

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER:

**CUANTIFICACIÓN Y PREDICCIÓN DE LLUVIA A PARTIR DE
INFORMACIÓN OBTENIDA DE RADIOENLACES DE
MICROONDAS Y COMUNICACIONES SATELITALES APLICANDO
MACHINE LEARNING**

Titulación: Máster Universitario en Ingeniería en Tecnologías de la Telecomunicación

Autor: D. Gabriel Ojeda Suárez

Tutor: Dr. Juan Luis Navarro Mesa

Tutor: Dr. Antonio Gabriel Ravelo García

Fecha: Julio de 2023

Agradecimientos

Por un lado, me gustaría agradecer a mis dos tutores del Trabajo Fin de Máster, al Dr. Juan Luis Narro Mesa y al Dr. Antonio Gabriel Ravelo García, por guiarme a lo largo de este proyecto y ayudarme a afrontar este reto de la manera más profesional posible. Gracias por acompañarme y transmitirme calma y seguridad en todo momento. Gracias por exigirme mi tres mil por ciento, por hacer míos sus conocimientos y por inculcarme su voluntad para la búsqueda insaciable de mejorar, curiosear y plantearme nuevos retos. Por otro lado, dado a que aquí se acaba mi camino por el Máster Universitario en Ingeniería de Telecomunicación, quiero agradecer a todos aquellos profesores que han dedicado su tiempo y esfuerzo en hacerme crecer como ingeniero cumpliendo su labor de tal manera que me ha sido posible hacer acopio de todo el conocimiento del que disponía y más para sacar adelante este trabajo.

"No tengo ningún talento especial. Solo soy apasionadamente curioso".

- Albert Einstein

Resumen

Este proyecto surge como continuación del proyecto europeo VIMETRI-MAC, con el objetivo de estudiar los riesgos relacionados con eventos meteorológicos que generen situaciones de alarma sobre la población. Promueve una adaptación frente al cambio climático por medio de la gestión de riesgos mediante la aportación de datos de redes de sensores meteorológicos con el fin de crear mecanismos automatizados que ayuden a cumplir los objetivos de detección, previsión y cuantificación del estado del clima en una región de manera telemática.

Como aportación principal dentro del proyecto VIMETRI-MAC se me ha encomendado la tarea de mejorar los sistemas de detección de precipitaciones atmosféricas y la creación de un sistema de predicción de estas basándose únicamente en la atenuación del nivel de potencia de los radioenlaces terrenos y satelitales disponibles.

El factor de mejora se ha producido optimizando la arquitectura mediante dos redes LSTM mejorando las métricas de calidad en un 5,88% respecto a nuestros modelos y resultados de referencia. De igual modo, se ha creado una arquitectura de red neuronal capaz de predecir eventos de lluvia a 20 minutos en el futuro con la calidad suficiente como para considerarla funcional.

Este documento contiene toda la base teórica sobre electromagnetismo, fenómenos meteorológicos, Machine Learning y análisis estadístico necesario como para poder crear las arquitecturas antes mencionadas e interpretar sus resultados.

Palabras clave

Detección de precipitaciones, predicción de precipitaciones, Machine Learning, redes LSTM, parámetros de calidad estadísticos, proyecto VIMETRI-MAC.

Abstract

This project is part of the European project VIMETRI-MAC, aimed at studying risks associated with weather events that generate alarming situations among the population. It promotes climate change adaptation through risk management by utilizing data from networks of meteorological sensors to create automated mechanisms that aid in achieving the objectives of detection, forecasting, and quantification of the regional climate conditions in a telematic manner.

As a key contribution within the VIMETRI-MAC project, I have been entrusted with the task of enhancing the detection systems for atmospheric precipitation and developing a prediction system based solely on the power level attenuation of available terrestrial and satellite radio links.

By adopting two LSTM networks as the new architecture, we have achieved significant improvements, enhancing quality metrics by 5.88% compared to previous models. Additionally, we have successfully developed a neural network capable of accurately predicting rain events 20 minutes into the future, reaching a level of functionality that warrants its practical application.

This document comprehensively covers the theoretical foundations of electromagnetism, meteorological phenomena, Machine Learning, and statistical analysis required to design the aforementioned architectures and interpret their results.

Keywords

Detection of precipitation, prediction of precipitation, Machine Learning, LSTM networks, statistical quality metrics, VIMETRI-MAC project.

Contenido

Listado de Figuras	VIII
Listado de Tablas.....	XI
Listado de Ecuaciones	XII
Listado de Códigos	XIII
Listado de Anexos	XIV
Acrónimos	XV
Prólogo	1
PARTE I.....	2
MEMORIA.....	2
Capítulo 1. Introducción.....	3
1.1. Descripción del proyecto	4
1.2. Objetivos.....	6
1.3. Estado del arte.....	6
1.4. Estructura del documento	8
Capítulo 2. Fundamentos teóricos	11
2.1. Ondas electromagnéticas	12
2.2. Atenuación de la señal	14
2.3. Fenómenos meteorológicos	16
2.3.1. Atmósfera.....	16
2.3.2. Niebla	18
2.3.3. Nubes.....	19
2.3.4. Precipitaciones	21
Capítulo 3. Proyecto VIMETRI-MAC	29

CONTENIDO

3.1. Introducción	30
3.2. Proyectos anteriores	32
3.3. Línea de investigación actual.....	33
Capítulo 4. Sistema de captación de datos	34
4.1. Introducción	35
4.2. Radioenlaces terrestres.....	35
4.2.1. Red TETRA.....	37
4.2.2. RESCAN	40
4.3. Enlace satelital	41
4.3.1. Prototipo satelital.....	43
4.4. AEMET.....	45
4.5. Importación de datos.....	47
4.5.1. Estaciones AEMET	47
4.5.2. Estación propia.....	49
4.5.3. Estación satelital.....	50
Capítulo 5. Procesado de datos.....	53
5.1. Introducción	54
5.2. Base de datos de la red satelital	54
5.2.1. Sincronismo.....	55
5.2.2. Obtención del umbral óptimo de detección.....	57
5.2.3. Detección y corrección de GAPS	58
5.3. Base de datos de la red terrestre.....	67
5.3.2. Sincronismo.....	68
5.3.3. Obtención de secuencias de precipitaciones	69
Capítulo 6. Redes LSTM para detección y predicción.....	70

CONTENIDO

6.1. Introducción	71
6.2. Algoritmos de Machine Learning	71
6.2.1. Redes Neuronales	73
6.2.2. Tipos de aprendizaje de Machine Learning	81
6.2.3. Parámetros de calidad.....	83
6.3. Red LSTM	86
Capítulo 7. Experimentos y Resultados.....	89
7.1. Algoritmo de detección.....	90
7.1.1. Variables principales	90
7.1.2. Estructura del algoritmo	93
7.1.3. Resultados	100
7.2. Algoritmo de predicción	105
7.2.1. Variables principales	106
7.2.2. Estructura del algoritmo	106
7.2.3. Resultados	110
Capítulo 8. Conclusiones	114
8.1. Algoritmo de detección.....	115
8.2. Algoritmo de predicción	115
8.3. Líneas futuras.....	116
PARTE II:.....	121
PRESUPUESTO.....	121
P.1.Introducción.....	122
P.2. Recursos hardware.....	122
P.3. Recursos software.....	123
P.4. Recursos humanos	124

CONTENIDO

P.5. Material fungible	124
P.6. Redacción del documento.....	125
P.7. Aplicación de impuestos y coste final	125
PARTE III:	127
PLIEGO DE CONDICIONES	127
C.1. Introducción y objetivos.....	128
C.2. Especificaciones de los materiales	128
C.3. Reglamentación y normativa aplicables.....	129
PARTE IV:	130
ANEXOS	130

Listado de Figuras

Figura 1. Onda electromagnética.....	12
Figura 2. Espectro electromagnético.	14
Figura 3. Composición de la atmósfera terrestre.....	17
Figura 4. Atenuación específica debida a los gases.....	18
Figura 5. Tipos de nube y alturas respecto de la base	21
Figura 6. Gráficas de k y α para ambas polarizaciones.	27
Figura 7. Localización de las estaciones meteorológicas en Gran Canaria.....	31
Figura 8. Arquitectura TETRA.....	39
Figura 9. Funcionamiento TETRA.....	39
Figura 10. Flota HISPASAT y área de cobertura.....	42
Figura 11. Accionistas principales de HISPASAT.....	42
Figura 12. Vaisala Rain Gauge RG13.	43
Figura 13. Estación meteorológica y antena.....	44
Figura 14. Conexionado del sistema de captación y transmisión de datos.....	45
Figura 15. Distribución de los vanos en la isla de Gran Canaria.....	47
Figura 16. Ejemplo de adaptación de archivos en formato csv a xls.	48
Figura 17. Aplicación web para obtener los datos de la estación del Pabellón B.	49
Figura 18. Ejemplo de los datos meteorológicos recogidos en el Pabellón B.....	50
Figura 19. Evento de lluvia no nulo.	50
Figura 20. Aplicación web para obtener los datos de la estación satelital.	51
Figura 21. Archivos xlsx del radioenlace satelital.....	51
Figura 22. Ejemplo de los datos satelitales obtenidos.	51
Figura 23. Diagrama de flujo del procesado de datos del Pabellón B.....	55
Figura 24. Comparación entre muestras de potencia y meteorológicas.	56
Figura 25. Diagrama de flujo para calcular el mejor umbral de detección de lluvia.....	58
Figura 26. GAP producido en los niveles de precipitación.	59
Figura 27. Señal de niveles de potencia original.....	61
Figura 28. Resultados de pasar la señal original por el filtrado de cierre.....	63

LISTADO DE FIGURAS

Figura 29. Señal de niveles de potencia filtrada.....	64
Figura 30. Lluvia real frente a lluvia detectada y coincidencias para un umbral dado.	65
Figura 31. Matriz de confusión para un valor de umbral dado.....	66
Figura 32. Evolución del parámetro F1-Score para valores de umbrales.....	67
Figura 33. Diagrama de flujo para el procesado de datos de las estaciones AEMET.	68
Figura 34. Secuencia con todos los niveles de potencia y al menos un caso de lluvia.	69
Figura 35. Áreas de la Inteligencia Artificial.	72
Figura 36. Neurona biológica y neurona artificial.....	75
Figura 37. Funciones de activación de un perceptrón.	76
Figura 38. Modelo de perceptrón multicapa.....	78
Figura 39 Modelo de red neuronal recurrente.	79
Figura 40. Arquitectura de red neuronal recurrente básica.	80
Figura 41. Red neuronal recurrente expandida.....	80
Figura 42 Predicción basada en la cerveza.....	81
Figura 43. Red LSTM.....	87
Figura 44. Línea principal de la red LSTM.	87
Figura 45. Almacenamiento en una red LSTM.	88
Figura 46. Salida de una red LSTM.	88
Figura 47. Resultados de los experimentos.	91
Figura 48. Evolución de la validación del modelo.	92
Figura 49. Número de experimentos realizados.	93
Figura 50. Ejemplo de métricas de calidad de un modelo.....	99
Figura 51. Histograma de Sensibilidad del detector.....	100
Figura 52. Histograma de Especificidad del detector.....	101
Figura 53. Histograma de Exactitud del detector.	101
Figura 54. Histograma de F1-Score del detector.....	102
Figura 55. Histograma de Cohen´s kappa del detector.....	102
Figura 56. Mejores parámetros de detección.....	104
Figura 57. Resultados del mejor detector.	105
Figura 58. Resultados del predictor con la matriz de confusión global.	110
Figura 59. Resultados del predictor con la matriz de confusión diversa.	110

LISTADO DE FIGURAS

Figura 60. Resultados para el mejor predictor de 150 según el mejor detector.	111
Figura 61. Resultados para el mejor predictor de 350 según el mejor detector.	111
Figura 62. Resultados para el mejor predictor de 150.....	113
Figura 63. Resultados para el mejor predictor de 350.....	113

Listado de Tablas

Tabla 1. Coeficientes para polarización horizontal.	26
Tabla 2. Coeficientes para polarización vertical	26
Tabla 3. Datos recibidos del radioenlace.....	40
Tabla 4. Parámetros de diseño de un radioenlace satelital.	41
Tabla 5. Datos proporcionados por la AEMET.....	46
Tabla 6. Matriz de confusión.....	84
Tabla 7. Clasificación del coeficiente de Cohen´s kappa.....	86
Tabla 8. Recursos hardware y su vida útil.....	122
Tabla 9. Coste de equipos hardware.....	122
Tabla 10. Recursos software y su vida útil.....	123
Tabla 11. Coste de equipos hardware.....	123
Tabla 12. Coste de recursos humanos.....	124
Tabla 13. Coste del material fungible.....	124
Tabla 14 Coste de la redacción del documento.....	125
Tabla 15. Presupuesto total.....	125

Listado de Ecuaciones

<i>(Ec. 1) Ley de Gauss para el campo eléctrico.....</i>	<i>12</i>
<i>(Ec. 2) Ley de Gauss para el campo magnético</i>	<i>12</i>
<i>(Ec. 3) Ley de Faraday.....</i>	<i>12</i>
<i>(Ec. 4) Ley de Ampère-Maxwell</i>	<i>12</i>
<i>(Ec. 5) Pérdida básica de transmisión en el espacio libre en los sistemas radar.....</i>	<i>15</i>
<i>(Ec. 6) Intensidad de una precipitación</i>	<i>25</i>
<i>(Ec. 7) Atenuación específica del agua</i>	<i>25</i>
<i>(Ec. 8) Variable k.....</i>	<i>25</i>
<i>(Ec. 9) Variable α.....</i>	<i>25</i>
<i>(Ec. 10) Variable k. Caso general</i>	<i>27</i>
<i>(Ec. 11) Variable α. Caso general.....</i>	<i>27</i>
<i>(Ec. 12) Umbral de lluvia.....</i>	<i>64</i>
<i>(Ec. 13) F1-Score</i>	<i>66</i>
<i>(Ec. 14) Salida del perceptrón.....</i>	<i>76</i>
<i>(Ec. 15) Función de error.....</i>	<i>83</i>
<i>(Ec. 16) Error cometido por la red</i>	<i>83</i>
<i>(Ec. 17) Precisión de una red neuronal</i>	<i>85</i>
<i>(Ec. 18) Exactitud de una red neuronal</i>	<i>85</i>
<i>(Ec. 19) Sensibilidad de una red neuronal.....</i>	<i>85</i>
<i>(Ec. 20) Especificidad de una red neuronal.....</i>	<i>85</i>
<i>(Ec. 21) F1-Score de una red neuronal.....</i>	<i>85</i>
<i>(Ec. 22) Cohen's kappa de una red neuronal.....</i>	<i>86</i>

Listado de Códigos

Código 1. Creación xlsx de Detector_Text_CTR_COMBINADO_Gabriel_01.m.....	93
Código 2. Cálculo del Cohen's kappa de ParametrosKappa.m.....	94
Código 3 Parámetro layers de Detector_Text_CTR_COMBINADO_Gabriel_01.m.....	95
Código 4. Parámetro options de Detector_Text_CTR_COMBINADO_Gabriel_01.m.....	97
Código 5. Matriz de confusión de Detector_Text_CTR_COMBINADO_Gabriel_01.m...	99
Código 6. Búsqueda del mejor Cohen's kappa del BuscarMejorNN.m.	103
Código 7. Variables principales de Lanza_Detector_Lluvia_COMBINADO8_4.m.....	107
Código 8. Fase de predicción de predictor_v1.m.....	108
Código 9. Matrices de confusión predictor_v1.m.....	109

Listado de Anexos

Anexo 1. Función GeneradorMesesf.	131
Anexo 2. Script Sincronismo_Satelital_Mes_COMBINADO6.m	134
Anexo 3. Script Obtención_koptValue_Mes_RSLd.m	136
Anexo 4. Función cierre.m	142
Anexo 5. Función dilata.m	143
Anexo 6. Función erosiona.m.....	144
Anexo 7. Script Lanza_Detector_Lluvia_COMBINADO8_4.m	145
Anexo 8. Script Detector_Test_CRT_COMBINADO_Gabriel_01.m.....	147
Anexo 9. Script ParametrosKappa.m	156
Anexo 10. Script BuscarMejorNN.m	158
Anexo 11. Script predictor_v1.m	160

Acrónimos

<i>Siglas</i>	<i>Descripción</i>
AEMET	<i>Agencia Estatal de Meteorología</i>
AI	<i>Artificial Intelligence</i>
BS	<i>Base Station</i>
CDTI	<i>Centro para el Desarrollo Tecnológico e Industrial</i>
CMNs	<i>Commercial Microwave Networks</i>
CNAF	<i>Cuadro Nacional de Atribución de Frecuencias</i>
DL	<i>Deep Learning</i>
DQPSK	<i>Dual-polarization Quadrature Phase Shift Keying</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
ISDN	<i>Integrated Services Digital Network</i>
LLC	<i>Logical Link Control</i>
LSTM	<i>Long Short-Term Memory</i>
MAC	<i>Media Access Control</i>
ML	<i>Machine Learning</i>
NaN	<i>Not a Number</i>
NN	<i>Neuronal Networks</i>
OMM	<i>Organización Meteorológica Mundial</i>
PSTN	<i>Public Switched Telephone Network</i>
RESCAN	<i>Red de Emergencias y Seguridad de Canarias</i>
RHS	<i>Relativity Humidity Sensor</i>
ROC	<i>Receiver Operating Characteristic</i>
RTD	<i>Resistance Temperature Detector</i>
SEPI	<i>Sociedad Estatal de Participaciones Industriales</i>
SGD	<i>Stochastic Gradient Descent with Momentum</i>

ACRÓNIMOS

SHF	<i>Super High Frequency</i>
TDMA	<i>Time-Division Multiple Access</i>
TETRA	<i>Terrestrial Trunked Radio</i>
TFM	<i>Trabajo Fin de Máster</i>
TIC	<i>Tecnologías de la Información y la Comunicación</i>
TP	<i>True Positive</i>
ULPGC	<i>Universidad de Las Palmas de Gran Canaria</i>

Prólogo

Pese a su ausencia en el conjunto de obras de las que se compone la lista bibliográfica de este trabajo, debo añadir *El Discurso del Método*, del filósofo, matemático y físico francés, René Descartes, dado a que su filosofía y método de cuestionarse todo lo que supo para reconstruir su conocimiento en base a deducciones lógicas, será empleada en este documento. Para el inicio de su elaboración comenzaremos olvidándonos de lo conocido e iremos edificando solidamente el conjunto de páginas de las que se compone este trabajo. Es por ello, que contendrá capítulos que expliquen desde el conocimiento más básico hasta el más elaborado, permitiendo así, al lector, marcar su propio ritmo de lectura y será su capacidad y ganas de entender lo que aquí se expone, lo único que limite su avance en las páginas posteriores.

El trabajo, no solo deja en claro mi colaboración en el proyecto, sino que hace acopio de toda la información que ha pasado por mis manos además de registrar cada una de las experiencias y los pasos que se han dado para llegar al resultado final. Siendo todo esto, debidamente documentado y justificado intentando ser lo más objetivo posible.

El objetivo de documentarlo y explicarlo todo desde el más básico nivel hasta llegar al punto en el que me encuentro mientras redacto estas páginas, no es otro sino el de que sirva como base para futuros proyectos. Se intenta por ello, que todo quede realmente cristalino ante la vista de cualquiera y que, si se sigue los pasos marcados y se progresa adecuadamente, cualquiera pueda ser capaz de llegar a las mismas conclusiones que se redactan al final del trabajo.

La transparencia que se emplea en la redacción de este proyecto junto a las explicaciones, desde la más sencilla hasta la más compleja, hace, bajo mi punto de vista, que sea fácilmente replicable por cualquiera que se atreva a embarcarse en un proyecto semejante al que dejo expuesto en este Trabajo de Fin de Máster.

PARTE I.

MEMORIA

Capítulo 1. Introducción

En este capítulo se introduce el contexto de la realización de este TFM, su estructura, los objetivos a cumplir, un resumen del estado del arte sobre los campos que se tratan en este trabajo y una explicación de la estructura de este proyecto.

1.1. Descripción del proyecto

La meteorología se define como el estudio científico de los procesos naturales que acontecen en la atmósfera y ejercen influencia sobre las condiciones atmosféricas y climáticas a nivel global [1]. En este análisis se consideran múltiples variables ambientales que conllevan a la manifestación de fenómenos meteorológicos tales como la precipitación pluvial, la formación de nieve, el granizo, la presencia de calima, entre otros. Estos eventos, tanto de manera individual como en combinación, afectan de forma habitual aspectos como la intensidad lumínica, la humedad atmosférica, los factores eléctricos, entre otros elementos relevantes.

Dada la intrincada naturaleza y la amplitud de variables que requieren ser consideradas en el análisis meteorológico, se comprende que la supervisión de las variables ambientales despierte el interés de diversos actores sociales y empresariales. Esto se debe a que un aumento en la precisión de las predicciones de precipitaciones en una determinada región tiene un impacto directo en la optimización de la gestión de recursos para hacer frente a posibles condiciones climáticas adversas [2]. De esta manera, es posible mitigar los riesgos mediante la implementación de medidas preventivas por parte de los servicios de emergencia y los organismos encargados de la gestión ambiental. Asimismo, las predicciones pueden facilitar mejoras en la administración de los recursos hídricos, como la planificación de sistemas de riego o la regulación de embalses [2].

En los últimos años, el crecimiento exponencial de las metodologías de análisis de datos ha propiciado avances significativos en el campo de las técnicas de Aprendizaje Máquina (Machine Learning, conocido como ML en su traducción del inglés). Estas innovadoras técnicas han demostrado su capacidad para abordar con éxito una amplia gama de desafíos en los que se involucra un elevado número de variables [3].

El Aprendizaje Máquina constituye una disciplina dentro del campo de la inteligencia artificial que se focaliza en el desarrollo de sistemas computacionales capaces de aprender y mejorar su desempeño sin requerir una programación explícita para una tarea específica [3]. En lugar de ser codificados con un conjunto determinado de reglas, estos

sistemas se entrenan mediante la exposición a grandes volúmenes de datos, y a través de algoritmos matemáticos, son capaces de identificar patrones y relaciones inherentes a dichos datos, permitiéndoles realizar predicciones y tomar decisiones informadas [3].

Como fundamentación para dicho sistema computacional, se proporciona acceso a una base de datos que almacena los niveles de potencia correspondientes a una serie de enlaces de radio, tanto terrestres como satelitales, que facilitan la expansión de los sistemas y servicios de comunicación. Estos sistemas operan en la banda de frecuencia SHF (Super High Frequency en inglés) del espectro electromagnético, la cual abarca un rango de frecuencias que va desde los 3 GHz hasta los 30 GHz.

La motivación detrás de esta investigación se enfoca en la fluctuación de la señal en la banda de microondas utilizada en los enlaces de radio. Esta señal es altamente sensible a la presencia de fenómenos meteorológicos como la humedad, presión y temperatura, así como a otros agentes atmosféricos [4]. Es posible caracterizar su variación cuando es afectada por hidrometeoros como la lluvia, la nieve o el granizo, así como por litometeoros como la calima y los remolinos, entre otros [5]. En este Trabajo Fin de Máster, se investigarán diversas estrategias con el objetivo de caracterizar las variaciones en la señal de los enlaces de radio y asociarlas a fenómenos de lluvia, además de estimar la cantidad de precipitación.

Este Trabajo Fin de Máster representa un proyecto centrado en sistemas de telecomunicaciones terrestres y satelitales desplegados. Su objetivo principal es agregar una nueva función que contribuya a la investigación climatológica. De esta manera, se propone una utilidad adicional al funcionamiento tradicional de los sistemas, al tiempo que se establece una alternativa complementaria a los métodos de medición basados en estaciones meteorológicas [6]. Esto genera un valor agregado a las redes de comunicación al incorporar un propósito adicional y beneficios en términos de contribución al campo de la climatología.

Las líneas de investigación de este trabajo se llevan a cabo en el marco del Proyecto VIMETRI-MAC. En consecuencia, se utilizan los enlaces de radio terrestres y satelitales desplegados en las Islas Canarias como parte integral de la investigación.

En líneas generales, este proyecto presenta una novedad al ampliar el alcance temporal en los sistemas de predicción meteorológica de lluvia a través de la utilización de redes de comunicación satelital desplegadas en las Islas Canarias [7]. Este enfoque pretende establecer una herramienta con potencial futuro que contribuya en la prevención de catástrofes, como sequías e inundaciones causadas por eventos climáticos extremos [8].

1.2. Objetivos

El objetivo principal del proyecto se basa en la implementación de algoritmos de ML para la cuantificación y predicción de precipitaciones de lluvia basado en los niveles de potencia involucrados en comunicaciones terrenas y satelitales a frecuencias de microondas distribuidas por diversas zonas de observación en las Islas Canarias. Estos objetivos se desglosan en los siguientes objetivos específicos:

- O1. Estudiar la influencia de las precipitaciones en los radioenlaces.
- O2. Estudiar los algoritmos para la cuantificación y predicción de lluvia.
- O3. Gestionar el sistema VIMETRI respecto a datos de hidrometeoros.
- O4. Gestionar conjuntos de datos de radioenlaces y estaciones meteorológicas.
- O5. Diseñar del esquema general del sistema global.

1.3. Estado del arte

En este proyecto, se trabajan diversos aspectos de la meteorología junto a su clasificación y predicción por medio de técnicas de ML. Estos pueden incluir cuestiones tales como las aberraciones que pueden provocar ciertos fenómenos meteorológicos y como aprovecharlos para crear un sistema que los identifique y pueda ayudar a crear alarmas ampliando la ventana de visión de estos fenómenos a un futuro cercano. De igual modo, entra en debate sobre qué tipo de red neuronal puede llegar a ser más eficiente para el caso que se estudia. Es por ello, que el estado del arte se dividirá en dos partes principales, una primera que informe acerca de los fenómenos meteorológicos y como

afecta a la señal de los radioenlaces y una segunda sobre los tipos de modelos regresivos usados para la detección y predicción de estos agentes atmosféricos.

En primer lugar, se referencia a la Agencia Estatal de Meteorología (AEMET) cuya función es la de proporcionar información meteorológica y climatológica precisa y actualizada a la sociedad española, así como realizar investigaciones en el campo de la meteorología y la climatología. Entre sus actividades destacan la emisión de boletines y pronósticos del tiempo, alertas meteorológicas, informes climatológicos y la gestión y mantenimiento de una extensa red de observación meteorológica distribuida por todo el territorio español [5]. Parte de los datos se obtienen por medio de esta entidad al igual que la información referente a fenómenos atmosféricos. Como avance científico, se sitúa el proyecto VIMETRI-MAC cuyos objetivos se alinean con los de este proyecto que resulta de la continuación de proyectos similares anteriores como el de Elena Encina Pérez [9], el de Teresa Gubern González [10], el de Naira Dolores Rosales Hernández [11] y el de Cristian Tacoronte Rivero [8] del cuál este trabajo es su continuación.

En todos los trabajos anteriores, entre otros más, se basaban, al igual que este, en hacer detecciones climatológicas por medio de técnicas de ML. La viabilidad de esta idea se ve reforzada con resultados los vistos en el artículo de G. Kaur [12], dónde se centra en la aplicación de técnicas de minería de datos en el campo de la meteorología y examina en detalle diferentes enfoques, como la clasificación, la regresión, la agrupación y las redes neuronales. También se exploran técnicas específicas de minería de datos, como el árbol de decisión, las máquinas de soporte vectorial, los algoritmos genéticos y el análisis de series temporales. Además, describe las técnicas analizando sus ventajas y limitaciones, y proporcionando ejemplos de aplicaciones prácticas en meteorología. Otro ejemplo en el que se ha basado este proyecto es en el trabajo de M.G. Cortina Januchs [13]. En este proyecto anterior se demuestra la viabilidad de los modelos empleados para observar la evolución de los contaminantes atmosféricos empleando técnicas de ML. Finalmente, cabe resaltar la obra que establece la base sobre los fundamentos de los estudios acerca de fenómenos atmosféricos de Wallace, John M. y HOBBS, Peter Victor [1].

En segundo lugar, respecto a las técnicas de ML empleadas se han estudiado los trabajos de Acosta-Coll, M., Ballester-Merelo, F. y De la Hoz-Franco, E. [14], dónde su

red es capaz de dar el servicio de alarma para un posible evento climatológico que pueda poner en riesgo a la población. El artículo de Alley, R. B, K. A, Emanuel y Zhang, F. [15], dónde se emplea una red LSTM para estimar la altura de un río en base a fenómenos climatológicos y prevenir su desbordamiento llegando a hacer predicciones que no superan el medio metro de diferencia con la realidad. Otro proyecto de interés para formar la base de este trabajo es el realizado por Liberman, Y., Samuels, R., Alper, P. y Messer, H. [16]. En este artículo se consideran los aspectos positivos y negativos de los métodos de radar y ML con el fin de desarrollar un nuevo algoritmo para mejorar la medición y el mapeo de la lluvia, basado en la fusión de datos de diferentes fuentes. La integración se basa en un promedio ponderado óptimo de los dos conjuntos de datos, teniendo en cuenta la ubicación, el número de enlaces, la intensidad de la lluvia y el intervalo de tiempo. Los resultados indican que, mediante el uso del nuevo método, no solo se generan reconstrucciones de lluvia en 2D más precisas en comparación con las intensidades reales de la lluvia en el espacio, sino que también se extienden los mapas reconstruidos al área de cobertura máxima. Al analizar tres eventos de lluvia significativos, se demuestra que el método empleado en dicho estudio supera a las redes comerciales de microondas (CMNs) o al radar en la estimación de la tasa de lluvia, tanto para mediciones espaciales instantáneas como para el cálculo de la lluvia acumulada total. Finalmente, se recurre al estudio realizado por Iván D. Guerra Moreno [3]. Es dicho estudio, concluye con gran parte del procesado de los datos y marco físico del que se basa el código de este trabajo.

1.4. Estructura del documento

Este proyecto se divide en diversas partes, en primer lugar, un apartado previo a la memoria que incluye la portada, la hoja de firmas, un resumen del documento, un prólogo y los distintos índices de capítulos y otros elementos que lo compone. En segundo lugar, se tiene como PARTE I la memoria del documento que contiene toda la información del proyecto, objetivos, experimentos y resultados entre otros. En tercer lugar, se encuentra en la PARTE II el presupuesto que hace acopio de todos los recursos que han sido necesarios para realizar este proyecto. En cuarto lugar, la PARTE III con el pliego de condiciones que describe las características bajo las que se han realizado los experimentos. Finalmente, en quinto lugar, la PARTE IV, los anexos que ayudan a completar la información expuesta en la memoria.

PARTE I. MEMORIA

Capítulo 1. Introducción

En este capítulo se introduce el contexto de la realización de este TFM, su estructura, los objetivos a cumplir, un resumen del estado del arte sobre los campos que se tratarán en este trabajo y una explicación de la estructura de este proyecto.

Capítulo 2. Fundamentos teóricos

Las ondas electromagnéticas son un fenómeno fundamental en el campo de la física y la ingeniería, con una amplia gama de aplicaciones en diversas áreas de la ciencia y la tecnología. Este capítulo, explora los fundamentos de las ondas electromagnéticas y examina específicamente su relación con las atenuaciones causadas por los hidrometeoros y los litometeoros.

Capítulo 3. Proyecto VIMETRI-MAC

En este capítulo, se aborda en detalle la idea del proyecto VIMETRI-MAC, y se menciona algunas de las actividades llevadas a cabo por el equipo. Se proporciona una amplia descripción de los aspectos clave del proyecto y se destacan las acciones realizadas para su implementación.

Capítulo 4. Sistema de captación de datos

Descripción de los sistemas utilizados para la obtención de los datos que se han utilizado para llevar a cabo los entrenamientos de las redes neuronales y sus comprobaciones.

Capítulo 5. Procesado de datos

Explicación de las transformaciones que sufren los datos de entrada y proceso que siguen hasta convertirse en datos útiles que faciliten el entrenamiento y comprobación de los posteriores experimentos.

Capítulo 6. Redes LSTM para detección y predicción

Descripción de las técnicas y herramientas empleadas en Machine Learning para la creación de modelos de detección y predicción y la medición de calidad de estas.

Capítulo 7. Experimentos y Resultados

Presentación de los modelos creados en la fase de experimentación junto a sus resultados para la detección y predicción de fenómenos de lluvia.

Capítulo 8. Conclusiones

Resumen e interpretación de los resultados obtenidos en la fase de experimentación al igual que la exposición de las líneas futuras de investigación.

Bibliografía

Compilación de todas aquellas fuentes, como trabajos académicos, libros, páginas web, guías y otros recursos, que han brindado respaldo y han fundamentado los conocimientos expuestos en este trabajo.

PARTE II. PRESUPUESTO

Incluye un desglose del presupuesto necesario para la realización de este Trabajo de Fin de Máster. Se presenta una recopilación de los costos requeridos para adquirir los materiales necesarios para construir la infraestructura de recopilación de datos, así como el costo de redacción de este documento entre otros.

PARTE III. PLIEGO DE CONDICIONES

Se detallan los parámetros y límites dentro de los cuales se llevará a cabo la acción del proyecto. En esta sección, se describirán las características mínimas necesarias de los materiales, las versiones de los programas utilizados y el conjunto de leyes y normativas que se seguirán durante el desarrollo del proyecto.

PARTE IV. ANEXOS

Por último, se encuentran los anexos al final del documento, los cuales proporcionan información adicional para aquellos que deseen complementar los detalles presentados en el trabajo.

Capítulo 2. Fundamentos teóricos

Este capítulo explora los fundamentos de las ondas electromagnéticas y examina su relación con las atenuaciones causadas por los hidrometeoros, los litometeoros y otros fenómenos atmosféricos y climatológicos.

2.1. Ondas electromagnéticas

Las ondas electromagnéticas son perturbaciones que se propagan a través del espacio como una combinación de campos eléctricos y magnéticos oscilantes. Son un fenómeno fundamental en la física y se generan por la aceleración de cargas eléctricas. Estas ondas no requieren de un medio material para su propagación, lo que significa que pueden moverse a través del vacío, a diferencia de otras formas de ondas, como las ondas sonoras [17].

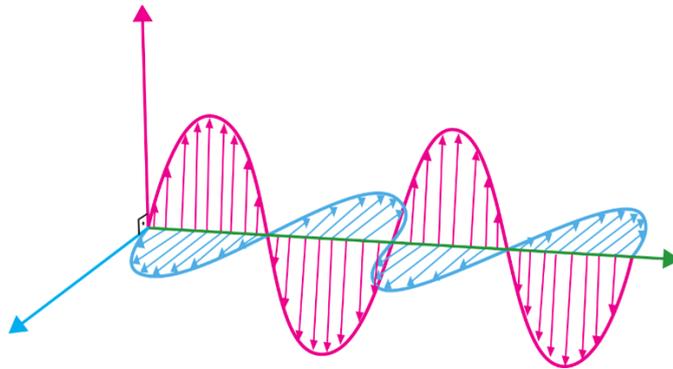


Figura 1. Onda electromagnética.
Fuente: Pixabay, 2023. <https://pixabay.com/>

Las ondas electromagnéticas se describen mediante las ecuaciones de Maxwell, que son un conjunto de cuatro ecuaciones diferenciales parciales formuladas por el físico James Clerk Maxwell en el siglo XIX [17]. Estas ecuaciones relacionan los campos eléctricos y magnéticos con las fuentes de carga y corriente, y establecen cómo estos campos interactúan y se propagan (ver Figura 1). Estas ecuaciones, se conocen popularmente como un conjunto de cuatro ecuaciones como se explican a continuación:

$$\oint E \cdot dA = \frac{1}{\epsilon_0} \int \rho \cdot dV \quad \text{V/m} \quad (\text{Ec. 1})$$

$$\oint B \cdot dA = 0 \quad \text{T} \quad (\text{Ec. 2})$$

$$\oint E \cdot dl = -\frac{d(\int B \cdot dA)}{dt} \quad \text{V/m} \quad (\text{Ec. 3})$$

$$\oint B \cdot dl = \mu_0 \cdot \left(\int J \cdot dA + \epsilon_0 \cdot \frac{d \int E \cdot dA}{dt} \right) \quad \text{T} \quad (\text{Ec. 4})$$

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

La primera ecuación (Ec. 1), conocida como la Ley de Gauss para el campo eléctrico, establece que el flujo eléctrico a través de una superficie cerrada es proporcional a la carga encerrada por esa superficie. Donde E es el campo eléctrico, dA es un elemento diferencial de área en la superficie cerrada, ϵ_0 es la permitividad del vacío, ρ es la densidad de carga eléctrica y dV es un elemento diferencial de volumen encerrado por la superficie [17].

La segunda ecuación (Ec. 2), conocida como la Ley de Gauss para el campo magnético, establece que el flujo magnético a través de una superficie cerrada es siempre cero. Esto significa que no existen fuentes magnéticas puntuales (monopolos magnéticos) en la naturaleza. Donde B es el campo magnético y dA vuelve a ser un elemento diferencial de área en la superficie cerrada [17].

La tercera ecuación (Ec. 3), conocida como la Ley de Faraday, establece que un cambio en el campo magnético a través de una superficie produce un campo eléctrico circulante en esa superficie. Donde E es el campo eléctrico, dl es un elemento diferencial de longitud en una trayectoria cerrada, B es el campo magnético, dA es un elemento diferencial de área encerrado por la trayectoria y dt es el cambio en el tiempo [17].

La cuarta ecuación (Ec. 4), conocida como la Ley de Ampère-Maxwell, establece que una corriente eléctrica o un cambio en el campo eléctrico a través de una superficie produce un campo magnético circulante en esa superficie. Donde B es el campo magnético, dl es un elemento diferencial de longitud en una trayectoria cerrada, J es la densidad de corriente eléctrica, dA es un elemento diferencial de área encerrado por la trayectoria, μ_0 es la permeabilidad del vacío y ϵ_0 es la permitividad del vacío [17].

La propagación de las ondas electromagnéticas se produce a la velocidad de la luz, que es una constante fundamental en el universo y tiene un valor aproximado de 299,792,458 metros por segundo en el vacío. Esto significa que las ondas electromagnéticas pueden viajar distancias extremadamente grandes en períodos muy cortos de tiempo [17].

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

Las ondas electromagnéticas abarcan un amplio rango de longitudes de onda (λ) y frecuencias (f), esto se conoce como el espectro electromagnético. Este espectro incluye desde ondas de radio de baja frecuencia y larga longitud de onda, hasta rayos gamma de alta frecuencia y corta longitud de onda, como se puede observar en la Figura 2. Entre estos extremos se encuentran otras formas de radiación electromagnética, como las microondas, los infrarrojos, la luz visible y los rayos X [17].

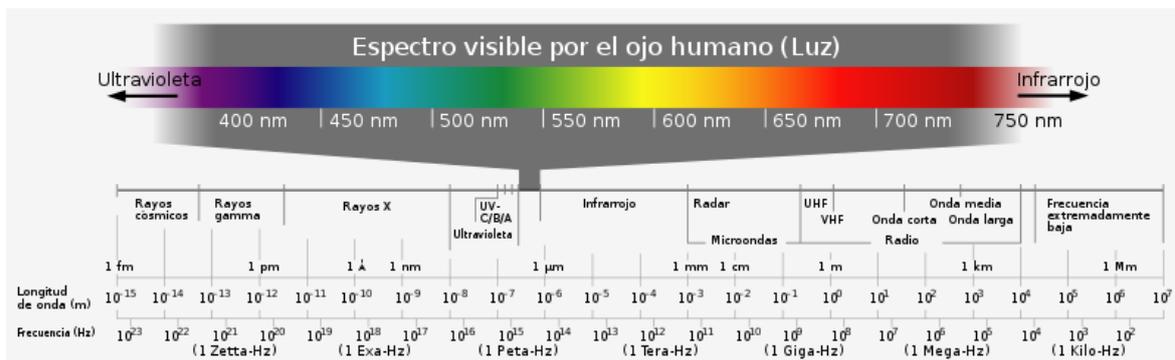


Figura 2. Espectro electromagnético.

Fuente: https://es.wikipedia.org/wiki/Espectro_electromagn%C3%A9tico

Cada tipo de onda electromagnética tiene propiedades específicas y se utiliza en diversas aplicaciones. Por ejemplo, las ondas de radio se utilizan en la radiodifusión y las comunicaciones inalámbricas, mientras que la luz visible permite la visión y es esencial para la fotografía y la óptica. Las aplicaciones médicas utilizan rayos X y radiación gamma para diagnósticos y tratamientos, y las microondas se utilizan en tecnologías como los hornos de microondas y las comunicaciones satelitales [16].

2.2. Atenuación de la señal

Los radioenlaces utilizados en este proyecto propagan ondas de la banda SHF a través de la atmósfera, por encima de los 8 GHz. Las pérdidas de señal en la banda SHF ocurren debido a diversos factores. Entre ellos se encuentra la atenuación del espacio libre que provoca pérdidas en la potencia de la señal radiada debido a que la onda experimenta una dispersión de su potencia en el espacio recorrido suponiendo una mayor pérdida cuanto mayor sea la distancia que recorra [17]. Esta atenuación puede ser calculada siguiendo la Recomendación ITU-R P.525 [18] que explica en detalle la (Ec. 5).

$$L_{br} = 103,4 + 20 \log(f) + 40 \log(d) - 10 \log(\sigma) \quad \text{dB} \quad (\text{Ec. 5})$$

La (Ec. 5) es la pérdida básica de transmisión en el espacio libre en los sistemas de radar que trabajan en la misma frecuencia que los radioenlaces utilizados. Dónde la atenuación de la señal L_{br} se mide en dB y depende de la frecuencia del sistema f en Mhz, la distancia al blanco d en km y la sección transversal del blanco σ en m². Este último parámetro es la relación de potencia total dispersada isotrópicamente y la potencia incidente [18].

Otra de las pérdidas de señal en la banda SHF pueden ocurrir debido a obstáculos y obstrucciones de los objetos físicos como edificios, árboles, montañas u otros que puedan bloquear o debilitar la señal SHF [3]. Estos obstáculos de momento son ignorados debido a que las condiciones del experimento se encuentran en zonas despejadas y los radioenlaces están en visión directa unos con otros.

Por otro lado, se tiene las pérdidas por multipath se producen cuando las señales se reflejan en diferentes superficies antes de llegar al receptor. Estos reflejos pueden superponerse y causar interferencias constructivas o destructivas, lo que resulta en una pérdida de señal o distorsión [17]. Al igual que en el caso anterior se desestimarán estas pérdidas.

Finalmente, la lluvia resulta ser un factor importante en la atenuación de señales en la banda SHF. Las gotas de lluvia actúan como pequeños obstáculos y absorben, dispersan y difractan la señal, lo que puede llevar a una pérdida significativa de potencia. Del mismo modo, la niebla densa también puede afectar la propagación de señales SHF debido a la concentración de partículas de agua como se mencionó anteriormente. De manera similar sucede con la dispersión atmosférica debido a cambios en la densidad, temperatura y humedad del aire. Estos cambios pueden causar que la señal SHF se desvíe y se disperse, lo que lleva a pérdidas adicionales [3].

Es por esto por lo que, a continuación, se desglosan las pérdidas del párrafo anterior ya que estas suponen la base para la experimentación posterior.

2.3. Fenómenos meteorológicos

Los fenómenos meteorológicos tienen un impacto en la atenuación de la señal debido a varias razones. Un ejemplo es la absorción de energía con fenómenos como la lluvia y la niebla que contienen partículas de agua que pueden absorber parte de la energía de la señal. Estas partículas interactúan con la señal electromagnética y causan una pérdida de energía en forma de calor. Cuanto más denso o abundante sea el fenómeno meteorológico, mayor será la absorción de energía y, por lo tanto, mayor será la atenuación de la señal. Del mismo modo, los fenómenos meteorológicos pueden dispersar la energía de la señal en múltiples direcciones. Por ejemplo, las mismas gotas de lluvia pueden dispersar la señal de origen, lo que lleva a una distribución más amplia de la energía lo que disminuye la intensidad de la señal recibida en un punto específico. Por otro lado, también pueden causar difracción de la señal [1]. La difracción es el fenómeno en el cual la señal se curva alrededor de los obstáculos pudiendo resultar en una pérdida de energía de la señal a medida que se aleja de la trayectoria directa entre el transmisor y el receptor [17].

Los fenómenos meteorológicos, como las nubes densas, pueden reflejar o refractar la señal. La reflexión ocurre cuando la señal rebota en la superficie de una nube o de las gotas de agua, cambiando su dirección. La refracción ocurre cuando la señal atraviesa diferentes capas de la atmósfera con diferentes propiedades ópticas, lo que puede causar desviaciones y cambios en la trayectoria de la señal [4]. También se debe considerar que la atmósfera puede cambiar las propiedades del medio a través del cual se propaga la señal. Por ejemplo, las variaciones en la temperatura y la humedad del aire pueden alterar la densidad del medio, lo que afecta la velocidad de propagación de la señal pudiendo provocar pérdidas de señal durante su trayectoria [1].

2.3.1. Atmósfera

La atmósfera es la capa gaseosa de un cuerpo celeste. Los gases que contiene en el interior son atraídos por la gravedad del cuerpo manteniéndose suspendidas a lo largo de su superficie y evitando ser barridas en su totalidad por el viento solar [19].

En nuestro planeta, la atmósfera se conforma una mezcla de distintos gases con distintas propiedades y que van acompañados de partículas sólidas y líquidas en suspensión

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

como puede ser moléculas de agua, cenizas, polen, residuos de combustión, etc. Entre los gases que componen la atmósfera, destacan tres componentes que son los más abundantes, como muestra la Figura 3, son el nitrógeno, el oxígeno y el argón [13]. El resto de los componentes, suponen un menor porcentaje de la composición de la atmósfera que a efectos del experimento a realizar resultan despreciables [7].

