitos de los usuarios (*high-level requirements*) a los recursos necesarios de la red (*low-level requirements*), para

mejorar la capacidad del proveedor de servicios para cumplir con los correspondientes compromisos de SLA. Además, es necesario considerar un mecanismo para la generación dinámica de Plantillas SLA con objetivos de nivel de servicio iniciales adaptados a cada servicio del proveedor (Kapassa et al., 2018), (Sama et al., 2016).

El ciclo de vida del SLA en el dominio 5G es una parte importante de la provisión de servicios; es administrado por las plataformas de servicios 5G que lo acompañan y es un proceso dinámico que comprende cuatro etapas clave: arquitectura, compromiso, operación y terminación, tal y como se muestra en la figura 10. El ciclo de vida del SLA está totalmente alineado con los principios de 5G y se ejecuta en paralelo con el ciclo de vida del servicio de red (Kapassa et al., 2019), (Van Rossem et al., 2017).



Figura 10. Ciclo de vida de los SLA en 5G (Kapassa et al., 2019).

La primera fase comienza con la selección de un servicio de red y la definición de requisitos por parte del desarrollador. Normalmente, el operador es el responsable de examinarlos, tener en cuenta necesidades comerciales importantes e implementar Plantillas de SLA, como oferta inicial a los clientes del servicio (Kapassa et al., 2019).

Durante la segunda fase, la selección de diferentes servicios de red es el resultado de aspectos comerciales, que son la base para diferentes restricciones de QoS, que también se pueden definir como requisitos del acuerdo. La preferencia de un operador o proveedor de servicios de red depende del servicio de red deseado, sus características, restricciones presupuestarias y así sucesivamente. Las expectativas de QoS, hace que los usuarios negocien con sus operadores o proveedores de servicios los niveles de QoS. Después de un proceso de negociación exitoso, se crea un SLA para describir los parámetros de QoS acordados (Kapassa et al., 2019), (Touloupou et al., 2019), (Kapassa et al., 2018).

Posteriormente, tiene lugar la fase de operación. Esta fase comprende el despliegue real del servicio, la población de los respectivos servicios con datos en ejecución, el establecimiento de canales de comunicación y actividades operativas adicionales. Además, la fase de operación monitorea datos en tiempo real, con el propósito de evitar o gestionar infracciones inesperadas de los SLA (Kapassa et al., 2019), (Touloupou et al., 2019), (Kapassa et al., 2018).

Finalmente, la fase de terminación se ocupa del final de la relación entre el operador o proveedor de servicios y el cliente, incluido el fin de la relación jurídica. Esta última fase incluye la evaluación de alternativas, compromisos de liquidación y terminación, exportación de datos, atención al cliente y diligencia, y supresión de datos. Todo lo anterior debe considerarse si el servicio de red fue rescindido, o se violó el SLA (Kapassa et al., 2019), (Touloupou et al., 2019), (Kapassa et al., 2018).

Existen varios trabajos activos sobre la gestión de SLA para infraestructuras en la nube y 5G; en estas investigaciones se buscan soluciones para gestionar los parámetros de QoS y monitorear de manera eficiente los SLO (Papageorgiou et al., 2020), (Kapassa et al., 2019), (Parada et al., 2018), (Gramaglia et al., 2016). En la figura 11 se muestra el posicionamiento del Gestor de SLA dentro de un sistema 5G con *network slicing*.

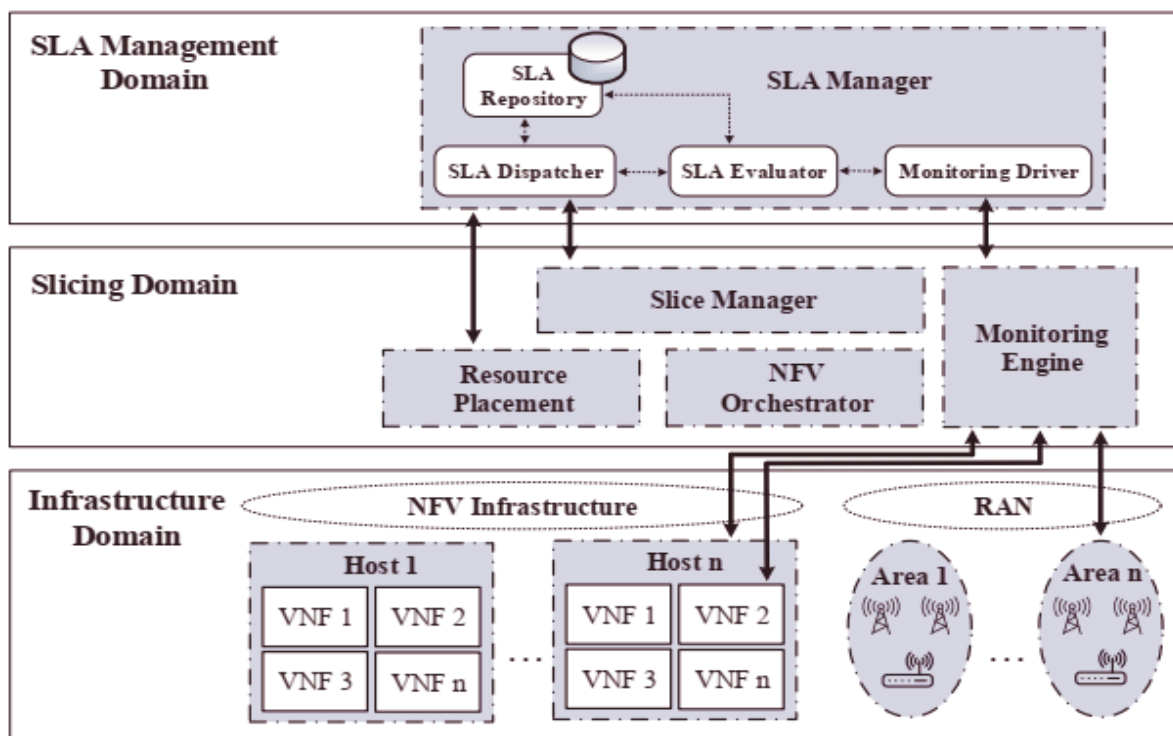


Figura 11. Posicionamiento del Gestor de SLA en un sistema 5G con *network slicing* (Papageorgiou et al., 2020).

Internamente, el Gestor de SLA está compuesto por los siguientes módulos de software (Papageorgiou et al., 2020):

- ✓ Despachador de SLA: Contiene la API para conectarse con otros módulos de la orquestación, por ejemplo, para recibir nuevas instancias de SLA o activación de la verificación de SLA y difundir información sobre violaciones.
- ✓ Evaluador de SLA: Calcula los valores de todos los parámetros de los SLO de los SLA activos, y verifica que estén dentro de los límites acordados, generando alarmas sobre violaciones.
- ✓ Repositorio de SLA: Almacena los SLA, así como su estado y posibles infracciones.
- ✓ Controlador de Monitoreo: Se conecta con el módulo de Monitoreo de la plataforma, para recibir valores en tiempo real de las métricas que se estén monitoreando, según las instrucciones del Evaluador de SLA.

Externamente, el Gestor SLA requiere una interfaz con el Gestor de Segmentos (*Slice Manager*), que es la entidad que asigna, provisiona y gestiona los segmentos de red. Entre otros, después de creado el segmento de red, el Gestor de SLA recibe desde el Gestor de Segmentos, la información del SLA asociado a dicho segmento de red. Además, existe una conexión opcional entre el Gestor de SLA y el módulo de Ubicación de Recursos, lo cual permite que el Gestor SLA tenga un impacto directo en la asignación de las VNF para garantizar el nivel de servicio acordado. Por último, existe una conexión entre el Gestor de SLA y el módulo de Monitoreo mediante la cual se obtiene en tiempo real información de la infraestructura NFV, las VNF y la RAN (Papageorgiou et al., 2020), (Kapassa et al., 2019), (Parada et al., 2018).

2.4 Requerimientos de desempeño

Los casos de uso de 5G y los requisitos relacionados con 5G pertenecientes a ITU, NGMN y 3GPP han sido caracterizados a escala global por sus partes interesadas desde su evolución. Varios escenarios requieren el soporte de velocidades de datos o densidades de tráfico muy altas del sistema 5G. En este caso los escenarios abordan diferentes áreas de servicio: áreas urbanas, áreas rurales, oficinas, hogares y despliegues especiales (por ejemplo, reuniones masivas, *broadcast*, residenciales, y vehículos de alta

velocidad). Asimismo, varios escenarios requieren el soporte de una latencia muy baja y una disponibilidad de servicio muy alta, lo cual implica una fiabilidad muy alta. La latencia general del servicio depende de varios factores, como son: el retraso en la interfaz de radio, la transmisión dentro del sistema 5G, la transmisión a un servidor que puede estar fuera del sistema 5G, y el procesamiento de datos. En la tabla 1, se puede ver un resumen de los diferentes escenarios e indicadores de desempeño asociados a los servicios genéricos eMBB, mMTC y uRLLC en 5G (3GPP, 2021a), (ETSI, 2021a), (ETSI, 2019).

Tabla 1. Escenarios e indicadores de desempeño de varios servicios 5G (elaboración propia).

Servicios genéricos	Escenarios	KPI
<p>Altas velocidades de datos y altas densidades de tráfico</p> <p>(eMBB y mMTC)</p>	<ul style="list-style-type: none"> - Macro urbana - Macro rural - Zona de interiores - Acceso de banda ancha en una multitud - Urbano denso - Servicios similares a la radiodifusión - Tren de alta velocidad - Vehículo de alta velocidad - Conectividad de aviones 	<ul style="list-style-type: none"> - Experiencia de velocidad de datos (DL) - Experiencia de velocidad de datos (UL) - Capacidad de tráfico del área (DL) - Capacidad de tráfico del área (UL) - Densidad general de usuarios - Factor de actividad - Velocidad del equipo de usuario - Cobertura
<p>Baja latencia y alta confiabilidad</p> <p>(uRLLC)</p>	<ul style="list-style-type: none"> - Control de movimiento - Automatización discreta - Automatización de procesos - Automatización para distribución eléctrica - Infraestructura <i>Backhaul</i> inalámbrica en carretera para Sistemas de Transporte Inteligente - Control remoto - Comunicaciones ferroviarias 	<ul style="list-style-type: none"> - Latencia máx. permitida de extremo a extremo - Tiempo de supervivencia - Disponibilidad del servicio de comunicación - Fiabilidad - Velocidad de datos experimentada por el usuario - Tamaño de la carga útil - Densidad de tráfico - Densidad de conexión - Dimensión del área de servicio

En el presente trabajo los escenarios de interés son áreas suburbanas (*Urban Macro*) y áreas urbanas de *Downtown (Dense Urban)* para el servicio genérico eMBB. En la tabla 2 se muestran los valores objetivo de los KPI para estos escenarios (3GPP, 2021a), (ETSI, 2021a), (ETSI, 2019).

Tabla 2. Valores de los KPI para los escenarios *Urban Macro* y *Dense Urban* (elaboración propia).

Escenario	Velocidad de datos (descarga)	Velocidad de datos (subida)	Capacidad de tráfico del área (descarga)	Capacidad de tráfico del área (subida)	Densidad de usuarios	Factor de actividad	Velocidad del UE	Cobertura
Macro urbana	50 Mbit/s	25 Mbit/s	100 Gbit/s/km ² (Nota 2)	50 Gbit/s/km ² (Nota2)	10000/km ²	20 %	Peatones y usuarios en vehículos (hasta 120 km/h)	Red completa (Nota 1)
Urbana densa	300 Mbit/s	50 Mbit/s	750 Gbit/s/km ² (Nota 2)	125 Gbit/s/km ² (Nota 2)	25000/km ²	10 %	Peatones y usuarios en vehículos (hasta 60 km/h)	<i>Downtown</i> (Nota 1)

Nota 1: Para los usuarios de vehículos, el UE (equipo de usuario) se puede conectar a la red directamente o mediante una estación base móvil incorporada.

Nota 2: Estos valores se derivan en función de la densidad general de usuarios. Se puede encontrar información detallada en (NGMN Alliance, 2016).

Nota 3: Todos los valores de esta tabla son valores objetivos y no requisitos estrictos.

2.5 Análisis de los factores de impacto sobre el *throughput* en 5G

Desde su debut comercial en 2019, más de 173 operadores han desplegado servicios 5G en todo el mundo (Ookla5GMap, 2021). Los servicios comerciales 5G se implementan en modo no autónomo (*Non-Standalone*, NSA por sus siglas en inglés) y modo autónomo (*Standalone*, SA por sus siglas en inglés) tal y como se muestra en la figura 12.

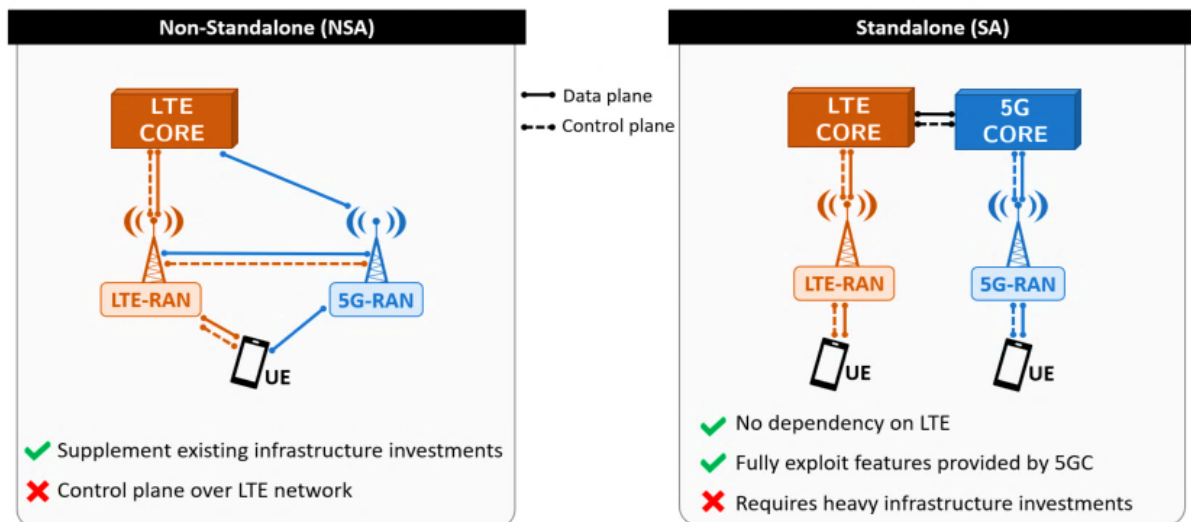


Figura 12. Modos de despliegue de redes 5G: modo no autónomo (izquierda) y modo autónomo (derecha) (Narayanan, Ramadan, Carpenter, et al., 2020).

En el caso del modo NSA, las redes 5G se implementan con antenas 5G, pero comparten la infraestructura central de paquetes 4G. En el modo SA, tanto las antenas como la infraestructura se implementan con tecnología 5G (Narayanan, Ramadan, Mehta, et al., 2020).

El despliegue de las redes 5G por parte de los operadores se puede llevar a cabo en las distintas bandas de frecuencias de radio (banda baja, banda media, banda alta). La banda de frecuencia de operación de la red 5G tiene gran incidencia en la cobertura y el rendimiento, tal y como se muestra en la tabla 3 (Keysight, 2021), (Narayanan, Ramadan, Carpenter, et al., 2020).

Tabla 3. Bandas de frecuencias de operación de las redes 5G (elaboración propia).

Bandas	Sensibilidad a la obstrucción	Cobertura	Ancho de Banda
Banda Alta (24 GHz y mayores)	Alta	Menor a 1 km	100 MHz
Banda Media (1 GHz a 6 GHz) (Posteriormente extendida hasta 7.125 GHz)	Media	15 km	20 MHz
Banda Baja (menor a 1 GHz)	Baja	30 km	10 MHz

Las redes 5G de banda baja y media constituyen la mayoría de los despliegues de servicios 5G en el mundo, estos ofrecen moderadamente un mayor ancho de banda que los servicios 4G / LTE o LTE *Advanced* existentes. Por el contrario, 5G de banda alta ofrece un ancho de banda de hasta 20 Gbps en teoría, pero en la práctica el ancho de banda es considerablemente menor (Narayanan, Ramadan, Mehta, et al., 2020).

Si bien el 5G de banda baja y banda media incluye todas las frecuencias de generaciones anteriores de tecnologías inalámbricas, el rango de frecuencia de banda alta es bastante nuevo en el ecosistema de comunicación inalámbrica convencional. Las redes 5G de banda alta utilizan la tecnología de radio de onda milimétrica (*mmWave*) que trabaja a frecuencias desde 24 GHz a 53 GHz (3GPP, 2017). Pese al gran ancho de banda disponible en esta banda, la pequeña longitud de onda de las señales las hace vulnerables a la atenuación. Lo anterior se solventa con la utilización de arreglos de antenas en fase para formar haces altamente direccionales (Narayanan, Ramadan, Carpenter, et al., 2020).

Las señales *mmWave* son sensibles a bloqueos por parte de los peatones y los vehículos en movimiento. El paso desde la línea de visión (LoS, por sus siglas en inglés) a línea de no visión (NLoS, por sus siglas en

inglés) debido a un bloqueo puede causar una caída significativa de la velocidad de datos o incluso un apagón total, a pesar del algoritmo de formación de haces que intenta generar haces buscando una trayectoria NLoS reflectiva (Nitsche et al., 2015), (Narayanan, Ramadan, Carpenter, et al., 2020).

En (Narayanan, Ramadan, Mehta, et al., 2020) y en (Narayanan, Ramadan, Carpenter, et al., 2020), a través de experimentos de campo, se revelan numerosos factores que afectan el rendimiento de las redes 5G. Estos factores son mucho más sofisticados que los que afectan a las redes 4G / LTE, ya que más allá de afectar de forma independiente el rendimiento, estos factores pueden causar una interacción compleja que es difícil de modelar analíticamente. A continuación, se exponen brevemente estos factores.

2.5.1 Impacto de la geolocalización y de los factores geométricos

En las redes 3G y 4G, la ubicación geográfica es el factor dominante para determinar el *throughput* y la cobertura de la red (Narayanan, Ramadan, Mehta, et al., 2020), (Alimpertis et al., 2019), (Bui et al., 2017). Sin embargo, en las redes 5G el *throughput* fluctúa enormemente, incluso para áreas que tienen garantizado el servicio 5G.

En (Narayanan, Ramadan, Mehta, et al., 2020) se realiza un estudio empírico que llega a la conclusión de que, además de la geolocalización, se debe considerar la dirección de movimiento para lograr una mejor caracterización del rendimiento en redes 5G, sobre todo si son redes 5G *mmWave*. En dicho estudio se realizan muchas pruebas estadísticas que utilizan otras métricas para confirmar la conclusión anterior.

Existen diferentes factores geométricos que se deben tener en cuenta entre el UE y el panel 5G. El primero es la distancia entre el usuario y el panel 5G. La relación cuantitativa distancia-rendimiento difiere de un panel 5G a otro ya que depende de la línea de visión entre el UE y el panel, la cual puede perderse debido a obstáculos. Cuando el UE recupera la línea de visión aumenta el rendimiento y viceversa. Incluso, en caso de NLOS pudiera darse el caso de que un panel 5G más lejano ofrezca mejor desempeño que un panel más cercano, esto se debe a que los rebotes de la señal pudieran permitir una mejor comunicación NLOS para el panel más lejano (Narayanan, Ramadan, Mehta, et al., 2020).

Otro factor geométrico importante es la posición del usuario con respecto a la cara frontal del panel 5G (ángulo de posición entre la línea normal al panel y la línea que conecta el UE al panel 5G). Si el UE se

coloca frente al panel 5G este ángulo será igual a cero, y generalmente en esta posición se exhibe un *throughput* mucho mejor que en otras posiciones (Narayanan, Ramadan, Mehta, et al., 2020).

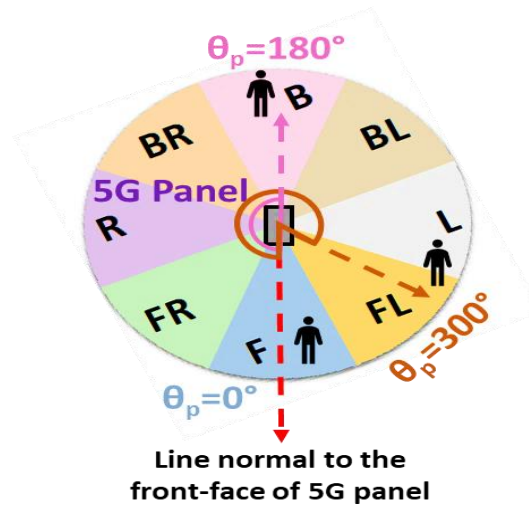


Figura 13. Ángulo de posición entre el UE y el panel 5G (Narayanan, Ramadan, Mehta, et al., 2020).

Por último, el otro factor geométrico de importancia es el ángulo de movilidad entre el UE y el panel 5G (ángulo entre la línea normal a la cara frontal del panel y la trayectoria del UE). Si el usuario se mueve hacia el frente del panel 5G, este ángulo es 180 grados, y 0 grados si se aleja. En general, si el UE se mueve hacia el frente del panel 5G el rendimiento de la red será bueno (Narayanan, Ramadan, Mehta, et al., 2020).

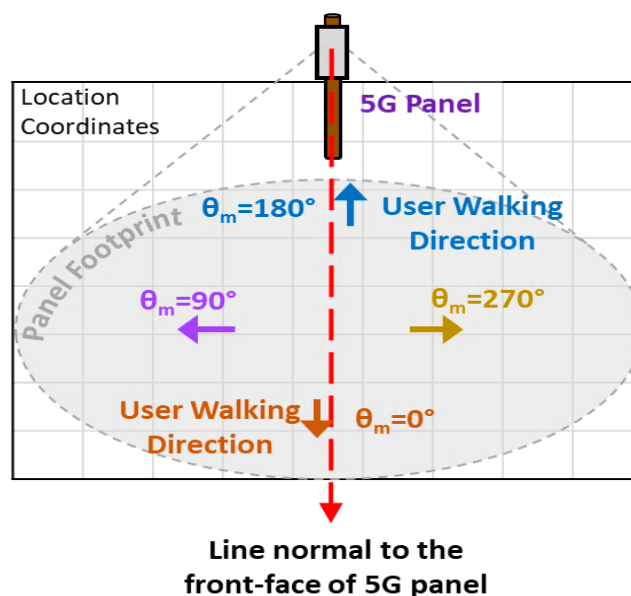


Figura 14. Ángulo de movilidad entre el UE y el panel 5G (Narayanan, Ramadan, Mehta, et al., 2020).

2.5.2 Impacto del medio ambiente

Las condiciones climáticas comunes como la lluvia y la nieve también afectan el rendimiento de 5G *mmWave* ya que sus señales se debilitan por las partículas o incluso la humedad en el aire (Kevin Fogarty, 2019). Por otra parte, la geografía del lugar, los árboles, las edificaciones y otros objetos provocan obstrucciones, las cuales generalmente son las de mayor impacto en el rendimiento de 5G *mmWave*. En el caso de las redes 5G de banda media, el impacto del medio ambiente es mucho menor en el rendimiento, debido a la naturaleza omnidireccional de su señal de radio (Narayanan, Ramadan, Carpenter, et al., 2020).

Las señales 5G *mmWave* difícilmente pueden penetrar en los cuerpos humanos (incluyendo la mano que sostiene al UE), trenes, estructuras de pilares y vidrios polarizados, provocando NLoS entre el transmisor y el receptor. Por otra parte, cuando el UE está dentro de una mochila, una caja de cartón, o un vidrio transparente, las señales 5G pueden penetrar estos contenedores (experimentado para una distancia de menos de 100 metros al panel 5G con LoS). Además, se ha podido comprobar que la señal 5G *mmWave* funciona en vehículos, ya que el parabrisas delantero suele ser de vidrio transparente (Narayanan, Ramadan, Carpenter, et al., 2020).

En ocasiones, a pesar de la NLoS creada por las obstrucciones, los edificios cercanos pueden reflejar las señales y crear múltiples rutas inalámbricas, provocando que las señales reflejadas aún pueden alcanzar el UE. En otras situaciones, si las habitaciones tienen ventanas con protección UV, las señales 5G reflejadas serán atenuadas por lo que el multirayecto se vuelve ineficaz. En caso de que exista NLoS y el multirayecto sea inefectivo se producirá el *handoffs* desde 5G a 4G, lo cual implica una importante degradación del rendimiento (Narayanan, Ramadan, Carpenter, et al., 2020).

2.5.3 Impacto de la movilidad y el *handoff*

La movilidad constituye un desafío técnico importante en 5G debido a las características de la capa física (sobre todo en 5G *mmWave*) que hacen que las señales sean altamente fluctuantes, lo que provoca variaciones bruscas en el rendimiento (Keysight, 2021), (Narayanan, Ramadan, Carpenter, et al., 2020).

La movilidad provoca cambios continuos en la distancia y orientación entre el panel y el UE. Además se pueden producir cambios aleatorios LoS / NLoS debido a los obstáculos, y la ocurrencia de *handoffs* es

más frecuente en comparación con las redes 4G (Narayanan, Ramadan, Carpenter, et al., 2020). Más allá de los 5 km/h, el rendimiento de 5G sufre una gran degradación de tal manera que el *throughput* medio de 5G tiende al *throughput* de 4G, que oscila entre 60 Mbps y 164 Mbps. Al mismo tiempo, el rendimiento máximo para velocidades de movimiento entre 5 y 30 km/h está por encima de 850 Mbps, lo que sugiere que otros factores podrían aumentar el *throughput* de la red (Narayanan, Ramadan, Mehta, et al., 2020).

Sin embargo, el rendimiento de la red no solo depende de la velocidad, sino que también el modo de transporte es determinante. El rendimiento máximo al caminar es capaz de alcanzar altos niveles de *throughput*, más de 1.8 Gbps en todo el rango de velocidad de movimiento de 0 a 7 km/h. Al mismo tiempo, el rendimiento medio al caminar es consistentemente mejor (de 148 a 457 Mbps) que mientras se conduce. Un rendimiento tan deficiente mientras se conduce no es sorprendente ya que las señales *mmWave* deben llegar al UE propagándose a través de la carrocería del automóvil (por ejemplo, parabrisas o ventanas laterales), que atenúa la intensidad de la señal provocando una degradación del rendimiento (Narayanan, Ramadan, Mehta, et al., 2020).

Bajo entornos urbanos típicos donde el UE está estacionario, el rendimiento promedio de 5G *mmWave* supera significativamente al del 5G de banda media. Sin embargo, el 5G comercial actual ofrece poca mejora de la latencia debido a su modelo de implementación no autónomo (NSA), que comparte gran parte de la infraestructura 4G existente con 5G. Otra causa de lo anterior, es el uso limitado de soporte en el borde de la red que ayude a acortar la latencia de extremo a extremo (Narayanan, Ramadan, Carpenter, et al., 2020).

La 5G *mmWave* comercial ofrece un rendimiento mucho más alto que 4G (mejora de $\sim 10x$). No obstante, debido a la naturaleza de las señales de onda milimétrica, el rendimiento en 5G *mmWave* muestra una variación mucho mayor que 4G incluso bajo una línea de visión clara. Por otra parte, las señales 5G *mmWave* se pueden bloquear fácilmente con las manos o el cuerpo humano. A pesar de lo anterior, en entornos urbanos realistas, los reflejos de la señal circundante a menudo pueden mitigar la degradación del rendimiento, permitiendo que 5G funcione bajo NLoS (Narayanan, Ramadan, Carpenter, et al., 2020).

En 5G *mmWave*, cuando el UE se está moviendo (usuario caminando o conduciendo) se suelen activar con frecuencia los *handoffs* 5G / 4G por el cambio de condición de la red o el tráfico de usuario. Incluso, en condiciones de baja movilidad (caminata), un teléfono inteligente puede experimentar más de 30 *handoffs* 5G / 4G en menos de 8 minutos. Una gran cantidad de los conmutadores pueden confundir las aplicaciones

(por ejemplo, la lógica de adaptación de la velocidad de bits de video) y brindar experiencias de usuario altamente inconsistentes (Narayanan, Ramadan, Carpenter, et al., 2020).

En comparación con 5G *mmWave*, 5G de banda media ofrece un mejor rendimiento en condiciones de movilidad debido a su radio omnidireccional. Por la misma razón, 4G también exhibe mayor estabilidad cuando el UE se está moviendo. Lo anterior indica la necesidad de utilizar conjuntamente 5G *mmWave* y 5G / 4G con radio omnidireccional en escenarios de movilidad (Narayanan, Ramadan, Carpenter, et al., 2020).

Los *handoffs* en 5G difieren de los de 4G / 3G, tanto en los *handoffs* horizontales como en los verticales. Un *handoff* horizontal (HHO) ocurre cuando la asociación del UE cambia de un panel 5G a otro panel 5G. De manera general un HHO ocurre cuando el cambio de asociación por parte del UE no implica que haya un cambio de tecnología (*Intra-Network Handoff*), tal y como se muestra en la figura 15. En 5G, los HHO pueden ocurrir frecuentemente debido a la menor cobertura de los paneles 5G en comparación con las torres 4G. Por otra parte, un *handoff* vertical (VHO) se activa cuando cambia la tecnología inalámbrica a la que se asocia el UE (*Inter-Network Handoff*), por ejemplo: de 5G a 4G y viceversa. Los VHO también son comunes en 5G ya que las señales son más inestables que en 4G (Narayanan, Ramadan, Carpenter, et al., 2020), (Goyal et al., 2017), (Zenalden et al., 2017),.

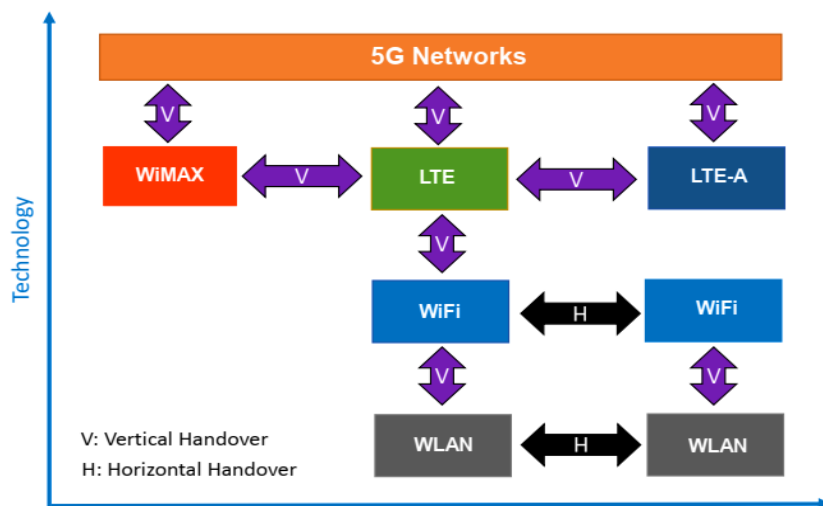


Figura 15. Tipos de *handoff* en redes heterogéneas (Zenalden et al., 2017).

Por el momento, las implementaciones de redes 5G no pueden proporcionar una cobertura perfecta en comparación con las redes 4G, que han sido optimizadas durante muchos años. Debido a lo anterior se

hace necesario el inter-funcionamiento entre las redes 5G y 4G para garantizar la experiencia del usuario y la continuidad del servicio. El rendimiento del inter-funcionamiento es también un aspecto muy importante para la selección de los modos SA y NSA (Samsung, 2021), (G. Liu et al., 2020), (Zakeri et al., 2020), (GSMA, 2018).

En la implementación de 5G NSA, el dispositivo UE siempre está anclado en la red LTE, y no hay inter-funcionamiento 4G / 5G (a nivel de núcleo de red). En el caso del servicio de voz este puede ser compatible con el servicio de voz sobre LTE (VoLTE) existente en 4G (Samsung, 2021), (G. Liu et al., 2020), (Zakeri et al., 2020), (GSMA, 2018).

Por otra parte, en la implementación de 5G SA, el dispositivo UE se ancla en la red 5G NR (5G *New Radio*) cuando su cobertura está disponible, mientras que el UE realiza un *handoff* a la red LTE cuando sale de la cobertura 5G. El inter-funcionamiento entre 4G y 5G se realiza a través de la interfaz N26. En el caso del servicio de voz de 5G SA este se puede prestar de dos formas: la primera es utilizando voz sobre NR (VoNR) y la segunda es utilizando el respaldo de EPS (*Evolved Packet System fallback*). Para VoNR, el servicio de voz es atendido directamente por 5G NR, y el UE se transfiere a la red LTE solo cuando se pierde la cobertura 5G. En el caso de VoNR, el servicio de voz se puede utilizar simultáneamente con otros servicios 5G; sin embargo, para EPS *fallback* el UE se conecta a la red LTE cuando se utiliza el servicio de voz, no siendo posible en este caso la utilización de los servicios de voz y datos 5G simultáneamente (Samsung, 2021), (G. Liu et al., 2020), (Zakeri et al., 2020), (GSMA, 2018).

Otro posible escenario es cuando el UE entra a un área de cobertura que no sea compatible con VoLTE y VoNR, por ejemplo una red 3G, entonces se utiliza SRVCC (*5G-Single Radio Voice Call Continuity*), que proporciona un servicio de voz continuo a través de UTRAN (Samsung, 2021).

En la figura 16 se muestra la continuidad del servicio de voz cuando el UE realiza VHO entre redes móviles 5G, 4G y 3G:

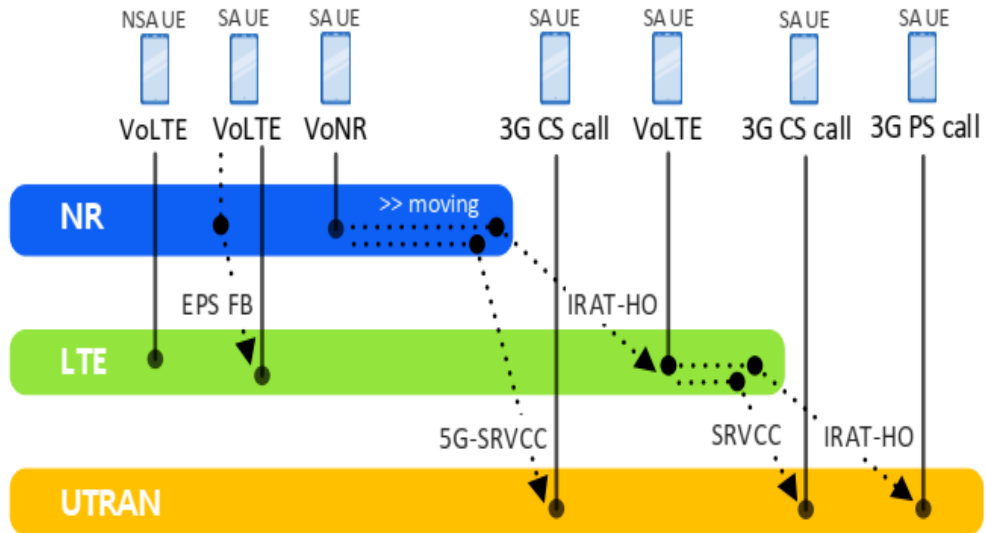


Figura 16. Continuidad del servicio de voz entre redes 5G, 4G y 3G (Samsung, 2021).

