



Universidad Internacional de la Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Redes neuronales y la transformada Wavelet aplicados a las curvas de luz de la misión Kepler

Trabajo Fin de Máster

presentado por: Xingqiang Chen

Dirigido por: Roberto Baena Gallé

Ciudad: Madrid

Fecha: 13 de febrero de 2023

Índice de Contenidos

| | |
|--|------------|
| Resumen | vi |
| Abstract | vii |
| 1 Introducción | 1 |
| 2 Contexto y estado del arte | 7 |
| 2.1 Marco teórico | 7 |
| 2.2 Métodos de detección de exoplanetas | 7 |
| 2.2.1 Velocidad radial | 7 |
| 2.2.2 Imagen directa | 8 |
| 2.2.3 Astrometría | 9 |
| 2.2.4 Microlente gravitatoria | 10 |
| 2.2.5 Método del tránsito | 11 |
| 2.3 Teoría de wavelets | 12 |
| 2.3.1 Introducción | 12 |
| 2.3.2 Funciones Wavelets | 13 |
| 2.3.3 Transformada Wavelet | 14 |
| 2.3.4 Transformada Wavelet continua (CoWT) | 15 |
| 2.3.5 Transformada Wavelet Discreta (DWT) | 16 |
| 2.4 Trabajos previos sobre la transformada wavelet y la detección de exoplanetas | 17 |
| 2.4.1 Wavelet aplicado a los coeficientes espectrales | 18 |
| 2.4.2 Transformada wavelet aplicado al filtrado de ruidos correlacionados con el tiempo en las curvas de luz | 19 |
| 2.4.3 Otros trabajos relacionados con la detección de exoplanetas mediante técnicas de IA | 21 |
| 3 Objetivos | 23 |
| 3.1 Objetivo General | 23 |
| 3.2 Objetivos específicos | 23 |
| 4 Desarrollo | 25 |
| 4.1 Base de datos de origen | 25 |

| | | |
|----------|---|-----------|
| 4.2 | Entorno de ejecución: Google Colab | 27 |
| 4.3 | Librerías usadas | 28 |
| 4.4 | Preprocesamiento de los datos | 32 |
| 4.4.1 | Filtrado inicial | 32 |
| 4.4.2 | Descarga y ajustes de los datos | 33 |
| 4.4.3 | Plegado par e impar de las curvas | 33 |
| 4.4.4 | Filtrado por número de puntos | 34 |
| 4.4.5 | Aplicación de la transformada wavelet para la descomposición de las curvas de luz | 36 |
| 4.4.6 | Obtención del dataframe final | 41 |
| 4.5 | Construcción de la red neuronal | 42 |
| 4.5.1 | Normalización de los datos | 42 |
| 4.5.2 | Formateo en array | 43 |
| 4.5.3 | Separación de datos | 44 |
| 4.5.4 | Formato de los resultados | 44 |
| 4.5.5 | Capas de las redes neuronales | 46 |
| 4.5.6 | Modelos de las redes neuronales | 49 |
| 5 | Resultados | 51 |
| 5.1 | Descripción de los resultados | 51 |
| 5.2 | Métricas de evaluación | 53 |
| 5.2.1 | Métricas empleadas | 53 |
| 5.2.2 | Métricas obtenidas | 54 |
| 5.3 | Ánalisis de conjuntos de curvas adicionales | 56 |
| 5.3.1 | Conjunto de curvas test | 56 |
| 5.3.2 | Predicción del conjunto de curvas candidatas | 56 |
| 6 | Conclusiones y líneas de trabajo futuros | 58 |
| A | Artículo científico | 63 |

Índice de Figuras

| | | |
|-----|--|----|
| 1.1 | Modelo heliocéntrico de Copérnico. <i>Fuente:</i> <i>De revolutionibus orbium coelestium</i> , 1543. | 1 |
| 1.2 | Diagrama Hertzsprung-Russell. <i>Fuente:</i> <i>sci.esa.int</i> | 2 |
| 1.3 | Recuento del número de exoplanetas descubiertos a fecha septiembre de 2022. <i>Fuente:</i> <i>Nasa Exoplanet Archive</i> | 3 |
| 1.4 | Comparación entre el Modelo de Kasting et al. (línea continua negra) y el Modelo de Temperatura Constante (línea continua roja). <i>Fuente:</i> <i>Determinación de la zona de habitabilidad. Denis Alexander Poffo, Universidad Nacional de Córdoba</i> | 4 |
| 1.5 | Zona de habitabilidad de una estrella (resaltada en el centro). <i>Fuente:</i> <i>National Geographic</i> | 5 |
| 1.6 | Espectro de luz obtenido la con espectroscopia de transmisión. <i>Fuente:</i> <i>Nasa Exoplanet Archive</i> | 5 |
| 2.1 | Efecto Doppler y el método de la velocidad radial <i>Fuente:</i> <i>European Southern Observatory</i> | 8 |
| 2.2 | Eclipse solar artificial empleando un coronógrafo y captado por el telescopio SOHO LASCO C2. <i>Fuente:</i> <i>Solar and Heliospheric Observatory, NASA</i> . . . | 9 |
| 2.3 | Ejemplo de un Starshade. <i>Fuente:</i> <i>ResearchGate</i> | 9 |
| 2.4 | Variaciones de la estrella Gliese 876 detectadas por el telescopio Hubble. <i>Fuente:</i> <i>Nasa Exoplanet Archive</i> | 10 |
| 2.5 | Efecto de microlente gravitacional para la detección de un exoplaneta. <i>Fuente:</i> <i>Nasa Exoplanet Archive</i> | 10 |
| 2.6 | Curvas de luz del exoplaneta confirmado por la misión Kepler con ID 10419211. <i>Fuente:</i> <i>Elaboración propia</i> | 11 |
| 2.7 | Curvas de luz del exoplaneta confirmado por la misión Kepler con ID 10337517. <i>Fuente:</i> <i>Elaboración propia</i> | 12 |
| 2.8 | Codificación por niveles y en sub-bandas. <i>Fuente:</i> <i>Sistema de reconocimiento de personas mediante su patrón de Iris Basado en la Transformada Wavelet. Rafael Coomonte, 2006</i> | 15 |
| 2.9 | Señal no estacionaria <i>Fuente:</i> <i>Sobre wavelets e imágenes. Universidad Tecnológica Nacional, 2006</i> | 16 |

| | | |
|------|---|----|
| 2.10 | CoWT aplicada a la señal no estacionaria de la Figura 2.9. <i>Fuente: Sobre wavelets e imágenes. Universidad Tecnológica Nacional, 2006.</i> | 16 |
| 2.11 | Descomposición de una imagen con la 2D-DWT <i>Fuente: González González, R. A. (2010). Algoritmo basado en Wavelets aplicado a la detección de incendios forestales. Universidad de las Américas Puebla.</i> | 17 |
| 2.12 | Detección automática de “planetas modelos” aplicando wavelet <i>Fuente: The Astrophysical Journal, 2004.</i> | 19 |
| 2.13 | Datos de series temporales de Spitzer aplicando wavelet. <i>Fuente: The American Astronomical Society, 2016.</i> | 20 |
| 2.14 | Frames, antes (arriba) y después (abajo) del filtrado de ruidos mediante la transformada wavelet continua, de la estrella de β Pictoris. <i>Fuente: Institute for Particle Physics and Astrophysics (ETH Zurich), 2021.</i> | 21 |
| 4.1 | Cronografía de las misiones espaciales realizadas desde 1990 por la National Aeronautics and Space Administration (NASA) y la European Space Agency (ESA) <i>Fuente: European Space Agency, 2022.</i> | 25 |
| 4.2 | Base de datos cumulativos de las curvas de luz obtenidos por la misión Kepler <i>Fuente: NASA Exoplanet Archive, 2022.</i> | 26 |
| 4.3 | Dataframe de las 2750 curvas de luz pertenecientes a estrellas del tipo espectral G <i>Fuente: Elaboración propia.</i> | 27 |
| 4.4 | Ejemplos de arrays de diferentes dimensiones: vector (izquierda), matriz (centro) y cubo (derecha). <i>Fuente: Aprende con Alf. Recursos Educativos Libres, 2022.</i> | 29 |
| 4.5 | Ejemplo de un dataframe y sus componentes. <i>Fuente: PYnative, Python Programming, 2021.</i> | 29 |
| 4.6 | Ejemplo del método de K-medias clustering aplicando Scikit-learn. <i>Fuente: scikit-learn.org, 2022.</i> | 31 |
| 4.7 | Ejemplo del método de evaluación mediante la curva ROC para un algoritmo de clasificación multiclas aplicando Scikit-learn, 2014. <i>Fuente: scikit-learn.org.</i> | 31 |
| 4.8 | Plegado par (amarillo) e impar (azul) de la curva de luz perteneciente al exoplaneta confirmado con Kepler ID 10717220 y un total de 25827 puntos. <i>Fuente: Elaboración propia.</i> | 34 |
| 4.9 | Código en Python que muestra los puntos que tienen las curvas con ID 10797460 (arriba) e ID 11446443 (abajo). <i>Fuente: Elaboración propia.</i> | 34 |
| 4.10 | Curvas de luz pertenecientes los exoplanetas confirmados con Kepler ID 10797460 (arriba) e ID 11446443 (abajo). El tránsito está resaltado en rojo. <i>Fuente: Elaboración propia.</i> | 35 |
| 4.11 | Funciones wavalet pertenecientes a la familia Symlets. <i>Fuente: Researchgate, 2014.</i> | 37 |

| | |
|---|----|
| 4.12 Los 7 niveles de descomposición correspondientes a las curvas de luz del exoplaneta confirmado con Kepler ID 2713049. <i>Fuente: Elaboración propia.</i> | 38 |
| 4.13 Los 7 niveles de descomposición wavelet y el número de puntos por nivel. <i>Fuente: Elaboración propia.</i> | 39 |
| 4.14 Los 7 niveles de descomposición wavelet una vez solucionado los dos problemas anteriores. <i>Fuente: Elaboración propia.</i> | 40 |
| 4.15 Primeras entradas pertenecientes a las curvas de nivel 3 tras aplicar la transformada wavelet. <i>Fuente: Elaboración propia.</i> | 41 |
| 4.16 Curvas de luz par e impar de nivel 3 normalizados. <i>Fuente: Elaboración propia.</i> | 42 |
| 4.17 Función en Python para concatenar la curva par e impar de una curva de luz en un mismo array . <i>Fuente: Elaboración propia.</i> | 43 |
| 4.18 Visualización de los datos en formato array . <i>Fuente: Elaboración propia.</i> | 43 |
| 4.19 Ejemplo de la operación de convolución de dos dimensiones con un kernel predefinido de tamaño 3x3 aplicado a una imagen de tamaño 7x6 y con stride = 1 para obtener una matriz final de dimensiones 4x5. <i>Fuente: Dive into Deep Learning, 2020.</i> | 46 |
| 4.20 Ejemplo de convolución después de aplicar padding . <i>Fuente: Rubén Rodríguez Abril, LMO, 2022.</i> | 47 |
| 4.21 Ejemplo de una red neuronal con capas densas en el que todas las neuronas de una capa están conectadas con todas las neuronas de la capa anterior. <i>Fuente: Datacamp, 2021.</i> | 47 |
| 4.22 Ejemplo de maxpooling de tamaño 2x2 aplicado a una matriz de 4x4 padding . <i>Fuente: Computer Science Wiki, 2018.</i> | 48 |
| 4.23 Una red neuronal sin dropout (izquierda) comparado con una red neuronal con dropout (derecha). <i>Fuente: Github, 2022.</i> | 48 |
| 4.24 Ejemplo de la operación de Flatten . <i>Fuente: Bootcamp AI, 2019.</i> | 48 |
| 5.1 Curvas de luz con la descomposición wavelet. A la izquierda exoplaneta confirmado (Kepler ID 3935914), en el centro falso positivo (Kepler ID 11913073) y a la derecha candidato por determinar (Kepler ID 10028127). <i>Fuente: Elaboración propia.</i> | 51 |
| 5.2 Resultado de la predicción de las 3 curva anteriores. <i>Fuente: Elaboración propia.</i> | 52 |
| 5.3 Curvas de accuracy (rojo) y loss (azul) para cada uno de los modelos. <i>Fuente: Elaboración propia.</i> | 55 |
| 5.4 Matriz de confusión de los resultados de las 100 curvas de test. <i>Fuente: Elaboración propia.</i> | 56 |
| 6.1 Curva de luz del falso positivo con Kepler ID 10419211 (a la izquierda) y exoplaneta confirmado con Kepler ID 7935997 (a la derecha). <i>Fuente: Elaboración propia.</i> | 59 |

Resumen

La principal pregunta que se cuestiona la humanidad, como especie, es sobre la existencia de vida inteligente extraterrestre. Para poder contestar a esta pregunta, es imprescindible saber si existen planetas habitables fuera de nuestro Sistema Solar: los exoplanetas.

En este trabajo vamos a analizar, usando un modelo que combina los resultados de 5 redes neuronales, las curvas de luz de pertenecientes a las estrellas del tipo G como el Sol captadas por la misión Kepler (2009 - 2018). Para obtener los conjuntos de datos, se ha aplicado la transformada Wavelet a cada una de las curvas. La precisión obtenida se sitúa en torno al 0,7 mientras que la pérdida se estabiliza alrededor del 0,6. Esto es debido a que algunas de las curvas de luz pertenecen a falsos positivos son muy similares a las curvas de luz de exoplanetas confirmadas. Los resultados van bien encaminados, pero están sujetos a posibles mejoras de cara a futuras líneas de trabajos futuros.

Palabras Clave: Astrofísica, exoplanetas, Wavelets, curvas de luz.

Abstract

The main question that humanity, as a species, is asking about the existence of intelligent life off Earth. In order to answer this question, it is essential to know if there are habitable planets outside our Solar System: the exoplanets.