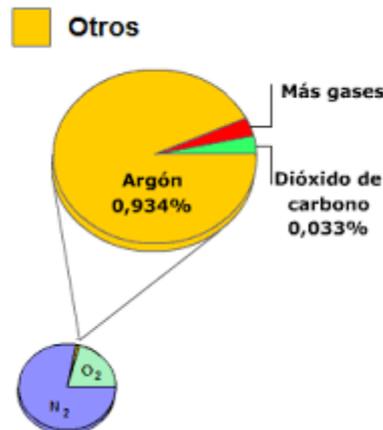
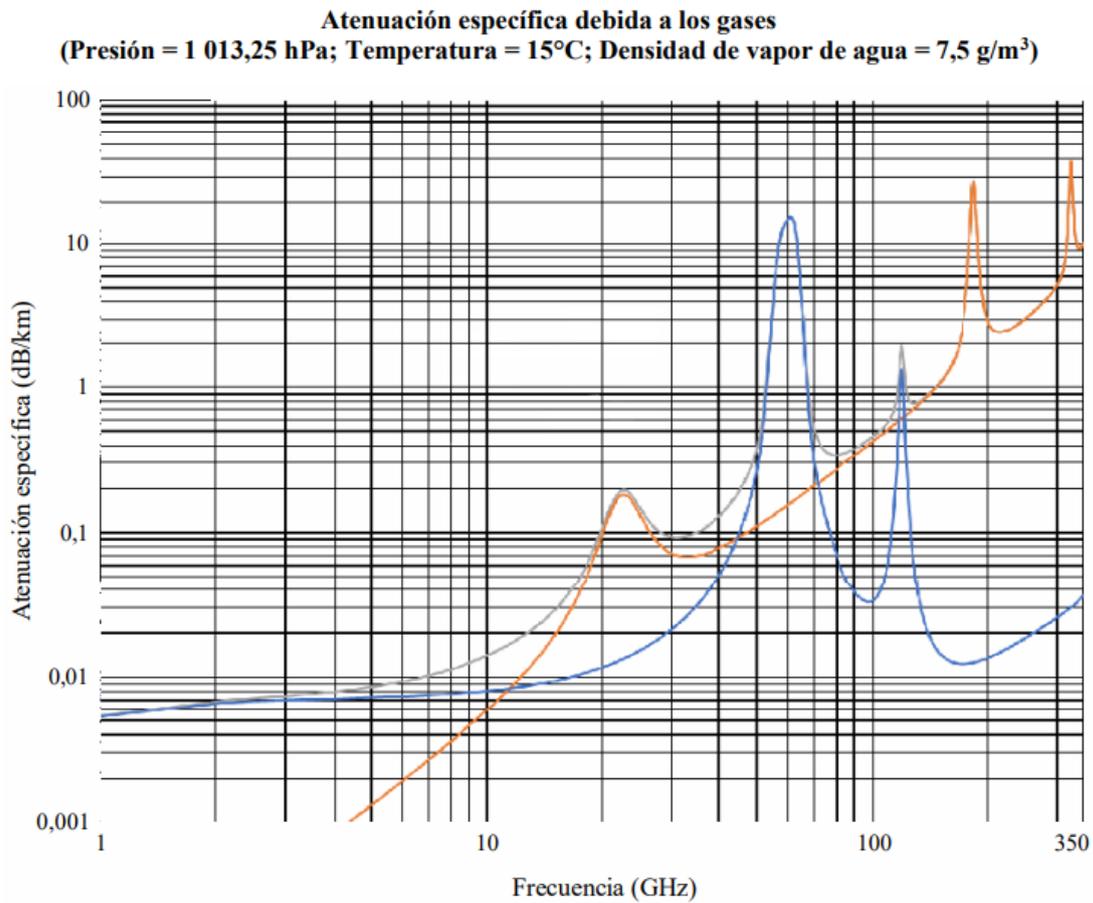


Figura 3. Composición de la atmósfera terrestre.
Fuente: Tacoronte, Cristian, 2022 [8].

Para cuantificar la atenuación de los gases atmosféricos sobre la propagación de la señal se toman en cuenta los dos factores más influyentes, el vapor de agua y el oxígeno. Según la Recomendación UIT-R P.676-12 [19], se obtiene la Figura 4, en la que se puede apreciar cómo se relaciona la frecuencia de la señal con la atenuación provocada por el vapor de agua (rojo) y el oxígeno (azul). De la Figura 4 se puede concluir que, a mayor frecuencia usada, mayores las atenuaciones totales (gris). Dentro del rango de SFH (3-30 GHz) lo más notable es el pico que se produce sobre los 25 GHz dónde debido a una mayor absorción de la señal por parte del vapor de agua lo que resulta ideal para obtener una diferencia significativa a la hora de evaluar las pérdidas de señal y detectar grandes volúmenes de vapor de agua en la atmósfera [8].



*Figura 4. Atenuación específica debida a los gases.
Línea roja: Vapor de agua. Línea azul: Oxígeno. Línea gris: Total.
Fuente: Recomendación UIT-R P.676-12, [19].*

2.3.2. Niebla

La niebla es una forma de condensación de vapor de agua que se produce cerca de la superficie terrestre. Se forma cuando el aire húmedo se enfría y alcanza su punto de rocío, lo que provoca que el vapor de agua en el aire se condense en pequeñas gotas suspendidas en la atmósfera [5]. La niebla se compone de pequeñas gotas de agua líquida, generalmente con un diámetro de menos de 50 micrómetros. Estas gotas son lo suficientemente pequeñas como para permanecer suspendidas en el aire y formar una capa densa de nubes bajas cerca del suelo [5].

La formación de niebla está asociada con condiciones de alta humedad relativa y enfriamiento del aire. Esto puede ocurrir durante la noche o en regiones cercanas a cuerpos de agua, donde la evaporación del agua y la humedad en el aire son altas. La niebla puede reducir la visibilidad impidiendo ver más allá de un kilómetro y dificultar las labores de

navegación terrestre y aérea [5]. Además, puede afectar la propagación de señales de comunicación inalámbrica al dispersar y atenuar las ondas electromagnéticas. La atenuación causada por la niebla depende de la densidad de las gotas de agua en suspensión y puede variar desde una ligera pérdida de señal hasta una atenuación significativa, especialmente en frecuencias más altas [20].

Dado el clima de Canarias junto a la localización de los radioenlaces, se desestiman los efectos de la niebla en la señal, esto se debe a que la probabilidad de que las condiciones para que se produzca una niebla que provoquen una obstrucción en los radioenlaces son prácticamente nulas. No obstante, dado a que este proyecto continúa en evolución, se puede incluir en futuros trabajos [8].

2.3.3. Nubes

Las nubes son conjuntos de moléculas gotas de agua, cristales de hielo, o una combinación de ambas. El diámetro de la partícula líquida es del orden de 200 micrómetros siendo las gotas de mayor tamaño el fenómeno conocido como lluvia [21].

Las nubes se encuentran de forma general en la troposfera, con excepción de ciertos tipos de nubes como las nubes nacaradas y las noctilucenles o la aparición ocasional de cirrus en la estratosfera baja. Las nubes se forman mediante la condensación de vapor de agua en la atmósfera por medio del desplazamiento vertical de aire que surge mediante el fenómeno de convección (corrientes ascendentes o descendentes de fluidos debido a diferencia de densidades), o el asociado con depresiones y frentes [21]. En el campo de la meteorología, existen varios tipos de nubes con características distintas como se muestra en la Figura 5. Estas se clasifican en cuatro grupos principales: nubes altas, nubes medias, nubes bajas y nubes de desarrollo vertical [21]. A continuación, se expone cada tipo y sus características:

- Nubes Altas.

Las nubes altas se forman en alturas por encima de los 6,000 metros. Estas nubes se componen de cristales de hielo y se asocian a condiciones atmosféricas estables, indicando buen tiempo. Los tres tipos principales de nubes altas son:

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

- i. Cirrus: Nubes delgadas, blancas y fibrosas con apariencia de filamentos o hebras formadas por la congelación del vapor de agua a grandes altitudes.
- ii. Cirrostratus: Nubes delgadas y traslúcidas que cubren grandes áreas del cielo. Pueden aparecer como un velo blanco o gris y a menudo se asocian con la aproximación de un sistema frontal (conocido comúnmente como mal tiempo).
- iii. Cirrocumulus: Nubes en forma de pequeñas masas blancas y esponjosas dispuestas en grupos o filas. Suelen indicar una inestabilidad atmosférica leve.

- Nubes Medias.

Las nubes medias se forman entre los 2,000 y 6,000 metros de altura y están compuestas tanto por gotas de agua como por cristales de hielo. Pueden ser indicadores de cambios en el clima. Los dos tipos principales de nubes medias son:

- i. Altostratus. Nubes grises y relativamente uniformes que cubren todo el cielo. Pueden bloquear parcialmente la luz solar y suelen preceder a la llegada de sistemas frontales.
- ii. Altocumulus. Nubes en forma de masas blancas o grises con sombras definidas. Pueden tener un aspecto ondulado y suelen asociarse con un tiempo estable, aunque también pueden indicar inestabilidad atmosférica.

- Nubes Bajas.

Las nubes bajas se forman por debajo de los 2,000 metros, y generalmente están compuestas por gotas de agua. Estas nubes suelen estar asociadas con condiciones húmedas y pueden generar precipitación. Los dos tipos principales de nubes bajas son:

- i. Stratus. Nubes de color gris uniforme que cubren todo el cielo y tienen una apariencia de neblina. Pueden estar asociadas con llovizna o niebla.
- ii. Stratocumulus. Se presentan en forma de masas bajas, separadas y grises. A menudo se encuentran en capas y pueden generar llovizna ligera.

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

- Nubes de Desarrollo Vertical.

Estas nubes tienen un gran desarrollo vertical y pueden extenderse a través de varias capas de la atmósfera. Están asociadas con el fenómeno de convección y son comunes en días calurosos e inestables. Los dos tipos principales de nubes de desarrollo vertical son:

- Cumulus. Son nubes blancas y esponjosas con bordes definidos. Suelen indicar tiempo estable, pero pueden desarrollarse en nubes de tormenta (cumulonimbus) si hay una fuerte convección.
- Cumulonimbus. Estas nubes son las más grandes y desarrolladas verticalmente. Tienen forma de una montaña con una parte superior en forma de yunque. Son nubes de tormenta y pueden generar fuertes lluvias, relámpagos e incluso granizo.

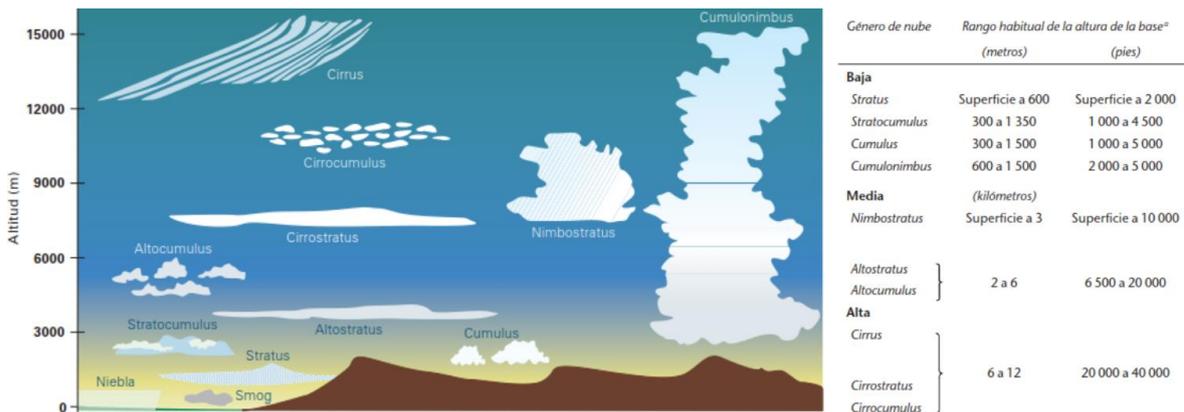


Figura 5. Tipos de nube y alturas respecto de la base
Fuente: Paredes Tavares, Jorge, 2019 [21].

2.3.4. Precipitaciones

Para definir lo que es una precipitación se necesita definir con anterioridad lo que son los meteoros, los hidrometeoros, los litometeoros, los fotometeoros y los electrometeoros. Según la Organización Meteorológica Mundial (OMM) define los conceptos anteriores de la siguiente forma [22]:

“Un meteoro es un fenómeno observado en la atmósfera o sobre la superficie de la tierra, que consiste en una suspensión, una precipitación, o un depósito de partículas líquidas, acuosas o no, o de partículas sólidas, o un fenómeno de la naturaleza de manifestación óptica o eléctrica” [OMM, 2019].

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

“Un hidrometeoro es un meteoro que consiste en un conjunto de partículas de agua líquida o sólida, suspendidas en la atmósfera o cayendo a través de ella, o que son empujadas por el viento desde la superficie de la tierra, o depositadas sobre objetos que se encuentran sobre el suelo o en el aire libre.” [OMM, 2019].

“Un litometeoro es un meteoro que consiste en un conjunto de partículas, la mayoría de las cuales son sólidas y no acuosas. Las partículas están más o menos suspendidas en el aire, o son levantadas por el viento desde el suelo.” [OMM, 2019]

“Un fotometeoro es un fenómeno luminoso producido por la reflexión, refracción, difracción o interferencia de luz proveniente del Sol o la Luna.” [OMM, 2019].

“Un electrometeoro es una manifestación visible o audible de la electricidad atmosférica. Los electrometeoros corresponden a descargas eléctricas discontinuas (relámpagos, truenos) o se producen como fenómenos más o menos continuos.” [OMM, 2019].

En resumen, la precipitación consiste en la caída de un conjunto de partículas, que cumplan con las características mencionadas anteriormente, desde la atmósfera hasta la superficie terrestre o siendo estas partículas evaporadas durante su trayectoria [22].

La formación de las precipitaciones se basa en un ciclo llamado ciclo hidrológico o ciclo del agua. Este describe el proceso de circulación del agua entre las capas de la hidrosfera donde sólo actúa el movimiento de las partículas de agua por medio de cambios en su estado físico. Estos ciclos varían la distribución y el volumen de una cantidad de agua determinada sobre distintas superficies. En función de la cantidad de moléculas de agua concentradas en una región, estas pueden adquirir diferentes formas y tamaños dependiendo del proceso de condensación por el que hayan pasado. Estos se pueden clasificar en [22]:

- Lluvia. Precipitación de un grupo de moléculas de agua en estado líquido entre 0.1 y 5 mm de diámetro. Según su intensidad y duración se puede clasificar a su vez en primer lugar, en llovizna si el diámetro de las gotas de agua es inferior a 0.5mm. Esta lluvia fina tiene poca intensidad y no suele prolongarse durante largos periodos en el tiempo. En segundo lugar, están los chubascos con gotas de mayor

diámetro que caen con gran intensidad, pero de corta duración. En tercer lugar, se encuentran los aguaceros que caen con poca intensidad, pero durante largos periodos de tiempo. Finalmente, en cuarto lugar, se encuentran las trombas que caen con mucha intensidad y con largas duraciones hasta su final.

- Nieve. Precipitación de un grupo de moléculas de agua en estado sólido. Se clasifican en función del diámetro y forma del hidrometeoro. Por un lado, está la nieve, que son cristales de hielo de forma hexagonal y tamaño del orden de centímetros de diámetro. Por otro lado, está la escarcha, siendo esta, pequeños gránulos de hielo que se han terminado de formar sobre objetos de la superficie terrestre. Finalmente, se encuentra el granizo cuya precipitación es en forma de masas de hielo, por lo general esferoides irregulares de 5 a 125mm de diámetro.

Otra forma de clasificar las precipitaciones es mediante su mecanismo de formación. La clasificación tiene en cuenta las condiciones climatológicas en las que originan las acumulaciones de moléculas de agua tales como la masa del aire o el grado de humedad. Esta clasificación se divide en [22]:

- Precipitación convectiva. Este tipo de precipitación ocurre cuando los rayos solares inciden sobre la superficie terrestre lo que provoca un aumento de su temperatura calentando el aire próximo a la superficie lo que lo vuelve menos denso y generando vapor de agua. Este aire asciende a capas superiores de la atmósfera facilitando la formación de nubes de desarrollo vertical. Luego, el vapor de agua se enfría dada su altitud y forma casos de importancia de chubascos y pudiendo también llegar a generar granizos.
- Precipitación orográfica. Este tipo de precipitación tiene lugar cuando corrientes de aire oceánicas se tropiezan con obstáculos, lo que obliga a estas corrientes de aire húmedo a ascender por medio de una elevación mecánica del aire. Este ascenso al igual que en el caso anterior, acarrea una bajada de temperatura del aire que provoca la condensación de las partículas de agua hasta precipitarse. El descenso por la otra parte del obstáculo hace que esta masa se vuelva a calentar originando un tiempo seco y cálido.

- Precipitación frontal. Este tipo de precipitación se puede llegar a entender por una parte como un frente frío o precipitaciones ciclónicas si tiene lugar cuando entran en contacto una masa de aire frío con otra masa de aire caliente siendo denominada la superficie de contacto como frente. Este frente frío se produce si el desplazamiento de ambas masas de aire se encuentra viajando en sentidos contrarios. Siendo la masa de aire frío la que eleva la masa de aire caliente creando una diferencia de presión y generando nubes propias de chubascos y trombas. Por otra parte, se puede llamar como frente cálido o precipitaciones anticiclónicas si ambas masas de aire viajan en el mismo sentido volviendo a ascender la masa de aire caliente por su menor densidad y al ascender la masa de aire cálido se enfría y condensa las partículas de agua que carga ocasionando lloviznas y chubascos moderados.

La región en la que se encuentra la base de experimentación son las Islas Canarias que se encuentran bajo un clima subtropical dada su localización geográfica. Esta zona en particular se encuentra entre la trayectoria de un viento húmedo que procede del noroeste y recibe el nombre de los vientos alisios. Esta corriente de aire deja una capa húmeda sobre una capa seca y genera cambios bruscos en la temperatura. Del mismo modo, esta variación da a lugar a nubes de tipo estratocúmulos e impide la proliferación de nubes de desarrollo vertical. Esto concluye en que las Islas Canarias es una región de escasas precipitaciones. No obstante, dada la vegetación de las islas que captan gran parte de la humedad de estos vientos, como los bosques de laurisilva, generan una presencia continua de nubes bajas que al encontrarse con zonas montañosas generan precipitaciones del tipo orográficas. Un ejemplo de este suceso son Las Breñas en La Palma que ha llegado a registrar entre 1000 y 1500 litros de agua al año [5].

La cuantificación de las precipitaciones se lleva a cabo mediante la medida de tres variables, la intensidad, la duración y la frecuencia que se explicarán a continuación [22]:

- La intensidad. Se define como la cantidad de agua caída por unidad de tiempo. Su expresión es la que se muestra en la (Ec. 6) donde I es la intensidad de la precipitación y se mide en mm/h, P es la altura del agua de precipitación en mm y t la duración de la precipitación en horas.

$$I = P \cdot t \quad \text{mm/h} \quad (\text{Ec. 6})$$

- La duración. Se define como el tiempo que transcurre desde el inicio hasta el fin de la precipitación. Este parámetro se obtiene del pluviograma que es un gráfico o diagrama que representa la intensidad de la precipitación en función del tiempo. Esta gráfica se saca por medio de un pluviógrafo, un instrumento meteorológico diseñado para medir y registrar de manera continua la cantidad y la intensidad de la precipitación.
- La frecuencia. Este parámetro representa el número de apariciones de una precipitación con unas características determinadas de intensidad y duración. Suele representarse mediante el número de apariciones por año.

La atenuación específica debida a la lluvia (γ_R) puede calcularse siguiendo la Recomendación UIT-R P.838-3 [23] Como se muestra en la (Ec. 7). Dicha atenuación se obtiene por medio de una relación entre la intensidad de lluvia R (mm/h) y dos valores k y α que son dependientes de la polarización de la señal emitida, la geometría de la gota de agua y su densidad volumétrica. La expresión que determina la atenuación es la que se muestra en la (Ec. 7). Por otro lado, los valores de k y α se obtienen a partir de (Ec. 8) y (Ec. 9) respectivamente dónde la frecuencia f es en GHz.

$$\gamma_R = k \cdot R^\alpha \quad \text{dB/km} \quad (\text{Ec. 7})$$

$$\log(k) = \sum_{j=1}^4 a_j \cdot \exp \left[- \left(\frac{\log f - b_j}{c_j} \right)^2 \right] + m_k \log f + c_k \quad (\text{Ec. 8})$$

$$\alpha = \sum_{j=1}^5 a_j \cdot \exp \left[- \left(\frac{\log f - b_j}{c_j} \right)^2 \right] + m_\alpha \log f + c_\alpha \quad (\text{Ec. 9})$$

Los valores restantes de la (Ec. 8) y (Ec. 9) pueden consultarse en la Tabla 1 si se desea calcular para una polarización horizontal o en la Tabla 2 para polarización vertical.

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

Tabla 1. Coeficientes para polarización horizontal.
Fuente: UIT-R P.838-3 [23].

	j	ai	bi	ci	mk	ck	mα	cα
k	1	-5,33980	-0,10008	1,13098				
	2	-0,35351	1,26970	0,45400				
	3	-0,23789	0,86036	0,15354	- 0,18961	0,71147	---	---
	4	-0,94158	0,64552	0,16817				
α	1	-0,14318	1,82442	- 0,55187				
	2	0,29591	0,77564	0,19822				
	3	0,32177	0,63773	0,13164	---	---	0,67849	- 1,95537
	4	-5,37610	0,96230	1,47828				
	5	16,17210	- 3,29980	3,43990				

Tabla 2. Coeficientes para polarización vertical
Fuente: UIT-R P.838-3 [23].

	j	ai	bi	ci	mk	ck	mα	cα
k	1	-3,80595	0,56934	0,81061				
	2	-3,44965	-0,22911	0,51059				
	3	-0,39902	0,73042	0,11899	- 0,16398	0,63297	---	---
	4	0,50167	1,07319	0,27195				
α	1	-0,07771	2,33840	- 0,76284				
	2	0,56727	0,95545	0,54039				
	3	-0,20238	1,14520	0,26809	---	---	-0,05374	0,83433
	4	-48,29910	0,79167	0,11623				
	5	48,58330	0,79146	0,11648				

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

Para casos de otras geometrías de polarización que no sean horizontal o vertical, como en polarizaciones lineales o circulares, se pueden calcular los valores k y α mediante una forma más general tal y como se muestra en (Ec. 10) y (Ec. 11) dónde θ es el ángulo de elevación del trayecto y τ el ángulo de elevación del trayecto respecto a la horizontal, ambos casos, en grados. Si se sigue la recomendación antes mencionada, se puede llegar al valor de las nuevas variables faltantes por medio de gráficas que se resumen en la Figura 6.

$$k = \frac{k_H + k_V + (k_H - k_V)\cos^2(\theta)\cos(2\tau)}{2} \quad (\text{Ec. 10})$$

$$\alpha = \frac{k_H\alpha_H + k_V\alpha_V + (k_H\alpha_H - k_V\alpha_V)\cos^2(\theta)\cos(2\tau)}{2k} \quad (\text{Ec. 11})$$

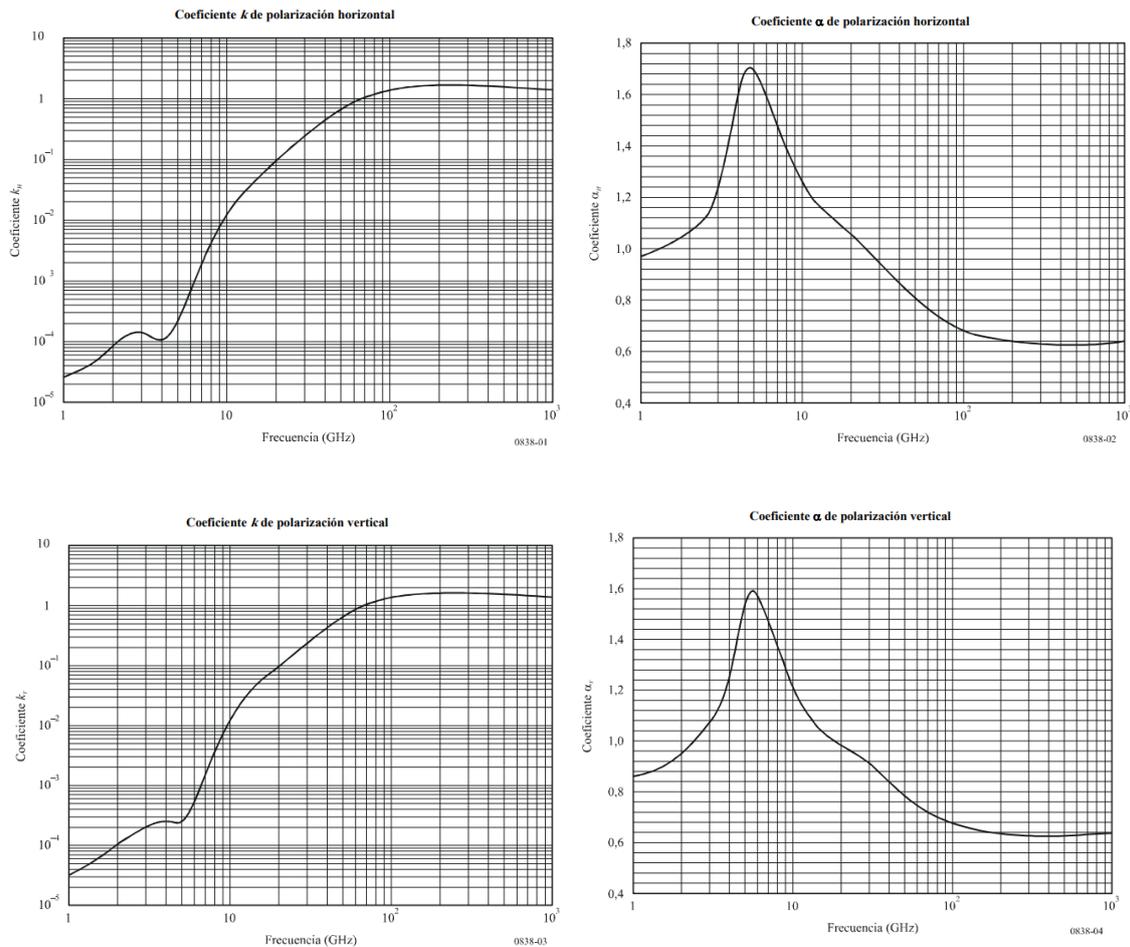


Figura 6. Gráficas de k y α para ambas polarizaciones.
Fuente: UIT-R P.838-3 [23].

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

Dentro del proyecto VIMETRI-MAC se muestrea de forma continua el estado de la atmosfera en busca de casos de lluvia para su detección y predicción. Este muestreo es variable, acomodando un número de muestras adecuado a cada situación. En el caso de que no haya una previsión de emergencia puede llegar a ampliar el margen de muestreo de 20 a 60 minutos. Mientras que, en una situación de emergencia se requiere un menor tiempo entre muestra y muestra para entender la gravedad de la alerta. Es por ello por lo que, uno de los elementos cruciales para obtener conclusiones lógicas del estado del tiempo con una ventana de visión próxima (sobre los 20 minutos) se debe tener en cuenta la tasa de muestreo y obtener datos como máximo con un margen de 5 minutos. Esto se debe ya que una tasa de muestreo menor no presentaría cambios significativos en el estado del tiempo dado a que la evolución de este, pese a ser continuo, es de igual modo, lento [9].

Una vez terminada esta sección se ha logrado completar el primer objetivo O1. Estudiar la influencia de las precipitaciones en los radioenlaces. De esta forma, se dispondrá de la suficiente base teórica como para poder relacionar los resultados con los posibles casos de precipitación conociendo su nivel de potencia de señal en un receptor en los posteriores experimentos.

Capítulo 3. Proyecto VIMETRI-MAC

En este capítulo se aborda en detalle la idea del proyecto VIMETRI-MAC, y se menciona algunas de las actividades llevadas a cabo por el equipo. Se proporciona una amplia descripción de los aspectos clave del proyecto y se destacan las acciones realizadas para su implementación.

3.1. Introducción

Este trabajo supone una parte del proyecto europeo denominado “Sistema de Vigilancia Meteorológica para el Monitoreo de Riesgos Ambientales” (VIMETRI-MAC) aprobado en la primera convocatoria dentro del Programa de Cooperación Territorial con código MAC/3.5b/065 cuya financiación procede del programa INTERREG-MAC V A España- Portugal 2014-2020. Con el objetivo de estudiar los riesgos relacionados con eventos meteorológicos que generen situaciones de alarma sobre la población por casusas de peligro por adversidades ambientales como pueden ser tormentas de gran poder, las denominadas olas de calor, entre otros [11]. La idea original parte de promover una adaptación frente al cambio climático por medio de la gestión de riesgos y prevención mediante la aportación de datos de redes de sensores meteorológicos, para la creación de mecanismos de automatización que ayuden a cumplir los objetivos de detección, previsión y cuantificación del estado del clima en una región de manera telemática [7].

Esta labor se realiza por medio de la observación de la calidad de las conexiones en las redes de telecomunicaciones o de los sistemas de comunicación vía satélite. La idea que sigue este trabajo se centra en la atenuación que sufren las señales durante los eventos de precipitaciones [11]. Como ya se ha comentado, estas situaciones son relevantes cuando se habla de redes de comunicaciones en la banda SHF ya que sus frecuencias son sensibles a ser atenuadas [17]. Estas diferencias de potencia de señal entre cielo despejado y momentos de precipitación son de ayuda para cuantificar, detectar y predecir las precipitaciones dentro de una región [7].

Los datos usados para la experimentación son suministrados por parte de la AEMET al proyecto VIMETRI-MAC. Estos datos proceden de distintas estaciones meteorológicas repartidas entre todas las islas [11]. En el caso de Gran Canaria, por ejemplo, se encuentran localizadas 25 estaciones meteorológicas con la distribución que se muestra en la Figura 7. Estas estaciones recaban datos acerca de las direcciones y velocidades promedias del viento al igual que registra su dirección y velocidad máxima. Del mismo modo, se crean archivos que recogen estos datos junto a las fechas y horas de los registros, el nombre de la estación que captó dichos datos, la humedad relativa, la localización de la estación (longitud, latitud y altitud) y de la misma forma, otros datos de

CAPÍTULO 3. PROYECTO VIMETRI-MAC

interés como la humedad relativa, la medición de las precipitaciones, la presión y la temperatura [7].

Esta base de datos será de vital importancia ya que serán cruciales para los datos de control de las redes neuronales que se entrenen en posteriores experimentos. Una gran variedad de fuentes de datos, permiten en ciertas ocasiones combatir los efectos prolongados de los denominados gaps. Esto gaps son brechas en la base de datos debido a la sucesiva falta de entradas para alimentar la DB (Data Base) lo que genera ventanas de tiempo sin entradas que pueden llegar a impedir un buen entrenamiento del modelo final [7].



*Figura 7. Localización de las estaciones meteorológicas en Gran Canaria.
Fuente: Google Earth. <https://www.google.com/intl/es/earth/>*

En este trabajo, para la creación de un modelo al que se le pueda asociar un buen desempeño, se debe tener en cuenta la configuración de los parámetros más relevantes y adecuados para el modelo. Es por ello por lo que se tiene en cuenta durante la creación de la red neuronal datos de entrada como la posición geográfica ya que en el caso especial de las Islas Canarias se producen grandes variaciones en las condiciones climatológicas a medida que uno se desplaza a lo largo de la isla, lo que se conoce comúnmente como microclimas [24].

3.2. Proyectos anteriores

Dado al elevado número de trabajos que se han llevado a cabo durante el transcurso del proyecto VIMETRI-MAC, a continuación, se exponen los tres últimos que han servido de motivación para llevar a cabo este trabajo.

Por orden de antigüedad, se presenta en primer lugar el trabajo realizado por Sonia Fuente González, en julio de 2020 bajo el nombre “Técnicas de aprendizaje máquina para detección y cuantificación de lluvia mediante radioenlaces terrenos a frecuencia de gigahercios” [7]. Este tiene como conclusiones que una red neuronal es el método que mejores resultados de detección y cuantificación presenta. Para las ocho pruebas realizadas de los ocho radioenlaces escogidos resulta ser el método que mejores valores presenta individualmente teniendo algunos resultados mayores del 98% para los distintos parámetros de calidad (sensibilidad, exactitud, Especificidad y F1-Score).

El segundo trabajo para resumir es el de Naira Dolores Rosales Hernández, de julio de 2021, llamado “Técnicas de aprendizaje máquina para la detección y cuantificación de lluvia sobre imágenes de videovigilancia” [11]. Este trabajo llega a la misma conclusión que el anterior y añadiendo que también resulta ideal para cuantificar las precipitaciones basándose en cámaras de vigilancia con un margen de confianza del 95%.

Finalmente, en tercer lugar, está el trabajo más reciente por parte de Cristian Tacoronte Rivero, de julio de 2022, con el nombre “Diseño e integración de un sistema para la detección y cuantificación de lluvia mediante radioenlaces terrenos y satelitales aplicando técnicas de Machine Learning” [8]. En este proyecto se modifican parámetros como la ventana de visión o la longitud de la secuencia de entrada para llegar a un equilibrio entre el cómputo necesario para dar lugar a la red y su margen de confianza. En este caso se concluye que, variando el tamaño del lote, el batch size, un hiperparámetro que se elige al entrenar una red neuronal y puede variar dependiendo del conjunto de datos y de los recursos computacionales disponible y aumentando el número de iteraciones (epoch) se obtienen mejores resultados que con redes con otros parámetros.

3.3. Línea de investigación actual

Actualmente, el proyecto sigue todas las aportaciones que han hecho los compañeros anteriormente y se continúa experimentando para llegar a una red que se encuentre con un equilibrio entre tiempo de cómputo y margen de confianza. Para ello, se toma en especial consideración la tesis de Iván Daniel Guerra Moreno [3]. Esta parte principalmente de una investigación exhaustiva acerca del preprocesado de los datos junto a las técnicas de manejos de grandes volúmenes de datos para tener en cuenta como cubrir las ventanas de tiempo en el que se pierde la conexión con las estaciones meteorológicas siendo márgenes en los que no hay datos. Esto puede llegar a suponer un gran problema porque no solo se tienen que manipular los datos acerca de los niveles de potencia de los radioenlaces, además tienen que ser solapados correctamente con los partes meteorológicos aportados por la AEMET [3].

Una vez marcada la base de la investigación actualmente se siguen probando nuevos hiperparámetros para dar con una red que se pueda llamar óptima. No solo eso, se maneja un nuevo campo dentro de este proyecto que es la predicción, detección y cuantificación de predicciones basándose únicamente en las pérdidas de energía de la señal de radioenlaces terrestres y satelitales.

En un primer caso, está el avance en la detección de precipitaciones utilizando lotes de entrenamiento cada vez más pequeños intentando no empeorar el margen de confianza de la red resultante. Esta labor supone un reto computacional ya que la solución óptima resulta ser una idea más que una realidad. Se puede suponer que ha llegado a la mejor solución, pero nunca llega a ser realmente la mejor [15]. La complejidad del número de variables necesarias para efectuar un parte meteorológico coherente y preciso requiere de un gran número de variables que suponen en sí, un mayor gasto computacional [3].

La referencia anterior viene en mayor medida por el avance en la creación de un modelo predictivo de precipitaciones que dé una alarma fiable con el mayor tiempo de antelación posible. El caso estudiado se ha llegado a predecir con 20 minutos de antelación la aparición de las primeras lluvias con un margen de confianza de más del 85% promedio según las métricas de calidad. Esta aplicación resulta de gran relevancia ya que ese margen es suficiente para actuar ante una situación de riesgo por causas meteorológicas [2].

Capítulo 4. Sistema de captación de datos

En ese capítulo se describe los sistemas para la obtención de los datos que se han utilizado para llevar a cabo los entrenamientos de las redes neuronales y sus comprobaciones.

4.1. Introducción

En un principio, el entrenamiento de cualquier modelo, en este caso detección y predicción, es necesario primero tener una base de datos lo suficientemente amplia como para que dicho modelo sea capaz de identificar patrones y ajustar sus características internas para desempeñar la tarea por lo que ha sido diseñado [3]. En este capítulo se explica cómo se han obtenido los datos para los modelos creados que se mostrarán más adelante.

En los sucesivos experimentos, la base de datos consta principalmente de dos puntos claves para la creación de los modelos. En este caso, se pretende relacionar el factor de atenuación de una señal con un fenómeno de precipitación. Es por ello, que los principales proveedores de datos tengan que suministrar información acerca del nivel de potencia de la señal incidente respecto a la inicial emitida y el estado del tiempo de la región que ha atravesado la señal durante su recorrido desde el emisor hasta el receptor [7].

Como se acaba de mencionar, se trabajará con dos tipos de datos. Por un lado, los datos meteorológicos que son los recogidos por las estaciones meteorológicas de la AEMET. Y, por otro lado, los datos de la potencia de la señal de los radioenlaces recogidos por los equipos proporcionados por el Gobierno de Canarias, Dirección General de Telecomunicaciones, obtenidos dentro de su plan de Red de Emergencias y Seguridad de Canarias conocido como RESCAN [8].

4.2. Radioenlaces terrestres

Los radioenlaces terrestres, también conocidos como enlaces de microondas, son sistemas de comunicación inalámbrica que utilizan ondas de radio de alta frecuencia para transmitir señales de voz, datos y vídeo entre dos puntos en la Tierra. Estos enlaces se establecen mediante la instalación de antenas en lugares estratégicos que permitan una línea de visión directa entre los dos puntos, el de transmisión y el de recepción [7].

Por una parte, los radioenlaces terrestres ofrecen varias ventajas y utilidades en el ámbito de las comunicaciones. Como transmitir grandes volúmenes de datos a velocidades muy altas y establecer conexiones de comunicación confiables de larga distancia entre ubicaciones remotas que pueden estar separadas por montañas o cuerpos de agua entre

otros. Del mismo modo, los radioenlaces terrestres ofrecen flexibilidad en términos de implementación y escalabilidad ya que se pueden utilizar para establecer conexiones temporales o permanentes, y es posible aumentar la capacidad de transmisión agregando más enlaces o utilizando equipos de mayor rendimiento [7].

Por otra parte, los radioenlaces terrestres, pese a ofrecer grandes ventajas, se deben tener en cuenta algunos aspectos antes de su instalación. Un ejemplo de ello es que requieren una línea de visión directa entre las antenas transmisoras y receptoras, por lo que cualquier obstáculo físico puede afectar la calidad de la señal. Otro ejemplo sería la elección de la frecuencia de operación ya que diferentes frecuencias ofrecen diferentes características de propagación y ancho de banda siendo las frecuencias más altas, las que pueden proporcionar mayores velocidades de transmisión, pero también son más susceptibles a la atenuación por lluvia y otros fenómenos atmosféricos, en este caso, la banda SHF se ve afectada en gran medida por dichos fenómenos [8]. La capacidad del enlace, es decir, la cantidad de información que puede transmitir es otro punto a tener en cuenta ya que la comunicación entre radioenlaces terrestres se ve influenciada por la técnica de modulación utilizada. No obstante, como la base del experimento se centra en analizar los cambios de la señal inicial durante el trayecto, es indiferente la cantidad de información transmitida. Lo que sí resulta interesante, tal y como se mencionó anteriormente, es el tipo de modulación y la polarización empleada para estimar las pérdidas de una forma teórica siguiendo la Recomendación UIT-R P.838-3 [23].

Para finalizar, algo que se debe tener en cuenta con los radioenlaces terrestres es que pueden ser vulnerables a interferencias y ataques maliciosos que provoquen datos falsos a la hora de analizar la señal. Por lo tanto, es importante implementar medidas de seguridad, como la comprobación por pares y una tasa de ingresos de datos lo suficientemente buena como para descartar posibles fallos en el sistema y tener un alto grado de seguridad en la integridad de la señal transmitida. Finalmente, la instalación de radioenlaces terrestres requiere infraestructura adecuada para montar las antenas. Además, es necesario realizar un mantenimiento regular para garantizar un buen rendimiento y solucionar problemas ocasionales [4].

4.2.1. Red TETRA

La red TETRA, del inglés, Terrestrial Trunked Radio, es un estándar de comunicaciones móviles digitales diseñado con la misión de satisfacer las necesidades de las organizaciones de servicios públicos, como fuerzas de seguridad, servicios de emergencia, transporte público entre otros. Esta red proporciona una comunicación confiable y segura y se ha ampliado en todo el mundo desde su desarrollo a mediados de la década de 1990 [8].

La red TETRA fue desarrollada por el Instituto Europeo de Normas de Telecomunicaciones (ETSI) dada la necesidad de un sistema de comunicación móvil dedicado para organizaciones de servicios públicos. Su planteamiento inicial fue el de reemplazar los sistemas de radio analógicos existentes de la época con limitaciones en términos de capacidad, seguridad y funcionalidad. TETRA se introdujo inicialmente en Europa y se ha expandido a nivel mundial desde entonces [9].

La red TETRA es utilizada en una variedad de aplicaciones y sectores, incluyendo las fuerzas de seguridad y servicios de emergencia donde TETRA es ampliamente utilizado por varios servicios de seguridad y emergencia para facilitar la comunicación efectiva entre los equipos en el campo. Proporciona funciones críticas, como transmisión multimedia, (texto, audio, vídeo entre otros) y capacidades de localización. De igual modo, se utiliza en sistemas de transporte público permitiendo la comunicación entre el personal de control, y otros empleados lo que aumenta la eficiencia a la hora de coordinar las operaciones de los servicios de transporte. Otro ejemplo sería la implementación de la red TETRA en empresas que operan infraestructuras críticas como, por ejemplo, plantas de energía o aeropuertos [25].

La red TETRA ofrece una capacidad de canales significativamente mayor en comparación con los sistemas de radio analógicos, lo que permite el soporte de múltiples usuarios y grupos de usuarios simultáneamente con seguridad. Proporciona autenticación de usuarios, cifrado de voz y mensajes, y medidas de protección contra escuchas y ataques cibernéticos. También está diseñada para ser interoperable, lo que significa que los dispositivos y sistemas de diferentes fabricantes pueden comunicarse entre sí sin problemas facilitando la colaboración entre diferentes organizaciones y servicios de

emergencia durante las operaciones conjuntas [25]. Por medio de una comunicación a través de la red TETRA, se puede llegar a establecer un canal dúplex o semi-dúplex con una tasa binaria máxima de 36 kbps y una modulación $\pi/4$ -DPSK permitiendo la interconexión con redes públicas, como son las redes ISDN (Integrated Services Digital Network) y PSTN (Public switched telephone network) [9].

En la capa física de la red TETRA se recoge en el Cuadro Nacional de Atribución de Frecuencias (CNAF) con tres bandas de trabajo, estas están dedicadas a servicios de seguridad y emergencia (380/385 – 390/395 MHz) y a servicios privados o terceros (410/470 MHz y 870/888 – 915/933 MHz) [25]. En la red TETRA se utilizan técnicas de multiplexación en el tiempo las llamadas Time-Division Multiple Access (TDMA), donde cada equipo es asignado con un tiempo determinado con acceso finito para comunicarse. Además, hace uso de la modulación por desplazamiento de fase en cuadratura de polarización dual (de sus siglas en inglés, DQPSK) con un canal de radio de 25 kHz [25].

En la capa de enlace, los datos se organizan en dos tramas siendo dividida a su vez en dos niveles. El primer nivel de Link de Control Lógico (de sus siglas en inglés, LLC) que se ocupa de la transmisión y retransmisión de datos. El segundo nivel de Control de Acceso al Medio (de sus siglas en inglés, MAC) con la función de controlar el acceso a los canales y su codificación y decodificación al igual que encargarse de otras tareas como el interleaving, routing y la multiplexación [25].

En la capa red se dividen entre un plano de usuario que se encarga de gestionar el contenido que envían los usuarios y un plano de control que gestiona la señalización, los datos de control al igual que tiene la función de control de procedimientos de la red [25].

En la arquitectura de una red TETRA, como se muestra en la Figura 8, cada celda tiene su frecuencia (o conjunto de ellas) y cada usuario accede a la celda que le proporciona un mejor servicio, siendo traspasado automáticamente a la celda que mejore su condición de servicio. Esta red basada en nodos de conmutación junto a las estaciones base reduce los tiempos de espera y garantiza la calidad de transmisión de información ya que se comparte de manera automatizada los recursos del sistema de una manera organizada [25].

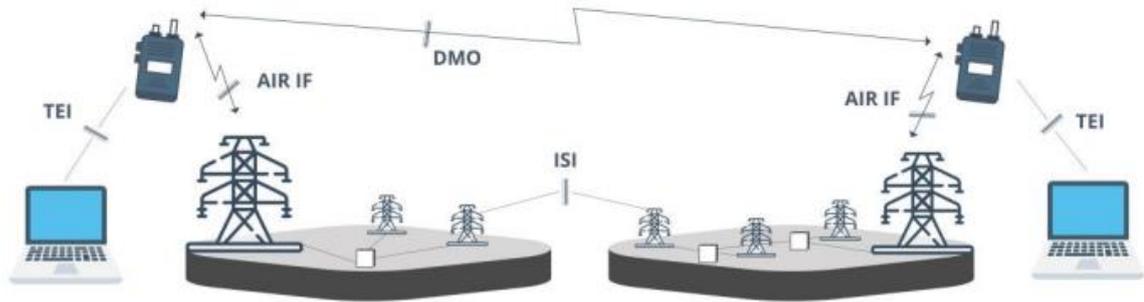


Figura 8. Arquitectura TETRA.
Fuente: INCIBE, 2023 [25].