En la figura 17 se muestran las distintas opciones de despliegue de las redes 5G para los modos NSA y SA según el 3GPP:

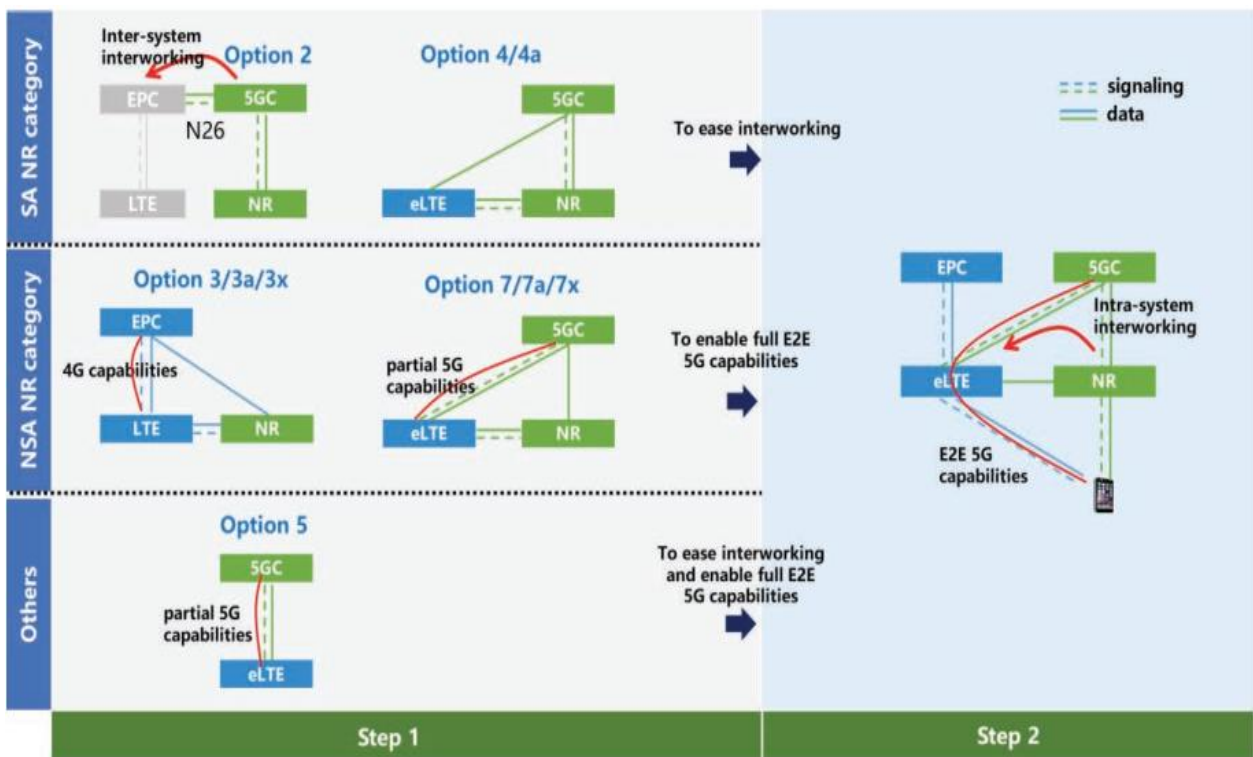
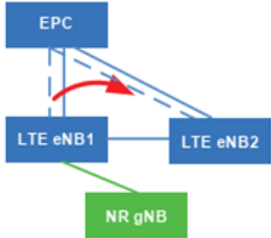
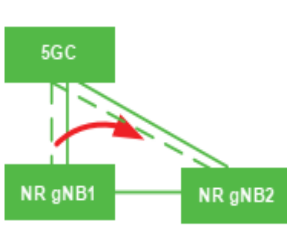
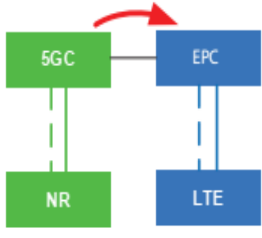


Figura 17. Arquitecturas 5G NSA / SA y potenciales caminos de migración para el inter-funcionamiento entre 4G y 5G (G. Liu et al., 2020).

En la tabla 4 se muestran los valores de la latencia y el tiempo de interrupción causado por el movimiento de un UE en diferentes escenarios NSA y SA. En el primer caso el UE se mueve desde la cobertura del NR gNB (conectado al LTE eNB1, desde donde se maneja la señalización) hacia la cobertura del LTE eNB2, se trata de un escenario NSA donde ocurre un *handoff* intrasistema EPC (*Evolved Packet Core*). En el segundo caso el UE se mueve desde la cobertura del NR gNB1 hacia el NR gNB2, se trata de un escenario SA donde ocurre un *handoff* intrasistema 5GC (*5G Core*). Por último, en el tercer caso, el UE pasa de la cobertura de una red 5G NR a la cobertura de una red LTE, se trata de un escenario SA donde ocurre un *handoff* intersistema (desde 5GC hacia EPC) (Samsung, 2021), (G. Liu et al., 2020), (Zakeri et al., 2020).

Según los resultados mostrados en la tabla 4, el modo NSA presenta mejor desempeño porque el interfuncionamiento entre 4G y 5G en NSA se lleva a cabo a través de los *handoff* intrasistema (aun cuando ocurra un VHO entre celdas 4G / 5G todo ocurre dentro del sistema EPC). Sin embargo, en el modo SA para que haya interfuncionamiento entre 4G y 5G se necesita de la ocurrencia de *handoff* intersistemas, lo cual aumenta la latencia del *handoff* y el tiempo de interrupción del servicio (Samsung, 2021), (G. Liu et al., 2020), (Zakeri et al., 2020).

Tabla 4. Análisis del desempeño del *handoff* en diferentes escenarios (G. Liu et al., 2020).

Aspectos	NSA	SA sin interfuncionamiento	SA con interfuncionamiento
Diagrama del escenario			
Latencia del <i>handoff</i>	<100 ms	<100 ms	400 ~500 ms
Interrupción del servicio	20 ms	0~20 ms	30~50 ms
Duración de la paginación inalcanzable	0	0	350~450 ms

2.6 Aplicación del aprendizaje automático a la gestión de redes 5G

El aprendizaje automático, tiene aplicación en una amplia gama de campos: procesamiento de imágenes, audio, finanzas, economía, análisis de comportamiento social, gestión de redes de telecomunicaciones,

siendo esta última una de las aplicaciones más prometedoras de esta rama de la inteligencia artificial (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

La evolución del ML como disciplina ha sido tal que actualmente permite que las redes de telecomunicaciones aprendan y extraigan conocimiento al interactuar con los datos. Esto último es de vital importancia para nuevos estándares de redes inalámbricas tales como 5G (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

Hasta el momento, el desarrollo del aprendizaje automático y de las redes de telecomunicaciones se había llevado a cabo como campos de investigación diferentes, pero la aplicación del ML en las redes 5G ha demostrado el potencial presente y futuro de esta combinación de paradigmas y tecnologías. Lo anterior se ha hecho evidente en los servicios basados en la ubicación, almacenamiento caché perimetral, redes contextuales, análisis de datos masivos, informática perimetral móvil y control de tráfico de red (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

El aprendizaje automático es ideal en problemas complejos donde la solución requiere de mucho ajuste manual o en problemas donde no existe una solución tradicional. Los problemas anteriores pueden abordarse sustituyendo el software convencional que contiene un gran número de reglas por software (que contenga rutinas de ML) que sea capaz de aprender automáticamente de los datos. Dentro de las diferencias entre los algoritmos de ML y los algoritmos cognitivos tradicionales se encuentra la extracción automática de características, la detección de anomalías, la predicción de escenarios futuros, la adaptación a entornos fluctuantes, la obtención de información sobre problemas complejos con grandes cantidades de datos y el descubrimiento de patrones (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

En el caso de las redes inalámbricas y móviles, existe un gran número de parámetros que se calculan mediante métodos heurísticos porque no existe una solución analítica o porque esta no es viable debido a su complejidad y costos asociados. En tales escenarios los algoritmos de ML pueden contribuir a la solución mediante la predicción y estimación de parámetros y funciones basado en los datos disponibles (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

No obstante a las ventajas que supone el uso de ML, ésta tecnología presenta desafíos que limitan su despliegue en comunicaciones inalámbricas, tales como: la interpretabilidad de los resultados, la dificultad para obtener datos relevantes, la potencia de cálculo requerida, la complejidad introducida, los largos tiempos de entrenamiento de algunos algoritmos, entre otros, que provocan que el costo, tiempo, latencia

y demora introducidos son incompatibles con algunas aplicaciones en tiempo real (Morocho-Cayamcela et al., 2019).

Se puede afirmar que, pese a los desafíos por resolver, ya el aprendizaje automático es ampliamente utilizado en el modelado de diversos problemas técnicos de sistemas de última generación, como comunicaciones de baja latencia, MIMO a gran escala, redes de dispositivo a dispositivo (D2D), redes heterogéneas constituidas por femtoceldas y celdas pequeñas, redes vehiculares, entre otros (Tayyaba et al., 2020), (T. Li et al., 2020), (Bega et al., 2019), (Morocho-Cayamcela et al., 2019), (Asadi et al., 2018), (Balevi and Gitlin, 2018), (J. Li et al., 2018), (Klautau et al., 2018), (Sim et al., 2018), (Ben Yahia et al., 2017), (Jiang et al., 2017).

En la figura 18, se muestra la relación a alto nivel entre aprendizaje profundo (*Deep Learning*), aprendizaje automático e inteligencia artificial (IA). Básicamente la IA lo constituye cualquier técnica que permita a las computadoras imitar el comportamiento humano; el ML son aquellos algoritmos que utilizan técnicas estadísticas para permitir que las máquinas “aprendan” a través de experiencias; y el *Deep Learning* es una subcategoría de ML que implica la utilización de redes neuronales de varias capas para llevar a cabo la ejecución de los algoritmos (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

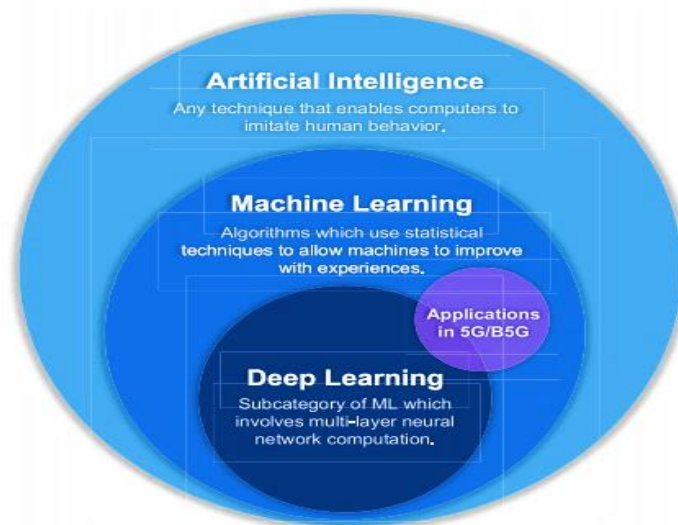


Figura 18. Diagrama de Venn de la relación entre inteligencia artificial, aprendizaje automático y el aprendizaje profundo, así como su aplicación en redes 5G y futuras (Morocho-Cayamcela et al., 2019).

El aprendizaje automático se subdivide según el nivel de supervisión que el procedimiento de ML requiere en la etapa de entrenamiento. Las tres categorías de ML existentes son: aprendizaje supervisado, aprendizaje sin supervisión, y el aprendizaje por refuerzo. En la figura 19 se muestra la comparación entre

la programación tradicional y los tres tipos de aprendizaje automático (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017).

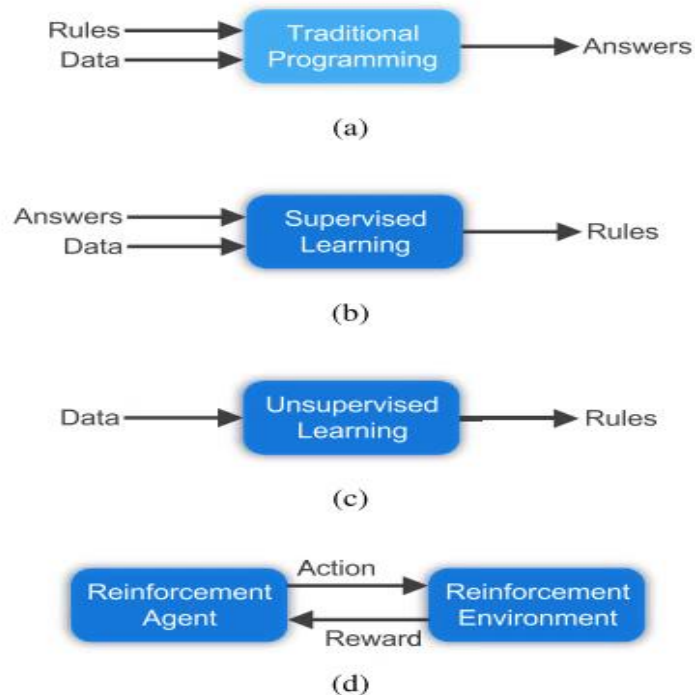


Figura 19. Comparación de la programación tradicional con los tres tipos de aprendizaje automático: (a) programación tradicional, (b) aprendizaje supervisado, (c) aprendizaje no supervisado y (d) aprendizaje por refuerzo (Morocho-Cayamcela et al., 2019).

2.6.1 Aprendizaje supervisado

El aprendizaje supervisado se basa en modelos conocidos que pueden soportar la estimación de parámetros desconocidos. Estos modelos pueden utilizarse para la estimación de canales en MIMO masivo, detección de espectro y detección de espacios en blanco en radio cognitiva y filtrado adaptativo en el procesamiento de señales para comunicaciones 5G. También se pueden utilizar en aplicaciones de capa superior, como aquellas que brindan servicio de ubicación y comportamiento de los usuarios de dispositivos móviles, que pueden ayudar a los operadores de red a mejorar la calidad de sus servicios. En la tabla 5, se muestra un resumen de los algoritmos de aprendizaje supervisado y su aplicación a la gestión de las redes 5G (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017):

Tabla 5. Algoritmos de aprendizaje supervisado y su aplicación en 5G (elaboración propia).

Técnicas de aprendizaje	Características principales	Aplicación en 5G
Modelos de Regresión	- Estimar las relaciones entre las variables - Regresión lineal y logística	- Aprendizaje (Energía)
KNN	- Voto mayoritario de vecinos	- Aprendizaje (Energía)
SVM	- Mapeo no lineal para alta dimensión - Clasificación separada de hiperplano	- Aprendizaje (Canal MIMO)
Aprendizaje Bayesiano	- Cálculo de distribución a posteriori - GM, EM, HMM	- Aprendizaje (MIMO masivo) - Aprendizaje (Espectro cognitivo)

2.6.2 Aprendizaje no supervisado

El aprendizaje no supervisado se basa en los propios datos de entrada de manera heurística, o sea el modelo se ajusta a las observaciones. Se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. Estos modelos se pueden utilizar en el agrupamiento de celdas en redes cooperativas ultradensas, en la asociación de puntos de acceso en redes WiFi ubicuas, en la agrupación heterogénea de estaciones base y en el equilibrio de carga en HetNets. También se puede aplicar en la detección de fallas, anomalías, intrusiones y en la clasificación del comportamiento de los usuarios. En la tabla 6, se muestra un resumen de los algoritmos de aprendizaje no supervisado y su aplicación a la gestión de las redes 5G (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017):

Tabla 6. Algoritmos de aprendizaje no supervisado y su aplicación en 5G (elaboración propia).

Técnicas de aprendizaje	Características principales	Aplicación en 5G
Agrupamiento K-means	- Agrupación de particiones - Algoritmo de actualización iterativo	- Redes heterogéneas
PCA	- Transformación ortogonal	- <i>Smart Grid</i>
ICA	- Revelar factores independientes ocultos	- Aprendizaje (Espectro Radio Cognitiva)

2.6.3 Aprendizaje por refuerzo

El aprendizaje por refuerzo se basa en una dinámica de aprendizaje iterativo y proceso de toma de decisiones. Se puede utilizar para inferir la toma de decisiones de los usuarios móviles en condiciones de red desconocidas, acceso al canal bajo condiciones de disponibilidad desconocidas en el uso compartido del espectro, para la asignación de recursos distribuidos bajo condiciones desconocidas de calidad de recursos en redes de femtoceldas, y en la asociación de estaciones base bajo un estado de energía desconocido en redes de recolección de energía. En la tabla 7, se muestra un resumen de los algoritmos

de aprendizaje por refuerzo y su aplicación a la gestión de las redes 5G (Morocho-Cayamcela et al., 2019), (Jiang et al., 2017):

Tabla 7. Algoritmos de aprendizaje por refuerzo y su aplicación en 5G (elaboración propia).

Técnicas de aprendizaje	Características principales	Aplicación en 5G
MDP / POMDP	<ul style="list-style-type: none"> - Maximización de la ecuación de Bellam - Algoritmo de iteración de valor 	- Recolección de energía
Aprendizaje Q	<ul style="list-style-type: none"> - Modelo de transición de sistema desconocido - Maximización de la función Q 	- Femtoceldas y celdas pequeñas
MAB	<ul style="list-style-type: none"> - Exploración vs. explotación 	- Redes dispositivo a dispositivo

2.7 Redes neuronales artificiales

Las redes neuronales artificiales (ANN) tienen como objetivo reproducir el comportamiento complejo del aprendizaje del cerebro humano. La popularidad de las ANN se debe al hecho de que la mayor parte de la optimización del algoritmo se basa en el método de descenso de gradiente, que se puede implementar computacionalmente de manera eficiente. Para ello, se ha construido hardware específico para facilitar y para mejorar el rendimiento de las redes neuronales, como son las GPU y las TPU. Estas están ampliamente disponibles en el mercado y pueden asociarse fácilmente con infraestructuras en la nube que permiten gestionar grandes volúmenes de datos y operaciones de *Big Data* (Fernández, 2020), (Trinh, 2020), (Géron, 2019).

En una ANN el elemento básico es la neurona. La neurona más simple está representada por el perceptrón, que realiza operaciones matemáticas básicas. Los perceptrones se pueden combinar para formar una capa, y las capas se puede conectar para construir redes complejas de múltiples capas (Fernández, 2020), (Trinh, 2020), (Géron, 2019), (Peyré, 2019).

Para un conjunto de datos de m pares de entrada y salida $(x_i, y_i), i = 1, \dots, m$: el perceptrón más simple toma una entrada $x_i = [x_1, x_2, \dots, x_n]$, y produce una salida $\hat{y}_i(w, x_i)$, realizando una combinación lineal con su peso w_i y la función de activación no lineal $f(\cdot)$. También, se le agrega una entrada adicional permanentemente activada a la que se la denominaba *bias* o sesgo (denotada con la letra b). En la ecuación 1 se expone matemáticamente lo anterior y en la figura 20 se muestra gráficamente el modelo del perceptrón (Trinh, 2020), (Géron, 2019), (Peyré, 2019).

$$\hat{y}_i = f\left(\sum_{i=1}^m w_i \cdot x_i + b\right) \quad (1)$$

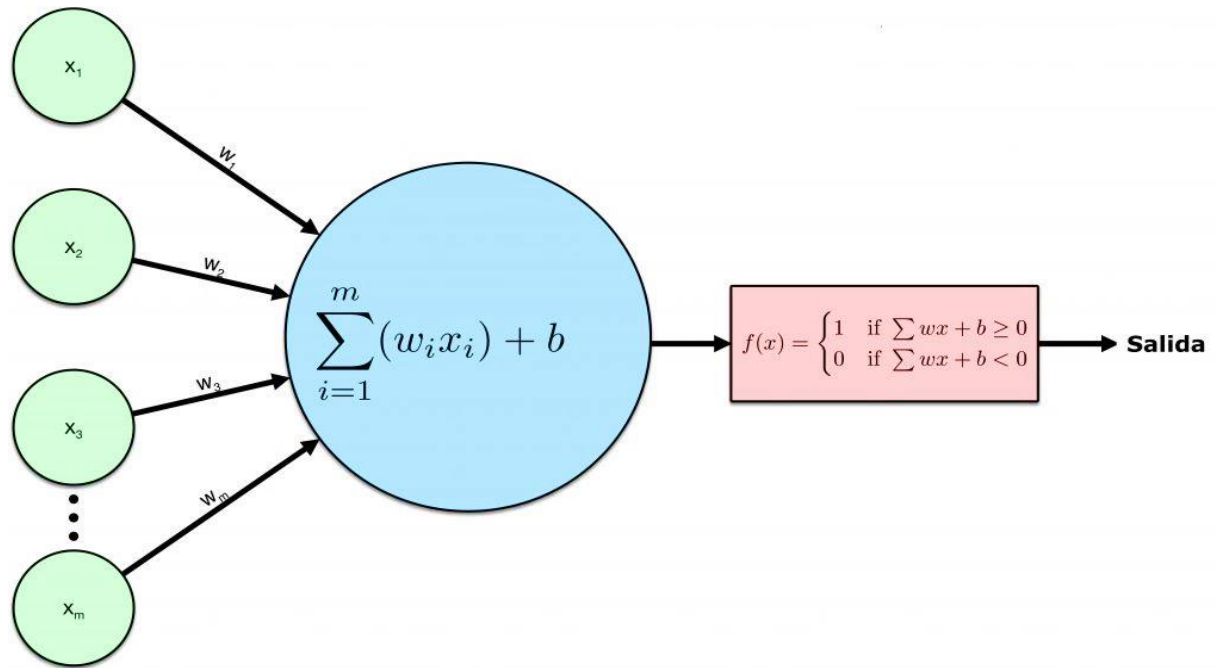


Figura 20. Modelo del perceptrón (Alvarez, 2018).

Se pueden combinar varios perceptrones para formar una arquitectura en capas, normalmente denominado perceptrón multicapa (MLP), tal y como se muestra en la figura 21. En el caso más simple, cada neurona realiza las mismas operaciones matemáticas, utilizando la misma función de activación no lineal. Por lo general, la entrada y la salida se consideran, respectivamente, la primera y la última capa de la red neuronal. Las capas intermedias se denominan capas ocultas, las neuronas de esta capa se denominan neuronas ocultas y su número define el tamaño de la capa. El término oculto se refiere al hecho de que no se tiene el valor real en el proceso de entrenamiento para estas unidades, en contraste con las capas de entrada y salida, de las cuales se conoce los valores reales de x_i , y_i (Trinh, 2020), (Peyré, 2019).

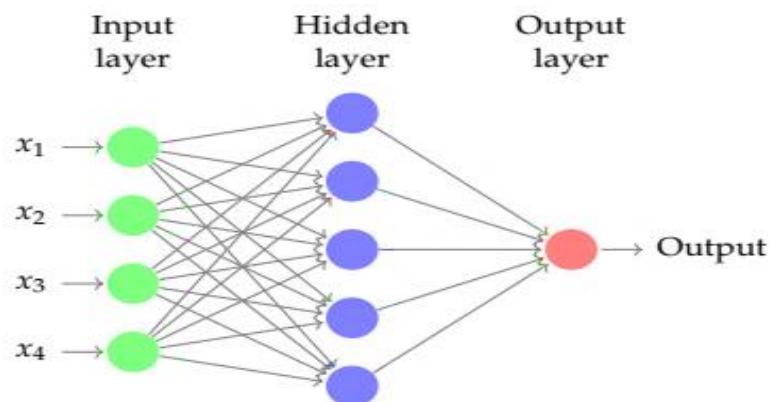


Figura 21. Ejemplo de arquitectura MLP totalmente conectada con 1 capa oculta y 5 neuronas ocultas (Trinh, 2020).

La potencia y complejidad de las ANN está dada por su comportamiento no lineal, el cual se logra de acuerdo con el tipo de función de activación usada por las neuronas (Gildardo, 2006). Las funciones de activación se usan para propagar la salida de los nodos de una capa hacia la siguiente capa. Uno de los grandes avances en el campo de las ANN fue el desarrollo de la neurona *sigmoide* o *sigmoidal*; esta neurona es muy parecida al perceptrón en su estructura, pero al contrario de este (el perceptrón admite valores de entrada de 0 y 1) puede tener entradas de cualquier valor numérico. Esta neurona utiliza como función de activación la función *sigmoid*, la cual viene representada por la ecuación 2:

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

La función *sigmoid* también es conocida como función logística, está en un rango de valores de salida entre 0 y 1, por lo que la salida es interpretada como una probabilidad. Si se evalúa la función con valores de entrada muy negativos la función será igual a 0, si se evalúa en $x = 0$ la función es igual a 0.5 y para valores altos es aproximadamente igual a 1. Por lo general esta función se usa en la última capa para clasificar datos en dos categorías. Actualmente la función *sigmoid* no es una función muy utilizada debido a que no está centrada, esto provoca el problema de desaparición del gradiente el cual afecta el aprendizaje y entrenamiento de la neurona (Freire and Silva, 2019).

Otra función de activación de la familia *sigmoid* es la tangente hiperbólica o *tanh*, que tiene un rango de valores de salida entre -1 y 1. Esta función es un escalamiento de la función logística, por lo que a pesar de que está centrada también tiene el problema de desaparición del gradiente (Géron, 2019), (Gildardo, 2006).

La función de unidad lineal rectificada: $\text{ReLU}(x) = \max(0, x)$ es continua pero desafortunadamente no diferenciable en $x = 0$ (la pendiente cambia abruptamente, lo que puede hacer que el gradiente descendente rebote), y su derivada es 0 para $x < 0$. En la práctica funciona muy bien y tiene la ventaja de ser rápida de calcular. Otra ventaja importante de esta función es el hecho de que no tiene un valor máximo de salida, lo cual ayuda a reducir algunos problemas durante el cálculo del gradiente descendente (Géron, 2019).

En la figura 22 se muestran las funciones *step*, *sigmoid*, *tanh*, ReLU y sus derivadas:

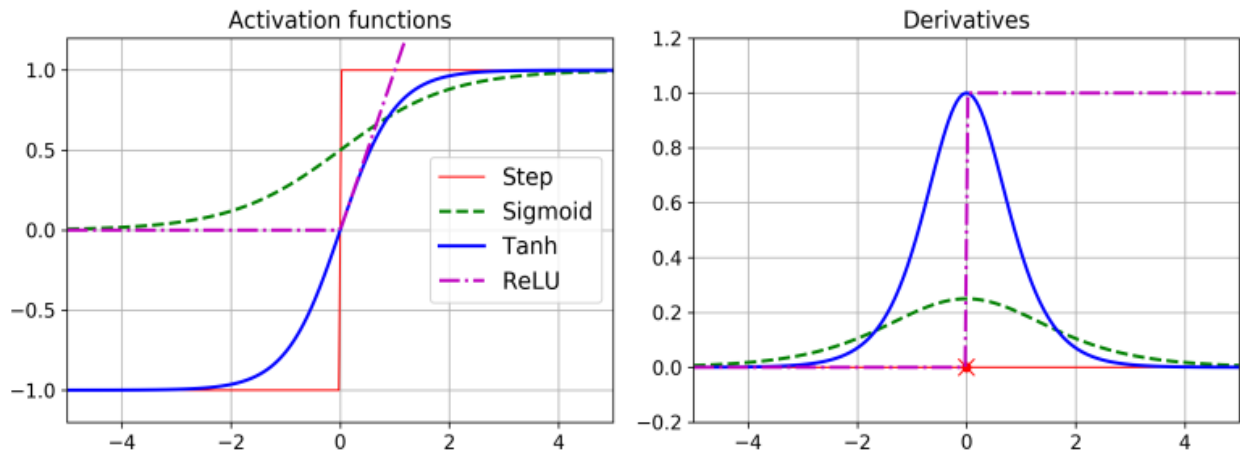


Figura 22. Funciones de activación *step*, *sigmoid*, *tanh*, ReLU y sus derivadas (Géron, 2019).

Otra función de activación muy utilizada es *softmax* (o función exponencial normalizada) la cual convierte un vector de K valores reales en un vector de K valores reales que suman 1. Los valores de entrada pueden ser positivos, negativos, cero o mayores que uno, pero la función *softmax* los transforma en valores entre 0 y 1, para que puedan interpretarse como probabilidades. A veces a *softmax* se le denomina función *softargmax* o regresión logística de clases múltiples, debido a que es una generalización de la regresión logística que se puede utilizar para la clasificación de clases múltiples (tienen que ser mutuamente excluyentes), y su ecuación es muy similar a la función *sigmoid* (Trinh, 2020), (Géron, 2019).

Es habitual añadir una función *softmax* como capa final de la red neuronal. Lo anterior se debe a que muchas redes neuronales multicapa presentan una penúltima capa que genera a la salida valores reales que no están convenientemente escalados, por lo que se dificulta el trabajo con ellos. La utilidad de aplicar *softmax* a la última capa está dada en que convierte dichos valores reales en una distribución de probabilidad normalizada (Trinh, 2020), (Géron, 2019).

Dentro de las redes ANN, se pueden destacar tres tipos de redes, las redes neuronales prealimentadas (*Feed Forward Networks*, FFN), las redes neuronales recurrentes (*Recurrent Neural Networks*, RNN) y las redes neuronales convolucionales (*Convolutional Neural Networks*, CNN) (Vinayakumar et al., 2017).

2.7.1 Redes neuronales prealimentadas

Las FNN son de las primeras redes neuronales diseñadas y son las más sencillas. En estas redes solo existe propagación en un solo sentido sin realimentación. La propagación se produce desde los nodos de entrada, pasando por los nodos ocultos y terminando en los nodos de salida (Vinayakumar et al., 2017), (Haykin, 2001).

En las FNN las operaciones de cada capa se pueden denotar con una matriz de peso W_l , donde l se refiere al número de la capa (se considera $l = 0$ para la capa de entrada). Las dimensiones de esta matriz son igual al número de entradas multiplicado por el número de neuronas ocultas. En cada capa l , se denota la combinación lineal con Z_l y la activación con A_l , tal y como se indica en las ecuaciones 3 y 4 (Trinh, 2020), (Vinayakumar et al., 2017), (Haykin, 2001).

$$Z_l = W_l \cdot x_i + b_l \quad (3)$$

$$A_l = f(Z_l) \quad (4)$$

La elección de la función de activación $f(\cdot)$ depende del problema que se esté intentando resolver. En caso de un problema de regresión, se pueden usar las funciones *tanh*, *sigmoid* o *ReLU*, mientras que para problemas de clasificación es común aplicar la función *softmax* (Trinh, 2020), (Vinayakumar et al., 2017), (Haykin, 2001).

2.7.2 Redes neuronales recurrentes

Las RNN son una clase de redes neuronales para procesar datos secuenciales y que pueden predecir el futuro. Estas redes pueden analizar datos de series de tiempo y realizar predicciones tales como precios de acciones financieras, anticipación de trayectorias en vehículos autónomos, pronóstico del tráfico en redes de telecomunicaciones, pronóstico de variables meteorológicas, etc. Pueden funcionar con secuencias de entrada de longitudes arbitrarias (en lugar de con entradas de tamaño fijo como sucede en otras redes), por ejemplo: pueden tomar oraciones, documentos o muestras de audio como entrada, lo que las hace extremadamente útiles para el procesamiento del lenguaje natural (NPL, por sus siglas en inglés), sistemas como traducción automática, análisis de voz a texto, entre otros (Trinh, 2020), (Géron, 2019), (Olah, 2015).

La principal diferencia entre una red típica multicapa y una RNN es que, en vez de alimentar completamente las conexiones, una RNN puede tener conexiones que retroalimentan las capas anteriores (o a la misma capa); esta retroalimentación permite que las RNN mantengan una memoria de entradas pasadas, o sea permite que la información persista. Las RNN pueden verse también como múltiples copias de la misma red, cada una de ellas pasando información a su sucesora. En cada momento del tiempo t , la red recibe como entrada tanto x_t como su propia salida h_{t-1} en el instante $t-1$. La representación expandida de una RNN se muestra en la figura 23, donde A es una red neuronal, con entrada x_t y salida h_t (Trinh, 2020), (Géron, 2019), (Olah, 2015).

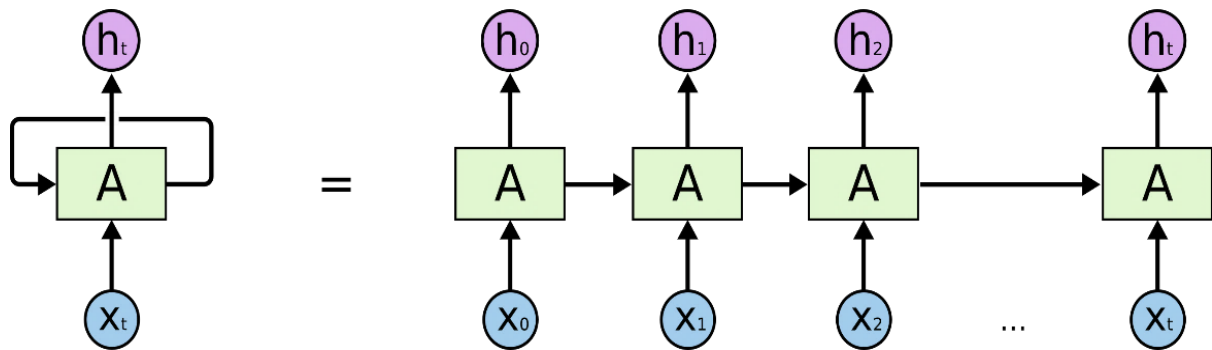


Figura 23. Arquitectura RNN expandida (Olah, 2015).

En teoría, las RNN son capaces de manejar dependencias a largo plazo, sin embargo, en la práctica cuando aumenta el número de capas el entrenamiento de la red utilizando el algoritmo de retro propagación puede ser difícil. Lo anterior se debe a que el gradiente tiende a hacerse más pequeño a medida que se avanza hacia atrás a través de las primeras capas ocultas; lo cual significa que las neuronas de las capas anteriores aprenden mucho más lentamente que las neuronas en capas posteriores. El problema anterior es conocido como desvanecimiento del gradiente y está presente en todos los tipos de redes neuronales, pero es más perjudicial en redes cuyo objetivo es el pronóstico de series temporales (Pascanu et al., 2013), (Olah, 2015).

