



Aplicaciones del Aprendizaje: Modelado y Pronóstico
de la Serie Temporal de ^{222}Rn en el Laboratorio
Subterráneo de Canfranc (LSC)

Autor : D. Iván Méndez Jiménez

Director : Dr. Miguel Ángel Gutiérrez Naranjo

Codirector: Dr. Miguel Cárdenas Montes

Máster Universitario en Lógica, Computación e Inteligencia Artificial

Trabajo Fin de Máster

Curso Académico 2017/2018

Trabajo Fin de Máster

Aplicaciones del Aprendizaje: Modelado y Pronóstico de la Serie Temporal de ^{222}Rn en el Laboratorio Subterráneo de Canfranc (LSC)

Autor : D. Iván Méndez Jiménez

Director : Dr. Miguel Ángel Gutiérrez Naranjo

Codirector : Dr. Miguel Cárdenas Montes

La defensa del presente Proyecto Fin de Máster se realizó el día de
de 2018, siendo calificado por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Sevilla, a de de 2018

Autorización de difusión

Por la presente,

D. Iván Méndez Jiménez, con D.N.I. 51482398-A, como autor del Trabajo de Fin de Máster titulado 'Aplicaciones del Aprendizaje: Modelado y Pronóstico de la Serie Temporal de ²²²Rn en el Laboratorio Subterráneo de Canfranc (LSC)', del Máster Universitario en Lógica, Computación e Inteligencia Artificial, correspondiente al curso 2017/2018.

Autoriza,

a la Universidad de Sevilla a su depósito y consulta, en cualquier formato, tanto de forma completa como en parte, en la biblioteca de la Escuela Técnica Superior de Ingeniería Informática.

Y para que conste, firma en Madrid a de Junio de 2018.

Firma: D. Iván Méndez Jiménez

La mejor forma de predecir

el futuro es implementarlo

David Heinemeier Hansson

Agradecimientos

El primer párrafo de mis agradecimientos va dirigido a mis directores del Trabajo de Fin de Máster. Al Dr. Miguel Ángel Gutierrez Naranjo, por su disposición y por aceptar la dirección y el seguimiento a distancia del presente trabajo, tarea que no resulta fácil cuando se trata de proyectos de esta envergadura. Y al Dr. Miguel Cárdenas Montes, por todo lo aprendido con él a lo largo de estos dos últimos años y por transmitirme un continuo entusiasmo en sus labores investigadoras. Gracias a ambos por vuestro tiempo, apoyo, dedicación y confianza.

Quiero dar las gracias a mis padres por su amor, su paciencia y su apoyo siempre incondicional, tanto en los buenos como en los malos momentos. Gracias por enseñarme la importancia de la responsabilidad, el esfuerzo y la constancia. Y a mi hermana Marta, por su cariño y por creer siempre en mis posibilidades para alcanzar los objetivos que me propongo, por su escucha y por ser mi mayor confidente.

Asimismo, quiero dar las gracias a Estela y a Carlos, por todos los buenos momentos compartidos y por escucharme, animarme y estar siempre ahí. Gracias a mis compañeros y amigos del CIEMAT, porque con ellos las risas y los buenos momentos siempre están garantizados. Y gracias a todos los demás compañeros y amigos por sus consejos y por ayudarme a desconectar cuando realmente lo necesitaba.

También agradezco al Dr. Roberto Santorelli tanto su aportación en la motivación del presente trabajo, como el habernos proporcionado los datos que serán analizados en el mismo. Gracias a todos los demás doctores y compañeros en el CIEMAT por su valiosa ayuda en el proyecto realizado, y a todo el profesorado del máster por estos años académicos tan enriquecedores.

Finalmente, quiero agradecer a la Universidad de Sevilla las enseñanzas adquiridas, al CIEMAT la experiencia laboral conseguida y a la Unidad de Excelencia María de Maeztu el haber financiado el artículo escrito en relación a este trabajo.

Resumen

El nivel de ^{222}Rn (isótopo de Radón) en laboratorios subterráneos, dónde se alojan los experimentos de Física de bajo fondo, constituye la mayor fuente de dicho fondo. Así, esta se considera la principal distorsión a la hora de obtener resultados de alta precisión. En España, el Laboratorio Subterráneo de Canfranc alberga experimentos innovadores como el *Argon Dark Matter-It Experiment*, encaminado a la búsqueda directa de materia oscura en el universo. Para las colaboraciones que explotan estos experimentos, el análisis, modelado y pronóstico del nivel de ^{222}Rn es una tarea muy relevante para alcanzar una planificación eficiente de las actividades de instalación y mantenimiento dentro del laboratorio. En el presente trabajo, se analizan cuatro años completos de valores de ^{222}Rn medidos cada diez minutos entre Julio de 2013 y Junio de 2017 en el Laboratorio Subterráneo de Canfranc. Para dicho análisis, y posterior pronóstico, se utilizan metodologías clásicas como: el suavizado exponencial *Holt-Winters*, los Promedios Móviles Integrados Autorregresivos y la Descomposición en Estacionalidad y Tendencia usando el método *Loess*. Además, se usan técnicas de aprendizaje automático y profundo tales como: Redes Neuronales *Feed-Forward*, Redes Neuronales Recurrentes y Redes Neuronales Convolucionales. Con el objetivo de evaluar el rendimiento de estos métodos, se van a usar tanto el Error Cuadrático Medio como el Error Absoluto Medio. Ambas métricas determinan que las técnicas basadas en aprendizaje automático y profundo son las que mejores resultados predictivos presentan. Es importante señalar que esta es la primera vez que se investigan los datos mencionados y constituye un excelente ejemplo de series temporales científicas con implicaciones relevantes para la calidad de los resultados obtenidos en experimentos destacados.

Palabras clave: *Series temporales, Aprendizaje Automático y Profundo, Redes Neuronales Recurrentes, Redes Neuronales Convolucionales, Pronóstico, ^{222}Rn , Laboratorio Subterráneo de Canfranc*

Abstract

The ^{222}Rn (Radon isotope) level at underground laboratories, where Physics experiments of low-background are installed, is the largest source of background. Thus, it is the main distortion for obtaining high accuracy results. At Spain, the Canfranc Underground Laboratory hosts ground-breaking experiments, such as Argon Dark Matter-1t aimed at the dark matter direct searches in the universe. For the collaborations exploiting these experiments, the analyzing, modelling and forecasting of the ^{222}Rn level is a very relevant task for efficient planning activities of installation and maintenance in the laboratory. In this work, four complete years of values of ^{222}Rn level measured every ten minutes between July 2013 and June 2017 in the Canfranc Underground Laboratory, are analysed. For this analysis, and later forecasting, classic methodologies are used, such as: Holt-Winters exponential smoothing, AutoRegressive Integrated Moving Averages, and Seasonal and Trend Decomposition using *Loess*. Furthermore, Machine and Deep Learning techniques are also used, such as: Feed-Forward Neural Networks, Recurrent Neural Networks and Convolutional Neural Networks. In order to evaluate the performance of these methods, both the Mean Squared Error and the Mean Absolute Error are used. Both metrics determine that the Machine and Deep Learning techniques, are those which obtain the best predictive results. It is important to note that this is the first time that the mentioned data are investigated, and it constitutes an excellent example of scientific time series with relevant implications for the quality of the scientific results of featured experiments.

Keywords: *Time Series, Machine and Deep Learning, Recurrent Neural Networks, Convolutional Neural Networks, Forecasting, ^{222}Rn , Canfranc Underground Laboratory*

Índice general

1. Introducción	1
1.1. Motivación del estudio de la serie temporal de ^{222}Rn en el LSC	3
1.2. Estructura de la memoria	4
2. Objetivos	5
2.1. Objetivos específicos	6
3. Estado del arte	9
3.1. Metodologías de Pronóstico	9
3.1.1. Técnicas Clásicas	11
3.1.2. Técnicas basadas en Aprendizaje Automático y Profundo	18
3.2. Lenguajes y <i>Frameworks</i>	25
4. Configuración y preprocesamiento de datos	27
4.1. Agrupamiento de la información y correlación con el Hall B del LSC	30
4.2. Correlaciones con las variables climatológicas internas	34
5. Resultados	37
5.1. Métricas de evaluación de rendimiento	38
5.2. Resultados experimentales por técnicas	39
5.2.1. Resultados experimentales con <i>Holt-Winters</i>	39
5.2.2. Resultados experimentales con <i>ARIMA</i>	40
5.2.3. Resultados experimentales con Descomposición STL	41
5.2.4. Resultados experimentales con Redes Neuronales <i>Feed-Forward</i>	45
5.2.5. Resultados experimentales con Redes Neuronales Recurrentes	47

5.2.6. Resultados experimentales con Redes Neuronales Convolucionales . . .	49
5.3. Comparativa general de metodologías	54
6. Conclusiones	57
6.1. Trabajos futuros	58
6.2. Publicaciones	59
Bibliografía	63

Índice de figuras

3.1. Estructuras más relevantes de Redes Neuronales Recurrentes.	21
3.2. Ejemplo de estructura de una CNN con dos capas de convolución para el procesamiento de una imagen.	24
3.3. Ejemplo de unidad de <i>pooling</i> aprendiendo a ser invariante ante las transformaciones de la entrada.	25
4.1. Datos en bruto: valores de ^{222}Rn en Bq/m^3 en el Hall A del LSC.	28
4.2. Boxplots de las medianas mensuales del ^{222}Rn en el Hall A del LSC.	31
4.3. Niveles de ^{222}Rn en el Hall A del LSC agrupados por meses.	31
4.4. Boxplots de las medianas mensuales del ^{222}Rn en el Hall B del LSC.	32
4.5. Medianas mensuales del ^{222}Rn en el Hall A y en el Hall B del LSC.	32
4.6. Correlación Hall A - Hall B de las medianas mensuales de ^{222}Rn	33
4.7. Correlación de las medianas mensuales del ^{222}Rn con las de la temperatura. . .	34
4.8. Correlación de las medianas mensuales del ^{222}Rn con las de la presión del aire. .	35
4.9. Correlación de las medianas mensuales del ^{222}Rn con las de la humedad relativa. .	35
5.1. Comparativa entre datos de evaluación reales y pronosticados por <i>Holt-Winters</i> para el periodo comprendido entre Julio de 2016 y Junio de 2017.	40
5.2. Comparativa entre datos de evaluación reales y pronosticados por <i>ARIMA</i> para el periodo comprendido entre Julio de 2016 y Junio de 2017.	41
5.3. Descomposición STL periódica original y a la misma escala para la serie temporal de las medianas de ^{222}Rn	43
5.4. Descomposición STL no periódica original y a la misma escala para la serie temporal de las medianas de ^{222}Rn	44

5.5. Comparativa entre datos de evaluación reales y pronosticados por la descomposición STL periódica y no periódica, para el periodo comprendido entre Julio de 2016 y Junio de 2017.	45
5.6. Comparativa entre los datos de evaluación reales y pronosticados por una Red Neuronal <i>Feed-Forward</i> , para el periodo comprendido entre Julio de 2016 y Junio de 2017.	46
5.7. Comparativa entre datos de evaluación reales y pronosticados por una Red Neuronal Recurrente con neuronas LSTM, para el periodo comprendido entre Julio de 2016 y Junio de 2017.	48
5.8. Ejemplo de imagen (utilizada como muestra de entrenamiento) formada a partir de las componentes normalizadas de la descomposición STL no periódica de la serie temporal de medianas mensuales de ^{222}Rn . Las filas, de arriba a abajo, hacen referencia a la Estacionalidad, la Tendencia y la Componente Aleatoria. .	53
5.9. Comparativa entre datos de evaluación reales y pronosticados por una CNN unidimensional, una CNN unidimensional que hace uso de las componentes de la descomposición STL no periódica del ^{222}Rn , y una CNN bidimensional que hace uso de lo mismo, para el periodo comprendido entre Julio de 2016 y Junio de 2017.	53
5.10. Comparativa general entre los datos de evaluación reales empleados y los pronosticados por todos y cada uno de los modelos analizados en el presente estudio, para el periodo comprendido entre Julio de 2016 y Junio de 2017. . . .	55

Índice de cuadros

4.1. Periodos de ausencia de datos más largos.	29
5.1. Predictores usados, media y desviación estándar del ECM y el EAM para cada uno de ellos tras 25 ejecuciones.	54

Capítulo 1

Introducción

Una serie de tiempo es una secuencia de observaciones de una o más variables, tomada a lo largo de una ventana temporal concreta y equitativamente espaciada, cronológicamente hablando. No obstante, la mayoría de las veces, cuando trabajamos con amplios conjuntos de datos, puede ocurrir que muchos de esos datos no se encuentren disponibles.

Entre las principales tareas que se pueden llevar a cabo cuando se trabaja con series de tiempo, cabe destacar su modelado y la predicción o pronóstico de las observaciones ausentes y el comportamiento futuro de dichas series temporales.

Antes de entrar en detalles acerca de los métodos y técnicas más relevantes para llevar a cabo estas labores, es necesario introducir las características principales de las series temporales, cómo se clasifican y de qué se componen. De esta manera, se puede decir que existen dos grandes grupos de series temporales: las discretas y las continuas. Si sus valores pueden ser predichos de manera exacta, diremos que nuestra serie temporal es determinista. Por su parte, si los valores no pueden ser predichos de manera exacta y tenemos que hacer uso de valores pasados con objeto de calcular los valores futuros, entonces decimos que nuestra serie temporal es estocástica.

Por otro lado, y continuando con la taxonomía de las series temporales, se puede hablar de series estacionarias y no estacionarias. Dentro del grupo de las estacionarias las series son estables, de forma que la media y la varianza de las observaciones que las componen permanecen constantes a lo largo del tiempo. En el caso de las series no estacionarias, la media y la varianza cambian en el tiempo. De esta manera, este segundo grupo introduce el concepto de tendencia en la evolución de dichas series.

Los principales objetivos del análisis de series temporales son los indicados a continuación:

1. En primer lugar, es necesario llevar a cabo una descripción de los datos. Una de las maneras más comunes y que más nos ayudan a realizar esta labor, es empezar representando gráficamente nuestra serie temporal. Así, por medio de un simple gráfico, empezamos a considerar las medidas descriptivas básicas de la misma. De esta manera, la forma de los datos se corresponde con la tendencia, la periodicidad hace referencia a la estacionalidad, y la existencia de observaciones extrañas o fuera de lugar con respecto al resto de datos se corresponde con la detección de anomalías.
2. En segundo lugar, una vez representados gráficamente los datos en bruto y teniendo en mente cómo estos se comportan de manera visual, se procede a modelar la serie temporal estudiada, analizando tanto los distintos tipos de anomalías existentes en la misma, como las posibles correlaciones con datos similares medidos en lugares cercanos, o con variables ambientales por las que se pueda ver influenciada.
3. Por último, cuando trabajamos con grandes volúmenes de información –algo que resulta muy típico hoy en día–, no solo nos interesa estudiar y modelar dicha información, sino también predecir la información futura o tratar de averiguar la información que nos falta, como si de recomponer un puzzle se tratara. Para ello, necesitamos explorar una serie de técnicas, desde las más clásicas y consolidadas hasta las más emergentes, de manera que podamos determinar cuál es la mejor o cuál aplicar en función del caso de estudio al que nos enfrentemos.

Con todos los conceptos mencionados anteriormente, se puede decir que el presente Trabajo de Fin de Máster se centra en el estudio y pronóstico de una serie temporal no completa, los niveles de ^{222}Rn en el Laboratorio Subterráneo de Canfranc (LSC). Dado que estamos realmente interesados en el modelado y en los valores futuros de dicha serie temporal, trataremos de predecir su comportamiento, centrándonos y haciendo uso de algunas de las técnicas de aprendizaje automático y profundo más comunes en la actualidad, pero sin renunciar a las técnicas clásicas de pronóstico con el objetivo de establecer comparativas y concluir cómo de eficientes son las primeras en comparación con las segundas, ya bien exploradas desde hace bastantes años.

Como se explicará mas adelante en el capítulo 4, los niveles de ^{222}Rn pertenecientes a los meses comprendidos entre Julio de 2013 y Junio de 2017 en el LSC, son recogidos a fin de usarlos como base para la elaboración del presente proyecto. A partir de estos datos, determinaremos cuál es el mejor de los modelos analizados, lo que nos permitirá alcanzar la mejor robustez y precisión en nuestros resultados.

1.1. Motivación del estudio de la serie temporal de ^{222}Rn en el LSC

El ^{222}Rn es un radionucleido producido por las cadenas de decaimiento del ^{238}U (isótopo de Uranio) y del ^{232}Th (isótopo de Torio). Al ser un gas a temperatura ambiente, puede ser emanado por las rocas y el hormigón del laboratorio subterráneo, difundiéndose así en la sala experimental donde, actualmente, se encuentra en desarrollo el experimento *Argon Dark Matter-It*¹ enfocado en la búsqueda de materia oscura. Esta contaminación presente en el aire es una fuente potencial de fondo², tanto de manera directa, como a través de los descendientes radiactivos de larga vida generados en las cadenas de decaimiento comentadas, que pueden adherirse a las superficies experimentales. La contaminación de ^{222}Rn en el aire se puede reducir en unos órdenes de magnitud sólo en áreas cerradas limitadas, purgando el nitrógeno puro atmosférico (N_2) o aire 'libre de Radón' producido por estructuras dedicadas.

En los laboratorios subterráneos profundos, la actividad promedio del ^{222}Rn depende de las condiciones locales y debe monitorizarse constantemente. Por lo general, varía de decenas a cientos de $\frac{\text{Bq}}{\text{m}^3}$, con variaciones periódicas y no periódicas³. Además, en algunos casos, se ha observado una fuerte dependencia estacional o modulación anual del observable [1]. Una comprensión detallada de la periodicidad del ^{222}Rn puede ser fundamental para un entendimiento preciso de los antecedentes de los experimentos de búsqueda de eventos excepcionales.

¹<http://darkmatter.ethz.ch/>

²Una fuente potencial de fondo es cualquier fuente que produzca radiaciones ionizantes que no se correspondan con lo que se pretende medir.

³El Becquerelio (Bq) es unidad derivada del Sistema Internacional de Unidades que mide la actividad radiactiva por segundo, o fenómeno físico por el cual los núcleos de algunos elementos químicos denominados radiactivos, emiten radiaciones.

Al mismo tiempo, la predicción de la evolución de la concentración del ^{222}Rn en el laboratorio puede ser importante para organizar correctamente las operaciones que prevén la exposición de los materiales del detector al aire, minimizando, de este modo, la deposición de los radionucleidos en las superficies.

1.2. Estructura de la memoria

El resto de la memoria de este trabajo se organiza como se especifica a continuación:

- En el capítulo 2 se introducen los objetivos del proyecto, qué es lo que se trata de conseguir de manera general y cuáles son los objetivos específicos en los que se ha desglosado dicho objetivo general, para poder alcanzarlo. Además, se presentan las metodologías usadas y los lenguajes en los que se implementan.
- En el capítulo 3 se exploran y explican las metodologías empleadas en el presente trabajo, tanto las clásicas como las basadas en aprendizaje automático y profundo, que dan pie al mismo. Por otro lado, también se explican los lenguajes y *frameworks* usados para hacer posible y aplicar posteriormente dichas metodologías.
- En el capítulo 4 se detalla el proceso de manipulación, preprocesado y visualización de la información, así como el estudio de las correlaciones tanto en los niveles de Radón entre dos salas contiguas, como entre los niveles de Radón en la sala estudiada y las variables climatológicas internas medidas en la misma.
- Por su parte, en el capítulo 5 se presentan las métricas usadas para evaluar el rendimiento/eficiencia de las metodologías empleadas, se detallan el diseño y los parámetros de configuración de nuestros algoritmos, y se especifican y muestran los resultados obtenidos por cada una de las técnicas. Finalmente, se realiza una comparativa general.
- Por último, en el capítulo 6 se proporcionan las conclusiones del proyecto, explicando qué es lo que se ha conseguido en base a lo pretendido en el presente proyecto, y estableciendo la línea de trabajos futuros a realizar a partir de lo desarrollado hasta el momento.