El funcionamiento de la red se muestra en la Figura 9 y se inicia cuando una estación móvil se activa dentro de una zona de cobertura de una estación base (de sus siglas en inglés, BS), primero busca un canal de control y se registra en la red. Segundo, la BS se encarga de enviar información a la estación móvil cuando tiene permitido el acceso. La permisividad de una estación móvil va en función del orden de prioridad que tenga su comunicación, dando la prioridad más alta y, por tanto, mayor velocidad para acceder a aquellos servicios o estaciones móviles que estén registrados como más urgentes [25].

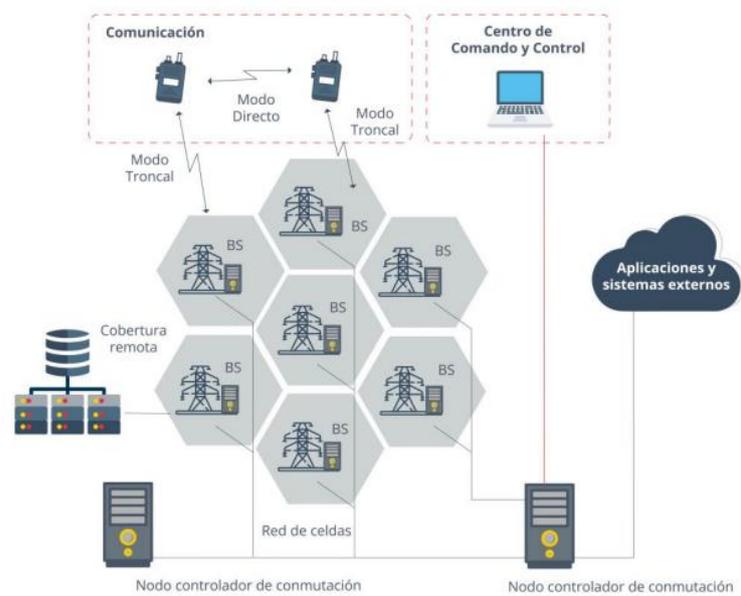


Figura 9. Funcionamiento TETRA.
Fuente: INCIBE, 2023 [25].

En tercer lugar, si el acceso es permitido, la estación móvil hace un intento de contacto mediante la técnica de acceso Aloha Ranurada y la estación base remite un mensaje con las especificaciones para reservar el canal. Finalmente, en cuarto lugar, se intercambia información para adecuar la potencia de transmisión a la estación móvil y transmitir los datos [25].

4.2.2. RESCAN

La red RESCAN es una propuesta por parte del Gobierno de Canarias para el desarrollo de las Tecnologías de la Información y la Comunicación, TIC. Esta propuesta nace como una acción para afrontar problemas de seguridad o emergencias en las Islas Canarias. Habiendo invertido más de 11 millones de euros en la instalación de esta red, la idea es ponerla a disposición de todas las administraciones locales y servicios de emergencias. RESCAN emplea una red TETRA para comunicar los avisos de emergencia a los usuarios objetivos y equipados con dispositivos habilitados [24]. La información a través de esta red genera datos cada 5 minutos por lo que se obtienen 12 muestras por hora bajo los parámetros de transmisión que se muestran en la Tabla 3 [8].

Tabla 3. Datos recibidos del radioenlace.
Fuente: Elaboración propia.

Valor	Descripción	Unidades
RSL	Nivel de potencia recibida	dBm
PwrDetTemperature	Temperatura en equipo transmisor	Celcius
IF Temperature	Temperatura del circuito de frecuencia intermedia en el equipo transmisor (debe tener el mismo valor que <i>PwrDetTemperature</i>)	Celcius
Forward Power	Potencia de transmisión	dBm
ODU -48V supply	Tensión de alimentación del ODU	Voltios
ATPC Tx Power	Control de potencia adaptativo en el transmisor	dBm
Rx Temperatura	Temperatura en el equipo receptor	Celcius
SNR	Relación señal a ruido	dB
ATPC Fade Margin	Margen de desvanecimiento en el control automático de potencia	dB

4.3. Enlace satelital

Los enlaces entre estaciones terrenas y satelitales o entre dos estaciones satelitales son similares, en parte, a los enlaces entre estaciones terrestres. No obstante, en este caso, la diferencia central entre los radioenlaces terrestres se encuentra en la dirección de transmisión, es decir, el sentido de la comunicación, dónde se pueden apreciar tres tipos de enlaces, el enlace de subida (desde la estación terrestre a la estación satelital), el enlace de bajada (desde la estación satelital a la estación terrestre) y enlace inter-satelital (desde la estación satelital a otra estación satelital) [7]. Los principales parámetros para el diseño de radioenlace satelital son los siguientes que se muestran en la Tabla 4 [8].

*Tabla 4. Parámetros de diseño de un radioenlace satelital.
Fuente: Elaboración propia.*

PARÁMETRO	UNIDAD
Inclinación	°
Distancia del vano	Km
Frecuencia RF	Hz
Díámetro de la antena parabólica receptora	M
Eficiencia estimada	%
Longitud de onda	M
Pérdidas de espacio libre	%
Atenuación debida a la absorción y dispersión por hidrometeoros	dB
Potencia transmitida y potencia recibida	dBm

El operador de comunicación usado para los experimentos es HISPASAT. Este operador dota de cobertura en toda Europa, América y norte de África. Poseen más de 25 años de experiencia y forma una fuente de sólida como proveedor de datos para los posteriores experimentos. La flota de HISPASAT cuenta con 8 satélites, tal y como muestra la Figura 10, capaces de ofrecer más de 1250 canales de televisión y radio. Por un lado, la sociedad HISPASAT, se compone de una serie de sociedades dependientes como son HISPASAT Canarias, HISPAMAR y las sociedades HISDESAT Servicios Estratégicos y Galileo Sistemas y Servicios. Por otro lado, la compañía cuenta con presencia en el sector público español Sociedad Estatal de Participaciones Industriales

Capítulo 4. SISTEMA DE CAPTACIÓN DE DATOS

(SEPI), Centro para el Desarrollo Tecnológico e Industrial (CDTI) y Red Eléctrica como se ve en el gráfico de accionistas principales de la Figura 11 [26].

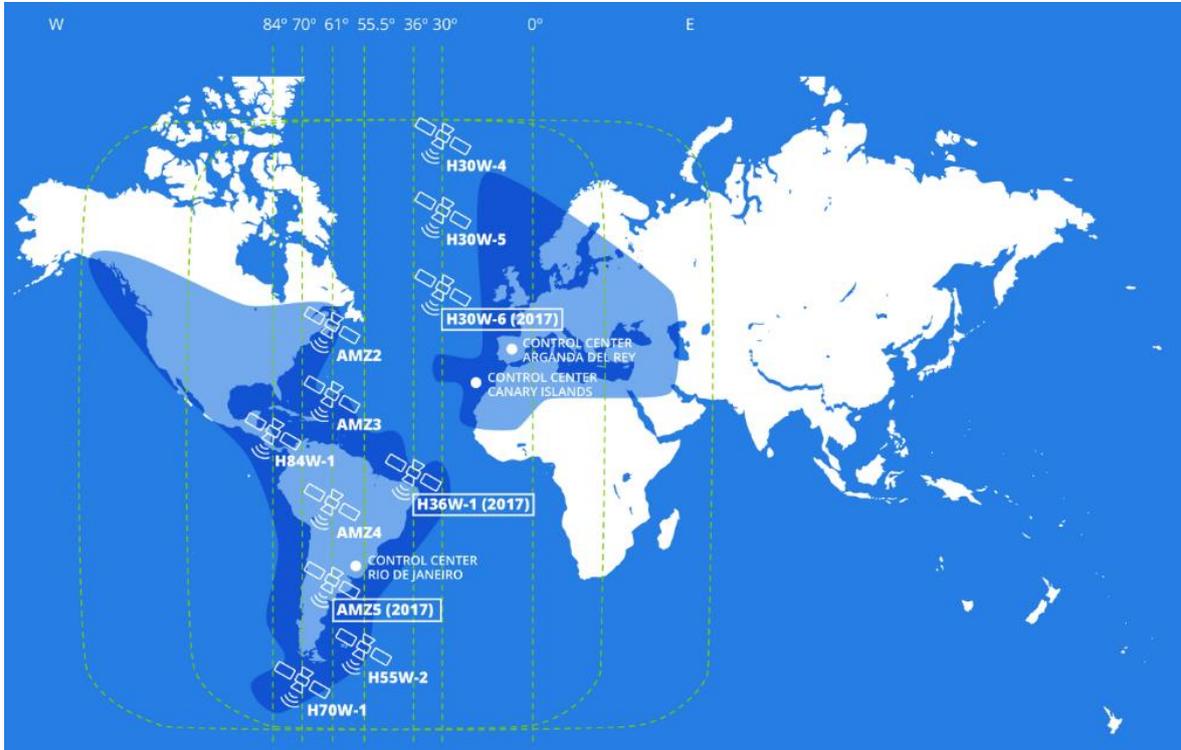


Figura 10. Flota HISPASAT y área de cobertura.
Fuente: HISPASAT,2023 [26].

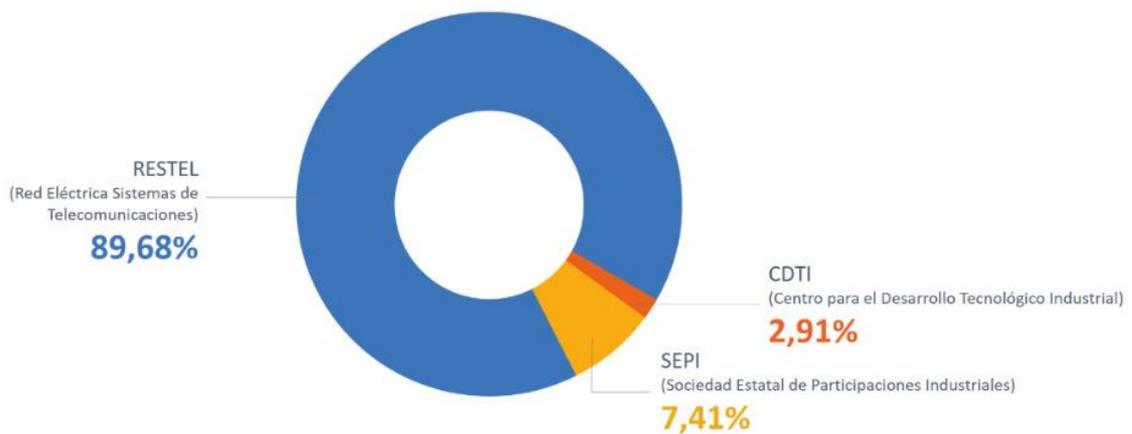


Figura 11. Accionistas principales de HISPASAT.
Fuente: HISPASAT,2023 [26].

4.3.1. Prototipo satelital

Los instrumentos utilizados para la obtención de datos por parte de cada estación meteorológica varía ya que se usan distintos modelos respecto a los aparatos de medición. No obstante, todas poseen redes de sensores de temperatura, humedad relativa, anemómetros, veletas y pluviómetro [5].

En la azotea del Pabellón B del Edificio de Electrónica y Telecomunicación de la ULPGC, se encuentra instalada una antena parabólica orientada hacia el satélite HISPASAR 30W-6. Junto a esta antena, se dispone de una estación meteorológica de marca VAISALA. Dicha estación se compone de un pluviómetro Vaisala Rain Gauge RG13 (ver Figura 12) que utiliza un mecanismo de balancín para producir un cierre de contacto cuando se recibe una cantidad predeterminada de lluvia a elegir entre 0.1mm, 0.2mm y 0.5mm [27]. El mecanismo se basa en un cubo dividido pivotado respecto a su centro dónde la lluvia se almacena en su mitad superior. Una vez lleno, el mecanismo se inclina y descarga el agua almacenada, lo que permite a la otra mitad del cubo comenzar a llenarse de manera similar a lo que sucede con el “Shishi-Odoshi”, un sistema que se ve tradicionalmente en jardines japoneses como un elemento decorativo y para asustar a los animales no deseados cuyo sonido del golpe y el movimiento rítmico del tubo de bambú añaden un elemento tranquilo y meditativo al entorno del jardín [28].



*Figura 12. Vaisala Rain Gauge RG13.
Fuente: VAISALA, 2023 [27].*

Otro dispositivo con el que se encuentra equipado el sistema son sensores de temperatura PT 100. Estos sensores son un detector de temperatura por resistencia (de sus siglas en inglés, RTD) fabricados con platino, un material con una resistencia eléctrica de 100 ohmios a una temperatura de 0 °C, este material cambia de resistencia cuando se cambia la temperatura sensiblemente, lo que sirve para determinar la temperatura a la que se encuentra sabiendo la resistencia que produce al paso de una corriente eléctrica. De igual modo, la estación posee un sensor de humedad relativa (de sus siglas en inglés, RHS) mide la humedad relativa en el rango de 0 a 100% y una resolución de 0,1% de humedad relativa. Continuando con los instrumentos de medida, está el anemómetro compuesto por 3 cazoletas cónicas sobre un eje de acero inoxidable. El transductor del anemómetro consiste en un led y un fototransistor que proporciona un pulso con una frecuencia proporcional a la velocidad del viento con una precisión de 0.1 m/s. Siguiendo con las medidas del viento, se encuentra una veleta de aluminio unida a un eje con un imán diametral que actúa sobre un sensor proporcionando una salida equivalente a la dirección indicada por la veleta, es decir, los grados de diferencia respecto al norte magnético [7]. La instalación final de la estación meteorológica y la antena se puede ver en la Figura 13.



*Figura 13. Estación meteorológica y antena.
Fuente: Elaboración propia.*

El punto 3 de la Figura 13 hace referencia a una caja de registro que contiene dos Raspberry Pi que establecen una conexión vía Ethernet a un switch dentro de la misma caja de registro. El equipo de mediciones meteorológicas se conectan a una de las dos unidades Raspberry Pi mientras que la antena parabólica se conecta a la otra. La información es enviada al Laboratorio 202 del Pabellón B que es donde se encuentra el servidor VIMETRI-MAC que almacena los datos como muestra el diagrama de la Figura 14 [8].

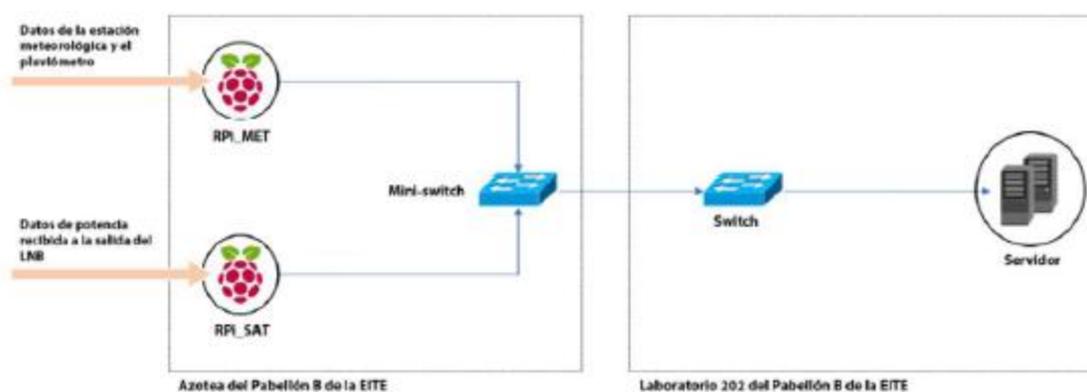


Figura 14. Conexión del sistema de captación y transmisión de datos.
Fuente: Elaboración propia.

4.4. AEMET

La AEMET es el organismo oficial encargado de la predicción y observación del tiempo y el clima en España. Es una agencia gubernamental adscrita al Ministerio para la Transición Ecológica y el Reto Demográfico de España. La AEMET tiene como objetivo proporcionar información meteorológica y climatológica precisa y actualizada a los diversos sectores del país. Entre sus funciones incluyen la observación y registro de datos meteorológicos, la emisión de pronósticos del tiempo, alertas meteorológicas, estudios climáticos, la elaboración de informes y boletines. Del mismo modo, avanza sobre asuntos como la investigación y desarrollo en el campo de la meteorología y la climatología [5].

La AEMET opera una red de estaciones meteorológicas distribuidas por los distintos sectores de España. Además, utiliza datos de satélites, radares y modelos numéricos para realizar pronósticos. Su trabajo es usado para proteger la población frente a

condiciones atmosféricas que puedan evolucionar en fenómenos peligrosos, así como apoyar la toma de decisiones en sectores como la agricultura, la aviación, la navegación marítima, la gestión del agua y la energía, entre otros [5].

Durante el estudio realizado, los datos meteorológicos que han sido usados para comprobar el estado del tiempo durante la transmisión han sido proporcionados a través del portal web de la AEMET [8]. Los datos requeridos sólo son accesibles por medio de dicho portal web que no ofrece la opción de descargar un archivo con el parte meteorológico, obligando en un primer momento a tener que estar copiando y pegando la información de manera manual. Para facilitar esta acción se ha utilizado una técnica de Web scraping, este es un algoritmo que simula el uso del navegador encargándose de visitar la página, localizar los datos y remitirlos hacia el servidor del VIMETRI-MAC donde se almacenan en forma de archivo de texto para su posterior procesado y uso [9].

La técnica de Web scraping utilizada analiza sintácticamente el código fuente de la página web objetivo. Por ejemplo, en caso de querer obtener los datos de las últimas 24 horas de la estación meteorológica de Tafira solo se tendría que acceder al portal web de la AEMET y buscar los datos con los parámetros antes mencionados [3]. Los datos que se pueden obtener de las estaciones se presentan en la Tabla 5.

*Tabla 5. Datos proporcionados por la AEMET.
Fuente: Elaboración propia.*

Parámetro	Unidad
Fecha y hora	aaaa-mm-dd hh:mm:ss
Dirección del viento	°, respecto al norte magnético
Velocidad del viento	km/h
Dirección de máxima velocidad del viento	°, respecto al norte magnético
Temperatura	Celcius
Humedad relativa	%
Presión atmosférica	hPa
Precipitación	mm
Precipitación volumétrica	l

Pese a tener la opción de acceder a una variada serie de otros parámetros, los datos requeridos para la investigación posterior son la fecha y hora de la medición y precipitación medida en ese instante de tiempo. Los datos recogidos en la base de datos se clasifican en función de la región de la que procedan, por ejemplo, una de estas estaciones se localiza en la Urbanización Zurbarán, próxima al Campus de Tafira [8].

4.5. Importación de datos

4.5.1. Estaciones AEMET

Las estaciones AEMET brindan un documento que registra las mediciones bajo un periodo muestral de diez minutos que se almacenan en un documento en formato csv. El vano utilizado principalmente es el vano Hospital Militar – Pico I (Pico DGTNT), distribuido sobre Gran Canaria como se muestra en la Figura 15 [8].

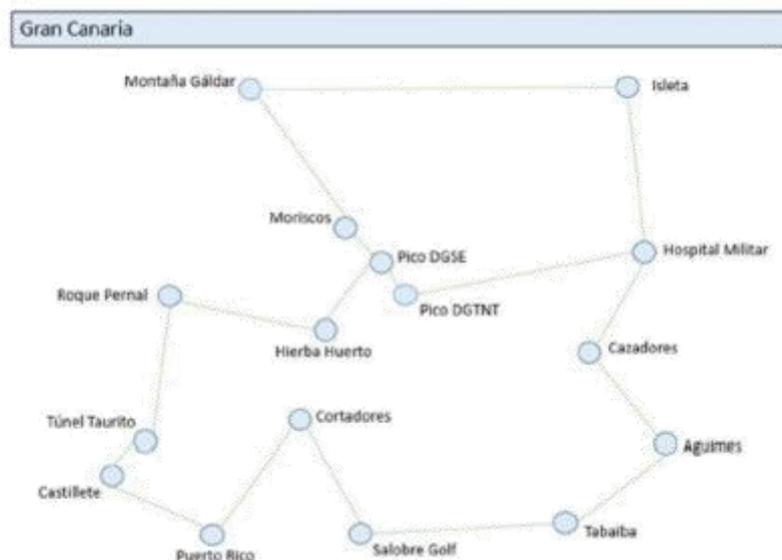
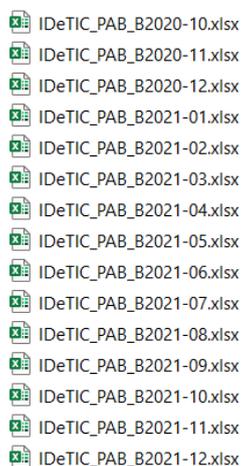


Figura 15. Distribución de los vanos en la isla de Gran Canaria.
Fuente: Tacoronte, Cristian, 2022 [8].

Tal y como se ha comentado en el texto anterior, los archivos se obtienen en formato csv dónde se guardan las mediciones por vano de manera continuada. Es por ello, que se requiere cambiar a un formato xlsx para su posterior uso y clasificarlo en función de los meses en los que se han realizado las mediciones para un más fácil manejo en los experimentos posteriores [8]. Para ello, se ha diseñado con anterioridad un programa en MATLAB R2022b cuya función es la de cumplir con dicha tarea dejando como resultado

Capítulo 4. SISTEMA DE CAPTACIÓN DE DATOS

un conjunto de archivos como los que se muestra en la Figura 16 [8]. En este caso, el conjunto de datos pertenece a la estación meteorológica ubicada en San Mateo, Corral de los Juncos.



*Figura 16. Ejemplo de adaptación de archivos en formato csv a xls.
Fuente: Elaboración propia.*

La investigación dentro de este proyecto requiere que la importación de los datos se divida en dos partes, en un lado, se tienen los datos de los niveles de radioenlaces terrestres y en otro lado los datos referentes a mediciones de condiciones meteorológicas. Al tener dos bases de datos separadas, es necesario crear una correlación entre ambas seleccionando aquellos meses en los que coinciden ambos conjuntos de datos [16]. Para que sea fácilmente comprensible, si se desea, por ejemplo, incluir los datos de un mes en la base de datos final para crear los modelos de detección y predicción, es necesario que dicho mes se encuentre disponible en el conjunto de datos del radioenlace terrestre y a su vez en la estación meteorológica seleccionada. Para corroborar que la base de datos se compone de medidas ambientales junto a los niveles de potencia de los radioenlaces en un mismo tiempo, se crea un programa, función `GeneradorMesesf.m` (Anexo 1), encargado de buscar aquellos meses en los que se cuenta con ambos tipos de datos generando al final un vector con los meses comunes [9].

La función `GeneradorMesesf.m` toma los valores de entrada del vano a estudiar y del tipo de sistema operativo que se emplea. Luego el programa sigue la ruta de las carpetas hasta llegar a la base de datos que debe comparar y comprueba que existen datos tanto de potencia de los radioenlaces por un lado como valores de medidas meteorológicas por el otro. Si se da dicha

condición de guarda el mes en un vector que contiene los meses de datos que coinciden por ambas partes. Finalmente, tras su ejecución, los meses que cumplen la condición son: 11-2020, 12-2020, 01-2021, 02-2021, 03-2021, 07-2021, 08-2021, 09-2021, 11-2021, 12-2021 [9].

4.5.2. Estación propia

Los datos que se obtienen por medio de la estación meteorológica situada en la azotea del Pabellón B se pueden obtener por medio de una aplicación web ya creada con anterioridad (Figura 17). En dicha aplicación se selecciona un intervalo de tiempo y se obtiene un archivo csv con los datos adquiridos, luego se pasa a formato xlsx bajo un nombre denominado como “IDeTIC_PAB_B” + año + “-“ + mes + “_Estacion” [8].

En la Figura 18 se puede observar un ejemplo del resultado final de la obtención de los datos de la estación meteorológica del Pabellón B transcritas en formato xlsx. En la figura se puede apreciar que la frecuencia de muestreo es de 15 segundos y proporciona información sobre la dirección y velocidad del viento, la temperatura del entorno, el porcentaje de humedad relativa, la presión, una columna que marca si hay registrado un evento de lluvia con una cantidad mayor que cero o si no ha detectado nada (valor cero). Si suceden las precipitaciones, guarda también el volumen registrado, si no, guarda valor cero ya que no ha podido registrar nada [11].



Figura 17. Aplicación web para obtener los datos de la estación del Pabellón B.
Fuente: Elaboración propia.

Capítulo 4. SISTEMA DE CAPTACIÓN DE DATOS

1	Time	WDir	WV	Temperature	Humidity	Pressure	Precipitation	VolumetricPrecipitation
2	2021-07-01 00:00:10	35	6,7	17,6	75,7	981,7	0	0
3	2021-07-01 00:00:25	3	13,5	17,6	75,8	981,7	0	0
4	2021-07-01 00:00:40	342	9,6	17,6	75,8	981,7	0	0
5	2021-07-01 00:00:55	1	10,5	17,6	75,8	981,7	0	0
6	2021-07-01 00:01:10	8	10,6	17,6	75,9	981,7	0	0
7	2021-07-01 00:01:25	331	10,3	17,6	75,9	981,7	0	0
8	2021-07-01 00:01:40	340	6,6	17,6	75,9	981,7	0	0
9	2021-07-01 00:01:55	11	3,8	17,6	76	981,7	0	0
10	2021-07-01 00:02:10	108	3,8	17,6	76	981,7	0	0
11	2021-07-01 00:02:25	354	15,6	17,6	76	981,6	0	0
12	2021-07-01 00:02:40	3	5,5	17,6	76	981,7	0	0

Figura 18. Ejemplo de los datos meteorológicos recogidos en el Pabellón B.

Fuente: Elaboración propia.

En meteorología se suele marcar un evento de lluvia cuando se registran niveles de lluvia suficientemente alto como, por ejemplo, mayores de 0'2 mm/h. Sin embargo, cuando el nivel es bajo se puede considerar como rocío o agua que ha arrastrado el viento haciendo caer las gotas en horizontal sin llegar a considerarse como lluvia [5]. En la Figura 19 se puede ver con claridad este caso dónde se han registrado precipitaciones sin que cuenten como un evento de lluvia ya que su valor volumétrico es cero.

	A	B	C	D	E	F	G	H
1	Time	WDir	WV	Temperature	Humidity	Pressure	Precipitatio	VolumetricPrecipitation
157125	2020-08-30 01:00:49	216	9	19,8	75,1	974,9	0,04	0
158418	2020-08-30 06:24:04	161	5,1	18,7	82	973,3	0,01	0
158554	2020-08-30 06:58:04	88	6,6	18	84,4	973,5	0,01	0

Figura 19. Evento de lluvia no nulo.

Fuente: Elaboración propia.

4.5.3. Estación satelital

La obtención de los datos procedentes del radioenlace satelital siguen la misma metodología comentada anteriormente, es decir, se obtienen del mismo modo de una aplicación web, como se muestra en la Figura 20, y se traspasan a formato xlsx. En este caso, bajo el patrón de nombre "IDeTIC_PAB_B" + año + "-" + mes " (Figura 21). Estos datos, tal y como se representan en la Figura 22, tienen un periodo de muestro sobre los catorce o dieciséis segundos siendo la columna E la interesante para los experimentos ya que representan la intensidad de la señal incidente sobre la antena parabólica de la azotea del Pabellón B [8].



Figura 20. Aplicación web para obtener los datos de la estación satelital.
Fuente: Elaboración propia.

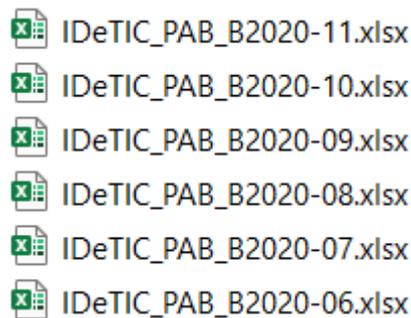


Figura 21. Archivos xlsx del radioenlace satelital.
Fuente: Elaboración propia.

	A	B	C	D	E
1	Time	FreqFI	FreqRF	DesFI	RSL
2	2020-11-01 00:00:05	1402,412434	11152,41243	412,43	-43,93
3	2020-11-01 00:00:21	1402,399936	11152,39994	399,94	-43,73
4	2020-11-01 00:00:36	1402,237462	11152,23746	237,46	-44
5	2020-11-01 00:00:52	1402,383272	11152,38327	383,27	-43,91
6	2020-11-01 00:01:07	1402,70822	11152,70822	708,22	-44,03
7	2020-11-01 00:01:23	1402,487422	11152,48742	487,42	-43,99
8	2020-11-01 00:01:38	1402,074988	11152,07499	74,99	-43,89
9	2020-11-01 00:01:53	1402,183304	11152,1833	183,3	-43,96

Figura 22. Ejemplo de los datos satelitales obtenidos.
Fuente: Elaboración propia.

La importación de los datos procedentes del enlace satelital se realiza por medio del script `Sincronismo_Satelital_Mes_COMBINADO6.m` (Anexo 2). Este programa lo primero especifica los nombres de los archivos con el patrón de nombre que se mencionó anteriormente [7].

En este caso, se tienen niveles de precipitaciones y de potencia desde el mes de agosto de 2019 hasta el mismo mes del 2021. Luego, se establecen las rutas de los directorios para localizar los archivos y se importan los datos con extensión `.mat` [8].

La función de usar esta extensión es para cargar los datos en el entorno de programación directamente en lugar de, más adelante, hacer que el programa buque el valor entre todos los elementos de la base de datos. Por lo tanto, esta decisión se toma para agilizar el proceso de futuros experimentos ya que los datos bajo la extensión `.mat` están optimizados para el trabajo dentro del entorno MATLAB [3].

Una vez terminada esta sección se ha logrado completar tanto el tercer objetivo O3. Gestionar el sistema VIMETRI respecto a datos de hidrometeoros, como el cuarto objetivo O4. Gestionar conjuntos de datos de radioenlaces y estaciones meteorológicas. De esta forma, se considera que los datos generados tanto por radioenlaces terrestres como radioenlaces satelitales sean capaces de introducirse dentro del sistema del proyecto para generar una base de datos con la que se pueda experimentar posteriormente.

Capítulo 5. Procesado de datos

En este capítulo se explican las transformaciones que sufren los datos de entrada y proceso que siguen hasta convertirse en datos útiles que faciliten el entrenamiento y comprobación de los posteriores experimentos.

5.1. Introducción

Los datos que se han ido obteniendo en pasos anteriores deben ser tratados en primer lugar antes de que puedan convertirse en datos relevantes para el entrenamiento de los modelos de detección y predicción de precipitaciones [3].

Es por lo comentado anteriormente, que los datos originales necesitan pasar primero por una secuencia de transformaciones para que pasen a ser datos útiles. Por ello, a continuación, se describirá la secuencia que han seguido los datos originales para que, una vez alcanzado su etapa final en el modelo de Machine Learning, presenten la estructura adecuada que facilite su interpretación buscando mejorar las prestaciones de la red de detección y predicción [11].

Por un lado, se detallarán los aspectos de la primera fase en dónde se referencia la estructura empleada durante el procesado de datos que provienen de la estación meteorológica y el prototipo satelital VIMETRI-MAC. Por otro lado, se explicará de igual modo la estructura empleada para los datos obtenidos por parte de las estaciones meteorológicas de la AEMET, especialmente, el radioenlace terrestre al que pertenece el vano Hospital Militar – Pico I [8].

5.2. Base de datos de la red satelital

Los datos que se recogen por parte de la red satelital y que llegan a la estación situada en el Pabellón B, siguen una secuencia de transformaciones que se muestra en la Figura 23. El diagrama de flujo presentado marca las pautas y los pasos por los que tiene que pasar los datos de entrada para generar una base de datos confiable ya que pueden ocasionarse errores por parte de la toma de datos dónde se puedan perder muestras por error o mal funcionamiento del sistema o cualquier otro error [13].

En términos generales, el diagrama se compone de una primera fase encargada de leer los datos de entrada, luego, comprueba si los datos se encuentran sincronizados para continuar haciendo un procesado de la señal que calcule el valor óptimo para definir lo que es un evento de lluvia y no un error por fallos en el sistema. Finalmente, el esquema acaba creando los datos necesarios para el posterior entrenamiento del modelo [8].

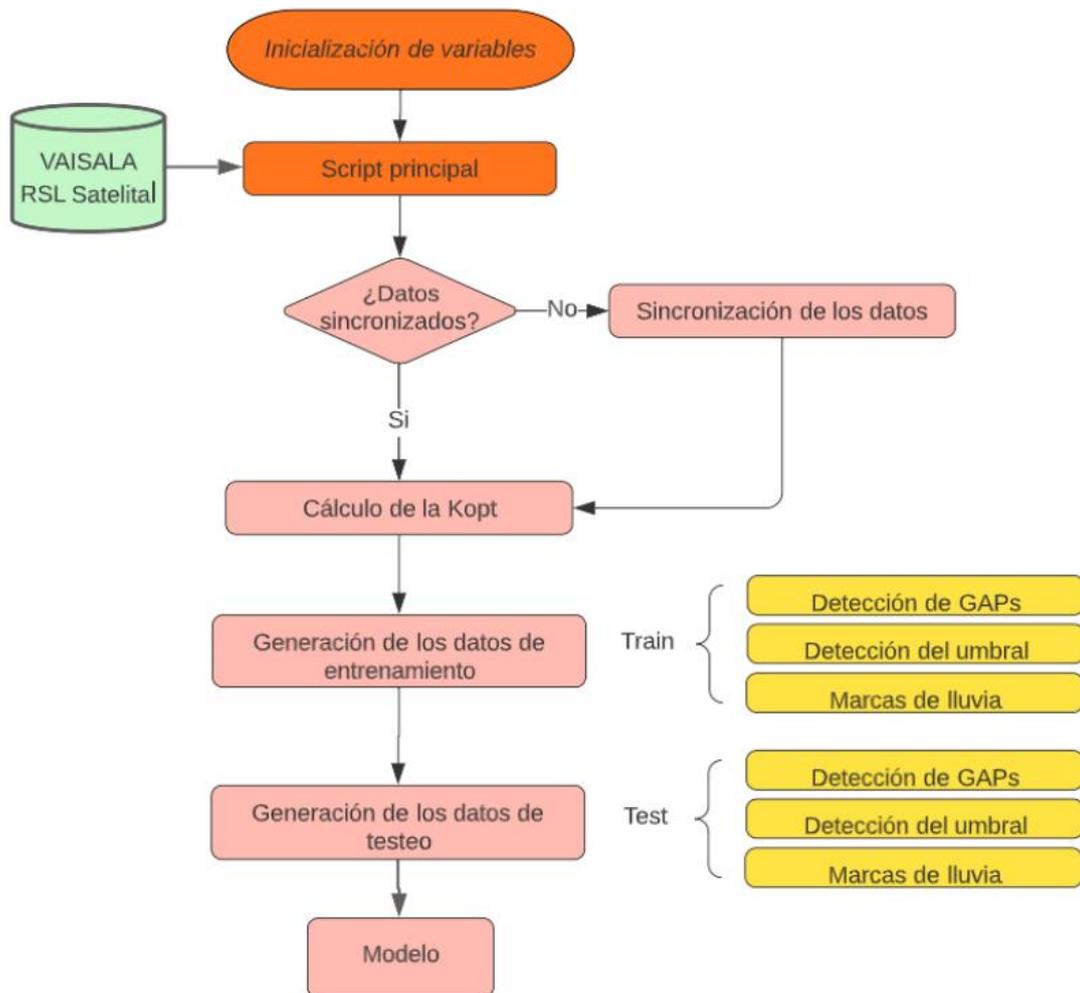


Figura 23. Diagrama de flujo del procesado de datos del Pabellón B
Fuente: Tacoronte, Cristian, 2022 [8].

5.2.1. Sincronismo

Comenzando con la primera parte del diagrama anterior, el inicio transcurre una vez los datos de entrada son importados. Estos datos son el compendio de muestras de los sistemas de recolección de datos clasificados por meses. Por lo tanto, el siguiente paso del diagrama es la comprobación de la sincronización de los datos [7].

El sincronismo buscado hace referencia a una coincidencia temporal entre las muestras de datos meteorológicos tomadas y los niveles de potencia de los radioenlaces satelitales. La probabilidad de que una muestra de potencia recibida coincida perfectamente con una muestra meteorológica es casi nula [3]. No obstante, las condiciones meteorológicas, aunque fluctúen con facilidad y tengan la apariencia de ser

CAPÍTULO 5. PROCESADO DE DATOS

impredecibles, también tienen la característica de que son lentas, es decir, el tiempo no cambia de un segundo a otro, por norma general suele mantenerse como mínimo, un breve periodo de tiempo antes de cambiar [5].

Pese a que los datos de la estación meteorológica y los del prototipo satelital están muestreados con el mismo periodo, suceden en instantes diferentes. Por suerte esa diferencia no es abismal, en realidad, es relativamente pequeña, del orden de diez segundos como mucho. Dado el caso, es necesario hacer que las muestras de ambas fuentes de información coincidan para realizar el etiquetado de las señales con sus respectivos fenómenos de lluvia o no lluvia. Como se acaba de comentar esta falta de sincronismo no resulta excesivamente grave, dado el bajo retardo entre muestras. No obstante, se deben sincronizar igualmente y correlacionarlas bajo un marco que defina los límites de hasta dónde termina el tiempo de una muestra y comienza la otra. Finalmente, se toma la decisión de ajustar temporalmente ambas secuencias con un marco de quince segundos como periodo de entrada entre una muestra de nivel de señal y otra meteorológica [3].

El marco impuesto de quince segundos se debe, a como se ve en la Figura 24, que tras analizarse ambas bases de datos se concluye que el tiempo promedio de registro de datos es de quince segundos y, por lo tanto, la diferencia temporal entre un par de muestras no debe aumentar dicho orden para ser asignadas correctamente [8]. La Figura 24 muestra dos ficheros xlsx, a la izquierda, los datos de potencia de la estación satelital y a la derecha, los datos de la estación meteorológica. Ambos ficheros corresponden a la misma fecha, en este caso, noviembre de 2020.

	A	B	C	D	E		A	B	C	D	E	F	G	H	
1	Time	FreqFI	FreqRF	DesFI	RSL		1	Time	WDir	WV	Temperature	Humidity	Pressure	Precipitation	VolumetricPrecipitation
2	2020-11-01 00:00:05	1402,412434	11152,41243	412,43	-43,93		2	2020-11-01 00:00:14	264	4,9	22	24,8	984	0	0
3	2020-11-01 00:00:21	1402,399936	11152,39994	399,94	-43,73		3	2020-11-01 00:00:29	277	5,3	22	24,8	984	0	0
4	2020-11-01 00:00:36	1402,237462	11152,23746	237,46	-44		4	2020-11-01 00:00:44	268	6,5	22	24,9	984	0	0
5	2020-11-01 00:00:52	1402,383272	11152,38327	383,27	-43,91		5	2020-11-01 00:00:59	283	6,7	22	24,8	984	0	0
6	2020-11-01 00:01:07	1402,70822	11152,70822	708,22	-44,03		6	2020-11-01 00:01:14	288	6,7	22,1	24,7	984	0	0
7	2020-11-01 00:01:23	1402,487422	11152,48742	487,42	-43,99		7	2020-11-01 00:01:29	280	6	22,1	24,5	984	0	0
8	2020-11-01 00:01:38	1402,074988	11152,07499	74,99	-43,89		8	2020-11-01 00:01:44	272	5,1	22,1	24,4	984	0	0
9	2020-11-01 00:01:53	1402,183304	11152,1833	183,3	-43,96		9	2020-11-01 00:01:59	267	7,8	22,1	24,3	984	0	0
10	2020-11-01 00:02:09	1402,570742	11152,57074	570,74	-43,94		10	2020-11-01 00:02:14	270	4,7	22,1	24,4	984,1	0	0
11	2020-11-01 00:02:24	1402,64573	11152,64573	645,73	-44,01		11	2020-11-01 00:02:29	269	5,1	22,2	24,4	984,1	0	0

Figura 24. Comparación entre muestras de potencia y meteorológicas.

Fuente: Elaboración propia.

5.2.2. Obtención del umbral óptimo de detección

El siguiente paso del camino de los datos durante el procesado prosigue por la clasificación de las muestras como eventos de lluvia o eventos de no lluvia. Para esta acción se aplica un procesado de señal, particularmente, un procesado de detección de umbrales. El objetivo final de este punto es determinar el valor del umbral para que, en caso de ser sobrepasado se pueda considerar correctamente la muestra como evento de lluvia. De manera teórica, este umbral debe ser superado durante una serie consecutiva de muestras ya que no podemos clasificar correctamente como lluvia una única muestra. Se puede considerar como precipitación como un posible crecimiento de la precipitación hasta que supere el umbral, luego se mantenga un cierto tiempo y luego finalizar el evento de lluvia cuando comience a decaer varias muestras después. En cierta forma, el tamaño de la ventana determina cuándo se aproxima un evento de lluvia y cuando esta por acabar [7].

El umbral escogido se trata del tipo adaptativo dónde teniendo en cuenta los valores que se encuentran dentro de una ventana, cuya longitud es fijada inicialmente, se cogen varios datos de muestras anteriores y promedia mientras guarda el resultado en el instante de estudio [3]. Este umbral se ha llamado k_{optimo} y su valor se determina mes a mes. De esta forma, se evitan los problemas relacionados con los cambios cíclicos de las estaciones [3]. No obstante, es posible modificar el tiempo con el que se renueva esta variable por si se estimara oportuno en un futuro realizar experimentos calculando su valor cada tres meses, cada año, o el margen que se desee establecer. Este proceso se realiza a través del script `Obtención_koptValue_Mes_RSLd.m` que se adjunta en el Anexo 3.

El procedimiento del código anterior que reflejado en la Figura 25 dónde se facilita un diagrama de flujo. El primer paso inicia con un vector que almacena los valores k , siendo k los valores que establecen el umbral óptimo. Luego, a través de un bucle se calculan unos parámetros de calidad para comprobar el mejor valor para el periodo [7]. En el estudio realizado, se ha decidido que el vector de umbral empiece en 0.95 y finalice en 3, con una escala de aumento de 0.05. Por tanto, este proceso se repite 42 veces.

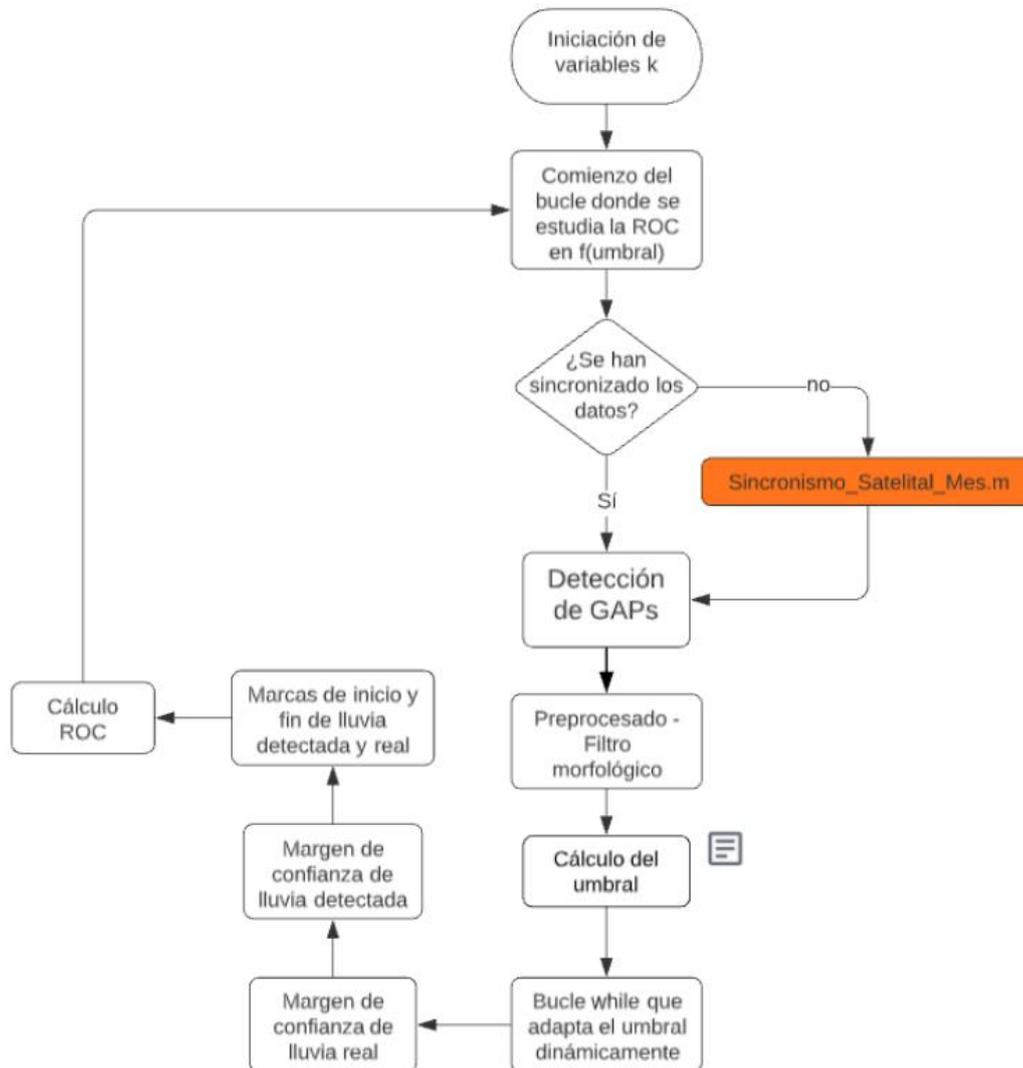


Figura 25. Diagrama de flujo para calcular el mejor umbral de detección de lluvia.
Fuente: Tacoronte, Cristian, 2022 [8].