Para evitar el problema del desvanecimiento del gradiente que se presenta en las redes RNN, Hochreiter y Schmidhuber (1997) diseñaron las redes LSTM (*Long Short Term Memory*), que son un tipo particular de RNN. La capacidad de aprender dependencias a largo plazo se debe a la estructura de las unidades LSTM, que, a diferencia de las neuronas simples, incorpora puertas que regulan el proceso de aprendizaje. Las capas ocultas de una unidad LSTM son celdas de memoria (MC); estas tienen la capacidad de almacenar u olvidar la información sobre los estados pasados de la red mediante el uso de estructuras llamadas

compuertas, que consisten en una cascada de una neurona con función de activación *sigmoid* y un bloque de multiplicación. En esta arquitectura, la salida de cada celda de memoria depende de la secuencia completa de estados pasados, lo que hace que las redes LSTM sean adecuadas para el procesamiento de series de tiempo con dependencias de tiempo prolongadas (Olah, 2015), (Pascanu et al., 2013), (Gers et al., 2000). En la figura 24 se muestra el diagrama de una celda LSTM estándar:

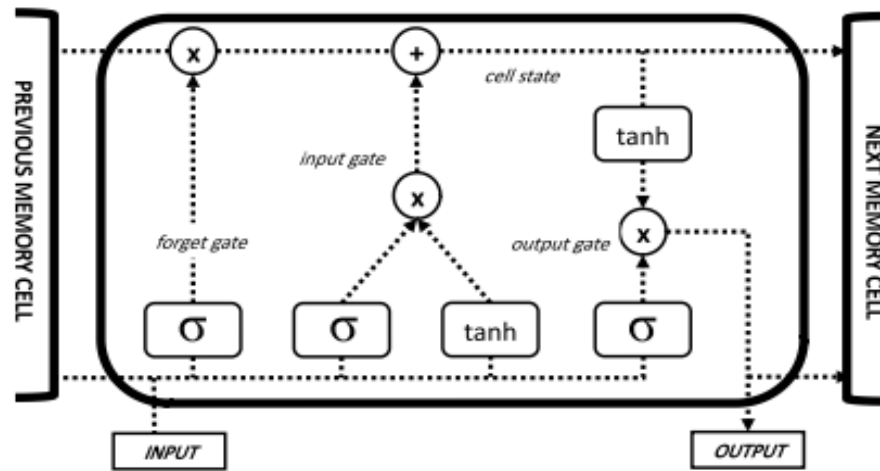


Figura 24. Unidad LSTM estándar (Trinh, 2020).

Las operaciones básicas se realizan mediante la puerta de entrada i_t , la puerta de olvido f_t y la puerta de salida o_t . A continuación se muestran las ecuaciones que describen las operaciones (Trinh, 2020), (Géron, 2019), (Wang et al., 2017), (Zhang and Patras, 2017), (Olah, 2015), (Graves and Jaitly, 2014), (Zaremba and Sutskever, 2014):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) = \sigma(W_{x_i} \cdot x_t + W_{h_i} \cdot h_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) = \sigma(W_{x_f} \cdot x_t + W_{h_f} \cdot h_{t-1} + b_f) \quad (6)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) = \sigma(W_{x_o} \cdot x_t + W_{h_o} \cdot h_{t-1} + b_o) \quad (7)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) = \tanh(W_{x_c} \cdot x_t + W_{h_c} \cdot h_{t-1} + b_c) \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (9)$$

$$h_t = o_t \odot \tanh(c_t) \quad (10)$$

En las ecuaciones anteriores, $\sigma(\cdot)$ es la función *sigmoid*; $\tanh(\cdot)$ es la función tangente hiperbólica; W es la matriz de pesos; b es el sesgo de las puertas i , f , o y del estado de celda c ; el subíndice t es el índice de tiempo y el símbolo \odot denota el producto de Hadamard (producto por elemento) de dos vectores.

La unidad LSTM combina la salida de la unidad LSTM anterior (h_{t-1}) con la entrada actual x_t usando las puertas de entrada, salida y de olvido para actualizar la memoria de la celda (ver ecuaciones 5, 6, 7 y 8). Las variables i_t y f_t representan respectivamente la información que deben guardarse u olvidarse del pasado y de la entrada actual. El estado de celda c_t se actualiza sumando el estado de celda anterior c_{t-1} y el estado de celda candidato \tilde{c}_t , ponderado respectivamente con f_t e i_t (ver ecuación 9). Después, se obtiene la salida actual h_t aplicando a c_t la función *tanh* y multiplicándola por o_t (ver ecuación 10). Por último, la salida h_t se pasa a la siguiente unidad LSTM y se combina con la entrada x_{t+1} (Trinh, 2020), (Géron, 2019), (Wang et al., 2017), (Zhang and Patras, 2017), (Olah, 2015), (Graves and Jaitly, 2014), (Zaremba and Sutskever, 2014).

2.7.3 Redes neuronales convolucionales

Las CNN son un tipo de redes neuronales que se especializan en procesar datos que tienen una topología similar a una cuadrícula (ejemplo: una imagen). Estas redes son capaces de capturar con éxito las dependencias espaciales y temporales en una imagen mediante la aplicación de filtros relevantes, también llamados núcleos (*kernel*) de convolución. La arquitectura se adapta mejor al conjunto de datos de imágenes debido a la reducción en el número de parámetros involucrados y la reutilización de pesos (MathWorks, 2021), (Trinh, 2020), (Géron, 2019).

En matemáticas, la convolución es una operación matemática que toma dos funciones y genera una nueva función, la cual expresa como la forma de una es modificada por la segunda. La primera función usualmente es la función sobre la cual se ejerce el cambio (entrada) mientras que la segunda función es la que modifica a la primera (núcleo o *kernel*). En el dominio del aprendizaje automático, tanto la entrada como el *kernel* son matrices, por lo que la convolución es una convolución de matrices (MathWorks, 2021).

Las CNN están compuestas por una capa de entrada, una capa de salida y por una o más capas ocultas (convolución y agrupación), tal y como se muestra en la figura 25. La capa convolucional se compone de tres partes principales y una operación extra (MathWorks, 2021), (Trinh, 2020), (Géron, 2019):

- ✓ La entrada: se compone de múltiples canales que entran en forma de matrices.
- ✓ El *kernel*: es el filtro que se aplica sobre los canales de entrada.
- ✓ La salida: es el resultado de aplicar la convolución del *kernel* sobre los canales de entrada después de aplicar la función de activación de no linealidad (normalmente la función ReLU).
- ✓ Agrupación o submuestreo: es una operación extra que consiste en reducir la dimensión de los canales resultantes con el objetivo de resumir estadísticamente los valores cercanos y de esta manera mejorar el tiempo de entrenamiento de la red. Dentro de las funciones de agrupamiento más utilizadas están *Max-Pooling* y *Average-Pooling*.

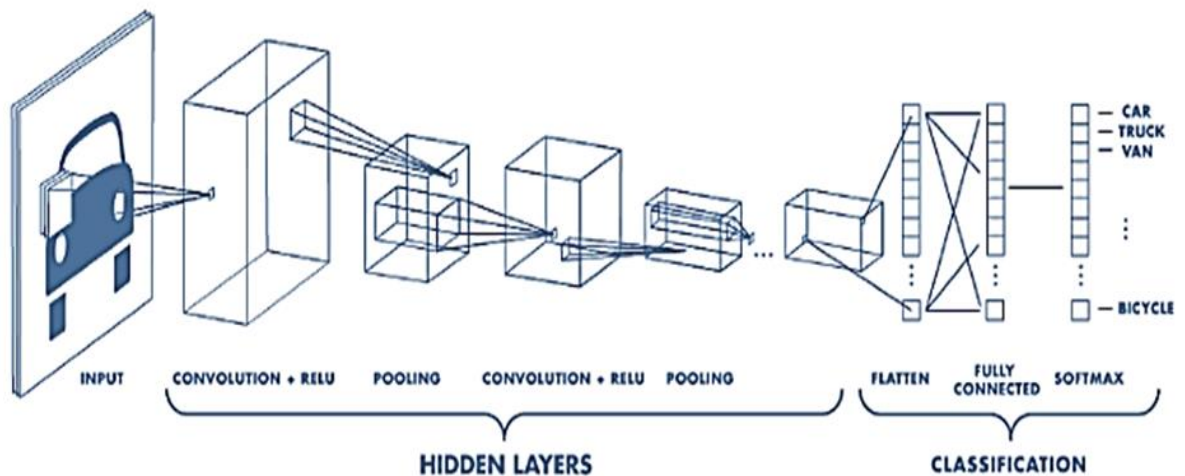


Figura 25. Arquitectura de una CNN (MathWorks, 2021).

El *kernel* actúa como un filtro cuyos pesos se reutilizan (pesos compartidos), esto hace que la estructura de conectividad de la red sea escasa por lo que un pequeño conjunto de parámetros es suficiente para mapear la entrada. Debido a lo anterior, la complejidad computacional de una CNN es considerablemente

menor con respecto a otras redes neuronales, tales como las redes LSTM (Trinh, 2020), (Géron, 2019), (Gadaleta and Rossi, 2018).

Generalmente, cuando se hace referencia a una red neuronal convolucional se asume que es una CNN bidimensional (CNN 2D), pero también existen las CNN unidimensionales y tridimensionales. A continuación, se resumen las características principales de estas redes neuronales (Géron, 2019), (Verma, 2019), (Lane, 2018):

- ✓ CNN 1D: el *kernel* se mueve en una dirección. Los datos de entrada y salida son bidimensionales (una dimensión espacial y una dimensión temporal). Se utilizan principalmente para procesar datos de series temporales.
- ✓ CNN 2D: el *kernel* se mueve en dos direcciones. Los datos de entrada y salida son tridimensionales (dos dimensiones espaciales y una dimensión correspondiente al canal de color). Se utilizan principalmente para procesar datos de imágenes.
- ✓ CNN 3D: el *kernel* se mueve en tres direcciones. Los datos de entrada y salida tienen cuatro dimensiones (tres dimensiones espaciales y una dimensión correspondiente al número de canales de color). Se utilizan principalmente para procesar imágenes 3D tales como resonancia magnética, tomografías computarizadas, video, etc.

En la figura 26 se muestra la convolución de una matriz 2D de tamaño 8x8 con un *kernel* 2D de tamaño 3x3:

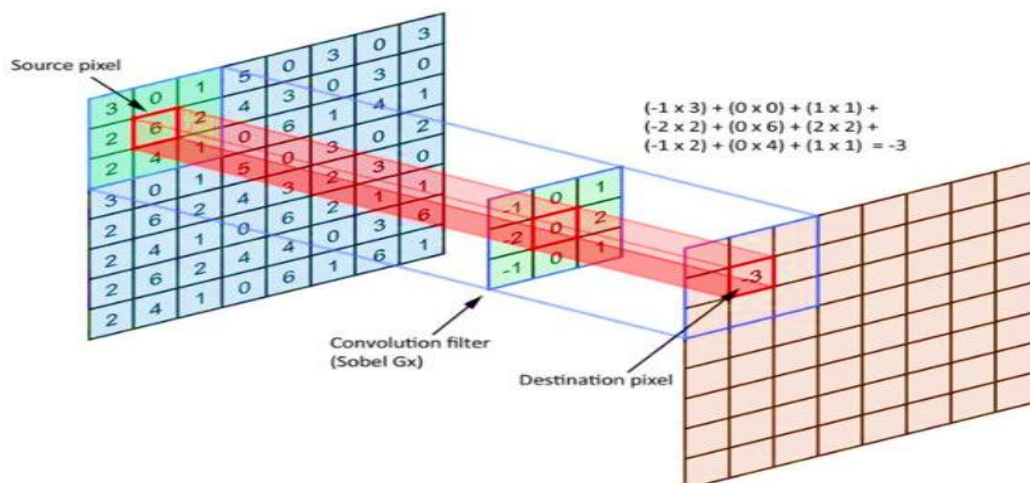


Figura 26. Convolución 2D utilizando un *kernel* 3x3 (Roman, 2020).

Las CNN 2D ganaron gran popularidad debido a los éxitos obtenidos en el campo de la visión por computadora, donde se utilizan imágenes como entradas, las cuales tienen dos dimensiones espaciales (alto y ancho). Sin embargo, recientemente se han obtenido muchos éxitos en la aplicación de las CNN 1D a las tareas de procesamiento del lenguaje natural (NLP), donde se utiliza texto como entradas, y este tiene patrones a lo largo de una sola dimensión espacial. Las CNN 1D se pueden utilizar como alternativas más eficientes a las RNN tradicionales, tales como LSTM y GRU, en el modelado de series de tiempo tanto univariantes como multivariantes (MathWorks, 2021), (Géron, 2019), (Verma, 2019).

2.8 Conclusiones

En este capítulo se presentaron los aspectos teóricos sobre redes 5G; también se abordó la aplicación del aprendizaje automático a la gestión de estas redes. Se puede concluir lo siguiente:

- ✓ Las redes 5G presentan tres casos de uso genéricos fundamentales definidos por la ITU: eMBB, mMTC y urLLC.
- ✓ La segmentación de las redes 5G es una característica vital para cumplir con los requerimientos de desempeño de los presentes y futuros servicios de telecomunicaciones.
- ✓ Entre los aspectos del sistema 5G que se ven afectados por la adopción de la segmentación de red se encuentran los procedimientos en torno a la gestión y la aplicación de los SLA.
- ✓ Dentro de los factores de impacto sobre el *throughput* en 5G, el *handoff* vertical es uno de los que mayor impacto tiene, sobre todo en los escenarios de alta movilidad.
- ✓ El empleo de técnicas de ML es un factor clave para mejorar la gestión de las redes 5G.

En el próximo capítulo, se propone un modelo para la predicción de las violaciones de los SLA a partir del pronóstico del *throughput* en aplicaciones de red del segmento eMBB.

Capítulo 3. Propuesta del modelo de predicción

En este capítulo se presenta la propuesta de un modelo para la predicción de las violaciones de los SLA en aplicaciones de red que operen en el segmento de red genérico eMBB. Para ello se procede a diseñar una solución de ML utilizando redes neuronales que será capaz de pronosticar el *throughput*, el cual es el KPI de mayor relevancia en el segmento eMBB. A partir de la predicción del *throughput*, el modelo realiza la predicción de las violaciones de los SLA. También se aborda la combinación de la predicción del *throughput* y el VHO para la prevención de violaciones de los SLA en escenarios de alta movilidad.

3.1 Requisitos para la predicción del *throughput*

Los modelos de aprendizaje automático basados en datos se han utilizado desde hace tiempo en la predicción del *throughput*, tanto en redes cableadas como en redes inalámbricas (Narayanan, Ramadan, Mehta, et al., 2020), (Vinayakumar et al., 2017), (Mirza et al., 2007), (He et al., 2005). En el caso de las redes móviles 5G, la existencia de múltiples factores de impacto sobre el *throughput* y su compleja interacción, hacen mucho más difícil a los modelos de ML realizar la predicción. A continuación, se expondrá la selección de cada uno de los componentes y características necesarios para el diseño e implementación de la solución que se propone.

3.1.1 Selección del *dataset*

Los modelos de aprendizaje profundo (tanto supervisados como no supervisados) requieren conjuntos de datos (*dataset*) para su entrenamiento y pruebas. Sin embargo, adquirir un buen *dataset*, en algunos casos, sigue siendo un desafío considerable. En la tabla 8 se muestra un resumen de los diferentes *dataset* utilizados en varias investigaciones previas, donde se aplica la IA a la solución de problemas en las redes 5G (Santos et al., 2020).

Según el estudio llevado a cabo en (Santos et al., 2020), la mayoría de los trabajos donde se aplica la IA a la solución de problemas en las redes 5G, utilizaron la simulación para generar su *dataset*.

Tabla 8. Fuente de los datos utilizados en las investigaciones previas (elaboración propia).

Fuente de los datos del Dataset	Número de artículos de investigación
Generados a través de simulación	24
Datos reales (generados utilizando prototipos o <i>datasets</i> públicos)	18
Sintéticos (generados aleatoriamente)	4
No se aclara (el trabajo de investigación no provee información acerca del <i>dataset</i> utilizado)	10

Lo anterior se justifica por el hecho de que la mayoría de los autores no disponen de una variedad adecuada de datos pertenecientes a redes 5G en operación, ya que es una tecnología novedosa y se está implementando lentamente desde finales de 2019. No obstante, dieciocho artículos de investigación utilizaron *dataset* reales para entrenar sus modelos. En algunos trabajos se obtuvieron los datos a través de experimentos utilizando plataformas propias, mientras que otros trabajos utilizaron *dataset* públicos disponibles en Internet. Cuatro trabajos utilizaron datos sintéticos, donde algunos parámetros del entorno 5G evaluado fueron generados aleatoriamente. Finalmente, diez trabajos no describieron la fuente de los datos utilizados para entrenar a los modelos propuestos.

Dentro de los *dataset* de redes 5G disponibles en Internet para ser utilizados se encuentran:

- ✓ *Dataset* 5G/4G obtenido por investigadores de la Universidad de Cork, Irlanda. La información se recopiló desde las redes de dos importantes operadores móviles irlandeses. El conjunto de datos se genera a partir de dos patrones de movilidad (estático y automóvil) a través de dos patrones de aplicación (transmisión de video y descarga de archivos). El *dataset* se compone de KPI del lado del cliente compuestos por métricas relacionadas con el canal, métricas relacionadas con el contexto, métricas relacionadas con las celdas e información del *throughput*. Estas métricas se generan a partir de una conocida aplicación de monitoreo de red de Android (*G-NetTrack Pro*). En el momento de llevar a cabo esta investigación, este es el primer *dataset* disponible públicamente que contiene información de rendimiento, canal y contexto para redes 5G (Raca, Leahy, et al., 2020) (Raca et al., 2018).
- ✓ *Dataset* 5G obtenido por investigadores de la Universidad de Minnesota. El *dataset* se obtuvo a partir de un estudio de medición del rendimiento comercial de redes 5G *mmWave* de operadores de Estados Unidos. Se analizan los mecanismos de transferencia en 5G y su impacto en el

throughput de la red. También se estudia el rendimiento de aplicaciones de navegación web y descargas HTTP sobre la red 5G. Se hace énfasis en aspectos propios de la señal de onda milimétrica (desvanecimiento, interferencias, etc) (Narayanan, Ramadan, Mehta, et al., 2020), (Narayanan, Ramadan, Carpenter, et al., 2020).

Los dos *datasets* descritos anteriormente son recientes y brindan información valiosa para llevar a cabo los análisis y las predicciones utilizando *Deep Learning*. No obstante, hay que destacar que las trazas del *dataset* de la Universidad de Minnesota son relativamente cortas (300 segundos aproximadamente) y la información del canal y del contexto no se publicó. Por otro lado, las trazas del *dataset* de la Universidad de Cork lograron recolectar información del ancho de banda del enlace ascendente y descendente, así como suficiente información del canal y del contexto. Una de las desventajas de este último *dataset* es que no cuenta con trazas 5G recolectadas en el sistema de transporte público (sí están disponibles para el *dataset* 4G), sin embargo las trazas disponibles para el escenario de movilidad en automóvil son suficientes para los requerimientos de la presente investigación (Mei et al., 2021), (Raca, Leahy, et al., 2020).

Dado que las redes 5G aún no tienen una cobertura completa, la práctica actual es recurrir a 4G / LTE siempre que un UE se mueve fuera de la cobertura 5G, esto conlleva a que sea común el *handoff* vertical en los escenarios de movilidad en automóvil, autobús y tren. Lo anterior representa un desafío importante en la predicción del *throughput*, especialmente cuando el modo de acceso a la red cambia entre 5G y 4G, tanto en un sentido como en otro. En este aspecto, ambos *dataset* presentan información acerca de los *handoff* que ocurren, tanto de los HHO como de los VHO, sin embargo el *dataset* de la Universidad de Cork brinda información más detallada sobre los VHO que ocurren cuando el UE se mueve por diferentes áreas de una ciudad (Mei et al., 2021), (Narayanan, Ramadan, Carpenter, et al., 2020), (Raca, Leahy, et al., 2020).

Teniendo en cuenta las ventajas y desventajas asociadas a cada *dataset*, enunciadas anteriormente, para la presente investigación se opta por el *dataset* de la Universidad de Cork.

3.1.2 Selección de los escenarios

La presente investigación se centra en el segmento de red correspondiente al servicio genérico eMBB. Los escenarios seleccionados son el Macro Urbano (áreas suburbanas) y el Urbano Denso (centro de ciudades).

Esta elección está condicionada a que estos son los escenarios contemplados en las trazas del *dataset* de la Universidad de Cork. En ambos escenarios se contempla la movilidad del UE (en automóvil).

3.1.3 Selección de los KPI

Los KPI de interés, y que se encuentran presentes en el *dataset* seleccionado, son los siguientes:

- ✓ Velocidad de datos de descarga (*throughput* de descarga).
- ✓ Velocidad de datos de subida (*throughput* de subida).
- ✓ Velocidad del UE.
- ✓ Parámetros del canal de radio: *Signal to Noise Ratio* (SNR), *Received Signal Strength Indicator* (RSSI), *Reference Signal Received Quality* (RSRQ), *Reference Signals Received Power* (RSRP), RSRQ y RSRP para las celdas vecinas (NRxRSRQ y NRxRSRP), *Channel Quality Indicator* (CQI), tipo de radio, identificador de celda (CellId), estado de la conexión.
- ✓ Latitud
- ✓ Longitud

Dentro de los KPI seleccionados, el *throughput* es fundamental para el cumplimiento de los SLA de los servicios y aplicaciones correspondientes al segmento de red genérico eMBB. En la figura 27 se muestran las capacidades clave para cada uno de los servicios genéricos de 5G; se puede observar cómo para el segmento eMBB el parámetro *user experienced data rate* (KPI *throughput*) tiene asignado “importancia alta” (ITU-R, 2015).

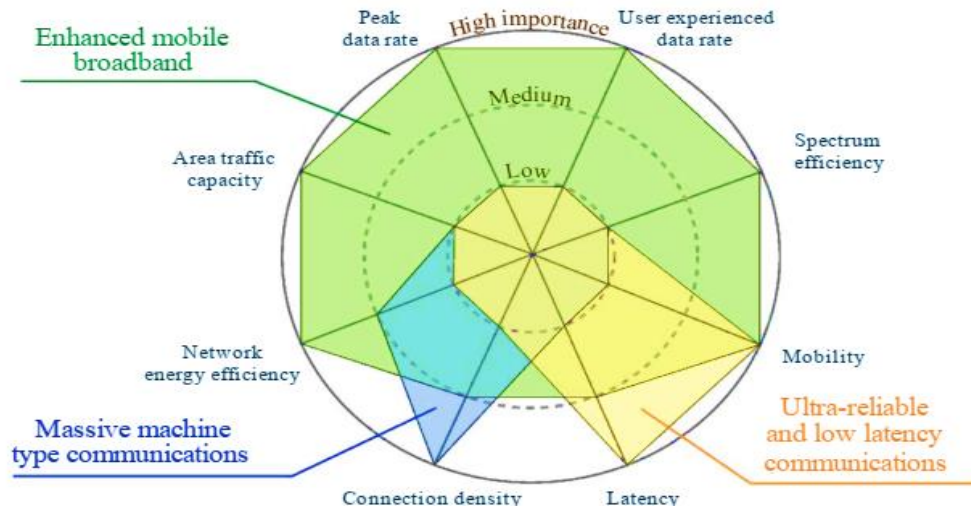


Figura 27. Capacidades claves para los servicios 5G genéricos (ITU-R, 2015).

Es importante esclarecer que todas las capacidades clave no necesariamente deben cumplirse simultáneamente, y algunos requisitos de casos de uso pueden incluso ser mutuamente excluyentes dentro de una única configuración de red. Lo anterior depende en gran medida de cuan diferenciados están los servicios para soportar casos de uso particulares para cada cliente, lo cual es perfectamente viable si se utiliza *network slicing* (Delgado, 2020), (ITU-R, 2015).

La medición del *throughput* es útil para evaluar la carga del sistema en un segmento de red de extremo a extremo. Si el *throughput* de un segmento de red específico no cumple con los requisitos de rendimiento correspondientes (definidos en el SLA para los servicios y/o aplicaciones que utilizan dicho segmento), entonces es necesario realizar acciones sobre el *network slice* (reconfiguración, reubicación de capacidad, etc.) (3GPP, 2021b), (ETSI, 2021b).

Por otra parte, el *throughput* percibido por el UE en la NG-RAN (RAN de próxima generación), es un parámetro muy importante para el funcionamiento de la red 5G. Si el *throughput* experimentado por el UE en la celda 5G NR no cumple con los requisitos de rendimiento, entonces es necesario realizar algunas acciones en la red tales como la reconfiguración o el aumento de capacidad. Lo anterior hace necesario definir correctamente los valores del KPI *throughput* en la RAN, de tal manera que se pueda evaluar la satisfacción de los clientes. Este KPI también cubre los escenarios de 5G NSA, donde el gNodeB está conectado a través del núcleo de la red 4G. La NG-RAN soporta la segmentación de red, por lo que es

importante el monitoreo del *throughput* de descarga y de subida percibido por el UE en cada segmento de red, con el objetivo de que el operador pueda identificar los problemas de desempeño específicos (3GPP, 2021b), (ETSI, 2021b).

Para 5G NR, la tasa de datos aproximada (*throughput*) para un número dado de portadoras agregadas en una banda o combinación de bandas, se calcula utilizando la siguiente ecuación (3GPP, 2021c), (ETSI, 2021c):

$$5G\ NR\ Throughput(Mbps) = 10^{-6} \cdot \sum_{j=1}^J (v_{Layers}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{max} \cdot \frac{N_{PRB}^{BW^{(j)},\mu} \cdot 12}{T_s^\mu} \cdot (1 - OH^{(j)})) \quad (11)$$

donde:

✓ J : número de portadoras agregadas en una banda o combinación de bandas

✓ R_{max} : 948/1024

Para la j -ésima portadora:

✓ v_{Layers}^j : es el número máximo de capas admitidas.

✓ Q_m^j : orden de modulación máximo, toma valor de 2 para QPSK, 4 para 16QAM, 6 para 32QAM y 8 para 256QAM.

✓ $f^{(j)}$: factor de escala, puede tomar cualquier valor de 1 / 0.8 / 0.75 / 0.4.

✓ μ : Numerología 5G NR, puede tomar cualquier valor de 0 a 5.

✓ T_s^μ : Duración media del símbolo OFDM en una subtrama para el valor μ , $T_s^\mu = 10^{-3} / (14 * 2\mu)$.

✓ $N_{PRB}^{BW^{(j)},\mu}$: Asignación máxima de PRB en ancho de banda $BW^{(j)}$ con numerología (μ). $BW^{(j)}$ es el ancho de banda máximo soportado por UE en una banda determinada o en

combinaciones de bandas. PRB es *Physical Resource Blocks*, donde cada PRB consta de 12 subportadoras.

- ✓ $OH^{(j)}$: Sobrecarga que toma cualquiera de los siguientes valores: 0.14 para rango de frecuencia FR1 para DL; 0.18 para rango de frecuencia FR2 para DL; 0.08 para rango de frecuencia FR1 para UL; 0.10 para rango de frecuencia FR2 para UL.

La distribución de las funciones de la NG-RAN entre el sitio de la antena de radio y las ubicaciones centrales también juega un papel fundamental en los requisitos de transporte. Estas funciones incluyen el procesamiento de la señal de radiofrecuencia (RF), y otras capas de la pila de protocolos tales como: capa física (PHY); control de acceso al medio (MAC); control de radioenlace (RLC); protocolo de convergencia de paquetes de datos (PDCP); y control de recursos de radio (RRC). En la arquitectura NG-RAN propuesta por el 3GPP, la funcionalidad de la estación base gNodeB se divide en dos unidades lógicas: una unidad central (CU) y una unidad distribuida (DU). Por otra parte, la ITU-T adoptó para 5G una arquitectura de red de transporte diferente, que se compone de tres elementos lógicos: CU, DU y unidad remota (RU), los cuales se pueden combinar como se muestra en la figura 28 (Tezergil and Onur, 2021), (5G Americas, 2020), (Bartelt et al., 2017).

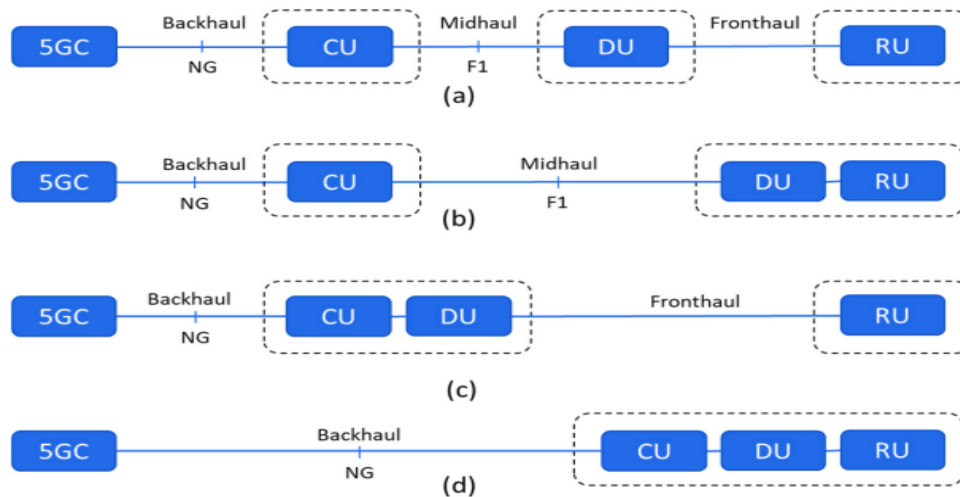


Figura 28. Posibles combinaciones de CU,DU y RU en la arquitectura NG-RAN propuesta por la ITU-T (5G Americas, 2020).

La red de transporte entre el 5GC y el CU se conoce como *backhaul*, e implementa la interfaz 3GPP NG. Del mismo modo, la red de transporte entre el CU y el DU se conoce como *midhaul*, e implementa la interfaz 3GPP F1. Finalmente, la red de transporte entre el DU y el RU es conocida como *fronthaul*. Varias interfaces de red *fronthaul* se han definido hasta la fecha, siendo las dos más utilizadas la interfaz de radio pública común (CPRI) y la CPRI mejorada (eCPRI) (5G Americas, 2020), (Bartelt et al., 2017).

El cálculo a priori del *throughput* (extremo a extremo) para un servicio de red específico constituye un gran desafío, ya que sobre este KPI inciden varios factores, desde los factores de impacto analizados en el capítulo 2, hasta el tipo de arquitectura NG-RAN implementada, el modo 5G desplegado y el patrón de consumo de datos del usuario final.

Para un servicio de red específico (o una aplicación) dentro del segmento genérico eMBB, la predicción del *throughput* es esencial para garantizar la correcta asignación de recursos de red virtualizados, y así evitar posibles violaciones de los SLA. Esta predicción sería un trabajo muy engorroso utilizando técnicas analíticas tradicionales como las descritas en (Alzate, 2004), siendo más factible el uso de técnicas de ML, más específicamente el uso de redes neuronales, las cuales son capaces de encontrar autónomamente las complejas relaciones entre cada una de las variables. Precisamente, en la presente investigación se utilizan redes neuronales para realizar el pronóstico del *throughput*.

3.1.4 Selección, agrupamiento y combinación de los factores de impacto

Los factores que tienen impacto sobre el *throughput* en las redes 5G y 5G *mmWave*, analizados en el capítulo 2, se pueden agrupar por categorías, tal y como se expone a continuación (Narayanan, Ramadan, Mehta, et al., 2020):

- ✓ Factores basados en la geolocalización: latitud, longitud.

- ✓ Factores basados en la movilidad: velocidad de movimiento del UE, dirección cardinal (brújula) del UE.

- ✓ Factores basados en la radio base: distancia UE-panel, ángulo de posición UE-panel, ángulo de movilidad UE-panel.

- ✓ Factores basados en la conexión: *throughput* pasado, tipo de radio, información de la fortaleza de la señal de radio, *handoffs*.

En la presente investigación se utilizarán los factores basados en la geolocalización, movilidad y conexión. Los factores basados en la radio base no se utilizan ya que no están disponibles en el *dataset* elegido. Estos parámetros son manejados por las estaciones base (eNodeB y gNodeB en 4G y 5G respectivamente) que componen la RAN. Desafortunadamente, los investigadores que obtuvieron el *dataset* no tuvieron acceso a dichos parámetros, puesto que se utilizó una red 5G en explotación y el acceso a esa información es exclusiva del operador de dicha red. Tampoco está incluido en el *dataset* la dirección cardinal del UE.

En las redes 3G / 4G, la ubicación geográfica es el factor dominante para indicar el *throughput* y la cobertura de la red; sin embargo, en las redes 5G el *throughput* fluctúa enormemente. Debido a lo anterior se hace necesario tener en cuenta los distintos grupos de factores de impacto, y su combinación, para predecir el *throughput* futuro con el objetivo de detectar proactivamente posibles violaciones de los SLA. En el presente trabajo se utilizan las siguientes combinaciones de grupos de factores de impacto:

- 1) Factores de geolocalización
- 2) Factores de geolocalización + factores de movilidad
- 3) Factores de geolocalización + factores de movilidad + factores de conexión

3.1.5 Selección de la aplicación y del patrón de movilidad

Los datos del *dataset*, obtenido en el estudio llevado a cabo por (Raca, Leahy, et al., 2020), corresponden a dos aplicaciones: la primera es una descarga continua de un fichero de gran tamaño (*file download*) y la segunda es el *streaming* de video desde las plataformas Netflix y Amazon Prime. La aplicación *file download* tiene un rango de variación y rendimiento promedio del *throughput* más alto. Por otra parte, el *streaming* de video desde Netflix y Amazon Prime consume significativamente menos ancho de banda, lo cual es una consecuencia del comportamiento de la aplicación de *streaming*, que descarga segmentos del video solo durante la fase ON (llenado del *buffer*); además, la demanda de ancho de banda está limitada por la máxima calidad del video codificado. En general, Netflix consume más ancho de banda que Amazon

Prime para ambos patrones de movilidad, como resultado de una mayor calidad de codificación, lo cual significa segmentos de video de mayor tamaño (Raca, Leahy, et al., 2020).

En la tabla 9 se muestra la comparación entre las métricas de rendimiento de las aplicaciones *file download* y *streaming* para los patrones de movimiento estático y automóvil, obtenidas del análisis del *dataset* de la Universidad de Cork.

Tabla 9. *Throughput* promedio y rango de variación para diferentes patrones de movilidad para las aplicaciones *file download* y *streaming* (Raca, Leahy, et al., 2020).