Capítulo 2

Objetivos

En el presente trabajo final de Máster, y referente a lo comentado en el capítulo anterior, se pretenden alcanzar dos grandes objetivos: por un lado, modelar la serie temporal de las observaciones de ^{222}Rn presentes en el LSC, a fin de detectar peculiaridades en la misma y mantener un control de sus niveles, garantizando y verificando consecuentemente que dichos niveles no influyen, intervienen ni falsean los resultados obtenidos en el experimento *Argon Dark Matter-It*, actualmente en desarrollo en el LSC. Por otro lado, y dada la cantidad de información de la que se dispone para llevar a cabo el presente estudio, se pretende explorar y usar las técnicas más novedosas y extendidas actualmente en el campo del aprendizaje profundo, como son las Redes Neuronales Profundas, que se han ido posicionando como un modelo computacional eficiente a la hora de resolver diversos problemas tanto de clasificación como de regresión.

Además, se pretende explicar tanto los *frameworks* y herramientas, como los modelos de computación novedosos usados, sin olvidar los modelos clásicos, que siguen resultando a día de hoy muy útiles para resolver una gran cantidad de problemas y que se encuentran mucho más afianzados y explorados. Con esto, se podrán establecer comparativas y se demostrará el gran potencial de algunas de las técnicas más novedosas de aprendizaje profundo, como alternativa real a las técnicas clásicas de clasificación y pronóstico, de cara a su afianzamiento e implantación como técnicas perfectamente viables para su uso en empresas y otras instituciones, a fin de obtener unos mejores resultados y una mayor aplicabilidad en distintos campos como la economía, la tecnología o las ciencias de la salud, entre otros.

Para el desarrollo del proyecto, se usarán los lenguajes de programación Python y R, ya que

disponen de una gran cantidad de módulos y librerías para trabajar con técnicas de Inteligencia Artificial, hacer test estadísticos para corroborar los resultados obtenidos y extraer gráficas representativas de los mismos, ya que al trabajar con series temporales, la representación visual de estas resulta de gran ayuda para su comprensión, estudio y seguimiento. El *framework* usado en Python para trabajar con redes neuronales, será *Keras*¹, recientemente desarrollado y por lo general bastante intuitivo, que actúa como capa intermedia entre el usuario y el *backend* (en este caso, *TensorFlow*²) haciendo la programación más agradable y sencilla. Entre otros *frameworks* comúnmente usados para aprendizaje profundo, caben destacar: *Caffe*³ y *Torch*⁴. Finalmente, en cuanto a modelos de computación se refiere, se usarán los siguientes:

1. **Modelos clásicos:** *Holt-Winters* [2, 3, 4], *ARIMA* [5, 6] y Descomposición en Estacionalidad y Tendencia usando *Loess* (Descomposición STL) [7]. Estos modelos se usarán, y sus resultados serán obtenidos, por medio del lenguaje R.
2. **Modelos basados en aprendizaje automático y profundo:** Redes Neuronales *Feed-Forward* (FFNN, por su acrónimo en inglés) [8], Redes Neuronales Recurrentes (RNN, por su acrónimo en inglés) con *Long Short-Term Memory* (LSTM) [9] y Redes Neuronales Convolucionales (CNN, por su acrónimo en inglés) [9]. Estas redes serán implementadas en Python haciendo uso del *framework Keras* comentado.

2.1. Objetivos específicos

El análisis y pronóstico de series temporales, como parte del tan mencionado *Big Data*, se está haciendo hoy en día cada vez más relevante. El incremento del volumen de datos en la industria, el comercio online, las redes sociales o la ciencia, está demandando un esfuerzo continuo en la comprensión de nueva información que nos permita extraer conclusiones y desarrollar nuevos modelos. Además, muchos de los conjuntos de datos existentes se encuentran ordenados cronológicamente, haciendo que puedan ser analizados mediante técnicas basadas en series temporales, tanto clásicas como las emergentes en el área del aprendizaje profundo.

¹<https://keras.io/>

²<https://www.tensorflow.org/>

³<http://caffe.berkeleyvision.org/>

⁴<http://torch.ch/>

Recientes trabajos realizados en este último campo, concretamente con Redes Neuronales Recurrentes y Convolucionales, demuestran la enorme capacidad que tienen estas técnicas de capturar la suficiente cantidad de información como para alcanzar posteriormente una exactitud en la predicción bastante elevada. Aunque es cierto que las redes convolucionales son más frecuentes en los estudios de clasificación de imágenes o datos bidimensionales, también pueden ser aplicadas al procesamiento de series temporales o datos unidimensionales. Así, cuando se tiene esto último, suele ser típico trabajar con muestras creadas a partir de dichos datos, sobre las cuales la red tratará de detectar posibles patrones.

Para llevar a cabo lo comentado en la sección anterior de objetivos generales, primero se requiere de una fase de asentamiento y exploración de las técnicas basadas en aprendizaje profundo y de los lenguajes y *frameworks* a usar, así como de lectura y conocimiento de las técnicas clásicas de pronóstico. Por este motivo, en los puntos siguientes, se presenta una primera parte teórica de los modelos a usar, a fin de reflejar la base de conocimiento necesaria para poder comprender e introducir el uso de las metodologías empleadas en el presente estudio.

Por otro lado, es importante notar que uno de los objetivos implícitos de este Trabajo de Fin de Máster, es profundizar en el conocimiento de las técnicas aplicadas y de los lenguajes Python y R al mismo tiempo, mayoritariamente usados en la actualidad en campos de la ciencia de los datos, como son el *Data Mining* y el *Big Data*. Estos lenguajes, aparte de proporcionar los *frameworks* y librerías necesarios para la realización del presente proyecto, nos van a permitir realizar todo el preprocesamiento de los datos manipulados y extraer gráficas y representaciones que permitan visualizar y clarificar los resultados obtenidos.

En lo referente al estudio y la extracción de los resultados en sí, tratamos de explorar cada una de las técnicas por separado e ir avanzando en ellas de manera gradual en términos de dificultad, con el objetivo de dominar y adquirir conocimiento suficiente como para poder decidir cuáles son las mejores técnicas a aplicar en cada caso y tener más posibilidades de atacar con éxito otros problemas futuros. Con todo ello, dichos resultados se podrán someter a comparativas generales, dándonos así la capacidad de obtener las oportunas conclusiones y deducir qué tareas se podrían realizar en el futuro en base al trabajo realizado.

Finalmente, y dado el carácter investigador tanto del máster como del presente trabajo final, se ha escrito un artículo de congreso en base al mismo, teniendo en cuenta las restricciones de espacio de la conferencia seleccionada. Así, la conferencia elegida para la cual se ha aceptado

una comunicación con algunos de los aspectos más relevantes del trabajo aquí realizado, es *Hybrid Artificial Intelligence Systems* (HAIS 2018). Dicha comunicación se titula *Modelling and Forecasting of the ^{222}Rn Radiation Level Time Series at the Canfranc Underground Laboratory* [10].

Capítulo 3

Estado del arte

3.1. Metodologías de Pronóstico

Podemos decir que el pronóstico o la predicción de datos es una actividad llevada a cabo hoy en día en un gran número de situaciones. Así, en el caso de la meteorología, nos puede interesar saber cuál va a ser el tiempo en las próximas horas o días o cuál será la probabilidad de precipitaciones futura. En el caso del consumo energético, resulta relevante el pronóstico con objeto de evitar agotar los recursos disponibles en el medio ambiente, tomar medidas para reducir la contaminación atmosférica o predecir nuestros próximos gastos en facturas de electricidad, gas y agua. Por otro lado, otro ejemplo muy típico sería el predecir el número de accidentes de tráfico en determinados tramos de carretera, basándose en información previa de accidentes registrados en esas zonas. Esto podría resultar muy útil de cara a tomar medidas, a fin de reducir la tasa de mortalidad por accidentes. Finalmente, como último ejemplo muy representativo de pronóstico de series temporales, cabría destacar la predicción de los próximos valores bursátiles de una empresa, a fin de lograr un incremento en sus ganancias.

Independientemente del caso de estudio tratado, el pronóstico es una técnica muy relevante a la hora de querer lograr una planificación eficiente. Sin embargo, no es una tarea trivial, dado que la predictibilidad de una variable depende de algunos factores, como:

1. El conocimiento y la comprensión existentes acerca de los factores que influyen en la predicción.
2. El volumen de datos disponible.

3. El entorno en el que tienen lugar las predicciones, del cual necesitamos asumir que está cambiando continuamente. De esta manera, se dice que un buen modelo de pronóstico es aquél que es capaz de adaptarse al entorno en el cual se aplica, considerando no solo las variables, sino también la manera en que estas cambian y/o evolucionan.

De manera general, el proceso de pronóstico engloba dos fases bien diferenciadas, detalladas a continuación:

1. En primer lugar, se necesita definir el problema y recoger y preprocesar los datos que van a ser analizados. En ocasiones, estas tareas pueden resultar bastante duras, puesto que se necesita determinar qué tipos de variables van a ser predichas y cómo va a trabajar la función de pronóstico.

Por otro lado, resulta también difícil recolectar toda la información y darle un formato adecuado para su correcto procesamiento. Normalmente, gran parte del tiempo dedicado a los problemas de pronóstico, reside en la recopilación y el preprocesado de la información a manipular. Hay que darse cuenta que, en este tipo de problemas, es muy importante disponer de una cantidad representativa de datos, a fin de extraer buenas predicciones. No obstante, en el análisis de series temporales –y en *Data Mining* en general–, la presencia de *gaps* o datos ausentes es bastante frecuente.

Como parte final de este primer paso, toda la información recogida se suele representar gráficamente. Esto resulta de gran utilidad a la hora de detectar posibles patrones, tendencia, estacionalidad o la presencia o ausencia de valores *outliers* o anómalos, siendo esto último muy relevante en algunos casos de estudio como por ejemplo la detección de fraudes. En resumen, la información gráfica nos va a permitir decidir cómo empezar a procesar nuestros datos.

2. En segundo lugar, se deben decidir los métodos o modelos de pronóstico que se van a utilizar. Actualmente, y dado el avance existente tanto en técnicas clasificatorias como de regresión, tenemos un amplio abanico de posibilidades. En el presente proyecto, nos centraremos en las técnicas mencionadas en el capítulo 2 para llevar a cabo nuestras predicciones.

Hay que darse cuenta de que la fortaleza de los modelos de pronóstico, depende en gran medida de la disponibilidad de datos pasados. Como este proyecto tiene una componente

exploratoria, se analizarán primero los modelos bien conocidos de pronóstico, para luego entrar en las técnicas más novedosas, extrayendo unos resultados que nos permitirán concluir qué modelos son mejores para nuestro caso de estudio en particular, pero fomentando su uso para otros casos de aplicación de manera genérica.

Cuando el pronóstico se lleva a cabo haciendo uso de las técnicas que se explicarán a continuación, es muy común el uso de intervalos de confianza. De alguna forma, la utilización de intervalos constituye una manera de garantizar que los valores finales extraídos del pronóstico se encuentran dentro de los límites de los mismos. Así, aunque los resultados de nuestras predicciones no sean los deseados, podemos asegurar que van a estar dentro de los intervalos establecidos. A pesar de que cualquier porcentaje puede ser utilizado como límite, típicamente se suele trabajar con intervalos de confianza del 80 % y del 95 %.

3.1.1. Técnicas Clásicas

En este primer apartado de técnicas, se introducen las metodologías clásicas de pronóstico usadas. Se corresponden con técnicas bastante probadas en la actualidad, que además nos permiten modelar la serie temporal usada en nuestro caso de estudio.

Holt-Winters

Normalmente, el método *Holt-Winters* [2, 3, 4], es usado para pronosticar nuevos datos en series temporales que presentan una cierta estacionalidad, que es la repetición de un patrón en intervalos regulares de tiempo.

Para comprender correctamente cómo funciona este método de manera general, es necesario detallar otros métodos similares más simples. De esta forma, a continuación, se empezará introduciendo el método de suavizado exponencial simple, para luego explicar el suavizado exponencial doble y terminar concluyendo con el triple, que es el denominado *Holt-Winters* como tal.

Además, es importante darse cuenta de que las fórmulas que se indicarán y explicarán a continuación, hacen referencia a la versión simplificada de las mismas para los instantes de tiempo t y $t - 1$ en cada uno de los modelos:

1. **Simple Exponential Smoothing.** Tomando como punto de referencia el método de pronóstico de Medidas Móviles con Peso [11], podemos considerar una serie temporal tal que todas las observaciones que la componen tienen un peso asignado. Así, en este primer método, se producirá un decaimiento de dichos pesos a lo largo del tiempo. Por ejemplo, si partimos de 0.6 –un primer peso comprendido entre 0 y 1–, la evolución del mismo en los sucesivos pasos temporales, sería de la forma: $0,6^1$, $0,6^2$, $0,6^3$, etc. y así sucesivamente. El principal problema al que conlleva esto es que, solo la suma de esos 3 primeros valores, ya excede el valor de 1. Sin embargo, una forma de resolverlo consiste en hacer uso de la fórmula del suavizado exponencial simple, que es:

$$\hat{y}_t = \alpha \cdot y_t + (1 - \alpha) \cdot \hat{y}_{t-1}$$

Como se puede observar en esta fórmula, el resultado final del pronóstico es el resultado de la suma de 2 productos. En primer lugar, tenemos el producto entre el coeficiente de suavizado (α) y la observación actual de nuestra serie de tiempo. Por otro lado, el segundo producto hace referencia al anterior valor pronosticado multiplicado por $1 - \alpha$, siendo el primer valor pronosticado de la serie, idéntico al valor real inicial de la misma.

Respecto al coeficiente de suavizado (α), podemos observar cómo ahora la suma de α y $1 - \alpha$, no supera el valor de 1. Por otra parte, haciendo uso de este coeficiente, podemos decidir qué valor influirá más en el resultado del nuevo pronóstico, el valor real o el último valor predicho. Encontrar el valor óptimo de α es, por tanto, una condición necesaria para la obtención de mejores pronósticos.

Haciendo uso de este método, somos capaces de predecir sólo una nueva observación en el futuro. Sin embargo, en la mayor parte de los casos de estudio de pronóstico con series temporales, estamos interesados en predecir un conjunto de valores futuros. Así, en los siguientes puntos se introducen técnicas capaces de obtener más de un valor.

2. **Double Exponential Smoothing.** En este segundo tipo de métodos exponenciales de suavizado, se van a introducir 2 nuevos conceptos que se deben tener en cuenta a la hora de pronosticar un par de valores futuros. El primer concepto es el nivel, cuyo valor es muy similar al obtenido en el pronóstico final resultado de aplicar la fórmula del método de

suavizado exponencial simple. La peculiaridad es que, en este caso, este no es el resultado de la predicción, sino solamente una parte de ella. Se define el nivel como:

$$l_t = \alpha \cdot y_t + (1 - \alpha) \cdot (l_{t-1} + b_{t-1}), \text{ con } 0 \leq \alpha \leq 1$$

Vemos que, en esta fórmula, se debe considerar ahora la pendiente de la serie en la observación previa (b_{t-1}). Aunque existen muchas maneras de inicializar este valor al principio, todas ellas se basan en cálculos de medias entre los valores de la serie de tiempo original.

Así pues, el segundo concepto al que se acaba de dar pie es la pendiente. En el análisis de series temporales, la pendiente –típicamente denotada por b –, se entiende como la diferencia entre 2 puntos adyacentes de una serie temporal ($b = y_t - y_{t-1}$), por lo que al enfoque aditivo respecta. En el enfoque multiplicativo, por su parte, se tendría que $b = y_t/y_{t-1}$. En la práctica, el enfoque aditivo es más simple que el multiplicativo y se hará referencia a dicho enfoque para el análisis de nuestra serie temporal, previo a su pronóstico.

Para el cálculo de la pendiente, se usa la siguiente fórmula:

$$b_t = \beta \cdot (l_t - l_{t-1}) + (1 - \beta) \cdot b_{t-1}, \text{ con } 0 \leq \beta \leq 1$$

En esta segunda ecuación, se introduce un nuevo parámetro, el coeficiente de pendiente (β). Como se comentaba para el parámetro α , elegir el mejor valor de β también influye en el resultado de la predicción.

Finalmente, con estas 2 ecuaciones, podemos calcular un nuevo par de puntos futuros de nuestra serie temporal como sigue: $\hat{y}_{t+m} = l_t + m \cdot b_t$.

3. **Triple Exponential Smoothing.** Este último tipo de suavizado exponencial constituye el método *Holt-Winters* tal y como lo conocemos e introduce un último nuevo concepto: la estacionalidad. Para poder usar este método, es necesario que nuestra serie temporal presente un comportamiento estacional, que consiste en la aparición de repeticiones en intervalos regulares de tiempo.

De esta manera, necesitamos una tercera ecuación que compute el efecto estacional de la serie, el cuál también influirá en la predicción final. Así, podemos resumir las ecuaciones usadas en el suavizado exponencial triple de la siguiente manera:

$$l_t = \alpha \cdot (y_t - s_{t-L}) + (1 - \alpha) \cdot (l_{t-1} + b_{t-1}), \text{ para el nivel.}$$

$$b_t = \beta \cdot (l_t - l_{t-1}) + (1 - \beta) \cdot b_{t-1}, \text{ para la pendiente.}$$

$$s_t = \gamma \cdot (y_t - l_t) + (1 - \gamma) \cdot s_{t-L}, \text{ para la estacionalidad.}$$

En la primera de las ecuaciones anteriores, hay que notar el uso de la componente estacional en el cálculo del nivel. Además, se introduce un tercer y último parámetro, el coeficiente estacional γ , del cual se intenta obtener también el mejor valor a fin de minimizar el Error Cuadrático Medio (ECM) y el Error Absoluto Medio (EAM) de los resultados. Por último, y_t hace referencia a la observación actual en un instante t , y la letra L es usada para denotar la longitud estacional.

La fórmula final del método *Holt-Winters* en la que intervienen nivel, pendiente y estacionalidad, y a partir de la cuál podemos predecir tantos valores futuros (\hat{y}_{t+m}) como queramos, sin restricciones en su número, es:

$$\hat{y}_{t+m} = l_t + m \cdot b_t + s_{t-L+m}$$

Respecto a los valores iniciales para la pendiente, existe una fórmula general para calcularlos, que es:

$$b_0 = \frac{1}{L} \cdot \left(\frac{y_{L+1} - y_1}{L} + \frac{y_{L+2} - y_2}{L} + \dots + \frac{y_{L+L} - y_L}{L} \right)$$

Mientras tanto, los valores iniciales referentes a la estacionalidad, requieren de un proceso más elaborado para obtenerlos¹.

¹<http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc435.htm>

ARIMA

Los modelos *ARIMA* (*Autoregressive Integrated Moving Average*) [5, 6] son, además de los métodos exponenciales de suavizado, otra de las maneras más comunes que existen para llevar a cabo el pronóstico de series temporales. La principal diferencia existente entre ellos es que, mientras los métodos de suavizado exponencial se centran en el nivel, la pendiente y la estacionalidad, los modelos *ARIMA* trabajan y se basan en las autocorrelaciones existentes en los datos.

Estos modelos fueron desarrollados y sistematizados por George Box y Gwilym Jenkins en 1976, y es por eso por lo que también se les conoce como modelos *Box-Jenkins*. Actualmente, estos modelos se usan en una gran variedad de aplicaciones, tanto industriales como en distintos campos de investigación.

Típicamente, un modelo *ARIMA* se representa por medio de una 3-tupla de la forma $ARIMA(p, d, q)$, donde p hace referencia al número de términos autorregresivos estacionales, d es el número de diferencias estacionales, y q es el número de términos de medias móviles estacionales. Además, cuando una serie temporal presenta patrones estacionales cada s periodos, o pautas que se repiten en forma de ciclos, entonces el modelo *ARIMA* se representa de la forma $ARIMA(p, d, q)(P, D, Q)_s$, donde (p, d, q) se refiere a la parte regular de la serie, y $(P, D, Q)_s$ se corresponde con la parte estacional. En el presente estudio se trabaja con una serie temporal estacional, como se observará en los siguientes puntos.

Cuando se hace uso de los modelos *ARIMA* para pronosticar las observaciones futuras de una serie temporal, este proceso implica cuatro fases, que son:

1. Como primer paso, se debe determinar la estructura del modelo, o lo que es lo mismo, dar valores a las variables p , d y q , y en el caso de que la serie sea estacional, también dar valores a la correspondiente parte estacional $(P, D, Q)_s$.
2. Posteriormente, se deben estimar los parámetros no conocidos del modelo, de manera que se obtengan sus errores estándar y los residuos del mismo.
3. El penúltimo paso consiste en diagnosticar los residuos estimados, obtenidos del proceso anterior.
4. Finalmente, se pronostican los valores futuros haciendo uso de la información previa y

disponible de la serie temporal, que se usará como dato de entrenamiento. Normalmente, para obtener buenos resultados en nuestros pronósticos, es muy típico disponer de aproximadamente unas 50 o más observaciones en nuestra serie temporal, tal y como se menciona en [12].