5.2.3. Detección y corrección de GAPs

Siguiendo el diagrama de la Figura 25, el siguiente paso identifica los GAPs. Los GAPs hacen referencia a la ausencia de valores en el conjunto de datos provocados por fallos en el sistema, interferencias, entre otros. Para identificar los GAPs se pueden encontrar, por una parte, es la estación satelital en aquellas muestras cuyo valor de potencia de señal es nulo o en la estación meteorológica cuando del pluviómetro se identifica con el valor -1 [8].

CAPÍTULO 5. PROCESADO DE DATOS

Un ejemplo de GAP se puede observar en la Figura 26 correspondiente a la estación meteorológica de Las Palmas de Gran Canaria Tafira-Zurbarán con fecha de febrero de 2016 dónde se registran diversos casos con un nivel de precipitación de -1 repitiéndose con bastante frecuencia lo largo de los días pudiendo generar escenarios como el respectivo al 8 de marzo dónde se estuvo casi dos días completos sin funcionar la estación [7].

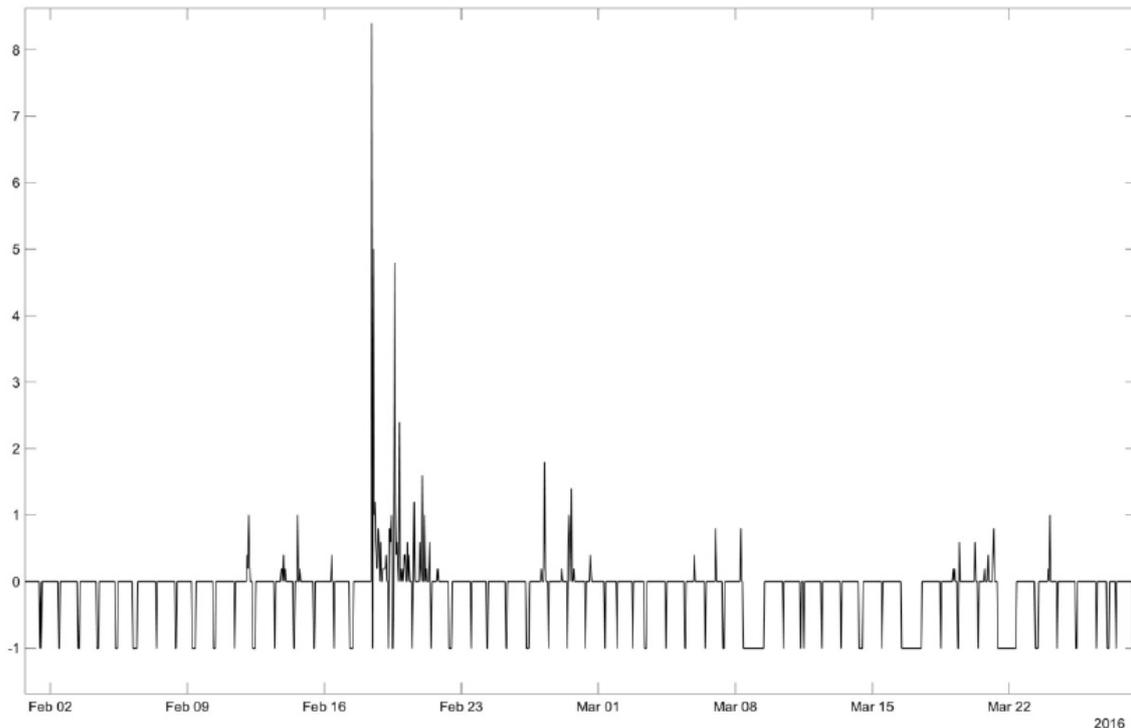


Figura 26. GAP producido en los niveles de precipitación.
Fuente: Fuente, Sonia, 2020 [7].

La clasificación que se podría crear en base a los GAPs se rigen en función de la manera en la que se presenten estos pueden aparecer debido a [3]:

- Ausencia de datos. Base temporal correcta, pero sin dato real de precipitación. Para detectar fácilmente estos valores, se introducen valores aberrantes que sean reconocibles a simple vista si se graficara como sucede con en el ejemplo de la Figura 26.
- Error de fecha. Falta de datos en periodos temporales completos. Para distinguirlos con mayor facilidad se localiza la fecha de inicio y final de la interrupción para obviar ese conjunto de datos.

Los GAPS aparecen por diferentes factores como ya se ha mencionado como [3]:

- Vida útil de la batería. Finalizan los ciclos de carga de la batería del sistema impidiendo su funcionamiento. Por eso se recomienda que siempre que sea posible, estos sistemas de medición deben ir conectados a una red eléctrica.
- Vida útil del equipo. La longevidad de los dispositivos repercute significativamente en su eficacia al realizar un trabajo de campo. Esto puede provocar que dejen de funcionar llegado el momento o incluso dar datos falsos.
- Lluvias torrenciales. En este TFM se ha mencionado reiteradamente la relación entre las precipitaciones y la atenuación de la señal. Un fenómeno meteorológico de grandes precipitaciones puede provocar que la energía de la señal se disipe de tal forma que el receptor no sea capaz de detectarla.
- Interferencias. Las estaciones transmiten en banda libre a frecuencia de MHz por lo que es sensible a otras señales que se entrometan en la misma banda que hagan que decaiga la integridad de la transmisión.
- Accidentes y vandalismo. Este caso cuenta con todas aquellas interacciones entre el mundo exterior y las estaciones de medición que repercutan directamente en la integridad de los dispositivos. Es por ello por lo que se recomienda que las estaciones estén localizadas en zonas de difícil acceso.

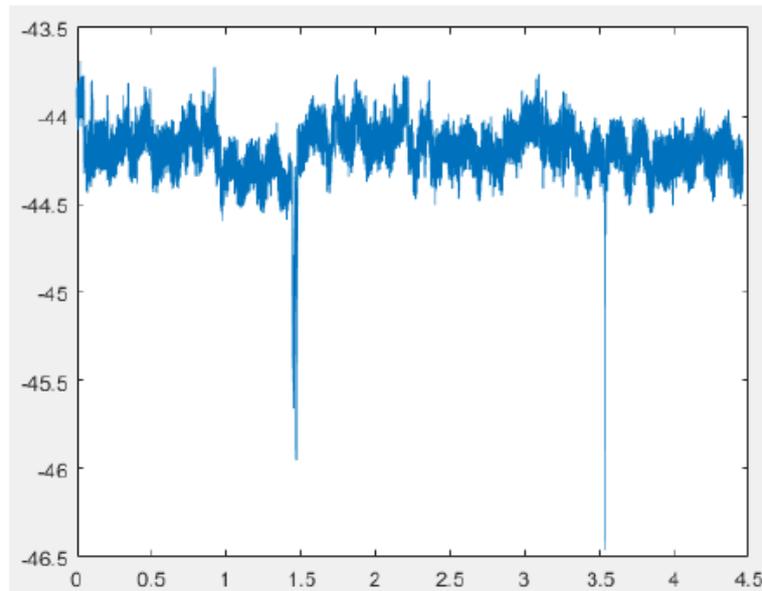
Hasta la fecha, los GAPS detectados a lo largo de toda la base de datos se pueden clasificar como ausencia de datos producidos por fenómenos como fallos en los equipos, relativo a la vida útil de los mismos, o fallos en la comunicación por accidentes como interrupciones de la red eléctrica [8].

Otro problema que requiere solución respecto a la base de datos consiste en la supresión de las componentes de baja frecuencia. Este fenómeno se puede asociar por los cambios que suceden durante la transición cíclica entre la noche y el día. Para el estudio posterior se requieren únicamente las componentes de alta frecuencia que son las que tienen relación con las atenuaciones de la señal provocadas por las precipitaciones [8].

El filtrado morfológico es una técnica que se emplea normalmente para el análisis y tratamiento de estructuras en imágenes digitales, gráficos, entre otros. En este proyecto, se

ha adoptado un filtrado de cierre. Este tipo de filtrado se compone en una encadenación de dos operaciones, en un inicio una operación de dilatación y posteriormente la erosión [3].

A modo de ejemplo gráfico de estas operaciones, se aporta en primer lugar, la señal original que va a ser filtrada (Figura 27).



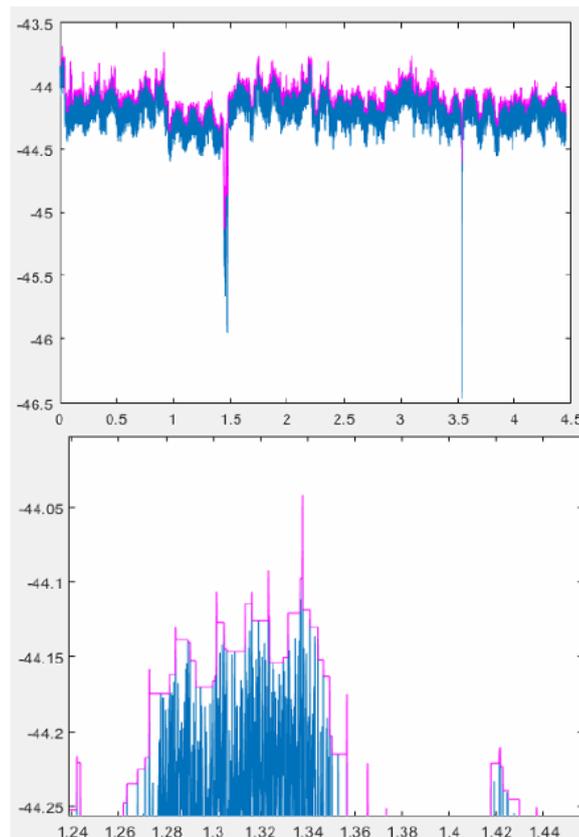
*Figura 27. Señal de niveles de potencia original.
Fuente: Elaboración propia.*

En segundo lugar, se aporta el resultado de la operación del filtrado morfológico que se muestra en la Figura 28 donde la imagen superior muestra en un color distinto aquellos valores seleccionados como borde superior tras la operación de cierre.m y la imagen inferior es una ampliación de la imagen superior para ver con más detalle el proceso. La operación del filtrado de cierre que se puede encontrar bajo el nombre de la función cierre.m en el Anexo 4. Función cierre.m Esta operación, como ya se ha comentado, se compone de dos acciones.

Por una parte, bajo el nombre de la función dilata.m (Anexo 5) se toman dos argumentos, uno primero vec, que es el vector en el que se realizará la dilatación, y n_elem, que es el tamaño del elemento estructurante utilizado en la dilatación. Luego longitud del vector de entrada vec se almacena en la variable long y se inicializa un vector llamado dil con ceros, que tendrá la misma longitud que el vector de entrada vec. Posteriormente, se crea un vector llamado ind que contiene los índices del 1 al n_elem. La

función continúa creando un vector llamado `seg` concatenando el vector de entrada `vec` con `n_elem-1` copias del último elemento de `vec`. Esto es necesario para que la dilatación se realice correctamente en los bordes del vector [3]. Después se inicia un bucle `for` que itera desde 1 hasta la longitud del vector de entrada `vec` dónde, en cada iteración del bucle, se asigna al elemento correspondiente de `dil` el valor máximo de los elementos en el segmento de `seg` indicado por `ind` para terminar actualizando el vector `ind` agregando 1 a cada uno de sus elementos. Esto se hace para mover el elemento estructurante a lo largo del vector. Finalmente, una vez que se completa el bucle, se devuelve el vector `dil`, que contiene el resultado de la operación de dilatación.

Por otra parte, bajo el nombre de la función `erosiona.m` (Anexo 6) se vuelven a tomar dos argumentos, uno primero llamado `vec`, que es el vector en el que se realizará la erosión, y `n_elem`, que es el tamaño del elemento estructurante utilizado en la erosión. La longitud del vector de entrada `vec` se almacena de igual modo en la variable `long`. Luego se inicializa un vector llamado `ero` con ceros, que tendrá la misma longitud que el vector de entrada `vec` para terminar creando un vector llamado `ind` que contiene los índices del 1 al `n_elem`. A continuación, crea un vector llamado `seg` concatenando `n_elem` copias del primer elemento de `vec` con el vector de entrada `vec`. Esto es necesario para que la erosión se realice correctamente en los bordes del vector al igual que pasaba con la dilatación [3]. Luego se inicia un bucle `for` que itera desde 1 hasta la longitud del vector de entrada `vec` dónde cada iteración del bucle se asigna al elemento correspondiente de `ero` el valor mínimo de los elementos en el segmento de `seg` indicado por `ind`. Finalmente, se actualiza el vector `ind` agregando 1 a cada uno de sus elementos. Esto se hace para mover el elemento estructurante a lo largo del vector y terminado el bucle, se devuelve el vector `ero`, que contiene el resultado de la operación de erosión.



*Figura 28. Resultados de pasar la señal original por el filtrado de cierre.
Fuente: Elaboración propia.*

En tercer lugar, en la Figura 29 se sustraen ambas señales (original y filtrada) y se añade un umbral de detección. Como se observa en dicha figura, la señal de potencia no fluctúa de la misma manera con las componentes de baja frecuencia a lo largo del tiempo, por consiguiente, se mejora la claridad de detección de los eventos de lluvia [8]. Estos eventos relacionados con las precipitaciones son fácilmente distinguibles como los picos que se observan en la propia figura.

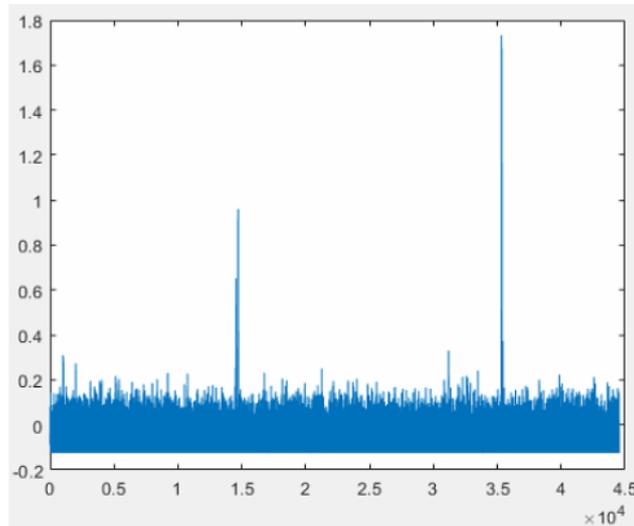


Figura 29. Señal de niveles de potencia filtrada.
Fuente: Elaboración propia.

De la Figura 29, se puede sacar en conclusión que la señal se compone principalmente de valores centrados en 0 con desviaciones de 0,125dB de manera superior como inferior. Esta fluctuación se debe al ruido de cuantificación que se genera cuando se digitalizan los niveles de potencia de entrada. En cualquier caso, este error de cuantificación se puede tomar también como un umbral para comenzar a determinar si el dato obtenido se trata de una posible precipitación [7]. Este error se incluye dentro del cálculo del umbral como se muestra en la (Ec. 12. En dicha ecuación se calcula basándose los parámetros RSL_Umb que es un vector que contiene los valores de potencia de la señal del radioenlace satelital, ind es un vector que selecciona las 25 muestras anteriores al instante de interés de estudio, k_0 es el coeficiente de umbral óptimo en el primer instante que inicialmente, tiene el valor de 0,95, finalmente, q_Error es el valor de 0,125dB procedente del error de cuantificación.

$$umbral = mean(RSL_Umb(ind) + k_0 \cdot q_Error) \quad (Ec. 12)$$

Dentro del script que calcula el mejor valor del umbral, una vez sea determinado el primer umbral, este valor entra en un bucle dónde itera hasta recorrer la longitud total de la señal RSL_Umb . Este bucle, se encarga de detectando pendientes positivas y adaptar el valor del umbral. Contando como una detección aquellos valores de potencia que superen el umbral con el que comienza. El bucle finaliza generando dos vectores que almacenan los

CAPÍTULO 5. PROCESADO DE DATOS

instantes de lluvia real y los instantes de lluvia detectada y se calcula un margen de confianza [8]. Finalmente, tras la obtención de ambas marcas de lluvia para un umbral determinado, se calculan los parámetros de calidad. Esto se realiza para comprobar el valor del umbral que presenta mejores resultados para el mes o periodo de estudio [3].

En la Figura 30, se observan los resultados de la primera iteración del estudio del umbral óptimo para ser considerada una detección de precipitación. Esta figura se compone de tres gráficas siendo la primera, la que se encuentra más arriba los instantes de lluvia real detectadas por el pluviómetro. La segunda, la gráfica central, son todos aquellos casos que bajo un determinado umbral se ha considerado como detección de lluvia. Finalmente, se muestra en tercer lugar, la gráfica inferior, representa todas aquellas coincidencias entre las dos gráficas anteriores. El caso dado, el primer valor de umbral resulta ineficiente debido a que se han contabilizado un gran número de eventos de lluvia que no corresponden con la realidad. Después de cada iteración se calcula de igual modo la matriz de confusión como se puede ver en la Figura 31 para que al finalizar todas las iteraciones se pueda obtener de manera gráfica y numérica el mejor valor para el umbral [8].

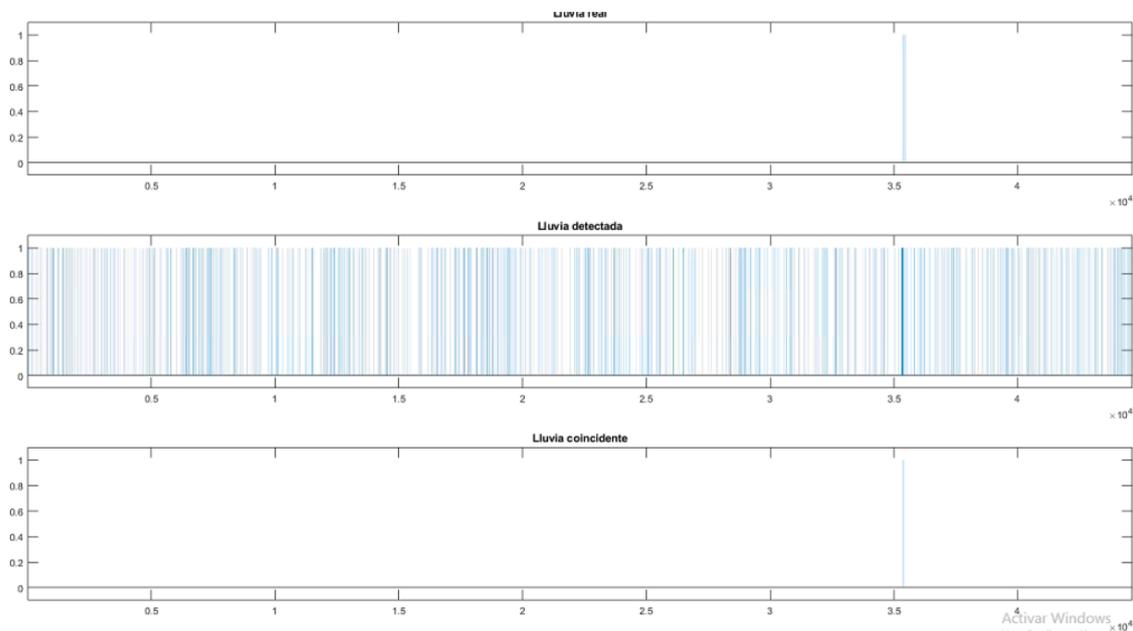


Figura 30. Lluvia real frente a lluvia detectada y coincidencias para un umbral dado.
Fuente: Elaboración propia.

En la Figura 31 se puede observar que un umbral de bajo valor aumenta en gran medida las detecciones falsas, es decir, los falsos positivos. No obstante, a medida que este umbral aumenta, desciende dicho número hasta cierto punto dónde si este umbral sigue creciendo, puede llegar a no reconocer un fenómeno de precipitación real [7].

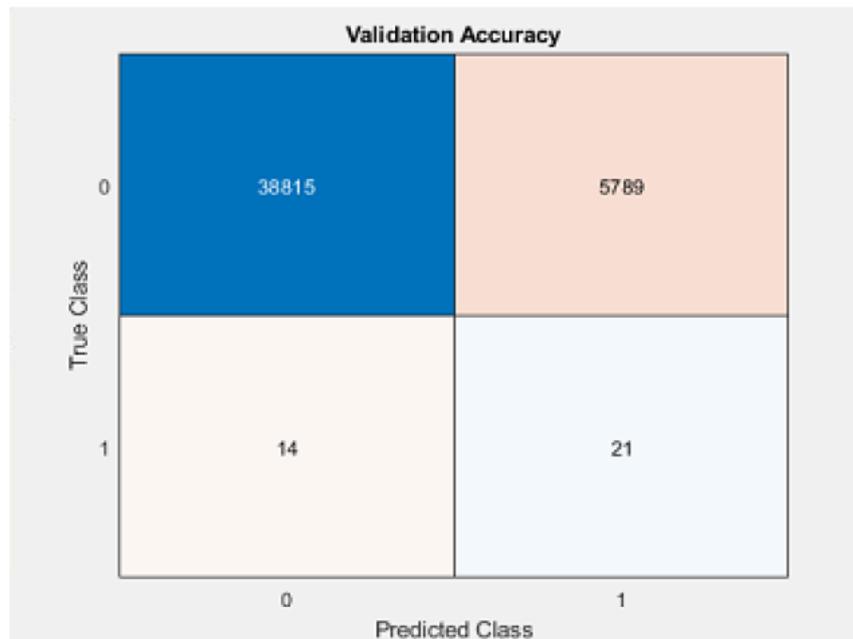
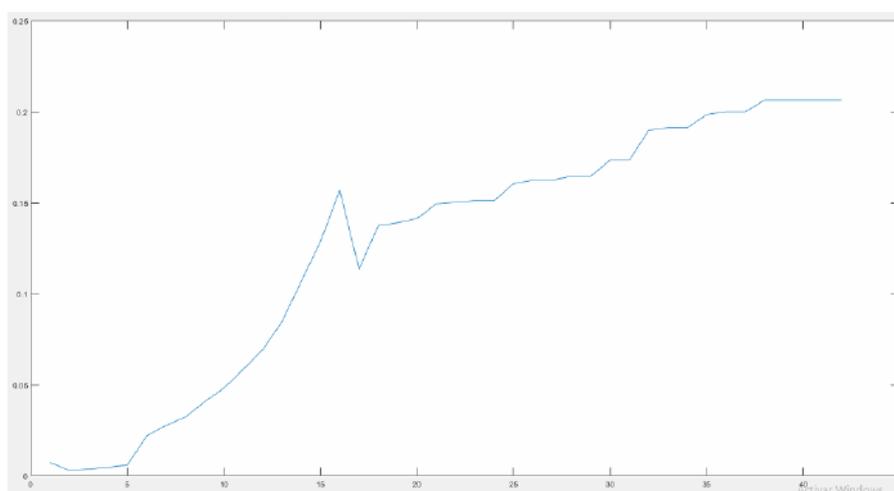


Figura 31. Matriz de confusión para un valor de umbral dado.
Fuente: Elaboración propia.

Una vez finalizadas todas las iteraciones con los diferentes valores de los umbrales, se almacena el mejor valor en la variable k_{opt} que se usará más adelante para el modelo de detección. La elección de este valor se basa en el parámetro F1-Score. El F1-score es una medida comúnmente utilizada en problemas de clasificación para evaluar el rendimiento de un modelo. Es una métrica que combina la precisión y el recall (sensibilidad o tasa de verdaderos positivos) de un modelo en un solo valor. Este parámetro se calcula de la forma en la que se presenta en la (Ec. 13) dónde el valor buscado (F1) se llega por medio del número de detecciones correctas (True Positive, TP), el número de casos positivos clasificados como negativos (False Negative, FN) y el número de casos negativos clasificados positivos (False Positive, FP).

$$F1 = \frac{TP}{TP + 0,5 \cdot (FP + FN)} \quad (Ec. 13)$$

Finalmente, se obtiene un valor k_{opt} de 2,8. Este valor es el que presenta una mayor puntuación. Se ha decidido tomar dicho valor ya que cuando llega a este, al aumentar el número de iteraciones se mantiene constante por algún tiempo, por lo que se utiliza el primer caso con el que se llega a dicho valor para así reducir el tiempo de cómputo de posteriores experimentos [3]. En la Figura 32 se muestra la evolución del F1-Score a medida que aumenta el umbral siendo la iteración 38 que da k_{opt} de 2,8.



*Figura 32. Evolución del parámetro F1-Score para valores de umbrales.
Fuente: Elaboración propia.*

El objetivo de todos estos pasos es concluir en la generación de un valor de umbral que se pueda considerar como adecuado para la detección y a su vez, la generación de dos vectores de misma longitud que asocien los eventos reales de lluvia junto a su nivel de potencia del radioenlace [9].

5.3. Base de datos de la red terrestre

Para el procesado de los datos que provienen de la red RESCAN y la AEMET se suprime el cálculo del umbral óptimo ya que se considera que las detecciones por parte de la AEMET son precisas y no requieren de otro umbral para determinar si un fenómeno es de lluvia o no. En caso de necesitarse, se puede reciclar la misma metodología del apartado anterior [5]. Por tanto, el desarrollo del procesado de estas estaciones sigue de la forma descrita en el diagrama de flujo de la Figura 33 usando los datos que proceden del radioenlace correspondiente al vano Hospital Militar – Pico I (Pico DGTNT) (Figura 15).



Figura 33. Diagrama de flujo para el procesamiento de datos de las estaciones AEMET.
Fuente: Tacoronte, Cristian, 2022 [8].

5.3.2. Sincronismo

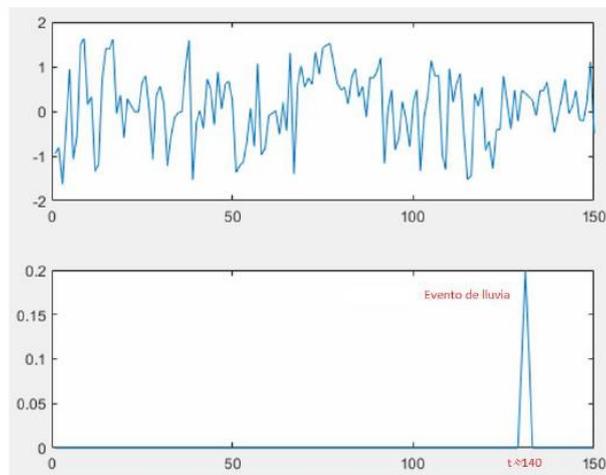
De manera similar al caso de la estación satelital, el primer paso se compone principalmente de cargar los datos de niveles de potencia a la base de datos. A continuación, se vuelve hacer la misma estrategia de cargar carpeta a carpeta los archivos con las mediciones meteorológicas por parte de la AEMET con el script de Sincronismo_Satelital_MesCOMBINADO6.m [7]. En esta ocasión en el código del Anexo 2 se especifica que mediante la tabla anterior y la actual se generen datos cada 5 minutos siendo estos de un valor igual a la interpolación entre ellos [8].

Al finalizar el programa, se obtiene una tabla cuya primera columna contiene tanto fecha como tiempo y es seguida por tantas columnas como estaciones meteorológicas haya. Cada columna es sincronizada con la anterior, siendo la dirección de sincronización de izquierda a derecha. Al terminar, se obtiene como resultado en la última columna un vector con todos los datos sincronizados de todas las estaciones meteorológicas disponibles

para un vano determinado [8]. Los registros de valor 0 se pasan al tipo NaN (Not a Number) para solo tener en cuenta los registros mayores de 0 [7]. Finalmente, esta serie de parámetros meteorológicos se encuentra junta a la serie de potencias de la señal transmitida sincronizadas con una muestra cada 5 minutos. Para el estudio resulta más fácil de acceder a una tabla donde se encuentra en un mismo instante su valor de potencia de señal y el nivel de precipitación asociado [3].

5.3.3. Obtención de secuencias de precipitaciones

Una vez relacionados niveles de potencia con niveles de precipitación se escogen aquellas secuencias que tengan por lo menos un evento de lluvia y un nivel de potencia asociado. Para esto, se da una ventana de longitud `SequenceLength` y se escogen todas las secuencias de dicho tamaño que tengan, como mínimo, una precipitación dentro de esta ventana y no falten sus niveles de potencia de la señal como la secuencia que se muestra en la Figura 34 dónde la primera gráfica muestra los niveles de potencia y la de abajo, los eventos de lluvia registrados.



*Figura 34. Secuencia con todos los niveles de potencia y al menos un caso de lluvia.
Fuente: Elaboración propia.*

Finalmente, dado a que hay un número considerable mayor de secuencias sin lluvias registradas que secuencias con eventos de lluvia, se decide etiquetar las secuencias con precipitaciones con un 1 y sin precipitaciones con un 0 para más adelante poder balancear las clases a la hora de crear el modelo de detección y predicción [3].

Capítulo 6. Redes LSTM para detección y predicción

Descripción de las técnicas y herramientas empleadas en Machine Learning para la creación de modelos de detección y predicción y la medición de calidad de estas.

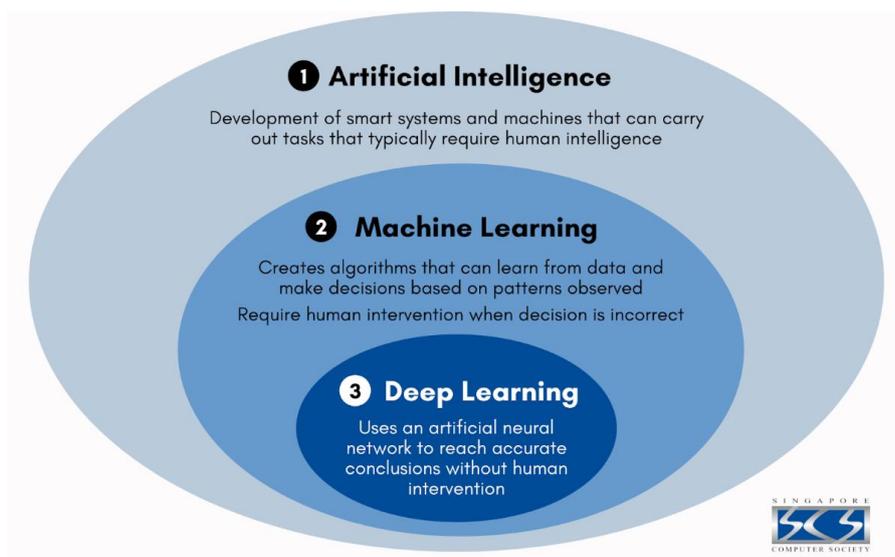
6.1. Introducción

El abanico de posibilidades que se presentan en el ámbito de las TIC existen diversos de investigación. Uno de ellos, que se ha vuelto extremadamente popular en los últimos años, es la Inteligencia Artificial (de sus siglas en inglés, AI). Dentro de dicho campo coexisten varias especializaciones como pueden ser las redes neuronales, máquinas de vectores soportados o procesamiento de datos entre otros [3]. En el caso que ocupa este proyecto, se tratarán herramientas para la detección y predicción de precipitaciones basándose en los niveles de potencia de los radioenlaces.

Los avances en el campo de la AI han conseguido diseñar y crear sistemas con capacidades superiores a las humanas, cada vez en más aspectos de la realidad. Estos sistemas son expertos en copiar o imitar características de la capacidad mental humana como puede ser relacionar reglas de la sintaxis de un lenguaje hablado o escrito de forma más rápida y eficiente superando las capacidades de cualquier persona que se enfrenta a la tarea de aprender un nuevo idioma y usarlo con soltura [29].

6.2. Algoritmos de Machine Learning

Una forma de ver las divisiones del área de conocimiento de la AI se muestra en la Figura 35. Esta figura muestra como un todo a la AI, luego, dentro de esta área de conocimiento se encuentra una especialización en el aprendizaje máquina (ML), encargado de crear algoritmos que puedan aprender patrones y tomar decisiones en función de una base de datos dada o unos parámetros impuestos para luego un humano vea el progreso y pueda corregirlo en caso de que se haya generado una red poco eficiente. Luego la especialización continúa con los mecanismos de Deep Learning (DL) siendo esta la más avanzada y la que más recursos consume dónde el propio algoritmo aprende por sí mismo los patrones, diferencias o reglas sin necesidad de la intervención humana [29].



*Figura 35. Áreas de la Inteligencia Artificial.
Fuente: Singapore Computer Society, 2020 [29].*

Dentro de las tareas de Machine Learning, se encuentra la de extraer conocimiento sobre algunas propiedades no observadas de un caso basándose en propiedades observadas del mismo caso o un caso similar. Esto puede escalar hasta tal punto llegando a predecir comportamientos futuros a partir de conocer lo ocurrido [29]. Este caso de predicción se retomará en capítulos posteriores.

Este tipo de aprendizaje que se realiza como una de las técnicas de ML, llamado aprendizaje inductivo, acarrea varios problemas. Normalmente se suele recurrir a dos tipos de redes [29]. Estos tipos de redes, relacionadas con el proyecto pueden ser clasificadas como:

- **Regresión.** Predice un valor, tanto futuro como presente, con un cierto margen de error. Por ejemplo, predecir las condiciones meteorológicas al atardecer teniendo previamente el valor de todas las condiciones meteorológicas pasadas.
- **Clasificación.** Predice la clase de un objeto en base a un conjunto de clases fijadas y los parámetros de cada una de ellas. Por ejemplo, clasificar si en un momento determinado hay un evento de lluvia o no a través del nivel de potencia de la señal de un radioenlace satelital o estaciones meteorológicas.

Dentro de este proyecto, según se ha ido comentando a lo largo del documento, queda claro que el caso que se ocupa en estos momentos el proyecto VIMETRI-MAC es

un problema de clasificación binaria, ya que se clasifica una muestra como de la clase lluvia o de la clase no lluvia. Si se quisiera, se podrían añadir más componentes haciendo una clasificación multiclase y clasifique las mediciones como llovizna, chubascos o trombas.

La herramienta básica con la que se desarrolla la AI son las redes neuronales (Neuronal Networks, NN). El sistema en el que se basa esta herramienta son las propias neuronas biológicas responsables del aprendizaje en los seres vivos [29]. Las redes neuronales se dividen en varias capas dónde cada una está formada por un conjunto de neuronas. Aumentar el número de capas significa hacer una red cada vez más profunda, lo que aumenta la capacidad de aprendizaje, no obstante, el costo computacional de dicha actividad también se incrementa pudiendo llegar hasta tal punto que se puede considerar un desperdicio de recursos. Cuando una NN se enfrenta a un problema complejo puede llegar a requerirse un nivel muy profundo debido a la necesidad de capas de aprendizaje con las que afrontar el problema. Estas NN profundas se conocen como Deep Learning o aprendizaje profundo pudiendo llegar a resolver desafíos que ni la inteligencia humana puede llegar a resolver [3].

Las redes creadas bajo el área de las DL pueden aprender a reconocer y clasificar conceptos complejos con facilidad e incluso llegar a conclusiones sobre contextos y entornos desafiantes. Un ejemplo de su capacidad se puede ver cuando son utilizadas en vehículos autónomos que identifican objetos circundantes, aparcan de manera autónoma o incluso llegar a conducirse por ellos mismos [29].

6.2.1. Redes Neuronales

Una Red Neuronal es un modelo matemático inspirado en el comportamiento biológico de las neuronas organizándose de manera similar a la estructura del cerebro. Se calcula que en un cerebro hay, aproximadamente, 100 mil millones de neuronas en la corteza cerebral formando un entramado de más de 500 billones de conexiones neuronales, pudiendo una neurona alcanzar 100 mil conexiones [8]. El cerebro pese a dotar a un ser vivo de cierta inteligencia no es capaz de competir con la precisión de las máquinas actuales para realizar cálculos complejos [3]. No obstante, de manera biológica, el cerebro sigue siendo el sistema más complejo descubierto. La superioridad del cerebro biológico

frente a la inteligencia artificial se centra en que, el cerebro biológico, puede llevar a cabo tareas muy complejas como el reconocimiento y clasificación de patrones que resultan un reto para la inteligencia artificial mientras que, para los seres vivos es algo común. Actualmente esto sigue siendo un campo de investigación con el objetivo de hacer sistemas cada vez más potentes que lleguen a resolver esos desafíos igual que una persona, como en el caso de reconocer un rostro entre una multitud o mantener una conversación de manera fluida [16].

En la Figura 36 se muestra en la figura de arriba, un dibujo sobre cómo es una neurona biológica frente a una neurona artificial que se expone en la parte de abajo de la figura. Esta figura está colocada de tal manera que se pueda apreciar de forma directa las similitudes de funcionamiento entre ambas.

Por una parte, la neurona biológica se compone del cuerpo central (Cell Body) denominado como soma, este cuerpo contiene el núcleo celular y compacta la estructura de la neurona. El núcleo celular es el encargado de recibir la información y transformarla en otra que guarda relación con la original. A la izquierda se localizan unas ramificaciones llamadas dendritas (Dendrites) que sirven de terminales por dónde le llega la información al núcleo de la neurona por parte de otras neuronas a las que esté conectada. Finalmente, a la derecha, se encuentra el axón (Axon), una prolongación que conecta la salida de la información procesada en el núcleo a otras neuronas. El paso de la información entre el axón de una neurona y las dendritas de otra es lo que se conoce como sinapsis. Este es el proceso por el cual se transmiten los impulsos nerviosos encargados de los procesos biológicos de la percepción y el pensamiento además de controlar todos los sistemas del cuerpo humano [8].

Por otra parte, se encuentra un perceptrón. El perceptrón es la forma más simple de una red neuronal, es decir, una única neurona. Usualmente se usa para clasificaciones sencillas fácilmente separables del mismo modo que hace un árbol de toma de decisiones. De una manera matemática, hablamos de dos clases de un linealmente separables. De manera práctica, se puede decir que un perceptrón simple es un umbral ajustable [30]. Como similitud con la neurona biológica se puede apreciar cómo se compone de una serie de dendritas a la izquierda (Señales de entrada) que pasan a un núcleo (Pesos sinápticos,

Unión sumadora y Función de activación) para finalmente generar una señal de salida (Salida).

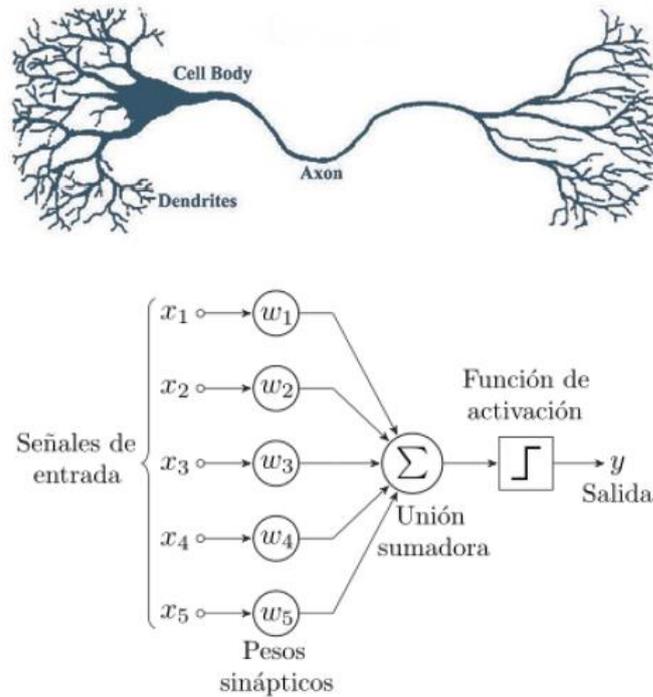


Figura 36. Neurona biológica y neurona artificial.
Fuente: Tacoronte, Cristian, 2022 [8].

El perceptrón consiste en una neurona con pesos sinápticos y umbrales ajustables. El algoritmo que se emplea para el ajuste de los parámetros libres surge por primera vez en un procedimiento de aprendizaje desarrollado por Rosenblatt en 1958. Las conclusiones de dicho trabajo demostraron que, si los patrones usados para entrenar el perceptrón son de dos clases linealmente separables, entonces el algoritmo del perceptrón converge tomando como superficie de decisión un hiperplano que separa ambas clases. Este resultado es lo que se conoce como el teorema de convergencia del perceptrón [3]. Dicho perceptrón se limita a realizar la clasificación de dos clases ya que este sólo posee una salida, si se aumentara el número de salidas, aumentaría de igual modo el número de clases para clasificar ya que posteriormente se podría conectar a más perceptrones. No obstante, ya que es un modelo simple, para poder garantizar su funcionamiento, las clases añadidas deben seguir siendo linealmente separables entre ellas [30]. La (Ec. 14) muestra la operación matemática con la que se obtiene la salida del perceptrón.

$$y_i = f_i(h_i) = f_i\left(\sum_{j=0}^n w_{ij} \cdot x_j - \theta_i\right) \quad (\text{Ec. 14})$$

Dónde x_n son los valores de los que se compone el vector de entrada, w_i son los pesos sinápticos que se asignan a cada entrada, f_i la función de activación e y_i la salida del perceptrón. La función de activación es un parámetro que se deja a la experiencia del programador y que depende de la tarea realizada por la neurona. Dentro de la herramienta de MATLAB se deja una gran selección de funciones de activación entre las que se encuentran las resumidas en la Figura 37. El tipo de función depende inicialmente del experimento o el tipo de clase que se utiliza [30].

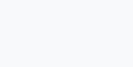
Name	Plot	Function, $f(x)$
Identity		x
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1 + e^{-x}}$
Hyperbolic tangent (tanh)		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Rectified linear unit (ReLU) ^[10]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x \cdot 1_{x>0}$
Gaussian Error Linear Unit (GELU) ^[4]		$\frac{1}{2}x \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$ $= x\Phi(x)$
Softplus ^[11]		$\ln(1 + e^x)$
Exponential linear unit (ELU) ^[12]		$\begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$ with parameter α
Scaled exponential linear unit (SELU) ^[13]		$\lambda \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$ with parameters $\lambda = 1.0507$ and $\alpha = 1.67326$
Leaky rectified linear unit (Leaky ReLU) ^[14]		$\begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Parameteric rectified linear unit (PReLU) ^[15]		$\begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$ with parameter α
Sigmoid linear unit (SiLU) ^[4] , Sigmoid shrinkage, ^[16] SiL ^[17] or Swish-1 ^[18]		$\frac{x}{1 + e^{-x}}$

Figura 37. Funciones de activación de un perceptrón.
Fuente: Wikipedia, 2021, [31].

Históricamente, un único perceptrón no era capaz de afrontar problemas complejos y por lo general el proceso de entrenamiento y uso de este resultaba menos eficiente y se requería más potencia de cálculo para su ajuste que un árbol de decisiones. Es por ello por lo que la evolución natural del perceptrón es el perceptrón multicapa. Este resulta ser una generalización del perceptrón simple. En 1969, Minsky y Papert demostraron que la combinación de varios perceptrones simples llegaba a una solución adecuada para tratar con algunos problemas de clasificación no lineales [3]. Actualmente, el perceptrón multicapa es una de las arquitecturas más recurrentes para la resolución de problemas complejos debido a su capacidad para aproximarse a una solución real y sencillez de programación y aplicación [30].

Las arquitecturas que comúnmente se pueden encontrar del perceptrón multicapa se caracterizan por neuronas agrupadas en diversos niveles, también llamados capas como se puede ver en la Figura 38. Cada capa se forma por un conjunto de neuronas clasificadas en tres niveles: capa de entrada, capa oculta y capa de salida. En primer lugar, en la capa de entrada, las neuronas sirven de recipientes donde se almacenan los datos de llegada al igual que pasaba con las dendritas en una neurona biológica. Solo se encargan de recibir y propagar la señal de entrada al resto de neuronas de la capa posterior. En segundo lugar, la capa oculta, a modo de núcleo neuronal, toma los datos de la capa de entrada y realizan un procesamiento no lineal de los patrones recibidos y envía dicho resultado de la señal de entrada a la capa oculta continua y así hasta llegar a la última capa. En tercer lugar, la capa de salida que se asemeja a los axones biológicos, siendo los encargados de transmitir la señal de entrada que se ha transformado en las capas previas dando lugar a la respuesta de los patrones de entrada [30].

En el caso de los perceptrones multicapa, las conexiones siempre se dirigen hacia delante, es decir, las neuronas de una capa transmiten la información siempre a capas posteriores. Este tipo de conexiones pueden conocerse como redes alimentadas hacia adelante o redes feedforward. Generalmente, la salida de cada neurona de una capa se conecta a la entrada de cada una de las neuronas de la siguiente capa como se ve en la Figura 38. Esta característica recibe el nombre de red totalmente conectada [30].

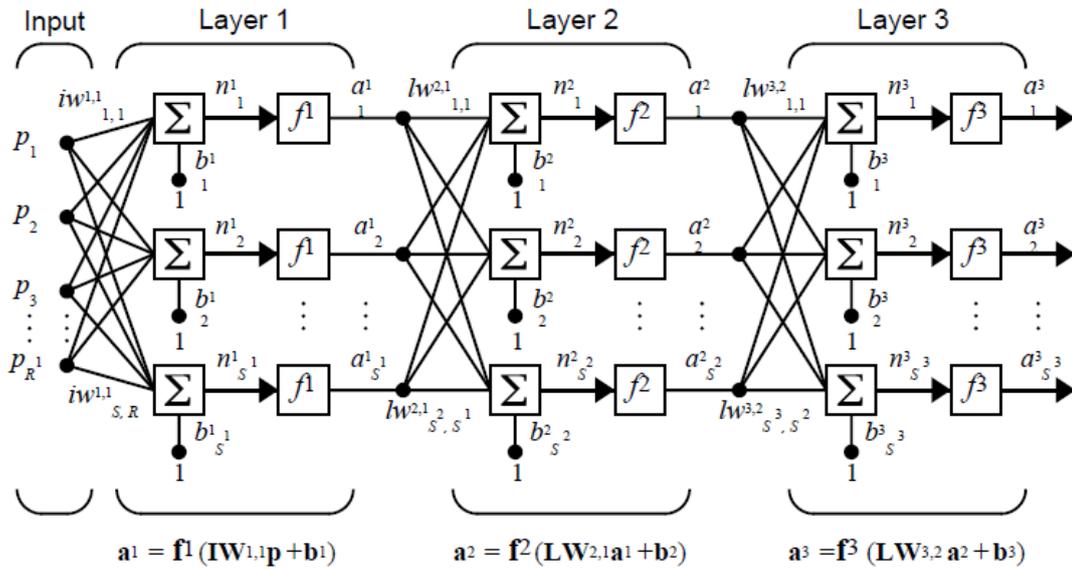


Figura 38. Modelo de perceptrón multicapa.
Fuente: Demuth, Howard; Beale, Mark, 2004 [30].