Aplicación	Estático				Automóvil			
	<i>Throughput</i> Promedio (Mbps)	<i>Throughput</i> Rango de Variación	No. de trazas	Duración de las trazas (m)	<i>Throughput</i> Promedio (Mbps)	<i>Throughput</i> Rango de Variación	No. de trazas	Duración de las trazas(m)
<i>file download</i>	66.9	(22.0, 202.5)	5	260	28.5	(3.0, 88.5)	16	459
<i>Streaming</i> (Netflix)	13.7	(0.5, 31.1)	10	576	7.5	(0.4, 19.9)	23	637
<i>Streaming</i> (Amazon Prime)	6.9	(0.3, 11.2)	8	582	1.3	(0.3, 2.7)	21	628

En la presente investigación se utilizará la aplicación *file download* debido a que esta no usa *buffer* y se basa en el principio del mejor esfuerzo (*best effort*), lo cual es deseable en la predicción del *throughput* en el enlace descendente, ya que este tendría un comportamiento aleatorio (semejante al *throughput* total debido al consumo de datos de múltiples aplicaciones simultáneas). Por el contrario, en la aplicación de *streaming* habría períodos de inactividad intercalados con otros períodos donde se descargan segmentos de video, lo cual haría menos compleja la predicción ya que el *throughput* se comportaría como una señal ON-OFF. Otra ventaja de utilizar una aplicación que no usa *buffer* y se basa en el principio del *best effort*, es que permite analizar el comportamiento del *throughput* y su relación con los factores de impacto sin tener en cuenta las capas superiores, o sea se puede realizar una abstracción de los protocolos de red y de transporte en los cuales está basado la aplicación.

En cuanto al patrón de movilidad, en el presente trabajo se utilizarán las trazas correspondientes al UE en movimiento (automóvil). La elección anterior se justifica en que para el segmento eMBB el parámetro *mobility* tiene asignado "importancia alta", tal y como se muestra en la figura 27. Además, la movilidad

del UE tiene un alto impacto sobre el *throughput* y determina el número de *handoffs* que se llevan a cabo durante la conexión.

3.2 Predicción del *throughput* utilizando redes neuronales

Existen trabajos previos, que han abordado los problemas asociados a la predicción del *throughput* en redes móviles celulares (Eyceyurt and Zec, 2020), (Kousias et al., 2020), (Raca, Zahran, et al., 2020), (Wei et al., 2018), (Raca et al., 2017), (Samba et al., 2017), (Samba et al., 2016), (Y. Liu and Lee, 2015), (Mirza et al., 2010). En la presente sección, se propone una solución de aprendizaje automático supervisado utilizando redes neuronales, para la predicción del *throughput* en el enlace descendente de una red 5G (segmento eMBB). Es pertinente esclarecer que la solución que se propone e implementa en la presente investigación constituye una solución experimental a escala de laboratorio, no apta aún para ser incorporada en un entorno de producción. Las fases de la solución se basan en la metodología seguida en (Kousias et al., 2020) y se describen a continuación:

- ✓ Inicialización: esta primera fase tiene dos objetivos, el primero es brindar soporte con todos los componentes de software requeridos, y el segundo, iniciar el proceso de importación de datos desde el *dataset*.
- ✓ Procesamiento de datos: en esta fase primeramente se dividen los datos del *dataset* en tres segmentos: datos de entrenamiento, datos de validación y datos de prueba. Posteriormente, se lleva a cabo la normalización de las variables de entrada al algoritmo de ML. La normalización consiste en comprimir o extender los valores de la variable para que estén en un rango definido. Por último, los datos se someten a funciones de transformación y remodelación para cumplir con los requisitos de entrada de la red neuronal.
- ✓ Optimización de hiperparámetros: es un proceso que consume bastante tiempo, pero es muy importante ya que es clave para lograr un rendimiento adecuado de la solución de ML. Ejemplos de hiperparámetros notables son las capas ocultas (HL) de la red neuronal, el número de neuronas, el número de épocas (*Epochs*), el tamaño del lote (BS) y la tasa de aprendizaje (LR). La optimización de los hiperparámetros se puede realizar a través del ajuste o búsqueda manual (MS) o a través de algoritmos de ajuste automatizado tales como: la búsqueda de cuadrícula (GS), que se basa en

la fuerza bruta; la optimización bayesiana (BOA), la cual se basa en un enfoque probabilístico; y la búsqueda aleatoria (RS).

- ✓ Arquitectura de la red neuronal: en el campo de las redes neuronales, el concepto de arquitectura contempla al número de capas neuronales, al número de neuronas en cada una de las capas, a la conexión entre neuronas o capas, al tipo de neuronas presentes e incluso a la forma en la que son entrenadas. Uno de los mayores retos del diseño de soluciones de ML basados en redes neuronales, ya sea de regresión o de clasificación, es que se desconoce la estructura de la red que se adapte de forma óptima al problema objeto de estudio (Bishop, 1995). Lo anterior se debe a que es difícil establecer la cantidad de capas, neuronas, conexiones o funciones de activación de forma automática, motivo por el cual generalmente se llega a la solución del problema basado en la experiencia del especialista que diseña la red neuronal o a través de múltiples pruebas en las que se varía alguno de los parámetros que afectan la arquitectura (Gaona and Ballesteros, 2012), (Bishop, 1995).

3.2.1 Propuesta de solución utilizando una RNN

En la figura 29 se puede observar el diagrama de alto nivel de la solución de ML para la predicción del *throughput*. Esta solución está basada en una red neuronal recurrente con celdas del tipo LSTM.

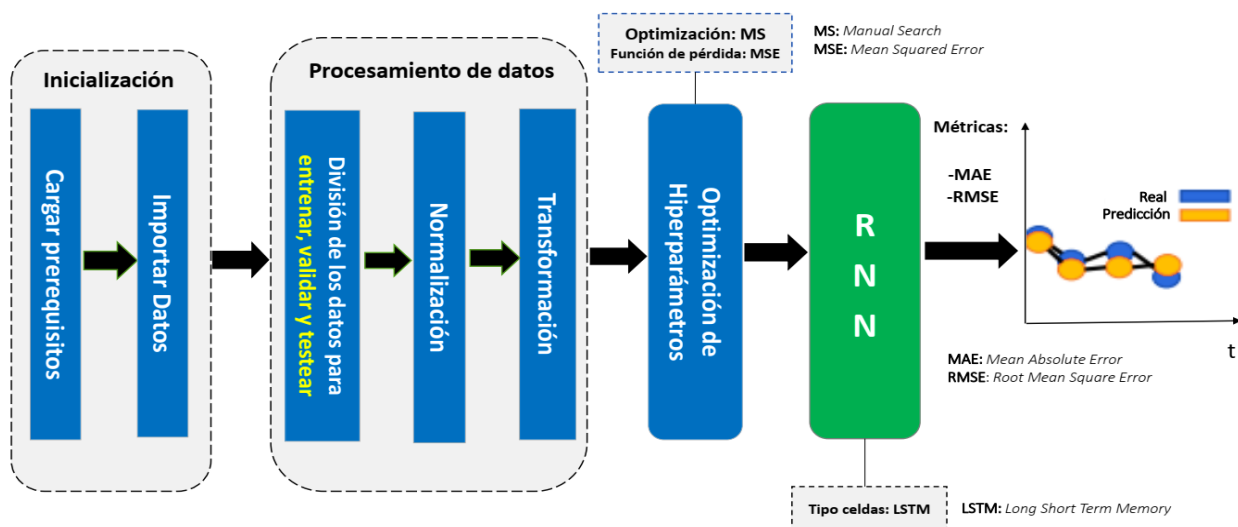


Figura 29. Diagrama de alto nivel de la solución de ML utilizando una RNN para la predicción del *throughput* (Kousias et al., 2020).

A continuación, se detallan las acciones realizadas en cada una de las fases que forman parte de la solución.

I. Inicialización.

En la presente investigación, la implementación de algoritmos de ML se realiza utilizando bibliotecas de código abierto. Se elige Python como lenguaje de programación principal, debido a que este se ha convertido en el lenguaje principal para el desarrollo de ML y ciencia de datos, y cuenta con las bibliotecas actualizadas más conocidas para ML (Alcalá, 2021). Además, se opta por la utilización de infraestructuras disponibles en la nube que permiten implementar, entrenar y probar las soluciones de ML diseñadas. Lo anterior se justifica en la existencia de servicios en la nube que permiten almacenar datos en Internet de forma gratuita; además de que la mayoría de las veces estos servicios integran núcleos GPU y TPU (unidad de procesamiento tensorial) gratis, que están diseñados específicamente para acelerar el entrenamiento de redes neuronales profundas. Normalmente, las versiones gratis de estos servicios en la nube tienen algunas limitaciones en cuanto a la capacidad de almacenamiento y el tiempo de uso de los núcleos GPU y TPU; sin embargo, los recursos ofertados por dichas infraestructuras son suficientes para la implementación de la solución que en este trabajo de investigación se propone.

En la tabla 10, se exponen las principales herramientas de software utilizadas en la implementación de la solución de ML:

Tabla 10. Bibliotecas y entornos de desarrollo de ML utilizados (elaboración propia).

Nombre de la herramienta	Tipo	Características
<i>NumPy</i>	Biblioteca de Python	Biblioteca que agrega soporte para vectores y matrices multidimensionales. Incluye una gran colección de funciones matemáticas de alto nivel para operar con matrices.
<i>Pandas</i>	Biblioteca de Python	Biblioteca para la manipulación y análisis de datos. Ofrece estructuras de datos y operaciones para manipular datos numéricos y series de tiempo.
<i>Matplotlib</i>	Biblioteca de Python	Biblioteca de graficado para <i>NumPy</i> . Proporciona una API orientada a objetos para incrustar gráficos en aplicaciones.
<i>Seaborn</i>	Biblioteca de Python	Biblioteca de visualización de datos de Python basada en <i>matplotlib</i> Proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos.
<i>Tabulate</i>	Biblioteca de Python	Biblioteca que permite mostrar datos de tablas de manera estética. No es parte de la biblioteca estándar de Python, por lo que es necesario instalarla de forma independiente.

<i>Datetime</i>	Biblioteca de Python	Biblioteca que proporciona clases para manipular fechas y horas. Si bien se admite la aritmética de fecha y hora, el enfoque de la implementación está en la extracción eficiente de atributos para formato de salida y manipulación.
<i>Keras</i>	Biblioteca de Python	Biblioteca para implementar redes neuronales. Es capaz de ejecutarse sobre <i>TensorFlow</i> , <i>R</i> , <i>Theano</i> y otros <i>Backends</i> de ML. Diseñado para permitir la experimentación rápida con DNN, se centra en ser modular, extensible y fácil de usar.
<i>TensorFlow</i>	Biblioteca de Python	Plataforma de extremo a extremo para el flujo de datos de ML. Tiene un amplio ecosistema de herramientas que permite a los investigadores impulsar el estado del arte en ML y a los desarrolladores construir e implementar aplicaciones de ML. Se utiliza tanto para la investigación como para la producción en Google.
<i>Google Colab</i>	Entorno de desarrollo	Servicio gratuito en la nube de Google, para desarrolladores de IA similar a <i>Jupyter Notebook</i> . Integra las principales bibliotecas de ML y permite contribuciones de múltiples usuarios. Brinda acceso gratuito a núcleos GPU y TPU para entrenamiento de redes neuronales (uso limitado a 12 horas diarias).

II. Procesamiento de datos.

División de los datos:

Las soluciones basadas en redes neuronales tienen en común que necesitan que se lleve a cabo una fase de aprendizaje o entrenamiento, para posteriormente poder obtener una mayor precisión en las predicciones. Por lo anterior es necesario dividir los datos del *dataset* en tres conjuntos de datos: datos de entrenamiento, datos de validación y datos de prueba. Los datos no se mezclan al azar antes de separarse, ya que se trata de series de tiempo y es necesario garantizar lo siguiente (TensorFlow Team, 2019):

- ✓ Garantizar que aún sea posible dividir los datos en ventanas de muestras consecutivas.
- ✓ Garantizar que los resultados de la validación / prueba sean más realistas y se evalúen en función de los datos recopilados después de que el modelo haya sido entrenado.

En la figura 30 se muestran los detalles de la división del *dataset* que se utilizó en la solución de ML:

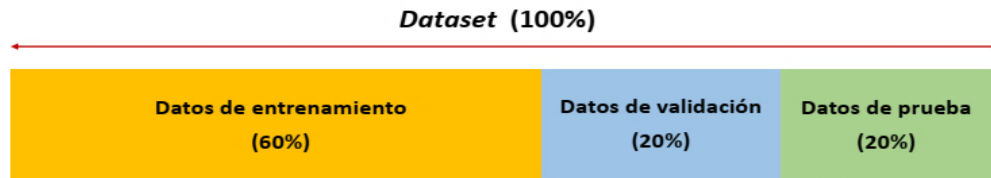


Figura 30. División del *dataset* en datos de entrenamiento, validación y prueba (elaboración propia).

El conjunto de datos de entrenamiento se utiliza para el entrenamiento de la red neuronal, proceso en el cual se calculan los pesos y sesgos. Por otra parte, el conjunto de datos de validación se utiliza para analizar como marcha el aprendizaje de la red neuronal, esto se logra calculando la precisión de la predicción al utilizar un conjunto de datos distinto al de entrenamiento. Lo anterior permite evaluar el modelo a la vez que se van ajustando los pesos y sesgos. Por último, el conjunto de datos de pruebas se utiliza una vez que finaliza el entrenamiento de la red neuronal, o sea estos datos se utilizan para dar la evaluación final del modelo, no sometida a sesgo (Shah, 2017).

Normalización:

Para que funcionen mejor muchos algoritmos de ML, se hace necesario normalizar las variables de entrada al mismo. Normalizar significa, en este contexto, comprimir o extender los valores de la variable para que estén en un rango definido. Dentro de los métodos más utilizados está el escalado de variables (*Feature Scaling* o *MinMax Scaler*) y el escalado estándar (*Standard Scaler*). A menudo a estos métodos se les llama escalamiento de datos y normalización de datos respectivamente. En el escalado de variables, se cambia el rango de distribución de los datos, mientras que en el escalado estándar, se cambia la forma de la distribución de los datos (Krukrubo, 2021).

En general, se opta por el escalado estándar si se va a utilizar una técnica de aprendizaje automático o estadística que asume que los datos se distribuyen normalmente (distribución gaussiana). Por otra parte, se utiliza el escalado de variables cuando se desea que las variables de entrada al modelo permanezcan en un rango uniforme, de modo que una no tenga preponderancia sobre otra. En el caso del modelo de ML propuesto, se opta por utilizar el escalamiento de datos mediante *MinMax Scaler*, definido en la ecuación 12. Este escalador comprime los datos de entrada entre unos límites empíricos (el máximo y el mínimo de las variables), y los valores resultantes oscilan entre cero y uno (Krukrubo, 2021).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (12)$$

Los valores máximos y mínimos de las variables de entrada deben calcularse utilizando los datos de entrenamiento para que el modelo no tenga acceso a los valores en los conjuntos de validación y prueba.

Transformación:

Finalmente, los datos se someten a funciones de transformación y remodelación para cumplir con los requisitos de entrada a la red neuronal. Las predicciones están basadas en una ventana de muestras consecutivas de los datos, tal y como se muestra en la figura 31.

La ventana de entrada estará compuesta por los valores de las variables de entrada (características) para cada paso de tiempo; el ancho de la entrada define la cantidad de pasos de tiempo que conformarán el historial; el *offset* define los pasos de tiempo futuro en donde estará enmarcada la predicción; y el ancho de etiqueta define los pasos de tiempo (su valor puede ser igual o menor que el *offset*, y funciona de derecha a izquierda) para los cuales se estimarán los valores futuros de la variable escogida para la predicción. Es importante destacar que en caso de que la serie de tiempo sea multivariante la predicción puede contemplar a todas las variables, un grupo de ellas o una en específico. En esta investigación la predicción se realiza para el *throughput* de descarga con el objetivo de prevenir violaciones de los SLA para una aplicación del segmento eMBB; no obstante, también en el epígrafe 3.4 se aborda la combinación de la predicción del *throughput* y del *handoff* vertical en la toma de decisiones para prevenir violaciones de los SLA en escenarios de alta movilidad.

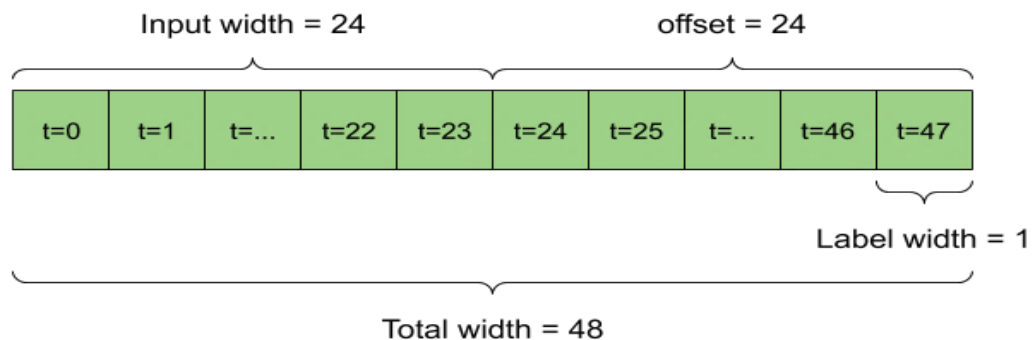


Figura 31. Ejemplo de una ventana para realizar una predicción en el futuro (t=47) a partir de un historial de 24 unidades de tiempo (TensorFlow Team, 2019).

Básicamente, la construcción de una ventana de datos como la mostrada en la figura 31 se puede llevar a cabo con la clase *WindowGenerator* definida en (TensorFlow Team, 2019). Esta clase puede:

- ✓ Manejar los índices y los *offsets*.
- ✓ Dividir la ventana de datos en los pares: características, etiquetas.
- ✓ Graficar el contenido de la ventana resultante.
- ✓ Generar de manera eficiente lotes de estas ventanas a partir de los datos de entrenamiento, validación y prueba, utilizando la API *tf.data.Dataset*, la cual admite la implementación eficiente de entradas *pipeline*.

III. Optimización de hiperparámetros.

Función de pérdida:

El objetivo de todo modelo de ML es ser capaz de describir la relación entre variables con la mayor precisión posible, o sea lograr que el valor de salida del modelo (valor estimado) se aproxime al valor real (valor esperado) con el mínimo error posible. Para lograr lo anterior se necesita de una función de pérdida o coste (*loss function*), la cual se debe minimizar utilizando un algoritmo de optimización conocido como gradiente descendente. Este algoritmo es un método iterativo que modifica los parámetros de una función objetivo (generalmente convexa) para encontrar su mínimo local. O sea, en el caso de una solución basada en redes neuronales, el método del gradiente descendente se utiliza para optimizar las ponderaciones (pesos y sesgos) de cada neurona, de tal forma que se garantice que la función de pérdida sea mínima (Trehan, 2020).

Matemáticamente el gradiente se calcula con derivadas parciales según la ecuación 13, y es necesario encontrar un valor de $\theta \in R^d$, donde $\theta = (\omega_1, b_1 \dots \omega_k, b_k)$, que minimice la función de pérdida $J(\theta)$. Para ello se realizan iteraciones sucesivas actualizando el valor de θ hasta que se satisfaga la condición del $\min_{\theta} J(\theta)$, tal y como lo expresa la ecuación 14 (Vinayakumar et al., 2017), (Vryniotis Vasilis, 2013):

$$\nabla_{\theta} J(\theta) = \frac{\partial J(\theta)}{\partial \theta} \quad (13)$$

$$\theta_{k+1} = \theta_k - \alpha \nabla_{\theta} J(\theta_k) \quad (14)$$

El parámetro α denota la tasa de aprendizaje y tiene un impacto directo en la velocidad de aproximación a la ponderación óptima. Cuanto menor es la tasa de aprendizaje, se consume más tiempo para la convergencia al $\min_{\theta} J(\theta)$. Si α es demasiado grande, se corre el riesgo de saltarse la solución óptima y no lograr la convergencia. A diferencia de θ , la tasa de aprendizaje es un parámetro que no es ajustado por el algoritmo de optimización del gradiente descendente, sino que debe ser configurado por separado. A este tipo de parámetros se los denomina hiperparámetros, y deben ser elegidos con cuidado para garantizar el buen desempeño de la red neuronal (Trehan, 2020).

En modelos de regresión se pueden utilizar las funciones de error cuadrático (*Mean Square Error*, MSE por sus siglas en inglés) y error absoluto (*Mean Absolute Error*, MAE por sus siglas en inglés) como funciones de pérdida, siendo la primera la más utilizada debido a que es más eficiente para el algoritmo de aprendizaje. La mayor eficiencia de MSE se debe a que el valor del gradiente es alto para errores significativos, y va decreciendo según se aproxima a cero, lo cual le da una mayor precisión al final del proceso de entrenamiento del modelo. Por el contrario, el valor del gradiente del MAE permanece constante en cada punto, tal y como se muestra en la figura 32 (Grover, 2018). Las ecuaciones 15 y 16 describen al MSE y al MAE respectivamente, donde y_i es el valor esperado y \hat{y}_i es el valor estimado:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (15)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (16)$$

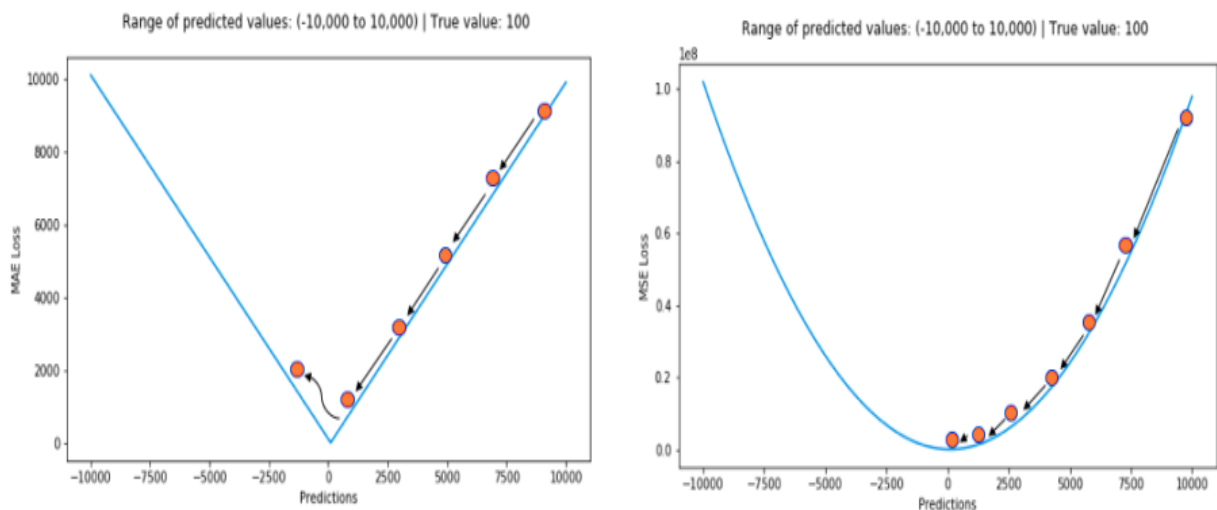


Figura 32. Representación de las funciones MAE (izquierda) y MSE (derecha) y diferencias en los valores del gradiente para cada punto (Grover, 2018)

Teniendo en cuenta los argumentos planteados anteriormente, en la presente investigación se selecciona al MSE como función de pérdida.

Técnica de validación:

La estimación del error para evaluar la efectividad del modelo se realiza posterior al entrenamiento de este. Sin embargo, evaluar la efectividad a posteriori no ofrece información de la capacidad de generalización del modelo frente a un *dataset* independiente. Para garantizar dicha capacidad, en la etapa de entrenamiento se utilizan métodos como el de la validación cruzada (*cross-validation*) (Kumar, 2020).

Cuando se trata de datos de series temporales, la validación cruzada tradicional (K iteraciones, aleatoria, dejar uno fuera) no debe usarse ya que es necesario mantener el orden cronológico de los datos. Debido a lo anterior, para los datos de series de tiempo se utiliza la validación cruzada de retención (*holdout cross-validation*), donde un subconjunto de los datos (dividido temporalmente) se reserva para validar el desempeño del modelo, tal y como se muestra en la figura 33. Este conjunto de validación evita que el modelo se sobreajuste al conjunto de prueba (Kumar, 2020), (Cochrane, 2018).

La importancia de tener un conjunto de datos de validación, además de mantener un conjunto de datos de prueba, se debe a que hay algunas decisiones y características del modelo que deben tomarse y ajustarse y que el algoritmo no es capaz de aprender. Lo anterior se refiere a los hiperparámetros, los cuales son parámetros que afectan la forma en que el modelo aprende sus datos de entrenamiento y crea un modelo capaz de generalizar lo aprendido. Los hiperparámetros deben ser determinados por el usuario, por lo que se hace necesario realizar la evaluación de cómo se está desempeñando el modelo con los datos que no se usaron para el entrenamiento, y a partir de ello ajustarlos hasta que se encuentre un modelo óptimo. Lo anterior implica que se termine sobreajustando ligeramente el modelo en función del conjunto de validación, por lo que la precisión que se obtiene del conjunto de validación no se considera la final. Debido a esto se hace necesario utilizar otro conjunto de datos de reserva, en este caso el conjunto de datos de prueba, para evaluar el modelo final. El error encontrado utilizando el conjunto de pruebas se considera como el error de generalización del modelo de ML (Cochrane, 2018).

Sin embargo, la validación cruzada de retención tiene como inconveniente que la elección del conjunto de prueba es arbitraria, y esa elección puede significar que el error del conjunto de prueba es una estimación pobre del error en un conjunto de prueba independiente. Para solucionar ese problema, se puede utilizar el método de la validación cruzada anidada (*nested cross-validation*).

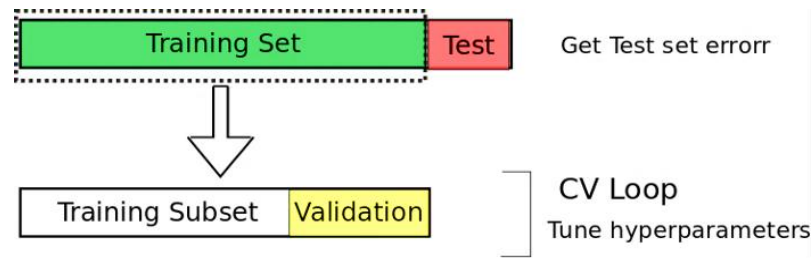


Figura 33. Representación gráfica del método *holdout cross-validation* (Cochrane, 2018).

Este método contiene un bucle externo para la estimación de errores y un bucle interno para el ajuste de parámetros, tal y como se muestra en la figura 34. El bucle interno funciona exactamente como el de la validación cruzada de retención y el bucle externo divide el conjunto de datos en varios conjuntos de prueba y entrenamiento diferentes; por último se promedia el error de cada división para obtener una estimación sólida del error del modelo (Cochrane, 2018).

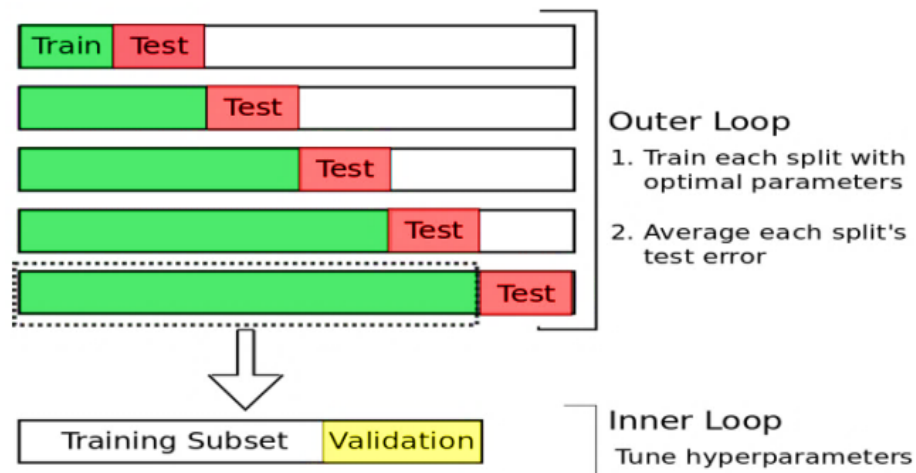


Figura 34. Representación gráfica del método *nested cross-validation* (Cochrane, 2018).

Es importante destacar que se utiliza el MAE y el RMSE (*Root Mean Square Error*) como métricas para calcular el error final del modelo utilizando el conjunto de datos de pruebas. El MAE se definió en la ecuación 16 y el RMSE es la raíz cuadrada del MSE, tal y como muestra la ecuación 17.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (17)$$

Tanto MAE como RMSE expresan el error promedio de predicción del modelo en unidades de la variable de interés y pueden oscilar entre 0 e ∞ , siendo además indiferentes a la dirección de los errores. Sin embargo, en RMSE los errores se elevan al cuadrado antes de promediarlos, lo que significa que los errores grandes tendrán mayor peso. Debido a lo anterior, RMSE es una métrica más útil cuando los errores grandes son particularmente indeseables. Por otra parte, desde el punto de vista de la interpretación el MAE es una métrica más fácil de comprender, y es preferible su uso en caso de que los valores atípicos no tengan una importancia significativa. Otro detalle importante es que el RMSE siempre será mayor o igual que el MAE, ocurriendo la igualdad cuando los errores tienen la misma magnitud (Chai and Draxler, 2014), (Willmott and Matsuura, 2005).

Optimización de hiperparámetros:

La optimización de hiperparámetros es clave para lograr un buen desempeño del modelo de ML. Dentro de los enfoques de optimización, los más populares son los siguientes (Kousias et al., 2020):

- ✓ Búsqueda manual (*Manual Search*, MS por sus siglas en inglés): no es un algoritmo de optimización por definición, el ajuste manual de hiperparámetros todavía es posible en casos en los que se tiene conocimiento sobre el sistema en estudio. Es la opción menos eficiente, ya que generalmente requiere una extensa prueba y error. Tiene como ventaja que implica un mínimo esfuerzo de codificación.
- ✓ Búsqueda de cuadrícula (*Grid Search*, GS por sus siglas en inglés): método algorítmico equivalente a la fuerza bruta, se refiere a probar todas las soluciones posibles en el camino al óptimo global. GS supera a todos los métodos de optimización, ya que siempre busca la mejor solución. Por otra parte, a medida que aumenta el número de hiperparámetros, sufre graves problemas de escalabilidad. Además, requiere considerables recursos, que incluso con una computadora potente, la optimización puede llevar días o incluso semanas.
- ✓ Búsqueda aleatoria (*Random Search*, RS por sus siglas en inglés): en lugar de evaluar todas las posibles combinaciones de hiperparámetros, RS itera sobre una muestra más pequeña. Como el número de iteraciones aumenta, la probabilidad de converger a una mejor solución también aumenta. El número de iteraciones necesario para llegar a una buena solución depende de varios factores, incluido el tamaño del espacio de hiperparámetros, rango de búsqueda, complejidad de datos, etc.

- ✓ Optimización bayesiana (*Bayesian Optimization*, BOA por sus siglas en inglés): este algoritmo presenta un enfoque probabilístico y es una alternativa diferente para funciones complejas y ruidosas, ya que incorpora aprendizaje para la optimización de hiperparámetros. BOA tiene sus raíces en el teorema de Bayes que explota el concepto de probabilidad condicional para hacer nuevas predicciones. Similar a RS, es necesario un número de iteraciones para converger a una buena solución.

En la presente investigación se opta por el ajuste manual de hiperparámetros, ya que implica el mínimo esfuerzo de codificación, lo cual se traduce en una menor complejidad de la solución. Además, la decisión anterior también tuvo en cuenta el tiempo disponible para llevar a cabo la investigación y el objetivo de esta. No obstante, si la solución fuera a ser implementada en un entorno de producción habría que valorar otro algoritmo de optimización que garantice la obtención de los mejores resultados. En la tabla 11 se muestran los principales hiperparámetros y sus respectivos valores:

Tabla 11. Valores de los hiperparámetros del modelo RNN (elaboración propia).

Hiperparámetros	Rango de valores	Descripción
<i>hidden_size</i>	3 a 7 (según modelo y contando las capas ocultas del codificador, vector de codificación y decodificador)	Tamaño de la representación oculta (capas ocultas)
<i>n_units</i>	64 a 96 (según modelo)	Número de celdas
<i>n_layers_enc</i>	1 a 3 (según modelo)	Número de capas del codificador
<i>n_layers_dec</i>	1 a 3 (según modelo)	Número de capas del decodificador
<i>tf</i>	$tf = \text{lambda } x: 1e-3 * 0.9^{**}x$ (<i>learning_rate</i> =1e-3, <i>momentum</i> =0.9)	Tasa de aprendizaje forzado
<i>patience</i>	8	Número de épocas sin mejora
<i>learning_rate</i>	1e-3	Tasa de aprendizaje del optimizador
<i>batch_size</i>	32	Tamaño del lote
<i>epochs</i>	20	Número de iteraciones. Define el número de pases hacia adelante y hacia atrás durante el proceso de entrenamiento del modelo de ML

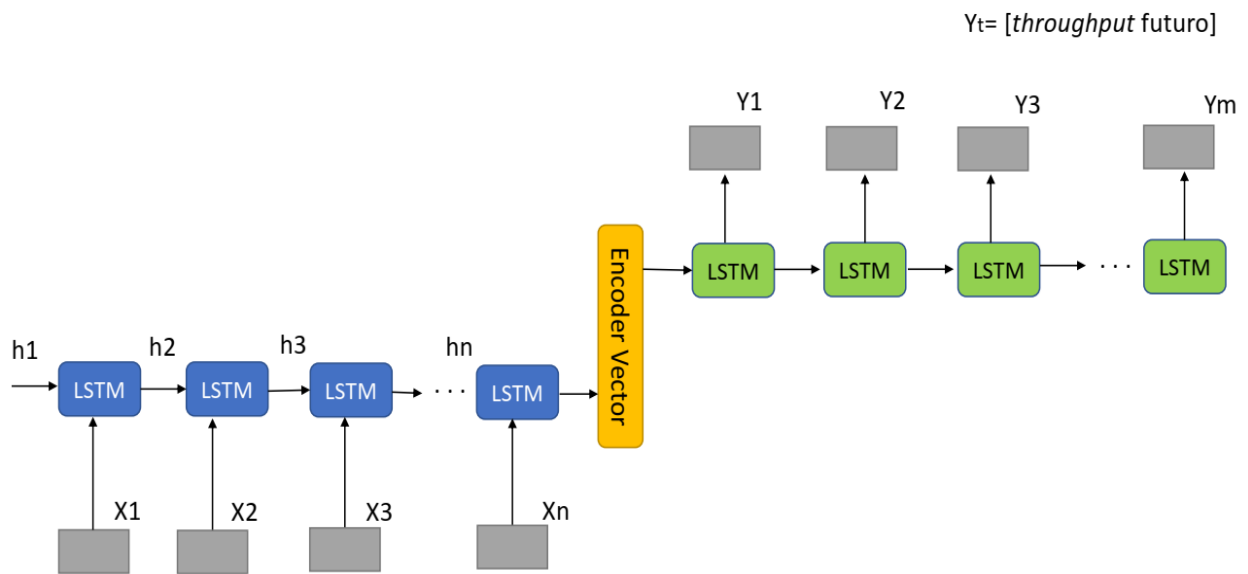
En la tabla 12 se muestra un resumen de otros parámetros de la solución propuesta:

Tabla 12. Otros parámetros del modelo RNN (elaboración propia).