Descomposición STL

La descomposición STL (*Seasonal and Trend Decomposition using Loess*) [7], es una forma de alcanzar un mejor grado de comprensión en el comportamiento de una serie temporal estacional, así como de sus cambios a lo largo del tiempo. El principal objetivo es descomponer una serie temporal en las distintas partes que la integran. Estas partes, como ya se ha ido mencionando en el presente trabajo, hacen referencia a diferentes categorías para un patrón en una serie de tiempo, como:

1. **Tendencia**, que se identifica con el suave movimiento de series temporales a largo plazo. Este movimiento suele ser incremental o decremental, aunque también podría ser lineal.
2. **Efecto Estacional**. En ocasiones, las series temporales pueden presentar una cierta periodicidad, entendida como la repetición de un patrón de manera regular en el tiempo. Por ejemplo, el número de personas desempleadas suele ser mayor en invierno que en verano. Este efecto se puede estudiar y entender fácilmente, e incluso ser eliminado por medio una desestacionalización de la serie temporal original.
3. **Componente Aleatoria**. Eliminando la tendencia y el efecto estacional, aún nos quedarían en la serie algunos valores aleatorios. El objetivo es estudiar su comportamiento haciendo uso de algún modelo probabilístico que los describa.

Con todo ello, podemos denotar una serie de tiempo de la manera que sigue:

$$y_t = S_t + T_t + E_t, \text{ para una descomposición aditiva, y}$$

$$y_t = S_t * T_t * E_t, \text{ para una descomposición multiplicativa,}$$

donde y_t es la serie temporal, S_t es la componente estacional, T_t es la tendencia y E_t es el error, ruido o componente aleatoria.

Aunque el proceso de descomposición STL es más completo, la descomposición de una serie de tiempo consta de 5 fases clásicas:

1. Primeramente se estima la tendencia, lo cuál se hace con un procedimiento de suavizado como es el cálculo de las medias móviles.
2. El segundo paso es eliminar la tendencia de la serie. Para el enfoque aditivo, en el cuál nos centramos dada la serie estudiada, esto se hace restando la tendencia estimada de la serie original. La decisión de usar un enfoque aditivo o multiplicativo, radica en comprobar si la serie temporal analizada presenta una varianza constante a lo largo del tiempo o no. Así, si la varianza de la serie es constante se debe usar un enfoque aditivo. En caso contrario, se usará un enfoque multiplicativo.
3. Posteriormente, se estiman los factores estacionales por medio de la serie extraída del paso anterior. Para datos mensuales, esto implica estimar un efecto para cada mes del año. El método más simple para estimar estos efectos es promediar los valores *desfasados* para una temporada específica. Por ejemplo, para obtener un efecto estacional para enero, promediamos los valores *desfasados* para todos los eneros en la serie, y así sucesivamente.
4. Usualmente, los efectos estacionales se ajustan de modo que promedien a 0 para una descomposición aditiva.
5. Finalmente, se extrae la componente aleatoria, restando de la serie original la tendencia y la estacionalidad.

Aún a pesar de que la descomposición STL está principalmente enfocada al estudio de una serie temporal, también se puede hacer uso de ella para pronosticar valores futuros de la misma. Cuando se lleva a cabo el pronóstico por medio de este modelo, la componente estacional (S_t) es predicha de manera aislada al resto de componentes. Así, dado que esta componente no suele cambiar o lo hace de manera muy gradual, es muy típico usar, como método base, el método Naïve estacional². Por su parte, el resto de componentes son predichas haciendo uso de métodos de pronóstico no estacional, como: *Random Walk* con modelo de deriva, el método Holt, modelos *ARIMA* no estacionales, etc.

El software R, mencionado en la sección 3.2, nos proporciona funciones para ajustar el modelo mediante la gestión de algunos parámetros intermedios para, a continuación, pronosticar los valores futuros de la serie temporal analizada mediante la metodología de *Random Walk*³.

²<https://www.otexts.org/fpp/2/3>

³http://cmaskm.ihmc.us/rid=1052458884462_996058812_7176/randomwalk.pdf

3.1.2. Técnicas basadas en Aprendizaje Automático y Profundo

En el presente apartado, se van a introducir las técnicas de aprendizaje automático y profundo que se han usado y que constituyen uno de los pilares fundamentales en los que se sustenta la elaboración de este trabajo. Llegados a este punto, y dada la gran cantidad de técnicas de regresión de las que disponemos hoy en día, hemos querido centrar nuestros esfuerzos en el uso de redes neuronales, desde sus versiones más sencillas, hasta los tipos más empleados en la actualidad en diversos campos de aplicación.

Como sabemos, las redes neuronales se caracterizan por ser un modelo computacional que consiste en un conjunto de unidades de procesamiento interconectadas llamadas neuronas. Así, estas unidades de procesamiento, tratan de imitar de la mejor manera posible el comportamiento del cerebro humano, aprendiendo a hacer una tarea dado un conjunto previo de datos de entrenamiento. Las redes neuronales presentan además algunas características relevantes como la robustez –por lo que resultan menos propensas a fallos–, o la capacidad de manejar diferentes tipos de datos –a pesar de existir ausencias o *gaps* en los mismos–.

En los puntos siguientes, se exponen los tipos de redes neuronales implementadas y usadas en este proyecto.

Redes Neuronales *Feed-Forward*

Las denominadas Redes Neuronales *Feed-Forward* son uno de los modelos más típicos y probados de redes neuronales cuando se llevan a cabo tareas de pronóstico donde se quiere introducir la Inteligencia Artificial [8]. Normalmente, las neuronas de este tipo de redes están dispuestas en 3 partes: la capa que contiene las unidades de entrada, un número arbitrario de capas intermedias u ocultas, y la capa de salida. En esencia, cada neurona está conectada a todas las neuronas de la capa siguiente y a ninguna de la misma capa. Además, las neuronas de la capa de entrada a la red, sólo suelen pasar un determinado vector de entrada x . Mientras tanto, las neuronas de las capas ocultas y de la capa de salida poseen lo que se conoce como función de activación, que en este tipo de redes típicamente se corresponde con la función sigmoide –aunque admite otras distintas–, detallada a continuación:

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

Por otra parte, las FFNN se basan en el algoritmo de entrenamiento de Propagación hacia Atrás (*Back-Propagation*), que es esencialmente un algoritmo de descenso de gradiente que engloba 5 fases:

1. Como primer paso, la red tiene que ser construida e inicializada. Para ello, se decide el número de unidades de procesamiento o neuronas que incorporará cada una de las capas de dicha red. Una vez se tiene el esqueleto, se seleccionan los pesos iniciales aleatorios y un determinado criterio de parada, que se corresponde con el número de épocas con el que se va a llevar a cabo el entrenamiento.
2. El segundo paso es el de *alimentación* directa de la red como tal. Este proceso, consiste en computar la salida para todas y cada una de las unidades que no sean de entrada, o lo que es lo mismo, para todas las neuronas de la capa oculta. Al mismo tiempo, se calcula el error de cada una de esas unidades y su contribución al error global alcanzado. Si llegados a este punto se alcanza el criterio de parada, el algoritmo termina. Si no, se continúa al paso 3.
3. Una vez que los errores han sido calculados, estos se propagan hacia atrás en la red para conseguir una mejora de la misma. Así, las neuronas que más hayan contribuido al error global, tendrán que ajustar mejor sus pesos, mientras que las neuronas que menos hayan contribuido a dicho error, mantienen los pesos que tenían.
4. Se lleva a cabo la actualización de los pesos de las neuronas en función de lo comentado en el punto anterior.
5. Finalmente, el algoritmo vuelve al paso 2, donde se volverá a comprobar el criterio de parada.

Además, suele ser típico el uso de *momentum* en estas redes neuronales para mejorar la tasa de aprendizaje y acelerar el proceso que se acaba de comentar, dado que la convergencia en estas redes suele ser lenta y en muchas ocasiones termina cayendo en mínimos locales [13].

Este tipo de redes, al igual que ocurrirá con las redes explicadas a continuación, serán implementadas haciendo uso del *framework Keras*, que nos proporciona funciones para configurar las características de la red neuronal y realizar los oportunos pronósticos.

Redes Neuronales Recurrentes con LSTM

Las Redes Neuronales Recurrentes [9, 14] son redes neuronales pensadas para el procesamiento secuencial de datos o valores. Esto resulta especialmente útil a la hora de trabajar con series temporales, ya que disponemos de valores cronológicamente ordenados en forma de array unidimensional. Además, al igual que las redes convolucionales –explicadas en el siguiente epígrafe–, que pueden escalar fácilmente a imágenes de gran tamaño y algunas pueden incluso procesar imágenes de tamaño variable, las redes recurrentes pueden escalar a secuencias mucho más largas cuando disponemos de un gran número de datos a analizar.

Para hacer posible este tipo de redes, partimos de una de las ideas básicas del aprendizaje automático y los modelos estadísticos: la compartición de parámetros a través de las diferentes partes de un modelo. Esta compartición nos permite generalizar dicho modelo para poder usarlo con muestras de formas distintas, sin atender a la longitud de una serie temporal concreta. Además, este tipo de redes extienden la idea de un grafo computacional cíclico, donde los ciclos representan la influencia del valor presente de una variable en su propio valor futuro.

Partiendo de estas ideas, podemos diseñar un amplio abanico de Redes Neuronales Recurrentes. Algunos de los patrones de diseño más relevantes son los que se detallan a continuación:

1. RNN que produce una salida por cada paso de tiempo, y dispone de conexiones recurrentes entre las unidades ocultas (figura 3.1(a)). Suele ser la estructura más típicamente usada.
2. RNN que produce una salida por cada paso de tiempo, y dispone de conexiones recurrentes sólo desde la salida en un paso de tiempo hasta las unidades ocultas del paso siguiente (figura 3.1(b)).
3. RNN con conexiones recurrentes entre unidades ocultas, capaces de leer una secuencia entera de valores y generar una única salida a partir de ello (figura 3.1(c)).

En los diseños mostrados en la figura 3.1 [9], se tiene que: x hace referencia a los valores de una secuencia de entrada; o son los valores de la secuencia de salida; L mide la diferencia entre el valor de salida obtenido y el valor de entrenamiento objetivo y ; y h hace referencia a las neuronas de la capa oculta.

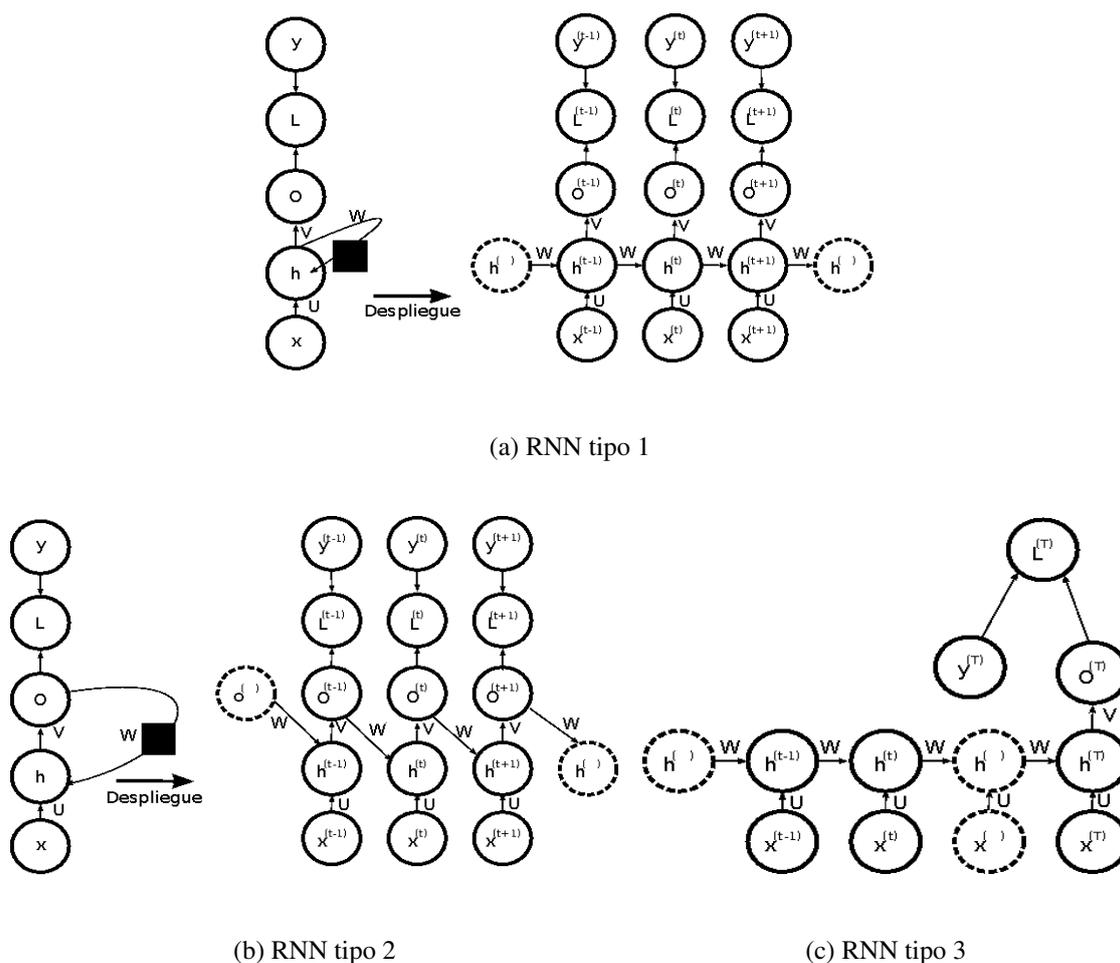


Figura 3.1: Estructuras más relevantes de Redes Neuronales Recurrentes.

Por su parte, las conexiones entre la entrada y la capa oculta están parametrizadas por una matriz ponderada U , las conexiones recurrentes entre los nodos de la capa oculta están parametrizadas por una matriz ponderada W , y las conexiones entre la capa oculta y la capa de salida están parametrizadas por una matriz ponderada V .

Aunque la mayor parte de las Redes Neuronales Recurrentes tienen estructura causal —entendiendo por causal que son capaces de capturar en un instante t , información del pasado y de la entrada en ese instante—, en ocasiones tenemos que lidiar con aplicaciones en las que queremos generar una determinada predicción que depende de toda una secuencia completa de entrada. Por ejemplo, esto es particularmente cierto cuando trabajamos en reconocimiento de voz, en el que la interpretación correcta de un sonido como un fonema, puede depender de los siguientes fonemas e incluso de las siguientes palabras debido a las dependencias lingüísticas entre palabras cercanas. De esta manera, es posible que tengamos que mirar hacia el futuro (y

el pasado) para llegar a un reconocimiento más preciso. Aunque en nuestro caso de estudio vamos a estar interesados en información pasada para poder hacer los pronósticos del ^{222}Rn , es relevante mencionar que también existen Redes Neuronales Recurrentes Bidireccionales que cubren las necesidades mencionadas [15].

Finalmente, cuando se trabaja con Redes Neuronales Recurrentes en aplicaciones prácticas, es muy común el uso de las denominadas Redes con Puertas –*Gated RNNs*–, que incluyen a las LSTM y a otras unidades recurrentes con puertas denominadas *Gated Recurrent Units* (GRU). La idea raíz de este tipo de redes, es almacenar información relevante durante un largo periodo de tiempo. Sin embargo, una vez que la información ha sido usada, queremos que sea la propia red la que decida cuándo liberar dicha información en lugar de hacerlo nosotros manualmente. En definitiva, estamos hablando de redes neuronales con memoria capaces de autogestionarse, olvidando el estado antiguo de la red para adquirir y almacenar nueva información. Así, los dos tipos más representativos de *Gated RNNs* son:

1. ***Long Short-Term Memory***. Introducen autobuses en la capa oculta, creando caminos donde la información puede fluir a lo largo de un gran periodo de tiempo [16]. Además, una de las contribuciones más relevantes a este tipo de unidades, ha sido el hacer que el peso en este autobús dependa del contexto en lugar de fijarlo [17]. De esta manera, se alcanza el control dinámico de estas unidades.
2. ***Gated Recurrent Units***. El funcionamiento de estas unidades es exactamente el mismo que el de las LSTM, con la diferencia de que son una simplificación de estas últimas. Así, la principal diferencia entre ambas, es que las GRU tienen una sola unidad que se encarga al mismo tiempo de controlar los pesos y de tomar la decisión de si actualizar o no el estado de la unidad. Una visión más amplia de estas redes se puede encontrar en [18, 19, 20].

Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales [9, 21] constituyen un tipo especial de redes neuronales principalmente destinadas al procesamiento de datos que poseen una estructura bidimensional (cuadrícula), aunque también para datos unidimensionales. Así, resultan bastante útiles cuando se trabaja con series temporales –interpretadas como una cuadrícula

unidimensional con muestras en intervalos de tiempo regulares–, o con imágenes –cuadrículas bidimensionales de píxeles–. El nombre de estas redes se debe a que, en lugar de llevar a cabo multiplicación general de matrices en todas las capas de la red, emplean un tipo especializado de operación lineal, denominada convolución, en al menos una de las capas.

Existen tres grandes ideas en la convolución que resultan de gran ayuda en sistemas que hacen uso de técnicas de aprendizaje profundo:

1. **Las interacciones dispersas.** Cuando trabajamos con la operación de convolución en aplicaciones de aprendizaje profundo, disponemos de una entrada –típicamente representada por un array multidimensional de datos–, y un kernel –un array multidimensional de parámetros que son adaptados por medio del algoritmo de aprendizaje–. Estos arrays, denominados tensores, no suelen ser del mismo tamaño, reduciendo así el número de interacciones entre unidades al procesarlos. Aunque en las redes neuronales tradicionales cada unidad de salida interactúa con cada unidad de entrada por medio de un producto de matrices, esto no es así en las CNN. Por ejemplo, cuando procesamos una imagen, podemos tener miles o millones de píxeles y, sin embargo, detectar un pequeño número de características de la misma que sólo ocupen cientos de píxeles o menos. Esto hace que se requieran muchos menos parámetros y un menor número de interacciones, reduciendo la cantidad de memoria usada y mejorando la eficiencia estadística.
2. **El uso compartido de parámetros.** La compartición de parámetros, por su parte, hace referencia al uso de un mismo parámetro para más de una función en un determinado modelo. En las redes neuronales tradicionales, cada elemento de la matriz de ponderación se usa exactamente una vez cuando se calcula la salida de una capa, y no se vuelve a visitar. En las CNN, en lugar de aprender un conjunto separado de parámetros para cada ubicación, aprendemos solo un conjunto general.
3. **Las representaciones equivalentes.** En el caso de la convolución, la forma existente de compartir parámetros da lugar a una propiedad denominada equivalencia. Decir que una función es equivalente significa que si la entrada cambia, la salida lo hace de la misma manera. De esta forma, tanto en el procesamiento de series temporales como en el de

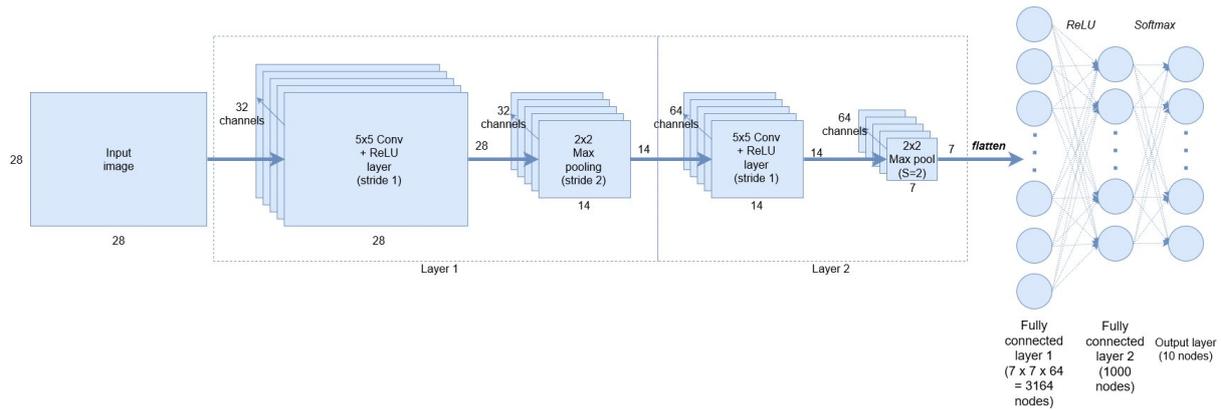


Figura 3.2: Ejemplo de estructura de una CNN con dos capas de convolución para el procesamiento de una imagen.

imágenes, el hecho de mover un determinado evento en la entrada, se traducirá en la aparición de dicho evento en la salida justo después.