In this work we are going to analyze, using a model that combines the results of 5 neural networks, the light curves of belonging to G-type stars like the Sun captured by the Kepler mission (2009 - 2018). To obtain the datasets, the Wavelet transform has been applied to each of the curves. The accuracy obtained is around 0.7 while the loss stabilizes around 0.6. This is because some of the light curves belonging to false positives are very similar to the light curves of confirmed exoplanets. The results are on the right track, but are subject to possible improvements for future lines of work.

Palabras Clave: Astrophysics, exoplanets, wavelets, light curves.

Capítulo 1

Introducción

¿Existe vida en otros planetas? El interés que tiene la comunidad científica por contestar esta pregunta ha aumentado considerablemente durante las últimas décadas debido a las nuevas herramientas surgidas por los importantes avances científicos en el área de la investigación espacial iniciado desde mediados del siglo pasado.

El Geocentrismo, creado en el siglo II a manos de Ptolomeo, en el que situaba la Tierra en el centro del Universo fue la teoría predominante hasta principios del siglo XVI cuando, a manos del astrónomo Nicolás Copérnico (Polonia, 1473-1543), se desarrolló la Teoría heliocéntrica. Esta teoría defendía que los planetas giraban alrededor del Sol en órbitas completamente circulares, mientras que el resto de las estrellas del firmamento estaban fijas en el cielo (Gómez-Martínez, 2016). Esto, aunque no cierto del todo, fue un gran avance en el ámbito científico.

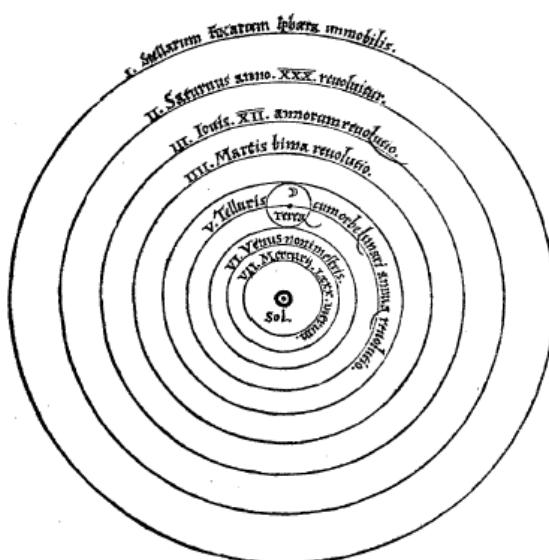


Figura 1.1: Modelo heliocéntrico de Copérnico. Fuente: *De revolutionibus orbium coelestium*, 1543.

Por otro lado, el astrónomo y teólogo Giordano Bruno (Italia, 1548-1600), dio un paso más sobre la Teoría heliocéntrica de Copérnico. Concretamente, defendía que el Sol no era más que una de las muchas estrellas que existían en el Universo y que, las otras estrellas, podrían tener sus propios planetas orbitando sobre ellas (Gaytán, 1997). Estos cuerpos celestes son lo que hoy en día se conoce como exoplanetas. Para que un objeto pueda considerarse como un planeta extrasolar o exoplaneta, es necesario que cumpla, según la Unión Astronómica Internacional, las siguientes tres características:

- La gravedad del objeto tiene que ser suficientemente grande para poder tener una forma esférica.
- Debe ser suficientemente grande como para poder despejar a otros objetos de tamaños similares de su órbita.
- Orbita alrededor de una estrella.

El primer descubrimiento de exoplanetas ocurrió en enero de 1992, cuando Aleksander Wolszczan y Dale Frail dieron con dos planetas rocosos orbitando alrededor de un púlsar (PSR B1 257+12) en la constelación de Virgo. Sin embargo, debido a la continua exposición a la radiación que sufrían estos dos exoplanetas, no era posible el desarrollo de cualquier forma de vida tal como se conoce hasta ahora (Walbolt, 2022).

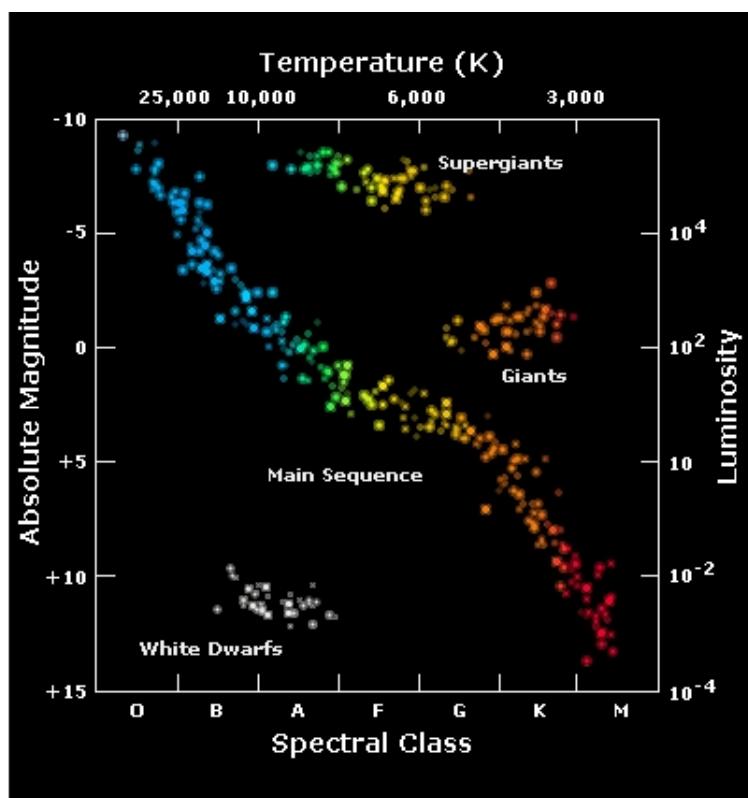


Figura 1.2: Diagrama Hertzsprung-Russell. *Fuente: sci.esa.int.*

Al año siguiente, se descubrió otro exoplaneta (PSR B1620-26 b), este orbitaba alrededor de un sistema binario compuesto por un púlsar y una enana blanca. Este cuerpo se encontraba a 1170 años luz de la Tierra y tenía aproximadamente 3300 veces su tamaño. Dos años más tarde, en 1995, se descubrió el primer planeta, bautizado como Dimidio, que orbitaba alrededor de una estrella de la secuencia principal, es decir, perteneciente a la región del diagrama de Hertzsprung-Russell en la que la mayoría de las estrellas usan el hidrógeno como su fuente principal de combustión (Walbolt, 2022) como se puede observar en la Figura 1.2.

A partir de entonces, el número de exoplanetas descubiertos fue en aumento hasta alcanzar, a fecha de septiembre de 2022, un total de 5171 exoplanetas confirmados y otros 5887 potenciales candidatos como se observa en la siguiente tabla:

Summary Counts

| | |
|--|------|
| All Exoplanets | 5171 |
| Confirmed Planets Discovered by Kepler | 2708 |
| Kepler Project Candidates Yet To Be Confirmed | 2056 |
| Confirmed Planets Discovered by K2 | 537 |
| K2 Candidates Yet To Be Confirmed | 969 |
| Confirmed Planets Discovered by TESS ¹ | 256 |
| TESS Project Candidates Integrated into Archive ² | 5887 |
| Current date TESS Project Candidates at ExoFOP | 5908 |
| TESS Project Candidates Yet To Be Confirmed ³ | 3941 |

¹ *Confirmed Planets Discovered by TESS* refers to the number planets that have been published in the refereed astronomical literature.

² *TESS Project Candidates* refers to the total number of transit-like events that appear to be astrophysical in origin, including false positives as identified by the TESS Project.

³ *TESS Project Candidates Yet To Be Confirmed* refers to the number of TESS Project Candidates that have not yet been dispositioned as a Confirmed Planet or False Positive.

Figura 1.3: Recuento del número de exoplanetas descubiertos a fecha septiembre de 2022.
Fuente: Nasa Exoplanet Archive.

La mayoría de estos exoplanetas se condensan en una región relativamente pequeña de la Vía Láctea. Según datos obtenidos por el telescopio espacial Kepler lanzado en 2009, se sabe que hay más exoplanetas que estrellas en nuestra galaxia (Brennan, 2021). Estos exoplanetas tienen composiciones relativamente similares a los planetas de nuestro sistema solar: algunos son ricos en gases como Júpiter o Saturno y otros, con una composición más rocosa, parecidos a los planetas interiores como la Tierra o Marte. Sin embargo, para dar el siguiente paso en la búsqueda de vida extraterrestre, se necesita analizar, de manera más detallada, diversos criterios para determinar las características de dichos cuerpos celestes.

Se conoce como la “zona habitable” de una estrella a la región alrededor de la misma en la que el agua puede mantenerse en estado líquido en su superficie durante un periodo de tiempo relativamente largo del orden de miles de millones de años. Normalmente, estas zonas están limitadas por una frontera interior y una frontera exterior.

Existen diferentes modelos como el de Kasting et al. (1993) para determinar los límites de la zona de habitabilidad estelar asumiendo condiciones extremas para el planeta. Otro de los modelos más destacados es el Modelo de Temperatura Constante, que calcula los límites de la zona de habitabilidad según las temperaturas de congelación y evaporación del agua del planeta (que depende de otros factores como la presión atmosférica). Aunque esta aproximación es más simple que la de Kasting et al, permite generalizar la estimación para estrellas de tipos espectrales entre F0 y K0 de forma razonablemente aceptable (Poffo, 2012). En la Figura 1.4 se puede ver una comparación entre ambos modelos.

Por ejemplo, en el caso del Sistema Solar, la zona habitable que se fijó inicialmente se extendía desde la órbita de Venus hasta la de Marte. Pero, actualmente se sabe que dicha zona podría ser mucho más grande o, incluso, haber múltiples zonas habitables para una misma estrella. Por ejemplo, la fuerza gravitatoria de planetas gigantes como Júpiter o Saturno, puede producir suficiente energía para calentar los núcleos de las lunas que orbitan sobre ellas. De manera que es posible desarrollar formas de vida capaces de sobrevivir en condiciones muy extremas, zonas en donde la luz de la estrella sea muy débil o lugares con temperaturas extremas como ocurre en algunas zonas de la propia Tierra.

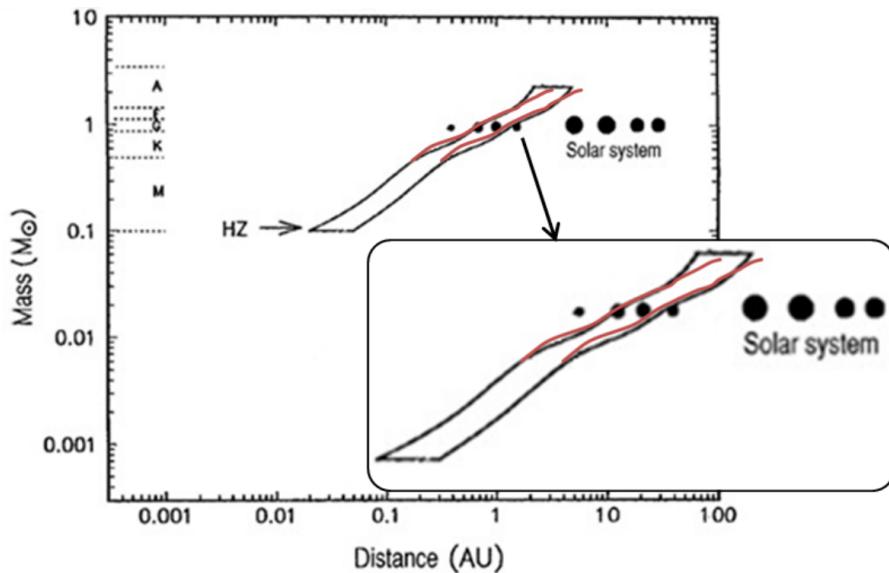


Figura 1.4: Comparación entre el Modelo de Kasting et al. (línea continua negra) y el Modelo de Temperatura Constante (línea continua roja). *Fuente: Determinación de la zona de habitabilidad. Denis Alexander Poffo, Universidad Nacional de Córdoba.*

En teoría, todas las estrellas deberían tener al menos una zona habitable (salvo las enanas marrones que no emiten suficiente energía). Por ejemplo, las estrellas del tipo solar tienen una esperanza de vida en torno a los diez mil millones de años mientras que las enanas rojas, que son las más longevas, pueden estar en su secuencia principal durante cientos o incluso miles de millones de años, lo cual es una gran ventaja para el desarrollo de vida. Además, se cree que el 85% de las estrellas de la galaxia son de este tipo (Poffo, 2012).

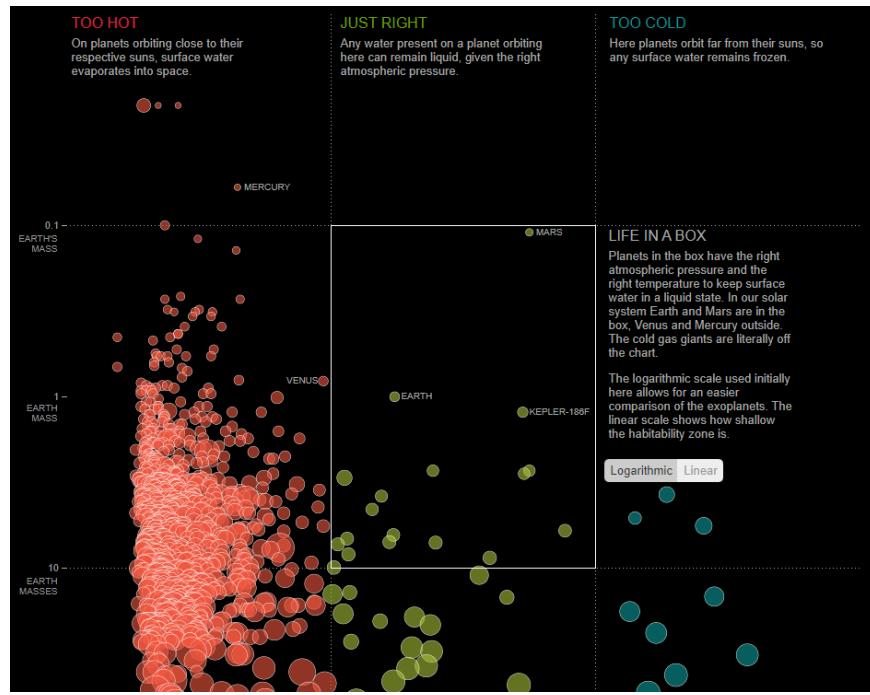


Figura 1.5: Zona de habitabilidad de una estrella (resaltada en el centro). *Fuente: National Geographic.*

Otros factores que influyen en el desarrollo de formas de vida son: la composición del planeta, el tamaño, los elementos retenidos en ella, el campo magnético o la composición atmosférica. Por ejemplo, para calcular la composición de la atmósfera se puede aplicar una técnica conocida como espectroscopía de transmisión en el que descompone la luz emitida por la estrella que pasa a través de la atmósfera del exoplaneta.