Parámetros del modelo de ML	Valor(es)	Descripción
Características	16	VARIABLES de entrada del modelo de ML
Normalización	<i>min-max</i>	Método de normalización utilizado
Tamaño de Ventanas	60; 30 120; 30 180; 30	Relación del número de pasos de tiempo de la ventana de entrada y de la ventana de etiquetas (variables cuyo valor se predice)
Celda	LSTM	Tipo de celda utilizada en la red neuronal
Arquitectura	<i>Encoder-Decoder</i>	Tipo de arquitectura de la red neuronal
Activación	<i>tanh</i>	Función de activación para activar el estado de la celda y el estado oculto
Activación recurrente	<i>sigmoid</i>	Función de activación para activar las compuertas <i>input</i> , <i>forget</i> , <i>output</i> de la celda LSTM
Optimización	<i>Adam</i>	Método de optimización del gradiente descendente utilizado
División del <i>dataset</i>	60%; 20%; 20%	División del <i>dataset</i> en (%): datos de entrenamiento, datos de validación y datos de prueba
Validación	<i>holdout cross validation</i>	Técnica de validación cruzada utilizada
Función de pérdida	MSE	Función que evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones utilizadas durante el aprendizaje.
Métricas de error	MAE; RMSE	Métricas de error utilizadas cuando se prueba el modelo final en un conjunto de datos de pruebas, nunca visto por el modelo de ML.

IV. Arquitectura.

La RNN propuesta presenta una arquitectura *encoder-decoder* y utiliza celdas LSTM, tal y como se puede observar en la figura 35. Esta arquitectura fue presentada por primera vez en 2014 por Google, y los primeros usos que se le dio fue la traducción automática y el procesamiento del lenguaje natural. De forma general esta arquitectura puede ser útil para resolver problemas del tipo secuencia a secuencia (*seq2seq*), por lo que se ha utilizado con éxito en el pronóstico de series de tiempo (Du et al., 2020), (Vadiraja and Chattha, 2020), (Gong et al., 2019).



$X_t = [\text{latitud, longitud, velocidad UE, throughput pasado, tipo de radio, fortaleza señal, \dots}]$

Figura 35. RNN con celdas LSTM en una arquitectura *Encoder-Decoder* para la predicción del *throughput* (elaboración propia).

La arquitectura consta de 3 partes (Kostadinov, 2019):

- ✓ **Codificador:** mapea una secuencia fuente de longitud variable a un vector de longitud fija. Está compuesto por varias unidades recurrentes (celdas LSTM en este caso), donde cada una de las celdas acepta un solo elemento de la secuencia de entrada, recopila información y realiza la propagación hacia delante de esa información. Los estados ocultos del codificador se pueden calcular según la ecuación 18. Es de destacar que esta ecuación representa el resultado de una RNN ordinaria, y solo se le aplica los pesos apropiados al estado anterior h_{t-1} y al vector de entrada x_t .

$$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{hx} \cdot x_t) \quad (18)$$

- ✓ **Vector del codificador:** es el estado oculto final producido por el codificador del modelo. Se calcula utilizando la ecuación 18. Este vector tiene como objetivo encapsular la información de todos los elementos de entrada para ayudar al decodificador a realizar predicciones precisas. Actúa como el estado oculto inicial de la parte del decodificador del modelo.

- ✓ Decodificador: mapea la representación del vector a una secuencia objetivo de longitud variable. Está compuesto por varias unidades recurrentes (celdas LSTM en este caso), donde cada una predice una salida y_t en un paso de tiempo t . Cada unidad recurrente acepta un estado oculto de la unidad anterior y produce una salida, así como su propio estado oculto. Los estados ocultos del decodificador se pueden calcular según la ecuación 19, donde se puede constatar que el cálculo de un estado oculto solo depende del estado anterior y su respectivo peso. Las salidas y_t se calculan utilizando el estado oculto en el paso de tiempo t con el peso respectivo W_s , tal y como se expone en la ecuación 20.

$$h_t = \tanh(W_{hh} \cdot h_{t-1}) \quad (19)$$

$$y_t = \tanh(W_s \cdot h_t) \quad (20)$$

La potencia de este modelo radica en el hecho de que puede mapear secuencias de diferentes longitudes entre sí, ya que las entradas y salidas no están correlacionadas y sus longitudes pueden diferir. Esto es algo muy importante en el caso de las series temporales, donde muchas veces se quiere predecir valores futuros a partir de valores históricos. En la presente investigación se utiliza esta arquitectura (con una, dos y tres capas de codificador-decodificador) para la predicción del *throughput*, utilizando entradas y salidas de diferente longitud.

3.2.2 Propuesta de solución utilizando una red neuronal mixta CNN-RNN

En la figura 36 se puede observar el diagrama de alto nivel de la solución de ML para la predicción del *throughput*, utilizando una red mixta que combina una red convolucional con una red recurrente.

Las fases I y II de esta solución coinciden con las fases I y II abordadas en el epígrafe 3.2.1. La fase III coincide en todo excepto en los hiperparámetros del modelo los cuales se exponen en la tabla 13 y en algunos parámetros generales que se exponen en la tabla 14.

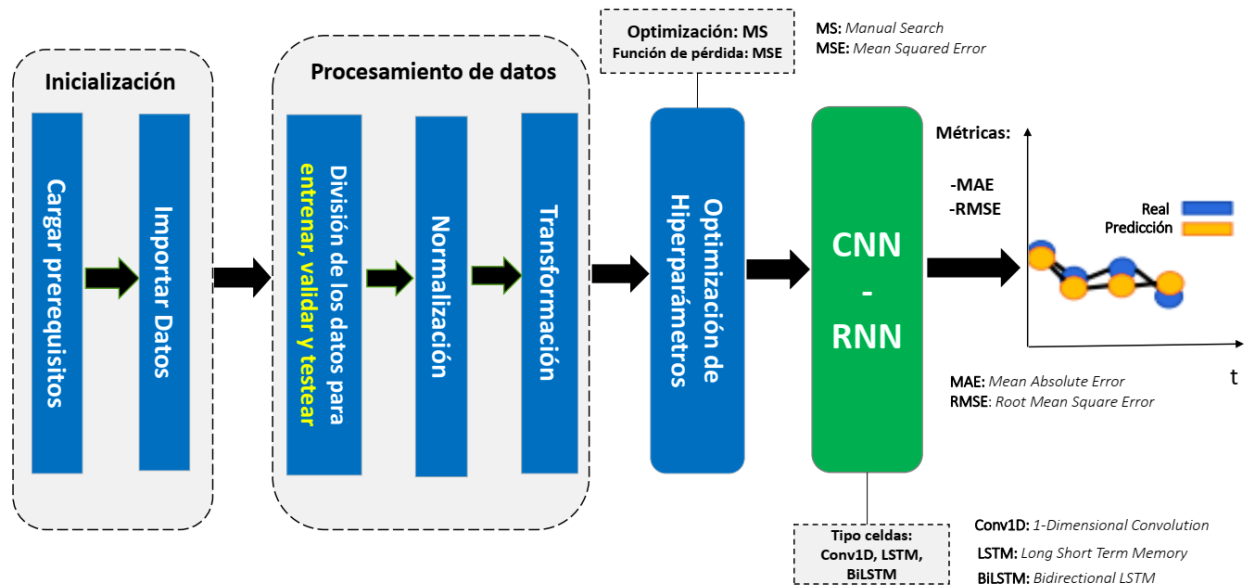


Figura 36. Diagrama de alto nivel de la solución de ML utilizando una red mixta CNN-RNN para la predicción del throughput (Kousias et al., 2020).

Tabla 13. Valores de los hiperparámetros del modelo mixto CNN-RNN (elaboración propia).

Hiperparámetros	Rango de valores	Descripción
<i>hidden_size</i>	14	Tamaño de la representación oculta (capas ocultas)
<i>n_filters</i>	512	Número de filtros convolucionales de la CNN
<i>kernel_size</i>	3	Tamaño del kernel de la CNN
<i>n_layers_cnn</i>	2 (una capa convolucional Conv1D y una capa de Max-Pooling 1D)	Número de capas de la red CNN
<i>n_units</i>	1184 (1024 LSTM y 160 BiLSTM)	Número de celdas de la RNN
<i>n_layers_rnn</i>	13 (8 capas LSTM y 5 capas BiLSTM)	Número de capas de la RNN
<i>tf</i>	$tf = \text{lambda } x: 1e-3 * 0.9^{**}x$ (learning_rate=1e-3, momentum=0.9)	Tasa de aprendizaje forzado
<i>patience</i>	8	Número de épocas sin mejora
<i>learning_rate</i>	1e-3	Tasa de aprendizaje del optimizador
<i>batch_size</i>	32	Tamaño del lote
<i>epochs</i>	20	Número de iteraciones: define el número de pases hacia adelante y hacia atrás durante el proceso de entrenamiento del modelo de ML

Tabla 14. Otros parámetros del modelo mixto CNN-RNN (elaboración propia).

Parámetros del modelo de ML	Valor(es)	Descripción
Características	16	VARIABLES de entrada del modelo de ML
Normalización	<i>min-max</i>	Método de normalización utilizado
Tamaño de Ventanas	60; 30 120; 30 180; 30	Relación del número de pasos de tiempo de la ventana de entrada y de la ventana de etiquetas (variables cuyo valor se predice)
Celda	Conv1D, LSTM, BiLSTM	Tipo de celda utilizada en la red neuronal
Arquitectura	Mixta (CNN-RNN)	Tipo de arquitectura de la red neuronal
Activación	<i>ReLU</i>	Función de activación para activar el estado de la celda y el estado oculto
Optimización	<i>Adam</i>	Método de optimización del gradiente descendente utilizado
División del <i>dataset</i>	60%; 20%; 20%	División del <i>dataset</i> en (%): datos de entrenamiento, datos de validación y datos de prueba
Validación	<i>holdout cross validation</i>	Técnica de validación cruzada utilizada
Función de pérdida	MSE	Función que evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones utilizadas durante el aprendizaje.
Métricas de error	MAE; RMSE	Métricas de error utilizadas cuando se prueba el modelo final en un conjunto de datos de pruebas, nunca visto por el modelo de ML.

La arquitectura para la implementación del modelo mixto CNN-RNN utiliza tres tipos de celdas. En la primera etapa se usan celdas convolucionales de una dimensión (Conv1D) para la extracción de las características de la serie de tiempo. En la segunda etapa se utilizan celdas LSTM y BiLSTM (LSTM bidireccionales) para la predicción del *throughput*. Este tipo de redes neuronales mixtas se han utilizado con éxito en trabajos recientes donde se aborda la predicción de series de tiempo (consumo de energía, precios de acciones) y en el procesamiento del lenguaje (Wu et al., 2021), (Lu et al., 2020), (Hassan and Mahmood, 2018). En la figura 37 se puede observar la arquitectura de esta red neuronal.

La red CNN está diseñada con una capa convolucional unidimensional, con 512 núcleos convolucionales. El tamaño del *kernel* convolucional se establece en 3 y el tamaño del paso se establece en 1. Después de que se realiza la operación de convolución, se extraen las características de los datos, pero las dimensiones de estas son muy altas. Para reducir las dimensiones de las características extraídas y reducir el coste de entrenamiento de la red, se agrega una capa de agrupación *Max-Pooling* posterior a la capa de convolución (Wu et al., 2021), (Lu et al., 2020). Esta capa evita el sobreajuste del modelo, sin embargo, en ocasiones puede provocar la pérdida de detalles importantes de las características (Hassan and Mahmood, 2018).

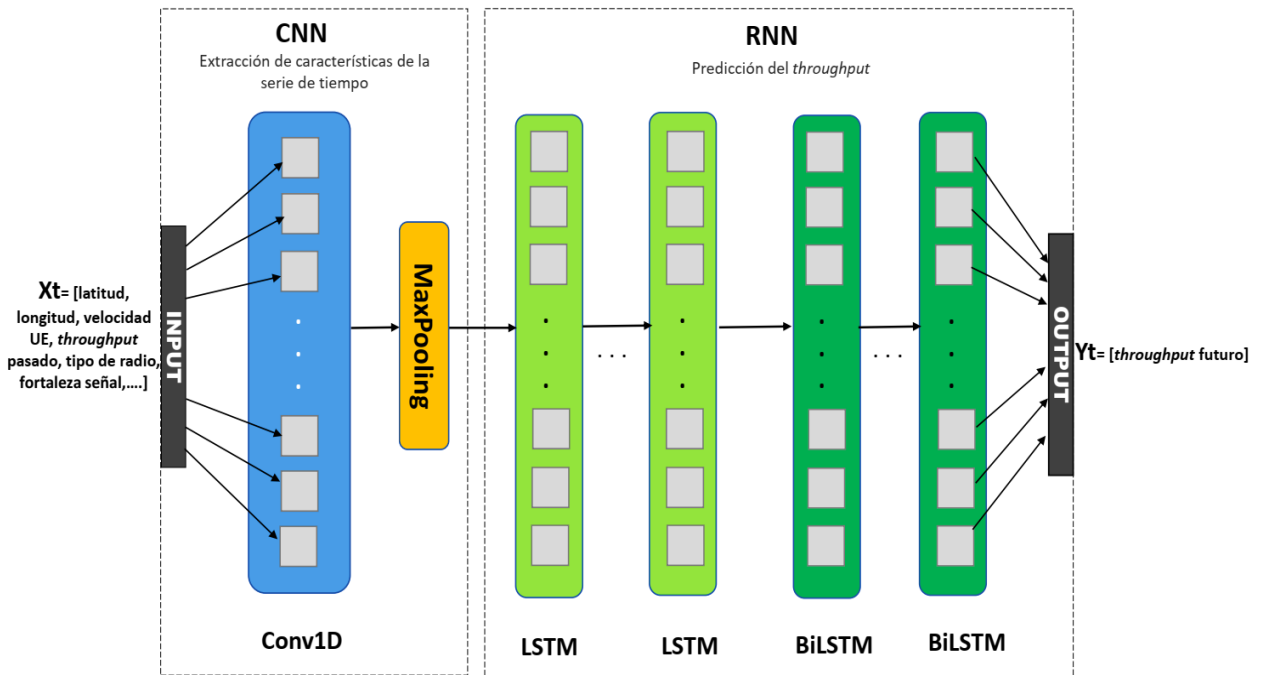


Figura 37. Red CNN-RNN con celdas Conv1D, LSTM y BiLSTM para la predicción del *throughput* (elaboración propia).

El cálculo del valor de salida después de la convolución se realiza según la ecuación 21:

$$l_t = \text{relu}(x_t \odot k_t + b_t) \quad (21)$$

donde l_t representa el valor de salida después de la convolución, relu es la función de activación, x_t es el vector de entrada, k_t y b_t son el peso y el sesgo del *kernel* de convolución respectivamente.

Las características extraídas por la CNN se toman como la entrada de ocho capas LSTM y cinco capas BiLSTM, las cuales realizan la predicción del *throughput*. El valor de salida de cada celda LSTM se calcula según la ecuación 10, y la salida de cada celda BiLSTM, que puede procesar simultáneamente datos secuenciales en diferentes direcciones, se calcula según las ecuaciones 22, 23 y 24 (Du et al., 2020):

$$\vec{h}_t = \vec{o}_t \odot \tanh(\vec{c}_t) \quad (22)$$

$$\overleftarrow{h}_t = \overleftarrow{o}_t \odot \tanh(\overleftarrow{c}_t) \quad (23)$$

$$h_t = \vec{h}_t \oplus \overleftarrow{h}_t \quad (24)$$

donde las flechas representan la dirección de procesamiento, y h_t denota la salida oculta final de cada celda BiLSTM, que es la concatenación de la salida hacia adelante \vec{h}_t y la salida hacia atrás \overleftarrow{h}_t .

La potencia de este modelo radica en el hecho de que la etapa convolucional se especializa en la extracción de las características más importantes de la serie de tiempo, dejando a la red recurrente la tarea de la predicción del *throughput*. La utilización de celdas BiLSTM en la etapa final permite realizar un mejor pronóstico cuando la longitud de las ventanas de entrada y salida es mayor, ya que se aumenta la capacidad de memoria y se mejora la atención sobre los valores de entrada menos recientes.

3.3 Premisas para la predicción de violaciones de SLA en el segmento eMBB a partir de la predicción del *throughput*

En la presente investigación se asumen premisas para, a partir del pronóstico del *throughput*, poder realizar la predicción de posibles violaciones de los SLA de las aplicaciones pertenecientes al servicio genérico eMBB. A continuación, se enuncian las premisas y se describen matemáticamente, también se realizan las acotaciones respectivas.

3.3.1 Premisas para la predicción

Se asume que el SLA para una determinada aplicación dentro del servicio genérico eMBB cuenta con umbrales diferenciados para el KPI *throughput* según el escenario de conexión, tal y como se expone en la ecuación 25:

$$SLA_{App} = \{SLA_{App_{escenario_1}}, SLA_{App_{escenario_2}}, \dots, SLA_{App_{escenario_N}}\} \quad (25)$$

Sea la siguiente definición de variables:

Th_p : *throughput* pronosticado

U_x : umbral del servicio genérico eMBB para un escenario de conexión X

U_y : umbral del servicio genérico eMBB para un escenario de conexión Y

Tal que: $U_y > U_x$

A continuación, se exponen los casos que se deben tener en cuenta:

Caso 1: Sea un UE, utilizando una aplicación del servicio genérico eMBB y conectado al escenario X.

✓ Si $Th_p > U_x \rightarrow$ se asume que ocurrirá una violación del SLA_{App} para el escenario X ($SLA_{App_{escenario.X}}$)

entonces si $U_y > Th_p > U_x \rightarrow$ se asume que el UE cambiará hacia el escenario Y, por tanto, la red debe garantizar los recursos necesarios para cumplir con el $SLA_{App_{escenario.Y}}$

Caso 2: Sea un UE, utilizando una App. del servicio genérico eMBB y conectado al escenario Y.

✓ Si $Th_p > U_y \rightarrow$ se asume que ocurrirá una violación del SLA_{App} para el escenario Y ($SLA_{App_{escenario.Y}}$)

✓ Si $U_y > Th_p < U_x \rightarrow$ se asume que el UE cambiará hacia el escenario X, por tanto, no habrá violación del SLA_{App} (habrá sobre aprovisionamiento de recursos, por tanto, la red deberá realizar una reasignación de recursos)

Nota 1: Se asume que los escenarios de conexión son Macro Urbano y Urbano Denso, y la transición entre ellos dependerá de los parámetros de movilidad y de cobertura.

Nota 2: Para cada escenario de conexión, además del umbral máximo (100%), se añaden umbrales de decisión de tal manera que se pueda acotar mejor los valores del *throughput* pronosticado. Se usan umbrales de decisión (empíricamente seleccionados) para el 35% y el 70% del umbral correspondiente a un escenario de conexión.

Nota 3: los umbrales utilizados son estáticos. Los umbrales máximos coinciden con los valores objetivo, definidos por el 3GPP para los escenarios Macro Urbano y Urbano Denso.

3.3.2 Función de costo por error de predicción

Sea:

Th_p : *throughput* pronosticado

Th_r : *throughput* real

U : umbral del servicio

Si el sistema de predicción propuesto se acopla a la plataforma 5G, habría que tener en cuenta lo siguiente:

Caso 1:

- ✓ Si $Th_p > U > Th_r$ → El sistema predice una violación del SLA erróneamente lo que implica un sobre aprovisionamiento de recursos de red (en este caso la Plataforma 5G asignará más recursos de los necesarios debido al error en el pronóstico).

Caso 2:

- ✓ Si $Th_p < U < Th_r$ → Se produce una violación del SLA sin ser detectada con anticipación por el sistema, lo que implica que el aprovisionamiento de recursos será insuficiente (en este caso la Plataforma 5G no asignará los recursos necesarios debido al error en el pronóstico).

Se pueden calcular las penalizaciones asociadas a los errores cometidos en la predicción utilizando una función de costo, la cual se expresa matemáticamente según la ecuación 26 (Bega et al., 2019).

$$L(x) = \begin{cases} \beta & \text{si } x \leq 0 \\ \gamma \cdot x & \text{si } x > 0 \end{cases} \quad (26)$$

donde:

- ✓ $L(x)$: función de pérdida por error de predicción.
- ✓ c_s^j : recursos que se aprovisionan según el pronóstico del desempeño del servicio para el tiempo t , *datacenter* j y *network slice* s .
- ✓ d_s^j : recursos que se necesitarían aprovisionar según desempeño real del servicio.
- ✓ $x = c_s^j - d_s^j$
- ✓ β : se asume como un costo fijo que representa la compensación que el operador tiene que pagar a un cliente cuando no se cumple el SLA.
- ✓ γ : es el costo unitario de los recursos que se asignan al *network slice*.

Entonces:

- ✓ Si $c_s^j > d_s^j$ → el operador sobre aprovisiona el *network slice*.
- ✓ Si $c_s^j < d_s^j$ → ocurre una violación del SLA para el *network slice* objetivo.

Es importante destacar, que una predicción perfecta de las violaciones de los SLA permite anticipar exactamente los recursos necesarios para garantizar la QoS asociada a la aplicación o servicio de red (entonces se cumple que $c_s^j = d_s^j$), evitando con ello cualquier penalización (Bega et al., 2019).

En la figura 38 se muestra la representación gráfica de la función de costo:

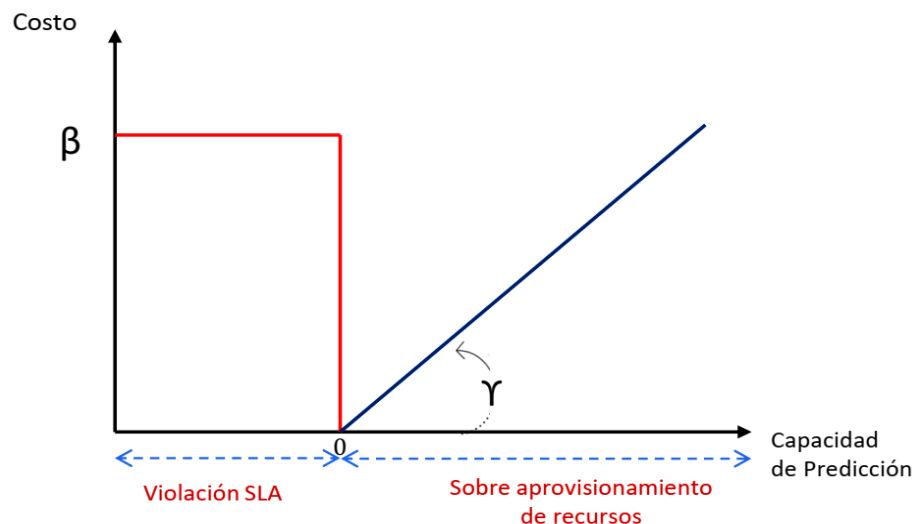


Figura 38. Representación gráfica de la función de costo asociada a los errores de predicción (Bega et al., 2019).

3.4 Combinación de la predicción del *throughput* y del VHO para prevenir violaciones de los SLA en escenarios de alta movilidad

Los diferentes modos de acceso y los *handoff* entre radio bases, debido a la movilidad del usuario, presentan desafíos adicionales para ofrecer un alto nivel de calidad de experiencia (QoE) para el usuario final; por otra parte, las aplicaciones móviles demandan cada vez más de un alto *throughput* (Phemina Selvi and Sendhilnathan, 2017), (Arshad et al., 2016). Debido a lo anterior, surge la necesidad de combinar la capacidad de predicción del *throughput* con la predicción del *handoff*, más específicamente con el VHO entre 4G y 5G. La combinación de la predicción de ambos parámetros brinda una mayor información para realizar ajustes proactivos a fin de evitar violaciones de los SLA (Mei et al., 2021).

En la presente investigación se lleva a cabo, además de la predicción del *throughput*, la predicción del VHO. Para ello se utilizan las dos arquitecturas propuestas en el epígrafe 3.2, con la diferencia de que se configuran las redes neuronales para que a la salida se obtenga la predicción del parámetro *network mode* (tipo de radio), o sea el KPI que dentro del *dataset* corresponde al *handoff vertical* entre 4G y 5G.

Por otra parte, en este caso también es importante definir umbrales de decisión para poder predecir con mayor precisión los posibles VHO que puedan ocurrir:

Sea la siguiente definición de variables:

Vh_p : VHO pronosticado

U_{5G} : umbral correspondiente al modo de red 5G

U_{4G} : umbral correspondiente al modo de red 4G

Tal que: $U_{5G} > U_{4G}$, donde se asume a $U_{5G} = 4.8$ y $U_{4G} = 4.2$

Caso 1: Sea un UE, conectado a la red móvil en modo de red (tipo de radio) 4G

✓ Si $U_{5G} > Vh_p > U_{4G}$ → se excedió el umbral correspondiente a 4G, pero, aunque es probable que ocurra un VHO, aun no se ha alcanzado el umbral de 5G.

Sino si $U_{5G} < Vh_p > U_{4G}$ → se efectuó un VHO desde 4G hacia 5G

Caso 2: Sea un UE, conectado a la red móvil en modo de red (tipo de radio) 5G

- ✓ Si $U_{5G} > Vh_p > U_{4G}$ → se excedió el umbral correspondiente a 5G (se aleja por la izquierda), pero, aunque es probable que ocurra un VHO, aun no se ha alcanzado el umbral de 4G (se acerca por la izquierda).

Sino si $U_{5G} > Vh_p < U_{4G}$ → se efectuó un VHO desde 5G hacia 4G

Nota: los valores de U_{5G} y U_{4G} se asumen empíricamente y están en correspondencia con la escala de valores que se utiliza por defecto en el *dataset*, donde se asignan los valores 5 y 4 a los tipos de radio 5G y 4G respectivamente.

Existen otros posibles casos de mayor complejidad como son el caso de los VHO del tipo *ping pong*. Debido a este efecto, hay *handoffs* innecesarios en la red celular, lo que reduce la eficiencia de energía y la QoS (Kene and Haridas, 2020), (Nyangaresi et al., 2020). El efecto *ping pong* es una condición muy peligrosa que debe ser cuidadosamente resuelta, pero ese aspecto no constituye un objetivo de esta investigación.

Por último, en la tabla 15 se muestra un resumen de la combinación de la predicción del *throughput* y el VHO para prevenir violaciones de los SLA en escenarios de alta movilidad. Se brindan además las posibles acciones de asignación de recursos de CN, y las causas probables de la variación del *throughput* con respecto al VHO y al tráfico demandado por las aplicaciones.

Tabla 15. Combinación de la predicción del *throughput* y del *handoff vertical* para prevenir violaciones de SLA en escenarios de alta movilidad (elaboración propia).

<i>Throughput</i>	<i>Handoff Vertical</i>	Acción	Causa
Aumenta	4G → 5G	Solicitar recursos de CN para garantizar QoS de extremo a extremo según estándar 5G.	Es muy probable que el <i>throughput</i> tiende a aumentar debido al <i>handoff vertical</i> , sobre todo en aplicaciones que funcionan bajo el principio <i>Best Effort</i> .
Disminuye	4G → 5G	Los recursos asignados son suficientes.	Es muy probable que la disminución del <i>throughput</i> se deba a la disminución de la actividad de la aplicación.
Aumenta	5G → 4G	Solicitar recursos de CN para garantizar QoS de extremo a extremo según estándar 4G	Es muy probable que el aumento del <i>throughput</i> se deba al aumento de la actividad de la aplicación.
Disminuye	5G → 4G	Solicitar recursos de CN para garantizar QoS de extremo a extremo según estándar 4G.	Es muy probable que el <i>throughput</i> tiende a disminuir debido al <i>handoff vertical</i> , sobre todo en aplicaciones que funcionen bajo principio <i>Best Effort</i> .

3.5 Diagrama general del modelo de predicción de violaciones de SLA

En la figura 39 se muestra el diagrama general del modelo de predicción de violaciones de SLA que se propone. Cada uno de los bloques funcionales del mismo ya fue analizado en detalle en los epígrafes del presente capítulo. Es importante destacar que, en el bloque funcional correspondiente a la solución de ML, los subbloques entrenamiento, validación y pruebas no estarían presentes si el modelo estuviera siendo utilizado en una plataforma 5G en producción, de ser ese el caso el modelo ya estuviera entrenado y listo para ser utilizado con cualquier conjunto de datos. En el caso del bloque funcional que implementa la lógica de predicción basado en umbrales de decisión, este tiene como fundamento las premisas descritas en el epígrafe 3.3. Por otra parte, el bloque que implementa la tabla de decisiones se basa en el análisis realizado en el epígrafe 3.4, que aborda la combinación de la predicción del *throughput* y el VHO en entornos de alta movilidad.

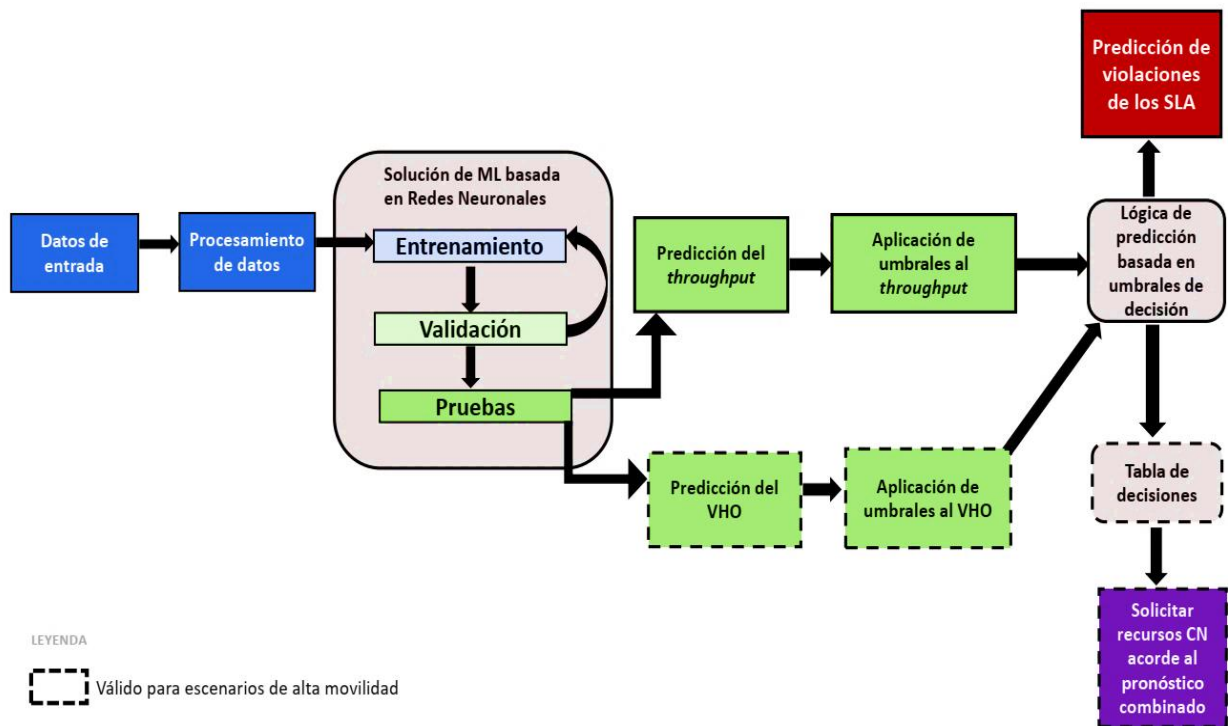


Figura 39. Diagrama general del modelo de predicción de violaciones de SLA (elaboración propia).

3.6 Conclusiones

En este capítulo se propuso un modelo de predicción de violaciones de SLA, el cual incluye la propuesta de dos arquitecturas de redes neuronales para la predicción del *throughput* y el VHO; y la definición de premisas para realizar la predicción de los SLA.

Se puede concluir lo siguiente:

- ✓ La predicción del *throughput* de extremo a extremo para un servicio de red específico constituye un gran desafío ya que sobre este inciden, además de los factores de impacto analizados en el capítulo 2, el tipo de arquitectura NG-RAN implementada, el modo 5G desplegado y el patrón de consumo de datos del usuario final.

- ✓ Existen varios trabajos previos que han abordado la predicción del *throughput* utilizando redes neuronales. Lo anterior sirve de base para diseñar nuevas redes neuronales que puedan ser utilizadas en la predicción del *throughput*.
- ✓ En el segmento de red genérico eMBB, el *throughput* es el KPI fundamental para el cumplimiento de los SLA de los servicios y aplicaciones de red. Por tanto, la predicción del *throughput* puede ser útil para predecir las violaciones de los SLA en dicho segmento.
- ✓ La predicción combinada del *throughput* y el VHO puede ser útil para prevenir violaciones de los SLA en escenarios de alta movilidad.

En el próximo capítulo se lleva a cabo la implementación y validación del modelo propuesto.

Capítulo 4. Validación del modelo propuesto

En el capítulo 3 se describieron de manera detallada cada uno de los bloques que componen el modelo propuesto, tal y como se muestra en la figura 39. En este capítulo se muestran los resultados de la validación del modelo propuesto, que van desde los resultados obtenidos en la predicción del *throughput* y el VHO, hasta la implementación de los bloques predictivos basados en umbrales de decisión según las premisas definidas en el epígrafe 3.3, y los resultados obtenidos en la predicción de las violaciones de los SLA.

4.1 Marco de la simulación

Con el objetivo de evaluar la propuesta realizada en el capítulo anterior es necesario, primeramente, implementar y entrenar las redes neuronales diseñadas, y posteriormente, implementar las premisas utilizando el lenguaje de programación Python.

La implementación y entrenamiento de las redes neuronales se realiza sobre el entorno de desarrollo *Google Colab* utilizando *TensorFlow* y otras bibliotecas de Python, las cuales fueron expuestas en la tabla 10. En el caso de la biblioteca *TensorFlow*, esta fue desarrollada por Google para construir y entrenar redes neuronales, y para detectar y descifrar patrones y correlaciones, siendo liberada a la comunidad como herramienta *Open Source* en 2015. Esta biblioteca proporciona flexibilidad gracias a su arquitectura, la cual se muestra en la figura 40 (TensorFlow Team, 2017).