Como se puede observar en la figura 3.2⁴, una capa de convolución típica consta de 2 grandes etapas. En la primera etapa, se llevan a cabo en paralelo las convoluciones propiamente dichas, que dan lugar a un conjunto de activaciones lineales que son a su vez ejecutadas por medio de una función de activación no lineal, típicamente la función de activación rectificadora (`relu`) en las CNN. Por su parte, en la segunda etapa, se usa una función de agrupación (`Pooling`) para modificar la salida obtenida. Esta función, reemplaza la salida de la red en una posición concreta, con una estadística resumen de las salidas cercanas. Por ejemplo, la operación de agrupación máxima (también de uso típico en las CNN), informa de la máxima salida en una vecindad rectangular. Sea cual sea la agrupación usada, esta función hace que las representaciones finales de nuestros datos sean invariantes a pequeños cambios en los datos de entrada, como se puede observar en la figura 3.3 [9]. Por último, antes de generar la salida de la red, se suelen aplanar los datos con el objetivo de crear una estructura de red neuronal propiamente dicha, con todos los nodos completamente conectados entre sí.

Finalmente, es importante destacar que los datos que se usan en una red convolucional se componen típicamente de múltiples canales, cada uno de ellos correspondientes a una observación distinta en algún punto del espacio o del tiempo. Así, una de las grandes ventajas de estas redes frente a las tradicionales, es que pueden procesar entradas con diferentes

⁴<http://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/>

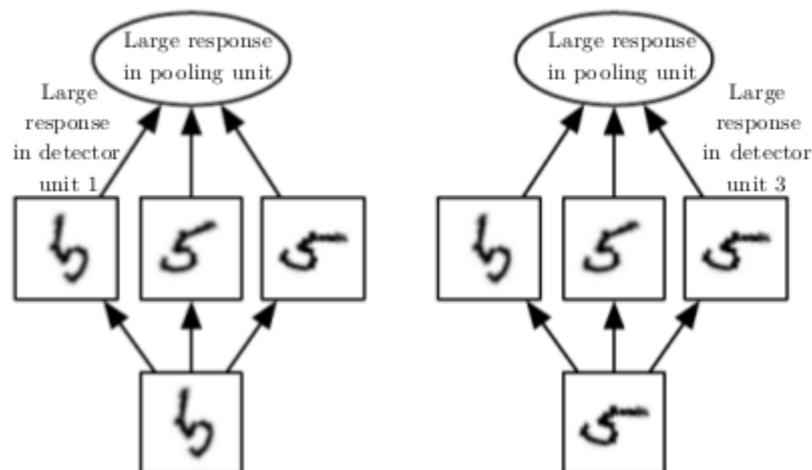


Figura 3.3: Ejemplo de unidad de *pooling* aprendiendo a ser invariante ante las transformaciones de la entrada.

dimensiones espaciales. Por ejemplo, si tenemos que procesar una colección de imágenes con distintas dimensiones haciendo uso de redes tradicionales, resulta complicado saber cómo modelar dichas entradas con una matriz ponderada de tamaño fijo. Sin embargo, en las redes convolucionales, el kernel se aplica un determinado número de veces dependiente del tamaño de la entrada, y la salida de la operación de convolución se escala en consecuencia.

3.2. Lenguajes y *Frameworks*

Como ya se ha venido comentando en los anteriores capítulos, los lenguajes en los que nos centraremos en el presente estudio serán R y Python.

R⁵ es un lenguaje ampliamente usado hoy en día para computación estadística y representación gráfica de calidad para diversos resultados. Proporciona un amplio número de librerías y paquetes para modelado, test estadísticos, análisis de series temporales, clasificación, regresión, *clustering*, entre otros. Además, se trata de una herramienta altamente extensible de código abierto, por lo que se puede usar libremente y contribuir a su crecimiento y mejora. Puede ser ejecutado en distintas plataformas UNIX y similares, así como en Windows y MacOS. Se usará para trabajar con las técnicas clásicas explicadas en el apartado 3.1.1.

⁵<https://www.r-project.org>

Por su parte, Python⁶ es un potente y claro lenguaje de programación orientado a objetos, comparable a otros como Perl, Ruby o Java. Se caracteriza por disponer de una sintaxis bastante fácil de comprender, un gran número de librerías para trabajar en diversos campos de estudio, y también llamadas y funciones para representar gráficamente los resultados. Además, al ser orientado a objetos, presenta clases y herencia múltiple, y posee una gran variedad de tipos que son asignados dinámicamente. Al igual que R, es un lenguaje de código abierto que puede ser modificado libremente y redistribuido bajo un determinado tipo de licencia de código abierto. Finalmente, puede ser ejecutado en UNIX, Linux, Windows y MacOS.

Por otro lado, el *framework* usado en Python para el desarrollo e implementación de las Redes Neuronales *Feed-Forward*, las Redes Neuronales Recurrentes y las Redes Neuronales Convolucionales –explicadas en el apartado 3.1.2–, será *Keras*.

Keras, es un API (*Application Programming Interface*) escrito a alto nivel para trabajar con redes neuronales. Es capaz de ejecutarse sobre *backends* como *TensorFlow*, *CNTK*⁷ o *Theano*⁸. *Keras* nos permite un prototipado rápido y sencillo, pudiendo trabajar con las redes neuronales más típicas hoy en día y ejecutarlo sin problemas tanto en la CPU como en la GPU. Entre sus características principales, caben destacar: es *user-friendly*, por lo que facilita la programación de las redes haciendo uso de un lenguaje más cercano al ser humano y más alejado de las máquinas; la modularidad, que nos permite trabajar con las componentes de una red neuronal de manera separada, para luego combinarlas y crear distintos modelos; la extensibilidad, que nos permite añadir nuevos módulos y ampliar los ya existentes, contribuyendo a la investigación avanzada; y trabaja con Python, lo que nos permite no tener que gestionar distintos archivos, consiguiendo un código compacto y fácil de depurar en caso de error.

⁶<https://www.python.org>

⁷<https://www.microsoft.com/en-us/cognitive-toolkit/>

⁸<http://www.deeplearning.net/software/theano/>

Capítulo 4

Configuración y preprocesamiento de datos

Una vez introducidas las técnicas de las que vamos a hacer uso para modelar y pronosticar la serie temporal analizada en el presente caso de estudio, es necesario introducir cómo se han manipulado y preprocesado los datos a fin de detectar peculiaridades en los mismos, tener una visión general acerca de cómo son y prepararlos para su uso en las metodologías previamente explicadas.

Como ya se ha comentado, en este trabajo, disponemos de un conjunto de medidas de ^{222}Rn tomadas entre Julio de 2013 y Junio de 2017, haciendo un total de cuatro años completos. Cada día se compone de 144 observaciones y, por lo tanto, al final de la semana se recogen un total de 1008 medidas, considerando que no haya observaciones ausentes o perdidas. La razón de que existan estos números de observaciones es que el dispositivo que las recoge, mide la concentración de ^{222}Rn cada 10 minutos. Este dispositivo, localizado a 850 metros de profundidad en el Hall A del LSC (Aragón, España), es un Alphaguard P30¹ que registra los valores tomados en un ordenador portátil que hace uso de *Data Expert*², un software muy simple que monitoriza y muestra gráficamente los niveles de ^{222}Rn . Aparte de ello, el Alphaguard P30 también es capaz de medir y registrar al mismo tiempo la temperatura en el medio, la presión del aire y la humedad relativa.

Con todo ello, en primer lugar, en la figura 4.1, se representan gráficamente los datos en

¹http://www.saphymo.com/ftp/ecatalogue/154/29294519EN_C_-_Data_Sheet_ALPHAGUARD.pdf

²<https://www.bertin-instruments.com/product/environmental-radiation-monitoring-systems/>

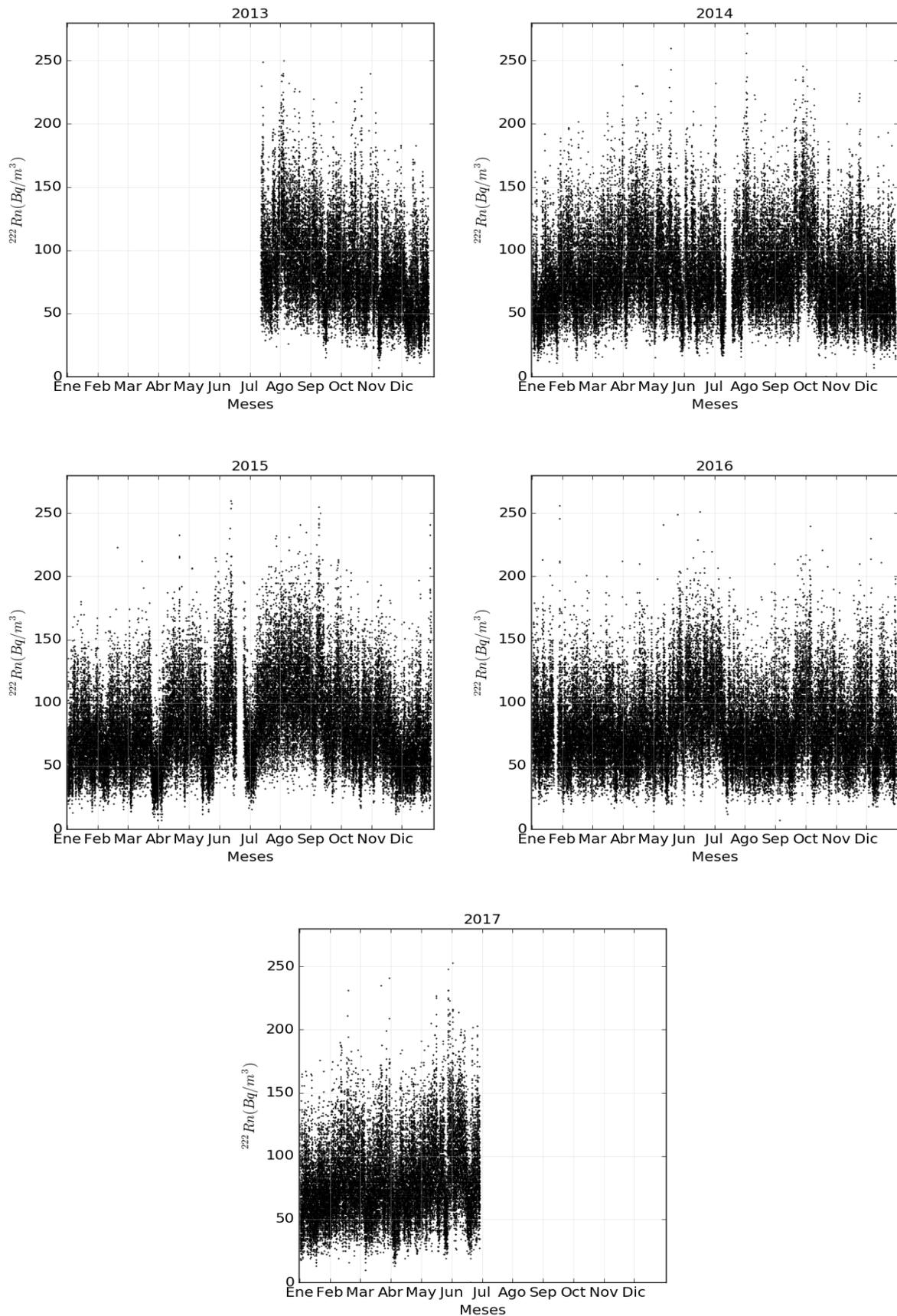


Figura 4.1: Datos en bruto: valores de ^{222}Rn en Bq/m^3 en el Hall A del LSC.

bruto para cada uno de los años analizados. En esta figura, como se puede observar, existe un elevado volumen de datos dónde se pueden apreciar algunos *gaps* y un alto nivel de ruido, que apenas nos deja visualizar posibles patrones en los mismos.

Una vez mostrados los datos en bruto de las concentraciones de ^{222}Rn , pasamos a detectar las posibles anomalías presentes en nuestra serie temporal, centrándonos en las siguientes:

1. En primer lugar, tenemos las medidas negativas de concentración de ^{222}Rn . Cualquier valor negativo de este gas debe ser descartado, dado que se trata de una medida falsa. El motivo de esto es que, dado que el Alphaguard P30 toma concentraciones de ^{222}Rn comprendidas entre los 2 Bq/m^3 y los $2.000.000 \text{ Bq/m}^3$, es imposible tener valores negativos de este gas.
2. El segundo tipo de anomalías, es la ausencia de algunos valores en la serie temporal. Por una parte, se pueden encontrar valores aislados perdidos y, por otra, *gaps* más amplios. En cualquier caso, para nuestra serie temporal, se han rellenado los huecos ausentes con valores a -1, que luego serán ignorados en la ejecución de nuestros algoritmos.

En cuanto a los grandes *gaps*, en la tabla 4.1, se especifican sus fechas de comienzo y el número de observaciones consecutivas perdidas desde entonces. En nuestro estudio, decidimos ignorar dichos *gaps* dado que los números de observaciones perdidas no son representativos en comparación a la cantidad de datos disponibles para cada año.

3. El tercer y último tipo de anomalías estudiadas son los valores *outlier* o atípicos. En este trabajo, hemos tratado detectar dichos valores por medio de un simple cálculo que consiste en determinar si las medidas de ^{222}Rn se encuentran, o no, en el rango comprendido entre el $\text{cuartil}^{25} - 1,5 * IQR$ y el $\text{cuartil}^{75} + 1,5 * IQR$. Los cuartiles 25 y 75 son dos de los tres valores que dividen un conjunto de datos ordenados en cuatro partes

Comienzo del <i>gap</i>	Observaciones consecutivas perdidas
14/07/2014 12:10:00	913
19/06/2015 14:30:00	1.053
22/01/2016 15:50:00	585

Cuadro 4.1: Periodos de ausencia de datos más largos.

porcentualmente iguales, cada una conteniendo, por tanto, un cuarto de la información. Por otro lado, el *IQR* es el Rango Inter Cuartílico, que se entiende como la diferencia entre el *cuartil*⁷⁵ y el *cuartil*²⁵. Finalmente, 1.5 es el valor más típicamente usado en este tipo de detección de anomalías³.

4.1. Agrupamiento de la información y correlación con el Hall B del LSC

Dada la gran cantidad de información disponible (más de 200.000 observaciones en total), la presencia de ruido elevado en nuestra serie temporal, y con objeto de reducir y hacer asequibles los tiempos de procesamiento de los algoritmos implementados, decidimos agrupar los valores de ²²²Rn por meses y calcular sus correspondientes medianas. La decisión del porqué usar las medianas y no las medias, radica en que las medianas son más robustas. Entendemos como robusto, el hecho de que una mediana no altera el valor representativo de ²²²Rn para un mes, en el caso de que haya alguna observación *outlier* en el mismo. Por su parte, las medias mensuales sí se verían afectadas por este hecho, llevándonos a valores más alejados de la realidad de un valor representativo para un mes y, por tanto, conduciéndonos a un estudio poco real de nuestra serie temporal. De este modo, basamos nuestro trabajo en torno a la serie temporal de medianas mensuales de ²²²Rn, que nos da un total de 48 observaciones para los meses comprendidos entre Julio de 2013 y Junio de 2017.

Con el fin de clarificar esto, en la figura 4.2, se muestran las medianas (líneas rojas) de nuestra serie temporal de ²²²Rn en el Hall A para cada uno de los meses comentados. En esta gráfica se puede apreciar una cierta modulación anual en la serie que se corresponde con un incremento de la concentración de Radón en los meses de verano y un decremento en los meses de invierno. Además, para mejorar el indicio visual de la existencia de esa modulación a partir de los niveles en bruto de ²²²Rn, en la figura 4.3 se han agrupado dichos valores por meses, observando el crecimiento en los meses de verano y el decrecimiento en los meses de invierno.

Por otro lado, y como manera de garantizar que los datos del Hall A son representativos y los podemos usar con seguridad en nuestro estudio y en nuestros algoritmos, el mismo preprocesado

³https://en.wikipedia.org/wiki/Box_plot

4.1. AGRUPAMIENTO DE LA INFORMACIÓN Y CORRELACIÓN CON EL HALL B DEL LSC31

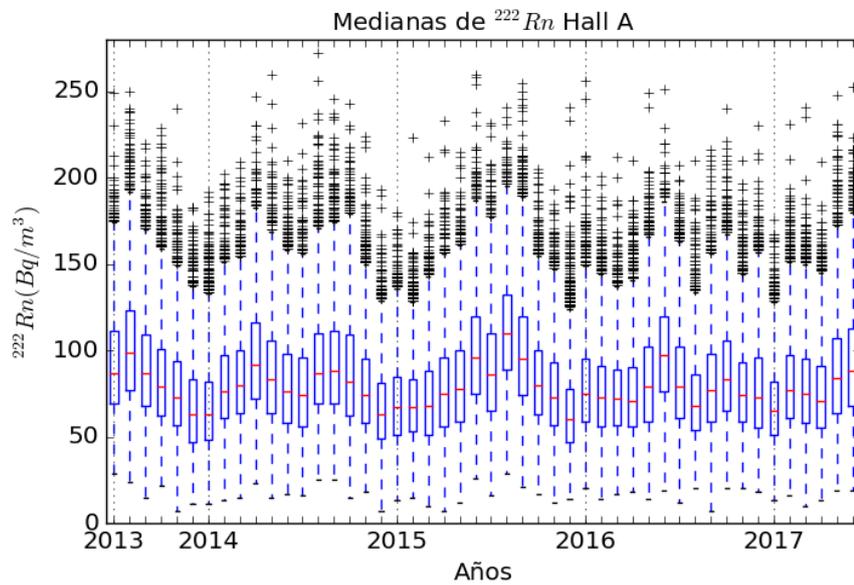


Figura 4.2: Boxplots de las medianas mensuales del ^{222}Rn en el Hall A del LSC.

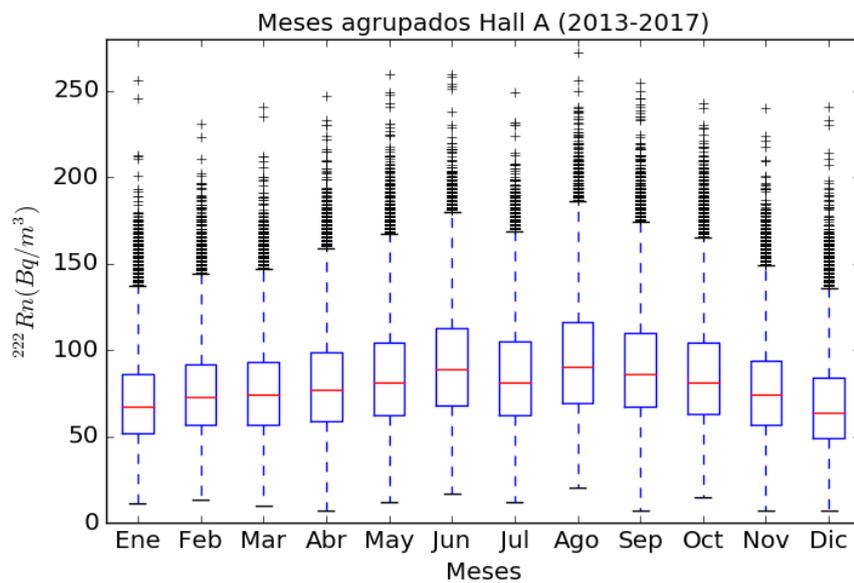


Figura 4.3: Niveles de ^{222}Rn en el Hall A del LSC agrupados por meses.