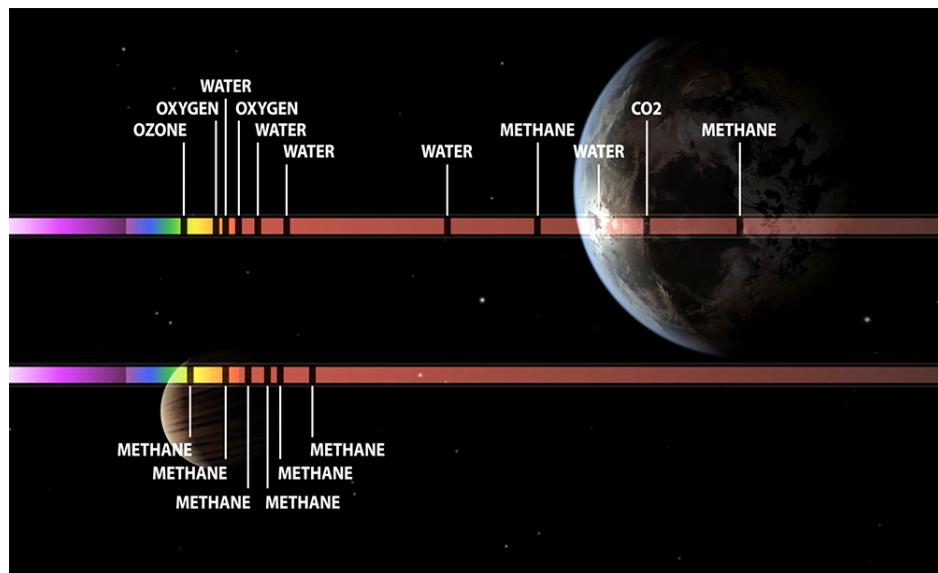


Figura 1.6: Espectro de luz obtenido mediante espectroscopía de transmisión. *Fuente: Nasa Exoplanet Archive.*

El espectro de luz obtenido puede indicar la composición de los gases de su atmósfera. Es decir, las franjas faltantes del espectro son los gases que componen dicha atmósfera. En la figura 1.6 se puede observar un ejemplo del espectro de luz con algunos de los elementos más importantes a tener en cuenta para el estudio de la composición.

Capítulo 2

Contexto y estado del arte

2.1 Marco teórico

Como se ha comentado en el capítulo anterior, la detección de exoplanetas es el primer paso para iniciar la búsqueda de vida fuera de la Tierra. Pero, debido a las escalas de distancias que se maneja en el ámbito de la exploración espacial, esta tarea no es en absoluto fácil. Ya desde tiempo inmemorables se ha especulado sobre la existencia de vida mucho más allá del límite de nuestros ojos, pero no fue hasta finales del siglo XX cuando realmente se dio el primer paso en la búsqueda de la misma. Gracias a la invención de nuevos instrumentos de alta precisión combinadas con avanzadas técnicas de inteligencia artificial se ha podido facilitar notablemente la correcta y eficaz detección de este tipo de cuerpos celestes.

2.2 Métodos de detección de exoplanetas

Actualmente existen numerosas técnicas que se usan para la detección de exoplanetas. La mayoría de ellas están basadas en la captación de variaciones inusuales en variables como la velocidad o la intensidad lumínica de la estrella sobre la que orbita el cuerpo celeste de interés. Con este tipo de análisis se consiguen resultados algo más difíciles de analizar e interpretar, pero pueden ser especialmente eficaces y precisas. También existen otras técnicas, más complejas a la hora de aplicarse, que pueden resultar muy interesantes. A continuación se detallan algunos de los principales métodos utilizados para el descubrimiento de los exoplanetas.

2.2.1 Velocidad radial

Este método se basa en la detección de leves variaciones en la velocidad de la estrella causadas por la fuerza gravitatoria que ejerce el posible exoplaneta sobre la propia estrella. A pesar de la gran diferencia de la masa entre ambos objetos, el cuerpo celeste en estudio ejerce su fuerza sobre la estrella provocándole una ligera perturbación con respecto al centro de masas del sistema en conjunto. Como esta variación es mínima y es muy difícil de detectar, este método se construye sobre el efecto Doppler.

Concretamente, aplicado al caso de las ondas luminosas, el efecto Doppler provoca ligeros desplazamientos en la longitud de onda del espectro aparente de la estrella cuando su velocidad varía. Por ejemplo, si un objeto luminoso se acerca al observador, la longitud de onda aparente de su luz disminuye, es decir, su frecuencia aumenta y su luz se desplaza hacia el color azul del espectro visible, mientras que si el objeto luminoso se aleja del observador, la longitud de onda aumenta y la luz se desplaza hacia el color rojo.

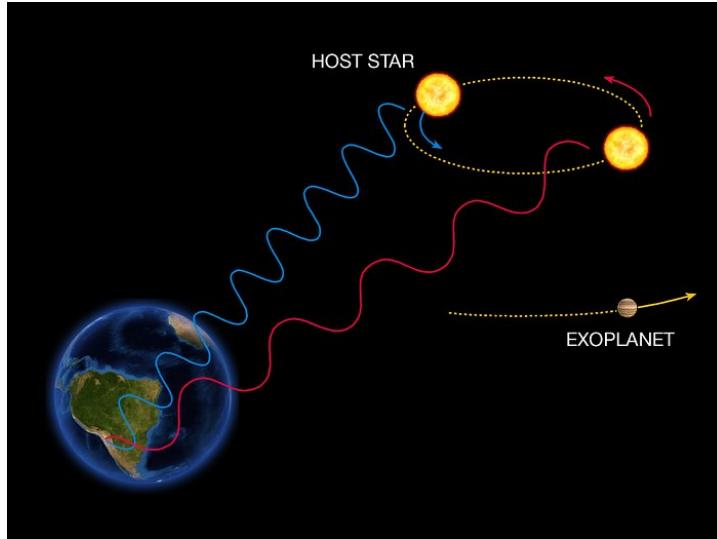


Figura 2.1: Efecto Doppler y el método de la velocidad radial *Fuente: European Southern Observatory.*

Por lo que, mediante el estudio de estas pequeñas variaciones en su longitud de onda a lo largo del tiempo, se puede determinar si existe una perturbación gravitatoria causada por otro cuerpo sobre la estrella. Sin embargo, estas alteraciones suelen ser casi imperceptibles salvo cuando el planeta provoca importantes perturbaciones gravitacionales. Por lo que este método solo es efectivo para aquellos cuerpos masivos de tipo gigante gaseoso, como Júpiter, que estén próximos a su estrella.

2.2.2 Imagen directa

Como su nombre indica, este método consiste en la observación directa de la luz visible o infrarroja que emiten las estrellas. Este método es el que más información puede proporcionar para el estudio. Sin embargo, también es el más difícil de aplicar pues la luz que refleja el objeto suele ser muy débil en comparación con la luz que emite la estrella sobre la que orbita. Por lo que, muchas veces, dicha luz reflejada es imperceptible. Aunque, durante los últimos años, se han desarrollado avanzadas técnicas en este campo que son capaces de bloquear gran parte de la luz de las estrellas anfitrionas (Sánchez, 2019). Entre las principales técnicas aplicables se pueden destacar:

1. Coronógrafos: Son dispositivos que se acoplan a los propios telescopios, estos bloquean

la luz emitida por un objeto central como una estrella y permite aumentar la visibilidad de los objetos cercanos a dicha estrella (ver Figura 2.2).

2. Starshade: se tratan de bloqueadores externos a los telescopios. Estos se sitúan a una distancia determinada y, ajustando al ángulo adecuado, bloquea la luz de la estrella antes de que lleguen al telescopio (ver Figura 2.3).

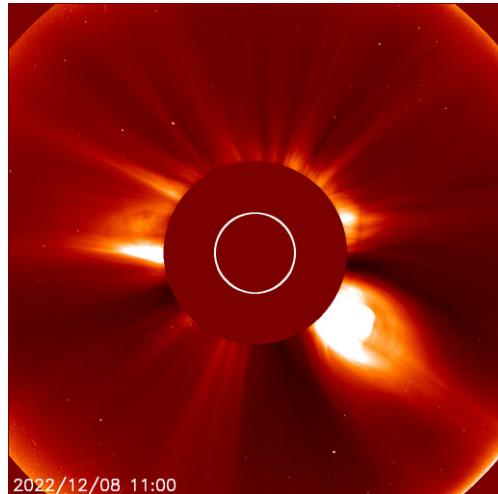


Figura 2.2: Eclipse solar artificial empleando un coronógrafo y captado por el telescopio SOHO LASCO C2. *Fuente: Solar and Heliospheric Observatory, NASA.*

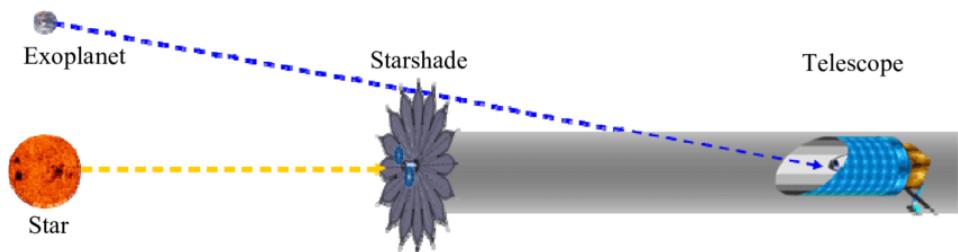


Figura 2.3: Ejemplo de un Starshade. *Fuente: ResearchGate.*

2.2.3 Astrometría

Dado que una estrella gira sobre su centro de masas, este método aprovecha las pequeñas variaciones que puede sufrir su posición tomando como referencia la posición aparente de la estrella con respecto a las otras estrellas lejanas del cielo. Como este método requiere unas mediciones muy exactas usando instrumentos muy precisos, entonces no es especialmente efectiva a fecha de hoy debido a las limitaciones técnicas. Las mediciones más precisas que se ha podido lograr con este método ha sido conseguido por el satélite Gaia de la ESA.

En la siguiente figura podemos observar las variaciones en la posición que ha experimentado la enana roja Gliese 876 perteneciente a la constelación de Acuario, situado a 15 años luz

de la Tierra.

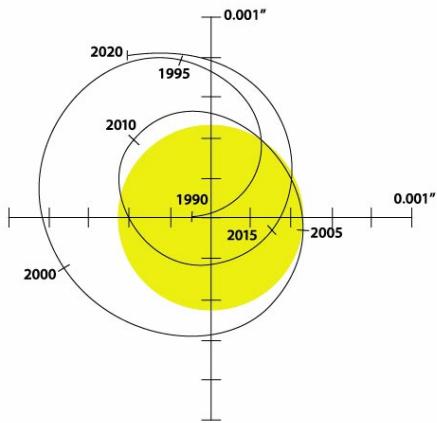


Figura 2.4: Variaciones de la estrella Gliese 876 detectadas por el telescopio Hubble.
Fuente: *Nasa Exoplanet Archive*.

2.2.4 Microlente gravitatoria

Este método aprovecha el efecto de lente gravitacional que provocan los objetos masivos para detectar la existencia de posibles exoplanetas. Cuando una estrella o un objeto que emite suficiente luz (por ejemplo, un cuásar) se alinea con un objeto con gran masa, este último puede curvar la luz que llega desde la primera estrella provocando una desviación en su trayectoria. En este caso, la microlente produce un efecto amortiguador entre la luz de la lente y los objetos medidos, revelándola o mejorando la captación de sus características como el brillo o el tamaño permitiendo el estudio de objetos tenues u oscuros, desde planetas hasta agujeros negros.

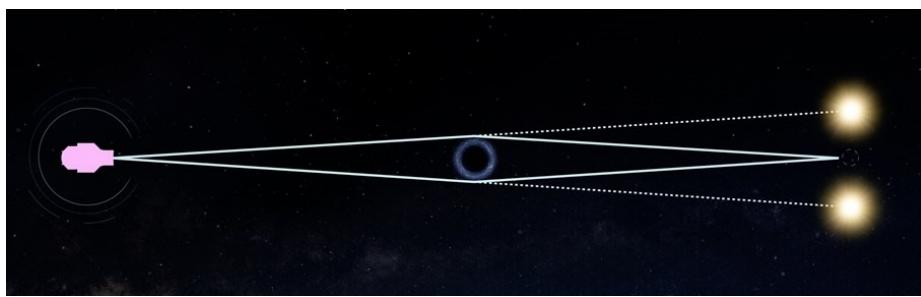


Figura 2.5: Efecto de microlente gravitacional para la detección de un exoplaneta. Fuente: *Nasa Exoplanet Archive*.

Sin embargo, en la práctica, este método es especialmente difícil de aplicar, ya que es complicado que se den todas las condiciones necesarias. Por lo tanto, generalmente se observan y se monitorizan millones de estrellas fuentes potenciales, normalmente de aquellas zonas del cielo con una densidad de estrellas grandes como es el centro de las galaxias durante periodos de días, meses o incluso años para detectar algunos pocos casos.