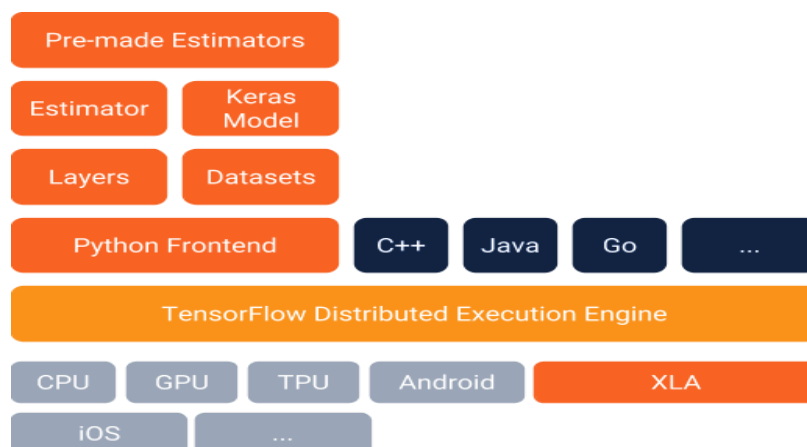


Figura 40. Arquitectura de *TensorFlow* (TensorFlow Team, 2017).

Para realizar los cálculos *Tensorflow* utiliza diagramas de flujo de datos que están representados por grafos, nodos y relaciones. Los nodos representan las operaciones matemáticas y las relaciones representan a los tensores, los cuales son objetos geométricos que describen relaciones lineales entre vectores geométricos, escalares y otros tensores. Los tensores y las matrices multidimensionales son tipos de objeto diferentes, mientras el primero es un tipo de función multilineal (la cual consta de varias variables vectoriales), el segundo es una estructura de datos adecuada para representar un tensor en un sistema de coordenadas (Analytics Vidhya, 2017) . En la figura 41 se muestra la representación gráfica de los tensores.

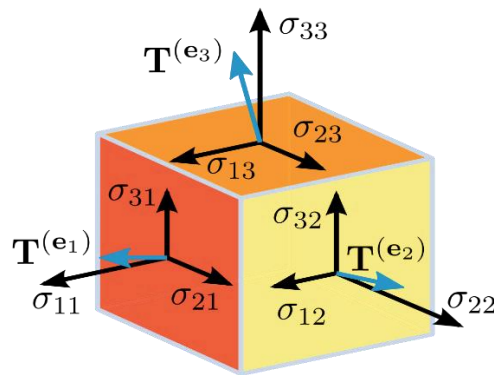


Figura 41. Representación gráfica de tres tensores (Analytics Vidhya, 2017).

Una vez que se tienen implementadas las redes neuronales diseñadas en el capítulo 3, se procede a su entrenamiento, validación y pruebas. Con el objetivo de garantizar que los resultados obtenidos sean estadísticamente válidos, y así poder comparar los mismos, el experimento se repite 30 veces para cada una de las métricas de error (MAE y RMSE), tamaño de ventana (60:30, 120:30, 180:30) y grupos de factores de impacto (geolocalización, geolocalización + movilidad, geolocalización + movilidad + factores de conexión). En los epígrafes 4.2 y 4.3 se muestran los resultados tabulados para la predicción del *throughput* y el VHO respectivamente. Es importante esclarecer que para el caso de la predicción del *throughput* se utilizan, además de los modelos de redes neuronales propuestos, dos modelos con arquitecturas genéricas: una CNN y una RNN, con el objetivo de enriquecer la comparativa.

Posteriormente, una vez que se realiza la predicción del *throughput*, se aplican los umbrales correspondientes según las premisas descritas en el epígrafe 3.3 del capítulo 3. Con la aplicación de estos

umbrales y utilizando una lógica predictiva basada en umbrales de decisión, se realiza la predicción de las violaciones de los SLA. En el epígrafe 4.4, se muestran las gráficas y tablas obtenidas en la predicción.

Por último, se realiza el análisis de la combinación de la predicción del *throughput* y el VHO, y se contrastan los resultados obtenidos con los casos descritos en la tabla 15 del epígrafe 3.4. Esta tabla puede ser considerada como una primera versión de la tabla de decisiones presente en el diagrama general del modelo mostrado en la figura 39; y a partir de las acciones en ella descritas se pueden solicitar los recursos CN necesarios para garantizar la QoS acordada en los SLA. Los casos y acciones descritos en la tabla 15 pueden aumentar en cantidad y complejidad, pero no se profundiza en ello pues no es el objetivo de esta investigación.

4.2 Resultados de las simulaciones para la predicción del *throughput*

En los experimentos llevados a cabo para la predicción del *throughput*, se obtuvieron los mejores resultados con un tamaño de ventana 120:30, o sea 120 pasos de tiempo a la entrada (en segundos) y 30 pasos de tiempo a la salida. Por otra parte, el grupo de factores de impacto que brindó los mejores resultados fue: geolocalización + movilidad + factores de conexión. En las tablas 16 y 17 se exponen los resultados estadísticos, tanto para la etapa de validación como la de pruebas.

Tabla 16. Resultados obtenidos en la predicción del *throughput* utilizando varios modelos de redes neuronales (métrica MAE) (elaboración propia).

	CNN	RNN	RNN-E1D1	RNN-E2D2	RNN-E3D3	RNN-CNN
Promedio (MAE Validación)	0.158373333	0.13328667	0.16114	0.16105667	0.14305667	0.123883333
Promedio (MAE Pruebas)	0.23816667	0.14405	0.2727	0.2604733	0.20342333	0.071783333
Desviación Estándar (MAE Validación)	0.013023902	0.02343498	0.03500785	0.03616678	0.03159805	0.005278981
Desviación Estándar (MAE Pruebas)	0.03591275	0.09232674	0.15409121	0.16369983	0.16285484	0.016549593

Tabla 17. Resultados obtenidos en la predicción del *throughput* utilizando varios modelos de redes neuronales (métrica RMSE) (elaboración propia).

	CNN	RNN	RNN-E1D1	RNN-E2D2	RNN-E3D3	RNN-CNN
Promedio (RMSE Validación)	0.197596667	0.20197667	0.28836333	0.22595333	0.23758	0.218546667
Promedio (RMSE Pruebas)	0.26404333	0.18725333	0.36891	0.24389	0.2494	0.130536667
Desviación Estándar (RMSE Validación)	0.010087633	0.01289666	0.05250238	0.03658961	0.03893687	0.00743861
Desviación Estándar (RMSE Pruebas)	0.0420363	0.08438681	0.14704959	0.16177375	0.16077869	0.019050685

En las figuras 42 y 43 se muestran los resultados obtenidos, para ambas métricas, en un gráfico de columnas:

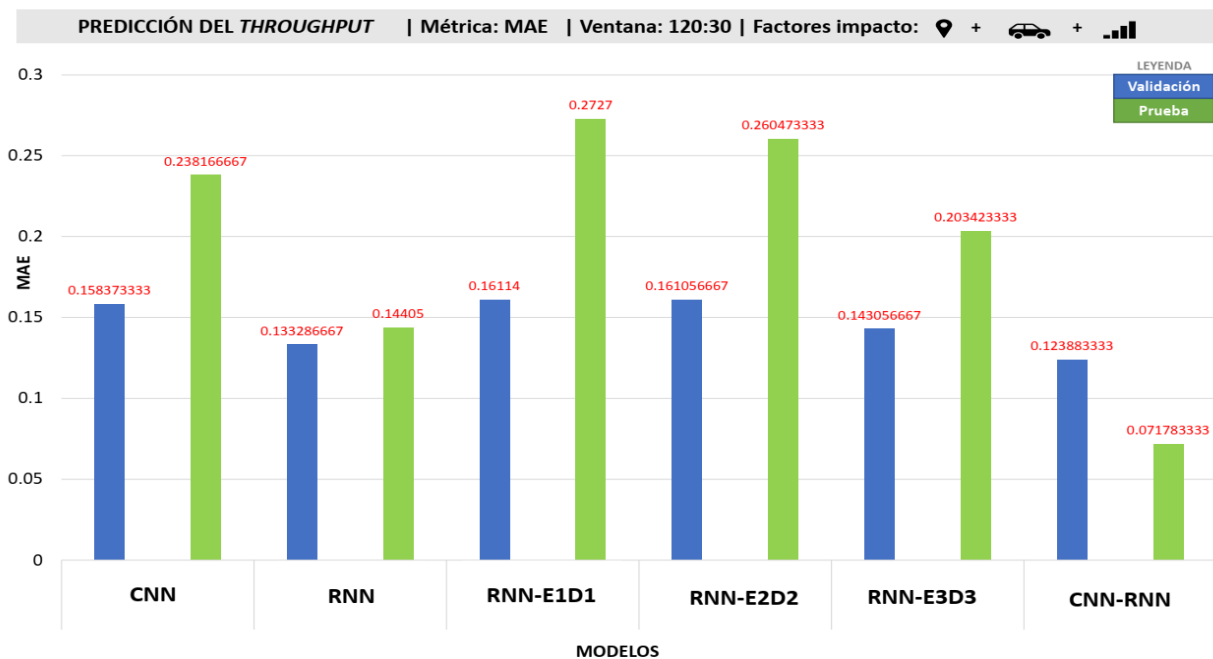


Figura 42. Resultados obtenidos en la predicción del *throughput* (métrica MAE) (elaboración propia).

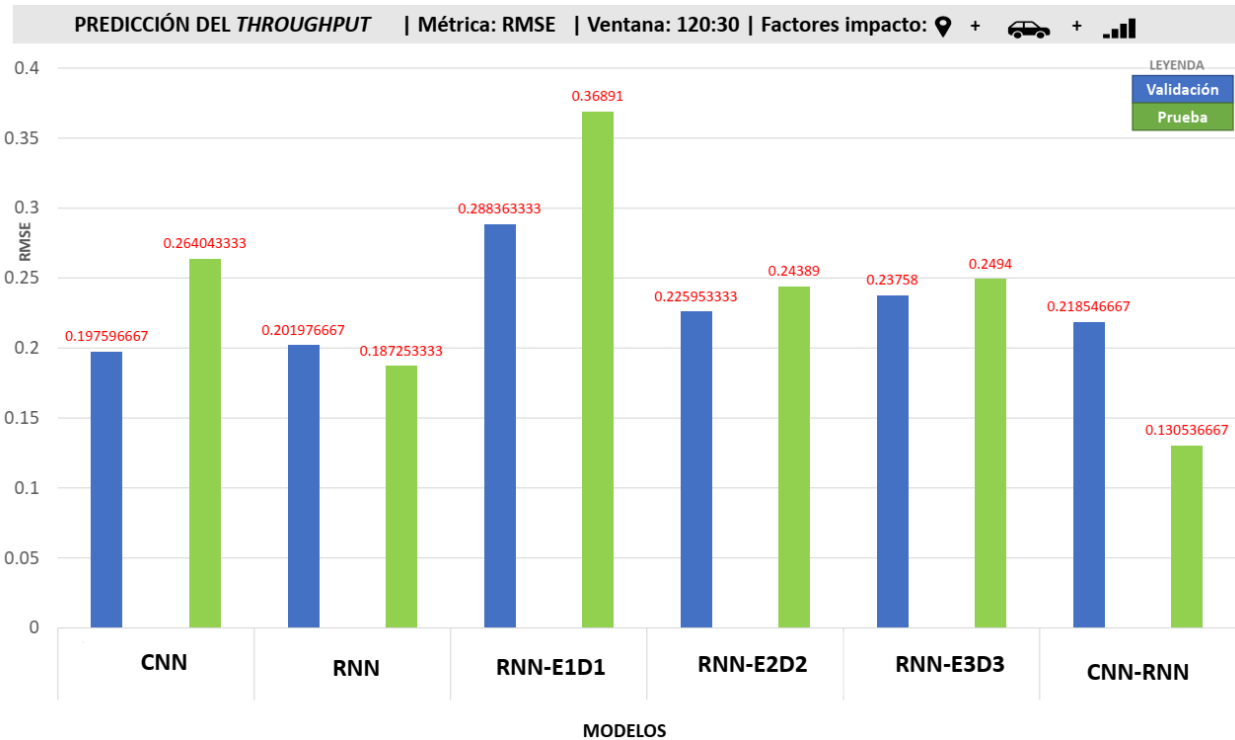


Figura 43. Resultados obtenidos en la predicción del *throughput* (métrica RMSE) (elaboración propia).

Se pueden resaltar los buenos resultados obtenidos, para ambas métricas, por el modelo mixto CNN-RNN. Sin embargo, el modelo basado en una RNN con arquitectura *Encoder-Decoder* no obtuvo buenos resultados, incluso cuando se compara con los modelos CNN y RNN con arquitecturas genéricas. Lo anterior puede deberse a una limitación presente en la arquitectura codificador-decodificador, la cual consiste en que el codificador debe comprimir toda la representación oculta de la información temporal pasada en un vector de contexto de longitud fija. Por lo tanto, la capacidad de predicción se degrada gradualmente a medida que aumenta la duración de la serie temporal de entrada (Bahdanau et al., 2014); esto unido a la multiplicidad de factores que tienen impacto sobre el *throughput* y a la complejidad de las relaciones entre ellos. El problema anterior puede ser resuelto con la incorporación de un mecanismo de atención, por ejemplo: una capa de atención temporal entre el codificador y el decodificador (Du et al., 2020). En el caso de la red mixta CNN-RNN, la capa de *Max-Pooling* funciona como un mecanismo de atención previo a las capas LSTM y BiLSTM, las cuales llevan a cabo la predicción del *throughput* (Wu et al., 2021).

4.3 Resultados de las simulaciones para la predicción del *handoff* vertical

En los experimentos llevados a cabo para la predicción del VHO, también se obtuvieron los mejores resultados con un tamaño de ventana 120:30. En este caso solo se utilizó el grupo de factores de impacto: geolocalización + movilidad + factores de conexión. Por otra parte, no se utilizaron en la comparativa los modelos CNN y RNN con arquitecturas genéricas, ya que no fueron capaces de predecir los cambios bruscos (transiciones) presentes en la serie de tiempo correspondiente al *network mode* (tipo de radio). En las tablas 18 y 19 se exponen los resultados estadísticos, tanto para la etapa de validación como la de pruebas.

Tabla 18. Resultados obtenidos en la predicción del VHO utilizando varios modelos de redes neuronales (métrica MAE) (elaboración propia).

	RNN-E1D1	RNN-E2D2	RNN-E3D3	RNN-CNN
Promedio (MAE Validación)	0.41154828	0.37675172	0.40927241	0.439244828
Promedio (MAE Pruebas)	0.33791724	0.39173793	0.35398276	0.421244828
Desviación Estándar (MAE Validación)	0.12592728	0.16179002	0.14760462	0.109399081
Desviación Estándar (MAE Pruebas)	0.13474806	0.17550807	0.18394602	0.078039169

Tabla 19. Resultados obtenidos en la predicción del VHO utilizando varios modelos de redes neuronales (métrica RMSE) (elaboración propia).

	RNN-E1D1	RNN-E2D2	RNN-E3D3	RNN-CNN
Promedio (RMSE Validación)	0.5754	0.60333667	0.57404	0.56544
Promedio (RMSE Pruebas)	0.51672667	0.55190667	0.53407	0.63109
Desviación Estándar (RMSE Validación)	0.1208223	0.14234074	0.15892338	0.08738036
Desviación Estándar (RMSE pruebas)	0.10269624	0.17436972	0.16481546	0.063924224

En las figuras 44 y 45 se muestran los resultados obtenidos, para ambas métricas, en un gráfico de columnas:

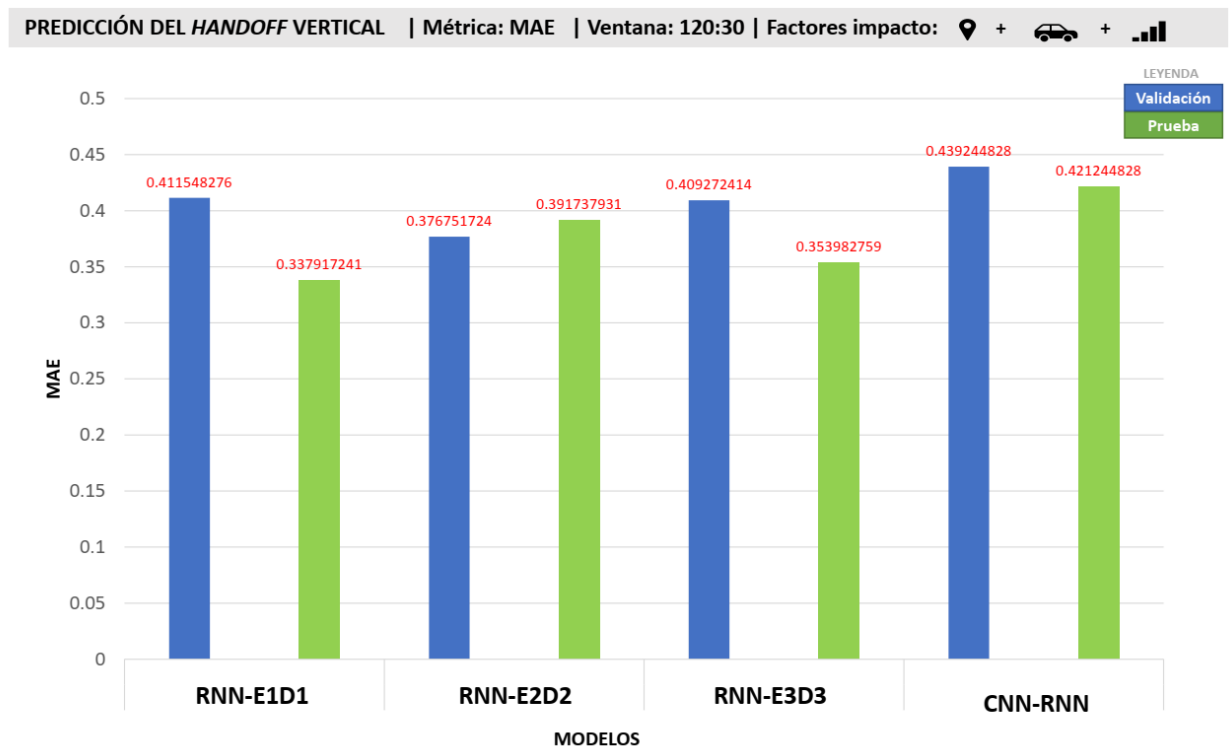


Figura 44. Resultados obtenidos en la predicción del VHO (métrica MAE) (elaboración propia).

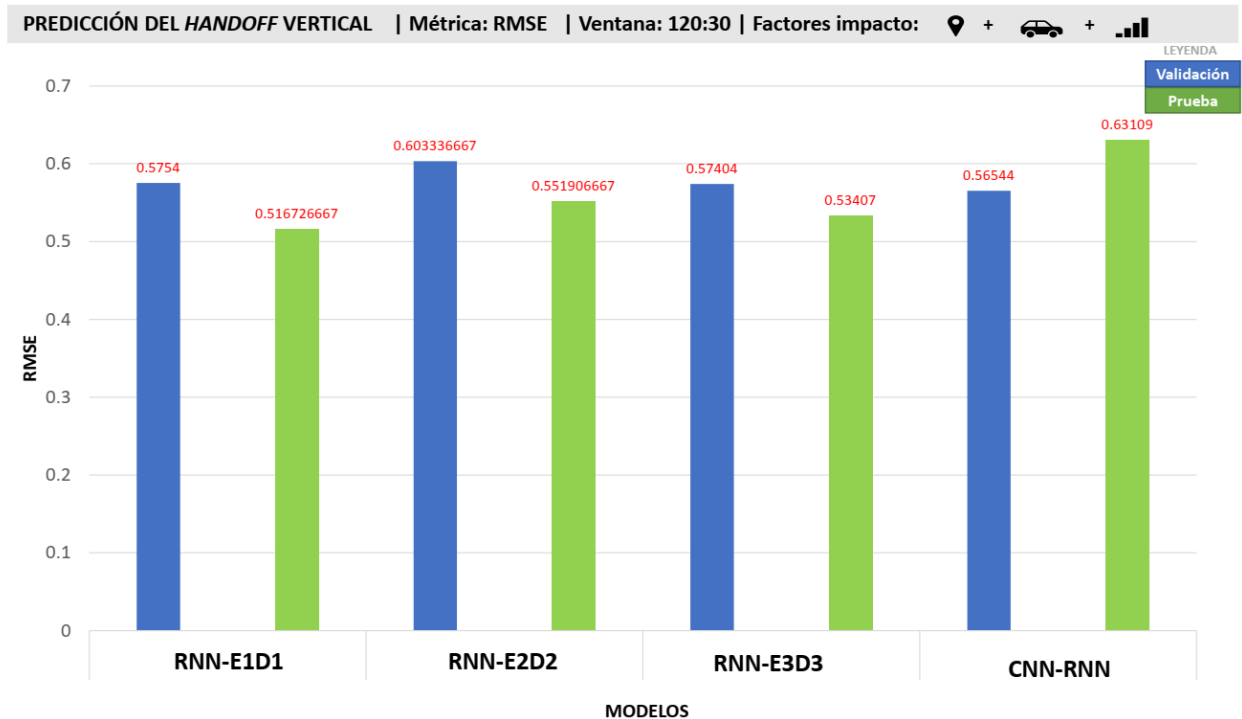


Figura 45. Resultados obtenidos en la predicción del VHO (métrica RMSE) (elaboración propia).

En el caso de la predicción del VHO, los modelos RNN con arquitectura *Encoder-Decoder* lograron predecir correctamente los cambios bruscos de la señal correspondiente al *network mode*, la cual es una señal del tipo ON-OFF. Cada vez que sucede un cambio brusco del *network mode* se produce un VHO, ya sea un cambio ascendente (VHO 4G→5G) o descendente (VHO 5G→4G). De los tres modelos *Encoder-Decoder* el de una capa (E1D1) fue el de mejores resultados, siendo además el más sencillo y por tanto el de más rápido entrenamiento.

La red mixta CNN-RNN también fue capaz de predecir con éxito los VHO, sin embargo, los resultados fueron ligeramente inferiores. Además, hay que destacar que la red CNN-RNN es más compleja que la red RNN-E1D1, por lo que la segunda constituye la mejor opción para realizar el pronóstico del VHO.

El éxito de la arquitectura *Encoder-Decoder* en la predicción del VHO, a diferencia de la predicción del *throughput*, se debe a que el *network mode* es una señal de solo dos valores de amplitud, y con menor variabilidad en el tiempo. Lo anterior le permite a esta arquitectura realizar buenas predicciones pese a carecer de un mecanismo de atención incorporado.

4.4 Predicción de las violaciones de los SLA

Una vez que las redes neuronales se encuentran correctamente entrenadas se procede a realizar la predicción. En la figura 46 se muestra un ejemplo de la predicción del *throughput* utilizando la red mixta CNN-RNN.

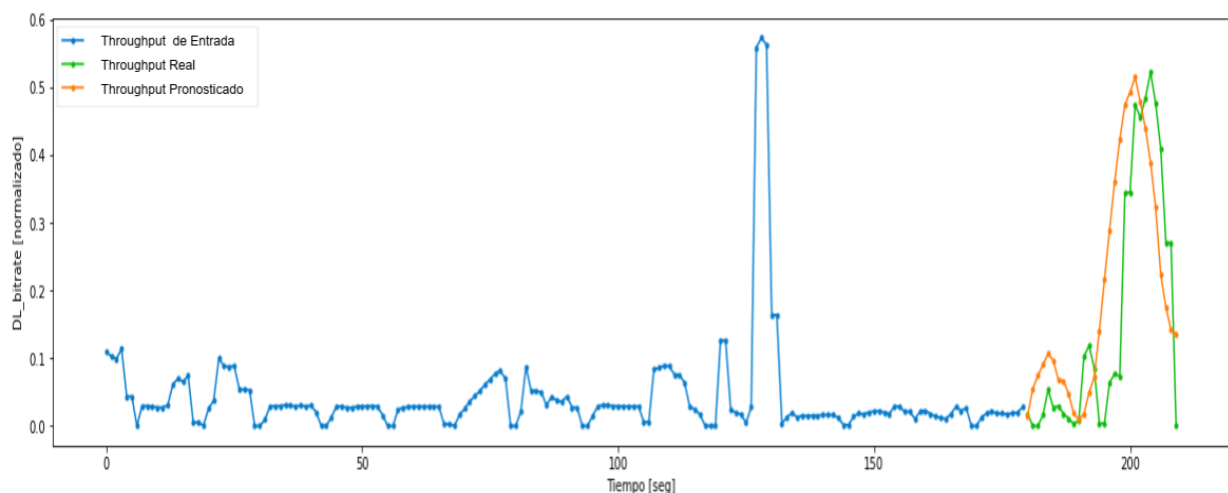


Figura 46. Predicción del *throughput* (elaboración propia).

En la figura 47 se muestra la aplicación de los umbrales máximos al *throughput*. En este caso los umbrales corresponden a los valores objetivos definidos por el 3GPP para los escenarios Zona Suburbana (Macro Urbana), Zona de *Downtown* (Urbano Denso) y Zona de interiores, siendo los dos primeros los que se tienen en cuenta para la predicción de las violaciones de los SLA.

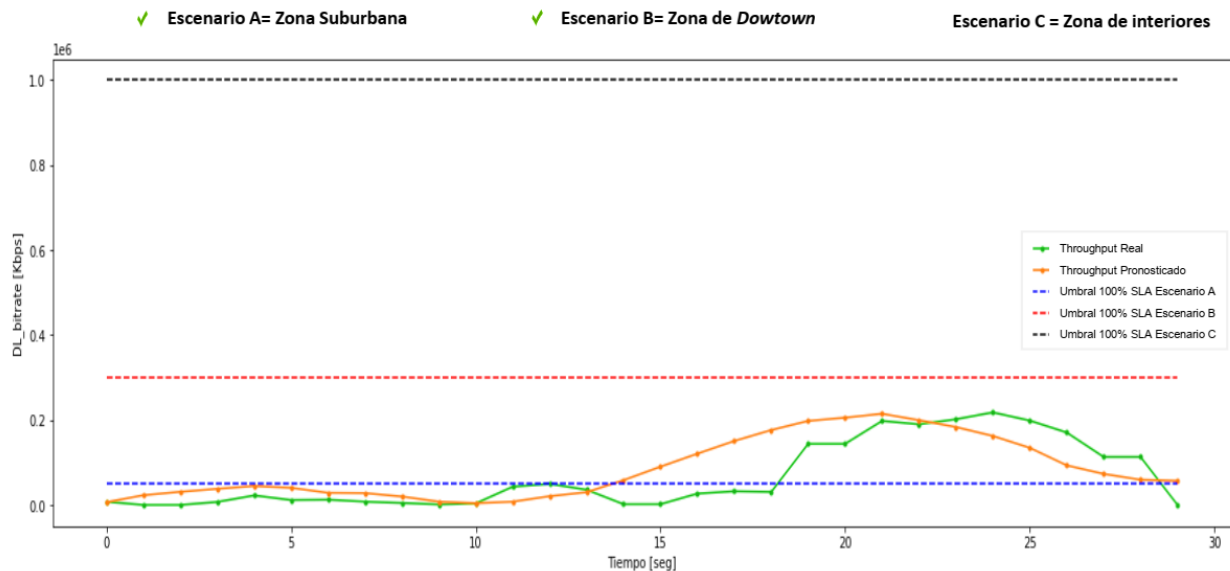


Figura 47. Aplicación de umbrales máximos al *throughput*. Escenarios: Zona Suburbana, Zona de *Downtown* y Zona de interiores (elaboración propia).

En la figura 48 se muestra la aplicación de los umbrales de decisión (35%, 70%, 100%) aplicados al *throughput* para el escenario Zona Suburbana. El objetivo de aplicar estos umbrales de decisión es acotar la señal del *throughput* y facilitar la toma de decisiones. Básicamente, el umbral máximo (100%) es el límite del escenario de conexión, por lo que si el *throughput* pronosticado excede el umbral máximo el sistema debe predecir una violación del SLA para dicho escenario. Los umbrales intermedios (35% y 70%) sirven para brindar información al *core* de la red (asumiendo que el sistema propuesto esté integrado a una plataforma 5G) del grado de utilización futuro (próximos 30 segundos) de los recursos de red asignados al servicio o aplicación del cliente.

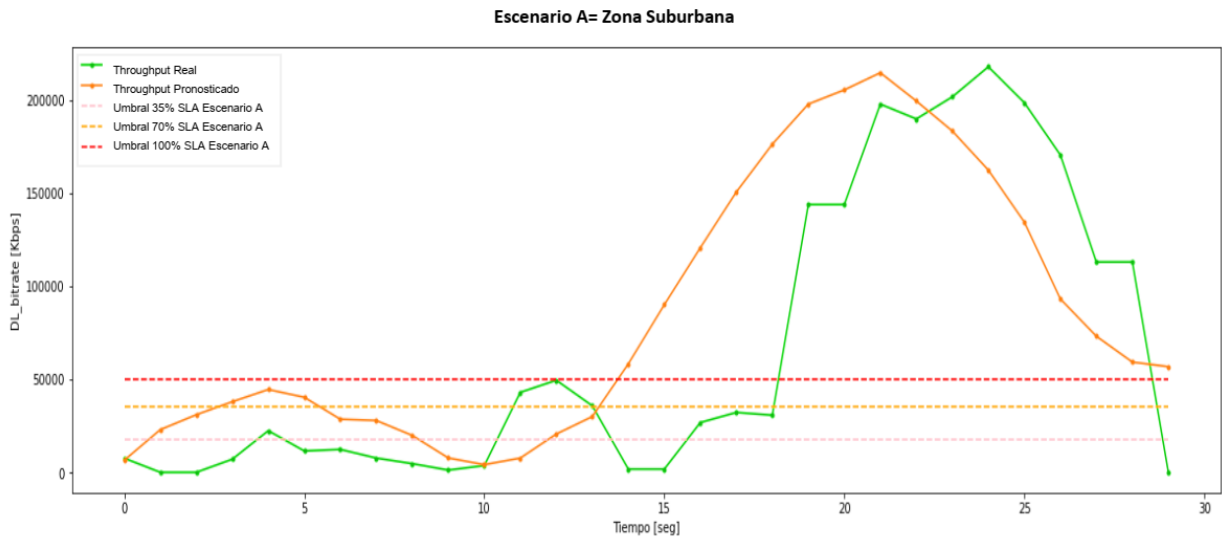


Figura 48. Aplicación de umbrales de decisión al *throughput* para el escenario Zona Suburbana (elaboración propia).

Posteriormente, utilizando una lógica predictiva basada en umbrales de decisión, se realiza el pronóstico de las violaciones de los SLA para el escenario en cuestión. La lógica predictiva se basa en las premisas del epígrafe 3.3, y se implementa a través de las ecuaciones 2.7 y 2.8, las cuales se muestran a continuación:

Sea la siguiente definición de variables:

U_d : umbral de decisión (%)

N_T : número total de muestras (coincide con el tamaño de la ventana de salida)

n : número de muestras que violan el umbral U_d

m_i : valor de la i -ésima muestra que viola el umbral U_d

P_v : probabilidad de violación del umbral U_d (%)

PED : promedio del exceso de demanda de tráfico para el umbral U_d (%)

$$P_v = \frac{n}{N_T} \cdot 100\% \quad (27)$$

$$PED = \frac{1}{n \cdot U_d} \cdot \sum_{i=1}^n m_i \cdot 100\% \quad (28)$$

Este bloque funcional, también implementa el cálculo de índices de efectividad con el objetivo de medir cuan efectivo es el modelo en la predicción de las violaciones de los SLA. Para ello también se calcula P_v y PED para el *throughput* real, y después se calculan los índices, tal y como se expone en las ecuaciones 29, 30 y 31.

Sea:

P_{vr} : probabilidad (real) de violación del umbral U_d (%)

PED_r : promedio del exceso de demanda de tráfico (real) para el umbral U_d (%)

I_v : índice de violaciones

I_{vr} : índice de violaciones reales

I_{pp} : índice de precisión de la predicción

$$I_v = P_v \cdot PED \cdot 10^{-4} \quad (29)$$

$$I_{vr} = P_{vr} \cdot PED_r \cdot 10^{-4} \quad (30)$$

$$I_{pp} = I_v - I_{vr} \quad (31)$$

donde si:

$I_{pp} = 0 \rightarrow$ predicción 100% efectiva

$I_{pp} > 0 \rightarrow$ sobre aprovisionamiento de recursos para el umbral U_d

$I_{pp} < 0 \rightarrow$ violaciones no detectadas para el umbral U_d

Nota 1: las violaciones de los SLA solo ocurren cuando el umbral $U_d = 100\%$ del valor objetivo definido por el 3GPP para un escenario determinado.

Nota 2: las violaciones de los umbrales $U_d = 35\%$ del valor objetivo y $U_d = 70\%$ del valor objetivo, solo brindan información sobre el grado de utilización futura de los recursos de red asignados a un cliente.

En la figura 49 se muestra la consola del sistema de predicción de violaciones de los SLA.

Cálculos para el KPI DL_Throughput para servicios genéricos eMBB en zonas Suburbanas

TABLA 1: ESTADÍSTICAS DE LA PREDICCIÓN

	Muestras que violan el umbral	Probabilidad de violación del Umbral(%)	Promedio del exceso de demanda de tráfico(%)
Umbral 35%	26	86.6667	544.782
Umbral 70%	19	63.3333	345.608
Umbral 100% SLA	16	53.3333	271.95

TABLA 2: ESTADÍSTICAS REALES

	Muestras que violan el umbral	Probabilidad de violación del Umbral(%)	Promedio del exceso de demanda de tráfico(%)
Umbral 35%	17	56.6667	648.742
Umbral 70%	13	43.3333	399.58
Umbral 100% SLA	10	33.3333	337.991

TABLA 3: CÁLCULO DE ÍNDICES DE EFECTIVIDAD

	Índice de violaciones (Iv)	Índice de violaciones reales (Ivr)	Índice de precisión de la predicción (Ipp)
Umbral 35%	4.72144	3.6762	1.04524
Umbral 70%	2.18885	1.73151	0.457339
Umbral 100% SLA	1.4504	1.12664	0.323761

NOTAS:

$Ipp = Iv - Ivr$

$Ipp = 0$ --> caso ideal (predicción 100% efectiva)

$Ipp > 0$ --> sobreaprovisionamiento de recursos

$Ipp < 0$ --> violaciones no detectadas de los umbrales

Figura 49. Consola del sistema de predicción de violaciones de SLA. En este caso se muestra el pronóstico de las violaciones de los SLA (fila roja) para el escenario Zona Suburbana (elaboración propia).

El mismo procedimiento de aplicar umbrales de decisión y calcular las violaciones se realiza para el escenario Zona de *Downtown*.