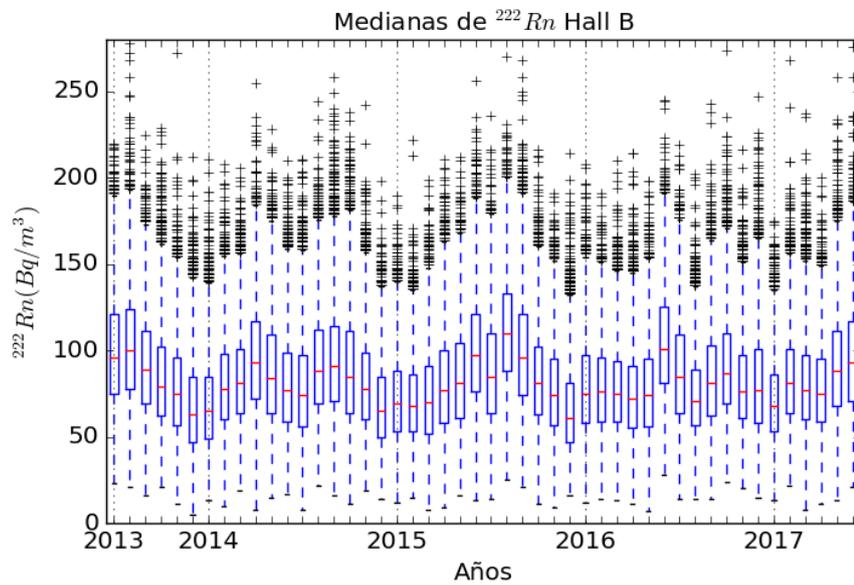


Figura 4.4: Boxplots de las medianas mensuales del ^{222}Rn en el Hall B del LSC.

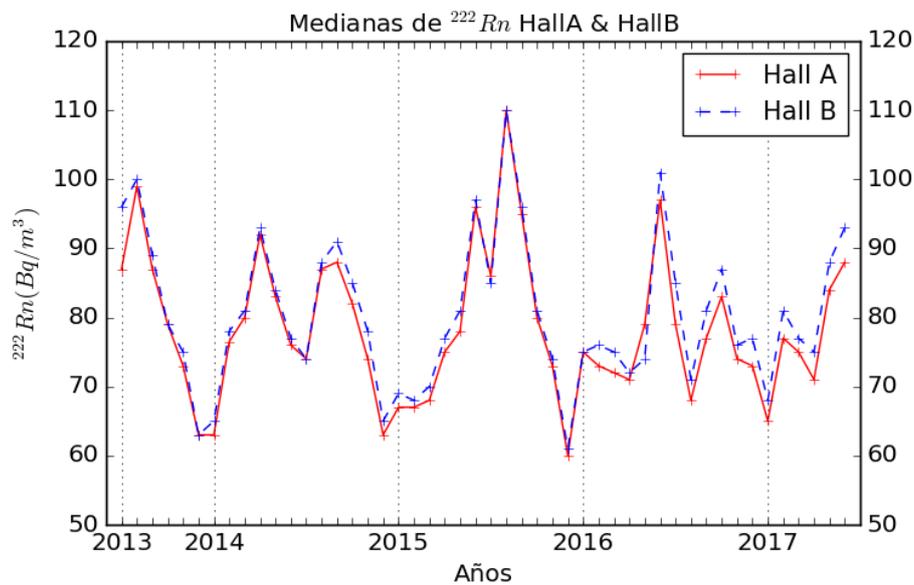


Figura 4.5: Medianas mensuales del ^{222}Rn en el Hall A y en el Hall B del LSC.

4.1. AGRUPAMIENTO DE LA INFORMACIÓN Y CORRELACIÓN CON EL HALL B DEL LSC33

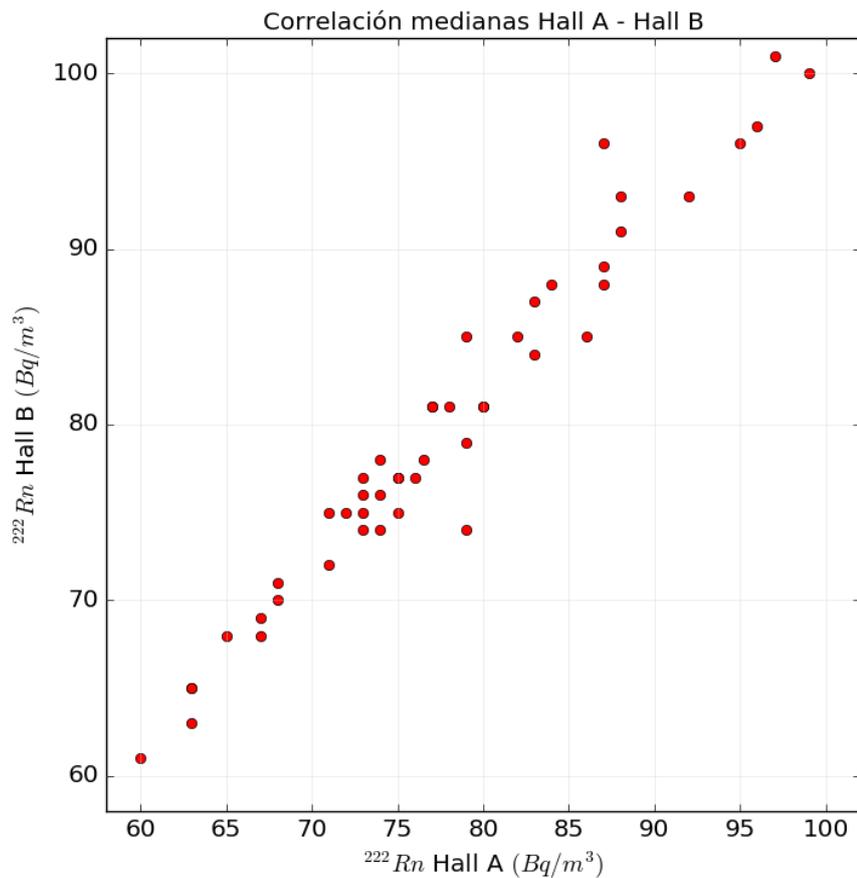


Figura 4.6: Correlación Hall A - Hall B de las medianas mensuales de ^{222}Rn .

de la información se ha llevado a cabo con los datos referentes al Hall B del LSC. En la figura 4.4, se muestran también las medianas mensuales de los niveles de ^{222}Rn del Hall B. Como se puede observar, la modulación es muy parecida (figura 4.5), pero para asegurarnos empíricamente de que ambas series son semejantes (pues se tratan de salas contiguas en el LSC), se ha comprobado la correlación entre ellas. Así, el coeficiente de correlación de Pearson extraído al comprobar ambas series es de 0.9808. Para saber si aceptar o rechazar la hipótesis nula de que las dos series no están correlacionadas, nos fijamos ahora en el p-value, obteniendo un valor de $2,4107 \cdot 10^{-34}$ [22, 23]. Este valor, nos permite rechazar la hipótesis nula planteada e indica que la pendiente del modelo lineal es significativamente no-nula para un nivel de confianza del 95 % (p-value menor de 0.05), como se puede comprobar en la figura 4.6. De este modo, ambas series están muy fuertemente correlacionadas y, por tanto, vamos a trabajar y a centrarnos en la serie temporal de las medianas de ^{222}Rn del Hall A del LSC.

4.2. Correlaciones con las variables climatológicas internas

Una vez que se han establecido los datos de las medianas de ^{222}Rn del Hall A del LSC como base para el estudio realizado en el presente trabajo, y sabiendo que el dispositivo Alphaguard P30 mide también los niveles de temperatura, presión del aire y humedad relativa dentro del Hall, en el presente apartado estamos interesados en determinar si existen posibles correlaciones entre los niveles de Radón registrados, y las variables climatológicas internas comentadas. A priori, no se esperan fuertes correlaciones entre estos datos debido a que las salas del LSC están perfectamente controladas y habilitadas para que factores externos e internos no alteren los resultados obtenidos en los experimentos llevados a cabo allí.

De esta manera, como podemos observar en las figuras 4.7 y 4.8, no existe correlación entre el ^{222}Rn y la temperatura, ni tampoco entre este gas y la presión del aire. Sin embargo, tal y como se observa en la figura 4.9, sí existe una cierta correlación entre el ^{222}Rn y la humedad, apreciando un incremento en los niveles del gas cuando el porcentaje de humedad relativa es alto.

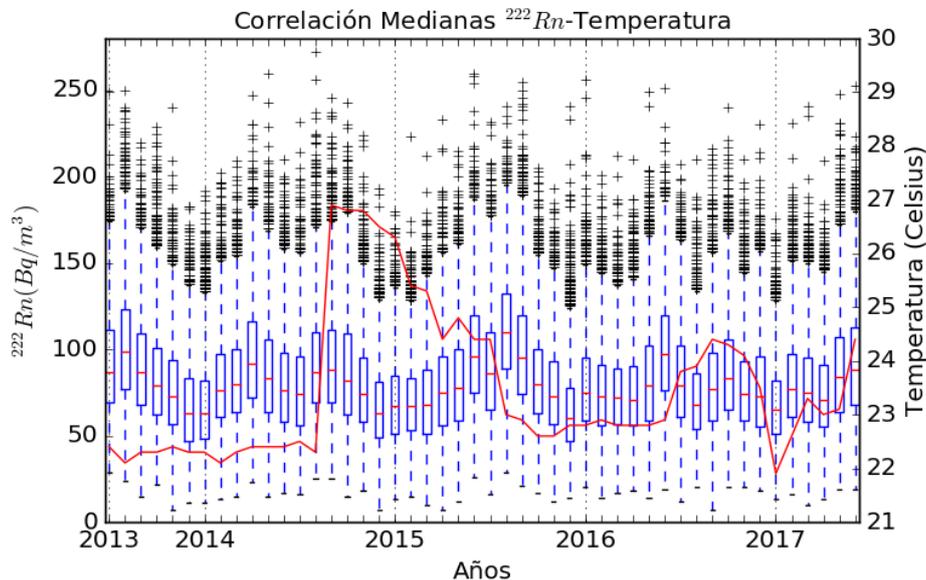


Figura 4.7: Correlación de las medianas mensuales del ^{222}Rn con las de la temperatura.

Como ocurría en el apartado anterior, a fin de demostrar numéricamente los resultados obtenidos en las figuras comentadas, se han calculado los coeficientes de correlación de Pearson para todas y cada una de las correlaciones estudiadas. Así, los coeficientes de correlación para

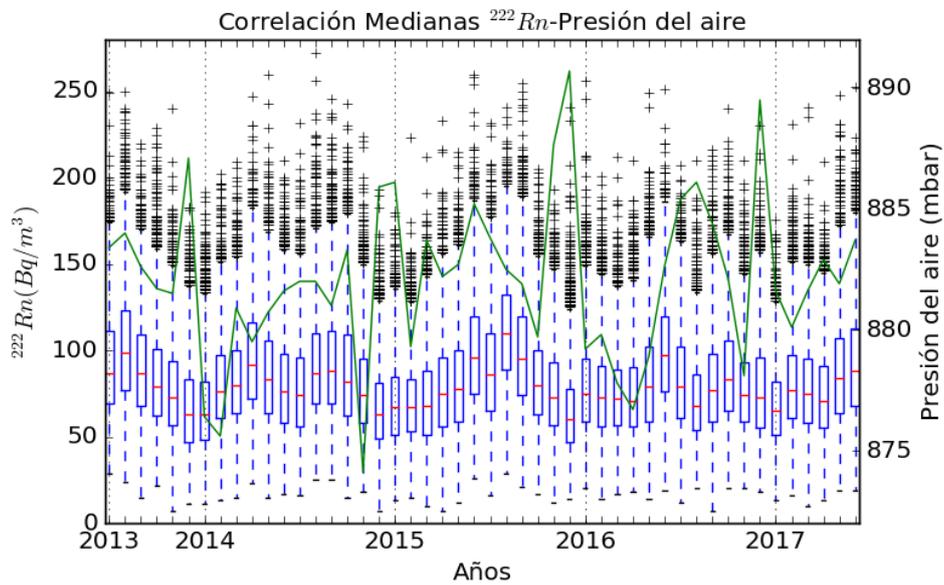


Figura 4.8: Correlación de las medianas mensuales del ^{222}Rn con las de la presión del aire.

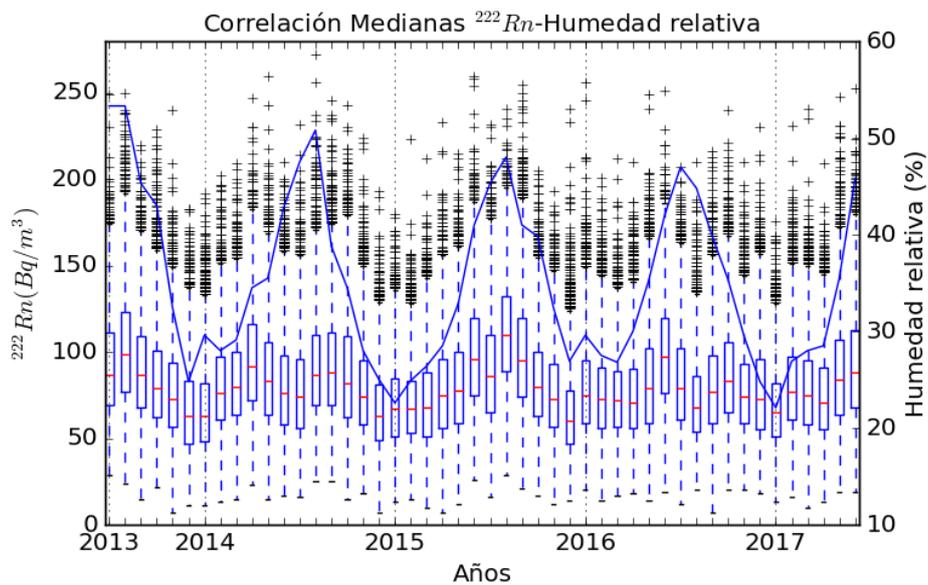


Figura 4.9: Correlación de las medianas mensuales del ^{222}Rn con las de la humedad relativa.

el caso de la temperatura y la presión del aire, son -0.1080 y -0.0516, respectivamente. Estos valores indican que no existe correlación alguna.

Una vez que se ha establecido la relación entre las variables, vamos a ver que sucede con la hipótesis nula: en los dos primeros casos comentados, las variables analizadas no están relacionadas entre sí, y por lo tanto la pendiente del modelo lineal es nula [22, 23]. En estos casos, el p-value para la correlación con la temperatura es de 0.4648, y para la correlación con la presión del aire es de 0.7274. Estos valores tan altos, nos impiden rechazar la hipótesis nula, llevándonos a concluir que tanto la correlación entre el ^{222}Rn y la temperatura, como la correlación entre el ^{222}Rn y la presión del aire, no puede ser significativamente establecida con un nivel de confianza del 95 % (p-value por debajo de 0.05)⁴.

Centrándonos ahora en la correlación entre el ^{222}Rn y la humedad relativa, la figura 4.9 si exhibe una correlación entre estas variables. En este caso, el coeficiente de correlación de Pearson es de 0.7013 y el p-value de $2,8261 \cdot 10^{-8}$. Este último valor nos permite rechazar la hipótesis nula, e indica que la pendiente del modelo lineal es significativamente no nula para un nivel de confianza del 95 %.

Como consecuencia de todo ello, podría decirse que la humedad es un predictor significativo del ^{222}Rn .

⁴En el presente estudio, usamos un nivel de confianza del 95 % (p-value menor de 0.05). Esto significa que es improbable que las diferencias en los resultados hayan ocurrido por casualidad con una probabilidad del 95 %.

Capítulo 5

Resultados

En el presente capítulo se presentan los resultados experimentales del trabajo realizado. El objetivo es mostrar los resultados de los pronósticos realizados con todos y cada uno de los modelos expuestos, y realizar una comparativa global entre ellos que nos permita saber cómo de buenas son las técnicas basadas en aprendizaje automático y profundo en comparación con los modelos clásicos ya explorados y muy explotados en la actualidad.

De esta manera, con el objetivo de conocer el mejor método para pronósticar los valores futuros del ^{222}Rn , los datos recogidos y previamente preprocesados han sido divididos en 2 grandes grupos:

1. El conjunto de datos de entrenamiento, que se corresponde con el 75 % del total de datos disponibles. Por lo tanto, los datos de entrenamiento están compuestos por todas las medianas mensuales de ^{222}Rn comprendidas entre Julio de 2013 y Junio de 2016 (3 años).
2. Por otro lado, el conjunto de datos de evaluación está formado por el 25 % de información restante, es decir, las medianas mensuales de ^{222}Rn comprendidas entre los meses de Julio de 2016 y Junio de 2017 (1 año). Estos datos nos permitirán evaluar la efectividad final de los modelos analizados haciendo uso del Error Cuadrático Medio y del Error Absoluto Medio.

5.1. Métricas de evaluación de rendimiento

Como se acaba de comentar, necesitamos introducir algunas métricas que nos permitan evaluar la precisión y el rendimiento de las metodologías de pronóstico explicadas en la sección 3.1.

En el presente estudio, el Error Cuadrático Medio (ECM) y el Error Absoluto Medio (EAM) han sido seleccionados para evaluar dichos rendimientos. La fórmulas empleadas se detallan a continuación:

1. Para el ECM tenemos:

$$\text{ECM} = \frac{1}{N} * \sum_{i=1}^N (\hat{y}_i - y_i)^2,$$

donde y_i denota el i -ésimo valor de la serie temporal observada; \hat{y}_i denota un valor en la misma posición, pero haciendo referencia a la serie temporal pronosticada; y, finalmente, N es el número de observaciones a pronosticar.

2. Para el EAM tenemos:

$$\text{EAM} = \frac{1}{N} * \sum_{i=1}^N |y_i - x_i|,$$

donde x_i denota el i -ésimo valor de la serie temporal observada; y_i denota un valor en la misma posición, pero haciendo referencia a la serie temporal pronosticada; y, finalmente, N es el número de observaciones a pronosticar.

Se puede concluir que unos valores más bajos del ECM y del EAM, nos indican que la serie temporal observada es más similar a la serie temporal pronosticada y, por lo tanto, que los resultados del pronóstico obtenido haciendo uso del modelo evaluado por estas métricas, son mejores.

No obstante, en nuestro caso de estudio en concreto, la medida del ECM es más representativa que la extraída por medio del EAM. Esto es así, porque estamos interesados en penalizar más los errores de aquellos modelos cuya capacidad o calidad predictiva sea más baja.

5.2. Resultados experimentales por técnicas

A continuación, se muestran los resultados extraídos de la evaluación y pronóstico de la serie temporal del ^{222}Rn haciendo uso de las metodologías comentadas a lo largo del presente proyecto.

Como ya fue comentado en la sección 3.2, los resultados referentes a las técnicas clásicas de pronóstico han sido extraídos haciendo uso del lenguaje R. Por su parte, los resultados propios de las técnicas de aprendizaje automático y profundo, han sido calculados diseñando la estructura de las redes neuronales en Python y haciendo uso del *framework Keras*.

Para cada una de las técnicas estudiadas, y siguiendo las pautas mencionadas en 5, se modela la serie temporal del Radón, se extrae una comparativa de pronóstico con la serie temporal original y se calculan el ECM y el EAM con el objetivo de hacer posible la posterior comparativa general de dichas técnicas.

5.2.1. Resultados experimentales con *Holt-Winters*

Como ya se explicó en el apartado 3.1.1, disponemos de 3 métodos de suavizado exponencial: el simple, el doble y el triple, este último también conocido como *Holt-Winters* propiamente dicho.

Así, para el cálculo de este primer resultado, se hace uso del método de suavizado exponencial triple. Para ello, empezamos separando las 48 medianas de ^{222}Rn en un conjunto de datos de entrenamiento y en un conjunto de datos de evaluación, tal y como se comentó al inicio del capítulo 5.

Una vez tenemos el conjunto de datos de entrenamiento creado, se le pasa como único argumento a la función `HoltWinters` que nos proporciona la herramienta R. Esta función computa el procedimiento para una serie temporal dada y nos devuelve un modelo que servirá para hacer los pronósticos y la posterior comparativa sobre los datos de evaluación. Además, al no fijar los parámetros α , β y γ del procedimiento, dejamos que sea él mismo el que trate de buscar los valores óptimos de dichos parámetros, minimizando los errores finales de la predicción. Así, estos parámetros han sido fijados automáticamente de la siguiente manera: $\alpha = 0,093$, $\beta = 0$ y $\gamma = 1$.