2.2.5 Método del tránsito

Cuando un cuerpo celeste pasa entre una estrella y el observador, la estrella sufre una disminución en la intensidad del flujo desde el punto de vista del observador. Este proceso se denomina tránsito y es la técnica que se ha empleado para obtener los datos de estudio de este trabajo.

Estos tránsitos son recogidos en las curvas de luz que se obtienen de la estrella y los objetos de interés. En nuestro caso, vamos a distinguir dos tipos de tránsitos:los pares y los impares (correspondientes a los eclipses alternos par e impar). Estos son utilizados en técnicas de detección automatizados para diferenciar los exoplanetas de los falsos positivos (Li, 2019). Es decir, cuando los tránsitos par e impar de un objeto muestran diferentes profundidades, entonces esto es suele indicar la existencia de dos estrellas en tránsito (Mallonn, 2022).

Si el objeto de estudio se tratase de un exoplaneta, con el tránsito se puede obtener información adicional muy relevante. Por ejemplo, con la siguiente ecuación se puede relacionar el flujo de una estrella con el radio del planeta:

$$\frac{\Delta F_*}{F_*} = \frac{R_p}{R_*} \quad (2.1)$$

Donde F_* es el flujo de la estrella, ΔF_* es la variación del flujo detectado durante el tránsito, R_p es el radio del planeta y R_* es el radio de la estrella.

Por otro lado, considerando una órbita circular, la distancia entre los centros de masa de ambos cuerpos en el punto más cercano de la órbita se calcula como:

$$d(\Phi = 0) = a \cdot \cos(i) \quad (2.2)$$

Donde Φ es la fase orbital en la que el planeta se encuentra más cerca del observador, d es la distancia entre ambos cuerpos, a la longitud del semieje mayor e i es la inclinación de la órbita.

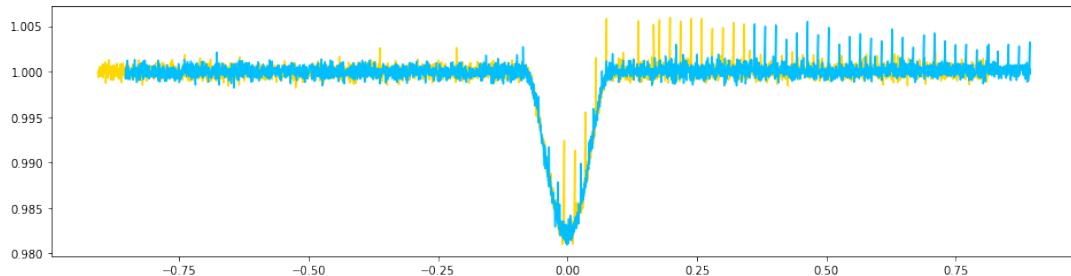


Figura 2.6: Curvas de luz del exoplaneta confirmado por la misión Kepler con ID 10419211.
Fuente: *Elaboración propia*.

En la figura 2.6 se puede observar la curva de luz del exoplaneta confirmado por la misión Kepler con ID 10419211 descompuesta con el Teorema de wavelet en curvas par (amarillo)

e impar (azul) correspondientes a los tránsitos par e impar respectivamente. En el centro de dichas curvas se puede ver un descenso notable en la intensidad del flujo de la estrella debido al tránsito del exoplaneta. Aunque muchas veces este tránsito no es tan trivial como se ha mostrado en la figura anterior. Por ejemplo, la siguiente figura muestra las curvas de luz del exoplaneta confirmado con kepler ID 10337517.

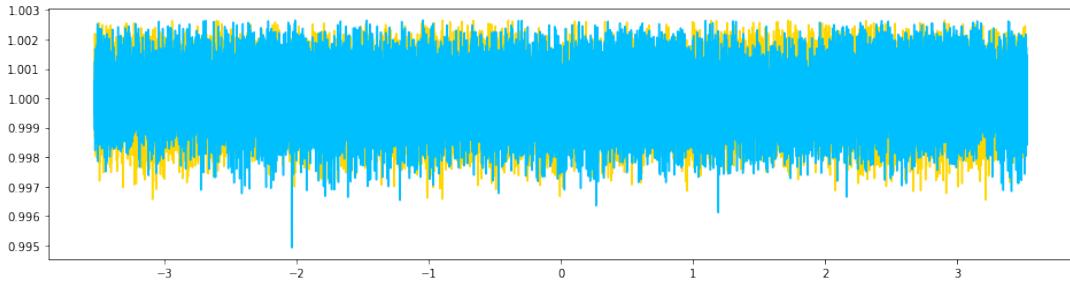


Figura 2.7: Curvas de luz del exoplaneta confirmado por la misión Kepler con ID 10337517.
Fuente: *Elaboración propia*.

Como se puede observar, no se puede apreciar ningún descenso notable del flujo por el tránsito. Por lo que, para poder obtener resultados más fácilmente analizables, se necesita depurar los datos obtenidos mediante técnicas específicas para ello. En el siguiente punto describiremos la técnica principal aplicada para el preprocesamiento de los datos.

2.3 Teoría de wavelets

2.3.1 Introducción

Se puede identificar una señal acústica constante y elemental mediante la frecuencia con la que vibra las partículas del aire que llega a nuestros oídos. Al ser la señal constante, es sencillo describir el movimiento de las partículas con una función trigonométrica como la que se define a continuación:

$$\sin(Kt + C) \quad (2.3)$$

Si la señal que se percibe se tratase de la superposición de varias señales elementales, entonces bastaría sumar las funciones trigonométricas de cada una de las señales elementales correspondientes. Esto es, en esencia, aplicar el análisis de Fourier. Sin embargo, los resultados son satisfactorios si y solo si las señales son invariantes con el tiempo. Sin embargo, como es evidente, esto no es aplicable a situaciones reales debido a la evidente mezcla de señales caóticas en nuestro entorno. Por lo que se necesita buscar herramientas más complejas para estudiar los diferentes tipos de señales que están presentes en nuestras vidas, sean sonoras o lumínicas.

Por ejemplo, estamos en una sala con mucha iluminación solar se enciende una luz. Y al

cabo de un rato, nos olvidamos totalmente de la bombilla. Y cuando de repente la bombilla se funde, nos damos cuenta de que estaba encendida. En este caso, el aviso que hemos recibido para darnos cuenta de ello es la transición entre los dos estados (el de encendido y el de apagado) y no por la propia luz. Con esto se pretende mostrar que, en algunas ocasiones, la información que se busca se encuentra en la variación de las frecuencias e intensidades.

Visto el ejemplo, se puede afirmar que, para estudiar las señales unidimensionales no estacionarias, se necesitan técnicas diferentes del análisis de Fourier. Técnicas que consiste en extraer los componentes simples de la señal, fundamentalmente de tipo tiempo-frecuencia o tiempo-escala. En el segundo caso, son conocidos como wavelets (Martín M., 2019).

El concepto de wavelet apareció en 1909 a manos de Alfred Haar. Sin embargo, esta wavelet se limitaba a aplicaciones concretas pues no era continua. Por lo que durante las siguientes décadas, concretamente entre los años 1930 y 1980, importantes científicos como Coifman, Goupillaud, Grossman, Levy, Marr, Morlet o Weis desarrollaron grandes avances sobre la teoría Wavelet. También hay que destacar que la década de 1980 fue de gran importancia para la evolución de las wavelets hasta lo que conocemos hoy en día (González G., 2010):

- En 1985, Stephane Mallat estableció la relación de los algoritmos piramidales, los filtros espejo de cuadratura y las bases orto-normales de las wavelets.
- Por otro lado, Yves Meyer, inspirado por el trabajo de Mallat, desarrolló las primeras wavelets continuas pero sin un soporte compacto.
- En 1988, Ingrid Daubechies publicó un conjunto de wavelets orto-normales que se convirtieron en el pilar de las aplicaciones wavelets.

2.3.2 Funciones Wavelets

Una de las principales características que destaca la teoría wavelet de los demás es su capacidad para realizar análisis en tiempo y frecuencia de señales tanto estacionarias como no estacionarias. Tanto la teoría de Fourier como el análisis wavelet se basa en la aproximación de señales mediante la superposición de señales elementales. La diferencia entre ambas radica en que las funciones wavelet varían tanto en frecuencia como en escala. Se puede ver las funciones wavelet como familias de funciones con una buena localización tanto en frecuencia como en tiempo.

Una wavelet es una señal oscilatoria, de corta duración, con una cantidad de energía finita y que se encuentra concentrada en un intervalo de tiempo determinado. Para que una función sea considerada como una wavelet $\psi(t)$, tiene que cumplir las siguientes condiciones:

1. Tiene que tener energía finita, es decir:

$$E = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty. \quad (2.4)$$

2. La función wavelet $\psi(t)$ tiene que cumplir el criterio de la constante de admisibilidad C_ψ :

$$C_\psi = \int_0^{\infty} \frac{|\psi(f)|^2}{f} df < \infty, \quad (2.5)$$

donde f es la frecuencia y $\psi(t)$ es la transformada de Fourier de la wavelet.

3. En el caso de las wavelets complejas, la transformada de Fourier $\psi(t)$ debe ser real y desvanecida para frecuencias negativas (González González, 2010).

2.3.3 Transformada Wavelet

La Transformada wavelet o WT genera, a partir de una función fija $\psi(t)$, bloques de información en escala y tiempo de una señal mediante operaciones de translación y dilatación como se muestra en la ecuación siguiente:

$$\psi_{a,b} = \frac{w\left(\frac{x-b}{a}\right)}{\sqrt{|a|}}; a, b \in \mathbb{R}, a \neq 0. \quad (2.6)$$

Donde a son las dilataciones y contracciones de la señal y b permite cambiar la posición de la señal en el tiempo.

El proceso de aplicar la transformada wavelet a una señal se le conoce como análisis mientras que el proceso inverso para reconstruir la señal a partir de la transformada se llama síntesis. El análisis de una señal mediante wavelets genera, dependiendo de las necesidades, diferentes niveles y diferentes sub-bandas. Dichas sub-bandas no son uniformes y se están divididas logarítmicamente como se muestra en la Figura 2.8.

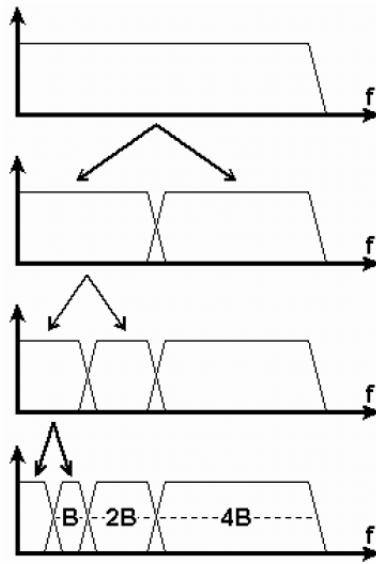


Figura 2.8: Codificación por niveles y en sub-bandas. *Fuente: Sistema de reconocimiento de personas mediante su patrón de Iris Basado en la Transformada Wavelet. Rafael Coomonte, 2006.*

Es importante destacar que el análisis WT, a diferencia del análisis de Fourier, proporciona una localización tiempo-frecuencia adaptiva. Es decir, a un nivel de escala grande se obtiene buena resolución en frecuencia mientras que a una escala baja se tiene una buena resolución en tiempo (González González, 2010).

2.3.4 Transformada Wavelet continua (CoWT)

Se define la CoWT como:

$$CoWT(b, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right)dt; a, b \in \mathbb{R}, a \neq 0. \quad (2.7)$$

donde $x(t)$ es la señal analizada, $\psi(t)$ es la wavelet madre (función prototipo que genera el conjunto de las funciones base en las transformadas wavelet), $a = f_0/f$ es el parámetro de dilatación, b es el parámetro de traslación y f_0 es la frecuencia central de la wavelet (González González, 2010).

En el ejemplo de la Figura 2.9 se puede observar una señal no estacionaria compuesta por 4 tipos de frecuencias: 5 Hz, 10 Hz, 20 Hz y 30 Hz. Por otro lado, en la Figura 2.10 se puede observar la transformada wavelet continua de la señal de la Figura 2.9. Los ejes corresponden a la transición y la escala. Este último, como ya se ha comentado en el punto anterior, se relaciona estrechamente con la frecuencia. Es decir, las escalas pequeñas se corresponden con frecuencias grandes y viceversa. Por otro lado, la traslación está estrechamente relacionada con el tiempo pues este indica la localización de la wavelet madre.

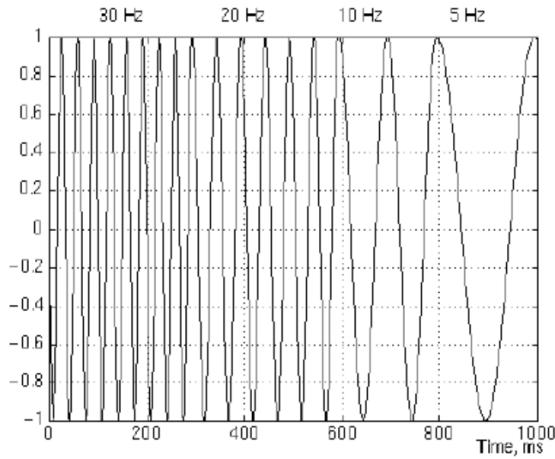


Figura 2.9: Señal no estacionaria *Fuente: Sobre wavelets e imágenes. Universidad Tecnológica Nacional, 2006.*

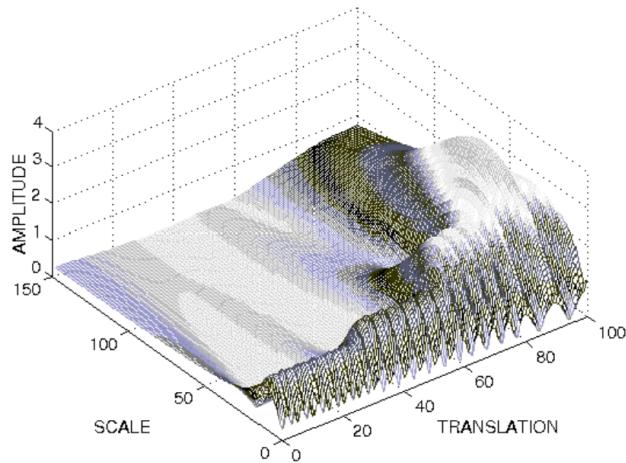


Figura 2.10: CoWT aplicada a la señal no estacionaria de la Figura 2.9. *Fuente: Sobre wavelets e imágenes. Universidad Tecnológica Nacional, 2006.*

2.3.5 Transformada Wavelet Discreta (DWT)

Una DWT se define como cualquier transformada wavelet en la que las wavelet son discretas o están discretizadas. En este caso se utiliza una familia de wavelets ortonormales (es decir, que las wavelets son ortogonales entre sí y todas tienen norma 1) definidas de la siguiente manera:

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k); j, k \in \mathbb{Z}. \quad (2.8)$$

donde j y k son parámetros que escalan (indica la anchura de la wavelet) y dilatan (determina la posición de la wavelet) la función ψ (función madre) para generar la familia de wavelets discretas.