El diseño de la arquitectura de la red perceptrón multicapa implica escoger la función de activación, el número de neuronas de cada capa y el número de capas de la red [30]. Por un lado, la función de activación suele basarse en la tarea a cumplir. Por norma general, esta elección influye en la eficacia y rapidez con la que el sistema es capaz de resolver el problema dado y no tanto en si será capaz de resolverlo. Por otro lado, el número de neuronas y de capas, suelen darse de igual forma por el problema. No obstante, en condiciones complejas, estos deben ser elegidos por el diseñador [3]. De inicio, el número de neuronas de la capa de entrada dependerá del número de parámetros o clases que se vayan a introducir en la red, mientras que, el número de neuronas de la capa de salida debe corresponder al número de clases que se espera. Por ejemplo, si se quisieran clasificar tres tipos de flores según la longitud y la anchura de su pétalo, sépalo y hoja, se tendría una red que de entrada tendría seis neuronas, una por cada dato de entrada, y tres neuronas en la capa de salida, una por cada tipo de flor a clasificar [30]. Es común encontrarse con problemas que requieren un gran número de entrada para unas pocas clases a la salida volviendo la estructura de la red demasiado compleja. En esos casos, resulta conveniente estudiar aquellos parámetros de entrada que aportan más información para la clasificación y descartar aquellos otros que no son tan influyentes para poder descartarlos sin que repercutan en gran medida en la clasificación posterior. Para conocer

las variables menos influyentes se recurren a técnicas de análisis de correlación, análisis de componentes principales, análisis de importancia relativa, análisis de sensibilidad entre otras [30]. Finalmente, el número de neuronas pertenecientes a la capa oculta y el número de capas ocultas que debe tener el sistema suelen basarse en experimentación y experiencia del diseñador. Hoy en día, sigue sin existir una metodología que determine el número óptimo de ambos parámetros para resolver un problema dado. Como norma general, se suele experimentar iniciando con una arquitectura básica para posteriormente aumentar o disminuir el número de neuronas y capas ocultas hasta alcanzar una arquitectura que resuelva el problema con el coste de cómputo y parámetros que al diseñador le parezca correcto [8].

Las redes neuronales recurrentes son otro tipo de NN capaces de reconocer y predecir secuencias de datos a lo largo del tiempo [30]. Estas redes funcionan mediante bucles que permiten que una salida de cualquier parte de la red se pueda utilizar como entrada de la propia red en la siguiente iteración como se puede ver en la Figura 39.

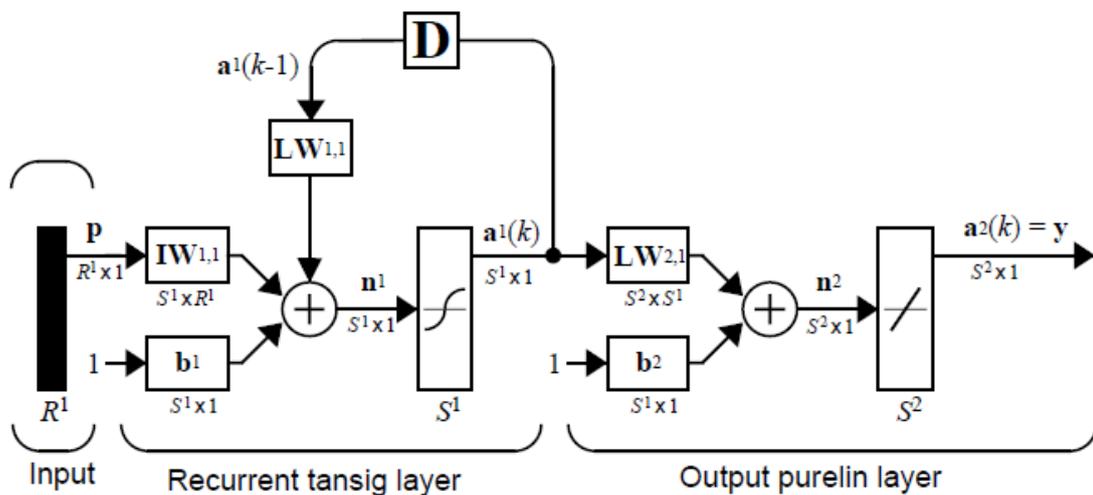


Figura 39 Modelo de red neuronal recurrente.
Fuente: Demuth, Howard; Beale, Mark, 2004 [30].

De manera simplificada, se explicará el funcionamiento de una red neuronal recurrente considerándola como un perceptrón multicapa de una sola capa oculta y haciendo que la salida sea utilizada como entrada en la siguiente evaluación como muestra la Figura 40. Este acto de tomar un dato anterior para usar posteriormente se asemeja con

el concepto de recordar. A medida que aumenta el número de capas se aumenta de igual modo la memoria del modelo por lo que será capaz de reconocer mayores secuencias [32]. En este caso, se tiene que X_t es la entrada de la red, A es una red neuronal y h_t la salida de la red. Como se puede observar en dicha imagen, parte de la salida se realimenta con la entrada. Se puede entender este tipo de arquitecturas como réplicas sucesivas de la misma red, pero cada una de ellas tiene a la entrada información añadida que procede de un estado anterior como se muestra en la Figura 41.

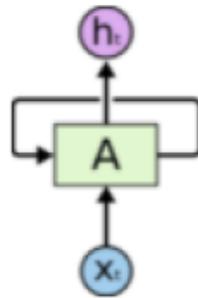


Figura 40. Arquitectura de red neuronal recurrente básica.
Fuente: Mañas, Andrés, 2019 [32].

Las redes neuronales recurrentes padecen de un impedimento conocido como el problema del desvanecimiento del gradiente. Este problema radica en que su efectividad de recordar es limitada, por lo que no va a poder aprender dependencias de largo alcance. Cuando se realiza dicha la propagación hacia atrás, los gradientes resultan disminuir a medida que la red avanza hacia atrás. En otras palabras, dada una salida h_t cuantos más instantes posteriores se produzcan estos aprenderán del instante h_t con menor rapidez que el instante inmediato de h_t [3].

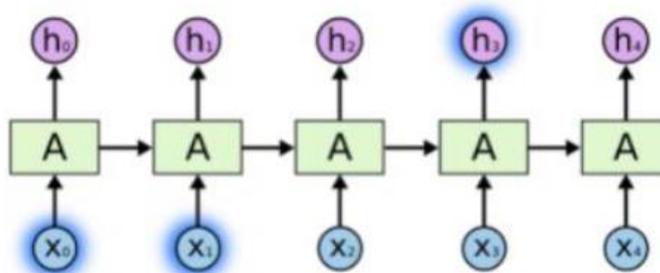


Figura 41. Red neuronal recurrente expandida.
Fuente: Mañas, Andrés, 2019 [32].

Un ejemplo de la mala memoria de estas redes se puede ver si se intenta crear un predictor de palabras dónde data una frase inicial se predice la siguiente palabra. En el ejemplo que se muestra en la Figura 42 se inicia con una frase, como, por ejemplo, “En Francia me tomé una cerveza... mientras me encontraba en” siendo la siguiente palabra Marsella. Se puede ver como en dicho ejemplo que su predicción es Baviera, una predicción incorrecta causada por la pérdida de información relevante debido a su distancia a un dato de importancia (como puede ser Francia para intentar predecir un lugar francés) y provocar un error (en este caso Baviera que se encuentra en Alemania). De esta forma, se puede ver de manera ilustrativa cómo el peso de la información de que me encontraba en Francia se ve opacada por información más reciente como cerveza dando lugar a una predicción basada en la cerveza y no en Francia. En resumen, estos errores se originan debido a que la distancia entre la información relevante y la predicción se alejan lo suficiente como para que ocurran errores como el antes mencionado [32].

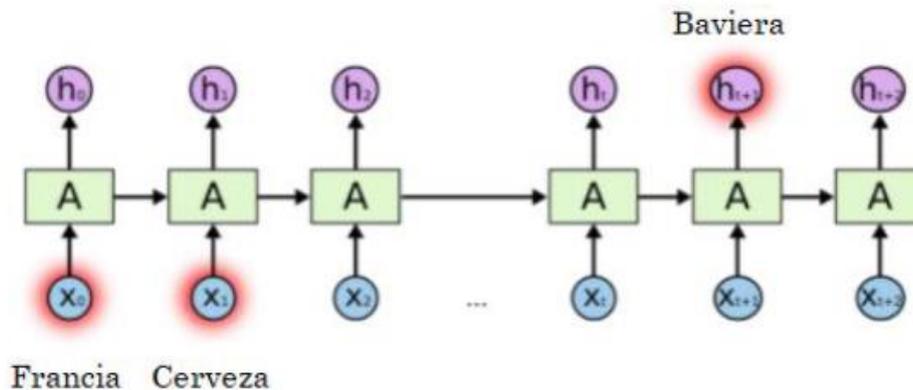


Figura 42 Predicción basada en la cerveza.
Fuente: Mañas, Andrés, 2019 [32].

6.2.2. Tipos de aprendizaje de Machine Learning

Durante el proceso de entrenamiento de una red neuronal, dónde se van modificando los umbrales para llegar a la solución del problema, se pueden llegar a establecer distintas metodologías para dicho proceso. El objetivo de una red será clasificar una entrada en alguna de las clases anteriormente establecidas construyendo un agente predictor que es evaluado al final. Dependiendo de la evaluación y modificación de este agente, se crean las técnicas de aprendizaje de Machine Learning y se clasifican en [30]:

- Aprendizaje supervisado. Los predictores y las respuestas son conocidos. Siendo ya clasificadas previamente, un conjunto de valores para cada clase de salida. Mediante esta técnica de aprendizaje se construyen modelos matemáticos para predecir observaciones posteriores. El punto de que se llamen supervisados radica en que el modelo se construye basándose en valores conocidos. Esta supervisión puede clasificarse en función de la variable de salida. Si la salida es un resultado cuantitativo, suele recurrirse a regresiones para su entrenamiento, mientras que si la salida es cualitativa se suelen utilizar como clasificadores.
- Aprendizaje no supervisado. La red aprende por cuenta propia la estructura de las clases y los patrones necesarios para resolver el problema. Estos métodos emplean técnicas de agrupación y métodos de análisis de componentes. Normalmente, para que la red de buenos resultados mediante este método, se requiere alimentarla con grandes volúmenes de datos.
- Aprendizaje semisupervisado. Combina ambos tipos anteriores de aprendizaje. Suele utilizarse cuando se requiere reforzar una clasificación o afinar un valor posterior.
- Aprendizaje por refuerzo. Este algoritmo funciona como el no supervisado, pero reforzando aquellas acciones acertadas y penalizando las incorrectas. Un ejemplo de esto sería la creación de un clasificador que distinga entre perros y gatos y cuando se revise sus resultados, se premien aquellos animales bien clasificados y castigando en los casos de confusión.
- Transducción. Similar al aprendizaje supervisado, pero con el objetivo de predecir las categorías en las que caen las siguientes entradas basándose en ejemplos de entrada. Dicho de otra forma, sería un aprendizaje supervisado dinámico.
- Aprendizaje multitarea. Hace acopio de todos los algoritmos de aprendizaje que usan un conocimiento previo aprendido por el sistema para asimilar nuevos problemas de características parecidas. Por ejemplo, la red anterior que distinguía entre perros y gatos, reutilizar su estructura para que distinga entre pulpos y medusas.

En resumen, un algoritmo de aprendizaje se encarga de adaptar y modificar los parámetros de la red con el objetivo de minimizar el error. En el caso propuesto del

perceptrón multicapa, se emplea un algoritmo de aprendizaje supervisado dónde la modificación de los parámetros se realiza de forma que la salida de la red sea próxima a la salida verdadera o deseada. Por tanto, por cada patrón de entrada se requiere del patrón de salida correcto. Es por ello, que el objetivo de la red es minimizar el error cometido a su salida respecto a la salida verdadera. Este error se puede calcular mediante la (Ec. 15) y la (Ec. 16). En estas ecuaciones, E es la función de error, mide la diferencia entre la salida del modelo y la salida real. N es el número de muestras y e(n) es el error cometido por la red para el patrón n. Dicho error se evalúa mediante los parámetros Y, la salida de la red y S la salida deseada, en ambos casos bajo el patrón n [30].

$$E = \frac{1}{N} \sum_{n=1}^n e(n) \quad (\text{Ec. 15})$$

$$e(n) = \frac{1}{n_c} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (\text{Ec. 16})$$

6.2.3. Parámetros de calidad

A la hora de determinar el comportamiento de una red frente a un problema existen unos parámetros de calidad que facilitan la tarea de medir el rendimiento del modelo. Estos parámetros de calidad muestran el rendimiento de la red para proporcionar las salidas correctas por medio de datos que no se han usado en el entrenamiento. Es común ver que cuando se estudia la base de datos para generar una red neuronal se hable de la partición de datos. Dicha partición divide los datos usados para entrenar el modelo en dos partes, los datos de entrenamiento y los datos de prueba. De toda la base de datos se suele coger una gran parte de los datos de dicha base para el entrenamiento (alrededor del 70% o más) y estos datos se entrenamiento, en algunos casos se pueden volver a dividir reservando una pequeña porción de estos asignándolos como datos de validación que se usan para ir comprobando el rendimiento del modelo durante su fase de entrenamiento. Luego se encuentran los datos de testeo que son usados cuando ya el modelo está entrenado y se introducen estas secuencias en la red para medir si los resultados a la salida son correctos.

CAPÍTULO 6. REDES LSTM PARA DETECCIÓN Y PREDICCIÓN

Es por ello, que con los resultados después de la fase de prueba se generen diversas métricas pensadas para graficar el rendimiento de la red neuronal [29].

La matriz de confusión es una herramienta básica para evaluar una red neuronal. Esta muestra los resultados que se obtienen tras la fase de prueba disponiéndolos en una matriz que contabiliza los verdaderos positivos, los falsos positivos, los falsos negativos y los verdaderos negativos. Con los que posteriormente se originan otras métricas de calidad [30]. A continuación, se explican cada uno de los componentes de la matriz de confusión cuyo resumen se puede ver en la Tabla 6:

- Verdaderos positivos (TP). El valor a la salida de la red es de una clase, que llamaremos positivo, y este concuerda con el valor real. Por ejemplo, la red clasifica a una persona enferma como enferma.
- Verdaderos negativos (TN). El valor a la salida de la red es de una clase, que llamaremos negativo, y este concuerda con el valor real. Por ejemplo, la red clasifica a una persona sana como sana.
- Falsos positivos (FP). El valor a la salida de la red es de la clase positivo pero su valor real era de la clase negativo. Por ejemplo, la red clasifica a una persona sana como enferma.
- Falsos negativos (FN). El valor a la salida de la red es de la clase negativo pero su valor real era de la clase positivo. Por ejemplo, la red clasifica a una persona enferma como sana.

Tabla 6. Matriz de confusión.

Fuente: Demuth, Howard; Beale, Mark, 2004 [30].

		Real	
		Positivo	Negativo
Predicción	Positivo	TP	FP
	Negativo	FN	TN

Otro parámetro para medir el funcionamiento de la red neurona es la precisión. Este parámetro mide la dispersión del conjunto de valores obtenidos tras la repetición de una magnitud. A menor dispersión, mayor la precisión. Se calcula mediante la (Ec. 17 [30]).

$$Precisión = \frac{TP}{TP + FP} \quad (Ec. 17)$$

De igual modo, se puede medir la exactitud. Este parámetro mide la proximidad de una salida al valor verdadero. En estadística, se hace referencia al sesgo de una estimación. Se calcula por medio de la (Ec. 18 [30]).

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (Ec. 18)$$

También se puede recurrir a la sensibilidad. Este parámetro mide la proporción de casos positivos correctamente asignados por el modelo entre todos los casos positivos que se encontraban en la base de datos. Se calcula por medio de la (Ec. 19 [30]).

$$Sensibilidad = \frac{TP}{TP + FN} \quad (Ec. 19)$$

De la misma forma se puede recurrir a la especificidad. Este parámetro mide la proporción de casos negativos correctamente asignados por el modelo entre todos los casos negativos que se encontraban en la base de datos. Se calcula por medio de la (Ec. 20 [30]).

$$Especificidad = \frac{TN}{TN + FP} \quad (Ec. 20)$$

Otro tipo de parámetro que ayuda a medir el rendimiento de una red es el F1-Score. Compara el rendimiento de la precisión y la exhaustividad entre varias soluciones. Se calcula mediante la (Ec. 21 [3]).

$$F1 = \frac{TP}{TP + 0,5 \cdot (FP + FN)} \quad (Ec. 21)$$

Finalmente, el coeficiente kappa, también conocido como el coeficiente de Cohen's kappa. Mide la concordancia entre dos clasificadores más allá de lo que se esperaría por casualidad. Proporciona una medida más robusta que la coincidencia porcentual, especialmente cuando las clases están desequilibradas en los datos. El coeficiente kappa varía entre -1 y 1, donde 1 indica una concordancia perfecta, 0 indica concordancia al azar y -1 indica una discordancia completa [33]. Su cálculo se reduce a la (Ec. 22).

$$kappa = \frac{2 \cdot (TP \cdot TN - FN \cdot FP)}{(TP + FP) \cdot (FP + TN) + (TP + FN) \cdot (FN + TN)} \quad (Ec. 22)$$

Este parámetro se usará en posteriores experimentos dónde la interpretación de su resultado se ha llevado a cabo basándose en el trabajo de Carletta, Jean (1996) [33] pero adaptando los umbrales al estudio de la manera que se muestra en la Tabla 7.

Tabla 7. Clasificación del coeficiente de Cohen's kappa.
Fuente: Elaboración propia.

Cohen's kappa	Calidad de la red
< 0,00	Concordancia nula
0,01 - 0,20	Concordancia mínima
0,21 - 0,40	Concordancia débil
0,41 - 0,60	Concordancia razonable
0,61 - 0,80	Concordancia buena
0,81 - 1,00	Concordancia casi perfecta

6.3. Red LSTM

Las redes neuronales recurrentes padecían de tener una baja capacidad de aprendizaje. No obstante, este problema se ha podido solucionar mediante la creación de las redes LSTM (Long Short-Term Memory) un tipo red neuronal recurrente con la capacidad de aprender dependencias a largo plazo [30]. Las redes LSTM son diseñadas con celdas de memoria capaces de mantener su estado a largo plazo, es decir, tras varias iteraciones. Su objetivo es combatir el problema del desvanecimiento del gradiente nombrado de las redes neuronales recurrentes [9]. Los LSTM, pese a tener una misma estructura que las anteriores, es decir, en forma de cadena, el módulo de repetición difiere al que tenían sus redes antecesoras. La red, en lugar de tener una sola capa ahora hay cuatro que interactúan entre ellas como se ve en la Figura 43. Los cuadros amarillos son una capa de red neuronal, los círculos representan un punto de operación y las líneas corresponden a vectores que pueden concatenarse o separarse [32].

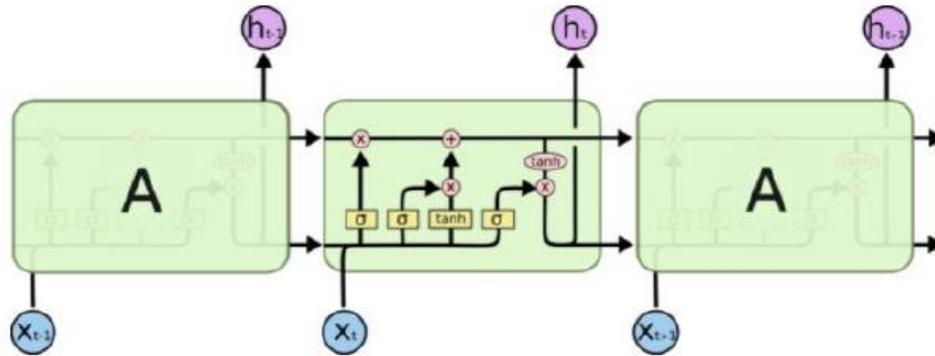


Figura 43. Red LSTM.
Fuente: Mañas, Andrés, 2019 [32].

La línea principal que se resalta en la Figura 44 representa el estado de la celda. Esta línea traspasa la celda recogiendo la información de cada capa y lo lleva a la salida [30].

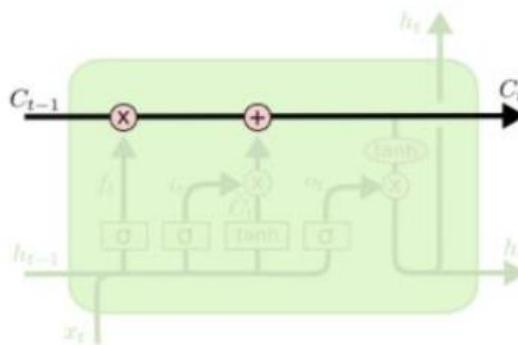


Figura 44. Línea principal de la red LSTM.
Fuente: Mañas, Andrés, 2019 [32].

La red LSTM es capaz de borrar o añadir información a una celda, por medio de puertas. Las puertas son estructuras que dejan pasar información o bloquear su flujo. Suelen estar compuestas por una capa de red neuronal sigmoidea y un multiplicador sobre la línea principal permitiendo que esta línea acarree la información de la red neuronal sigmoidea [3]. Esta red sigmoidea genera números entre 0 y 1. El valor determina cuánto valor de cada componente deja pasar. De esa forma, la red LSTM es capaz de decidir qué información debe ser eliminada del estado de la celda. Del mismo modo, se decide qué nueva información se almacena en el estado de la celda. Esta acción de almacenar, como se ve en la Figura 45, se procesa en primer lugar, por la capa sigmoidea que actúa como puerta de entrada decidiendo qué valores se actualizarán, y, en segundo lugar, actúa de

igual modo la capa tanh creando un vector de nuevos valores que se agregan al estado de la red [7]. Cuando se deciden las variables a actualizar, se calcula el nuevo valor de la celda multiplicando el viejo estado por la señal de la capa sigmoidea, permitiendo así borrar información que no es de interés y agregando la que se considera relevante [8].

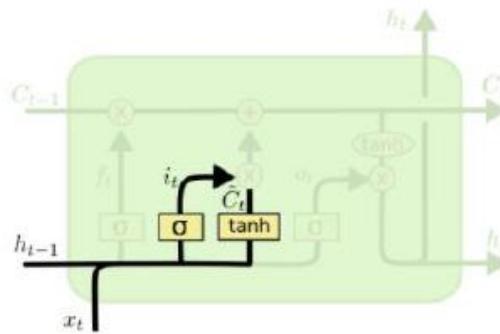


Figura 45. Almacenamiento en una red LSTM.
Fuente: Mañas, Andrés, 2019 [32].

Finalmente, se genera la salida de la celda basándose en el estado actual pero filtrado. Como se muestra en la Figura 46, una capa sigmoidea indica las partes a generar y estas se multiplican por una función tanh que marca la importancia de esa nueva información antes de su salida [8].

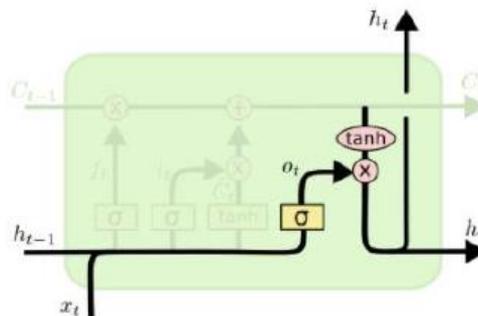


Figura 46. Salida de una red LSTM.
Fuente: Mañas, Andrés, 2019 [32].

Una vez terminada esta sección se ha logrado completar el segundo objetivo O2. Estudiar los algoritmos para la cuantificación y predicción de lluvia. De esta forma, se dispondrá de la suficiente base teórica como para poder generar las redes neuronales para la detección, cuantificación y predicción de precipitaciones.

Capítulo 7. Experimentos y Resultados

En este capítulo se presentan los modelos creados en la fase de experimentación junto a sus resultados para la detección y predicción de fenómenos de lluvia.

7.1. Algoritmo de detección

Uno de los puntos más relevantes de este proyecto está relacionado con los cambios que se han hecho respecto a antiguos trabajos. De este modo, se experimentan con otros tipos de estructuras en las redes propuestas con anterioridad con el objetivo de reducir su coste computacional u obtener mejores resultados [16].

En este capítulo se detalla el recorrido de los datos, a la salida del preprocesado, dentro de la red de detección. En otras palabras, se describe la estructura de la nueva red empleada, y los procesos de entrenamiento y prueba. A lo largo de los siguientes puntos se expondrá tanto la metodología antigua como la nueva y se compararán los resultados obtenidos por ambas metodologías. Finalmente, cabe destacar que las arquitecturas que se emplearán en ambos casos están basadas en redes LSTM y creadas en el entorno de MATLAB R2022b.

7.1.1. Variables principales

Dentro de la programación de la red neuronal creada para la detección, en primer lugar, en orden de lanzar los experimentos, se crea en primer lugar, un script que servirá para juntar las variables relevantes llamado `Lanza_Detector_Lluvia_COMBINADO8_4.m` que se adjunta en el Anexo 7. En este script, se juntan la mayoría de los parámetros con los que se experimentará posteriormente y se toman decisiones como el sistema operativo empleado, el número de meses previos con los que se entrena la red, el tamaño de las secuencias, entre otros. Una vez ejecutado este script, se llaman a otros programas que se mencionarán más adelante. Tras la finalización de su ejecución, se obtendrá una carpeta con el nombre del experimento y esta almacena las métricas y la última red creada en formato mat y replica una plantilla que genera un archivo xlsx dónde se guardan los datos de calidad de cada red generada como se muestra en la Figura 47.

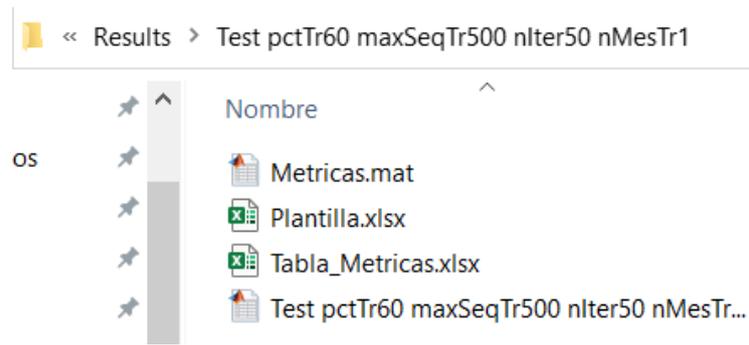
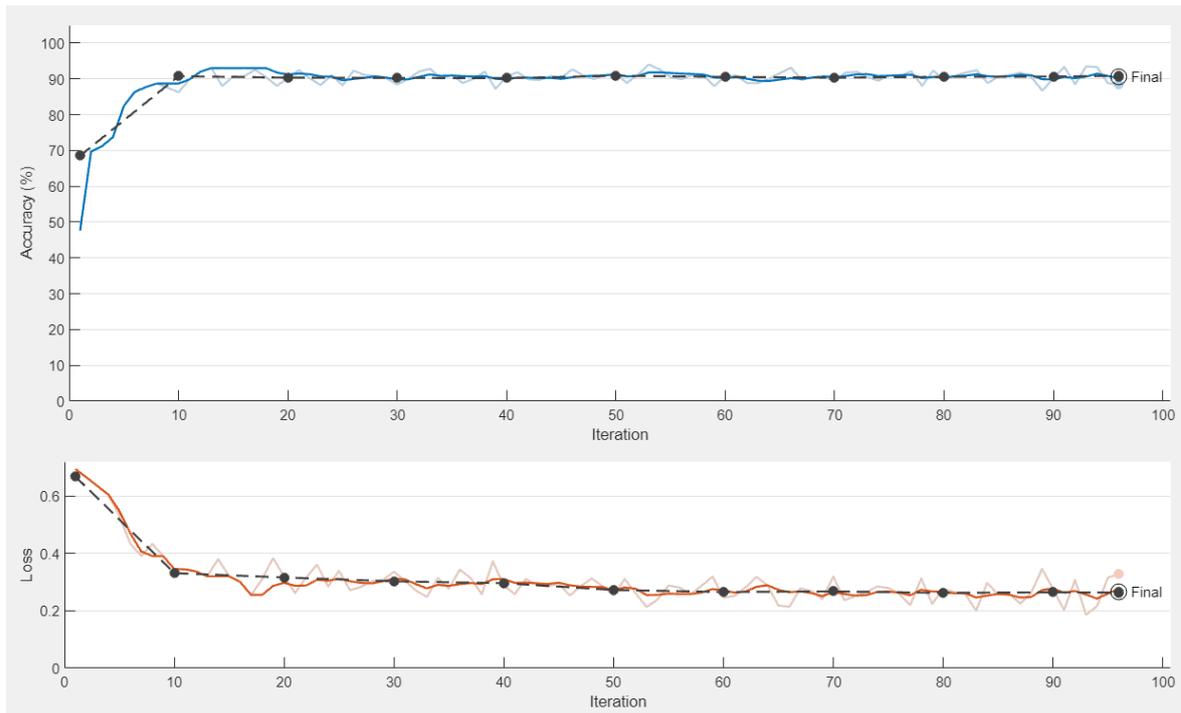


Figura 47. Resultados de los experimentos.
Fuente: Elaboración propia.

Dentro de las variables destaca `SequenceLength`. Esta variable designa la longitud de la secuencia, o ventana, de datos con la que se recorre por completo a la señal con el objetivo de extraer sus características. Como se menciona en capítulos anteriores, el radioenlace satelital genera datos con una media de 15 segundos. Si la variable dicha toma un valor de 150, la ventana utilizada es de 150 veces esos 15 segundos (37,50 minutos). Por otro lado, se encuentra la variable `sequenceOverlap`, esta determina el porcentaje con el que se solapan dos ventanas consecutivas. Este solape se iniciará con el 75%. En el bucle principal del programa, se localizan las variables `pctTr`, que marcará el porcentaje de los datos usados para el entrenamiento, `maxSeqTr`, indicando la longitud de la secuencia con la que se entrena, `nMesTr`, que marca el número de meses anteriores con los que se entrena la red y, finalmente, `nIter`, con el que se indica el número de iteraciones que debe hacer una determinada red la fase de entrenamiento y testeo.

El bucle principal del programa se realiza en un primer experimento dejando la variable `nIter` a 50. Este valor se mantiene constante ya que se considera suficiente como para obtener un valor simbólico dentro de un análisis estadístico de las métricas de calidad. De momento, no se han realizado experimentos con valores mayores de 50 debido a que el cómputo necesario para la creación de la red sería alto. No obstante, se justifica este valor por medio de la Figura 48 donde se puede apreciar que, pese a aumentar el número de iteraciones que la red entrena y testea, se puede ver que, al llegar a las 50 iteraciones, la red ya está bastante estable habiendo convergido a un valor final de validación. En caso de que llegado a 50, el valor de validación de la red fluctuara de la misma manera que en las primeras 30 iteraciones, se consideraría aumentar este número [3].

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS



*Figura 48. Evolución de la validación del modelo.
Arriba: Porcentaje de validación por cada iteración. Abajo: Pérdidas por cada iteración.
Fuente: Elaboración propia.*

Finalmente, se realiza una primera búsqueda del mejor modelo para la tarea variando el porcentaje de datos usados para el entrenamiento entre el 60% y el 90% en saltos del 10%, luego, se mira la eficiencia del modelo con secuencias de entrenamiento que van desde 500 hasta 1000 en pasos de 100 y finalmente se verifica con cuántos meses anteriores resulta que el modelo da mejores resultados empezando desde 1 mes anterior hasta 6. Tras la ejecución de este programa se generan tantas carpetas como experimentos dando lugar a un total de 144 experimentos cada uno con sus 50 redes y sus 50 métricas de calidad registradas en el xlsx como se demuestra en la Figura 49.

Nombre	Tamaño	Comprimido	Tipo
Test pctTr70 maxSeqTr900 nIter50 nMesTr3	2.816.201	2.809.553	Carpeta de archivos
Test pctTr70 maxSeqTr900 nIter50 nMesTr4	2.818.720	2.812.056	Carpeta de archivos
Test pctTr70 maxSeqTr900 nIter50 nMesTr5	2.816.812	2.810.137	Carpeta de archivos
Test pctTr70 maxSeqTr900 nIter50 nMesTr6	2.817.863	2.811.141	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr1	2.801.405	2.794.431	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr2	2.810.707	2.804.149	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr3	2.815.802	2.809.087	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr4	2.817.684	2.811.011	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr5	2.818.225	2.811.537	Carpeta de archivos
Test pctTr70 maxSeqTr1000 nIter50 nMesTr6	2.820.176	2.813.409	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr1	2.802.568	2.795.704	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr2	2.808.405	2.801.493	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr3	2.811.385	2.804.511	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr4	2.813.628	2.806.989	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr5	2.813.735	2.807.152	Carpeta de archivos
Test pctTr80 maxSeqTr500 nIter50 nMesTr6	2.813.851	2.807.318	Carpeta de archivos
Test pctTr80 maxSeqTr600 nIter50 nMesTr1	2.801.686	2.794.866	Carpeta de archivos
Test pctTr80 maxSeqTr600 nIter50 nMesTr2	2.808.028	2.801.292	Carpeta de archivos
Test pctTr80 maxSeqTr600 nIter50 nMesTr3	2.816.229	2.809.456	Carpeta de archivos
Test pctTr80 maxSeqTr600 nIter50 nMesTr4	2.814.734	2.807.940	Carpeta de archivos

Seleccionado 1 carpeta, 2.815.957 bytes Total 144 carpetas, 405.033.738 bytes

Figura 49. Número de experimentos realizados.
Fuente: Elaboración propia.

7.1.2. Estructura del algoritmo

Como ya se ha visto, el primer paso para la obtención de cada una de esas carpetas es la declaración de los parámetros con los que la red va a ser creado. Una vez iniciadas las variables principales, se llama a un segundo script que se puede encontrar en el Anexo 8 denominado `Detector_Test_CTR_COMBINADO_Gabriel_01.m` el cual se encarga de llamar al resto de funciones necesarias para la creación del umbral, el diseño de la red neuronal, el entrenamiento de esta y la obtención de métricas para su evaluación. Para que sea más sencillo el flujo que sigue este programa, se dividirá en secciones.

Sección 1. Creación del fichero `xlsx`.

Se genera un fichero `xlsx` dónde se guardarán los parámetros de calidad. Esto se hace para que sea fácilmente evaluable el desempeño de la red para la detección de lluvia.

Código 1. Creación `xlsx` de `Detector_Text_CTR_COMBINADO_Gabriel_01.m`.
Fuente: Elaboración propia.

```

81 filename = strcat(Dir_Resu, '\Tabla_Metricas', '.xlsx');
82 A_Met     = {'Mes', 'Sensibilidad', 'Especificidad', 'Exactitud', 'F1-Score'};
83 A_Con     = {'TN', 'FP', 'FN', 'TP', 'Tiempo'};

```

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

Como se puede observar en el Código 1, se especifica una variable llamada filename que recoge la ruta del archivo dónde se va a guardar, el nombre que va a tener el experimento y el tipo de archivo como.xlsx. Las siguientes dos variables, guardan el nombre de los parámetros de las métricas a extraer en función de los resultados de la red. En otras palabras, indican el encabezado de la tabla. Dentro de este encabezado se encuentra la variable Tiempo, este valor se encarga de medir el tiempo de ejecución del experimento. Se agrega con el propósito de ver que elementos han ayudado a mejorar el rendimiento del modelo.

Más adelante, se añadirá el parámetro de Cohen's kappa. Este se realiza posteriormente porque requiere de un cálculo previo ya que servirá como elemento que determine la calidad de la red [33]. El cálculo de este parámetro se realiza una vez la ejecución haya terminado llamando al script adjunto al Anexo 9. El script ParametrosKappa.m realiza una búsqueda de un experimento y toma los valores de la tabla para calcular el parámetro Cohen's kappa y realizar un valor medio y su desviación para elegir el mejor modelo como se puede ver en la extracción de su script del Código 2.

*Código 2. Cálculo del Cohen's kappa de ParametrosKappa.m.
Fuente: Elaboración propia.*

```
% Datos separados de la matriz de confusion de cada hoja
TN = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rTN);
FP = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rFP);
FN = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rFN);
TP = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rTP);

% Calculo de los parametros kappa
numerador = 2*(TP.*TN-FN.*FP);
denominador = (TP+FP).*(FP+TN)+(TP+FN).*(FN+TN);
kappa = numerador./denominador;

%Media y desviacion de kappa
mean_kappa = mean(kappa);
sigma_kappa = std(kappa);

%Guardado de los parametros kappa
title_kappa = 'kappa';
title_mean = 'Media';
title_sigma = 'Desviacion';
```

Sección 2. Entrenamiento del modelo.

El siguiente paso del flujo del programa entra en un bucle que de recorrer todos los meses donde comienza a cronometrar el tiempo de ejecución de cada iteración por medio de la función tic-toc de MATLAB para quedar representado en la columna Tiempo de las métricas de calidad que se han escogido para el xlsx final.

Más adelante comienza todo el procesado de los datos el que se había mencionado en el Capítulo 5 obteniendo finalmente un umbral funcional. Cuando termina su ejecución, se etiquetan los datos de forma aleatoria para ser usados en la fase de entrenamiento o de testeo. Los datos se agrupan en secuencias de tamaño SequenceLength y se reparten de forma que se cumpla que, en tanto por uno, el pctTR de los datos sean para entrenamiento y el resto para la comprobación del modelo.

Finalmente se entrena la red. Para entrenar una red en MATLAB se hace uso de la función `trainNetwork(sequence,response,layers, options)`. Esta requiere de cuatro parámetros para crear la red y entrenarla. Estos parámetros se dividen en `sequence`, conjunto de datos de entrada, `response`, conjunto de datos de salida, `layers`, arquitectura de la red y `options`, configuraciones del modelo. La entrada serán los niveles de potencia, la salida, son las clasificaciones hechas de la base de datos como eventos de lluvia, 1, o eventos de no lluvia, 0. Los otros dos parámetros necesarios son a elección del diseñador.

El parámetro `layer` como se ve en el Código 3 determina la arquitectura que tendrá el modelo a crear. Las opciones escogidas para este parámetro son [34]:

*Código 3 Parámetro layers de Detector_Text_CTR_COMBINADO_Gabriel_01.m.
Fuente: Elaboración propia.*

```
layers = [ ...  
    sequenceInputLayer( size(trainFeatureCell,1),"Name","Entrada" )  
    bilstmLayer(200,"OutputMode","sequence","Name","Capa 1")  
    bilstmLayer(100,"OutputMode","sequence","Name","Capa 2")  
    fullyConnectedLayer(2,"Name","fullyConnected")  
    softmaxLayer("Name","Softmax")  
    classificationLayer("Name","Classification")  
];
```

- `sequenceInputLayer`. Capa encargada de recibir los datos de entrada. Tiene dos parámetros de entrada donde el primero es dimensión de los datos de entrada y el segundo es el nombre de la capa.
- `bilstmLayer`. Tipo de dirección de la red LSTM para recordar información. Las redes LSTM unidireccionales conservan información pasada ya que solo están conectadas a las capas anteriores. Las LSTM bidireccionales, al estar conectadas a la entrada por las capas anteriores como posteriores, puede conservar información procedente de ambas direcciones. En este caso, este parámetro se ve duplicado porque el modelo se ha diseñado de tal forma que son dos capas `bilstmLayer` contiguas. Los puntos que contienen cada una de las redes con respectivamente 200 y 100 que corresponden al número de celdas ocultas encargadas de la cantidad de información a mantener en memoria. `OutputMode` es el modo de salida de los datos siendo del tipo secuencial donde se devuelve todas las secuencias, aunque se da la opción de que pueda ser del tipo `last` para devolver solamente la última secuencia de datos. Finalmente, igual que en el caso anterior, se da la opción de ponerle nombre a la capa.
- `fullyConnectedLayer`. Conecta todas las salidas de la capa anterior y con las entradas de la siguiente. El valor 2 referencia al tamaño de salida de la red LSTM en este caso eventos de lluvia y no lluvia convirtiendo al modelo en un clasificador binario.
- `softmaxLayer`. Capa que usa la función `softmax`. Esta función convierte un vector de números reales en otro con valores de entre 0 y 1. Esto se suele utilizar para salidas que quieran interpretarse como probabilidades en tanto por uno. Para el estudio realizado, lo cerca que está de la clase lluvia o la clase no lluvia.
- `classificationLayer`. Capa final por antonomasia encargada de realizar la clasificación.

El parámetro `options` como se ve en el Código 4 especifica los hiperparámetros del modelo que tendrá el modelo a crear. Las opciones escogidas para este parámetro son [34]:

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

Código 4. Parámetro options de Detector_Text_CTR_COMBINADO_Gabriel_01.m.
Fuente: Elaboración propia.

```
options = trainingOptions("adam", ...
    "MaxEpochs",maxEpochs, ...
    'ExecutionEnvironment','auto', ...
    "MiniBatchSize",miniBatchSize, ...
    "Shuffle","every-epoch", ...
    "Verbose",0, ...
    "SequenceLength",sequenceLength, ...
    "ValidationFrequency",10, ...
    "ValidationData",{validationFeatureCell,validationLabelCell}, ...
    "Plots","training-progress", ... %"training-progress"
    "LearnRateSchedule","piecewise", ...
    "LearnRateDropFactor",0.1, ...
    "LearnRateDropPeriod",5);
```

- MaxEpochs. Número de veces que se ejecutarán los algoritmos de forwardpropagation y backpropagation. En cada ciclo o epoch, todos los datos de entrenamiento pasan por la red neuronal. Forwardpropagation es la forma en la que se crean las predicciones, es decir, los datos de entrenamiento pasan por las neuronas hasta el final. Backpropagation, cambia la función de pérdida de las neuronas para mejorar las predicciones.
- MiniBatchSize. Método en que se empleará el descenso del gradiente. Actualiza los pesos y coeficientes disminuyendo el error entre valor predicho y valor real. En todos los experimentos realizados, comenzará con un valor de 32.
- SolverName. Especifica el solucionador de la red neuronal. Puede valer sgdm para que use el descenso de gradiente estocástico con impulso (Stochastic Gradient Descent with Momentum), rmsprop para usae el optimizador RMSPropr (Root Mean Square Propagation) donde usa ventana para tener en cuenta solo los gradientes recientes, Adam, actualización del optimizador RMSProp fijando la ratio de aprendizaje del SGD.
- ExecuteEnvironment. Hardware empleado para el entrenamiento de la red (auto, cpu, gpu).
- Shuffle. Configuración para mezclar los datos entre cada iteración. Puede ser once (una vez), never (nunca), every-epoch (cada iteración).

- `Verbose`. Muestra el progreso del entrenamiento en la ventana de comando. 1 se muestra, 0 no se muestra.
- `SequenceLength`. Rellena, trunca o divide las secuencias de entrada. Tiene tres argumentos distintos que son `longest`, dónde cada secuencia de datos es tan larga como la secuencia de mayor longitud, `shortest`, lo mismo que la anterior, pero con la más corta o, finalmente incluir un valor entero positivo que especifique la longitud de la secuencia. En el caso de los experimentos de decide del tamaño `sequenceLength`.
- `ValidationFrequency`. Cada cuanta iteración se realiza una validación. En este trabajo se elige el método de validación predeterminado de `train/test Split`, que valida con datos aleatorios del entrenamiento.
- `Plots`. Muestra el proceso de entrenamiento de forma gráfica como en la Figura 48.
- `LearnRateSchedule`. Permite que a lo largo del entrenamiento la ratio de aprendizaje baje o no, en caso de permitirlo se debe de pasar el argumento “`piecewise`”.
- `LearnRateDropPeriod`. Número de iteraciones al que actualiza la ratio de aprendizaje
- `LearnRateDropFactor`. Configura el factor de bajada de la ratio de aprendizaje.

Siendo ya configurado el modelo se pasa a la función antes mencionada. A la hora de entrenar un modelo es importante tener en cuenta de que no suceda un fenómeno llamado `overfitting` o sobreentrenamiento [8]. Este fenómeno ocurre cuando una red se adapta demasiado a los datos de entrenamiento provocando que de cara a futuro sólo sea capaz de interpretar los datos con los que ha sido entrenado. De esta forma el modelo es incapaz de reconocer nuevos datos de entrada [30]. Este caso indeseado suele reconocerse fácilmente cuando tiene resultados casi perfectos y una sensibilidad muy alta. Normalmente suele verse también tras la fase de testeo que, al introducirse nuevos datos, la red clasifica incorrectamente [7]. Este fenómeno se puede prevenir estableciendo más porcentaje de datos para la fase de prueba, teniendo clases equilibradas (para que la red no sea experta en reconocer un único tipo de clase), tener una base de datos amplia para que la fase de prueba sea lo más extensa posible y finalmente, cambiando la configuración de los hiperparámetros [29].

Sección 3. Prueba del modelo.