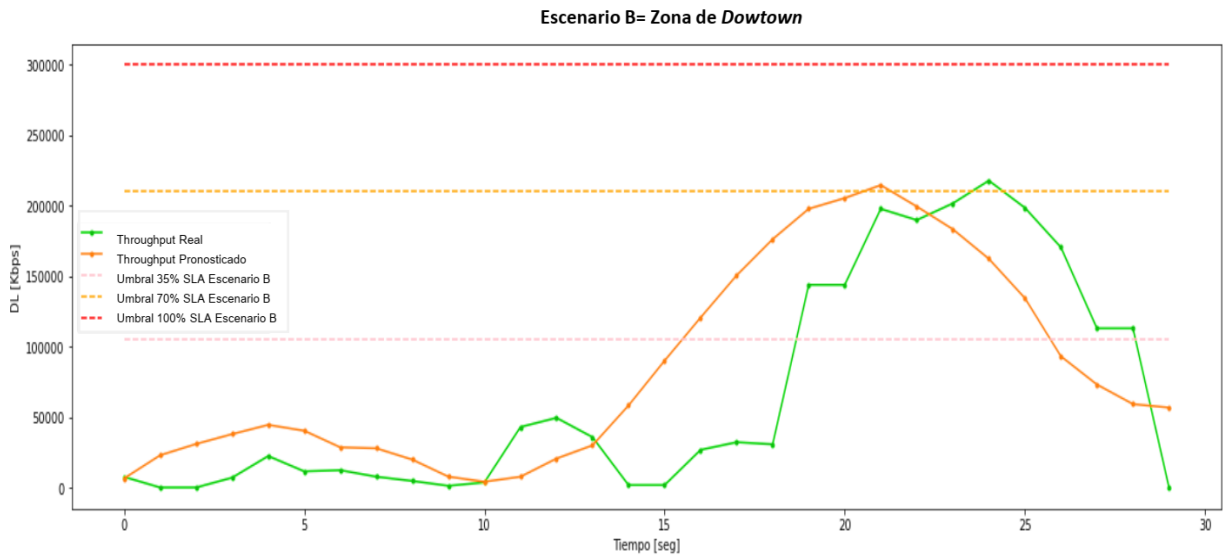


Figura 50. Aplicación de umbrales de decisión al *throughput* para el escenario Zona de *Downtown* (elaboración propia).

Cálculos para el KPI DL_Throughput para servicios genéricos eMBB en zonas Downtown

TABLA 1: ESTADÍSTICAS DE LA PREDICCIÓN

	Muestras que violan el umbral	Probabilidad de violación del Umbral(%)	Promedio del exceso de demanda de tráfico(%)
Umbral 35%	10	33.3333	166.202
Umbral 70%	1	3.33333	102.218
Umbral 100 SLA%	0	0	0

TABLA 2: ESTADÍSTICAS REALES

	Muestras que violan el umbral	Probabilidad de violación del Umbral(%)	Promedio del exceso de demanda de tráfico(%)
Umbral 35%	10	33.3333	160.948
Umbral 70%	1	3.33333	103.742
Umbral 100 SLA%	0	0	0

TABLA 3: CÁLCULO DE ÍNDICES DE EFECTIVIDAD

	Índice de violaciones (Iv)	Índice de violaciones reales (Ivr)	Índice de precisión de la predicción (Ipp)
Umbral 35%	0.554007	0.536494	0.0175129
Umbral 70%	0.0340728	0.0345808	-0.000508006
Umbral 100% SLA	0	0	0

NOTAS:
 Ipp=Iv-Ivr
 Ipp=0 --> caso ideal (predicción 100% efectiva)
 Ipp>0 --> sobreaprovisionamiento de recursos
 Ipp<0 --> violaciones no detectadas de los umbrales

Figura 51. Consola del sistema de predicción de violaciones de SLA. En este caso se muestra el pronóstico de las violaciones de los SLA (fila roja) para el escenario Zona de *Downtown* (elaboración propia).

Por último, se puede analizar la predicción del VHO para el mismo intervalo de tiempo futuro (30 segundos). En la figura 52 se puede observar que se pronostica un *handoff* vertical desde 4G hacia 5G.

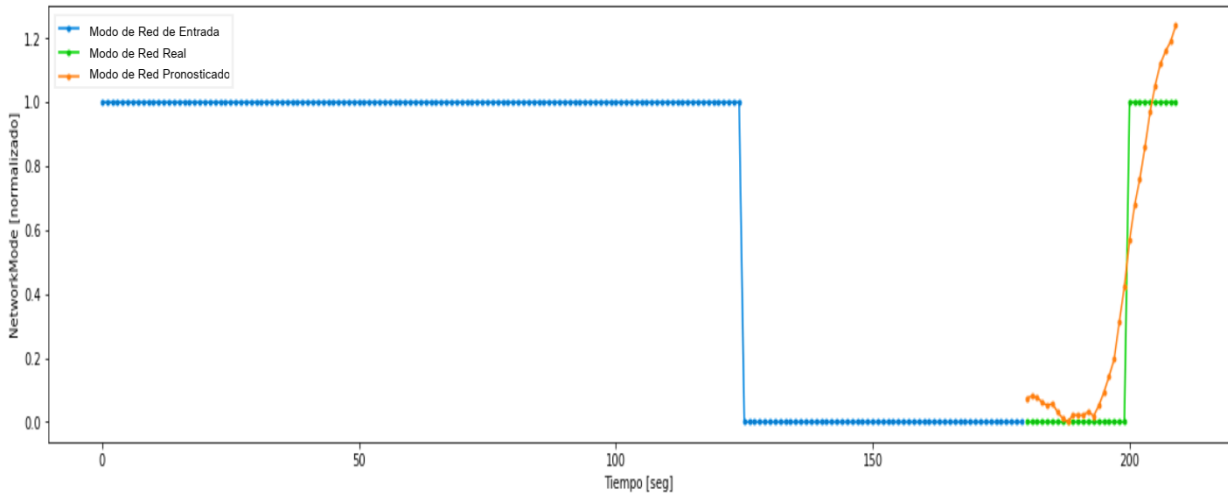


Figura 52. Predicción del *handoff* vertical (elaboración propia).

Al VHO también se le aplican umbrales de decisión para acotar la señal y facilitar la toma de decisiones. En la figura 53 se muestra la aplicación de los umbrales en los valores 4.8 y 4.2, tal y como se planteó en el epígrafe 3.4.

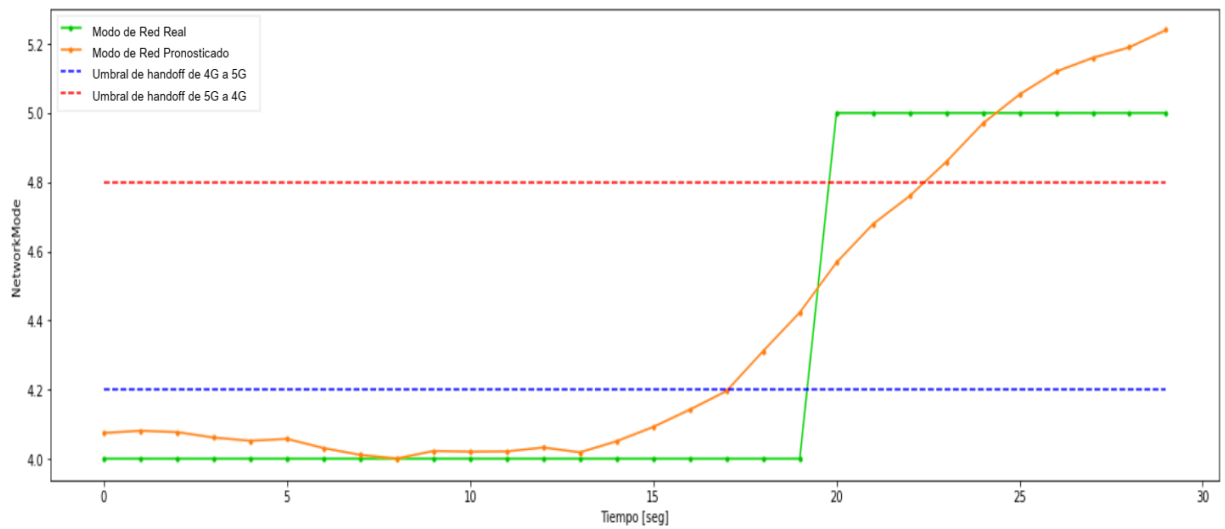


Figura 53. Aplicación de los umbrales al *handoff* vertical (elaboración propia).

Si combinamos el pronóstico anterior con el pronóstico mostrado en la figura 46, se puede observar que en el mismo intervalo de tiempo se pronostica un aumento brusco del *throughput*. El análisis combinado del pronóstico del VHO y el *throughput* a partir del ejemplo empírico anterior valida lo abordado en el epígrafe 3.4, donde se planteaba que la combinación de la predicción de ambos KPI podía ser útil para prevenir violaciones de los SLA en escenarios de alta movilidad, además de brindar información útil para realizar una asignación más eficiente de los recursos de red.

4.5 Conclusiones

En este capítulo se validó el modelo propuesto en el capítulo 3.

Se puede concluir que el modelo propuesto cumplió con el objetivo para el cual fue diseñado: la predicción de las violaciones de los SLA de servicios y aplicaciones de red del segmento genérico eMBB. Para ello fue necesario realizar primero la predicción del *throughput*, obteniéndose los mejores resultados en este aspecto con la red neuronal mixta CNN-RNN. También se realizó con éxito la predicción del VHO, obteniéndose los mejores resultados con la red neuronal RNN-E1D1; y comprobándose además que la combinación de la predicción del *throughput* y el VHO puede ser útil para prevenir las violaciones de los SLA en escenarios de alta movilidad.

Capítulo 5. Conclusiones y Recomendaciones

El análisis presentado en este trabajo y las publicaciones recientes que abordan problemas similares, permitieron establecer los principales aspectos teóricos sobre la gestión de la segmentación de red y los SLA en redes 5G. Asimismo, se abordaron los aspectos teóricos del aprendizaje automático y su aplicación en la gestión de las redes 5G.

Como resultado del estudio realizado, se propuso un modelo para la predicción de las violaciones de los SLA de aplicaciones de red del segmento genérico eMBB. Como parte del modelo propuesto, se seleccionan dos arquitecturas de redes neuronales para realizar la predicción del *throughput* y el *handoff* vertical.

Con respecto a las redes neuronales propuestas, el análisis de los resultados obtenidos a partir de los experimentos realizados permitió llegar a las siguientes conclusiones:

- ✓ La red neuronal mixta CNN-RNN fue la que brindó mejores resultados a la hora de predecir el *throughput*; obteniéndose los mejores resultados cuando se incluyeron todos los factores de impacto y una ventana de 2 minutos de observación y 30 segundos de predicción del *throughput* futuro.
- ✓ La red neuronal RNN con arquitectura *Encoder-Decoder* de una capa (RNN-E1D1) fue la que brindó mejores resultados a la hora de predecir el *handoff* vertical, obteniéndose los mejores resultados cuando se incluyeron todos los factores de impacto y una ventana de 2 minutos de observación y 30 segundos de predicción del *handoff* vertical futuro.

Con respecto a la predicción de las violaciones de los SLA, el análisis de los resultados de la validación del modelo propuesto permitió llegar a las siguientes conclusiones:

- ✓ La predicción de las violaciones de los SLA se realizó satisfactoriamente aplicando umbrales al *throughput* pronosticado según los valores objetivo de los escenarios Macro Urbano y Urbano Denso definidos por el 3GPP para el servicio genérico eMBB. El modelo propuesto es aplicable a todas las aplicaciones que operen en el segmento eMBB.

- ✓ Para escenarios de alta movilidad, el análisis combinado de la predicción del *throughput* y la predicción del *handoff* vertical ofrece información adicional para la predicción de las violaciones de los SLA, e incluso se pudieran prevenir dichas violaciones si se toman acciones correctivas basadas en la información brindada por ambos pronósticos.

5.1 Contribuciones al conocimiento

Esta investigación contribuye al problema de la predicción del comportamiento de indicadores clave de QoS en la creación de instancias de *network slicing* y el cumplimiento de los SLA asociados. Los aportes de este trabajo de tesis son:

- ✓ Se aborda el problema de la predicción de violaciones de los SLA a partir de la predicción del *throughput*.
- ✓ Se aborda el problema de la prevención de violaciones de los SLA a partir de la predicción combinada del *throughput* y el *handoff* vertical.
- ✓ Se proponen dos arquitecturas de redes neuronales para la predicción del *throughput* y el *handoff* vertical.
- ✓ Se propone un modelo para la predicción de las violaciones de los SLA de aplicaciones de red en el segmento genérico eMBB.
- ✓ Se desarrolla un marco de simulación para realizar el entrenamiento, validación y pruebas de las redes neuronales propuestas.
- ✓ Se realiza un análisis comparativo de los resultados obtenidos con las dos redes neuronales.
- ✓ Se desarrolla un marco de simulación para validar el modelo de predicción de violaciones de los SLA a partir de la implementación de una lógica predictiva basada en umbrales de decisión.

5.2 Limitaciones de la investigación

La siguiente sección describe las limitaciones de esta investigación:

- ✓ El modelo propuesto para la predicción de las violaciones de los SLA presenta la limitación de que el bloque funcional correspondiente a la solución de ML incluye a los subbloques entrenamiento, validación y pruebas; los cuales no estarían presentes si el modelo estuviera siendo utilizado en una plataforma 5G en producción, caso en el cual el modelo ya debe estar entrenado y listo para realizar la predicción con cualquier conjunto de datos.
- ✓ El ajuste de los hiperparámetros de las redes neuronales se realiza de forma manual, ya que implica el mínimo esfuerzo de codificación. Sin embargo, si el modelo va a ser utilizado en un entorno de producción es necesario utilizar otro algoritmo de optimización que garantice la obtención de los mejores resultados en el entrenamiento de la red.
- ✓ En la tabla 15 del epígrafe 3.4 del capítulo 3, los casos y acciones descritos pueden aumentar en cantidad y complejidad, pero no se profundiza en ello pues no es el objetivo de esta investigación. Dentro de los casos que no se tuvieron en cuenta y tienen un impacto significativo están los *handoff* verticales del tipo *ping pong*.
- ✓ Para la predicción de las violaciones de los SLA solo se tuvieron en cuenta los escenarios de conexión Macro Urbano y Urbano Denso.

5.3 Trabajo futuro

Después de realizar el análisis de los resultados obtenidos en la validación del modelo propuesto, se consideran los siguientes trabajos a futuro:

- ✓ Entrenar las redes neuronales utilizando un algoritmo que automatice el ajuste de los hiperparámetros.

- ✓ Incorporar un mecanismo de atención a la RNN con arquitectura *Encoder-Decoder*, repetir los experimentos y realizar un análisis comparativo con las demás redes neuronales analizadas en la presente investigación.
- ✓ Ampliar el estudio realizado mejorando la tabla 15 del epígrafe 3.4 del capítulo 3, de tal manera que se incluyan más casos y acciones para garantizar una prevención más efectiva de las violaciones de los SLA en entornos de alta movilidad.
- ✓ Ampliar el estudio incorporando el escenario de conexión de Zona de interiores.

Literatura citada

- 3GPP. 2017. 3GPP specification series: 38series. Retrieved June 29, 2021, from <https://www.3gpp.org/DynaReport/38-series.htm>
- 3GPP. 2018. 3GPP TR 28.801 V15.1.0; Study on management and orchestration of network slicing for next generation network (Release 15). Retrieved August 13, 2021, from 3GPP TR 28.801 V15.1.0 website: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3091>
- 3GPP. 2021a. 3GPP TS 22.261 V18.4.0- 5G ; Service requirements for the 5G system. Stage 1 (Release 18). Retrieved June 24, 2021, from 3GPP TS 22.261 V18.4.0 website: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107>
- 3GPP. 2021b. 3GPP TS 28.554 V17.4.0 - 5G; Management and orchestration; 5G end to end Key Performance Indicators (KPI). Retrieved September 29, 2021, from 3GPP TS 28.554 website: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415>
- 3GPP. 2021c. 3GPP TS 38.306 V16.6.0 - NR; User Equipment (UE) radio access capabilities. Retrieved September 30, 2021, from 3GPP TS 38.306 website: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3193>
- 5G Americas. 2016. Network Slicing for 5G Networks & Services,. Retrieved August 6, 2020, from https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Americas_Network_Slicing_11.21_Final.pdf
- 5G Americas. 2020. Innovations-in-5G-Backhaul-Technologies. Retrieved October 1, 2021, from <https://www.5gamericas.org/wp-content/uploads/2020/06/Innovations-in-5G-Backhaul-Technologies-WP-PDF.pdf>
- Alcalá, U. 2021. Lenguajes de programación para Data Science - Máster en Data Science. Retrieved October 13, 2021, from <https://www.master-data-scientist.com/lenguajes-programacion-data-science/>
- Alimpertis, E., Markopoulou, A., Butts, C. T., Psounis, K. 2019. City-wide signal strength maps: Prediction with random forests. En The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, May 13, 2019, Association for Computing Machinery, Inc, pp. 2536–2542. doi:10.1145/3308558.3313726
- Alvarez, J. M. 2018. El perceptrón como neurona artificial - Blog de Jose Mariano Alvarez. Retrieved August 15, 2021, from <http://blog.josemarianoalvarez.com/2018/06/10/el-perceptron-como-neurona-artificial/>

- Alzate, M. A. 2004. Modelos de tráfico en análisis y control de redes de Comunicaciones. *Revista Ingeniería*, 9(1), 63–87. doi:10.14483/23448393.2744
- Analytics Vidhya. 2017. Understanding Tensors and Graphs to get you started in Deep Learning. Retrieved October 31, 2021, from Analytics Vidhya website: <https://www.analyticsvidhya.com/blog/2017/03/tensorflow-understanding-tensors-and-graphs/>
- Arshad, R., ElSawy, H., Sorour, S., Al-Naffouri, T. Y., Alouini, M. S. 2016. Handover Management in 5G and Beyond: A Topology Aware Skipping Approach. *IEEE Access*, 4, 9073–9081. from <https://arxiv.org/abs/1611.07366v1>
- Asadi, A., Muller, S., Sim, G. H., Klein, A., Hollick, M. 2018. FML: Fast Machine Learning for 5G mmWave Vehicular Communications. En *Proceedings - IEEE INFOCOM*, October 8, 2018, Institute of Electrical and Electronics Engineers Inc., 2018-April, pp. 1961–1969. doi:10.1109/INFOCOM.2018.8485876
- Bahdanau, D., Cho, K., Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. from <https://arxiv.org/abs/1409.0473v7>
- Balevi, E., Gitlin, R. D. 2018. Unsupervised machine learning in 5G networks for low latency communications. En *2017 IEEE 36th International Performance Computing and Communications Conference, IPCCC 2017*, February 2, 2018, Institute of Electrical and Electronics Engineers Inc., 2018-Janua, pp. 1–2. doi:10.1109/PCCC.2017.8280492
- Barona, L., Maestre, J., García, L. 2017. An Approach to Data Analysis in 5G Networks. *Entropy*, 19(2), 74. doi:10.3390/e19020074
- Bartelt, J., Vucic, N., Camps-Mur, D., Garcia-Villegas, E., Demirkol, I., Fehske, A., Grieger, M., Tzanakaki, A., Gutiérrez, J., Grass, E., Lyberopoulos, G., Fettweis, G. 2017. 5G transport network requirements for the next generation fronthaul interface. *Eurasip Journal on Wireless Communications and Networking*, 2017(1). doi:10.1186/S13638-017-0874-7
- Bega, D., Gramaglia, M., Fiore, M., Banchs, A., Costa-Perez, X. 2019. DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning. En *Proceedings - IEEE INFOCOM*, April 1, 2019, Institute of Electrical and Electronics Engineers Inc., 2019-April, pp. 280–288. doi:10.1109/INFOCOM.2019.8737488
- Ben Yahia, I. G., Bendriss, J., Samba, A., Dooze, P. 2017. CogNitive 5G networks: Comprehensive operator use cases with machine learning for management operations. En *Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks, ICIN 2017*, April 13, 2017, Institute of Electrical and Electronics Engineers Inc., pp. 252–259. doi:10.1109/ICIN.2017.7899421
- Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc.: New York, NYUnited States.

- Bui, N., Cesana, M., Hosseini, S. A., Liao, Q., Malanchini, I., Widmer, J. 2017, July 1. A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques. *IEEE Communications Surveys and Tutorials*, Vol. 19. doi:10.1109/COMST.2017.2694140
- Chai, T., Draxler, R. R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. doi:10.5194/GMD-7-1247-2014
- Cochrane, C. 2018. Time Series Nested Cross-Validation. Retrieved October 22, 2021, from Towards Data Science website: <https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9>
- Delgado, D. 2020. Evaluación de redes móviles 5G en entornos con aplicaciones IoT (Universitat Oberta de Catalunya. Departamento de Telemática). Retrieved September 27, 2021, from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/106667/6/daviddTFM0120memoria.pdf>
- Du, S., Li, T., Yang, Y., Horng, S. J. 2020. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388, 269–279. doi:10.1016/J.NEUCOM.2019.12.118
- ETSI. 2019. ETSI TS 122 261 - V15.8.0; Service requirements for next generation new services and markets (3GPP TS 22.261 version 15.8.0 Release 15). Retrieved June 24, 2021, from ETSI TS 122 261 website: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>
- ETSI. 2021a. ETSI TS 122 261 - V16.14.0 - 5G; Service requirements for the 5G system (3GPP TS 22.261 version 16.14.0 Release 16). Retrieved June 24, 2021, from ETSI TS 122 261 - V16.14.0 website: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>
- ETSI. 2021b. ETSI TS 128 554 - V16.8.0- 5G; Management and orchestration; 5G end to end Key Performance Indicators (KPI) (3GPP TS 28.554 version 16.8.0 Release 16). Retrieved September 27, 2021, from 3GPP TS 28.554 V17.4.0 website: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>
- ETSI. 2021c. ETSI TS 138 306 - V16.5.0 - 5G; NR; User Equipment (UE) radio access capabilities (3GPP TS 38.306 version 16.5.0 Release 16). Retrieved September 30, 2021, from ETSI TS 138 306 website: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>
- Eyceyurt, E., Zec, J. 2020. Uplink Throughput Prediction in Cellular Mobile Networks. *International Journal of Electronics and Communication Engineering*, 14(6), 149–153. from https://www.researchgate.net/profile/Engin-Eyceyurt/publication/341984677_Uplink_Throughput_Prediction_in_Cellular_Mobile_Networks/links/5ee07f43a6fdcc47689442bc/Uplink-Throughput-Prediction-in-Cellular-Mobile-Networks.pdf
- Fernández, A. 2020. Uso de Machine-Learning en el control de congestión sobre redes 5G (Universidad de Cantabria). Retrieved August 13, 2021, from <https://repositorio.unican.es/xmlui/handle/10902/19021>

- Freire, E., Silva, S. 2019. Redes neuronales. Programa de Visión. Retrieved August 16, 2021, from by Bootcamp AI website: <https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb>
- Gadaleta, M., Rossi, M. 2018. IDNet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recognition*, 74, 25–37. doi:10.1016/J.PATCOG.2017.09.005
- Gaona, A. E., Ballesteros, D. M. 2012. An efficient selection of neural-network architectures using pruning and regularization techniques. *Tecnura*, 16(33), 158–172. from http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-921X2012000300012&lng=en&nrm=iso&tlng=es
- García, M. 2007. Propuesta de acuerdos de nivel de servicio en la red pública de datos de ETECSA (Universidad Central “Marta Abreu” de Las Villas. Facultad de Ingeniería Eléctrica. Departamento de Electrónica y Telecomunicaciones). Retrieved March 18, 2021, from <http://dspace.uclv.edu.cu:8089/xmlui/handle/123456789/6984>
- Géron, A. 2019. Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems. In N. Tache (Ed.), ISBN: 9781492032649 (2nd Edition). O’Reilly Media, Inc.: Sebastopol, California, USA. pp. 277–364.
- Gers, F. A., Schmidhuber, J., Cummins, F. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10), 2451–2471. doi:10.1162/089976600300015015
- Gildardo, M. A. 2006. Predicción de tráfico en redes de telecomunicaciones basado en técnicas de Inteligencia Analítica (CINVESTAV). Retrieved August 16, 2021, from <https://www.cs.cinvestav.mx/TesisGraduados/2006/tesisGildardoMillan.pdf>
- Gong, G., An, X., Mahato, N. K., Sun, S., Chen, S., Wen, Y. 2019. Research on Short-Term Load Prediction Based on Seq2seq Model. *Energies* 2019, Vol. 12, Page 3199, 12(16), 3199. doi:10.3390/EN12163199
- Goyal, P., Lobiyal, D. K., Katti, C. P. 2017. Vertical handoff in heterogeneous wireless networks: A tutorial. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, 2017-January, 551–556. doi:10.1109/CCAA.2017.8229862
- Gramaglia, M., Banchs, A., Pastor, A. U., Sciancalepore, V., Yousaf, Z., Mannweiler, C., Yu, L., Sayadi, B., Alberi-Morel, M.-L., Gajic, B. 2017, June 30. Definition and specification of connectivity and QoE/QoS management mechanisms. Final Report. Retrieved August 6, 2020, from http://www.it.uc3m.es/wnl/5gnorma/pdf/5g_norma_d5-2.pdf
- Gramaglia, M., Digon, I., Friderikos, V., Von Hugo, D., Mannweiler, C., Puente, M. A., Samdanis, K., Sayadi, B. 2016. Flexible connectivity and QoE/QoS management for 5G Networks: The 5G NORMA view. *2016 IEEE International Conference on Communications Workshops, ICC 2016*, 373–379. doi:10.1109/ICCW.2016.7503816
- Graves, A., Jaitly, N. 2014, June 18. Towards End-To-End Speech Recognition with Recurrent Neural Networks. Retrieved September 16, 2021, from Proceedings of the 31st International Conference on

Machine Learning, PMLR website: <https://proceedings.mlr.press/v32/graves14.html>

- Grover, P. 2018. 5 Regression Loss Functions All Machine Learners Should Know. Retrieved October 19, 2021, from Heartbeat website: <https://heartbeat.comet.ml/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>
- GSMA. 2018. Road to 5G: Introduction and Migration Whitepaper - Future Networks. Retrieved September 21, 2021, from <https://www.gsma.com/futurenetworks/resources/road-to-5g-introduction-and-migration-whitepaper/>
- Gutz, S., Story, A., Schlesinger, C., Foster, N. 2012. Splendid isolation: A slice abstraction for software-defined networks. HotSDN'12 - Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, 79–84. doi:10.1145/2342441.2342458
- Haddad, R., Viniotis, Y. 2007. 3-Tier service level agreement with automatic class upgrades. En GLOBECOM - IEEE Global Telecommunications Conference, Raleigh, NC, December 2007. doi:10.1109/GLOCOMW.2007.4437778
- Hassan, A., Mahmood, A. 2018. Convolutional Recurrent Deep Learning Model for Sentence Classification. IEEE Access, 6, 13949–13957. doi:10.1109/ACCESS.2018.2814818
- Haykin, S. 2001. Feedforward neural network: An Introduction. Retrieved August 15, 2021, from Wiley website: <https://catalogimages.wiley.com/images/db/pdf/0471349119.01.pdf>
- He, Q., Dovrolis, C., Ammar, M. 2005. On the predictability of large transfer TCP throughput. Computer Communication Review, 35(4), 145–156. doi:10.1145/1090191.1080110
- Hochreiter, S., Schmidhuber, J. 1997. Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. doi:10.1162/NECO.1997.9.8.1735
- ITU-R. 2015. Recommendation ITU-R M.2083-0 IMT Vision-Framework and overall objectives of the future development of IMT for 2020 and beyond M Series Mobile, radiodetermination, amateur and related satellite services. Retrieved August 6, 2020, from <http://www.itu.int/ITU-R/go/patents/en>
- ITU. 2018. Sentando las bases para la 5G: Oportunidades y desafíos. Retrieved June 19, 2021, from https://www.itu.int/dms_pub/itu-d/opb/pref/D-PREF-BB.5G_01-2018-PDF-S.pdf
- Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K. C., Hanzo, L. 2017. Machine Learning Paradigms for Next-Generation Wireless Networks. IEEE Wireless Communications, 24(2), 98–105. doi:10.1109/MWC.2016.1500356WC
- Kapassa, E., Touloupou, M., Kyriazis, Di. 2018. SLAs in 5G: A complete framework facilitating VNF- and NS-tailored SLAs management. En Proceedings - 32nd IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2018, July 20, 2018, Institute of Electrical and Electronics Engineers Inc., 2018-January, pp. 469–474. doi:10.1109/WAINA.2018.00130

- Kapassa, E., Touloupou, M., Stavrianos, P., Xylouris, G., Kyriazis, D. 2019. Managing and optimizing quality of service in 5G environments across the complete SLA lifecycle. *Advances in Science, Technology and Engineering Systems*, 4(1), 329–342. doi:10.25046/aj040132
- Kene, P., Haridas, S. L. 2020. Reducing ping-pong effect in heterogeneous wireless networks using machine learning. *Advances in Intelligent Systems and Computing*, 989, 697–705. doi:10.1007/978-981-13-8618-3_71
- Kevin Fogarty. 2019. Issues In Designing 5G Beamforming Antennas. Retrieved August 3, 2021, from <https://semiengineering.com/5g-beamforming-antennas-create-design-test-problems/>
- Keysight. 2021. Engineering the 5G World | Keysight. Retrieved June 29, 2021, from Keysight Technologies website: <https://www.keysight.com/mx/en/cmp/2020/engineering-the-5g-world.html>
- Klautau, A., Batista, P., Gonzalez-Prelcic, N., Wang, Y., Heath, R. W. 2018. 5G MIMO data for machine learning: Application to beam-selection using deep learning. En 2018 Information Theory and Applications Workshop, ITA 2018, October 23, 2018, Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ITA.2018.8503086
- Kostadinov, S. 2019. Understanding Encoder-Decoder Sequence to Sequence Model. Retrieved October 23, 2021, from Towards Data Science website: <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346>
- Kousias, K., Pappas, A., Alay, O., Argyriou, A., Riegler, M. 2020, November 20. Long Short Term Memory Networks for Bandwidth Forecasting in Mobile Broadband Networks under Mobility. Retrieved March 18, 2021, from arXiv website: <http://arxiv.org/abs/2011.10563>
- Krukrubo, L. A. 2021. Scaling vs. Normalizing Data – Towards AI. Retrieved October 16, 2021, from Towards AI website: <https://towardsai.net/p/data-science/scaling-vs-normalizing-data-5c3514887a84>
- Kumar, S. 2020. Understanding 8 types of Cross-Validation. Retrieved October 22, 2021, from Towards Data Science website: <https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d>
- Kurtz, F., Bektas, C., Dorsch, N., Wietfeld, C. 2018. Network Slicing for Critical Communications in Shared 5G Infrastructures - An Empirical Evaluation. En 2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, September 10, 2018, Institute of Electrical and Electronics Engineers Inc., pp. 262–266. doi:10.1109/NETSOFT.2018.8460110
- Lane, T. 2018. 1D & 3D Convolutions explained with... MS Excel! | by Thom Lane | Apache MXNet | Medium. Retrieved September 20, 2021, from <https://medium.com/apache-mxnet/1d-3d-convolutions-explained-with-ms-excel-5f88c0f35941>
- Li, J., Zhao, Z., Li, R. 2018. Machine learning-based IDS for softwaredefined 5G network. *IET Networks*, 7(2), 53–60. doi:10.1049/iet-net.2017.0212

- Li, T., Zhao, M., Wong, K. K. L. 2020. Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks. *Computer Communications*, 152, 109–118. doi:10.1016/j.comcom.2020.01.034
- Liu, G., Huang, Y., Chen, Z., Liu, L., Wang, Q., Li, N. 2020. 5G Deployment: Standalone vs. Non-Standalone from the Operator Perspective. *IEEE Communications Magazine*, 58(11), 83–89. doi:10.1109/MCOM.001.2000230
- Liu, Y., Lee, J. Y. B. 2015. An Empirical Study of Throughput Prediction in Mobile Data Networks. 2015 IEEE Global Communications Conference (GLOBECOM), 1–6. doi:10.1109/GLOCOM.2015.7417858
- Lu, W., Li, J., Li, Y., Sun, A., Wang, J. 2020. A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020. doi:10.1155/2020/6622927
- MathWorks, I. 2021. Introduction to Deep Learning: What Are Convolutional Neural Networks? Retrieved September 17, 2021, from MathWorks, Videos and Webinars website: <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>
- Mei, L., Gou, J., Cai, Y., Cao, H., Liu, Y. 2021, April 27. Realtime Mobile Bandwidth and Handoff Predictions in 4G/5G Networks. Retrieved November 30, 2021, from <https://arxiv.org/abs/2104.12959>
- Mirza, M., Sommers, J., Barford, P., Zhu, X. 2007. A machine learning approach to TCP throughput prediction. *IEEE Xplore*, 97–102. doi:10.1145/1254882.1254894
- Mirza, M., Sommers, J., Barford, P., Zhu, X. 2010. A machine learning approach to TCP throughput prediction. *IEEE/ACM Transactions on Networking*, 18(4), 1026–1039. doi:10.1109/TNET.2009.2037812
- Morocho-Cayamcela, M. E., Lee, H., Lim, W. 2019. Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. *IEEE Access*, 7, 137184–137206. doi:10.1109/ACCESS.2019.2942390
- Mullins, R., Taynnan, M., et. al. 2017, March 9. Cognitive Network Management for 5G. Retrieved August 6, 2020, from https://5g-ppp.eu/wp-content/uploads/2017/03/NetworkManagement_WhitePaper_1.pdf
- Napolitano, A., Giorgetti, A., Kondepu, K., Valcarengi, L., Castoldi, P. 2018. Network Slicing: An Overview. *IEEE 4th International Forum on Research and Technologies for Society and Industry, RTSI 2018 - Proceedings*. doi:10.1109/RTSI.2018.8548449
- Narayanan, A., Ramadan, E., Carpenter, J., Liu, Q., Liu, Y., Qian, F., Zhang, Z.-L. 2020. A First Look at Commercial 5G Performance on Smartphones. *En Proceedings of The Web Conference 2020*, New York, NY, USA, 2020, ACM, 12, pp. 894–905. from <https://doi.org/10.1145/3366423.3380169>
- Narayanan, A., Ramadan, E., Mehta, R., Hu, X., Liu, Q., Fezeu, R. A. K., Dayalan, U. K., Verma, S., Ji, P., Li,