Se define la función de escalamiento como:

$$\phi(t) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1} \psi(2t + k), \quad (2.9)$$

donde las c_k son los coeficientes wavelet. Esta función tiene como objetivo analizar el dominio de datos en diferentes resoluciones. Para simplificar los conceptos, se puede entender los c_k como filtros. Es decir, son matrices que se aplican a un vector de datos para realizar una transformación. Existen dos tipos de transformaciones: uno que actúa como un filtro de paso-bajo en el que atenúa los detalles y otro que actúa como un filtro de paso-alto en el que resaltan los detalles. Este concepto de análisis de una señal mediante filtros se conoce como descomposición de árbol de Mallat.

En la Figura 2.11 se muestra la descomposición de una imagen mediante la transformada wavelet Discreta de 2 dimensiones. Se puede observar la matriz de transformación LL en la parte superior izquierda, la submatriz LH de los detalles horizontales de la señal original, la submatriz HL correspondiente a los detalles verticales y la submatriz HH que son los detalles diagonales (González González, 2010).

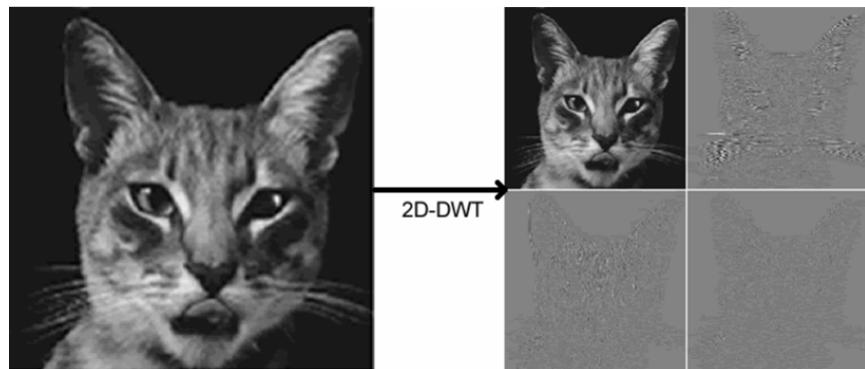


Figura 2.11: Descomposición de una imagen con la 2D-DWT *Fuente: González González, R. A. (2010). Algoritmo basado en Wavelets aplicado a la detección de incendios forestales. Universidad de las Américas Puebla.*

2.4 Trabajos previos sobre la transformada wavelet y la detección de exoplanetas

Como ya se ha comentado en las secciones anteriores, la transformada de wavelet es ampliamente utilizada en la actualidad debido a su efectividad. Por lo que no es de extrañar que existen numerosos trabajos de campo en el que se aplican estas técnicas. Por lo que en esta sección se comentarán algunos de los trabajos más destacados relacionados con el presente trabajo.

2.4.1 Wavelet aplicado a los coeficientes espectrales

En 2004 se publicó el artículo «Exoplanet recognition using a wavelet analysis technique» a manos de E. Masciadri y A. Raga. Se trata de uno de los primeros trabajos del milenio en el que aplica la transformada wavelet como técnica para la detección de exoplanetas. En dicho artículo explican las dificultades que se presentan en la detección de planetas extrasolares mediante técnicas como la de velocidad radial que hemos visto en el punto 2.2.1. Las complicaciones son provocadas, sobre todo, por la presencia de fotones y motas de polvo que existen entre el observador en la Tierra y objeto de interés. Cuanto más lejos se encontraba el objeto, más ruido aparecía. Por lo que dicha técnica, aun usando los telescopios más potentes del momento (de 8 a 10 metros de diámetro), solo era funcional para el estudio de objetos cercanos a la Tierra (del orden de 1 unidad astronómica).

Por otro lado, también hace referencia a otro artículo (Masciadri, 2004) sobre la posibilidad de detectar exoplanetas masivos (aproximadamente de 3 a 10 veces la masa de Júpiter) que orbiten alrededor de estrellas jóvenes (menores de 200 millones de años de edad). Sin embargo, este estudio sigue siendo extremadamente difícil debido, como ya se ha comentado anteriormente, a las interferencias que se produce. Por lo que sugiere la aplicación de las wavelets para poder automatizar este proceso de manera eficiente.

Hasta dicho momento, las wavelets ya fueron utilizadas en diversas aplicaciones como es el filtrado del ruido y la separación de fuentes de primer plano aplicado a datos obtenidos de la radiación de fondo de microondas (CMB) o el método clásico de la máxima entropía. Por lo que, siguiendo los resultados ya obtenidos en trabajos anteriores, las wavelets fueron de especial interés en dicho estudio debido a las ventajas que ofrecían frente al método del análisis de Fourier.

Para probar dicha capacidad, desarrollaron un procedimiento con la transformada wavelet aplicándolo sobre una imagen profunda obtenida con el Very Large Telescope (VLT) y el Sistema de Óptica Adaptativa¹ Nasmyth (NAOS), ambos pertenecientes a un programa de búsqueda de exoplanetas. Concretamente, parten de que un planeta se puede caracterizar por un conjunto de coeficientes espectrales:

$$a_i(x, y) \quad i = 1, 2, \dots$$

donde i es el tamaño característico de la wavelet y (x, y) indica la posición dentro de la imagen, medido en pixels.

Esta forma de identificación crea una especie de identidad para el planeta. En primer lugar, se calculan estos coeficientes para un “planeta modelo”. Posteriormente, se buscan otras características de la imagen profunda que tengan los mismos coeficientes espectrales.

¹Técnica que permite contrarrestar los efectos de la atmósfera para la captación de información en tiempo real

Y finalmente se aplica la transformada wavelet para intentar discriminar los “planetas modelos” y diferentes fuentes de ruidos.

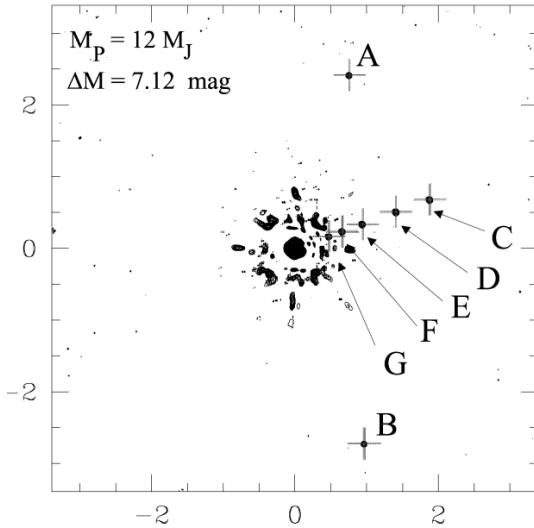


Figura 2.12: Detección automática de “planetas modelos” aplicando wavelet *Fuente: The Astrophysical Journal, 2004.*

Dicho estudio concluye que, al aplicar wavelet, se necesitan únicamente dos parámetros de entrada: el umbral de intensidad y la tolerancia. Este número es muy inferior con respecto a otras técnicas del momento por lo que resulta de especial interés cuando se disponen de datos limitados. Además, la técnica parece ser muy efectiva para aplicarlo en los sistemas binarios.

2.4.2 Transformada wavelet aplicado al filtrado de ruidos correlacionados con el tiempo en las curvas de luz

Durante las últimas dos décadas se han desarrollado gran cantidad de trabajos relacionados con la transformada wavelet aplicada al ámbito de astronomía. Entre los que se puede destacar, en especial, uno publicado en 2016. El trabajo titulado «On correlated-noise analyses applied to exoplanet light curves» fue escrito, en conjunto, por P. Cubillos, J. Harrington, T. J. Loredo, N. B. Lust, J. Blecic y M. Stemm. Dicho trabajo se centra en el análisis del ruido correlacionado con el tiempo, es decir, aquello que provoca alteraciones para la correcta medición de las datos a lo largo del tiempo. Este tipo de ruidos es de especial interés pues está muy presente en los datos que se obtienen de las curvas de luz de los exoplanetas y puede afectar tanto en la exactitud como en la precisión de los estimadores.

Además, el ruido puede proceder de diferentes fuentes: procedentes de los instrumentos de medida como los propios telescopios, provocados por variaciones de flujo estelar por fenómenos como las erupciones, causados por alteraciones de las condiciones climáticas

del lugar de observación o simplemente provocado por un exceso de datos. Por lo que una correcta filtración es esencial para interpretar de manera correcta la información que se obtiene de las mediciones. Por tanto, el pilar central del trabajo es el análisis de tres de los principales estimadores de ruido correlacionado más utilizado para la búsqueda de exoplanetas:

- **Promedio de tiempo:** se trata de un método que usan los propios n puntos de la curva de luz para realizar la estimación. Se considera al ruido como la suma de los cuadrados de dos componentes: una fuente incorrelada caracterizada por una desviación típica σ_w y una fuente correlada temporalmente caracterizada por σ_r . Entonces la incertidumbre de la medición se representa como:

$$\sigma_r = \sigma_d = \sqrt{\frac{\sigma_w^2}{n} + \sigma_r^2}$$

- **Permutación residual:** este método está inspirado en métodos de Bootstrapping no paramétricos. Usa datos muestreados para generar una distribución que se aproxime a la distribución de muestreo $p(y|\theta^*)$ para los parámetros θ^* . La idea es cambiar los datos mientras que preserva el orden del tiempo y, por lo tanto, preservando la estructura correlativa. Esta técnica se usa repetidamente para estimar las incertidumbres de los parámetros en el estudio de los exoplanetas.
- **Análisis wavelet:** esta técnica, desarrollada en 2009, modela el ruido correlacionado con el tiempo usando la transformada wavelet usando una base wavelet ortonormal en el que las variables fuera de la diagonal principal de la matriz de covarianza tienen poca importancia. Este método, ya comentado anteriormente, se basa en una serie de traslaciones y dilataciones sobre una wavelet madre.

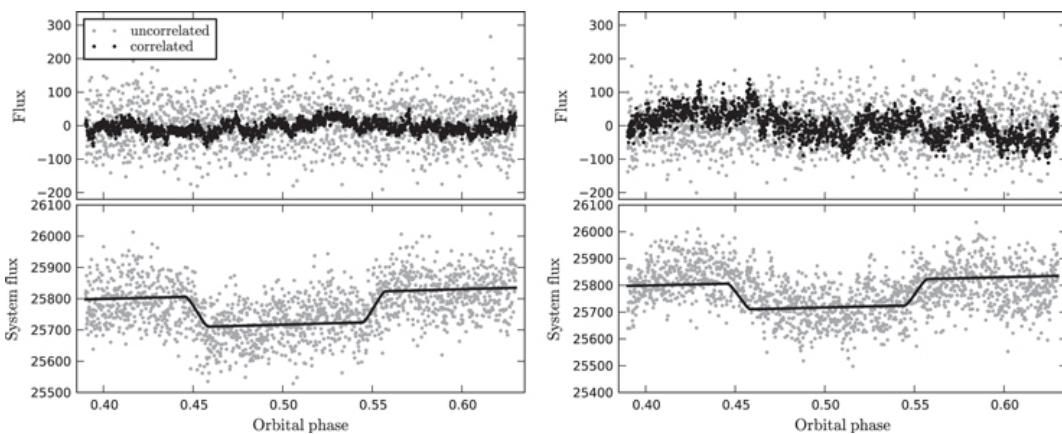


Figura 2.13: Datos de series temporales de Spitzer aplicando wavelet. *Fuente: The American Astronomical Society, 2016.*

Los resultados obtenidos en este estudio afirma que los 3 estimadores son eficientes, pero no son óptimos como muestran los resultados obtenidos. Pero, gracias al desarrollo de

avanzadas técnicas basados en los procesos gaussianos o el Análisis de Componentes Independientes, estos métodos están sujetos a notables mejoras futuras.

Por otro lado, en noviembre de 2021 se publicó un trabajo adicional sobre este último. Se trata del artículo «Wavelet based speckle suppression for exoplanet imaging. Application of a de-noising technique in the time domain» escrito por M. J. Bonse, S. P. Quanz y A. Amara. En dicho trabajo tiene como objetivo mejorar la relación señal-ruido (Signal and noise relation, SNR) en la detección de exoplanetas asistida por la óptica adaptativa. Concretamente, se ha aplicado la transformada wavelet al conjunto de datos obtenidos mediante imágenes directas en el dominio del tiempo.