Una de las partes más importantes a la hora de crear un modelo, es comprobar su eficacia [6]. Esto se hace en la fase de prueba del modelo dónde se le da al modelo unas secuencias de entrada totalmente nuevas para determinar lo preciso que es. Para ello, se crea una matriz de confusión, generada por entradas desconocidas por el modelo durante la fase de entrenamiento, y se emplean las métricas de calidad mencionadas anteriormente como se ve en el Código 5.

*Código 5. Matriz de confusión de Detector_Text_CTR_COMBINADO_Gabriel_01.m.
Fuente: Elaboración propia.*

```

%% MÉTRICAS DE CALIDAD
cm = confusionchart(detected_Labels_Dete,detected_Labels_Real,"title","Validation Accuracy");
[c,m] = size(cm);
MatrizConfusion = cm.NormalizedValues;
[c,m] = size(MatrizConfusion);
close; % Cerramos la ventana con la matriz de confusión
if c == 1
    if ( sum(detected_Labels_Real)==length(detected_Labels_Real) ) % Todas las muestras son de Sí lluvia
        TP = MatrizConfusion(1,1);
        TN = 0;
        FN = 0;
        FP = 0;
    elseif (sum(detected_Labels_Real)==0 ) % Todas las muestras son de NO lluvia
        TP = 0;
        TN = MatrizConfusion(1,1);
        FN = 0;
        FP = 0;
    end
else
    TN = MatrizConfusion(1,1);
    FN = MatrizConfusion(2,1);
    TP = MatrizConfusion(2,2);
    FP = MatrizConfusion(1,2);
end
end

```

Finalmente, una vez obtenido todas las métricas, se registra la información en el archivo xlsx creado con anterioridad y se continúa con este procedimiento hasta recorrer todos los meses y parámetros distintos que se colocaron en un inicio. Esto da como resultado una tabla como la que se ve en la Figura 50.

	A	B	C	D	E	F	G	H	I	J	K
1	Mes	Sensibilidad	Especificidad	Exactitud	F1-Score	TN	FP	FN	TP	Tiempo	kappa
2	2021-09	0,9358209	0,84047267	0,88789903	0,89252669	569	108	43	627	0,169371715	0,775904411
3	2021-09	0,93979442	0,85735736	0,89903489	0,9039548	571	95	41	640	0,340039557	0,797864241
4	2021-09	0,94014599	0,86253776	0,90200445	0,90704225	571	91	41	644	0,514645193	0,803702845
5	2021-09	0,93703148	0,83823529	0,88715664	0,89158345	570	110	42	625	0,68675862	0,774512007
6	2021-09	0,93916914	0,84843982	0,89383816	0,89850958	571	102	41	633	0,853572502	0,787661923
7	2021-09	0,94047619	0,84740741	0,89383816	0,89836532	572	103	40	632	1,024103177	0,78771949
8	2021-09	0,93265007	0,85240964	0,89309577	0,89844852	566	98	46	637	1,197355173	0,785915791
9	2021-09	0,94065282	0,84992571	0,89532294	0,89992903	572	101	40	634	1,369470015	0,790631687
10	2021-09	0,93880597	0,84342688	0,8908686	0,89537367	571	106	41	629	1,540692072	0,781840718

*Figura 50. Ejemplo de métricas de calidad de un modelo.
Fuente: Elaboración propia.*

7.1.3. Resultados

Para representar los resultados de los 144 experimentos, se realiza un histograma por cada métrica de calidad para acomodar los 64800 casos de estudio realizados en total (144 experimentos con 9 meses de estudio con 50 redes por mes) entre todos los experimentos creados. Las siguientes figuras tienen la función de mostrar la evolución global de todos los experimentos con el fin de ilustrar la calidad de las distintas redes [3]. Comenzando con la Figura 51, se puede observar cómo la evolución global de los distintos sistemas producen una gran cantidad de redes que no pasan del 50% de sensibilidad. La búsqueda de la mejor red, dado a que se trata en parte de un proyecto para generar una alarma, será alguna de las redes que forman parte de la derecha de la gráfica [6].

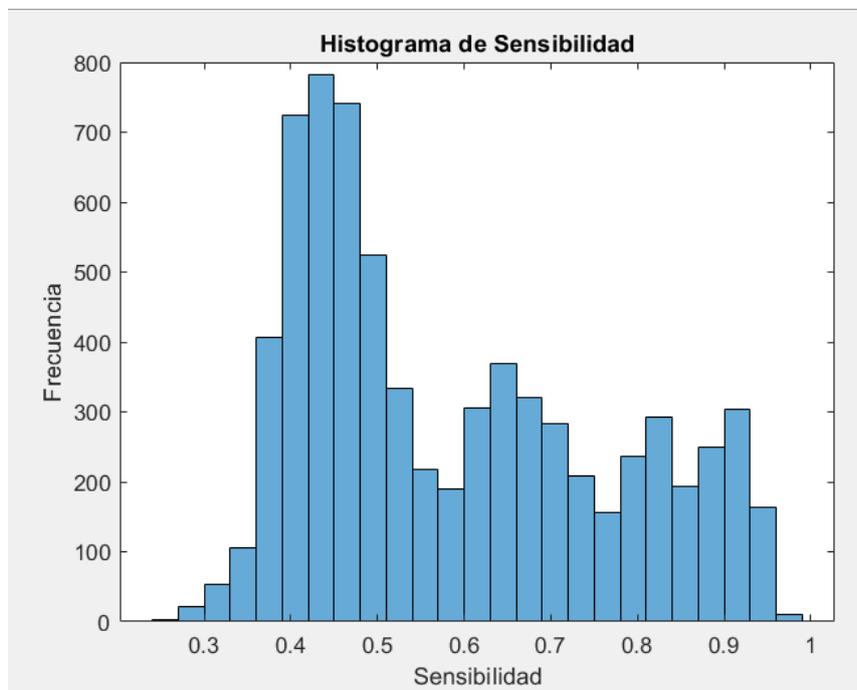


Figura 51. Histograma de Sensibilidad del detector.
Fuente: Elaboración propia.

Continuando con la Figura 52 se obtienen buenos parámetros en la mayoría de las redes creadas para la detección, demostrando que, en términos generales, las redes son capaces de cumplir la función para la que fueron diseñadas [8].

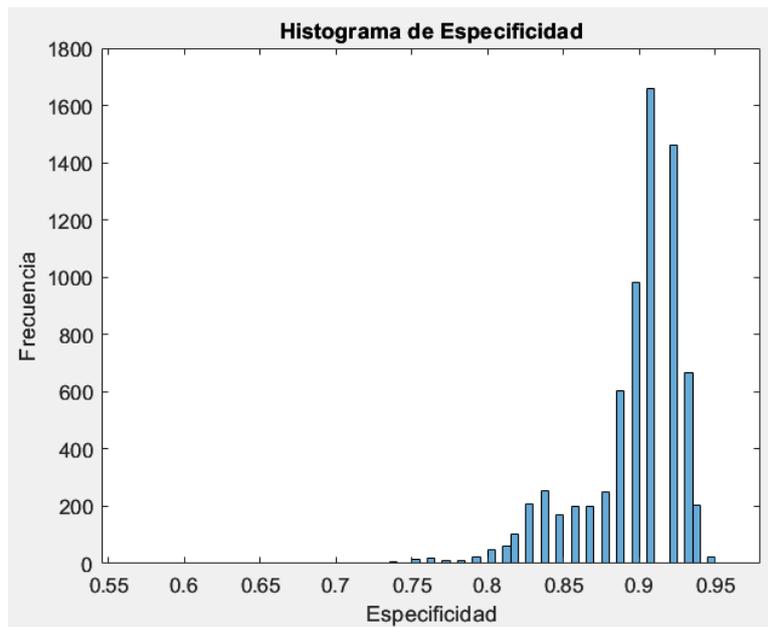


Figura 52. Histograma de Especificidad del detector.
Fuente: Elaboración propia.

Pasando a la Figura 53, se demuestran ratios similares a los expuestos con anterioridad con la especificidad. La mayor parte de las redes creadas, superan el 80% de exactitud.

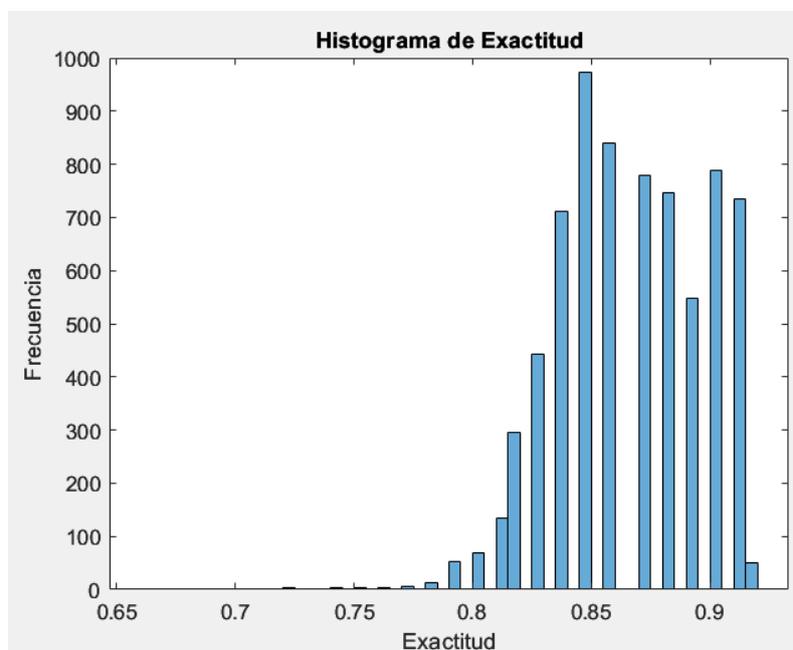


Figura 53. Histograma de Exactitud del detector.
Fuente: Elaboración propia.

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

Finalmente, los parámetros que se muestran en la Figura 54 y Figura 55 revelan la calidad general de las redes graficando el F1-Score y el Cohen's kappa respectivamente. Como se puede apreciar, ambas gráficas guardan cierta similitud dado a que miden de forma general la relación entre aciertos y fallos de la red entrenada [30]. Por lo general muestran escalas bastante bajas respecto a las deseadas, pero con algunos casos interesantes de redes que se pueden considerar como la mejor.

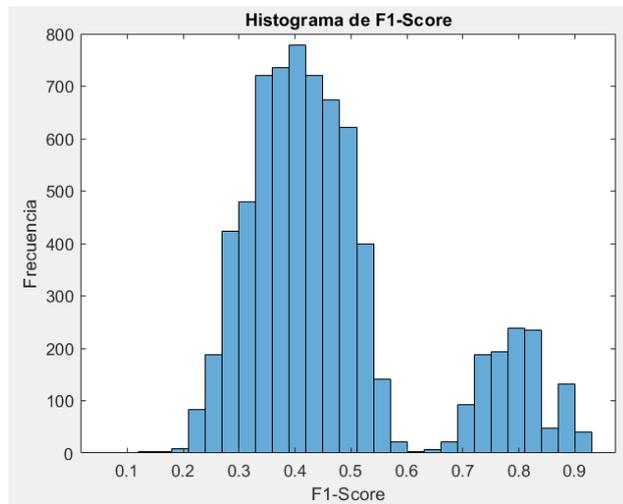


Figura 54. Histograma de F1-Score del detector.
Fuente: Elaboración propia.

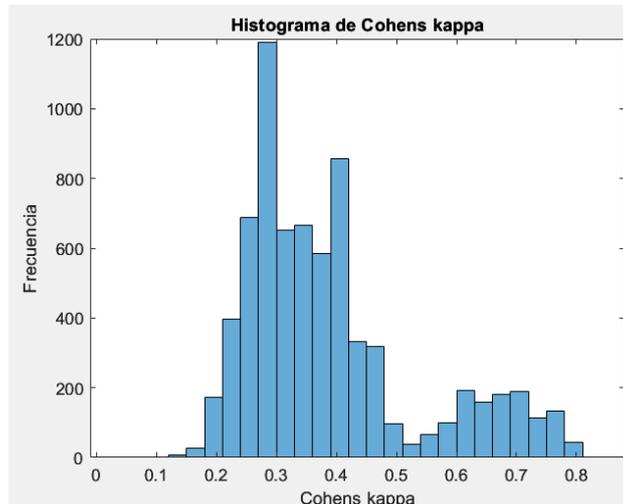


Figura 55. Histograma de Cohen's kappa del detector.
Fuente: Elaboración propia.

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

Una vez visto el estado global de todos los experimentos, se procede a buscar el mejor caso. De todas las redes creadas se tomará el valor más alto del parámetro Cohen's kappa como la mejor red [33]. Para ello, se crea el programa `BuscarMejorNN.m` que se puede encontrar en el Anexo 10. Este programa se introduce en todas las carpetas dónde se han guardado los resultados de los experimentos y realiza una comparación entre el valor promedio del parámetro Cohen's kappa de cada uno de los meses de cada experimento como se ve en el Código 6. De esta forma, se consigue localizar aquellos hiperparámetros que tras 50 iteraciones han dado los mejores resultados en lugar de basarse en un caso fortuito que de casualidad haya maximizado este valor. En otras palabras, si suponemos un caso en el que unos hiperparámetros logra que en una de las iteraciones se obtenga un Cohen's kappa de casi 1 mientras que en las restantes iteraciones consigue un valor que clasifica a la red de concordancia débil o menor, este caso queda descartado considerándose como un golpe de suerte en lugar de una demostración de que esos son los mejores hiperparámetros de entre todos los experimentos. Por otro lado, si se observa que de entre todas las iteraciones se consigue una clasificación de concordancia buena o mejor, se considera esos hiperparámetros como candidatos a ser los mejores de toda la experimentación.

*Código 6. Búsqueda del mejor Cohen's kappa del BuscarMejorNN.m.
Fuente: Elaboración propia.*

```
%% Cálculo de los parametros kappa
for i=1:Lsh
    % Datos asociados a la hoja de datos actual
    act_kappa = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', mean_kappa);
    act_std   = readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', std_kappa);
    act_expe  = strcat(Nom_Expe, ' en ', all_sheetnames{i});

    % Comparacion entre parametros kappa
    if act_kappa > max_kappa
        max_kappa = act_kappa;
        max_std   = act_std;
        max_expe  = act_expe;
        max_SeqTr = maxSeqTr;
        max_nMesTr = nMesTr;
        max_Mes   = all_sheetnames{i};
        max_Mes_i = i;
    end
end
```

Una vez finalizada la ejecución de este código devuelve el nombre del experimento que contiene los hiperparámetros que han resultado en todos sus casos, un Cohen's kappa mayor que el promedio. Siendo en este caso, el experimento de nombre `Test pctTr80`

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

maxSeqTr500 nIter50 nMesTr4. Para facilitar su búsqueda y el flujo del código, se ha decidido que el nombre de cada experimento se componga de los hiperparámetros utilizados. En este caso, se puede entender entonces, que, para la arquitectura diseñada, los mejores hiperparámetros encontrados son:

- Porcentaje de entrenamiento del 80%.
- Secuencia para el entrenamiento de longitud 500.
- Número de iteraciones de 50.
- Meses anteriores para el entrenamiento de 4.

Los valores de estas redes se pueden ver en la Figura 56 que muestra los resultados de las diez primeras iteraciones de la red con estos hiperparámetros para septiembre de 2021.

	A	B	C	D	E	F	G	H	I	J	K
1	Mes	Sensibilidad	Especificidad	Exactitud	F1-Score	TN	FP	FN	TP	Tiempo	kappa
2	2021-09	0,9358209	0,84047267	0,88789903	0,89252669	569	108	43	627	0,169371715	0,775904411
3	2021-09	0,93979442	0,85735736	0,89903489	0,9039548	571	95	41	640	0,340039557	0,797864241
4	2021-09	0,94014599	0,86253776	0,90200445	0,90704225	571	91	41	644	0,514645193	0,803702845
5	2021-09	0,93703148	0,83823529	0,88715664	0,89158345	570	110	42	625	0,68675862	0,774512007
6	2021-09	0,93916914	0,84843982	0,89383816	0,89850958	571	102	41	633	0,853572502	0,787661923
7	2021-09	0,94047619	0,84740741	0,89383816	0,89836532	572	103	40	632	1,024103177	0,78771949
8	2021-09	0,93265007	0,85240964	0,89309577	0,89844852	566	98	46	637	1,197355173	0,785915791
9	2021-09	0,94065282	0,84992571	0,89532294	0,89992903	572	101	40	634	1,369470015	0,790631687
10	2021-09	0,93880597	0,84342688	0,8908686	0,89537367	571	106	41	629	1,540692072	0,781840718
11	2021-09	0,95718654	0,84271284	0,8982925	0,90136789	584	109	28	626	1,709698423	0,797121382

Figura 56. Mejores parámetros de detección.
Fuente: Elaboración propia.

Una vez completada la tarea de buscar los mejores hiperparámetros, se busca a su vez, el mejor caso de la mejor red para volver a crearla y entrenarla [8]. Este proceso se hace mediante el programa CrearMejorNN_V2.m que sigue la misma metodología que hacía el programa Detector_Test_CTR_COMBINADO_Gabriel_01.m pero únicamente creando una sola red con los mejores hiperparámetros encontrados. De esta forma, se vuelve a generar esa misma red y se evalúa en los distintos escenarios del resto de meses. Los resultados finales de este proceso se muestran en la Figura 57.

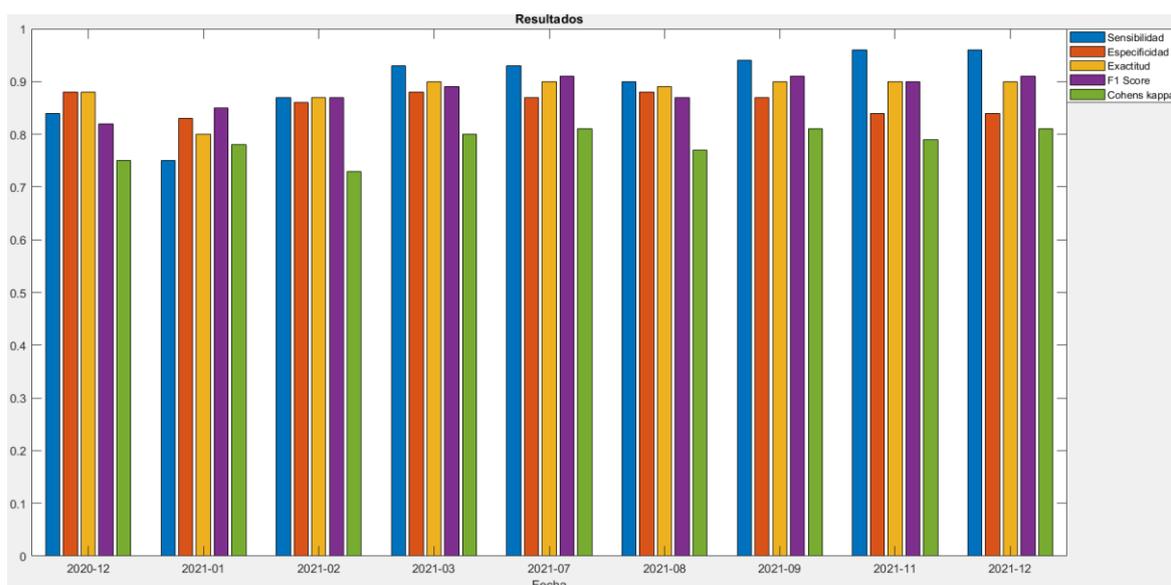


Figura 57. Resultados del mejor detector.
Fuente: Elaboración propia.

Como se puede apreciar en la Figura 57, los resultados de los experimentos han dado lugar a un modelo que resulta bastante adecuado para cada uno de los meses [7]. Siendo un parámetro importante la sensibilidad dado a que su condición de alarma debe procurar una falsa alarma de lluvia antes de dejar pasar por alto un fenómeno meteorológico que pueda ser un peligro. Por otro lado, se encuentran la especificidad y la exactitud de los modelos que por lo general suelen ser bastante altas, lo suficiente como para que esta red permita aportar datos relevantes del terreno al sistema VIMETRI-MAC. Finalmente, se observan los parámetros de F1-Score y Cohen's kappa que clasifican a todos los modelos, como mínimo, con una concordancia buena [33]. De esta forma, se demuestra que el modelo finalmente expuesto resulta viable para su uso como alarma de alerta por precipitaciones basándose en la atenuación del nivel de potencia de los radioenlaces.

7.2. Algoritmo de predicción

Para la creación del algoritmo de predicción, se seguirá la misma metodología que se siguió durante el proceso de creación del detector. Se puede suponer que un detector es en realidad un predictor de presente [30]. De esta forma, la fase de experimentación seguirá con el mismo flujo de desarrollo, pero en lugar de probar las redes en base a los

resultados de la función `trainNetwork()` de MATLAB, se hará un estudio de la salida de la función `predict()`.

7.2.1. Variables principales

Las variables principales de este segundo modelo seguirán siendo las mismas que en el caso del primero. En `Lanza_Detector_Lluvia_COMBINADO8_4.m` se mantendrán constantes los valores de los experimentos anteriores. No obstante, ya una vez obtenido los parámetros que resultan ser los mejores para la detección, se planteará un estudio sobre su eficacia frente a la tarea de detección. Esto se debe a que es lógico pensar que las características del mejor detector puedan dar resultados funcionales bajo la tarea de predicción con el entrenamiento adecuado.

El entrenamiento aplicado a estas nuevas redes de predicción mantienen los mejores parámetros del detector, pero se varía el número de meses anteriores con los que se entrena y se cambia también la longitud de la secuencia en la que se divide la señal para comprobar la mejoría de los resultados entre un modelo de predicción con una longitud de secuencia de 150 y otra de 350 al igual que la secuencia de entrenamiento a 750 por valor típico en los predictores ya que estos necesitan una ventana de tiempo mayor para predecir [16]. Es por ello, que los parámetros de inicio serán los siguientes:

- Porcentaje de entrenamiento del 80%.
- Secuencia para el entrenamiento de longitud 750.
- Número de iteraciones de 50.
- Meses anteriores para el entrenamiento de 1 a 6.
- Longitud de las secuencias en las que se divide la señal 150 y 350.

7.2.2. Estructura del algoritmo

Al igual que pasaba con el caso anterior del detector, en primer lugar, se inicializan las variables necesarias en el programa `Lanza_Detector_Lluvia_COMBINADO8_4.m`. La diferencia en este caso es que se llama a un nuevo programa denominado `predictor_v1.m` que se encuentra en el Anexo 11. Este nuevo programa se elabora basándose en el programa `Detector_Test_CTR_COMBINADO_Gabriel_01.m` y adaptándolo para que aporte las métricas durante la fase de predicción.

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

En segundo lugar, una vez que se inicialicen las variables del bucle principal como se muestra en el Código 7 se llama al script predictor_v1.m. Este segundo script vuelve a entrar en un bucle que recorre todos los meses cronometrando el tiempo de ejecución de cada iteración por medio de la función tic-toc de MATLAB. Más adelante, continua con todo el procesamiento de los datos mencionado en el Capítulo 5 para obtener un umbral funcional. Cuando termina su ejecución, vuelve a etiquetar los datos de forma aleatoria para ser usados en la fase de entrenamiento o de testeo. Los datos se agrupan en secuencias de tamaño SequenceLength que, en este caso, se estudiará su influencia variando entre 150 y 350, y se reparten entre datos que se destinan a la fase de entrenamiento y prueba cumpliendo la razón impuesta de 80% de los datos para entrenar y el 20% de los datos para la comprobación de su eficacia.

*Código 7. Variables principales de Lanza_Detector_Lluvia_COMBINADOS_4.m.
Fuente: Elaboración propia.*

```
5  
6  
7  _____
8
9
10
11
```

```
for sequenceLength = 150:200:350
    for nMesTr=1:1:6

        % Predictor
        pctTr      = 0.8;
        maxSeqTr   = 500;
        nIter      = 50;
```

En tercer lugar, se crea la red y se entrena. Para entrenar una red en MATLAB se vuelve hacer uso de la función trainNetwork(sequence,response,layers, options). Estos cuatro parámetros para crear la red y entrenarla se mantienen igual al modelo de detección.

En cuarto lugar, se pone a prueba el modelo creado, pero estudiando la matriz de confusión resultante de la función ypred = predict mdl,Xnew) de MATLAB. Esta función se usará de forma que añadiendo la red a ser evaluada como predictor y los datos de salida se devuelvan las predicciones para un instante posterior. En el caso de estudio, un instante posterior significa 4 muestras hacia delante. Como cada muestra del sistema se obtiene cada 5 minutos, en conclusión, se está haciendo una predicción a 20 minutos en el futuro, tiempo suficiente para poner en alerta a los servicios de emergencia [2].

La función predict en MATLAB es una función utilizada en el contexto de modelos de aprendizaje automático para realizar predicciones basadas en datos de entrada. La

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

función se utiliza comúnmente con modelos de regresión y clasificación como en el caso de estudio de un detector de precipitaciones en base al nivel de potencia de los radioenlaces ya que puede ser interpretada como un clasificador binario de eventos de lluvia o no lluvia. Los parámetros de esta función son [34]:

- mdl. El modelo previamente entrenado.
- Xnew. Matriz de características o conjunto de datos de entrada para los cuales se desean realizar las predicciones.
- ypred. Matriz que contiene las predicciones resultantes.

Como ya se ha comentado, la salida de esta función son las predicciones del modelo introducido frente a las mismas entradas con las que se entrenó el detector. En este caso se hace uso de la red previamente entrenada llamada rainDetectNet y como matriz de datos se agrega el conjunto de datos finales validados que enlazan los niveles de potencia con el fenómeno meteorológico del momento de la muestra de nombre testFeatureCell. Las salidas del predictor se guardan en la variable predictedLabels. En el Código 8 se muestra este proceso junto a otras variables que se usarán durante la fase de prueba.

Código 8. Fase de predicción de predictor_v1.m
Fuente: Elaboración propia.

```
%% Predictor
% Realiza la predicción
predictedLabels = predict(rainDetectNet, testFeatureCell);
pred_tmp       = {};
pred_confmat   = {};
pred_confmat_diver = {};
testlabelCell = testlabelCell';
```

La finalización de la fase de predicción genera un conjunto de valores que no son manejables ya que, al hacer una predicción de toda la base de datos, el número de predicciones que hace es igual de grande. Por ello, para obtener una idea de su utilidad final, se elabora la matriz de confusión de cada predicción y se obtiene un promediado de cada una de las 50 redes que se generan por experimento. Para ello se guardan los valores de predicción como una única matriz por caso y por experimento. Al hacer la predicción con toda la base de datos, surge el problema de una descompensación en las clases, siendo los eventos de no lluvia muy superiores a los de lluvia. En este caso, para corroborar el

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

buen funcionamiento del modelo, se generan dos tipos de matriz de confusión. Como se puede ver en el Código 9 existe una primera matriz de confusión que contiene todos los casos y luego otra bajo el sufijo `diver` que reúne todas las matrices distintas de 150,0,0,0. El valor de matriz 150,0,0,0 se puede entender como que el modelo ha predicho a la perfección la siguiente secuencia de 150 muestras como eventos de no lluvia. Esta predicción perfecta, en principio no debe ser considerada ya que en raras ocasiones aparece un evento de lluvia en la base de datos [7]. Dada la situación de que existen dos clases extremadamente descompensadas, nace la oportunidad de experimentar acerca de cómo de diferente sería la matriz de confusión si se contaran todos los casos, o aquellos casos dónde la matriz de confusión muestre un resultado diverso (de ahí el sufijo `diver`) a una secuencia sin eventos de lluvia, es decir, la matriz de confusión secundaria sólo cuenta en su promediado las matrices distintas de 150,0,0,0.

Código 9. Matrices de confusión predictor_v1.m
Fuente: Elaboración propia.

```
% Matriz de confusion
pred_confmat{ii} = confusionmat(pred_tmp{ii},testLabelCell{ii+1});

% Matriz de confusion de secuencias distintas a solo casos de lluvia (150,0;0,0)
if pred_confmat{ii}(1,1) == 150
else
    pred_confmat_diver{ii}=pred_confmat{ii};

    % Resultado sumado de las matrices distintas a (150,0;0,0)
    total_diver_TN = total_diver_TN + pred_confmat_diver{ii}(1,1);
    total_diver_FN = total_diver_FN + pred_confmat_diver{ii}(2,1);
    total_diver_FP = total_diver_FP + pred_confmat_diver{ii}(1,2);
    total_diver_TP = total_diver_TP + pred_confmat_diver{ii}(2,2);
end

% Resultado sumado de todas las matrices
total_TN = total_TN + pred_confmat{ii}(1,1);
total_FN = total_FN + pred_confmat{ii}(2,1);
total_FP = total_FP + pred_confmat{ii}(1,2);
total_TP = total_TP + pred_confmat{ii}(2,2);
end

% Medias de la matriz de confusion global
media_acc = (total_TP + total_TN) / (total_TP + total_TN + total_FP + total_FN); % Exactitud
media_sen = total_TP / (total_FN + total_TP); % Sensibilidad
media_spe = total_TN / (total_TN + total_FP); % Especificidad
media_F1 = total_TP/(total_TP + 0.5 * (total_FN + total_FP)); %F-Score
```

Finalmente, el último paso que resta es el de calcular las métricas de calidad y registrarlas en un `xlsx` clasificándolas mediante el nombre del experimento y su mes de ejecución. Por un lado, como se puede ver en la Figura 58, se mantienen los datos del

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

promediado de las métricas de calidad en función todas las matrices de confusión junto a la suma total de los aciertos y fallos para cada una de las clases y el tiempo de ejecución. Por otro lado, la Figura 59 muestra los mismos parámetros, pero haciendo el estudio de calidad usando únicamente aquellas matrices diversas en las que no se ha predicho correctamente 150 casos de no lluvia. Cabe recordar, que este promediado se hace debido a que por cada vez que se iteraba, el modelo hacía predicciones de la siguiente secuencia en la base de datos llegando a crear para un solo modelo más de 2000 matrices de confusión de 150 predicciones.

	A	B	C	D	E	F	G	H	I	J
1	Mes	Sensibilidad	Especificidad	Exactitud	F1-Score	TN	FP	FN	TP	Tiempo
2	2021-02	0,7591041	0,9829272	0,96160037	0,79023597	967970	16813	24985	78732	0,26264488
3	2021-02	0,786991	0,9856077	0,96674231	0,81802599	970932	14178	22023	81367	0,53790334
4	2021-02	0,7100897	0,98513461	0,95628939	0,77311029	959860	14484	33095	81061	0,81389335
5	2021-02	0,00998613	0,2974317	0,04227193	0,01817505	36364	85896	956591	9649	1,08261777
6	2021-02	0,67097302	0,9851496	0,95022967	0,74980257	953148	14368	39807	81177	1,36296776
7	2021-02	0,72542778	0,98364226	0,95763436	0,77524722	962852	16012	30103	79533	1,6426925
8	2021-02	0,5077524	0,98926263	0,91462655	0,64835186	909902	9876	83053	85669	1,92525764
9	2021-02	0,77235397	0,98452942	0,96425999	0,80502779	969283	15231	23672	80314	2,20772544
10	2021-02	0,72204897	0,98513933	0,95801286	0,78003821	961760	14508	31195	81037	2,48661836

Figura 58. Resultados del predictor con la matriz de confusión global.

Fuente: Elaboración propia.

	K	L	M	N	O	P	Q	R
1	Sensibilidad2	Especificidad2	Exactitud2	F1-Score2	TN2	FP2	FN2	TP2
2	0,759104101	0,982927203	0,96160037	0,79023597	967970	16813	24985	78732
3	0,786991005	0,985607699	0,96674231	0,81802599	970932	14178	22023	81367
4	0,710089702	0,985134614	0,95628939	0,77311029	959860	14484	33095	81061
5	0,009986132	0,297431703	0,04227193	0,01817505	36364	85896	956591	9649
6	0,670973021	0,9851496	0,95022967	0,74980257	953148	14368	39807	81177
7	0,725922	0,983639756	0,9577209	0,77546	962702	16012	29994	79442
8	0,510277321	0,989281851	0,91534392	0,65045061	909152	9850	82150	85598
9	0,773302758	0,984527065	0,96438083	0,80549994	969133	15231	23528	80258
10	0,723654046	0,985168818	0,95831441	0,78095569	961310	14472	30859	80809

Figura 59. Resultados del predictor con la matriz de confusión diversa.

Fuente: Elaboración propia.

7.2.3. Resultados

En el caso de la extracción de conclusiones para este conjunto de experimentos, al ser un número bastante reducido respecto al estudio anterior se plantea de una forma distinta dónde directamente se busca el mejor modelo, luego se vuelve a generar y se pasa por el resto de los meses para obtener sus resultados. Cómo se puede ver tanto en la Figura 60

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

como en la Figura 61 se puede comprobar que la idea de que el mejor detector sea necesariamente el mejor predictor, queda descartada. El mejor caso de la mejor red, como se puede ver en ambas imágenes sólo serviría como predictor para el mes de febrero como mucho.

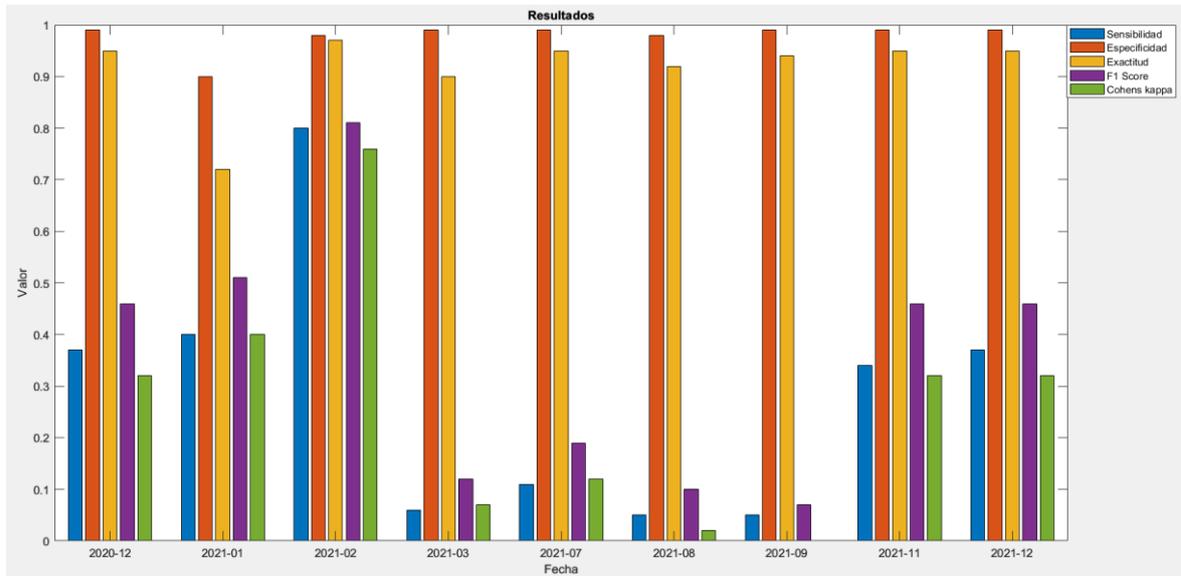


Figura 60. Resultados para el mejor predictor de 150 según el mejor detector.
Fuente: Elaboración propia.

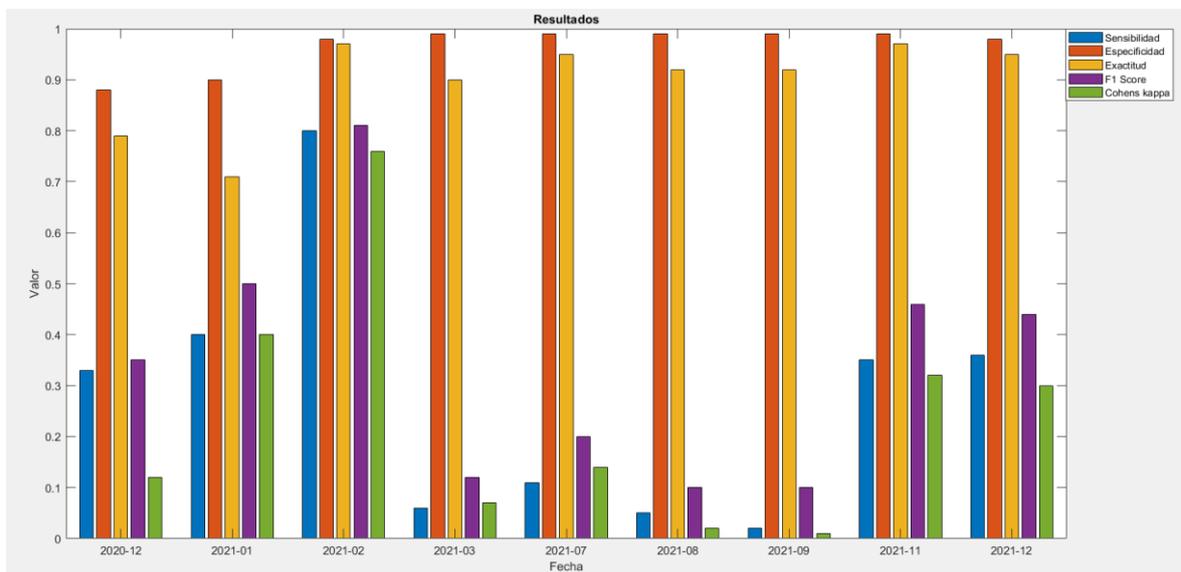


Figura 61. Resultados para el mejor predictor de 350 según el mejor detector.
Fuente: Elaboración propia.

Dado a los resultados que indican que la red creada no es funcional, se decide volver a la metodología del detector y calcular los distintos hiperparámetros que puedan

dar lugar a una red consistente para la tarea de hacer predicciones de precipitaciones para los 20 minutos futuros. En este caso, tras rehacer los experimentos nos encontramos que el mejor caso resulta ser el experimento de nombre Test pctTr80 maxSeqTr700 nIter50 nMesTr3 bajo una secuencia de tamaño 150 y 350.

Como se observa tanto en la Figura 62 como en la Figura 63 los valores de sensibilidad, F1-Score y Cohen's kappa aumentan, siendo el último el que ha determinado los experimentos actuales de la categoría concordancia razonable frente a los experimentos anteriores que en su mayoría eran de concordancia débil como mucho [33]. Del mismo modo, tanto para una secuencia de 150 como para 350 no hay mucha diferencia apreciable por lo que la elección del modelo a implementar en el sistema VIMETRI-MAC dependerá del cómputo que se desee destinar para esta labor, de momento, se elige el de longitud 350 ya que presenta mejores métricas de calidad necesarias para una alarma.

Finalmente, los resultados de los experimentos han dado lugar a un modelo que resulta bastante adecuado para cada uno de los meses [7]. Siendo la sensibilidad, del mismo modo que en el detector, un parámetro importante para alarmar un fenómeno meteorológico que necesite ser alertado. Por otro lado, se encuentran la especificidad y la exactitud de los modelos que, pese a tener un valor alto, son algo inferiores respecto al detector. Esto es esperable debido a que la condición de predictor lleva consigo un deterioro general de los parámetros en función se amplíe el tiempo de predicción [8]. De esta forma, se vuelve a demostrar que el modelo finalmente expuesto resulta viable para su uso como alarma de alerta por precipitaciones basándose en la atenuación del nivel de potencia de los radioenlaces.

CAPÍTULO 7. EXPERIMENTOS Y RESULTADOS

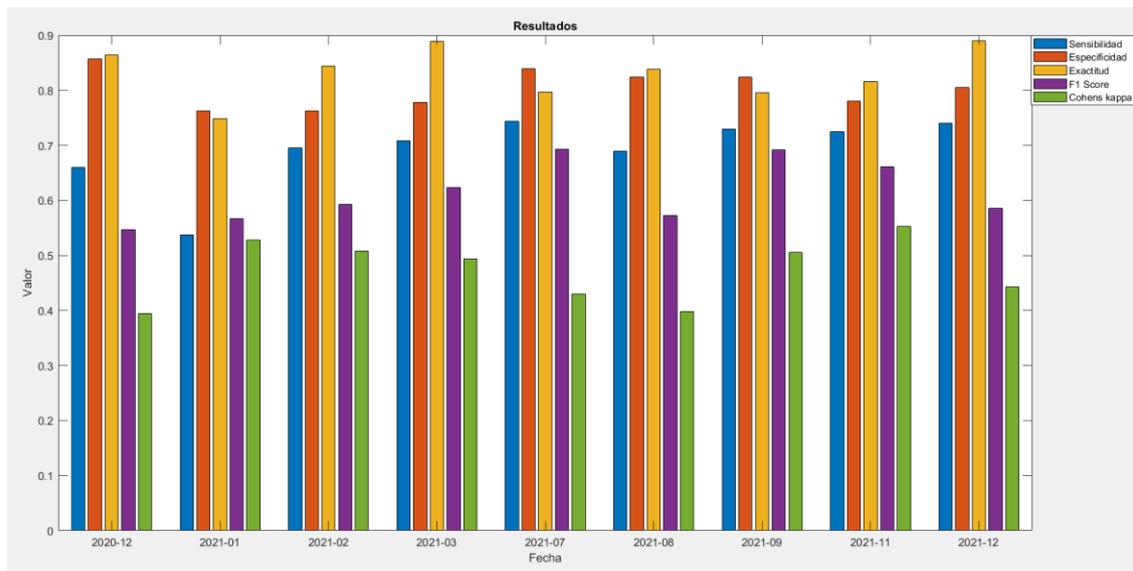


Figura 62. Resultados para el mejor predictor de 150.
Fuente: Elaboración propia.

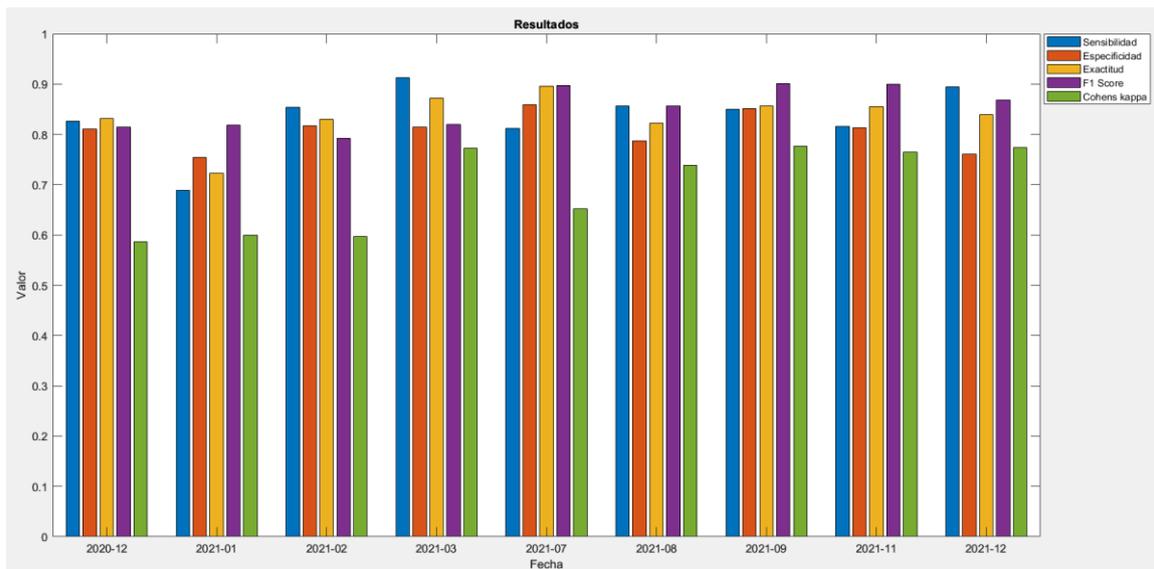


Figura 63. Resultados para mejor predictor de 350.
Fuente: Elaboración propia.

Una vez terminada esta sección se ha logrado completar el quinto objetivo O5. Diseñar del esquema general del sistema global. Ya que actualmente, estas redes de detección y de predicción forman parte del proyecto VIMETRI-MAC una vez ya han sido validadas en su fase de experimentación.

Capítulo 8. Conclusiones

En este capítulo se realiza un resumen con la interpretación de los resultados obtenidos en la fase de experimentación y se exponen las cuestiones que han quedado abiertas como líneas futuras.

8.1. Algoritmo de detección

Por una parte, como se ha podido comprobar en el capítulo anterior, es posible la creación de una red neuronal que cumpla con la tarea de detectar fenómenos de precipitaciones basándose únicamente en la atenuación de la señal de un radioenlace. De esta forma, se ha podido crear un sistema capaz de reconocer lluvia teniendo de entrada los niveles de potencia.

Por otra parte, se ha podido implementar una metodología capaz de crear suficientes números de experimentos como para que la muestra sea lo suficientemente grande como para poder llegar a una solución funcional.

Finalmente, dado al elevado número de experimentos se consigue concluir que uno de los parámetros clásicos en redes neuronales, como es marcar un entrenamiento del 80% de la base de datos, resulta suficiente como para crear modelos de manera efectiva. Del mismo modo, se prueba que la adaptación del parámetro Cohen's kappa del ámbito de la ciencia de la salud a una ciencia técnica como las telecomunicaciones resulta viable tanto o más que parámetros clásicos como el F1-Score.

8.2. Algoritmo de predicción

En primer lugar, al igual que se ha podido comprobar en el capítulo anterior, es posible la creación de una red neuronal que cumpla con la tarea de predecir fenómenos de precipitaciones basándose únicamente en la atenuación de la señal de un radioenlace. De esta forma, se ha podido crear un sistema capaz de predecir lluvia en ventanas de 20 minutos teniendo de entrada los niveles de potencia.

En segundo lugar, se ha podido comprobar que los mejores hiperparámetros para un detector no son necesariamente los mejores para implementar un predictor. De esta manera, se amplía la dificultad del área de la AI dado a que una solución para un problema no necesariamente resuelve otro.