- T., Qian, F., Zhang, Z.-L. 2020. Lumos5G: Mapping and Predicting Commercial mmWave 5G Throughput. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 18(20)*, 176–193. doi:10.1145/3419394.3423629
- NGMN Alliance. 2016. Recommendations for NGMN KPIs and Requirements for 5G by NGMN Alliance. Retrieved June 25, 2021, from https://www.ngmn.org/wp-content/uploads/Publications/2016/160603_Annex_NGMN_Liaison_to_3GPP_RAN_72_v1_0.pdf
- NGMN Alliance, Hedman (Ed.), P. 2016, January 13. Description of Network Slicing Concept by NGMN Alliance. Retrieved August 11, 2021, from https://ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf
- Nitsche, T., Flores, A. B., Knightly, E. W., Widmer, J. 2015. Steering with eyes closed: Mm-Wave beam steering without in-band measurement. *Proceedings - IEEE INFOCOM, 26*, 2416–2424. doi:10.1109/INFOCOM.2015.7218630
- Nyangaresi, V. O., Rodrigues, A. J., Abeka, S. O. 2020. Secure Handover Protocol for High Speed 5G Networks. *International Journal of Advanced Networking and Applications, 11(06)*, 4443–4450. doi:10.35444/IJANA.2020.11061
- Olah, C. 2015. Understanding LSTM Networks -- colah's blog. Retrieved September 11, 2021, from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Ookla5GMap. 2021. Ookla 5G Map - Tracking 5G Network Rollouts Around the World. Retrieved June 29, 2021, from Ookla5GMap. website: <https://www.speedtest.net/ookla-5g-map>
- Papageorgiou, A., Fernandez-Fernandez, A., Ochoa-Aday, L., Pelaez, M. S., Shuaib Siddiqui, M. 2020. SLA management procedures in 5G slicing-based systems. En *2020 European Conference on Networks and Communications, EuCNC 2020, June 1, 2020, Institute of Electrical and Electronics Engineers Inc.*, pp. 7–11. doi:10.1109/EuCNC48522.2020.9200904
- Parada, C., Bonnet, J., Fotopoulou, E., Zafeiropoulos, A., Kapassa, E., Touloupou, M., Kyriazis, D., Vilalta, R., Munoz, R., Casellas, R., Martinez, R., Xilouris, G. 2018. 5Gtango: A Beyond-Mano Service Platform. En *2018 European Conference on Networks and Communications, EuCNC 2018, August 20, 2018, Institute of Electrical and Electronics Engineers Inc.*, pp. 26–30. doi:10.1109/EuCNC.2018.8443232
- Pascanu, R., Mikolov, T., Bengio, Y. 2013, May 26. On the difficulty of training recurrent neural networks. Retrieved September 11, 2021, from <https://proceedings.mlr.press/v28/pascanu13.html>
- Peyré, G. 2019. Mathematics of Neural Networks. Retrieved August 15, 2021, from <https://mathematical-tours.github.io/book-basics-sources/neural-networks-en/NeuralNetworksEN.pdf>
- Phemina Selvi, M., Sendhilmathan, S. 2017. Minimizing handover delay and maximizing throughput by Heterogeneous Handover Algorithm (HHA) in telecommunication networks. *Applied Mathematics and Information Sciences, 11(6)*, 1737–1746. doi:10.18576/AMIS/110621

- Popovski, P., Trillingsgaard, K. F., Simeone, O., Durisi, G. 2018. 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. *IEEE Access*, 6, 55765–55779. doi:10.1109/ACCESS.2018.2872781
- Raca, D., Leahy, D., Sreenan, C. J., Quinlan, J. J. 2020. Beyond throughput, the next generation: A 5G dataset with channel and context metrics. En *MMSys 2020 - Proceedings of the 2020 Multimedia Systems Conference*, May 27, 2020, Association for Computing Machinery, Inc, pp. 303–308. doi:10.1145/3339825.3394938
- Raca, D., Quinlan, J. J., Zahran, A. H., Sreenan, C. J. 2018. Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics. En *Proceedings of the 9th ACM Multimedia Systems Conference*, New York, NY, USA, 2018, ACM, 18. from <https://doi.org/10.1145/3204949.3208123>
- Raca, D., Zahran, A. H., Sreenan, C. J., Sinha, R. K., Halepovic, E., Jana, R., Gopalakrishnan, V. 2017. Back to the future: Throughput prediction for cellular networks using radio KPIs. *HotWireless '17 Proceedings of the 4th ACM Workshop on Hot Topics in Wireless*, 37–41. doi:10.1145/3127882.3127892
- Raca, D., Zahran, A. H., Sreenan, C. J., Sinha, R. K., Halepovic, E., Jana, R., Gopalakrishnan, V. 2020. On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and Challenges. *IEEE Communications Magazine*, 58(3), 11–17. doi:10.1109/MCOM.001.1900394
- Roman, V. 2020, February 27. Convolutional Neural Networks. Introduction & Convolutions. Retrieved September 21, 2021, from *Towards Data Science* website: <https://towardsdatascience.com/convolutional-neural-networks-357b9b2d75bd>
- Sama, M. R., Beker, S., Kiess, W., Thakolsri, S. 2016. Service-based slice selection function for 5G. En 2016 *IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*, Washington, DC, USA, December 4, 2016, Institute of Electrical and Electronics Engineers Inc. doi:10.1109/GLOCOM.2016.7842265
- Samba, A., Busnel, Y., Blanc, A., Dooze, P., Simon, G. 2016. Throughput Prediction in Cellular Networks: Experiments and Preliminary Results. Retrieved October 11, 2021, from HAL website: <https://hal.archives-ouvertes.fr/hal-01311158v2>
- Samba, A., Busnel, Y., Blanc, A., Dooze, P., Simon, G. 2017. Instantaneous throughput prediction in cellular networks: Which information is needed? *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, 624–627. doi:10.23919/INM.2017.7987345
- Samsung, E. C. L. 2021. 5G Standalone Architecture. Retrieved September 29, 2021, from https://images.samsung.com/is/content/samsung/assets/global/business/networks/insights/white-papers/0107_5g-standalone-architecture/5G_SA_Architecture_Technical_White_Paper_Public.pdf
- Santos, G. L., Endo, P. T., Sadok, D., Kelner, J. 2020. When 5G Meets Deep Learning: A Systematic Review. *Algorithms*, 13(9), 208. doi:10.3390/a13090208

- Shah, T. 2017. About Train, Validation and Test Sets in Machine Learning. Retrieved October 16, 2021, from Towards Data Science website: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>
- Sim, G. H., Klos, S., Asadi, A., Klein, A., Hollick, M. 2018. An online context-aware machine learning algorithm for 5g mmwave vehicular communications. *IEEE/ACM Transactions on Networking*, 26(6), 2487–2500. doi:10.1109/TNET.2018.2869244
- Sohaib, R. M., Onireti, O., Sambo, Y., Imran, M. A. 2021. Network Slicing for Beyond 5G Systems: An Overview of the Smart Port Use Case. *Electronics* 2021, Vol. 10, Page 1090, 10(9), 1090. doi:10.3390/ELECTRONICS10091090
- Song, G., Wang, W., Chen, D., Jiang, T. 2018. KPI/KQI-driven coordinated multipoint in 5G: Measurements, field trials, and technical solutions. *IEEE Wireless Communications*, 25(5), 23–29. doi:10.1109/MWC.2018.1800041
- Tayyaba, S. K., Khattak, H. A., Almogren, A., Shah, M. A., Ud Din, I., Alkhalifa, I., Guizani, M. 2020. 5G vehicular network resource management for improving radio access through machine learning. *IEEE Access*, 8, 6792–6800. doi:10.1109/ACCESS.2020.2964697
- TensorFlow Team. 2017. Introduction to TensorFlow Datasets and Estimators. Retrieved October 31, 2021, from Google Developers Blog website: <https://developers.googleblog.com/2017/09/introducing-tensorflow-datasets.html>
- TensorFlow Team. 2019. Time series forecasting. Retrieved October 17, 2021, from TensorFlow.org website: https://www.tensorflow.org/tutorials/structured_data/time_series#multi-output_models
- Tezergil, B., Onur, E. 2021, March 15. Wireless Backhaul in 5G and Beyond: Issues, Challenges and Opportunities. Retrieved October 1, 2021, from <https://arxiv.org/abs/2103.08234v2>
- Touloupou, M., Kapassa, E., Symvoulidis, C., Stavrianos, P., Kyriazis, D. 2019. An Integrated SLA Management Framework in a 5G Environment. *En Proceedings of the 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2019, April 9, 2019, Institute of Electrical and Electronics Engineers Inc.*, pp. 233–235. doi:10.1109/ICIN.2019.8685916
- Trehan, D. 2020. Gradient Descent Explained. A comprehensive guide to Gradient. Retrieved October 20, 2021, from Towards Data Science website: <https://towardsdatascience.com/gradient-descent-explained-9b953fc0d2c>
- Trinh, H. D. 2020. Data analytics for mobile traffic in 5G networks using machine learning techniques (Universitat Politècnica de Catalunya. Department of Network Engineering). Retrieved August 13, 2021, from <http://www.tdx.cat/handle/10803/669204>
- Vadiraja, P., Chattha, M. A. 2020, August 13. A Survey on Knowledge integration techniques with Artificial Neural Networks for seq-2-seq/time series models. Retrieved October 24, 2021, from <https://arxiv.org/abs/2008.05972v1>

- Van Der Meer, S., Keeney, J., Fallon, L. 2018. 5G networks must be autonomic! En IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018, July 6, 2018, Institute of Electrical and Electronics Engineers Inc., pp. 1–5. doi:10.1109/NOMS.2018.8406185
- Van Rossem, S., Peuster, M., Conceição, L., Kouchaksaraei, H. R., Tavernier, W., Colle, D., Pickavet, M., Demeester, P. 2017. A network service development kit supporting the end-to-end lifecycle of NFV-based telecom services. En 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2017, December 7, 2017, Institute of Electrical and Electronics Engineers Inc., 2017-January, pp. 1–2. doi:10.1109/NFV-SDN.2017.8169859
- Verma, S. 2019. Understanding 1D and 3D Convolution Neural Network. Retrieved September 18, 2021, from Towards Data Science website: <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>
- Vinayakumar, R., Soman, K. P., Poornachandran, P. 2017. Applying deep learning approaches for network traffic prediction. 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-January, 2353–2358. doi:10.1109/ICACCI.2017.8126198
- Vryniotis Vasilis. 2013. Tuning the learning rate in Gradient Descent. Retrieved October 20, 2021, from DatumBox website: <http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/>
- Vyakaranam, N., Krishna, D. 2018. 5G: Network As A Service - How 5G enables the telecom operators to lease out their network | NETMANIAS. Retrieved August 6, 2020, from <https://netmanias.com/en/post/blog/13311/5g/5g-network-as-a-service-how-5g-enables-the-telecom-operators-to-lease-out-their-network>
- Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X., Yang, D. 2017. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. Proceedings - IEEE INFOCOM. doi:10.1109/INFOCOM.2017.8057090
- Wei, B., Kawakami, W., Kanai, K., Katto, J., Wang, S. 2018. TRUST: A TCP Throughput Prediction Method in Mobile Networks. 2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings. doi:10.1109/GLOCOM.2018.8647390
- Willmott, C. J., Matsuura, K. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82. doi:10.3354/CR030079
- Wu, K., Wu, J., Feng, L., Yang, B., Liang, R., Yang, S., Zhao, R. 2021. An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system. *International Transactions on Electrical Energy Systems*, 31(1). doi:10.1002/2050-7038.12637
- Zakeri, A., Gholipoor, N., Tajallifar, M., Ebrahimi, S., Reza Javan, M., Member, S., Mokari, N., Reza Sharafat, A. 2020, February 20. E2E Migration Strategies Towards 5G: Long-term Migration Plan and Evolution Roadmap. Retrieved September 21, 2021, from <https://arxiv.org/abs/2002.08984v1>

- Zaremba, W., Sutskever, I. 2014, October 17. Learning to Execute. Retrieved September 16, 2021, from Neural and Evolutionary Computing website: <https://arxiv.org/abs/1410.4615v3>
- Zenalden, F., Hassan, S., Habbal, A. 2017. Vertical Handover in Wireless Heterogeneous Networks. Retrieved August 4, 2021, from https://www.researchgate.net/publication/320945141_Vertical_Handover_in_Wireless_Heterogeneous_Networks
- Zhang, C., Patras, P. 2017. Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks. Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 231–240. from <https://arxiv.org/abs/1712.08083v1>

Anexos

Anexo A. Detalles de los modelos de ML propuestos.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 120, 16)]	0	[]
lstm (LSTM)	[(None, 32), (None, 32), (None, 32)]	6272	['input_1[0][0]']
repeat_vector (RepeatVector)	(None, 30, 32)	0	['lstm[0][0]']
lstm_1 (LSTM)	(None, 30, 32)	8320	['repeat_vector[0][0]', 'lstm[0][1]', 'lstm[0][2]']
time_distributed (TimeDistributed)	(None, 30, 16)	528	['lstm_1[0][0]']

Total params: 15,120
 Trainable params: 15,120
 Non-trainable params: 0

Figura 54. Resumen del modelo RNN con arquitectura *Encoder-Decoder* de una capa (RNN-E1D1) (elaboración propia).

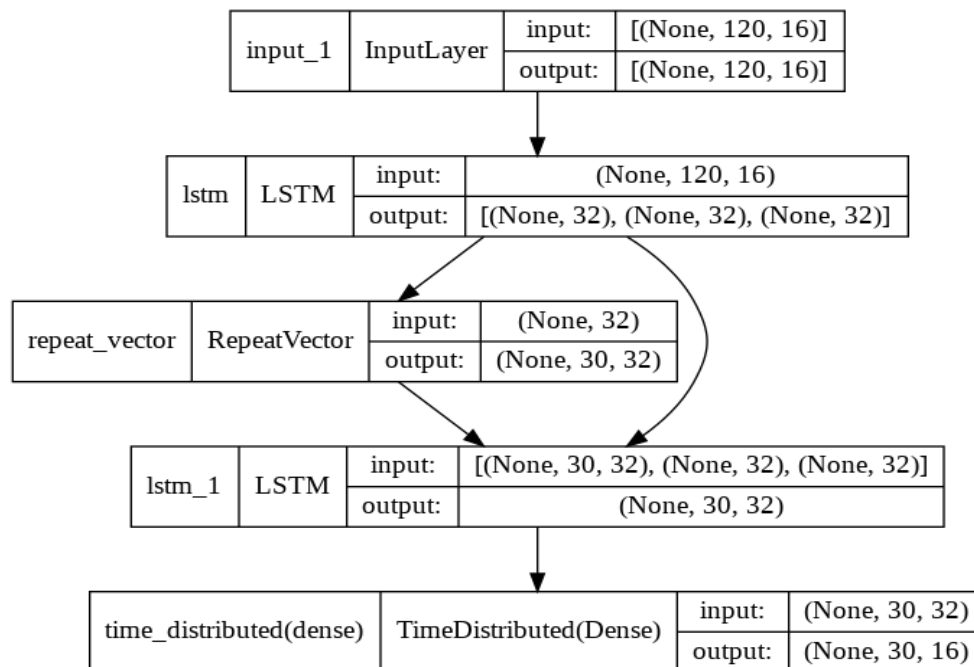


Figura 55. Detalles de las capas de la RNN con arquitectura *Encoder-Decoder* de una capa (RNN-E1D1) (elaboración propia).

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 120, 16)]	0	[]
lstm (LSTM)	[(None, 120, 16), (None, 16), (None, 16)]	2112	['input_1[0][0]']
lstm_1 (LSTM)	[(None, 16), (None, 16), (None, 16)]	2112	['lstm[0][0]']
repeat_vector (RepeatVector)	(None, 30, 16)	0	['lstm_1[0][0]']
lstm_2 (LSTM)	(None, 30, 16)	2112	['repeat_vector[0][0]', 'lstm[0][1]', 'lstm[0][2]']
lstm_3 (LSTM)	(None, 30, 16)	2112	['lstm_2[0][0]', 'lstm_1[0][1]', 'lstm_1[0][2]']
time_distributed (TimeDistributed)	(None, 30, 16)	272	['lstm_3[0][0]']

Total params: 8,720
Trainable params: 8,720
Non-trainable params: 0

Figura 56. Resumen del modelo RNN con arquitectura *Encoder-Decoder* de dos capas (RNN-E2D2) (elaboración propia).

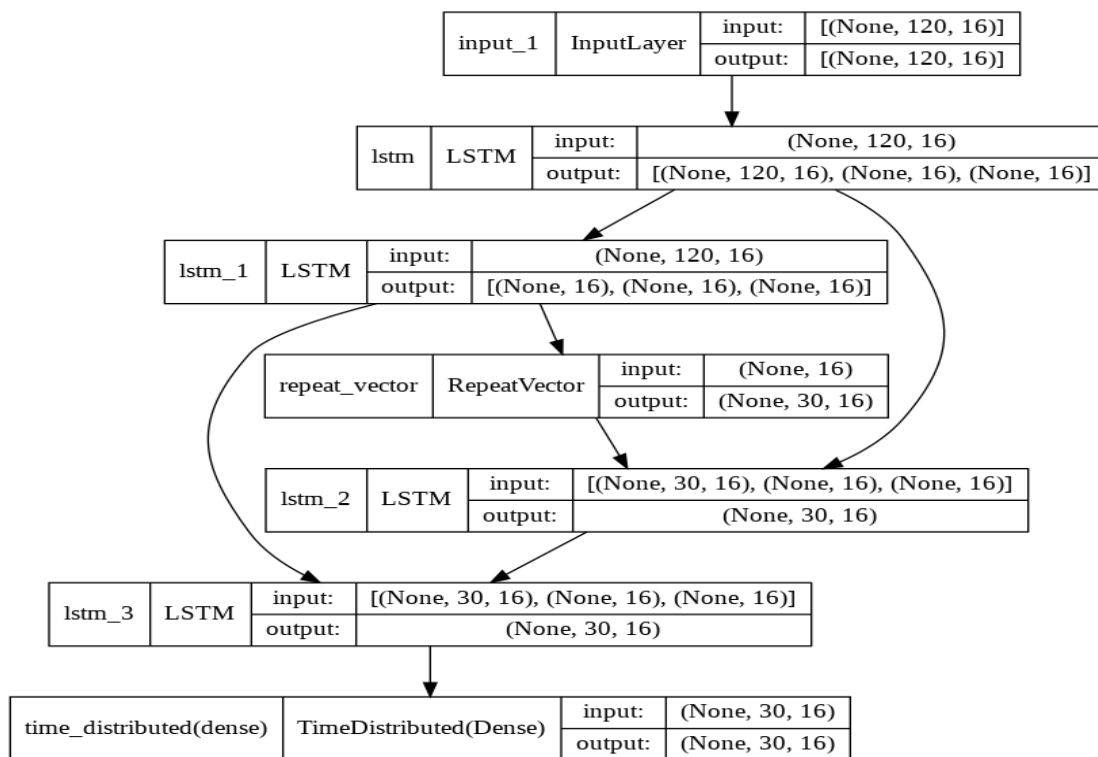


Figura 57. Detalles de las capas de la RNN con arquitectura *Encoder-Decoder* de dos capas (RNN-E2D2) (elaboración propia).

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 120, 16)]	0	[]
lstm (LSTM)	[(None, 120, 16), (None, 16), (None, 16)]	2112	['input_1[0][0]']
lstm_1 (LSTM)	[(None, 120, 16), (None, 16), (None, 16)]	2112	['lstm[0][0]']
lstm_2 (LSTM)	[(None, 16), (None, 16), (None, 16)]	2112	['lstm_1[0][0]']
repeat_vector (RepeatVector)	(None, 30, 16)	0	['lstm_2[0][0]']
lstm_3 (LSTM)	(None, 30, 16)	2112	['repeat_vector[0][0]', 'lstm[0][1]', 'lstm[0][2]']
lstm_4 (LSTM)	(None, 30, 16)	2112	['lstm_3[0][0]', 'lstm_1[0][1]', 'lstm_1[0][2]']
lstm_5 (LSTM)	(None, 30, 16)	2112	['lstm_4[0][0]', 'lstm_2[0][1]', 'lstm_2[0][2]']
time_distributed (TimeDistributed)	(None, 30, 16)	272	['lstm_5[0][0]']

Total params: 12,944
Trainable params: 12,944
Non-trainable params: 0

Figura 58. Resumen del modelo RNN con arquitectura *Encoder-Decoder* de tres capas (RNN-E3D3) (elaboración propia).

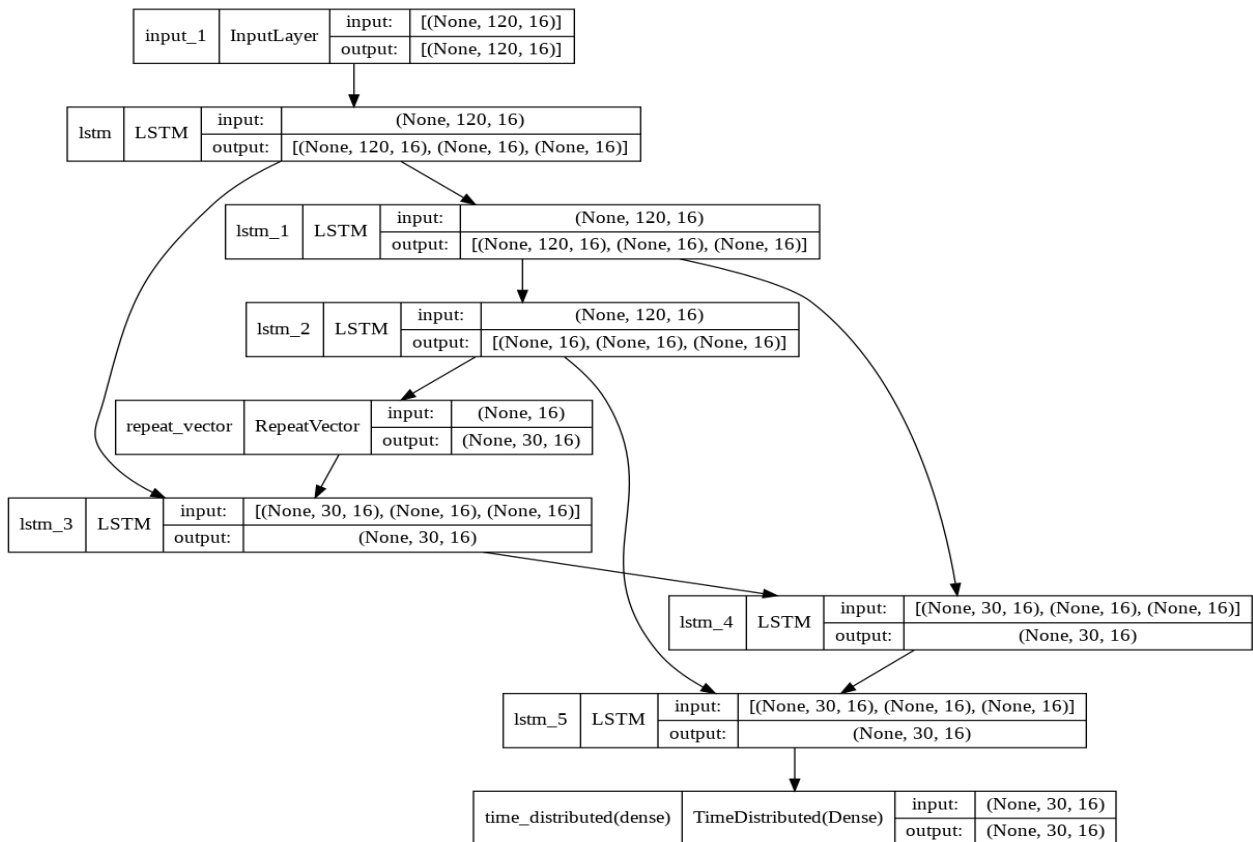


Figura 59. Detalles de las capas de la RNN con arquitectura *Encoder-Decoder* de tres capas (RNN-E3D3) (elaboración propia).

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 120, 512)	25088
max_pooling1d (MaxPooling1D)	(None, 60, 512)	0
lstm (LSTM)	(None, 60, 128)	328192
lstm_1 (LSTM)	(None, 60, 128)	131584
lstm_2 (LSTM)	(None, 60, 128)	131584
lstm_3 (LSTM)	(None, 60, 128)	131584
lstm_4 (LSTM)	(None, 60, 128)	131584
lstm_5 (LSTM)	(None, 60, 128)	131584
lstm_6 (LSTM)	(None, 60, 128)	131584
lstm_7 (LSTM)	(None, 60, 128)	131584
bidirectional (Bidirectional)	(None, 60, 64)	41216
bidirectional_1 (Bidirectional)	(None, 60, 64)	24832
bidirectional_2 (Bidirectional)	(None, 60, 64)	24832
bidirectional_3 (Bidirectional)	(None, 60, 64)	24832
bidirectional_4 (Bidirectional)	(None, 64)	24832
dense_1 (Dense)	(None, 480)	31200
lambda_1 (Lambda)	(None, 480)	0
reshape_1 (Reshape)	(None, 30, 16)	0
Total params: 1,446,112		
Trainable params: 1,446,112		
Non-trainable params: 0		

Figura 60. Resumen del modelo de red mixta CNN-RNN (elaboración propia).

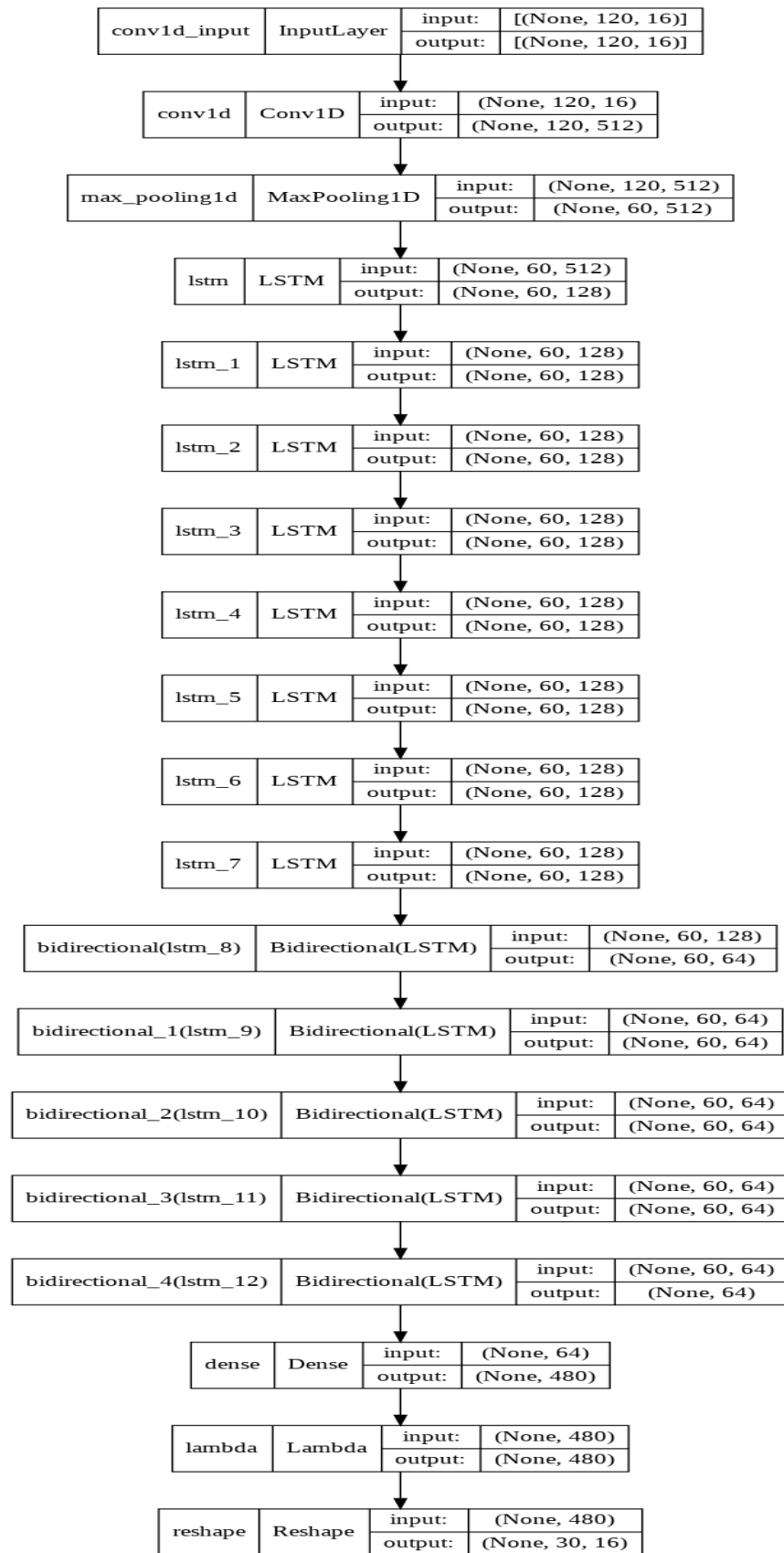


Figura 61. Detalles de las capas de la red mixta CNN-RNN (elaboración propia).

Anexo B. Análisis combinado a partir de la predicción del *throughput* y del VHO.

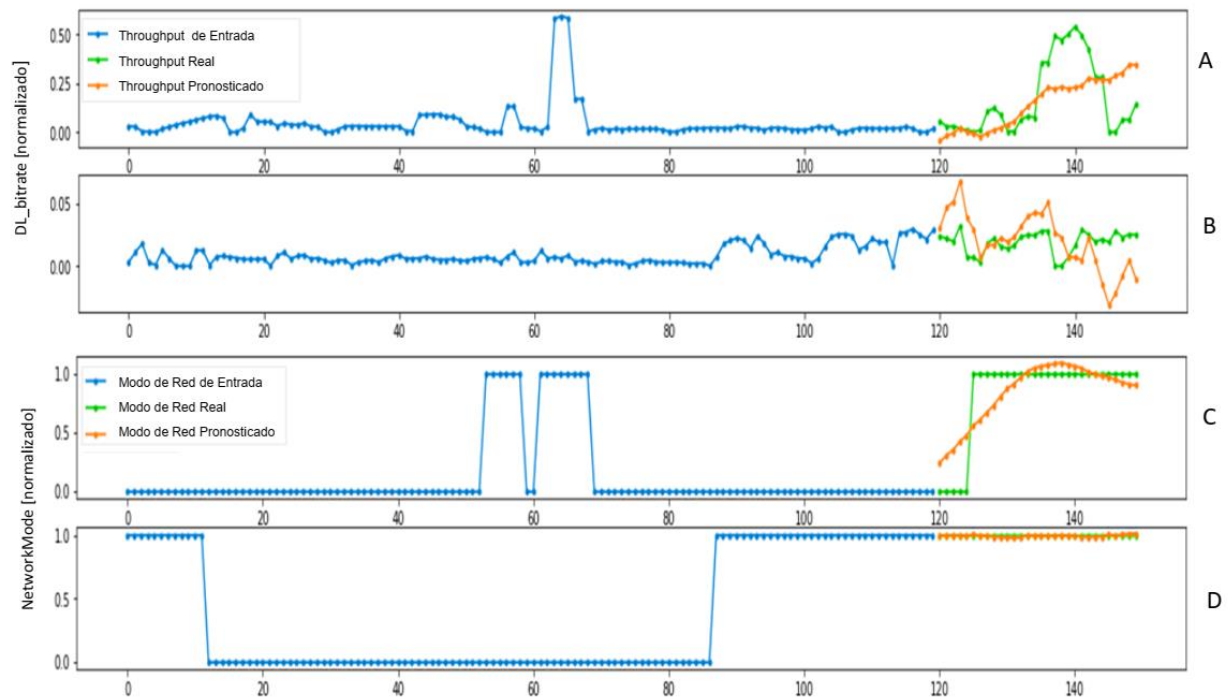


Figura 62. Predicción del *throughput* y del VHO para el análisis combinado. Ejemplo del aumento brusco del *throughput* debido a un VHO 4G→5G (combinación de A con C). Ejemplo de fluctuación del *throughput* sin ocurrencia de VHO (combinación de B con D) (elaboración propia).

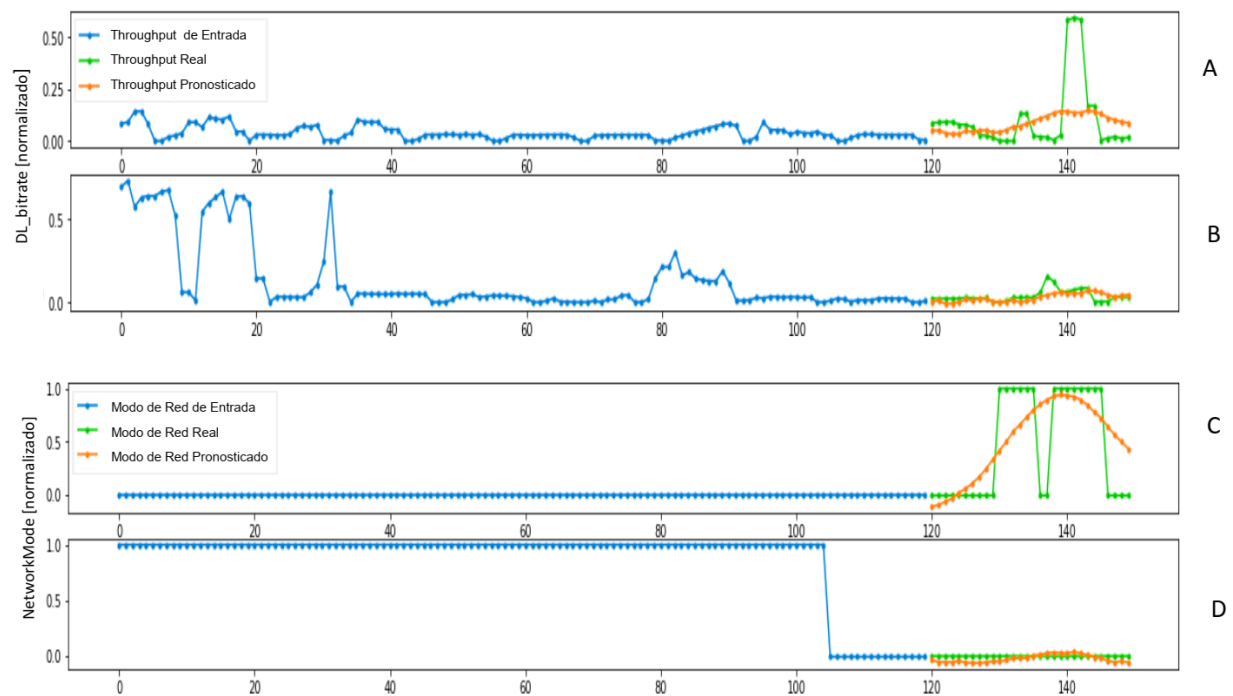


Figura 63. Predicción del *throughput* y del VHO para el análisis combinado. Ejemplo del aumento y disminución brusca del *throughput* debido a un VHO del tipo *ping pong* (combinación de A con C). Ejemplo de fluctuación del *throughput* sin ocurrencia de VHO (combinación de B con D) (elaboración propia).