Con el modelo creado y entrenado, podemos invocar a la función `forecast` para predecir

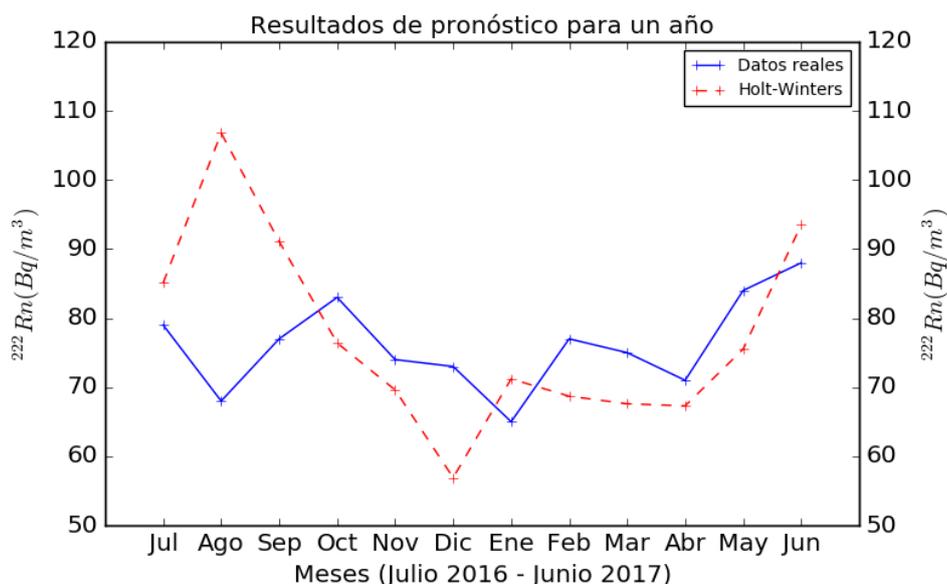


Figura 5.1: Comparativa entre datos de evaluación reales y pronosticados por *Holt-Winters* para el periodo comprendido entre Julio de 2016 y Junio de 2017.

los valores del último año completo y establecer el ECM y el EAM para este modelo, que son de $195,1 \pm 0$ y $10,5 \pm 0$, respectivamente.

En la figura 5.1, se establece la comparativa entre los datos de evaluación reales de la serie temporal, y los pronosticados haciendo uso de esta técnica.

5.2.2. Resultados experimentales con *ARIMA*

Para el caso de los modelos *ARIMA*, y volviendo a separar previamente los conjuntos de entrenamiento y de evaluación tal y como se ha explicado hasta ahora, vamos a hacer uso de la parte estacional de estos modelos. Esto se debe a que la serie temporal analizada presenta una modulación anual con incrementos de la concentración de ^{222}Rn en los meses de verano, y decrementos de su concentración en los meses de invierno, tal y como se puede observar en las figuras 4.2 y 4.3.

Por todo esto, estamos interesados en la fórmula de los modelos *ARIMA* estacionales, es decir: $ARIMA(p, d, q)(P, D, Q)_s$, donde (p, d, q) se refiere a la parte no estacional de la serie, y $(P, D, Q)_s$ se corresponde con la parte estacional de la misma (sección 3.1.1).

Uno de los grandes retos de estos modelos, es alcanzar una combinación adecuada de los parámetros $(p, d, q)(P, D, Q)$ que nos lleve al mejor pronóstico posible de la serie temporal

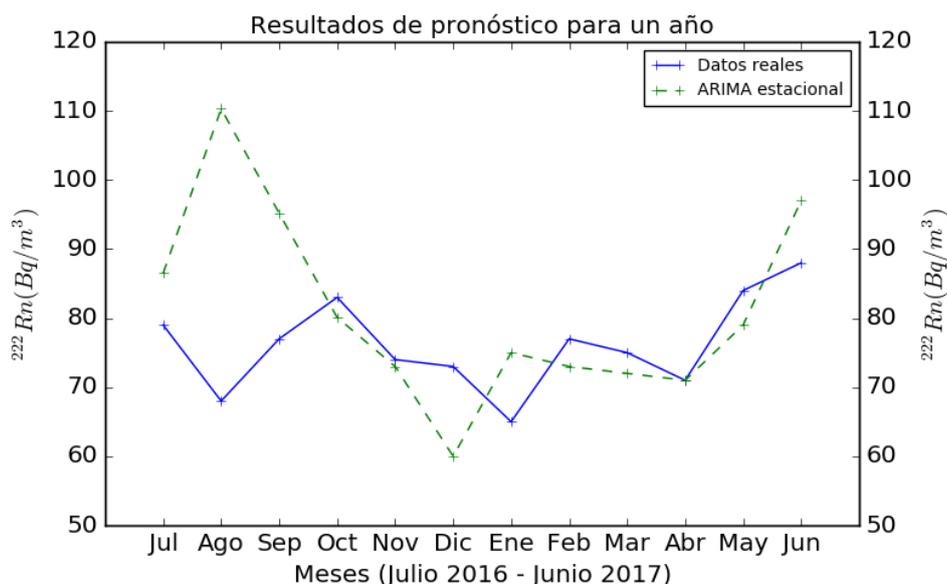


Figura 5.2: Comparativa entre datos de evaluación reales y pronosticados por *ARIMA* para el periodo comprendido entre Julio de 2016 y Junio de 2017.

estudiada. La determinación de dichos parámetros suele llevarse a cabo minimizando el *Akaike's Information Criterion* y el *Maximum Likelihood Estimation*. Una explicación en profundidad acerca de este tipo de modelos, puede encontrarse en [5].

Para nuestro caso de estudio, usaremos la función de R `auto.arima`, que recibe sólo los datos de entrenamiento y devuelve el mejor modelo *ARIMA* encontrado para nuestra serie temporal, en este caso, $ARIMA(1, 0, 0)(0, 1, 0)[12]$. El número 12 indica la frecuencia o número de observaciones por unidad de tiempo de la serie temporal observada, resultando del procesamiento de datos mensuales (medianas mensuales de ^{222}Rn) y de que el periodo natural es de un año [24, 25].

Una vez que dicho modelo ha sido creado, llevamos a cabo el pronóstico obteniendo un ECM de $215,3 \pm 0$ y un EAM de $9,7 \pm 0$.

En la figura 5.2, se establece la comparativa entre los datos de evaluación reales de la serie temporal, y los pronosticados haciendo uso de estos modelos.

5.2.3. Resultados experimentales con Descomposición STL

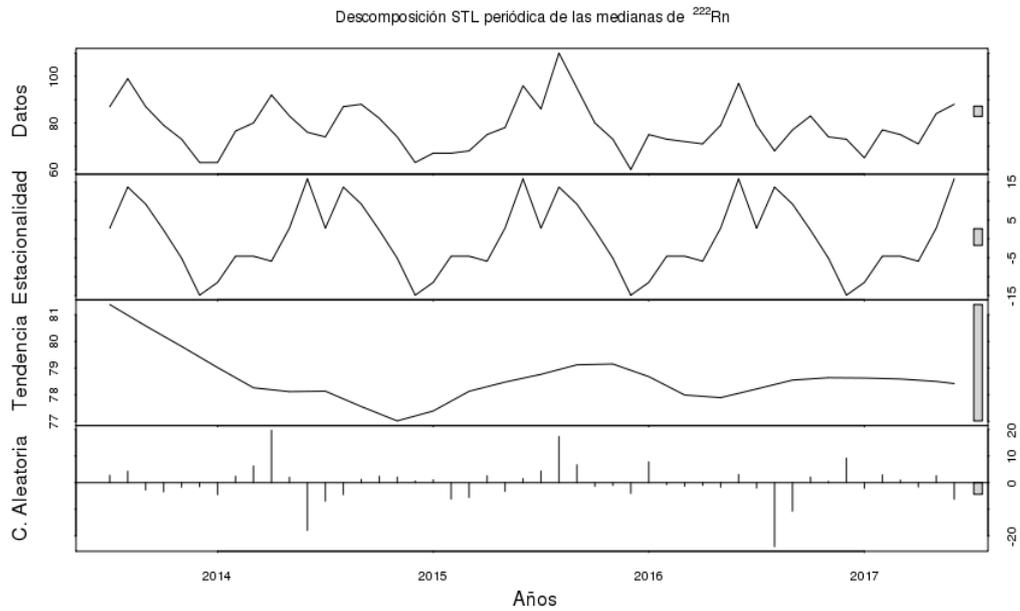
Basándonos en lo observado en las figuras 4.2 y 4.3, y en los análisis previos de los niveles de ^{222}Rn llevados a cabo en las tesis doctorales [24] y [25], cuyos periodos son de 379 ± 20 días

(ajuste sinusoidal para el intervalo comprendido entre Mayo de 2011 y Noviembre de 2012) y de 385 ± 1 días (ajuste cosenoidal para el intervalo comprendido entre Enero de 2012 y Enero de 2016), respectivamente, podemos considerar entonces que el ciclo de nuestra serie temporal es anual. De esta manera, podemos hacer uso de la descomposición STL como metodología válida de pronóstico. Para el presente estudio, se ha llevado a cabo una descomposición STL tanto periódica como no periódica de la serie temporal de las medianas mensuales de ^{222}Rn , cuyos resultados se representan en las figuras 5.3(a) y 5.4(a), respectivamente.

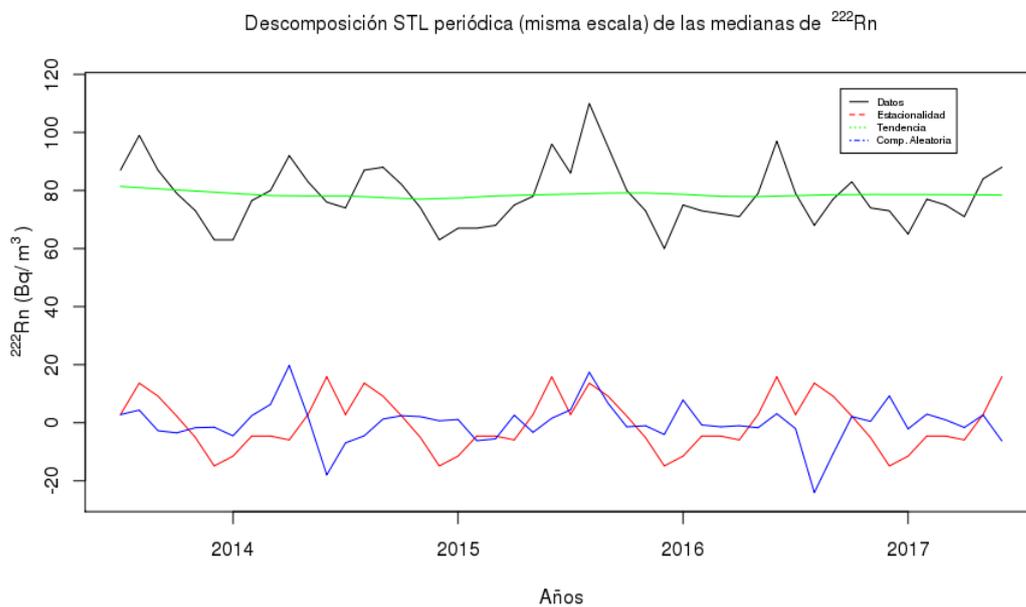
En dichas figuras, hay que darse cuenta además de que las escalas de las gráficas no son iguales. Así, la barra gris que aparece en la parte derecha de cada una de las componentes, nos indica la escala o magnitud relativa de las mismas con respecto a los datos originales. De esta manera, considerando por ejemplo la tendencia, la barra gris nos está indicando que si lo representásemos todo en la misma escala, esta componente se vería como una línea casi recta, como se puede comprobar en las figuras 5.3(b) y 5.4(b).

Como ya sabemos, los valores que aparecen en la serie temporal, son las medianas mensuales de ^{222}Rn comprendidas entre Julio de 2013 y Junio de 2017, debido a la robustez de estas en comparación a las medias. Tomando, como hasta ahora, los tres primeros años como conjunto de datos de entrenamiento y el último año como conjunto de datos de evaluación, vamos a llevar a cabo el pronóstico de la serie temporal distinguiendo los dos grandes casos comentados:

1. **Descomposición STL periódica.** En este primer caso, creamos un modelo invocando a la función `stl` de R con el parámetro `s.window` fijado a `periodic`. Esto indica que el suavizado realizado por la descomposición, es reemplazado por la media de las subseries creadas en dicho proceso (por ejemplo, la serie de todos los valores de enero, de febrero, y así sucesivamente). Con el modelo construido y haciendo uso del método Naïve comentado en la sección 3.1.1, obtenemos el pronóstico de nuestra serie temporal para el último año con un ECM de $117,5 \pm 0$ y un EAM de $7,3 \pm 0$.
2. **Descomposición STL no periódica.** En el segundo caso, volvemos a invocar a la misma función de R, pero esta vez asignándole al parámetro `s.window` el valor 7 por motivos de mejora en los resultados. Cuando dicho parámetro se proporciona en forma de lapso de la ventana de *Loess*, este número debe ser impar y al menos 7, de acuerdo a lo comentado

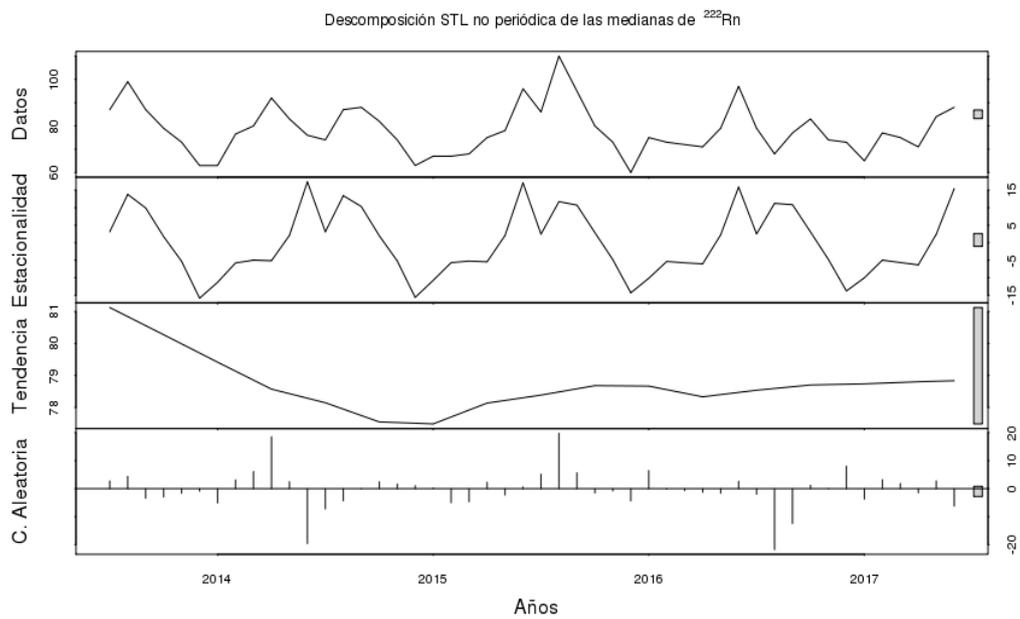


(a) Descomposición STL periódica para la serie temporal de las medianas de ^{222}Rn .

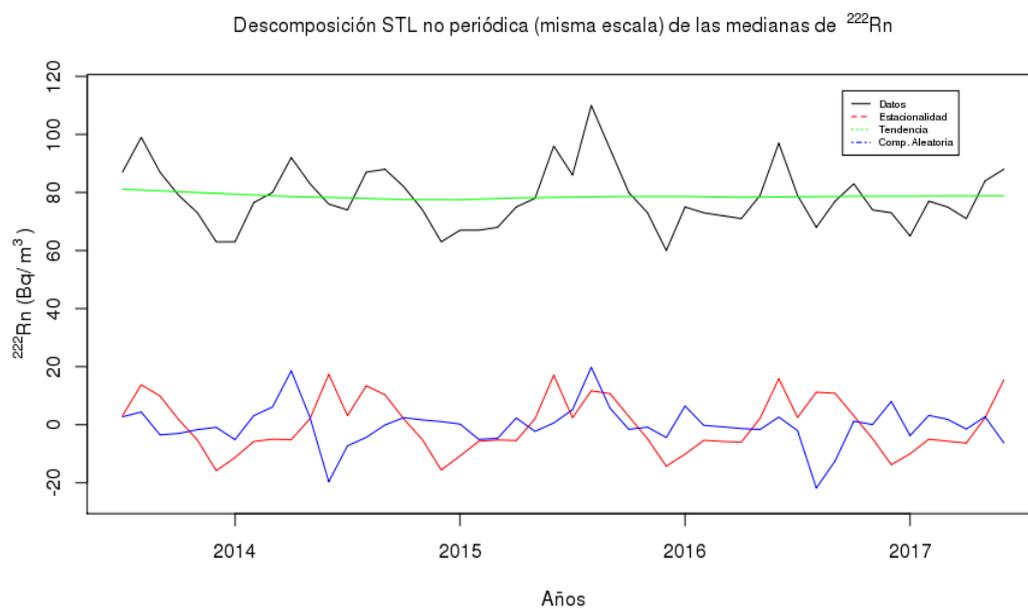


(b) Descomposición STL periódica a la misma escala para la serie temporal de las medianas de ^{222}Rn .

Figura 5.3: Descomposición STL periódica original y a la misma escala para la serie temporal de las medianas de ^{222}Rn .



(a) Descomposición STL no periódica para la serie temporal de las medianas de ^{222}Rn .



(b) Descomposición STL no periódica a la misma escala para la serie temporal de las medianas de ^{222}Rn .

Figura 5.4: Descomposición STL no periódica original y a la misma escala para la serie temporal de las medianas de ^{222}Rn .

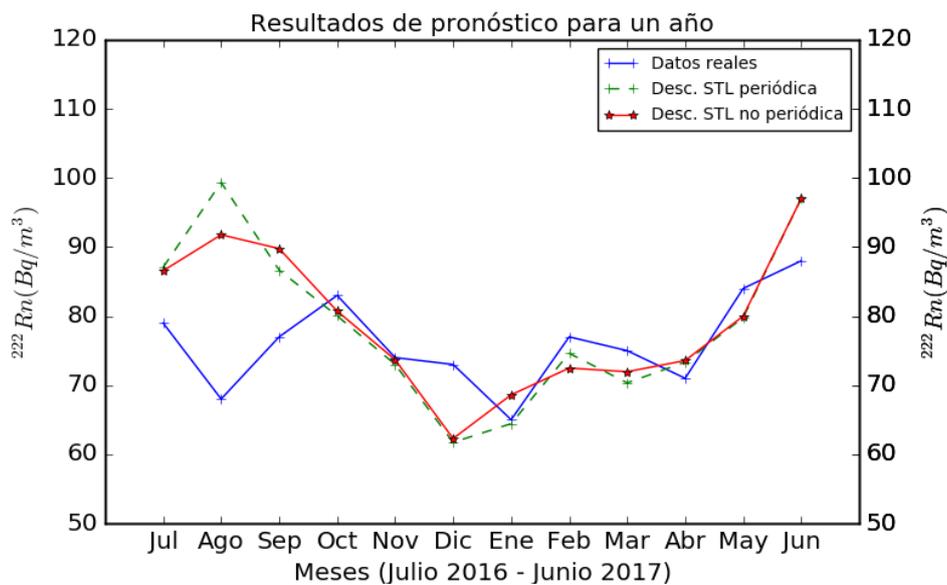


Figura 5.5: Comparativa entre datos de evaluación reales y pronosticados por la descomposición STL periódica y no periódica, para el periodo comprendido entre Julio de 2016 y Junio de 2017.

en [7]. Construyendo así el modelo y volviendo a hacer uso del método Naïve, llevamos a cabo el pronóstico obteniendo un ECM de $87,7 \pm 0$ y un EAM de $7,0 \pm 0$.

Como se acaba de comprobar, llevar a cabo el pronóstico haciendo uso de una descomposición STL no periódica, mejora significativamente los resultados de la predicción en comparación con la descomposición STL periódica. En la figura 5.5, se establece la comparativa entre los datos de evaluación reales y los pronosticados haciendo uso de ambos tipos de descomposición.