Los resultados muestran que se puede obtener una notable mejora, de un 40% a un 60%, sobre la imagen original, dependiendo de la rotación de datos disponibles (a mayor cantidad de rotación, mejor supresión de ruidos). En la siguiente figura, se puede ver un ejemplo de la imagen directa de la estrella β Pictoris de la Constelación de Pictor antes y después de aplicar la transformada wavelet continua²:

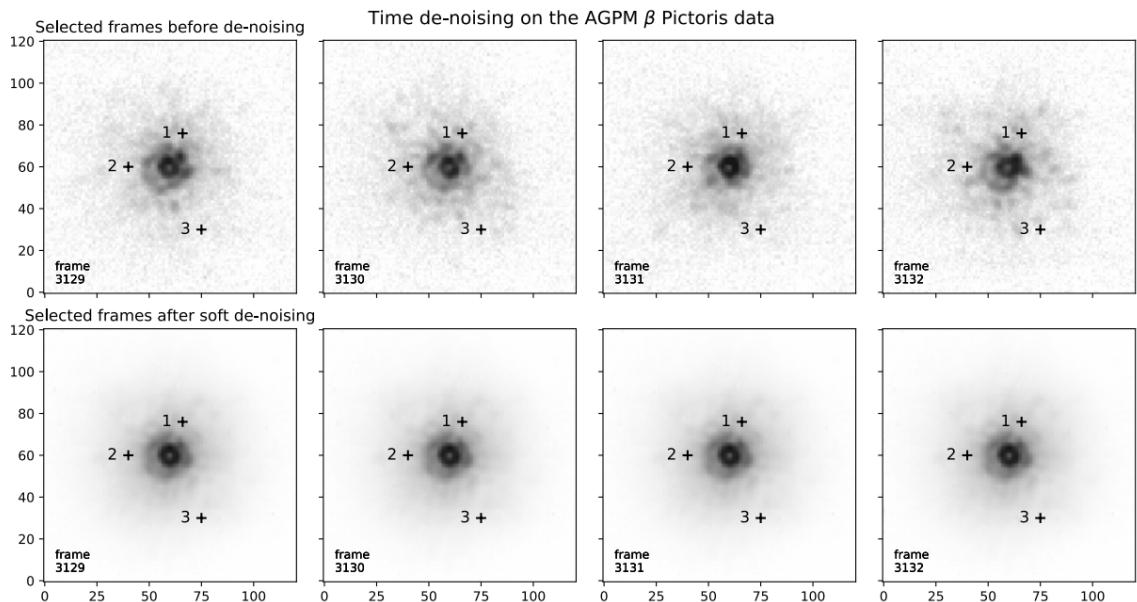


Figura 2.14: Frames, antes (arriba) y después (abajo) del filtrado de ruidos mediante la transformada wavelet continua, de la estrella de β Pictoris. Fuente: *Institute for Particle Physics and Astrophysics (ETH Zurich)*, 2021.

2.4.3 Otros trabajos relacionados con la detección de exoplanetas mediante técnicas de IA

Existen otros numerosos estudios relacionados con la detección de exoplanetas. Por ejemplo, el estudio titulado «Detección y caracterización de exoplanetas mediante el

²Ver el punto 2.3.4 para mayor detalle

método de los tránsito» realizado por el doctor Roi Alonso Sobrinos. En él estudió, mediante la técnica del tránsito, una zona concreta de la constelación de Lyra mediante el telescopio STARE durante un total de 49 noches. Como resultado, logró detectar un total de 16 candidatos a exoplaneta. De los cuales, 1 es el número de planetas confirmados (TrES-1). El estudio concluye resaltando la necesidad de mayor estudio de campo y el desarrollo de nuevas técnicas más eficaces para el futuro.

Otro de los trabajos que se puede destacar, sobre el que se acaba de mencionar, es el trabajo publicado por Raúl García Crespo de la Universidad Politécnica de Valencia (Crespo, 2021) en el que estudia los datos obtenidos por mediciones de flujo de las estrellas obtenidos por la misión Kepler y aplicando técnicas de data generator como el data augmentation. En dicho estudio, se han analizado diversas técnicas de redes neuronales para analizar su eficacia (LSTM, convolucional 1D y convolucional 2D). Y como resultado, se ha obtenido que las redes convoluciones funciona notablemente mejor (entre 82% a 93% de accuracy y 70.36% y 96% de Recall) que las Long-Short Term Memory (entre 65.82% a 69.71% de accuracy y 32.32% y 63.29% de Recall).

Capítulo 3

Objetivos

3.1 Objetivo General

El objetivo general de este trabajo es construir una red neuronal capaz de automatizar el proceso de análisis de las curvas de luz pertenecientes a las estrellas del tipo G, obtenidas por la misión Kepler, para determinar si se corresponden con exoplanetas o se tratan de falsos positivos.

3.2 Objetivos específicos

Para cumplir con el objetivo general, se han establecido los siguientes objetivos específicos:

- Análisis del estado del arte en el campo de la detección de exoplanetas aplicando la transformada Wavelet sobre las curvas de luz. De esta manera, se obtendrá una visión global sobre el enfoque del estudio y las posibles líneas de avance.
- Creación de un dataset de las curvas de luz pertenecientes a las estrellas del tipo G (como el Sol) adecuados para aplicar la transformada wavelet. Conociendo el tipo de datos disponibles, podremos entender mejor las características comunes que tienen dichas curvas de luz y optimizar los resultados.
- Preprocesamiento del dataset inicial mediante scripts en Python para obtener los 5 datasets finales de curvas de luz a ser analizados. De esta manera, unificaremos el criterio sobre las características a analizar para cada tipo de curva de cada red neuronal.
- Desarrollo de 5 modelos de redes neuronales que analizarán los 5 tipos de curvas de luz disponibles tras aplicar la transformada wavelet a cada una de las curvas del dataset inicial. Así obtendremos 5 resultados independientes para cada una de dichas curvas para ser considerado en el análisis final.
- Fijar los criterios de análisis para los 5 modelos anteriores para obtener un valor de probabilidad para considerar si un objeto es exoplaneta o no. De forma que

obtendremos resultados lo más precisos posibles teniendo en cuenta el output de cada uno de los modelos.

- Creación de un modelo final que unifican las 5 redes anteriores siguiendo los criterios establecidos. Así unificaremos los resultados iniciales para obtener un valor unificado que sea lo más representativo posible.
- Establecer métricas de evaluación para analizar los resultados obtenidos y las posibles líneas de mejora de cara a futuros trabajos.

Capítulo 4

Desarrollo

4.1 Base de datos de origen

Desde el descubrimiento de los primeros exoplanetas en la década de 1990, realizado por observatorios terrestres, el interés por la búsqueda de otras formas de vida fuera de la Tierra aumentó considerablemente. Tal fue que, desde entonces, entre la NASA y la ESA se diseñaron e implementaron gradualmente un total de 11 misiones involucrando a telescopios espaciales como se observa en la siguiente figura:

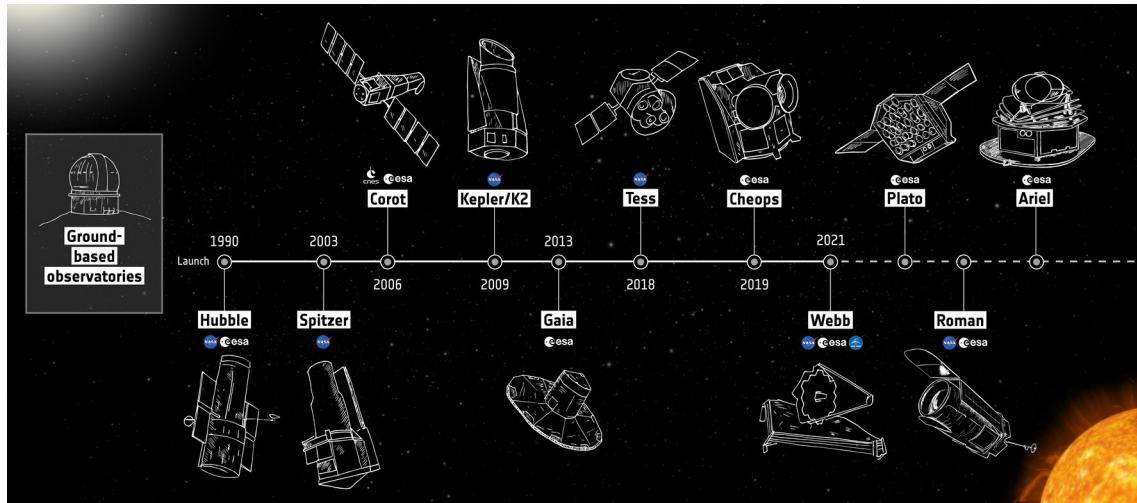


Figura 4.1: Cronografía de las misiones espaciales realizadas desde 1990 por la National Aeronautics and Space Administration (NASA) y la European Space Agency (ESA)
Fuente: European Space Agency, 2022.

Dentro de estas misiones, la de Kepler (2009 - 2018) fue la principal herramienta en el descubrimiento de exoplanetas, ocupando cerca del 52% de los exoplanetas confirmados hasta septiembre de 2022 (ver Figura 1.3). Un análisis posterior de los resultados estima que entre el 20% y el 50% de las estrellas visibles en el cielo podrían poseer posibles planetas rocosos similares a la Tierra ubicadas dentro de la zona habitable (Cofield, 2018).

Los datos utilizados en el presente estudio son los datos cumulativos almacenados en NASA Exoplanet Archive¹ que se han obtenido en la misión de Kepler comentado anteriormente.

| KepID | Kepler Name | Exoplanet Archive Disposition | Orbital Period [days] | Transit Epoch [BJD] | Stellar Effective Temperature [K] | Stellar Radius [Solar radii] |
|----------|--------------|-------------------------------|-----------------------|---------------------|--------------------------------------|---|
| 10797460 | Kepler-227 b | CONFIRMED | 9.48803557±2.775e-05 | 170.53875±0.00216 | 5455±81 | 0.927 ^{+0.105} _{-0.081} |
| 10797460 | Kepler-227 c | CONFIRMED | 54.4183827±0.0002479 | 162.51384±0.00352 | 5455±81 | 0.927 ^{+0.105} _{-0.081} |
| 10811496 | | CANDIDATE | 19.89913995±1.494e-0 | 175.850252±0.00058 | 5853 ⁺¹⁵⁸ ₋₁₇₈ | 0.868 ^{+0.233} _{-0.078} |
| 10848459 | | FALSE POSITIVE | 1.736952453±2.63e-07 | 170.307565±0.00011! | 5805 ⁺¹⁵⁷ ₋₁₇₄ | 0.791 ^{+0.201} _{-0.087} |
| 10854555 | Kepler-664 b | CONFIRMED | 2.525591777±3.761e-0 | 171.59555±0.00113 | 6031 ⁺¹⁶⁹ ₋₂₁₁ | 1.046 ^{+0.334} _{-0.133} |
| 10872983 | Kepler-228 d | CONFIRMED | 11.09432054±2.036e-0 | 171.20116±0.00141 | 6046 ⁺¹⁸⁹ ₋₂₃₂ | 0.972 ^{+0.315} _{-0.105} |
| 10872983 | Kepler-228 c | CONFIRMED | 4.13443512±1.046e-05 | 172.97937±0.0019 | 6046 ⁺¹⁸⁹ ₋₂₃₂ | 0.972 ^{+0.315} _{-0.105} |
| 10872983 | Kepler-228 b | CONFIRMED | 2.56658897±1.781e-05 | 179.55437±0.00461 | 6046 ⁺¹⁸⁹ ₋₂₃₂ | 0.972 ^{+0.315} _{-0.105} |
| 6721123 | | FALSE POSITIVE | 7.36178958±2.128e-05 | 132.25053±0.00253 | 6227 ⁺¹¹¹ ₋₁₂₄ | 1.958 ^{+0.322} _{-0.483} |
| 10910878 | Kepler-229 c | CONFIRMED | 16.06864674±1.088e-0 | 173.621937±0.00051 | 5031 ⁺⁷⁵ ₋₈₃ | 0.848 ^{+0.033} _{-0.072} |
| 11446443 | Kepler-1 b | CONFIRMED | 2.470613377±2.7e-08 | 122.763305±8.7e-06 | 5820±78 | 0.964±0.038 |
| 10666592 | Kepler-2 b | CONFIRMED | 2.204735417±4.3e-08 | 121.3585417±1.6e-05 | 6440 ⁺⁷⁶ ₋₈₉ | 1.952 ^{+0.099} _{-0.11} |
| 6922244 | Kepler-8 b | CONFIRMED | 3.522498429±1.98e-07 | 121.1194228±4.71e-0 | 6225 ⁺¹¹² ₋₁₃₇ | 1.451±0.11 |
| 10984090 | Kepler-466 c | CONFIRMED | 3.709214104±6.536e-0 | 133.98318±0.00143 | 5833 ⁺¹⁰⁵ ₋₁₁₇ | 1.022 ^{+0.143} _{-0.107} |

Figura 4.2: Base de datos cumulativos de las curvas de luz obtenidos por la misión Kepler
Fuente: *NASA Exoplanet Archive, 2022*.

En la Figura 4.2 se observa la tabla de la base de datos de NASA Exoplanet Archive en el que se visualiza una parte de la información que se ha obtenido mediante la misión Kepler. Las principales columnas de interés de dicha base de datos son:

- **kepid**: es el número de identificación del objeto compuesto por 8 cifras, único por objeto.
- **koi_disposition**: informa si el objeto se trata de falso positivo, exoplaneta confirmado o candidato por determinar.
- **period**: informa el periodo del tránsito del objeto en días.
- **epoch**: tiempo correspondiente al centro del primer tránsito detectado medido en Fecha Juliana Baricéntrica (BJD) restando un desplazamiento constante de 2.454.833 días correspondiente al 1 de enero de 2009.
- **koi_steff**: temperatura superficial de la estrella.
- **koi_srad**: el radio de la estrella medido en unidades de radio solar.

Por otro lado, se ha empleado una serie de filtros para conservar únicamente aquellos objetos cuya estrella sobre la que orbita es de tipo espectral *G* de la secuencia principal. Es decir, aquellas estrellas cuyo proceso principal es la fusión del Hidrógeno en Helio y que cumplen las siguientes características:

¹exoplanetarchive.ipac.caltech.edu

- La temperatura superficial que oscila entre 5200K y 6000K.
- La masa de la estrella oscila entre 0.8 y 1.2 veces la masa Solar.