8.3. Líneas futuras

A lo largo de toda la experimentación han ido surgiendo dudas y curiosidades sin resolver que se exponen como líneas futuras para que se puedan resolver en una futura investigación. Estas líneas futuras se resumen en:

- Mejorar la metodología. Dado el actual auge del interés en la AI es lógico pensar que en un futuro cercano se creen nuevas redes o nuevos protocolos de experimentación que superen al actual prueba y error que se ha utilizado a lo largo de este proyecto. Es por ello, que se plantea como continuación del trabajo, la creación de un nuevo sistema que llegue a los mismos resultados o mejores que los aquí expuestos de una forma más rápida y que requiera menos potencia de cómputo.
- Mejorar los resultados. Es posible implementar una nueva arquitectura que permita mejorar los resultados expuestos. Siendo posible que se detecte de una manera más eficaz usando otros parámetros o incluso añadiendo otros tipos de redes que permitan ampliar la ventana de previsión de condiciones meteorológicas a más de 20 minutos sin tener la contraparte de que empeoren las métricas de calidad.
- Implementación global. Pese a que este trabajo se incluya dentro del proyecto VIMETRI-MAC, puede llegar a surgir la posibilidad de combinar distintos tipos de redes de sensores que aporten más datos al sistema y con ellos crear nuevas estructuras de redes que combinen distintos datos para mejorar la detección y predicción.
- Estudios más finos. Este punto va desde crear nuevos bucles con saltos menores hasta llegar a contestar preguntas como cuánto degradaría el predictor si se aumenta la ventana a dos muestras o más, qué resultados se obtendrían si se crea un predictor que ha sido entrenado para predecir o cuánto se puede llegar a ver en el futuro si se implementan más entradas de datos.
- Cuantificación. Este trabajo forma parte de un sistema de alertas para emergencias dónde se considera que la predicción es prioritaria frente a la cuantificación. Es por ello por lo que, dada la necesidad de rehacer los experimentos del modelo de predicción, la cuantificación queda pendiente para un futuro proyecto.

Bibliografía

- [1] J. M. Wallace and P. V. HOBBS, Atmospheric science: an introduction survey, Academic Press, 1977.
- [2] R. G. Hernández and J. Parra López, “Plan especial de protección civil y atención de emergencias por inundaciones de la comunidad autónoma de Canarias,” 2017, p. 103.
- [3] I. D. G. Moreno, Algoritmos de aprendizaje aplicados a la monitorización meteorológica mediante redes inalámbricas de comunicaciones distribuidas, Universidad de las Palmas de Gran Canaria, 2017.
- [4] F. G. G. y R. D. Montoya, “Efecto de la Atenuación por Precipitación y Niebla para Sistemas Wi-Max en el rango de 10 a 66 GHz en entornos aplicables para Colombia,” no. p.9, 2008.
- [5] “AEMET. Clasificación de los Meteoros,” AEMET, [Online]. Available: http://www.aemet.es/es/noticias/2002/03/clasificacion_meteoros. [Accessed Febrero 2023].
- [6] M. G. Monagas, M. Gómez Crespo and A. León Méndez, “Sistema automatizado de alerta temprana ante el peligro de inundaciones,” *RIHA*, vol. 3, no. 3, p. 8, 2013.
- [7] S. F. González, Técnicas de aprendizaje máquina para detección y cuantificación de lluvia mediante radioenlaces terrenos a frecuencia de gigahercios, Grado en Ingeniería en Tecnologías de la Telecomunicación. Escuela de Ingeniería de Telecomunicación y Electrónica: ULPGC, 2020.
- [8] C. TACORONTE RIVERO, Diseño E Integración De Un Sistema Para La Detección Y Cuantificación De Lluvia Mediante Radioenlaces Terrenos Y Satelitales Aplicando Técnicas De Machine Learning, Universidad de Las Palmas de Gran Canaria: Tesis de Maestría, 2022.

BIBLIOGRAFÍA

- [9] E. E. Pérez, Diseño de un sistema de observación meteorológica combinando estaciones meteorológicas y radioenlaces satelitales, Grado en Ingeniería en Tecnologías de la Telecomunicación. Escuela de Ingeniería de Telecomunicación y Electrónica: ULPGC, 2021.
- [10] T. Gubern González, Análisis de Mapas de Inundabilidad de Áreas de Riesgo Potencial Significativo de Inundación (ARPSIs), Universidad de Las Palmas de Gran Canaria: Tesis de Maestría, 2021.
- [11] N. D. R. Hernández, “Técnicas de aprendizaje máquina para detección y cuantificación de lluvia sobre imágenes de videovigilancia,” 2021.
- [12] G. Kaur, “Meteorological data mining techniques: A survey.,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, pp. 325-327, 2012.
- [13] M. G. Cortina Januchs, Aplicación de técnicas de inteligencia artificial a la predicción de contaminantes atmosféricos, 2012.
- [14] M. Acosta-Coll, F. Ballester-Merelo and E. De la Hoz-Franco, “Real-Time Early Warning System Design for Pluvial Flash Floods—A Review,” *Sensors*, vol. 18, no. 7, p. 2255, 2018.
- [15] R. B. Alley, E. K. A and . F. Zhang, “Advances in weather prediction.,” *Science*, pp. 342-344, 2019.
- [16] Y. Liberman, R. Samuels, P. Alper and H. Messer, “New algorithm for integration between wireless microwave sensor network and radar for improved rainfall measurement and mapping,” *Atmospheric Measurement Techniques*, vol. 7, no. 10, pp. 3549-3563, 2014.
- [17] M. F. J. C. R. W. Reitz J. R, Foundations of electromagnetic theory, 4th. ed., AddisonWesley Pub. Co., 1913.
- [18] R. U.-R. P.525, Cálculo de la atenuación en el espacio libre, 2016.

BIBLIOGRAFÍA

- [19] R. U.-R. P.676-12, *Attenuation by atmospheric gases and related effects*, 2019.
- [20] E. S. Gómez, *Atenuación por lluvia*, Bogotá: Universidad El Bosque, 2013.
- [21] P. T. JORGE, *NUBES, FORMACIÓN Y CLASIFICACIÓN.*, Facultad de Planeación Urbana y Regional: Universidad Autónoma del Estado de México, 2019.
- [22] OMM, “ATLAS INTERNACIONAL DE NUBES,” in *MANUAL DE OBSERVACION DE NUBES Y OTROS METEOROS*, vol. I, Ginebra, ISBN 92-63-30407-6, 1993, pp. 3-5.
- [23] R. U.-R. P.838-3, *Modelo de la atenuación específica debida a la lluvia para los métodos de predicción*, 2005.
- [24] C. I. d. A. d. G. Canaria, "Mapas de peligrosidad y riesgo de inundación de las ARPSIS fluviales (Demarcación hidrográfica de Gran Canaria)".
- [25] INCIBE, *Estudio de ciberseguridad en redes TETRA*, <https://www.incibe.es/>, 2023.
- [26] HISPASAT, “HISPASAT,” HISPASAT, 2021. [Online]. Available: <https://www.hispasat.com/es>. [Accessed Julio 2023].
- [27] “VAISALA,” 2023. [Online]. Available: <https://www.vaisala.com/es>.
- [28] H. Sowa, *THE STUDY ON THE SOUNDSCAPE OF THREE JAPANESE GARDENS*, Kyoto University of Art & Design, Japan, 2012.
- [29] S. C. Society, “SIMPLIFYING THE DIFFERENCE: MACHINE LEARNING VS DEEP LEARNING,” 2020. [Online]. Available: <https://www.scs.org.sg/articles/machine-learning-vs-deep-learning>. [Accessed Julio 2023].
- [30] H. Demuth and M. Beale, *Neural Network Toolbox For Use with MATLAB*, The MathWorks, Inc, 2004.

BIBLIOGRAFÍA

- [31] “Activation function,” Wikipedia, 26 octubre 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Activation_function&oldid=1051952747. [Accessed julio 2023].
- [32] A. M. Mañas, “bookdown,” Notas sobre pronóstico del flujo de tráfico en la ciudad de Madrid, 19 junio 2019. [Online]. Available: <https://bookdown.org/amanas/traficomadrid/m%C3%A9todos-basados-en-deep-learning.html#lstm-univariado>. [Accessed julio 2023].
- [33] J. Carletta, “Assessing Agreement on Classification Tasks: The Kappa Statistic,” *Computational Linguistics*, vol. 22, no. 2, p. 249–254, 1996.
- [34] “MathWorks Help Center,” The MathWorks, Inc., 1994. [Online]. Available: <https://es.mathworks.com/help/>. [Accessed julio 2023].
- [35] ULPGC, “2023_retribuciones_pi_contratado_ulpgc.pdf,” ULPGC, 3 febrero 2023. [Online]. Available: <https://www.ulpgc.es>. [Accessed julio 2023].

PARTE II:
PRESUPUESTO

P.1.Introducción

En esta parte del documento se desglosa el coste económico asociado al desarrollo de este TFM. Los aspectos del trabajo que han contribuido a dicho coste son:

- Recursos hardware.
- Recursos software.
- Recursos humanos.
- Redacción del documento.

P.2. Recursos hardware

La Tabla 8, recoge los recursos hardware al igual que su vida útil. Seguidamente se encuentra Tabla 9 que muestra el coste de los equipos hardware en el momento de la adquisición, su amortización anual y el coste final de cada dispositivo.

*Tabla 8. Recursos hardware y su vida útil
Fuente: Elaboración propia.*

Recurso hardware	Vida útil (años)
Asus F541U	5

*Tabla 9. Coste de equipos hardware.
Fuente: Elaboración propia.*

Recursos hardware				
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final
Asus F541U	4 meses	749,00€	149,80€	37,45€
Total:		749,00€	149,80€	37,45€

El coste final de los recursos hardware asciende a TREINTA Y SIETE EUROS CON CUARENTA Y CINCO CÉNTIMOS.

P.3. Recursos software

La Tabla 10, recoge los recursos software al igual que su vida útil. Seguidamente se encuentra la Tabla 11 que muestra el coste de los equipos software en el momento de la adquisición, su amortización anual y el coste final de cada dispositivo.

*Tabla 10. Recursos software y su vida útil.
Fuente: Elaboración propia.*

Recurso software	Vida útil (años)
SO Windows 10	5
Office 365	1
MATLAB R2022b	1

*Tabla 11. Coste de equipos hardware.
Fuente: Elaboración propia.*

Recursos software				
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final
SO Windows 10	4 meses	126,37€	25,24€	6,31€
Office 365	4 meses	69,00€	69,00€	17,25€
MATLAB R2022b	4 meses	00,00€	00,00€	00,00€
Total:		195,37€	94,24€	23,56€

El coste final de los recursos software asciende a VEINTITRÉS EUROS CON CINCUENTA Y SEIS CÉNTIMOS.

P.4. Recursos humanos

Según la tabla “RETRIBUCIONES DEL PERSONAL INVESTIGADOR, DEL PERSONAL TÉCNICO Y DEL PERSONAL TÉCNICO DE APOYO CONTRATADO DE LA ULPGC CON CARGO A PROYECTOS, CONVENIOS Y CONTRATOS PARA EL AÑO 2023” del BOULPGC del 3 de febrero de 2023 [35]. Para un investigador de titulación de máster (MECES 3) bajo una jornada de 20 horas semanales el sueldo base queda reflejado en la Tabla 12.

*Tabla 12. Coste de recursos humanos.
Fuente: Elaboración propia.*

Recursos humanos			
Personal	Coste total mensual	Tiempo	Total
Investigador	1389,63€	4 meses	5558,52€

El coste final de los recursos humanos es de CINCO MIL QUINIENTOS CINCUENTA Y OCHO EUROS CON CINCUENTA Y DOS CÉNTIMOS.

P.5. Material fungible

Se entiende como material fungible los recursos asociados como miscelánea como son los folios, utensilios de escritura, tinta, entre otros. Su valor se ve en la Tabla 13.

*Tabla 13. Coste del material fungible.
Fuente: Elaboración propia.*

Material fungible	
Descripción	Total
Miscelánea	100€

El coste total del material fungible es de CIEN EUROS.

P.6. Redacción del documento

Dadas las horas asignadas a la redacción del documento en el anteproyecto y teniendo en cuenta el coste por hora según el BOULPGC del 3 de febrero de 2023 [35], los costes de redacción se recogen en la Tabla 14.

*Tabla 14 Coste de la redacción del documento.
Fuente: Elaboración propia.*

Redacción			
Personal	Coste total por hora	Tiempo	Total
Investigador	17,37€	90 horas	1563,33€

El coste de redacción asciende a un total de MIL QUINIENTOS SESENTA Y TRES EUROS CON TREINTA Y TRES CÉNTIMOS.

P.7. Aplicación de impuestos y coste final

Se recoge el coste final de este proyecto en la Tabla 15.

*Tabla 15. Presupuesto total.
Fuente: Elaboración propia.*

Presupuesto final	
Partidas	Totales (€)
Recursos hardware	37,45
Recursos software	23,56
Recursos humanos	5558,52
Material fungible	100
Redacción del documento	1563,33
Total	7282,86
IGIC (7%)*	509,80
Total, aplicando IGIC	7792,66€

*Todo el material anteriormente expuesto ha sido comprado previamente sin IGIC.

El coste total resulta SIETE MIL SETESIENTOS NOVENTA Y DOS EUROS CON SESENTA Y SEIS CÉNTIMOS.

PRESUPUESTO

Así lo declaro yo, Gabriel Ojeda Suárez, a día 20 de julio de 2023.

Las Palmas de Gran Canaria a 20 de julio de 2023

Firmado:

PARTE III:
PLIEGO DE CONDICIONES

C.1. Introducción y objetivos

En este apartado se reflejan los matices, normativas y procedimientos que se consideran necesarios para llevar a cabo la aportación que se ha hecho al proyecto VIMETRI-MAC. Por tanto, se aclararán todos los detalles que se estimen como indispensables para que se pueda replicar la experimentación con éxito.

El objetivo principal del proyecto se basa en la implementación de algoritmos de ML para la cuantificación y predicción de precipitaciones de lluvia basado en los niveles de potencia involucrados en comunicaciones terrenas y satelitales a frecuencias de microondas distribuidas por diversas zonas de observación en las Islas Canarias. Estos objetivos se desglosan en los siguientes objetivos específicos:

- O1. Estudiar la influencia de las precipitaciones en los radioenlaces.
- O2. Estudiar los algoritmos para la cuantificación y predicción de lluvia.
- O3. Gestionar el sistema VIMETRI respecto a datos de hidrometeoros.
- O4. Gestionar conjuntos de datos de radioenlaces y estaciones meteorológicas.
- O5. Diseñar del esquema general del sistema global.

C.2. Especificaciones de los materiales

El conjunto de los materiales que se han visto necesarios para la realización de este proyecto se dividen en:

- Base de datos. Se debe contar con una base de datos que contenga, como mínimo mediciones o partes meteorológicos que puedan dar la información necesaria como para que se pueda detectar con claridad que es un caso de precipitación. Del mismo modo, se debe contar con una base de datos que albergue los niveles de señal de, por lo menos, un radioenlace satelital para poder determinar si la atenuación producida entre la señal emitida y la recibida ha sido provocada por el paso de la señal a través de una tormenta.
- Ordenador. Se debe estar en posesión de un ordenador que permita la ejecución del programa MATLAB R2022b.

C.3. Reglamentación y normativa aplicables

El conjunto de normas que rigen este proyecto se basan en las Recomendaciones UIT-R. Dichos documentos aportan la base teórica necesaria como para poder estimar las pérdidas de la señal durante su trayectoria facilitando de sobremanera el procesado de los datos posteriores. Las Recomendaciones UIT-R empleadas en este proyecto son:

- Recomendación UIT-R P.525 [18]. Recomendación que detalla el proceso de estimación de cálculo de la atenuación de una señal en el espacio libre.
- Recomendación UIT-R P.676-12 [19]. Recomendación que detalla el proceso de cálculo de la atenuación de una señal debida a los gases atmosféricos y otros gases.
- Recomendación UIT-R P.838-3 [23]. Recomendación que detalla el proceso de cálculo de la atenuación de una señal provocada por fenómenos meteorológicos de precipitaciones del tipo lluvia.

PARTE IV:
ANEXOS

ANEXO

Anexo 1. Función *GeneradorMesesf*.

Fuente: Propia

```
function meses = GeneradorMesesf(S_0,Vano)
% Genera el vector meses en funcion de los meses comunes.
% Se indica primero el S0 y el vano.
% S_0 = 1, Linux; =2, Windows(Gabriel), =3 Windows(Juan Luis).
% Vano = 1; Vano Pico I - Hospital militar, = 2 Vano Pico I - Pico II.
% Dependiendo de lo escogido, se dirige a un directorio u otro.
if(S_0==1) % Linux

    % Directorio estaciones meteorológicas
    Dir_Estacion = 'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\';

    % Directorio niveles RSL satelital
    Dir_RSL_Sat = 'Base_Datos/Datos_RSL_Estaciones/RSL_Satelite/';

elseif(S_0==2) %Gabriel - Windows
    if Vano == 2
        Dir_Estacion =
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\';
        Dir_RSL_Sat =
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-    PicoII\';

    else
        Dir_Estacion =
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I\';
        Dir_RSL_Sat = 'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-
HospitalMilitar\'; % Directorio niveles RSL satelital
    end

elseif(S_0==3)% Juan Luis - Windows
    if Vano == 2
        Dir_Estacion=
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\';
        Dir_RSL_Sat =
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-PicoII\';

    else

Dir_Estacion='Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I\';
        Dir_RSL_Sat =
'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-
HospitalMilitar\';
    end
end

% Rutas de todas las carpetas.
if length(dir(Dir_Estacion))>=3
```

```
arch_est = dir(Dir_Estacion);
rutas_AEMET = [];

for i=3:length(dir(Dir_Estacion))

    if sum(strfind(string(strcat(Dir_Estacion,arch_est(i).name)), '.')) == 0

        rutas_AEMET = [rutas_AEMET;
string(strcat(Dir_Estacion,arch_est(i).name))];
        end
    end
end

meses = string(['IDeTIC_PAB_B2019-01.xlsx';'IDeTIC_PAB_B2019-
02.xlsx';'IDeTIC_PAB_B2019-03.xlsx';
'IDeTIC_PAB_B2019-04.xlsx';'IDeTIC_PAB_B2019-05.xlsx';'IDeTIC_PAB_B2019-
06.xlsx';
'IDeTIC_PAB_B2019-07.xlsx';'IDeTIC_PAB_B2019-08.xlsx';'IDeTIC_PAB_B2019-
09.xlsx';
'IDeTIC_PAB_B2019-10.xlsx';'IDeTIC_PAB_B2019-11.xlsx';'IDeTIC_PAB_B2019-
12.xlsx';
'IDeTIC_PAB_B2020-01.xlsx';'IDeTIC_PAB_B2020-02.xlsx';'IDeTIC_PAB_B2020-
03.xlsx';
'IDeTIC_PAB_B2020-04.xlsx';'IDeTIC_PAB_B2020-05.xlsx';'IDeTIC_PAB_B2020-
06.xlsx';
'IDeTIC_PAB_B2020-07.xlsx';'IDeTIC_PAB_B2020-08.xlsx';'IDeTIC_PAB_B2020-
09.xlsx';
'IDeTIC_PAB_B2020-10.xlsx';'IDeTIC_PAB_B2020-11.xlsx';'IDeTIC_PAB_B2020-
12.xlsx';
'IDeTIC_PAB_B2021-01.xlsx';'IDeTIC_PAB_B2021-02.xlsx';'IDeTIC_PAB_B2021-
03.xlsx';
'IDeTIC_PAB_B2021-04.xlsx';'IDeTIC_PAB_B2021-05.xlsx';'IDeTIC_PAB_B2021-
06.xlsx';
'IDeTIC_PAB_B2021-07.xlsx';'IDeTIC_PAB_B2021-08.xlsx';'IDeTIC_PAB_B2021-
09.xlsx';
'IDeTIC_PAB_B2021-10.xlsx';'IDeTIC_PAB_B2021-11.xlsx';'IDeTIC_PAB_B2021-
12.xlsx'];]);

% Meses con señal RSL
meses_rsl = [];
arch_rsl = dir(Dir_RSL_Sat);
for irsl=3:length(arch_rsl)
    meses_rsl = [meses_rsl string(arch_rsl(irsl).name)];
end

% Ficheros .xlsx
meses_rsl = meses_rsl(contains(meses_rsl, ".xlsx"));

% Recorre todas las carpetas y compara meses_pivot con el vector meses
for n=2:length(rutas_AEMET)
    meses_pivot = [];
    arch_estiii = dir(rutas_AEMET(n));
    for iii=3:length(dir(rutas_AEMET(n)))
        meses_pivot = [meses_pivot string(arch_estiii(iii).name)];
    end
end
```

```
end

% Compara y elimina los meses que no tienen en comun
for a=1:length(meses)
    if matches(meses(a), meses_pivot)==0 || matches(meses(a), meses_rsl)==0
        meses(a) = NaN;
    end
end

% Vector que contiene solamente los meses comunes en las distintas carpetas
meses = meses(~ismissing(meses));
end

% Se comprueba en los meses comunes si hay datos o no
% Si no tiene, se elimina del vector meses
for mesii=1:length(meses)
    cont = 0;
    for rutasi=1:length(rutas_AEMET)
        mes_est = strcat(rutas_AEMET(rutasi), '\', meses(mesii));
        table_est = readtable(mes_est);

        if sum(table_est.prec)==0
            cont = cont +1;
        else
            continue
        end
    end

    if cont==length(rutas_AEMET)
        disp(['Mes sin datos: ',meses(mesii)])
        meses(mesii) = NaN;
    end
end

meses = meses(~ismissing(meses));

for z=1:length(meses)
    meses(z)= replace(meses(z), '.xlsx', "");
end
```

ANEXO

Anexo 2. Script Sincronismo_Satelital_Mes_COMBINADO6.m

Fuente: Elaboración propia.

```
Mes_Satelite = strcat(Dir_RSL_Sat,[Meses(nn,:) '.xlsx']);
Data_Sat = readtable(Mes_Satelite);
modifyTimeZoneAndFormat = @(x)(datetime(Data_Sat(:,1),'Format','yyyy-MM-dd
hh:mm:ss'));
Data_Sat = table2timetable(Data_Sat);
RSL_BaseTemp = Data_Sat(:,2);

% GENERADOR DE RUTAS PARA CADA CARPETA.
if length(dir(Dir_Estacion))>=3
    arch = dir(Dir_Estacion);
    rutas_AEMET = [];

    for i=3:length(dir(Dir_Estacion))

        % Se busca solo la ruta de las carpetas
        if sum(strfind(string(strcat(Dir_Estacion,arch(i).name)), '.')) == 0
            rutas_AEMET = [rutas_AEMET;
string(strcat(Dir_Estacion,arch(i).name))];
        end
    end
end

% INICIO: SINCRONIZACIÓN DE LOS DATOS PROVENIENTES DE LAS ESTACIONES METEO
% Datos de la primera estación: rutas_AEMET(1)
Mes_Estacion = strcat(rutas_AEMET(1), '\',[Meses(nn,:)],'.xlsx');

Data_Est = readtable(Mes_Estacion);
modifyTimeZoneAndFormat = @(x)(datetime(Data_Est(:,1),'Format','yyyy-MM-dd
hh:mm:ss'));
PREC_BaseTemp = Data_Est(:,[1 14]); % COLUMNA DE LA LLUVIA AEMET 14 VAISALA 7
PREC_BaseTemp = convertvars(PREC_BaseTemp,{'Time'},'datetime');
PREC_BaseTemp = table2timetable(PREC_BaseTemp);

if length(dir(Dir_Estacion))>=3
    for ii=2:length(rutas_AEMET)
        Mes_Estacion = strcat(rutas_AEMET(ii), '\',[Meses(nn,:)],'.xlsx');
        Data_Est = readtable(Mes_Estacion);
        modifyTimeZoneAndFormat = @(x)(datetime(Data_Est(:,1),'Format','yyyy-
MM-dd hh:mm:ss'));
        PREC_BaseTemp2 = Data_Est(:,[1 14]); % COLUMNA:AEMET 14 VAISALA 7
        PREC_BaseTemp2 = convertvars(PREC_BaseTemp2,{'Time'},'datetime');
        PREC_BaseTemp2 = table2timetable(PREC_BaseTemp2);

% Datos cada 5 minutos
        PREC_BaseTemp2 =
synchronize(PREC_BaseTemp,PREC_BaseTemp2,'regular','linear','TimeStep',minutes(
5));
        PREC_BaseTemp = PREC_BaseTemp2;
    end
end
```

ANEXO

```
% FIN: SINCRPONIZACIÓN DE LOS DATOS PROVENIENTES DE LAS ESTACIONES METEO

tti = table2array(PREC_BaseTemp); % timetable a matriz
tti(find(tti==0)) = nan; % 0 a NaN para no contarlos en la media
tti = mean(tti, 2, 'omitnan');
tti(isnan(tti))=0; % NaN a 0
prec = tti/2; % Nivel de lluvia en cada instante de 5'

% NUEVO TIMETABLE
T3 = PREC_BaseTemp;
T3 = timetable2table(T3);
T4 = addvars(T3, prec); % Nueva table añadiendo la columna de la media
T4 = table2timetable(T4); % Paso a timetable para unirlo con PREC_BaseTemp

% ELIMINA TODAS LAS COLUMNAS EXCEPTO LA MEDIA
T4(:,1:end-1)=[];

PREC_BaseTemp = T4;

TT_tmp      = retime(RSL_BaseTemp,'regular','linear','TimeStep',minutes(5));
PREC_tmp    = retime(PREC_BaseTemp,'regular','linear','TimeStep',minutes(5));
TT          = synchronize(TT_tmp,PREC_tmp); % Esto permite dar datos cada 5 minutos

Lluvia_Temp = table2array(TT(:,2));
if ( isnan(Lluvia_Temp(1,1)) ) % Solo si el primer valor es NaN
    [nanx,nany] = find(~isnan(Lluvia_Temp(1:20)));
    lluvia_Temp(1,1) = mean(Lluvia_Temp(nanx));
end

for isn=0:10
    if ( isnan(Lluvia_Temp(end-isn)) ) % Para los ultimos valores NaN
        Lluvia_Temp(length(Lluvia_Temp)-isn) = 0;
    end
end

t          = 1:length(Lluvia_Temp);
Lluvia_Temp =
interp1(t(~isnan(Lluvia_Temp)),Lluvia_Temp(~isnan(Lluvia_Temp)),t);
Fichero_Est = strcat(Dir_Estacion,[Meses(nn,:) '.mat']);
save(Fichero_Est,'Lluvia_Temp');

RSL_Temp   = table2array(TT(:,1));
if ( isnan(RSL_Temp(1,1)) ) % Solo si el primer valor es NaN
    [nanx,nany] = find(~isnan(RSL_Temp(1:20)));
    lluvia_Temp(1,1) = mean(RSL_Temp(nanx));
end

t          = 1:length(RSL_Temp);
RSL_Temp   = interp1(t(~isnan(RSL_Temp)),RSL_Temp(~isnan(RSL_Temp)),t);
Fichero_Sat = strcat(Dir_RSL_Sat,[Meses(nn,:) '.mat']);
save(Fichero_Sat,'RSL_Temp');
```

ANEXO

Anexo 3. Script Obtención_koptValue_Mes_RSLd.m

Fuente: Elaboración propia.

```
k0_ini      = 0.95;
k0_fin      = 3;
k0_delta    = 0.05;
K_k0        = [k0_ini:k0_delta:k0_fin];
L_k0        = length(K_k0);
ROC_acum    = zeros(L_k0,8);
k0_ind      = 1;

for k0 = K_k0 % Bucle de estudio de la ROC en funcion del umbral
    LL_real = [];
    LL_dete = [];
    for mm=Mes_1:Mes_2
        fid0 = fopen(strcat(Dir_RSL_Sat,[Meses(mm,:) '.mat']));
        fid1 = fopen(strcat(Dir_Estacion,[Meses(mm,:) '.mat']));
        if ( (fid0>0) & (fid1>0) )
            load(strcat(Dir_RSL_Sat,[Meses(mm,:) '.mat']));
            fclose(fid0);
            load(strcat(Dir_Estacion,[Meses(mm,:) '.mat']));
            fclose(fid1);
        else % No se generan antes de la deteccion
            Sincronismo_Satelital_Mes;
        end

        % DETECCION DE GAPS
        [Irs1,Jrs1] = find(RSL_Temp == 0);
        RSL_Temp(Irs1) = NaN*ones(length(Irs1),1);
        [Illu,Jllu] = find(Lluvia_Temp == -1);
        Temp_Util = ones(size(RSL_Temp));
        gaps = unique([Irs1;Illu]);
        Temp_Util(gaps) = zeros(size(gaps));

        % FIN DE LA DETECCION DE GAPS

        RSL4 = RSL_Temp;
        % COMIENZO: ZONA DE PREPROCESADO, ANTES DE LA DETECCIÓN DE UMBRAL
        RSLp = -RSL4 + 2*mean(RSL4(1:50))*ones(size(RSL4));
        RSL_a = apertura(RSLp,e_est1);
        RSL_c = cierre(RSLp,e_est1);
        RSL_AC = mean([RSL_a RSL_c],2);
        RSLd = -(RSLp-RSL_c)-q_Error*ones(size(RSLp));
        Ly = length(RSL4);

        % FIN: ZONA DE PREPROCESADO, ANTES DE LA DETECCIÓN DE UMBRAL

        R = zeros(Ly,1);
        Marc = [];
        i = e_est2+1;
        ind = [1:e_est2];
        Inic = 1;
        Finc = LL;
        iindnd = [Inic:Finc];

        RSL_Umb = RSLd;
```

```

umbral = ( mean(RSL_Umb(ind)) + k0*q_Error );

umb_Pot = mean(RSL_AC(ind));
i       = LL+1;
flag    = 1;
gap_i   = 2;
KK      = 1;
Rsum    = 0;
Racum   = [];
Lluviasum = 0;
Lluviacum = [];
while (i <= Ly-1)
    if (Temp_Util(i) == 1) % Bucle de deteccion
        if ( RSL_Umb(i)>=umbral ) %Detectamos pendiente positiva
            ini     = i;
            Marc    = [Marc ini-1];
            UU      = 0;
            umbral2 = umbral;
            while ( (RSL_Umb(i)>umbral2) & (i<Ly) )
                i     = i+1;
                UU    = UU+1;
                if (UU==1)
                    umbral2 = umbral * 0.9;
                end
            end
            fin     = i-1;
            Marc    = [Marc fin];
            for jj=[ini+1:fin]
                difRSL = abs(umb_Pot-RSLp(jj));
                R(jj)  = ( (difRSL/aLs).^(1/beta) ) * Ct;
            end
        else
            if (flag==1) % El dato anterior existe, no era gap
                Inic   = i-LL+1;
                Finc   = i;
                ind     = [Inic:Finc];
            elseif (flag==0) % Uno de los 20 anteriores era gap
                Inic   = gaps(gap_i)+1;
                Finc   = i;
                ind     = [Inic:Finc];
                KK     = KK+1;
                if (KK == LL) % Amplia ventana hasta forma original
                    flag = 1;
                end
            end
            umbral = ( mean(RSL_Umb(ind)) + k0*q_Error );
            umb_Pot = mean(RSL_AC(ind));
            i       = i+1;
        end
        flag = 1;
    elseif (Temp_Util(i) == 0)
        i       = gaps(gap_i)+2; % Avanzamos dos muestras
        gap_i   = gap_i +2;
        flag    = 0;
    end
end

```

```

% Acumulacion de 30 muestras atras
if (i<=Ly-1)
    if (R(i)>0)
        for z = i-M_atras:i
            Rsum = Rsum + R(z);
            Racum = [Racum Rsum];
        end
    end
    if (Lluvia_Temp == 0)
        for z = i-M_atras:i
            Lluviasum = Lluviasum + R(z);
            Lluviacum = [Lluviacum Rsum];
        end
    end
end
end

% Lluvia_Temp; Registro de lluvia de la estacion meteorologica
% Marc: Marcas de inicio y fin de lluvia del registro RSL

% Crear ventana de confianza para el detector de 2 minutos antes y
% despues de cada deteccion

% MARGEN DE CONFIANZA CON LA LLUVIA REAL
Marc_real = [];
[Marca_real,Jm] = find(Lluvia_Temp > 0); % Marca_real
[Iv,Jv] = size(Marca_real');
Adap_Marc = zeros(2,1); % Limites inferior/superior
if (~isempty(Marca_real))
    Adap_Marc(1) = Marca_real(1) - Minu_Conf_R; % Limite inferior
    if ( Adap_Marc(1) < 0 ) Adap_Marc(1) = Marca_real(1);
    end
    Adap_Marc(2) = Marca_real(1) + Minu_Conf_R; % Limite superior
    Marc_real = [Marc_real [Adap_Marc(1) Adap_Marc(2)]];
% Primeras marcas de lluvia real, con su margen de confianza
for hh=2:Jv % Proseguimos desde la primera marca de lluvia real
    Temp_Marc = [Marca_real(hh)-Minu_Conf_R
Marca_real(hh)+Minu_Conf_R];
    if (Temp_Marc(1) > (Adap_Marc(2)+1) ) % Si no hay solape
        Adap_Marc(1) = Temp_Marc(1);% Actualizamos limite inferior
        Adap_Marc(2) = Temp_Marc(2);% Actualizamos limite superior
% Ponemos marcas de lluvia real, con su margen de confianza
        Marc_real = [Marc_real [Adap_Marc(1) Adap_Marc(2)]];
    else
        Adap_Marc(2) = Marca_real(hh)+Minu_Conf_R; % Si hay solape
    end
end
end

% MARGEN DE CONFIANZA CON LA LLUVIA DETECTADA
Marc_dete = [];
Adap_Marc = zeros(2,1);
if (~isempty(Marc))
    Adap_Marc(1) = Marc(1) - Minu_Conf_D; % Limite inferior
    if ( Adap_Marc(1) < 0 ) Adap_Marc(1) = Marc(1);end
    Adap_Marc(2) = Marc(2) + Minu_Conf_D; % Limite superior

```

```

Marc_dete      = [Marc_dete [Adap_Marc(1) Adap_Marc(2)]];
Jv            = length(Marc);
for hh=3:2:Jv
    Temp_Marc  = [Marc(hh)-Minu_Conf_D Marc(hh+1)+Minu_Conf_D];
    if (Temp_Marc(1) > (Adap_Marc(2)+1) ) % Si no hay solape
        Adap_Marc(1) = Temp_Marc(1); % Actualiza limite inferior
        Adap_Marc(2) = Temp_Marc(2); % Actualiza limite superior
        Marc_dete    = [Marc_dete [Adap_Marc(1) Adap_Marc(2)]];
    else
        Adap_Marc(2) = Marc(hh+1)+Minu_Conf_D;
    end
end
end

Ll_real = zeros (Ly-1,1);
Ll_dete = zeros (Ly-1,1);

Marc2by2_real = []; % marcas inferior/superior de lluvia real
Marc2by2_real = reshape (Marc_real,2,[]);
Marc2by2_dete = []; % marcas inferior/superior de lluvia detectada
Marc2by2_dete = reshape (Marc_dete,2,[]);
[Idet,Jdet] = size(Marc2by2_dete);

[Id,Jd] = size(Marc2by2_real);
if ( ~isempty(Marc2by2_real) )
    for ii=1:Jd
        Ll_real([Marc2by2_real(1,ii):Marc2by2_real(2,ii)],1) =
ones(size([Marc2by2_real(1,ii):Marc2by2_real(2,ii)]));
    end
end
[Id,Jd] = size(Marc2by2_dete);
for ii=1:Jd
    Ll_dete([Marc2by2_dete(1,ii):Marc2by2_dete(2,ii)],1) =
ones(size([Marc2by2_dete(1,ii):Marc2by2_dete(2,ii)]));
end

% Para cada umbral acumulamos Ll_real y Ll_dete y sobre estas variables
LL_real = [LL_real;Ll_real];
LL_dete = [LL_dete;Ll_dete];

if (0)
    figure(1)
    clf
    plot(RSL_Umb, 'm')
    grid
    hold on
    plot(RSLd, 'b')
    plot(RSLp-mean(RSLp(1:200)), 'r')
    plot(LL_dete, 'k')
    plot(LL_real, 'y')
    mm;
end

end

if (0)

```

```

figure
clf
subplot(311)
bar(LL_real);
title('Lluvia real');
axis([1 Ly-1 -0.1 1.1]);
subplot(312)
bar(LL_dete);
title('Lluvia detectada');
axis([1 Ly-1 -0.1 1.1]);
subplot(313)
bar(LL_real.*LL_dete);
title('Lluvia coincidente');
axis([1 Ly-1 -0.1 1.1]);
mm;
end

[a,b] = find(LL_real==1); % Si no hay lluvia
if (~isempty(a))

    cm =
confusionchart(num2str(LL_real),num2str(LL_dete),"title","Validation
Accuracy");
    MatrizConfusion = cm.NormalizedValues;
    close; % Cerramos la ventana con la matriz de confusión
    TN = MatrizConfusion(1,1);
    FN = MatrizConfusion(2,1);
    TP = MatrizConfusion(2,2);
    FP = MatrizConfusion(1,2);
    Sen = TP/(TP+FN);
    Spe = TN/(TN+FP);
    ACC = (TP+TN)/(TP+TN+FP+FN);
    F1 = 2*(Sen*Spe)/(Sen+Spe);

else
    TN = find (LL_real == 0 & LL_dete ==0); % Posiciones de los TN
    FP = find (LL_real == 0 & LL_dete ==1); % Posiciones de los FP

    TP_c = length(LL_real);
    TN_c = length(TN);
    FP_c = length(FP);
    FN_c = 0;
end
ROC_acum(k0_ind,1:4) = [TN, FP, FN, TP];
ROC_acum(k0_ind,5:8) = [Sen, Spe, ACC, F1];

if (1)
    disp(['Analizado el coeficiente de umbral nÂ° ' num2str(k0_ind) ' de '
num2str(L_k0) ' _ ' num2str(Jdet)]);
end

k0_ind= k0_ind+1;

end

if (0)

```

```
Sensitivity      = ROC_acum(:,1);
Specificity     = ROC_acum(:,2);

plot(K_k0,Sensitivity,K_k0,Specificity);

kk = 1;
while ( Sensitivity(kk) > Specificity(kk) && kk < length(Sensitivity) )
    kk = kk+1; % Si se cumple la condicion para el bucle
end

if (kk>1)
    k0_1 = kk-1;
    k0_2 = kk;
    k_opt = mean([K_k0(k0_1) K_k0(k0_2)]);
else
    k_opt = 1.3; % Valor promedio
end
end

[I_f1,J_f1] = max(ROC_acum(:,8));
k_opt = K_k0(J_f1);

% A la salida solo se necesita k0_opt
clear k0 k_k0 Marc2by2_real Ll_real Ll_dete LL_real LL_dete
```

ANEXO

Anexo 4. Función cierre.m

Fuente: Elaboración propia.

```
% PROYECTO VIMETRI-MAC.  
% ESTE SOFTWARE DE LOS QUE HACE SUSO SON PARA USO EXCLUSIVO DEL PROYECTO.  
% ULPGC. 2020  
%
```

```
function [cier,dil] = cierre(vec,n_elem)
```

```
long = length(vec);
```

```
vec = [vec;zeros(n_elem-1,1)];  
dila = dilata (vec,n_elem);  
cie = erosiona(dila,n_elem);
```

```
cier = zeros(long,1);  
cier(1:long-1) = cie(2:long);  
cier(long) = cie(long-1);
```

```
dil = zeros(long,1);  
dil(1:long-1) = dila(2:long);  
dil(long) = dila(long-1);
```

ANEXO

Anexo 5. Función dilata.m
Fuente: Elaboración propia.

```
% PROYECTO VIMETRI-MAC.  
% ESTE SOFTWARE DE LOS QUE HACE SUSO SON PARA USO EXCLUSIVO DEL PROYECTO.  
% ULPGC. 2020  
%  
  
function [dil] = dilata(vec,n_elem)  
  
long = length(vec);  
dil = zeros(long,1);  
  
ind = [1:n_elem];  
seg = [vec;vec(long)*ones(n_elem-1,1)];  
for t=1:long  
    dil(t,1) = max(seg(ind));  
    ind      = ind +1;  
end
```

ANEXO

Anexo 6. Función *erosiona.m*

Fuente: *Elaboración propia.*

```
% PROYECTO VIMETRI-MAC.  
% ESTE SOFTWARE DE LOS QUE HACE SUSO SON PARA USO EXCLUSIVO DEL PROYECTO.  
% ULPGC. 2020  
%  
  
function [ero] = erosiona(vec,n_elem)  
  
long = length(vec);  
ero = zeros(long,1);  
  
ind = [1:n_elem];  
seg = [vec(1)*ones(n_elem,1);vec];  
for t=1:long  
    ero(t,1) = min(seg(ind));  
    ind      = ind +1;  
end
```

ANEXO

Anexo 7. Script Lanza_Detector_Lluvia_COMBINADO8_4.m
Fuente: Elaboración propia.

```
% Bucle de lanzamiento para varios casos
% for pctTr=0.7:0.1:0.9
%     for maxSeqTr=500:100:1000
%         for nIter=50:5:100
%             %for nMesTr=1:1:6

                % Lanzamiento de un solo caso
                pctTr      = 0.8;
                maxSeqTr   = 500;
                nIter      = 50;
                nMesTr     = 4;

                % Limpiar todas las variables excepto las cuatro especificadas
                clearvars -except pctTr maxSeqTr nIter nMesTr;
                close all;
                clc
                warning('off','all');

                %% Variables de control del experimento 1
                S_0 = 2;% =1, Linux; =2, Windows(Gabriel), =3 Windows(Juan Luis)
                Vano = 1;% =1 Vano Pico I - Hospital militar, = 2 Vano Pico I - Pico II

                Version = 0; % valor para que, a partir de este, se incremente en 1
                las versiones del if(1)
                pctTr100=pctTr*100;
                Nom_Expe      = string(sprintf('Test pctTr%d maxSeqTr%d
nIter%d nMesTr%d', pctTr100, maxSeqTr, nIter, nMesTr));
                Dir_Resu      = 'Results';    % Carpeta de guardado
                LL             = 25;         % Tamaño de la ventana para
                calcular la media al calcular los umbrales
                M_atras       = 3;          % Número de muestras anteriores
                que tomamos para calcular la lluvia acumulada
                M_delante     = 10;         % Número de muestras
                posteriores que tomamos para calcular la lluvia acumulada
                Minu_Conf_R   = 3;          % Margen de confianza en la
                detección de lluvia real (dada por estaciones meteorologicas)
                Minu_Conf_D   = 3;          % Margen de confianza en la
                detección de lluvia por el detector (dada en el cálculo del umbral óptimo)
                sequenceLength = 350;      % Longitud de las secuencias en
                que se divide la señal
                hopLength     = 4;
                sequenceOverlap = sequenceLength-hopLength;
                RSL_Morf      = 0;

                Meses = GeneradorMesesf(S_0,Vano);
                Meses = char(Meses);

                [Im,Jm]      = size(Meses);
                Mes_Ini      = 1;
                [Mes_Fin,Im] = size(Meses);
                Num_Iter     = nIter;
                N_Mes_Tr     = nMesTr;    % Número de meses anteriores al
                actual usados para entrenar
```

ANEXO

```
Porc_Train          = pctTr; % Porcentaje de secuencias tomadas
para entrenamiento. Para test cogemos todas las secuencias del mes en curso
doTraining          = true;
doTesting           = true;
doValidation_Same_Tr= true; % Hacemos la validación (durante el
entrenamiento) con datos que entrenaron (true) o no (false)
doTesting_Same_Val  = true; % Hacemos el Test con los datos de
validación (true) o del mes actual (false) que no participó en el
entrenamiento.
maxEpochs          = 8;
maxSeqs_Train       = maxSeqTr; % Número máximo de secuencias de
entrenamiento. Con esto prevenimos que haya demasiadas y se relentice demasiado
el entrenamiento
miniBatchSize       = 32; % En muchos trabajos este valor se da
como bueno porque, en general, permite llegar a buenos resultados.

% Creacion de redes para la deteccion
%Detector_Test_CTR_COMBINADO_Gabriel_01;
%PlantillaV0;

% Busqueda de la mejor red y su creacion
%BuscarMejorNN;

% Creacion de la red de prediccion
%predictor_v1;
%PlantillaV0;

%end
%   end
% end
```

ANEXO

Anexo 8. Script Detector_Test_CRT_COMBINADO_Gabriel_01.m

Fuente: Elaboración propia.

```
if (S_0==1)
    Dir_Estacion =
    'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E/'; %
    Directorio estaciones meteorológicas
    Dir_RSL_Sat = 'Base_Datos/Datos_RSL_Estaciones/RSL_Satelite/';
    % Directorio niveles RSL satelital
    % Gabriel - Windows
elseif (S_0==2)
    if Vano == 2
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E\'; %
        Directorio estaciones meteorológicas
        Dir_RSL_Sat =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-PicoII\';
        % Directorio niveles RSL satelital
    else
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I\';
        Dir_RSL_Sat =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-
        HospitalMilitar\'; % Directorio niveles RSL satelital
    end
elseif (S_0==3)
    if Vano == 2
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E\'; %
        Directorio estaciones meteorológicas
    else
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I\';
    end
    Dir_RSL_Sat =
    'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-PicoII\';
    % Directorio niveles RSL satelital
end

%% GESTIÓN DEL PERIODO DE ANÁLISIS

% PARÁMETROS DE LOS EXPERIMENTOS
Nh1 = 1; % Análisis morfológico, número de horas previas
e_est1 = Nh1*60; % Primer análisis morfológico, análisis grueso
e_est2 = 25; % Segundo análisis morfológico, análisis fino
% Modelo de la IUT
Al_N = 1000; % Altura de las nubes
Al_A = 382; % Altura de la antena
alfa = 0.018; % Constante alpha, relación atenuación-lluvia,
Recommendation ITU-R P.838-3
beta = 1.21; % Constante beta, relación atenuación-lluvia,
Recommendation ITU-R P.838-3
Elev = 30*pi/180; % Elevación de la antena
Ls = (Al_N-Al_A)/sin(Elev); % Longitud recorrida por el haz a lo
largo de la columna de lluvia, supuesta en todo el recorrido, lluvia uniforme
aLs = alfa*Ls;
```