5.2.4. Resultados experimentales con Redes Neuronales *Feed-Forward*

Basándonos en los trabajos previos [8] y [26], en el presente proyecto se va a implementar una Red Neuronal *Feed-Forward* para llevar a cabo el pronóstico de los valores de ^{222}Rn referentes al último año. Este tipo de red constituye uno de los modelos más simples de redes neuronales existentes, y uno de los primeros en ser desarrollados.

Para nuestro caso de estudio en particular, tras recopilar los datos de las medianas mensuales de ^{222}Rn , hemos decidido crear muestras de entrada contiguas y solapadas de longitud 2, haciendo un total de 46. Esto nos permitirá disponer de un número adecuado de ejemplos con

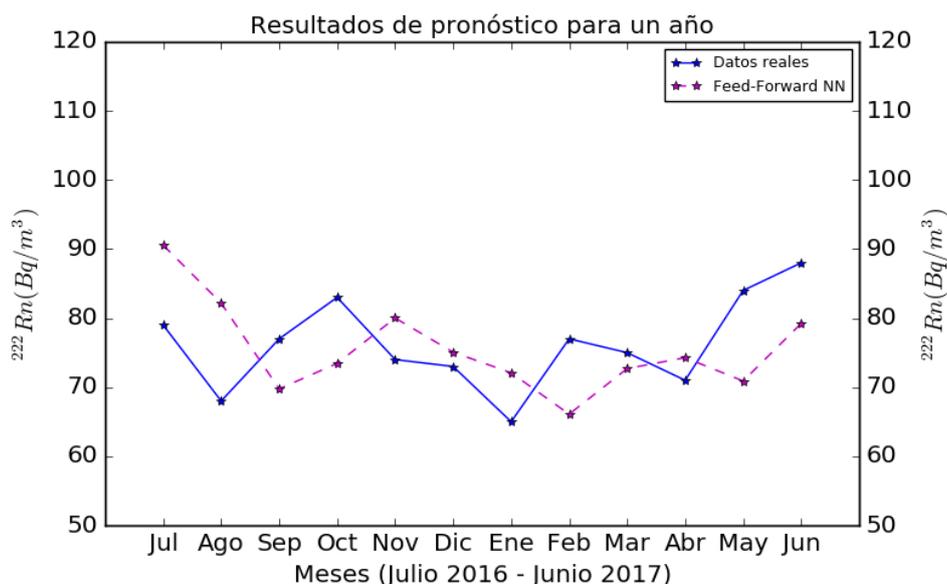


Figura 5.6: Comparativa entre los datos de evaluación reales y pronosticados por una Red Neuronal *Feed-Forward*, para el periodo comprendido entre Julio de 2016 y Junio de 2017.

los que entrenar la red.

Por su parte, en cuanto a parámetros de configuración se refiere y con el objetivo de obtener el mejor rendimiento final posible, se ha creado una estructura de red neuronal con 2 capas ocultas haciendo uso de la función `Dense`, que da lugar a capas de neuronas completamente conectadas. Para la primera capa se crean 24 neuronas con función de activación `relu`, mientras que para la segunda se crean 12 neuronas con la misma función de activación. Antes de crear la capa de salida, es necesario aplanar los datos por medio de la función `Flatten` para evitar problemas de dimensionalidad en la red. Tras dicho aplanamiento, se añade la capa de salida con una neurona y con una función de activación lineal.

Con todo el proceso anterior realizado, y tras compilar el modelo y crear los conjuntos de entrenamiento y de evaluación de la manera comentada al inicio del capítulo 5, se procede a entrenar la red con 100 épocas. Al llevar a cabo el pronóstico a partir de este modelo, obtenemos un ECM de $83,2 \pm 3,7$ y un EAM de $8,1 \pm 0,2$.

Finalmente, en la figura 5.6, se establece la comparativa entre los datos de prueba reales y los pronosticados haciendo uso de este tipo de Redes Neuronales Artificiales.

5.2.5. Resultados experimentales con Redes Neuronales Recurrentes

A diferencia de las Redes Neuronales *Feed-Forward* analizadas en el apartado anterior, las Redes Neuronales Recurrentes son capaces de usar sus estados internos para llevar a cabo el procesamiento de secuencias de entradas. Puesto que estamos trabajando con la serie temporal de las medianas mensuales del ^{222}Rn y creando muestras de tamaño fijo a partir de ella, el hecho de que este procesamiento sea posible resulta de especial interés para nuestro caso de estudio. Además, esto posibilita la aplicabilidad de este tipo de redes a muchos estudios de reconocimiento de escritura y de voz.

Como ya se explicó en el apartado dedicado a las Redes Neuronales Recurrentes dentro de la sección 3.1.2, el control de los estados internos de la red forma parte de unas unidades especiales que pueden ser las LSTM o las GRU. Para nuestro caso de estudio en concreto, hemos hecho uso de neuronas LSTM como unidades pertenecientes a las capas ocultas de la Red Neuronal Recurrente.

Antes de crear las muestras de entrada contiguas y solapadas de tamaño fijo, en este caso de longitud 2 y haciendo un total de 46, es necesario proceder a la normalización de los datos del ^{222}Rn , ya que este tipo de redes son muy sensibles a trabajar con valores que se encuentran muy alejados de los manipulados por las funciones de activación. La tangente hiperbólica (\tanh) es la función de activación más típicamente usada por las Redes Neuronales Recurrentes. Esto es así, porque su rango de valores comprendido entre -1 y 1, permite a la red decidir de manera automática qué información eliminar de su estado interno, y qué información mantener a largo plazo por resultar más relevante. Es por este motivo por el que normalizamos nuestra serie temporal a valores comprendidos dentro del rango comentado.

Por lo que a los parámetros de configuración de la red respecta y con el objetivo de minimizar el ECM y el EAM finales, se ha decidido crear una Red Neuronal Recurrente con 2 capas ocultas: la primera con 24 neuronas LSTM y función de activación \tanh ; y la segunda, con 12 neuronas LSTM y la misma función de activación. Por último, se añade una capa de salida con una neurona y función de activación lineal.

Además de lo comentado, entre las dos capas ocultas y entre la segunda capa oculta y la capa de salida, se han incorporado operaciones de `Dropout` con el objetivo de prevenir el sobreajuste de la red. Simplemente, consiste en establecer al azar una tasa fraccional de unidades de entrada a 0 en cada actualización durante el tiempo de entrenamiento. En este caso,

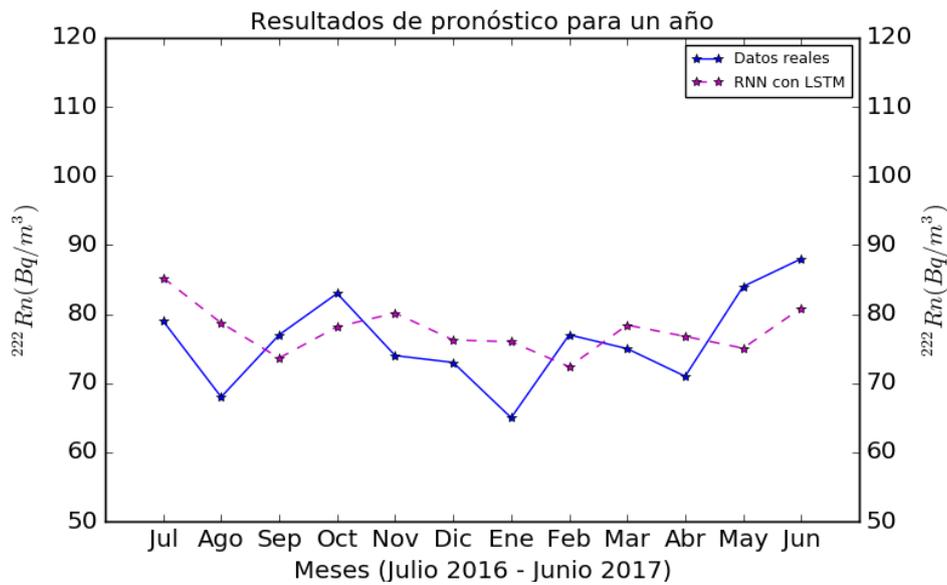


Figura 5.7: Comparativa entre datos de evaluación reales y pronosticados por una Red Neuronal Recurrente con neuronas LSTM, para el periodo comprendido entre Julio de 2016 y Junio de 2017.

el uso de esta técnica de regularización con una tasa de 0.6, sí contribuye a una mejora de los resultados. Sin embargo, esto no siempre tiene porqué ser así, y puede suceder que la operación de Dropout afecte al rendimiento cuando se da alguna de las siguientes situaciones¹:

1. Cuando se aplica justo antes de la última capa, puesto que en ese punto la red no tiene posibilidad de corregir los errores inducidos por la operación.
2. Cuando la red es pequeña en relación al conjunto de datos manipulado.
3. Cuando el tiempo de entrenamiento es limitado y no se tiene un número suficiente de épocas. Normalmente, ocurre que la operación de Dropout induce más errores al principio del entrenamiento que al final, momento en el que convergen los resultados.

En muchos casos, la operación de Dropout se suele sustituir por una técnica denominada *Batch Normalization*, cuya descripción en detalle se puede encontrar en [27].

Con todas las consideraciones anteriores, y tras compilar el modelo y crear los conjuntos de entrenamiento y de evaluación de la manera ya comentada, se procede a entrenar la red con 100 épocas.

¹<https://stats.stackexchange.com/questions/299292/dropout-makes-performance-worse>

Finalmente, se lleva a cabo el pronóstico de los datos de evaluación obteniendo un ECM de $47,7 \pm 0,9$ y un EAM de $6,3 \pm 0,1$, constituyendo esta una de las mejores técnicas aplicadas en el presente trabajo. En la figura 5.7, se establece la comparativa entre los datos de prueba reales y los pronosticados haciendo uso de este tipo de redes neuronales con neuronas LSTM.

5.2.6. Resultados experimentales con Redes Neuronales Convolucionales

Como ya sabemos, las Redes Neuronales Convolucionales son un tipo especializado de redes neuronales que hacen gran énfasis en el procesamiento de imágenes [9], aunque hoy en día están siendo ampliamente usadas también para el análisis de series temporales [28, 29].

Una Red Neuronal Convolutiva se caracteriza por estar compuesta de un conjunto de capas convolucionales que alternan filtros de convolución, funciones no lineales de activación y funciones de agrupación de vecindades. Todo ello, permite a la red extraer características relevantes de los objetos, independientemente de la posición que ocupan en las muestras de datos manipuladas. De alguna forma, se puede decir que las CNN construyen un modelo con un conjunto de filtros capaces de reconocer patrones específicos tanto en imágenes como en series temporales, capturando así información representativa de dichos elementos.

Para el caso de estudio actualmente tratado, en lugar de trabajar con un conjunto de imágenes, nuestra serie temporal de ^{222}Rn (o los elementos que vayamos a usar como entradas de la red) tiene que ser dividida en un conjunto de muestras contiguas y solapadas. Estas muestras tendrán un tamaño previamente fijado y constituirán los ejemplos a partir de los cuales la CNN tratará de encontrar los posibles patrones de nuestra serie para pronosticar, de la mejor manera posible, los datos de evaluación de la misma.

Con el objetivo de cubrir tanto la convolución unidimensional como la bidimensional, y dados los interesantes resultados de pronóstico obtenidos haciendo uso de la descomposición STL no periódica analizada en la sección 5.2.3 con un ECM de $87,7 \pm 0$, se han realizado dos grandes estudios con estas redes:

1. En el caso de la convolución unidimensional hacemos uso única y exclusivamente de la serie temporal de las medianas mensuales de ^{222}Rn . Así, al trabajar solamente con esta serie temporal, no va a ser necesario realizar una normalización de dichos datos.

La elección de los parámetros de configuración de dicha red, están elegidos con el objetivo

de conseguir los valores del ECM y del EAM finales más bajos posibles. Así, crearemos muestras contiguas solapadas de longitud 12, haciendo un total de 36. Se usarán dos capas de convolución con características semejantes salvo en el número de filtros. En la primera capa, se aplicarán 32 filtros de convolución de tamaño 3, funciones de activación `relu` y un agrupamiento máximo de vecindades de tamaño 2. Por su parte, para la segunda capa, usaremos 64 filtros también de tamaño 3 y unas funciones de activación y agrupamiento máximo similares a las de la capa anterior. Es importante mencionar que la elección de estos números debe ser cuidadosa a fin de evitar problemas referentes a la dimensionalidad de los datos manipulados. Finalmente, se aplanan los datos para construir una red neuronal completamente conectada y se añade la capa de salida con una neurona y función de activación lineal.

Con el modelo creado y compilado, y habiendo establecido los conjuntos de entrenamiento y evaluación como se ha mencionado hasta ahora (3 años y 1 año, respectivamente), se procede a entrenar la red con 10 épocas. Dado que al trabajar con las medianas mensuales de ^{222}Rn no tenemos un conjunto de datos demasiado amplio, utilizamos este número de épocas para evitar llegar al sobreajuste de la red.

Una vez realizado el entrenamiento, podemos llevar a cabo el pronóstico de los datos de evaluación, obteniendo un ECM de $59,4 \pm 4,2$ y un EAM de $6,6 \pm 0,6$.

2. En el segundo caso de estudio con CNN se pretenden dos cosas: por un lado, aprovechar los resultados obtenidos en la descomposición STL no periódica para intentar mejorar los resultados con redes convolucionales; y, por el otro, trabajar con la convolución bidimensional manipulando las series temporales que componen la serie de medianas mensuales de ^{222}Rn como si de una imagen se tratara.

Por este motivo, se han extraído los datos numéricos referentes a las series temporales de estacionalidad, tendencia y componente aleatoria de la gráfica 5.4(a). Así, en lugar de trabajar ahora con muestras de entrada creadas a partir de la serie temporal de medianas de ^{222}Rn , creamos dichas muestras a partir de las componentes en las que se descompone la serie. La salida de la red seguirá haciendo referencia a los valores de Radón de la serie original.

Sin embargo, al trabajar ahora con distintas componentes cuyos valores están bastante

alejados entre sí, como se puede comprobar en la figura 5.4(b) entre la tendencia y la estacionalidad o la componente aleatoria, vamos a distinguir dos subcasos (en este segundo caso de estudio) en función de la necesidad o no de normalizar los datos manipulados:

- a) En el primer subcaso, creamos tres redes convolucionales con características semejantes, una para la serie de la estacionalidad, otra para la tendencia y la última para la componente aleatoria. Al tener cada serie una red convolucional propia, no será necesario normalizar los datos de entrada, puesto que en cada una de ellas la función de agrupación máxima de la red se aplicará de manera separada.

En cuanto a los parámetros de configuración de la red, se han escogido los mismos valores del caso anterior. La diferencia es que ahora tenemos tres redes convolucionales con dos capas de convolución cada una: la primera con 32 filtros de tamaño 3, función de activación `relu` y función de agrupamiento máximo de tamaño 2; y la segunda con 64 filtros de tamaño 3 y las mismas funciones de activación y agrupamiento máximo. Otra diferencia es que se recibirán muestras de entrada contiguas y solapadas de longitud 12 de cada una de las tres series manipuladas. Tras aplanar las redes convolucionales y añadir sus respectivas capas de salida con una neurona y función de activación lineal, se procede a concatenar los modelos en un modelo final –haciendo uso de la función `Merge`– que, después de crear la última capa de salida final con una neurona y función de activación lineal, nos permitirá llevar a cabo el entrenamiento y posterior pronóstico de los datos de evaluación.

Por la misma razón que en el caso anterior, el entrenamiento se hará haciendo uso de 10 épocas, previa separación de los conjuntos de entrenamiento y prueba. Con todo ello, podemos llevar a cabo el pronóstico obteniendo un ECM de $45,0 \pm 7,2$ y un EAM de $4,9 \pm 0,5$.

- b) En el segundo subcaso, creamos una matriz bidimensional como resultado de apilar las tres series temporales en las que se descompone la serie de las medianas de ^{222}Rn . Sin embargo, previo a esta operación, es necesario normalizar los datos manipulados a valores entre 0 y 1 de manera separada. El objetivo de

esto es evitar que, a la hora de trabajar con las muestras creadas a partir de la matriz bidimensional, nuestras funciones de agrupación máxima seleccionen exclusivamente los elementos que componen la tendencia a la hora de llevar a cabo la convolución. Por su parte, la elección del rango de valores entre 0 y 1 se debe al uso de funciones de activación `relu`, cuyo rango válido está comprendido entre 0 e infinito.

Con las series normalizadas, creamos una matriz resultante de 3x48. Ahora, en cuanto a los parámetros de configuración de la red y considerando las restricciones de dimensionalidad en los datos, crearemos dos capas de convolución: la primera con 32 filtros de tamaño 3x3, función de activación `relu` y agrupamiento máximo de 1x2; y la segunda con 64 filtros de tamaño 1x3 y función de activación y agrupamiento máximo idénticas a las de la capa anterior. Por último, aplanamos los datos y creamos la capa de salida con una neurona y función de activación lineal.

Las muestras creadas serán de tamaño 3x12 (figura 5.8) dando lugar a un total de 36. A la hora de configurar la entrada de la red, el framework *Keras* admite dos estructuras de la misma, una en la que se determinan los canales de la imagen/matriz primero, y la otra en la que se determinan al final. Como en nuestro caso solo vamos a trabajar con un canal (tenemos una matriz de profundidad 1, no una imagen RGB por ejemplo), basta con preguntar con una llamada de *Keras* si tenemos configuración `channels_first` o `channels_last`, estableciendo la dimensionalidad correcta de la entrada en consecuencia.

Con todo lo mencionado hasta el momento, y con el modelo compilado, podemos entrenar la red dividiendo previamente el conjunto de datos en entrenamiento y evaluación, como siempre. Como en los otros casos, el número de épocas usadas para entrenar la red será de 10.

Una vez realizado el entrenamiento, se lleva a cabo el pronóstico de los datos de evaluación, obteniendo un ECM de $63,4 \pm 13,1$ y un EAM de $6,4 \pm 0,6$.

En la figura 5.9, se establece la comparativa entre los datos de prueba reales y los pronosticados haciendo uso de los tipos de CNN desarrollados. Una de las diferencias más representativas de este tipo de modelos (aparte de la obtención de unos valores más bajos para

las métricas analizadas) con respecto a los modelos clásicos, es la capacidad que tienen las redes convolucionales de predecir un comportamiento decremental de la serie durante el mes de agosto de 2016. Como se puede observar en las técnicas clásicas, el pronóstico para este mes estaba bastante alejado de la realidad. Esto nos fortalece en la idea del gran potencial que tiene el aprendizaje profundo a la hora de reconocer patrones y poder realizar pronósticos futuros de información de una manera bastante acertada.

	0.57	0.89	0.78	0.53	0.32	0.	0.14	0.30	0.33	0.32	0.54	1.	E.
3	1.	0.92	0.84	0.77	0.69	0.61	0.53	0.45	0.38	0.30	0.26	0.22	T.
	0.59	0.63	0.44	0.45	0.48	0.50	0.40	0.60	0.67	0.97	0.58	0.05	C.A.

12

Figura 5.8: Ejemplo de imagen (utilizada como muestra de entrenamiento) formada a partir de las componentes normalizadas de la descomposición STL no periódica de la serie temporal de medianas mensuales de ^{222}Rn . Las filas, de arriba a abajo, hacen referencia a la Estacionalidad, la Tendencia y la Componente Aleatoria.