Como resultado final, después de aplicar los filtros mencionados anteriormente, se han obtenido un total de 2750 curvas de luz como se muestra en la siguiente figura:

| | rowid | kepid | kepoi_name | kepler_name | koi_disposition |
|-------------------------|-------|----------|------------|--------------|-----------------|
| 0 | 1 | 10797460 | K00752.01 | Kepler-227 b | CONFIRMED |
| 1 | 2 | 10797460 | K00752.02 | Kepler-227 c | CONFIRMED |
| 2 | 3 | 10811496 | K00753.01 | NaN | CANDIDATE |
| 3 | 11 | 11446443 | K00001.01 | Kepler-1 b | CONFIRMED |
| 4 | 14 | 10984090 | K00112.02 | Kepler-466 c | CONFIRMED |
| ... | ... | ... | ... | ... | ... |
| 2745 | 9536 | 12117215 | K08296.01 | NaN | FALSE POSITIVE |
| 2746 | 9550 | 4645492 | K08095.01 | NaN | FALSE POSITIVE |
| 2747 | 9553 | 10028127 | K08193.01 | NaN | CANDIDATE |
| 2748 | 9559 | 10031643 | K07984.01 | NaN | FALSE POSITIVE |
| 2749 | 9560 | 10090151 | K07985.01 | NaN | FALSE POSITIVE |
| 2750 rows × 141 columns | | | | | |

Figura 4.3: Dataframe de las 2750 curvas de luz pertenecientes a estrellas del tipo espectral G
Fuente: *Elaboración propia*.

Estos 2750 objetos se clasifican en 3 grupos diferentes:

- **CONFIRMED**: son aquellos objetos que, mediante el análisis detallado con técnicas fuera del alcance del presente trabajo, han sido confirmados como exoplanetas.
- **FALSE POSITIVE**: corresponden a aquellos que, tras el análisis en detalle, finalmente han sido descartados como exoplanetas.
- **CANDIDATE**: son casos que aún está por determinar.

Los dos primeros grupos (confirmed y false positive) son los que se emplearán como datos de entrenamiento y test para elaborar el modelo de redes neuronales. Mientras que los objetos del tercer grupo (candidate) son los que se intentarán clasificar una vez terminada la red neuronal.

4.2 Entorno de ejecución: Google Colab

Colab es un servicio cloud que se basa en Jupyter Notebook. Permite utilizar aceleradores GPU² y TPU³ de Google y es compatible con múltiples librerías de IA como Scikit-

²Una GPU o Graphics Processing Unit es un coprocesador que proporcionan un alto rendimiento en operaciones con números con alta precisión.

³Una TPU o Tensor Processing Unit se focaliza en operaciones vectoriales y operaciones matriciales con números con baja precisión por lo que resulta especialmente útil en productos matriciales grandes.

learn, PyTorch, TensorFlow, Keras o OpenCV. Es la principal herramienta utilizada en el presente trabajo.

Una primera versión de Colab surgió en 2014 a manos de Google Research en colaboración con el equipo de desarrollo de Jupyter. Desde entonces, fue evolucionando dependiendo del uso y las necesidades que tuvo. Se centra en la compatibilidad con el lenguaje Python y está adaptado para realizar tareas de análisis de datos y aprendizaje automático.

Existe una versión gratuita con acceso a recursos limitados y dos versiones de pago con mayor recurso. En el presente trabajo se ha empleado una versión de Colab de pago con las siguientes características:

- **GPU:** GPU Nvidia T4 Tensor Core o GPU Nvidia V100 según disponibilidad.
- **TPU:** 8 cores y 1 worker.
- **RAM:** 25.5 GB (si se usa GPU) y 35.2 GB (si se usa TPU).
- **Memoria disco:** 166.8 GB (si se usa GPU) y 225.8 GB (si se usa TPU).
- **Tiempo de ejecución máxima:** 24 horas de forma ininterrumpida.

Como los datos que se manejan en este trabajo necesita un alto grado de precisión en sus operaciones (del orden de 10^{-8} para el flujo una vez normalizados los datos), entonces, finalmente se ha optado por el uso del acelerador GPU frente al TPU por las características que ofrece.

4.3 Librerías usadas

Dado el alto grado de desarrollo de los modelos de machine learning, actualmente, existen grandes cantidades de recursos disponibles para simplificar el trabajo de desarrollo. Estos recursos fueron desarrollados a medida que surgían las nuevas necesidades en la evolución del campo. Las principales librerías que se han utilizado en este trabajo son:

- **Numpy:** se trata de un módulo de extensión para Python escrito, en su mayor parte, en el lenguaje C. Esta librería introduce una nueva clase de objetos llamados **arrays**. Un **array** es una estructura que se compone de datos del mismo tipo. Estos datos pueden formar tablas de distintas dimensiones como muestra la siguiente figura:

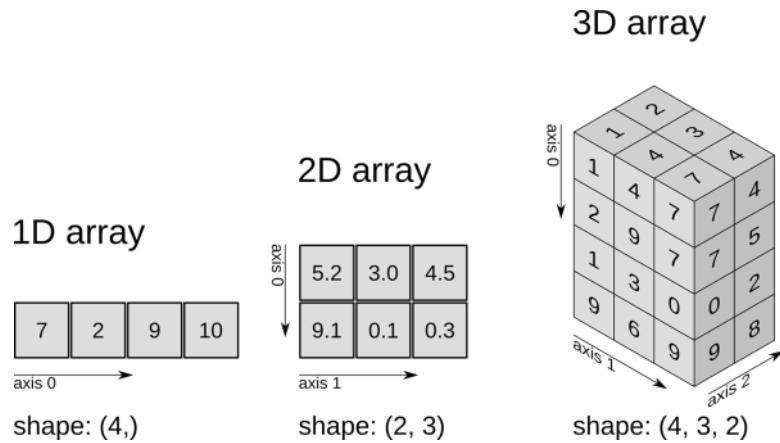


Figura 4.4: Ejemplos de **arrays** de diferentes dimensiones: vector (izquierda), matriz (centro) y cubo (derecha). *Fuente: Aprende con Alf. Recursos Educativos Libres, 2022.*

También incorpora una serie de operaciones algebraicas básicas para el tratamiento de estas matrices multidimensionales. La principal ventaja de esta librería es que la velocidad de procesamiento de los arrays es muy superior a la de las listas predefinidas en Python. Por lo que es ideal para tratar con datos de dimensiones grandes como es el caso de los modelos de redes neuronales.

- **Pandas:** es un paquete de Python creado en 2008. Está enfocado especialmente para el área del análisis de datos y aprendizaje automático, pues proporciona una estructura muy potente y flexible para tareas como el análisis, el modelado, la manipulación y el tratamiento de los datos.

| | | Column Label/ Header | | | | |
|---|----|----------------------|-----|-------|-------|----------|
| | | 0 | 1 | 2 | 3 | 4 |
| | | Name | Age | Marks | Grade | Hobby |
| 0 | S1 | Joe | 20 | 85.10 | A | Swimming |
| 1 | S2 | Nat | 21 | 77.80 | B | Reading |
| 2 | S3 | Harry | 19 | 91.54 | A | Music |
| 3 | S4 | Sam | 20 | 88.78 | A | Painting |
| 4 | S5 | Monica | 22 | 60.55 | B | Dancing |

Annotations explaining components:

- Index Label:** Points to the column index 0.
- Column Index:** Points to the header "Grade".
- Row Index:** Points to the row index 0.
- Column:** Points to the column "Marks".
- Element/ Value/ Entry:** Points to the value "85.10".

Figura 4.5: Ejemplo de un dataframe y sus componentes. *Fuente: PYnative, Python Programming, 2021.*

Pandas esta construido sobre Numpy, particularmente sobre los **arrays**, añadiendo nuevas funcionalidades sobre este último. Por ejemplo, facilita enormemente el trabajo con ficheros CSV, XLSX o bases de datos SQL. Permite acceder a los datos por el índices o por el nombre de las filas y columnas. También ofrece la posibilidad de reordenar, dividir y combinar conjuntos de datos o la manipulación de series temporales.

En la Figura 4.5 podemos ver un ejemplo de un dataframe (2D) usando Pandas. En ella podemos localizar cualquier elemento por filas, columnas o índices. Además de realizar cualquier tipo de operaciones básicas de las matrices.

- **Lightkurve:** se trata de un paquete de Python creado específicamente para facilitar el análisis de datos de series temporales relacionados con el brillo de objetos como planetas, estrellas o, incluso, galaxias. Particularmente, se enfoca para ofrecer soporte a los datos obtenidos por los telescopios espaciales *Kepler* y *TESS* de la NASA, pero también se puede aplicar para datos de curvas de luz obtenidos con cualquier otro telescopio. Incluye operaciones como el plegado, la agrupación o el trazado de las curvas de luz. En las figuras 2.6 y 2.7 podemos ver el trazado de dos curvas de luz una vez plegados en par e impar.
- **PyWavelets:** es una biblioteca de Python que integra las transformadas Wavelet para facilitar su uso en el análisis de datos de seires temporales. Incluye operaciones como Transformadas Wavelet Continuas (ver 2.3.4), Transformadas Wavelet Discretas (ver 2.3.5), filtros wavelet, cálculos de precisión simple y doble u operaciones con números complejos.
- **Tensorflow:** se trata de una de las principales bibliotecas dedicada al desarrollo de modelos de aprendizaje automático. Fue desarrollado por Google Brain en 2015 e integra diversas funciones dedicadas para la clasificación o el reconocimiento de imágenes, la incrustación de palabras, las redes neuronales, la traducción automática o simulaciones basadas en ecuaciones diferenciales parciales.

La gran ventaja de Tensorflow es que se puede ejecutar en prácticamente cualquier entorno: máquinas locales, en la nube o en dispositivos Android e IOS. Además, si se ejecuta en el propio entorno de Google (como por ejemplo, Google Colab), entonces se puede emplear los aceleradores TPU mencionados anteriormente.

- **Scikit-learn:** es una de las librerías básicas de Python para el machine learning. Fue creada en 2009 e incluye diversas herramientas como, por ejemplo, algoritmos de regresión, clasificación, clustering o reducción de dimensionalidad.

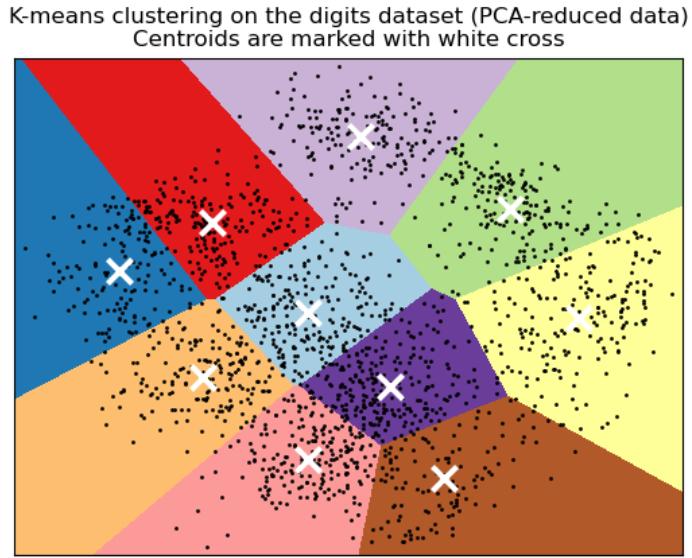


Figura 4.6: Ejemplo del método de K-medias clustering aplicando Scikit-learn. *Fuente: scikit-learn.org, 2022.*

Además, cuenta con diversas métricas de evaluación para los modelos de aprendizaje automático. Tales como RMSE, MAE y MSE para los modelos de regresión, la matriz de confusión y el accuracy para los modelos de clasificación, o la Silhouette para los algoritmos de partición.

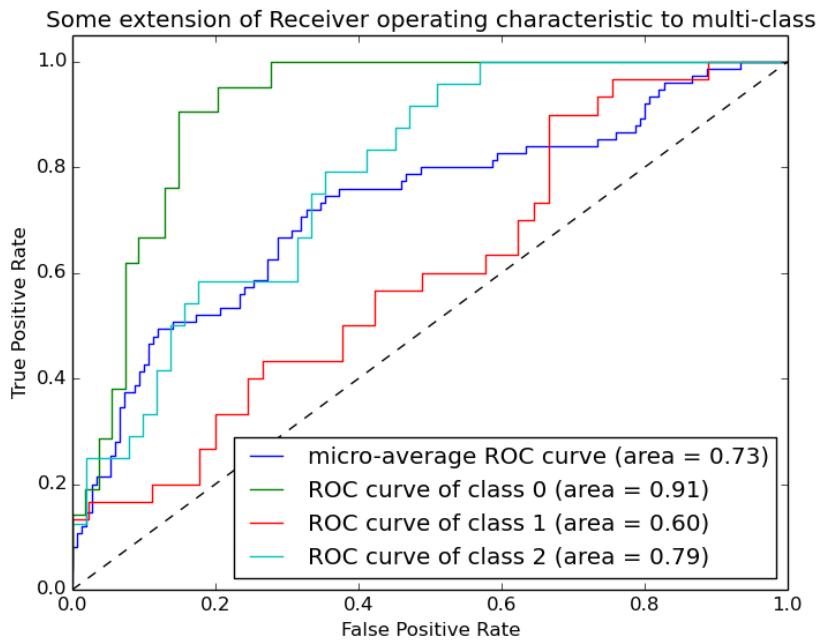


Figura 4.7: Ejemplo del método de evaluación mediante la curva ROC para un algoritmo de clasificación multiclas aplicando Scikit-learn, 2014. *Fuente: scikit-learn.org.*

- **Keras:** esta librería fue desarrollada en 2015 y tiene como objetivo simplificar los algoritmos basados en aprendizaje profundo para que sean más intuitivos y de alto nivel. Keras está diseñado para construir, por bloques, la arquitectura de cada red neuronal. Y, para ello, usa librerías de más bajo nivel como TensorFlow, Microsoft Cognitive Toolkit o Theano.
- **Matplotlib:** es una librería de Python desarrollada en 2002 y especializada en la creación de gráficos en dos dimensiones. Está construido sobre Numpy e integrado en Pandas.