ANEXO

```
% T          = 15.5;          % Periodo de muestreo de la RSL (en segundos)
T            = 1;            % Periodo de muestreo de la RSL ( en minutos)
Th           = 60;           % minutos/hora
Ct           = T/Th;         % Ancho temporal de integración para determinar
R(mm/h)
q_Error      = 0.25;         % 2*q_Error = margen dinámico del error de
cuantificación de los RSL
Num_Proc     = 1;           % Número de minutos anteriores que tomamos para
extraer características
WindowLength = 1;
Plot_Fig     = 1;

%% Lluvia_RealVSDete
% Creación de un directorio con el nombre Nom_Expe
s1 = 'ResultsV2\';
s2 = Nom_Expe;
Dir_Resu = strcat(s1,s2); % Dentro de la carpeta Results

DD = dir(Dir_Resu);          % dir mira los archivos que hay dentro
del directorio pasado
if ( isempty(DD) )          % Crea el archivo si no existe ya
    mkdir(Dir_Resu);
end

if (S_0==1)
    Metrics_str = [Dir_Resu '/Quality_Metrics'];
elseif (S_0==2)
    Metrics_str = [Dir_Resu '\Quality_Metrics'];
elseif (S_0==3)
    Metrics_str = [Dir_Resu '\Quality_Metrics'];
end

%fid0 = fopen(Metrics_str);
fid0 = fopen(strcat(Metrics_str(1),Metrics_str(2)));
if ( fid0<0 )
    Metricas_meses = zeros(Num_Iter,Im,8); % Numero de meses excels distintos
else
    load(Metrics_str);
    fclose(fid0);
end

% Se crea un excel en la carpeta creada denominada Tabla_Metricas.xlsx y
% crea las columnas A_Met y A_Con
filename = strcat(Dir_Resu,'\Tabla_Metricas','.xlsx');
A_Met    = {'Mes', 'Sensibilidad', 'Especificidad', 'Exactitud', 'F1-
Score'};
A_Con    = {'TN', 'FP', 'FN', 'TP', 'Tiempo'};

Metricas = [];

for nn=Mes_Ini:Mes_Fin      % Recorreremos los meses de interés

    Mes_Actual = Meses(nn,end-6:end);

    tic % inicio el calculo de tiempo de computo
```

```

% Apertura de la tabla de resultados con las métricas, mes a mes
Rango1 = ['A' num2str(1) ':' 'E' num2str(1)];
Rango2 = ['F' num2str(1) ':' 'J' num2str(1)];
% Escribe las columnas anteriores en el fichero excel
xlswrite(filename,A_Met,Meses(nn,end-6:end),Rango1);
xlswrite(filename,A_Con,Meses(nn,end-6:end),Rango2);

%Aquí se crea el excel y se añaden las columnas
fid0      = fopen(strcat(Dir_RSL_Sat,[Meses(nn,:) '.mat']));
fid1      = fopen(strcat(Dir_Estacion,[Meses(nn,:) '.mat']));
if ( (fid0>0) & (fid1>0) ) % Los datos de interés ya se han generado en una
ejecución previa ...
    load(strcat(Dir_RSL_Sat,[Meses(nn,:) '.mat']));
    fclose(fid0);
    load(strcat(Dir_Estacion,[Meses(nn,:) '.mat']));
    fclose(fid1);
else      % ... Si no se generan antes de la detección
    Sincronismo_Satelital_Mes_COMBINADO6;
end

%      Mes_1 = Mes_Ini; % Coge el mes inicial, a medida que se avanza se
sigue cogiendo TODOS los anteriores
Mes_1 = max(1,nn-N_Mes_Tr); % Primer mes usado para entrenar
Mes_2 = max(1,nn-1);      % Segundo mes usado para entrenar

if      (S_0==1)
    k_str = [Dir_Resu '/k_optimo'];
elseif (S_0==2)
    k_str = [Dir_Resu '\k_optimo'];
elseif (S_0==3)
    k_str = [Dir_Resu '\k_optimo'];
end

% Generan los archivos .mat con sus nombres
if (nn==1) % Si es el primer mes de toda la iteracion
    k_str = strcat(k_str,['_' Meses(nn,end-6:end) '_' Meses(nn,end-
6:end)]);
else
    for hh=Mes_1:Mes_2
        k_str = strcat(k_str,['_' Meses(hh,end-6:end)]);
    end
end

Bucle_Datos_Train_Test_Tris8;
if ( sum(Label_present)>0 )
    trainFeatureCell = {};
    trainLabelCell   = {};
    index            = 1;
    solape_Labels    = zeros(Lg,Lg);
    [Iseq,Jseq]      = size(rain_sequence);
    for ss=1:Jseq
        if ( Label_present(ss)==1 ) % Si hay lluvia se toma la secuencia
            trainFeatureCell{end+1} = rsl_sequence{ss};
            trainLabelCell{end+1}   = categorical(rain_sequence{ss});
            if ss ~= 1 % cuando ss toma el valor de 1, al hacer ss-1 da
error ya que no hay indice 0 en matlab

```

```

        if ( Label_present(ss-1)==0 ) % Si la secuencia
inmediatamente anterior es de no-lluvia se toma
            trainFeatureCell{end+1} = rsl_sequence{ss};
            trainLabelCell{end+1} =
categorical(rain_sequence{ss});
        end
    end
end
detected_Labels = zeros(1,Lg);
detected_Labels = mean(solape_Labels,1) > 0;

[Itr,Num_secs] = size(trainFeatureCell); %Apartado para generar las
muestras de entrenamiento y test de forma aleatoria, no heuristico
Num_secs = min(maxSeqs_Train,Num_secs); % Limitamos el número de
ejemplos que entrenan
Num_Perm = randperm(Num_secs);
Num_Tr = round(Porc_Train*Num_secs);
ind_Tr = Num_Perm(1:Num_Tr);

% Características y etiquetas en los lmeses de test
Mes_1 = nn; % Testeamos sobre el mes en curso
Mes_2 = nn; % Testeamos sobre el mes en curso

Bucle_Datos_Train_Test_Tris8;

if ( sum(Label_present)>0 )
    testFeatureCell = {};
    testLabelCell = {};
    index = 1;
    solape_Labels = zeros(Lg,Lg);
    [Iseq,Jseq] = size(rain_sequence);
    for ss=1:Jseq
        if ( Label_present(ss)==1 ) % Si hay lluvia se toma la
secuencia
            testFeatureCell{end+1} = rsl_sequence{ss};
            testLabelCell{end+1} = categorical(rain_sequence{ss});
            if ss ~= 1 % cuando ss toma el valor de 1, al hacer ss-1 da
error ya que no hay indice 0 en matlab
                if ( Label_present(ss-1)==0 ) % Si la secuencia
inmediatamente anterior es de no-lluvia se toma
                    testFeatureCell{end+1} = rsl_sequence{ss};
                    testLabelCell{end+1} =
categorical(rain_sequence{ss});
                end
            end
        end
        testFeatureCell = testFeatureCell(:);
        testLabelCell = testLabelCell(:);
        [Itr_ts,Num_secs_Ts] = size(testFeatureCell); % Apartado para
generar las muestras de entrenamiento y test de forma aleatoria, no heuristico
        Num_secs_Ts = min(Num_secs_Ts,maxSeqs_Train); % Como
mucho, habrá tntas secuencias de test como de train

        % Train the LSTM Network

```

```

if (S_0==1)
    k_str = [Dir_Resu '/Model'];
elseif (S_0==2)
    k_str = [Dir_Resu '\Model'];
end
if (nn==1)
    k_str = strcat(k_str,['_' Meses(nn,end-6:end) '_' Meses(nn,end-
6:end)]);
else
    for hh=Mes_1:Mes_2
        k_str = strcat(k_str,['_' Meses(hh,end-6:end)]);
    end
end
k_str = strcat(k_str, '.mat');
Metricas = [];
for iter=1:Num_Iter % En cada mes iteramos
    Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(1)];
    Rango2 = ['F' num2str(iter+1) ':' 'I' num2str(1)];
    if doTraining
        layers = [ ...
            sequenceInputLayer(
size(trainFeatureCell,1),"Name","Entrada" )
            bilstmLayer(200,"OutputMode","sequence","Name","Capa
1")
            bilstmLayer(100,"OutputMode","sequence","Name","Capa
2")
            fullyConnectedLayer(2,"Name","fullyConnected")
            softmaxLayer("Name","Softmax")
            classificationLayer("Name","Classification")
        ];

        if doValidation_Same_Tr
            validationFeatureCell = trainFeatureCell(1,ind_Tr).';
            validationLabelCell = trainLabelCell(1,ind_Tr).';
        else
            validationFeatureCell =
trainFeatureCell(1,Num_Perm(Num_Tr+1:end)).';
            validationLabelCell =
trainLabelCell(1,Num_Perm(Num_Tr+1:end)).';
        end
        options = trainingOptions("adam", ...
            "MaxEpochs",maxEpochs, ...
            'ExecutionEnvironment','auto', ...
            "MiniBatchSize",miniBatchSize, ...
            "Shuffle","every-epoch", ...
            "Verbose",0, ...
            "SequenceLength",sequenceLength, ...
            "ValidationFrequency",10, ...

"ValidationData",{validationFeatureCell,validationLabelCell}, ...
            "Plots","training-progress", ... %"none"
            "LearnRateSchedule","piecewise", ...
            "LearnRateDropFactor",0.1, ...
            "LearnRateDropPeriod",5);
    end
end

```

% ENTRENAMIENTO

```

        [rainDetectNet,netInfo] =
trainNetwork(trainFeatureCell(1,ind_Tr),trainLabelCell(1,ind_Tr),layers,options
);
        disp(['Mes ' Meses(nn,end-6:end) ' Iteración: '
num2str(iter) ' Validation accuracy: '
num2str(netInfo.ValidationAccuracy(end))])
        %save(k_str,'rainDetectNet','netInfo');
        save(Dir_Resu + '\\'
+Nom_Expe+'.mat','rainDetectNet','netInfo');
        %delete(findall(0)); % Cerrar la ventana de Training
Progress par evitar que se acumulen muchas
        end

        if doTesting
        % TEST
        if ( ~isempty(testFeatureCell) ) % Podría estar vacía de
eventos de lluvia
            if doTesting_Same_Val
                testFeatureCell = validationFeatureCell;
                testLabelCell = validationLabelCell;
            else
                testFeatureCell =
testFeatureCell([1:min(Num_secs_Ts,Num_secs)],1);
                testLabelCell =
testLabelCell([1:min(Num_secs_Ts,Num_secs)],1);
            end
            DetectedRain =
classify(rainDetectNet,testFeatureCell.);
            [Num_secs_Ts,Itr_ts] = size(testFeatureCell);
%Apartado para generar las muestras de entrenamiento y test de forma aleatoria,
no heurístico
            if (0) % Versión anterior
                solape_Labels_Real = zeros(Lg,Lg);
                solape_Labels_Dete = zeros(Lg,Lg);
                ind_lab = [1:sequenceLength];
                [Id,Jd] = size(DetectedRain);
                index = 1;
                for ii=1:Id
                    LL_DETECTED =
str2num(char(DetectedRain{ii}));
                    solape_Labels_Dete(index,ind_lab) =
LL_DETECTED.';
                    LL_TEST =
str2num(char(testLabelCell{ii}));
                    solape_Labels_Real(index,ind_lab) =
LL_TEST.';
                    ind_lab = ind_lab +
hopLength;
                    index = index +
hopLength;
                end
                detected_Labels_Dete = zeros(1,Lg);
                detected_Labels_Real = zeros(1,Lg);
            end
            %if (1) % Versión actual
            HopLength = hopLength-1;

```

```

solapeLength      = HopLength*(Num_secs_Ts-
1)+sequenceLength;
solape_Labels_Real = zeros(solapeLength,1);
solape_Labels_Dete = zeros(solapeLength,1);
ind_lab           = [1:sequenceLength];
for ii=1:Num_secs_Ts
    LL_DETECTED =
str2num(char(DetectedRain{ii}));
solape_Labels_Dete(ind_lab,1) = LL_DETECTED;
    LL_TEST =
str2num(char(testLabelCell{ii}));
solape_Labels_Real(ind_lab,1) = LL_TEST.';
    ind_lab = ind_lab +
HopLength;
end
%end
detected_Labels_Dete = zeros(1,solapeLength);
detected_Labels_Real = zeros(1,solapeLength);
detected_Labels_Dete = mean(solape_Labels_Dete,2) > 0;
% detected_Labels_Dete vs detected_Labels_Real
detected_Labels_Real = mean(solape_Labels_Real,2) > 0;
% detected_Labels_Dete vs detected_Labels_Real

%% MÉTRICAS DE CALIDAD
cm =
confusionchart(detected_Labels_Dete,detected_Labels_Real,"title","Validation
Accuracy");
%
[c,m] = size(cm);
MatrizConfusion = cm.NormalizedValues;
[c,m] = size(MatrizConfusion);
close; % Cerramos la ventana con la matriz de confusión
if c == 1
    if (
sum(detected_Labels_Real)==length(detected_Labels_Real) ) % Todas las muestras
son de Sí lluvia
        TP = MatrizConfusion(1,1);
        TN = 0;
        FN = 0;
        FP = 0;
    elseif (sum(detected_Labels_Real)==0 ) % Todas las
muestras son de NO lluvia
        TP = 0;
        TN = MatrizConfusion(1,1);
        FN = 0;
        FP = 0;
    end
else
    TN = MatrizConfusion(1,1);
    FN = MatrizConfusion(2,1);
    TP = MatrizConfusion(2,2);
    FP = MatrizConfusion(1,2);
end
Sen = TP/(TP+FN); % True Positive Rate - Sensitivity
Spe = TN/(TN+FP); % True Negative Rate- Specificity
ACC = (TP+TN)/(TP+TN+FP+FN); % Accuracy

```

```

        F1 = TP/(TP+0.5*(FN+FP));      % F1 scores is the
harmonic mean of precision and sensitivity
        Metricas_meses(iter,nn,1:4) = [TN, FP, FN, TP];
        Metricas_meses(iter,nn,5:8) = [Sen, Spe, ACC, F1];
    else
        Metricas_meses(iter,nn,:) = [-100, -100, -100, -100,-
100, -100, -100, -100]; % Con el -100 indicamos que no hay datos válidos por
falta de ejemplos de test
    end
end
%% ESCRIBO EN EL XLSX TRAS CADA ITERACION
% Para cada mes, guardamos las métricas resultantes de cada
% iteración

timerval = toc/60;
Rango0 = ['A' num2str(iter+1)];
Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];
Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
if nn == 1
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
else
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
end

Metricas_i = [Sen, Spe, ACC, F1, TN, FP, FN, TP, timerval];
[Inan]      = isnan(Metricas_i);
[Inan,Jnan] = find(Inan==1);
Metricas_i(Jnan) = -1*ones(size(Jnan));
Metricas = [Metricas; Metricas_i];

xlswrite(filename,{Mes_Actual}, Meses(nn,end-6:end), Rango0);
xlswrite(filename,[Sen, Spe, ACC, F1],Meses(nn,end-
6:end),Rango1);
xlswrite(filename,[TN, FP, FN, TP, timerval],Meses(nn,end-
6:end),Rango2);
%save(Metrics_str,'Metricas_meses');
save('Metricas_meses');
nn;

end
else
    iter = 1;
    % Para los meses en que no hay ningún evento de lluvia detectado
    TN      = -1;
    FN      = -1;
    TP      = -1;
    FP      = -1;
    Sen     = -1;
    Spe     = -1;
    ACC     = -1;
    F1      = -1;

    % Para cada mes, guardamos las métricas resultantes de cada
    % iteración
    timerval = toc/60;
    Rango0 = ['A' num2str(iter+1)];

```

```

Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];
Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
if nn == 1
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
else
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
end
xlswrite(filename,{Mes_Actual}, Meses(nn,end-6:end), Rango0);
xlswrite(filename,[Sen, Spe, ACC, F1],Meses(nn,end-6:end),Rango1);
xlswrite(filename,[TN, FP, FN, TP, timerval],Meses(nn,end-
6:end),Rango2);
save(Metrics_str, 'Metricas_meses');
nn;

Metricas_i = [Sen, Spe, ACC, F1, TN, FP, FN, TP, timerval];
Metricas = [Metricas; Metricas_i];

end
else
    iter = 1;
    TN = -1;
    FN = -1;
    TP = -1;
    FP = -1;
    Sen = -1;
    Spe = -1;
    ACC = -1;
    F1 = -1;

    % Para cada mes, guardamos las métricas resultantes de cada
    % iteración
    timerval = toc/60;
    Rango0 = ['A' num2str(iter+1)];
    Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];
    Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
    if nn == 1
        Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
    else
        Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
    end
    xlswrite(filename,{Mes_Actual}, Meses(nn,end-6:end), Rango0);
    xlswrite(filename,[Sen, Spe, ACC, F1],Meses(nn,end-6:end),Rango1);
    xlswrite(filename,[TN, FP, FN, TP, timerval],Meses(nn,end-
6:end),Rango2);
    save('Metricas_meses');
    nn;

    Metricas_i = [Sen, Spe, ACC, F1, TN, FP, FN, TP, timerval];
    Metricas = [Metricas; Metricas_i];

end

mat_file = matfile('Metricas.mat','Writable',true);
save(Dir_Resu + '\Metricas.mat','Metricas');
end

```

ANEXO

Anexo 9. Script ParametrosKappa.m

Fuente: Elaboración propia.

```
% for pctTr=0.6:0.1:0.9
%     for maxSeqTr=500:100:1000
%         %for nIter=50:5:100 %de momento fijarlo
%         for nMesTr=1:1:6

            clearvars -except pctTr maxSeqTr nIter nMesTr;
            close all;
            clc
            %Pruebas
            pctTr      = 0.8;
            maxSeqTr   = 500;
            nIter       = 50;
            nMesTr     = 4;
            pctTr100=pctTr*100;

            %% Ubicar el archivo de Excel y los datos
            % Posicionarse en el directorio
            Nom_Expe = string(sprintf('Test pctTr%d maxSeqTr%d nIter%d
nMesTr%d', pctTr100, maxSeqTr, nIter, nMesTr)); %Nombre: experimento
            s1 = 'ResultsV2\';
            s2 = Nom_Expe;
            Dir_Resu = strcat(s1,s2); % Direccion: carpeta del experimento

            %Nombre del Excel con los datos del experimento
            filename = strcat(Dir_Resu, '\Tabla_MetricasV2', '.xlsx');

            % Nombres de las hojas del Excel
            all_sheetnames = {'2020-12'; '2021-01'; '2021-02'; '2021-03'; '2021-
07'; '2021-08'; '2021-11'; '2021-12'};

            %Rango de datos para calcular parametros Kappa
            rTN = 'F2:F51';
            rFP = 'G2:G51';
            rFN = 'H2:H51';
            rTP = 'I2:I51';

            %% Calculo de los parametros kappa
            for i=1:length(all_sheetnames)

                % Datos separados de la matriz de confusion de cada hoja
                TN =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rTN);
                FP =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rFP);
                FN =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rFN);
                TP =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', rTP);

                % Calculo de los parametros kappa
                numerador = 2*(TP.*TN-FN.*FP);
                denominador = (TP+FP).*(FP+TN)+(TP+FN).*(FN+TN);
                kappa = numerador./denominador;
            end
        end
    end
end
```

```
    %Media y desviacion de kappa
    mean_kappa = mean(kappa);
    sigma_kappa = std(kappa);

    %Guardado de los parametros kappa
    title_kappa = 'kappa';
    title_mean = 'Media';
    title_sigma = 'Desviacion';

    writematrix(title_kappa,filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','K
1');

    writematrix(title_mean,filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','J5
2');

    writematrix(title_sigma,filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','J
53');
        writematrix(kappa,
filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','K2:K51');
        writematrix(mean_kappa,
filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','K52');

    writematrix(sigma_kappa,filename,"Sheet",cell2mat(all_sheetnames(i)),'Range','K
53');
        end

%         end
%         %end
%     end
% end
```

ANEXO

Anexo 10. Script BuscarMejorNN.m

```
clear all;
close all;
clc
warning('off','all');
%% Búsqueda del mejor parametro Kappa
%Mejor Kappa inicial
max_kappa      = 0;
max_std        = inf;
max_expe       = 'error';

nIter          = 50;
s1             = 'Results\';
all_sheetnames = {'2020-12';'2021-01';'2021-02';'2021-03';'2021-07';'2021-08';'2021-09';'2021-11';'2021-12'};
Lsh            = length(all_sheetnames);
%Rango de datos de los parametros Kappa
mean_kappa     = 'K52:K52';
std_kappa      = 'K53:K53';
for pctTr=0.6:0.1:0.9
    for maxSeqTr=500:100:1000
        %for nIter=50:5:100 %de momento fijarlo
        for nMesTr=1:1:6

            %pctTr      = 0.8;
            %maxSeqTr   = 750;
            %           nIter      = 50;
            %nMesTr     = 1;
            pctTr100=pctTr*100;

            %% Ubicar el archivo de Excel y los datos
            % Posicionarse en el directorio
            Nom_Expe = string(sprintf('Test pctTr%d maxSeqTr%d nIter%d
nMesTr%d', pctTr100, maxSeqTr, nIter, nMesTr)); %Nombre: experimento
            %           s1 = 'Results\';
            s2 = Nom_Expe;
            Dir_Resu = strcat(s1,s2); % Direccion: carpeta del experimento

            %Nombre del Excel con los datos del experimento
            filename = strcat(Dir_Resu, '\Tabla_Metricas', '.xlsx');

            % Nombres de las hojas del Excel
            all_sheetnames = {'2020-12';'2021-01';'2021-02';'2021-03';'2021-07';'2021-08';'2021-09';'2021-11';'2021-12'};

            %% Calculo de los parametros kappa
            for i=1:1:Lsh
                % Datos asociados a la hoja de datos actual
                act_kappa =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', mean_kappa);
                act_std   =
readmatrix(filename, 'Sheet', cell2mat(all_sheetnames(i)), 'Range', std_kappa);
                act_expe  = strcat(Nom_Expe, ' en ', all_sheetnames{i});

                % Comparacion entre parametros kappa
```

```
        if act_kappa > max_kappa
            max_kappa = act_kappa;
            max_std   = act_std;
            max_expe  = act_expe;
            max_SeqTr = maxSeqTr;
            max_nMesTr = nMesTr;
            max_Mes   = all_sheetnames{i};
            max_Mes_i = i;
        end
    end
end
end
%end
end
end
disp(max_expe);
disp(max_kappa);
disp(max_std);

%Se lanza el programa que crea la mejor red neuronal
S_0 = 2; % =1, Linux; =2, Windows(Gabriel), =3 Windows(Juan Luis)
Vano = 1; % =1 Vano Pico I - Hospital militar, = 2 Vano Pico I - Pico II
Meses = GeneradorMesesf(S_0,Vano);
Meses = char(Meses);
[Im,Jm] = size(Meses);
Mes_Ini = 1; % Aqui debe de ser 1
[Mes_Fin,Im] = size(Meses); % Aqui debe ser hasta el final de la
variable Meses (en principio)
Num_Iter = 1; % OJO: Poner a 50 (u otro valor de interés) cuando
acabe la fase de puesta a punto dle código
N_Mes_Tr = max_nMesTr; % Número de meses anteriores al actual
usados para entrenar
Porc_Train = pctTr; % Porcentaje de secuencias tomadas para
entrenamiento. Para test cogemos todas las secuencias del mes en curso
doTraining = true;
doTesting = true;
doValidation_Same_Tr = true; % Hacemos la validación (durante el entrenamiento)
con datos que entrenaron (true) o no (false)
doTesting_Same_Val = true; % Hacemos el Test con los datos de validación
(true) o del mes actual (false) que no participó en el entrenamiento.
maxEpochs = 8;
maxSeqs_Train = max_SeqTr; % Número máximo de secuencias de
entrenamiento. Con esto prevenimos qe haya demasiadas y se relentice demasiado
el entrenamiento
miniBatchSize = 32; % En muchos trabajos este valor se da como bueno
porque, en general, permite llegar a buenos resultados.
sequenceLength = 150; % Longitud de las secuencias en que se
divide la señal
hopLength = 4;
% CrearMejorNN
CrearMejorNN_V2
```

ANEXO

Anexo 11. Script predictor_v1.m

Fuente: Elaboración propia.

```
if (S_0==1)
    Dir_Estacion =
    'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E/'; %
    Directorio estaciones meteorológicas
    Dir_RSL_Sat = 'Base_Datos/Datos_RSL_Estaciones/RSL_Satelite/';
    % Directorio niveles RSL satelital
    % Gabriel - Windows
elseif (S_0==2)
    if Vano == 2
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E/'; %
        Directorio estaciones meteorológicas
        Dir_RSL_Sat =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-PicoII/';
        % Directorio niveles RSL satelital
    else
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I/';
        Dir_RSL_Sat =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-
        HospitalMilitar/'; % Directorio niveles RSL satelital
    end
elseif (S_0==3)
    if Vano == 2
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Pico_I_Pico_II\C611E/'; %
        Directorio estaciones meteorológicas
    else
        Dir_Estacion =
        'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\Hospital_Pico_I/';
    end
    Dir_RSL_Sat =
    'Base_Datos\Datos_RSL_Estaciones\RSL_Terreno\GRANCANARIAPicoI-PicoII/';
    % Directorio niveles RSL satelital
end

% PARÁMETROS DE LOS EXPERIMENTOS
Nh1 = 1; % Análisis morfológico, número de horas previas
e_est1 = Nh1*60; % Primer análisis morfológico, análisis grueso
e_est2 = 25; % Segundo análisis morfológico, análisis fino

s1 = 'ResultsV3\';
s2 = Nom_Expe;
Dir_Resu = strcat(s1,s2); % Dentro de la carpeta Results

DD = dir(Dir_Resu); % dir mira los archivos que hay dentro
del directorio pasado
if ( isempty(DD) ) % Crea el archivo si no existe ya
    mkdir(Dir_Resu);
end
```

ANEXO

```
if (S_0==1)
    Metrics_str = [Dir_Resu '/Quality_Metrics'];
elseif (S_0==2)
    Metrics_str = [Dir_Resu '\Quality_Metrics'];
elseif (S_0==3)
    Metrics_str = [Dir_Resu '\Quality_Metrics'];
end

fid0 = fopen(strcat(Metrics_str(1),Metrics_str(2)));
if ( fid0<0 )
    Metricas_meses = zeros(Num_Iter,Im,16); % Numero de meses excels distintos
else
    load(Metrics_str);
    fclose(fid0);
end

% Se crea un excel en la carpeta creada denominada Tabla_Metricas.xlsx y coloca
los titulos de los datos
filename = strcat(Dir_Resu,'\Tabla_Metricas','.xlsx');
A_Met = {'Mes', 'Sensibilidad', 'Especificidad', 'Exactitud', 'F1-
Score'};
A_Con = {'TN', 'FP', 'FN', 'TP', 'Tiempo'};
A_diver_Met = {'Sensibilidad2', 'Especificidad2', 'Exactitud2', 'F1-Score2'};
A_diver_Con = {'TN2', 'FP2', 'FN2', 'TP2'};

Metricas = [];

for nn=Mes_Ini:Mes_Fin % Recorreremos los meses de interés

    Mes_Actual = Meses(nn,end-6:end);

    tic % inicio el calculo de tiempo de computo
    % Apertura de la tabla de resultados con las métricas, mes a mes
    Rango1 = ['A' num2str(1) ':' 'E' num2str(1)];
    Rango2 = ['F' num2str(1) ':' 'J' num2str(1)];
    Rango3 = ['K' num2str(1) ':' 'N' num2str(1)];
    Rango4 = ['O' num2str(1) ':' 'R' num2str(1)];

    % Escribe las columnas anteriores en el fichero excel
    xlswrite(filename,A_Met,Meses(nn,end-6:end),Rango1);
    xlswrite(filename,A_Con,Meses(nn,end-6:end),Rango2);
    xlswrite(filename,A_diver_Met,Meses(nn,end-6:end),Rango3);
    xlswrite(filename,A_diver_Con,Meses(nn,end-6:end),Rango4);

    %Aquí se crea el excel y se añaden las columnas
    fid0 = fopen(strcat(Dir_RSL_Sat,[Meses(nn,:) '.mat']));
    fid1 = fopen(strcat(Dir_Estacion,[Meses(nn,:) '.mat']));
    if ( (fid0>0) & (fid1>0) ) % Los datos de interés ya se han generado en una
ejecución previa ...
        load(strcat(Dir_RSL_Sat,[Meses(nn,:) '.mat']));
        fclose(fid0);
        load(strcat(Dir_Estacion,[Meses(nn,:) '.mat']));
        fclose(fid1);
    else % ... Si no se generan antes de la detección
        Sincronismo_Satelital_Mes_COMBINADO6;
    end
end
```

```

Mes_1 = max(1,nn-N_Mes_Tr); % Primer mes usado para entrenar
Mes_2 = max(1,nn-1);      % Segundo mes usado para entrenar

if (S_0==1)
    k_str = [Dir_Resu '/k_optimo'];
elseif (S_0==2)
    k_str = [Dir_Resu '\k_optimo'];
elseif (S_0==3)
    k_str = [Dir_Resu '\k_optimo'];
end

% Generan los archivos .mat con sus nombres
if (nn==1) % Si es el primer mes de toda la iteracion
    k_str = strcat(k_str,['_' Meses(nn,end-6:end) '_' Meses(nn,end-
6:end)]);
else
    for hh=Mes_1:Mes_2
        k_str = strcat(k_str,['_' Meses(hh,end-6:end)]);
    end
end

Bucle_Datos_Train_Test_Tris8;
if ( sum(Label_present)>0 )
    trainFeatureCell = {};
    trainLabelCell   = {};
    index            = 1;
    solape_Labels    = zeros(Lg,Lg);
    [Iseq,Jseq]      = size(rain_sequence);
    for ss=1:Jseq
        if ( Label_present(ss)==1 ) % Si hay lluvia se toma la secuencia
            trainFeatureCell{end+1} = rsl_sequence{ss};
            trainLabelCell{end+1}   = categorical(rain_sequence{ss});
            if ss ~= 1 % cuando ss toma el valor de 1, al hacer ss-1 da
error ya que no hay indice 0 en matlab
                if ( Label_present(ss-1)==0 ) % Si la secuencia
inmediatamente anterior es de no-lluvia se toma
                    trainFeatureCell{end+1} = rsl_sequence{ss};
                    trainLabelCell{end+1}   =
categorical(rain_sequence{ss});
                end
            end
        end
        detected_Labels = zeros(1,Lg);
        detected_Labels = mean(solape_Labels,1) > 0;

        [Itr,Num_secs] = size(trainFeatureCell); %Apartado para generar las
muestras de entrenamiento y test de forma aleatoria, no heuristico
        Num_secs       = min(maxSeqs_Train,Num_secs); % Limitamos el número de
ejemplos que entrenan

% Características y etiquetas en los meses de test
Mes_1 = nn; % Testeamos sobre el mes en curso
Mes_2 = nn; % Testeamos sobre el mes en curso

```

```

Bucle_Datos_Train_Test_Tris8;

if ( sum(Label_present)>0 )
testFeatureCell = {};
testLabelCell   = {};
index           = 1;
solape_Labels   = zeros(Lg,Lg);
[Iseq,Jseq]     = size(rain_sequence);
for ss=1:Jseq
    if ( Label_present(ss)==1 ) % Si hay lluvia se toma la
secuencia
        testFeatureCell{end+1} = rsl_sequence{ss};
        testLabelCell{end+1}   = categorical(rain_sequence{ss});
        if ss ~= 1 % cuando ss toma el valor de 1, al hacer ss-1 da
error ya que no hay indice 0 en matlab
            if ( Label_present(ss-1)==0 ) % Si la secuencia
inmediatamente anterior es de no-lluvia se toma
                testFeatureCell{end+1} = rsl_sequence{ss};
                testLabelCell{end+1}   =
categorical(rain_sequence{ss});
            end
        end
    end
end
testFeatureCell = testFeatureCell(:);
testLabelCell   = testLabelCell(:);
[Itr_ts,Num_secs_Ts] = size(testFeatureCell); % Apartado para
generar las muestras de entrenamiento y test de forma aleatoria, no heuristico
Num_secs_Ts       = min(Num_secs_Ts,maxSeqs_Train); % Como
mucho, habrá tantas secuencias de test como de train

% Train the LSTM Network
if (S_0==1)
    k_str = [Dir_Resu '/Model'];
elseif (S_0==2)
    k_str = [Dir_Resu '\Model'];
end
if (nn==1)
    k_str = strcat(k_str,['_' Meses(nn,end-6:end) '_' Meses(nn,end-
6:end)]);
else
    for hh=Mes_1:Mes_2
        k_str = strcat(k_str,['_' Meses(hh,end-6:end)]);
    end
end
k_str = strcat(k_str, '.mat');

for iter=1:Num_Iter % En cada mes iteramos
    Num_Perm       = randperm(Num_secs);
    Num_Tr         = round(Porc_Train*Num_secs);
    ind_Tr         = Num_Perm(1:Num_Tr);

    Rango1         = ['B' num2str(iter+1) ':' 'E' num2str(1)];
    Rango2         = ['F' num2str(iter+1) ':' 'I' num2str(1)];
    if doTraining

```

```

        layers = [ ...
            sequenceInputLayer(
size(trainFeatureCell,1),"Name","Entrada" )
            bilstmLayer(200,"OutputMode","sequence","Name","Capa
1")
            bilstmLayer(100,"OutputMode","sequence","Name","Capa
2")
            fullyConnectedLayer(2,"Name","fullyConnected")
            softmaxLayer("Name","Softmax")
            classificationLayer("Name","Classification")
        ];

        if doValidation_Same_Tr
            validationFeatureCell = trainFeatureCell(1,ind_Tr).';
            validationLabelCell = trainLabelCell(1,ind_Tr).';
        else
            validationFeatureCell =
trainFeatureCell(1,Num_Perm(Num_Tr+1:end)).';
            validationLabelCell =
trainLabelCell(1,Num_Perm(Num_Tr+1:end)).';
        end
        options = trainingOptions("adam", ...
            "MaxEpochs",maxEpochs, ...
            'ExecutionEnvironment','auto', ...
            "MiniBatchSize",miniBatchSize, ...
            "Shuffle","every-epoch", ...
            "Verbose",0, ...
            "SequenceLength",sequenceLength, ...
            "ValidationFrequency",10, ...

"ValidationData",{validationFeatureCell,validationLabelCell}, ...
            "Plots","none", ... %"training-progress"
            "LearnRateSchedule","piecewise", ...
            "LearnRateDropFactor",0.1, ...
            "LearnRateDropPeriod",5);

        % ENTRENAMIENTO
        [rainDetectNet,netInfo] =
trainNetwork(trainFeatureCell(1,ind_Tr),trainLabelCell(1,ind_Tr),layers,options
);
        disp(['Mes ' Meses(nn,end-6:end) ' Iteración: '
num2str(iter) ' Validation accuracy: '
num2str(netInfo.ValidationAccuracy(end))])
            %save(k_str,'rainDetectNet','netInfo');
            save(Dir_Resu + '\'+Nom_Expe+'.mat','rainDetectNet','netInfo');
            %delete(findall(0)); % Cerrar la ventana de Training
Progress par evitar que se acumulen muchas

        end
        % Test: Predictor
        % Toma la red creada, luego por cada secuencia genera una
prediccion de la siguiente secuencia y crea la matriz de confusion.
        % Una vez esten todas las matrices de confusion (2949) hace la
media aritmetica de sus valores para crear una unica

```

```

% matriz promediada entre todas y escribe sus resultados en un
excel.

if doTesting
    if ( ~isempty(testFeatureCell) ) % Podría estar vacía de
eventos de lluvia
        %% Predictor
        % Realiza la predicción
        predictedLabels = predict(rainDetectNet,
testFeatureCell);

        pred_tmp          = {};
        pred_confmat      = {};
        pred_confmat_diver = {};
        testLabelCell = testLabelCell';

        % Resultado globales de todas las secuencias
        total_TN = 0;
        total_FN = 0;
        total_TP = 0;
        total_FP = 0;

        % Resultados globales de las secuencias de lluvia
        total_diver_TN = 0;
        total_diver_FN = 0;
        total_diver_TP = 0;
        total_diver_FP = 0;

        % BUCLE
        for ii=1:(length(predictedLabels)-1)
            lee_prob_0 = predictedLabels{ii}(1,:);
            lee_prob_1 = predictedLabels{ii}(2,:);
            detector = lee_prob_1 >= lee_prob_0;
            pred_tmp{ii} = categorical(detector, [0, 1], {'0',
'1'});

            % Matriz de confusion
            pred_confmat{ii} =
confusionmat(pred_tmp{ii},testLabelCell{ii+1});

            % Matriz de confusion de secuencias distintas a
solo casos de lluvia (150,0;0,0)
            if pred_confmat{ii}(1,1) == 150
                else
                    pred_confmat_diver{ii}=pred_confmat{ii};

                    % Resultado sumado de las matrices distintas a
(150,0;0,0)
                    total_diver_TN = total_diver_TN +
pred_confmat_diver{ii}(1,1);
                    total_diver_FN = total_diver_FN +
pred_confmat_diver{ii}(2,1);
                    total_diver_FP = total_diver_FP +
pred_confmat_diver{ii}(1,2);
                    total_diver_TP = total_diver_TP +
pred_confmat_diver{ii}(2,2);

```

```

end

% Resultado sumado de todas las matrices
total_TN = total_TN + pred_confmat{ii}(1,1);
total_FN = total_FN + pred_confmat{ii}(2,1);
total_FP = total_FP + pred_confmat{ii}(1,2);
total_TP = total_TP + pred_confmat{ii}(2,2);
end

% Medias de la matriz de confusion global
media_acc = (total_TP + total_TN) / (total_TP +
total_TN + total_FP + total_FN); % Exactitud
Sensibilidad
media_sen = total_TP / (total_FN + total_TP); %
Especificidad
media_spe = total_TN / (total_TN + total_FP); %
media_F1 = total_TP/(total_TP + 0.5 * (total_FN +
total_FP)); %F-Score

% Medias de las matrices de confusion con eventos
media_diver_acc = (total_diver_TP + total_diver_TN) /
(total_diver_TP + total_diver_TN + total_diver_FP + total_diver_FN);
media_diver_sen = total_diver_TP / (total_diver_FN +
total_diver_TP);
media_diver_spe = total_diver_TN / (total_diver_TN +
total_diver_FP);
media_diver_F1 = total_diver_TP/(total_diver_TP + 0.5
* (total_diver_FN + total_diver_FP));

testLabelCell = testLabelCell';
Metricas_meses(iter,nn,1:4) = [total_TN, total_FP,
total_FN, total_TP];
Metricas_meses(iter,nn,5:8) = [media_sen, media_spe,
media_acc, media_F1];
Metricas_meses(iter,nn,9:12) = [total_diver_TN,
total_diver_FP, total_diver_FN, total_diver_TP];
Metricas_meses(iter,nn,13:16) = [media_diver_sen,
media_diver_spe, media_diver_acc, media_diver_F1];
else
% Con el -100 indicamos que no hay datos válidos por
falta de ejemplos de test
Metricas_meses(iter,nn,:) = [-100, -100, -100, -100,-
100, -100, -100, -100,-100, -100, -100, -100, -100, -100];
end
end
%% GUARDAR EN EL XLSX TRAS CADA ITERACION
% Para cada mes, se guardan las metricas resultantes de cada
iteracion

timerval = toc/60;

Rango0 = ['A' num2str(iter+1)];
Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];
Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
Rango3 = ['K' num2str(iter+1) ':' 'N' num2str(iter+1)];
Rango4 = ['O' num2str(iter+1) ':' 'R' num2str(iter+1)];

```

```

        if nn == 1
            Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
        else
            Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
        end

        Metricas_i = [media_sen, media_spe, media_acc, media_F1,
total_TN, total_FP, total_FN, total_TP, timerval,...
                    media_diver_sen, media_diver_spe, media_diver_acc,
media_diver_F1, total_diver_TN, total_diver_FP, total_diver_FN,
total_diver_TP];
        [Inan]      = isnan(Metricas_i);
        [Inan,Jnan] = find(Inan==1);
        Metricas_i(Jnan)= -1*ones(size(Jnan));
        Metricas = [Metricas; Metricas_i];

        xlsxwrite(filename,{Mes_Actual}, Meses(nn,end-6:end), Rango0);
        xlsxwrite(filename,[media_sen, media_spe, media_acc,
media_F1],Meses(nn,end-6:end),Rango1);
        xlsxwrite(filename,[total_TN, total_FP, total_FN, total_TP,
timerval],Meses(nn,end-6:end),Rango2);
        xlsxwrite(filename,[media_diver_sen, media_diver_spe,
media_diver_acc, media_diver_F1],Meses(nn,end-6:end),Rango3);
        xlsxwrite(filename,[total_diver_TN, total_diver_FP,
total_diver_FN, total_diver_TP],Meses(nn,end-6:end),Rango4);
        %save(Metrics_str,'Metricas_meses');
        save('Metricas_meses');
        nn;

    end
else
    iter = 1;
    % Para los meses en que no hay ningún evento de lluvia detectado
    total_TN = -1;
    total_FN = -1;
    total_TP = -1;
    total_FP = -1;
    media_sen = -1;
    media_spe = -1;
    media_acc = -1;
    media_F1 = -1;

    total_diver_TN = -1;
    total_diver_FN = -1;
    total_diver_TP = -1;
    total_diver_FP = -1;
    media_diver_sen = -1;
    media_diver_spe = -1;
    media_diver_acc = -1;
    media_diver_F1 = -1;

    % Para cada mes, guardamos las métricas resultantes de cada
iteración
    timerval = toc/60;
    Rango0 = ['A' num2str(iter+1)];
    Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];

```

```

Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
if nn == 1
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
else
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
end

xlswrite(filename, {Mes_Actual}, Meses(nn, end-6: end), Rango0);
xlswrite(filename, [media_sen, media_spe, media_acc,
media_F1], Meses(nn, end-6: end), Rango1);
xlswrite(filename, [total_TN, total_FP, total_FN, total_TP,
timerval], Meses(nn, end-6: end), Rango2);
xlswrite(filename, [media_diver_sen, media_diver_spe,
media_diver_acc, media_diver_F1], Meses(nn, end-6: end), Rango3);
xlswrite(filename, [total_diver_TN, total_diver_FP, total_diver_FN,
total_diver_TP], Meses(nn, end-6: end), Rango4);

save(Metrics_str, 'Metricas_meses');
nn;

Metricas_i = [media_sen, media_spe, media_acc, media_F1, total_TN,
total_FP, total_FN, total_TP, timerval, ...
media_diver_sen, media_diver_spe, media_diver_acc,
media_diver_F1, total_diver_TN, total_diver_FP, total_diver_FN,
total_diver_TP];
Metricas = [Metricas; Metricas_i];

end
else
    iter = 1;
    total_TN = -1;
    total_FN = -1;
    total_TP = -1;
    total_FP = -1;
    media_sen = -1;
    media_spe = -1;
    media_acc = -1;
    media_F1 = -1;

    total_diver_TN = -1;
    total_diver_FN = -1;
    total_diver_TP = -1;
    total_diver_FP = -1;
    media_diver_sen = -1;
    media_diver_spe = -1;
    media_diver_acc = -1;
    media_diver_F1 = -1;

% Para cada mes, guardamos las métricas resultantes de cada iteración
timerval = toc/60;

Rango0 = ['A' num2str(iter+1)];
Rango1 = ['B' num2str(iter+1) ':' 'E' num2str(iter+1)];
Rango2 = ['F' num2str(iter+1) ':' 'J' num2str(iter+1)];
Rango3 = ['K' num2str(iter+1) ':' 'N' num2str(iter+1)];
Rango4 = ['O' num2str(iter+1) ':' 'R' num2str(iter+1)];

```

```
if nn == 1
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}$', 'match');
else
    Mes_xlsx = regexp(k_str, '\d{0,4}-\d{0,2}', 'match');
end

xlswrite(filename, {Mes_Actual}, Meses(nn, end-6:end), Rango0);
xlswrite(filename, [media_sen, media_spe, media_acc,
media_F1], Meses(nn, end-6:end), Rango1);
xlswrite(filename, [total_TN, total_FP, total_FN, total_TP,
timerval], Meses(nn, end-6:end), Rango2);
xlswrite(filename, [media_diver_sen, media_diver_spe, media_diver_acc,
media_diver_F1], Meses(nn, end-6:end), Rango3);
xlswrite(filename, [total_diver_TN, total_diver_FP, total_diver_FN,
total_diver_TP], Meses(nn, end-6:end), Rango4);

save('Metricas_meses');
nn;

Metricas_i = [media_sen, media_spe, media_acc, media_F1, total_TN,
total_FP, total_FN, total_TP, timerval, ...
media_diver_sen, media_diver_spe, media_diver_acc, media_diver_F1,
total_diver_TN, total_diver_FP, total_diver_FN, total_diver_TP];
Metricas = [Metricas; Metricas_i];

end

mat_file = matfile('Metricas.mat', 'Writable', true);
save(Dir_Resu + '\Metricas.mat', 'Metricas');
end
```