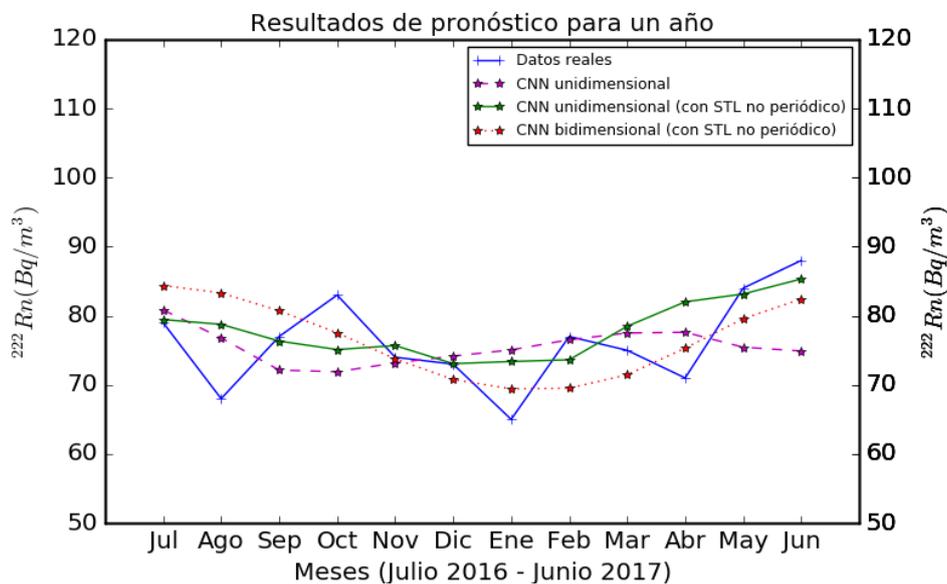


Figura 5.9: Comparativa entre datos de evaluación reales y pronosticados por una CNN unidimensional, una CNN unidimensional que hace uso de las componentes de la descomposición STL no periódica del ^{222}Rn , y una CNN bidimensional que hace uso de lo mismo, para el periodo comprendido entre Julio de 2016 y Junio de 2017.

5.3. Comparativa general de metodologías

En la presente y última sección de resultados, se presenta una comparativa general de las metodologías empleadas en el presente estudio, en aras de:

1. Determinar el mejor o mejores modelos existentes, de entre los estudiados, para llevar a cabo futuros pronósticos del ^{222}Rn .
2. Establecer cómo de eficientes y precisas son las metodologías basadas en aprendizaje automático y profundo en comparación con las metodologías clásicas de pronóstico.

Cuadro 5.1: Predictores usados, media y desviación estándar del ECM y el EAM para cada uno de ellos tras 25 ejecuciones.

Predictores	ECM	EAM
<i>Holt-Winters</i>	195,1 ± 0	10,5 ± 0
ARIMA estacional	215,3 ± 0	9,7 ± 0
Descomposición STL periódica	117,5 ± 0	7,3 ± 0
Descomposición STL no periódica	87,7 ± 0	7,0 ± 0
FFNN	83,2 ± 3,7	8,1 ± 0,2
RNN con neuronas LSTM	47,7 ± 0,9	6,3 ± 0,1
CNN unidimensional	59,4 ± 4,2	6,6 ± 0,6
CNN unidimensional (con STL no periódico)	45,0 ± 7,2	4,9 ± 0,5
CNN bidimensional (con STL no periódico)	63,4 ± 13,1	6,4 ± 0,6

De esta manera, en la tabla 5.1, se muestra una lista con las metodologías analizadas y sus correspondientes ECM y EAM² después de llevar a cabo los procesos de entrenamiento y pronóstico. Como se puede observar en dicha tabla, las técnicas basadas en aprendizaje profundo son las técnicas que han obtenido unos valores más bajos para el ECM y el EAM, seguidos de los pronósticos obtenidos haciendo uso de la FFNN y de la descomposición STL no periódica de la serie temporal del ^{222}Rn . Por lo tanto, estas metodologías –principalmente las RNN con neuronas LSTM y los tipos de CNN implementados– son las que mejor rendimiento

²Datos obtenidos (media y desviación estándar) tras 25 ejecuciones de cada uno de los modelos analizados.

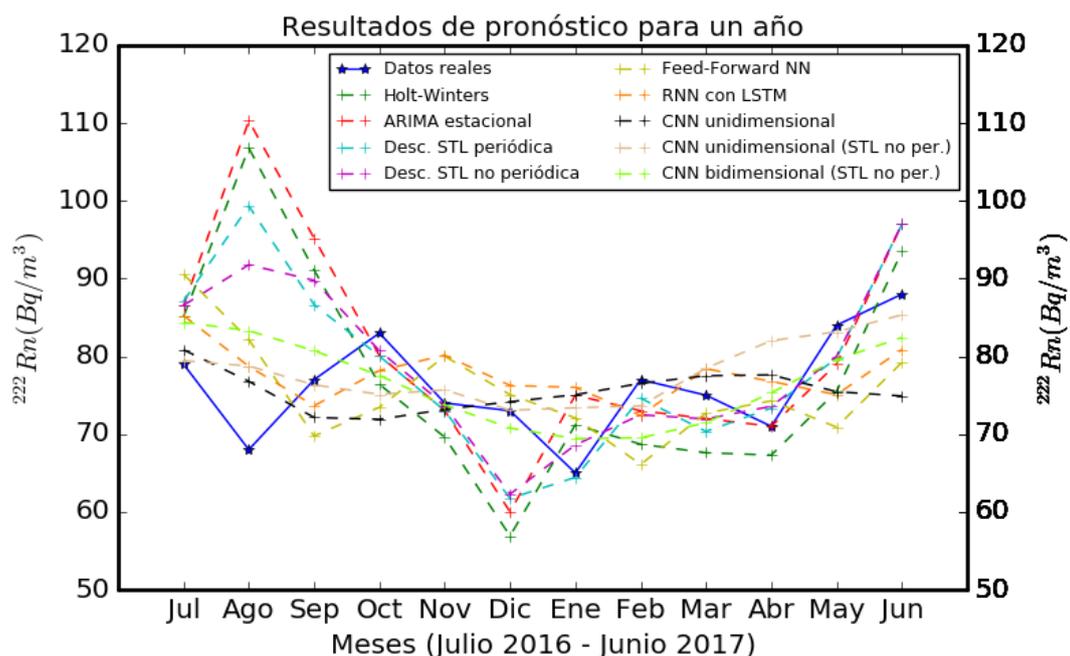


Figura 5.10: Comparativa general entre los datos de evaluación reales empleados y los pronosticados por todos y cada uno de los modelos analizados en el presente estudio, para el periodo comprendido entre Julio de 2016 y Junio de 2017.

predictivo presentan y, por este motivo, serán las potenciales candidatas a seleccionar a la hora de realizar futuros pronósticos del gas estudiado. Así, esto supondrá un paso de evolución de las técnicas clásicas de pronóstico hacia el uso de las emergentes técnicas de aprendizaje profundo.

Además de observar cómo el ECM y el EAM se reducen considerablemente en las técnicas basadas en aprendizaje profundo en comparación con las metodologías clásicas de pronóstico, uno de los resultados más relevantes logrados, como ya se ha mencionado con anterioridad y tal y como se puede observar en la figura 5.10, es la capacidad que tienen las técnicas basadas en aprendizaje profundo de tratar de adaptarse a la forma de la serie temporal original, con mención especial para los meses comprendidos entre Julio de 2016 y Septiembre de ese mismo año, patrón que las demás técnicas de pronóstico no perciben de igual manera al principio de los datos de evaluación.

Capítulo 6

Conclusiones

En el presente trabajo de fin de máster, se ha llevado a cabo el modelado y pronóstico de la serie temporal de la concentración de ^{222}Rn en el Laboratorio Subterráneo de Canfranc, haciendo uso de un amplio abanico de metodologías y técnicas. Tal y como se mencionó al principio de este documento, un alto nivel de concentración de este gas, puede conllevar errores y perturbar los resultados obtenidos en muchos experimentos que se desarrollan en laboratorios subterráneos. Así, es por este motivo, por el que se hace tan necesario el control, modelado y pronóstico de este tipo de variables ambientales.

Los datos recogidos que han hecho posible el estudio realizado, se extienden dentro del intervalo de tiempo comprendido entre Julio de 2013 y Junio de 2017, haciendo un total de cuatro años completos. Sin embargo, a día de hoy, estos datos se siguen recogiendo haciendo ya un total de casi cinco años. Las principales tareas realizadas incluyen tanto el preprocesado y limpieza de los datos en bruto recopilados, como el posterior análisis y comprensión de las características más relevantes de los mismos. Para dicho análisis y comprensión, se han utilizado técnicas clásicas de pronóstico y aquellas que están basadas en aprendizaje automático y profundo, a fin de establecer comparativas entre ellas y estudiar cómo estas últimas nos conducen a mejores resultados en comparación a las primeras, como se ha podido comprobar. De esta manera, tratamos de hacer visible el gran potencial que tienen las técnicas basadas en aprendizaje profundo de cara a su aplicación en diversos estudios, campos y empresas. Las técnicas estudiadas incluyen: suavizado exponencial *Holt-Winters*, *ARIMA* estacional, Descomposición en Estacionalidad y Tendencia usando el método *Loess*, Redes Neuronales *Feed-Forward*, Redes Neuronales Recurrentes y Redes Neuronales Convolucionales. Las pruebas realizadas,

demuestran la gran mejora de eficiencia existente en las técnicas basadas en aprendizaje profundo en comparación a las técnicas clásicas de pronóstico, siendo las Redes Neuronales Recurrentes y las Redes Neuronales Convolucionales unidimensionales que hacen uso de las componentes de la descomposición STL no periódica de la serie temporal, las que han obtenido unos valores más bajos para las métricas estudiadas y, por tanto, las que mejor rendimiento predictivo presentan.

Tras el preprocesamiento previo de los datos disponibles para elaborar el presente trabajo, se observó que los niveles de concentración de ^{222}Rn están perfectamente controlados, y que no existe una diferencia representativa entre los mismos en diferentes salas del laboratorio. Además, se comprobó que estos datos no se encuentran correlacionados con otras variables ambientales como la temperatura o la presión del aire. Sólo una correlación con los datos recogidos referentes a la humedad dentro de la caverna, fue hallada.

Por su parte, se utilizaron los lenguajes R y Python, este último junto con el *framework Keras* para redes neuronales. Ambas herramientas disponen de una gran cantidad de recursos para trabajar con representaciones gráficas y librerías para manipulación de datos. Además de los códigos adjuntos a esta memoria, se han desarrollado otros scripts necesarios para el procesamiento previo de la información y para la extracción de la mayoría de las gráficas mostradas en el presente documento, alcanzando con esto un grado de profundización en estos lenguajes mucho más elevado que en el momento de inicio de desarrollo del mismo.

Finalmente, y como se mencionó en el capítulo 2, se escribió un artículo científico relacionado con el presente trabajo y se envió como comunicación al congreso HAIS 2018, referente a sistemas basados en Inteligencia Artificial. Tras los comentarios de los revisores y las oportunas correcciones, el artículo fue aceptado para dicho congreso [10].

6.1. Trabajos futuros

El trabajo continuo de toma de datos comentado, requiere una continuidad en las tareas de análisis y modelado de los mismos. De esta forma, la incesante actividad de los detectores de ^{222}Rn capturando sus niveles de concentración cada 10 minutos, nos llevará a un incremento de la información disponible que, tras ser preprocesada, podrá ser incorporada a nuestras técnicas a fin de aumentar el conjunto de datos de entrenamiento y poder así mejorar la detección de

patrones y la eficiencia final.

Por su parte, incorporando más años completos de datos, los modelos propuestos no solo podrían incrementar su capacidad predictiva, sino que también habría que computar conjuntos de entrenamiento más amplios. Así, en el caso de las técnicas basadas en aprendizaje profundo, el tiempo de procesamiento podría constituir una barrera en el proceso de pronóstico. De esta manera, el uso de Unidades de Procesamiento Gráfico (GPU) cada vez más potentes, podría suponer una mejora en la eficiencia computacional del procesamiento de mayores cantidades de datos.

Además, se plantea añadir al estudio información adicional de variables meteorológicas exógenas, como por ejemplo la información ambiental (temperatura, precipitaciones, etc.) de localidades cercanas a la ubicación del laboratorio. Esto permitiría estudiar posibles correlaciones de datos externos con los niveles de ^{222}Rn dentro del laboratorio. De existir dichas correlaciones, se podrían utilizar algoritmos como *Random Forests* [30, 31, 32] u otras técnicas de reducción de dimensionalidad, para decidir qué variables exógenas son más representativas en relación con los datos de la serie temporal original y seleccionarlas para usarlas como conjuntos de datos de entrenamiento en las metodologías basadas en aprendizaje profundo.

Finalmente, se propone escribir nuevas comunicaciones para congresos y/o revistas científicas con todos y cada uno de los resultados novedosos y mejoras que se puedan ir extrayendo en base a los trabajos futuros aquí mencionados.

6.2. Publicaciones

Durante el periodo de realización del máster y del trabajo final asociado al mismo, se han venido desarrollando artículos y comunicaciones para congresos, tanto en relación con el trabajo aquí realizado, como aquellos relacionados con otros campos de la informática de gran interés.

Las publicaciones aceptadas y las propuestas como parte del trabajo futuro, se recogen a continuación:

1. En primer lugar, me gustaría destacar el artículo escrito en relación con el presente Trabajo de Fin de Máster. Este artículo, titulado *Modelling and Forecasting of the ^{222}Rn Radiation Level Time Series at the Canfranc Underground Laboratory* [10], fue aceptado como comunicación para el congreso HAIS 2018, celebrado del 20 al 22 de Junio

de 2018 en Oviedo (Asturias), España. Dicho artículo recoge algunos de los aspectos más relevantes del presente proyecto, desde el preprocesamiento y estudio previo de las correlaciones en los datos, hasta la comparativa en rendimiento de las metodologías clásicas de pronóstico con respecto a algunas de las técnicas de aprendizaje automático y profundo estudiadas.

2. En cuanto a las publicaciones y documentos propuestos en relación con la continuidad y seguimiento de este trabajo, se tienen los siguientes:

a) El primero es un informe técnico realizado en mi centro de trabajo que trata todos y cada uno de los aspectos recogidos en este TFM, incluyendo cierta temática más relacionada con el campo de la Física y una pequeña parte de los trabajos futuros mencionados en la sección anterior, como es el estudio de la correlación del ^{222}Rn con variables ambientales exógenas.

b) Un *Journal* o revista científica, como ampliación del trabajo presentado en HAIS 2018. Se añadirían las técnicas de aprendizaje profundo que no se han incluido en dicho artículo, el estudio completo de la correlación con las variables ambientales exógenas y la incorporación del casi ya disponible quinto año entero de datos, que nos permitirá disponer de más información a fin de mejorar el rendimiento de las metodologías empleadas. La elección de la revista elegida se basará en criterios de calidad, teniendo en cuenta un factor de impacto que la sitúe en el Q1 o Q2.

3. Finalmente, en cuanto a publicaciones aceptadas y en desarrollo, ajenas al problema aquí tratado y desarrolladas durante el periodo de realización del máster, tenemos:

a) Un artículo de congreso titulado *An Accuracy-Aware Implementation of Two-Point Three-Dimensional Correlation Function using Bin-Recycling Strategy on GPU*, presentado y defendido en *2nd International Workshop on Distributed Big Data Management (DBDM 2017)* como parte del *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2017)*, celebrado en Madrid (España) entre los días 14 y 17 de Mayo de 2017. En este artículo se computa la función de correlación cosmológica de dos puntos en tres dimensiones haciendo uso de una arquitectura de procesamiento gráfico Fermi C2075.

- b) Un artículo publicado en arXiv, titulado *Report: Performance comparison between C2075 and P100 GPU cards using cosmological correlation functions*, donde se evalúa la diferencia de rendimiento entre las arquitecturas de procesamiento gráfico Fermi C2075 y Pascal P100 a la hora de computar distintas funciones de correlación cosmológicas.
- c) Un artículo también aceptado para HAIS 2018 donde se propone una generalización del *Gaussian Mixture Model*, haciendo uso de un algoritmo evolutivo denominado Optimizador por Arrecifes de Coral, cuyo objetivo es optimizar los parámetros de una mezcla de distintas distribuciones de probabilidad. Dicho artículo se titula *Generalized Probability Distribution Mixture Model for Clustering*.
- d) Un artículo en desarrollo para la conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2018), que se celebrará en Granada (España) del 23 al 26 de Octubre de 2018. En este artículo, se quiere realizar una comparativa de rendimiento entre el uso directo de Redes Neuronales Convolucionales sobre distintas series temporales sin relación entre sí, y haciendo uso de sus descomposiciones STL previas en dichas redes.
- e) Finalmente, otro artículo en desarrollo para el *19th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2018)*, que se celebrará en Madrid (España) del 21 al 23 de Noviembre de 2018. En este artículo, se evalúa el rendimiento de una arquitectura de ensamble formada por distintas variantes de Redes Neuronales Recurrentes y Convolucionales para el pronóstico de series temporales, en comparación a las arquitecturas individuales que la integran.

Bibliografía

- [1] Bettini, A.: New underground laboratories: Europe, Asia and the Americas. Physics of the Dark Universe **4**(Supplement C) (2014) 36 – 40 DARK TAUP2013.
- [2] Kalekar, P.S.: Time series forecasting using Holt-Winters exponential smoothing. (01 2004)
- [3] Brown, R.G.: Exponential smoothing for predicting demand (1956)
- [4] Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Management Science **6**(3) (1960) 324–342
- [5] Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2014)
- [6] Newbold, P.: ARIMA model building and the time series analysis approach to forecasting. Journal of Forecasting **2**(1) (1983) 23–35
- [7] Cleveland, R.B., Cleveland, W.S., McRae, J., Terpenning, I.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics (1990) 3–73
- [8] Tang, Z., Fishwick, P.A.: Feed-Forward Neural Nets as models for time series forecasting. ORSA Journal on Computing (1993) 374 – 385
- [9] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
- [10] Méndez-Jiménez, I., Cárdenas-Montes, M.: Modelling and Forecasting of the ^{222}Rn Radiation Level Time Series at the Canfranc Underground Laboratory. In: Hybrid Artificial Intelligent Systems - 13th International Conference, HAIS 2018, Oviedo, Spain, June 20-22, 2018, Proceedings. (2018) 158–170

- [11] Shih, S.H., Tsokos, C.P.: A weighted moving average process for forecasting. *Modern Applied Statistical Methods* **7**(1) (2008) 187 – 197
- [12] Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: *Time Series Analysis: Forecasting and Control*. Wiley (2015)
- [13] Becker, S., Lecun, Y.: Improving the convergence of Back-Propagation learning with second-order methods. (01 1989) 29 – 37
- [14] E. Rumelhart, D., E. Hinton, G., J. Williams, R.: Learning representations by back propagating errors. **323** (10 1986) 533–536
- [15] Schuster, M., Paliwal, K.: Bidirectional Recurrent Neural Networks. *Trans. Sig. Proc.* **45**(11) (November 1997) 2673–2681
- [16] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. **9** (12 1997) 1735–80
- [17] Gers, F.A., Schmidhuber, J.A., Cummins, F.A.: Learning to forget: Continual prediction with LSTM. *Neural Comput.* **12**(10) (October 2000) 2451–2471
- [18] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* **abs/1406.1078** (2014)
- [19] Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* **abs/1412.3555** (2014)
- [20] Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML'15, JMLR.org* (2015) 2342–2350
- [21] le Cun, Y.: Generalization and network design strategies. Technical report, Department of Computer Science, University of Toronto (1989)
- [22] Walck, C.: Hand-book on statistical distributions for experimentalists. Technical report, Particle Physics Group, Fysikum, University of Stockholm (1996)

- [23] Sheskin, D.J.: Hand-book of Parametric and Nonparametric Statistical Procedures. 4 edn. Chapman & Hall/CRC (2007)
- [24] Soria, C.C.: ANAIS-0: Feasibility study for a 250 kg NaI(Tl) dark matter search experiment at the Canfranc Underground Laboratory. PhD thesis, Universidad de Zaragoza (2013)
- [25] Monge, M.A.O.: Design, scale-up and characterization of the data acquisition system for the ANAIS dark matter experiment. PhD thesis, Universidad de Zaragoza (2015)
- [26] Frank, R.J., Davey, N., Hunt, S.P.: Time series prediction and neural networks. *Journal of Intelligent and Robotic Systems* **31**(1-3) (2001) 91–103
- [27] Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR* **abs/1502.03167** (2015)
- [28] Gamboa, J.C.B.: Deep Learning for time-series analysis. *CoRR* **abs/1701.01887** (2017)
- [29] Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. *CoRR* **abs/1611.06455** (2016)
- [30] Breiman, L.: Random Forests. *Machine Learning* **45**(1) (Oct 2001) 5–32
- [31] Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8) (Aug 1998) 832–844
- [32] Liaw, A., Wiener, M.: Classification and Regression by Random Forests. **23** (11 2001)