Las versiones de las librerías, mencionadas anteriormente, que se han usado en este trabajo son:

- Numpy: 1.21.6
- Pandas: 1.3.5
- Lightkurve: 2.3.0
- PyWavelets: 1.4.1
- Tensorflow: 2.9.2
- Scikit-learn: 1.0.2
- Keras: 2.9.0
- Matplotlib: 3.2.2

4.4 Preprocesamiento de los datos

4.4.1 Filtrado inicial

Al aplicar los filtros mencionados en el apartado 4.1, se obtiene un total de 2750 curvas de luz de interés como se puede observar en la Figura 4.3.

Por otro lado, para realizar el análisis, se necesita separar las 2750 curvas de luz en dos grupos distintos:

- Se obtiene 2181 curvas de luz, de los cuales 1215 pertenecen a falsos positivos y 966 corresponden a exoplanetas confirmados. Este grupo de curvas es el que se destina para entrenar la red neuronal.
- Se dispone de un total de 569 curvas de luz pertenecientes a posibles candidatos por confirmar. Este conjunto de curvas es el destinado a ser analizado una vez terminada la red neuronal.

4.4.2 Descarga y ajustes de los datos

1. Descarga de los datos de las curvas

El siguiente paso, una vez ya filtrado las curvas de interés, es proceder a descargar los datos de cada una de las curvas de luz con los Kepler ID del apartado anterior correspondientes con exoplanetas confirmados o falsos positivos.

Para ello, se ha usado la librería Lightkurve comentado en el punto 4.3. Concretamente, se ha empleado el siguiente script:

```
lc_search = lk.search_lightcurve('KIC ' + str(kepler_id), mission = 'Kepler')
lc_collection = lc_search.download_all()
```

En la variable `lc_collection` se almacenan todos los datos de la curva de luz correspondientes que están disponibles en la base de datos.

2. Unión de las curvas y eliminación de valores anómalos o nulos

A continuación, se unen los datos de los diferentes trimestres en una única curva mediante el comando `stitch`. Para la eliminación de valores nulos (NaN) o valores anómalos, usamos los comandos `remove_nans` y `remove_outliers` respectivamente:

```
lc_ro = light_curve_collection.stitch()
lc_ro = lc_ro.remove_outliers(sigma=20, sigma_upper=4)
lc_nonans = lc_ro.remove_nans()
```

4.4.3 Plegado par e impar de las curvas

El siguiente paso a realizar es el plegado de las curvas para ajustarlo a un periodo definido (correspondiente al valor de `period` que se mencionó en el 4.1). Este proceso consiste primero en concatenar todos los tránsitos en una sola curva (usando el comando `stitch` como se ha comentado en el punto 2 del apartado 4.4.2). Una vez unida la curva, se procede a la extracción de la máscara impar (`odd`) y par (`even`) (correspondientes a las curvas par e impar comentado en el apartado 2.2.5) como se observa a continuación:

```
lc_fold = lc_nonans.fold(period = period, epoch_time = epoch)
lc_odd = lc_fold[lc_fold.odd_mask]
lc_even = lc_fold[lc_fold.even_mask]
```

Una vez plegadas estas curvas, se pueden representar en un gráfico de fase en el que el eje *X* representa la fase (con valores variable) y el eje *Y* representa el flujo normalizado del objeto como se puede observar en la siguiente figura:

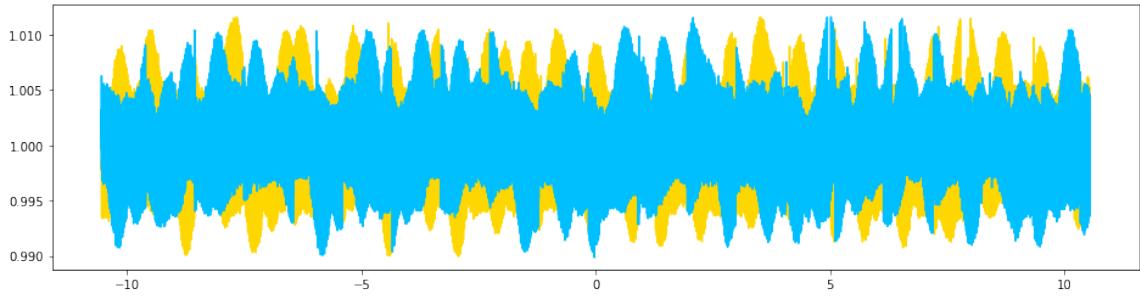


Figura 4.8: Plegado par (amarillo) e impar (azul) de la curva de luz perteneciente al exoplaneta confirmado con Kepler ID 10717220 y un total de 25827 puntos. *Fuente: Elaboración propia.*

4.4.4 Filtrado por número de puntos

Sin embargo, debido a un problema que se observó en una etapa posterior de análisis ya con las curvas plegadas, se necesita realizar un filtro adicional sobre el dataframe obtenido en el apartado anterior. Concretamente, se ha observado que el número de puntos de cada curva de luz presenta variaciones muy notables. Por ejemplo, la curva de luz con Kepler ID 11446443 tiene un total de 808070 puntos mientras que la curva con Kepler ID 10797460 tiene únicamente 32458 puntos como se observa en la figura siguiente:

```
[14] wavelet_family = 'sym5'
    level = 7
    period,epoch=df[df['kepid']== 10797460][['koi_period','koi_time0bk']].iloc[0]
    len_curva(10797460,period,epoch,wavelet_family,level,plot = False, plot_comparative=False)
32458

[13] wavelet_family = 'sym5'
    level = 7
    period,epoch=df[df['kepid']==11446443][['koi_period','koi_time0bk']].iloc[0]
    len_curva(11446443,period,epoch,wavelet_family,level,plot = False, plot_comparative=False)
808070
```

Figura 4.9: Código en Python que muestra los puntos que tienen las curvas con ID 10797460 (arriba) e ID 11446443 (abajo). *Fuente: Elaboración propia.*

Esta diferencia interfiere negativamente en el entrenamiento de la red neuronal, ya que las curvas pueden presentar variaciones muy grandes en las longitudes del periodo de tránsito si la diferencia entre el número de puntos de las curvas es muy grande. Por ejemplo, en la Figura 4.10 podemos ver las curvas de luz pertenecientes a los dos ID anteriores:

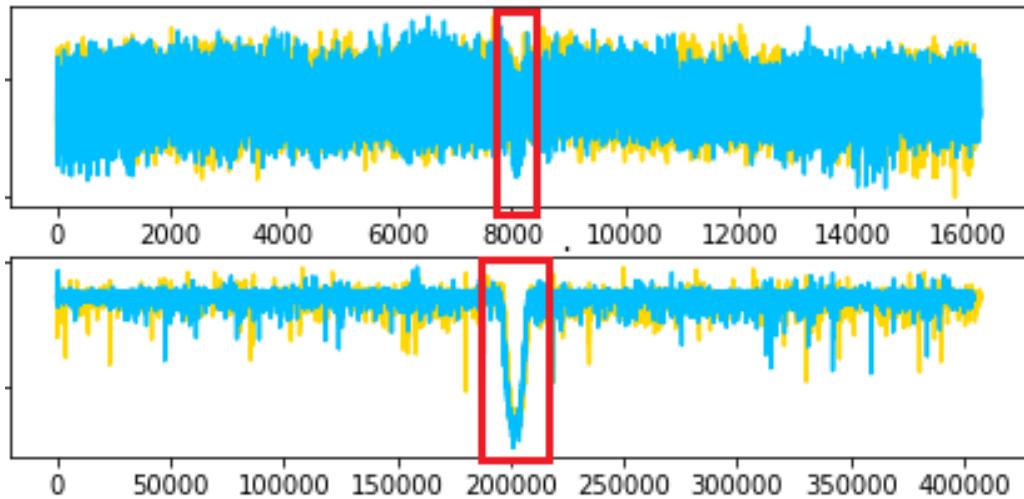


Figura 4.10: Curvas de luz pertenecientes los exoplanetas confirmados con Kepler ID 10797460 (arriba) e ID 11446443 (abajo). El tránsito está resaltado en rojo. *Fuente: Elaboración propia.*

En la curva superior, con un total de 32458 puntos, el periodo de tránsito (marcado en rojo) tiene, aproximadamente, de 300 a 600 puntos. Mientras que en la curva de abajo, de 808070 puntos, el periodo de tránsito ocupa entre 8000 y 12000 puntos aproximadamente. Por lo que, al intentar fijar un número de puntos determinados para formar el dataset para la red neuronal⁴, no se puede llegar a un número idóneo para los dos casos al mismo tiempo.

Por ejemplo, si elegimos los 1000 puntos centrales para estas dos curvas. Para la primera curva (la de 32458 puntos) se cubriría el periodo de tránsito completamente. Mientras que en la segunda curva solamente se habría captado la parte central del tránsito, sin proporcionar información del resto del tránsito por lo que se produciría una pérdida de información notable para la red neuronal.

En cambio, si cogemos, por ejemplo, los 20000 puntos centrales, entonces en la curva inferior se habría captado toda la información del tránsito. Mientras que en la curva superior, se habría cogido prácticamente toda la curva, añadiendo ruido de manera excesiva e innecesaria.

Por lo que la solución que se ha optado en este caso es, fijar un rango de valores para incluir en el dataset solamente aquellas curvas de luz que tengan un número de puntos entre ese rango. Después de adaptar el código mostrado en la Figura 4.8 con un bucle for, se ha aplicado dicho algoritmo a las 200 primeras curvas y se ha observado que la mayoría de las curvas tenían entre 25000 y 35000 puntos. Por lo que, para evitar el problema comentado anteriormente, se ha optado por fijar el rango en [25000, 35000].

⁴ya que todas las entradas tienen que tener la misma dimensión para poder entrenar la red

Aplicando dicho filtro adicional sobre el dataset obtenido en el apartado 4.4.1, se han obtenido los siguientes números de curvas de cada tipo:

- 935 falsos positivos (1215 anteriormente).
- 762 exoplanetas confirmados (966 anteriormente).
- 328 candidatos a determinar (569 anteriormente).

El segundo problema que se ha encontrado es que, aun disponiendo de la versión de pago de Colab, al añadir el algoritmo del filtrado del número de puntos, el coste computacional ha crecido notablemente y no se dispone de recursos suficientes para analizar todas las curvas de luz en el tiempo límite que se dispone con dicha versión de Colab (24 horas). Por lo que finalmente se decidió optar los siguientes números de curvas para cada caso:

- 500 falsos positivos y 500 exoplanetas confirmados para el entrenamiento y la validación.
- 50 falsos positivos y 50 confirmados para el test.
- 100 candidatos para probar la predicción.

De esta manera, se obtiene equidad para el dataset de entrenamiento y validación entre los casos confirmados y los falsos positivos para evitar posibles errores debido a la disparidad en los datos.

4.4.5 Aplicación de la transformada wavelet para la descomposición de las curvas de luz

El siguiente paso es la aplicación de la transformada wavelet a las curvas de luz del dataframe. En el apartado 2.3 ya se vio las características principales de las transformadas wavelet, tanto las discretas como las continuas. En este apartado se centrará en su aplicación práctica en las curvas de luz. Como ya se sabe, la finalidad de las transformadas wavelet sirven para aproximar de datos con variaciones o con discontinuidades abruptas.

A lo largo de los últimos años, se fueron agregando distintas funciones wavelet según las necesidades. Para el presente trabajo, se ha optado por la familia Wavelet Symlet, concretamente la transformada Symlet 5. La elección de esta función es debida a que es la función que obtuvo mejores métricas en un estudio anterior relacionado con el presente estudio (Guirado, 2022).

Los Symlets son wavelet casi simétricas derivadas de la familia de wavelet Daubechies. Estas últimas son wavelets ortonormales de soporte compacto enfocadas al análisis wavelet discreto. En la Figura 4.11 podemos ver las diferentes funciones wavelet pertenecientes a la familia Symlet.

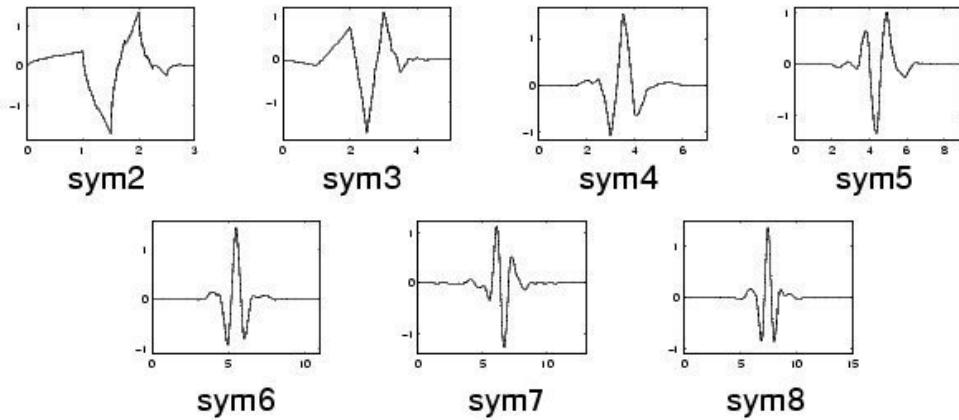


Figura 4.11: Funciones wavalet pertenecientes a la familia Symlets. *Fuente: Researchgate, 2014.*

Una vez elegida la transformada wavelet, se ha optado por calcular los 7 primeros niveles basándose en los resultados de un estudio previo realizado sobre el campo (Guirado, 2022). En la Figura 4.13 se pueden observar los 7 niveles de descomposición wavelet aplicado al exoplaneta confirmado con Kepler ID 2713049. Sin embargo, en este punto se presentan otros dos problemas:

1. Anomalías tras la aplicación de la transformada wavelet

El primer problema que se presenta es que, al aplicar la transformada wavelet a las curvas de luz, aparecen, sobre todo, en los últimos niveles de la descomposición, dos anomalías en las curvas de luz (una en cada extremo) como se observa en la Figura 4.12 (resaltado en rojo). Estas anomalías pueden interferir en las predicciones de la red neuronal porque producen una variación notable del flujo de la curva de luz que se puede confundir con un tránsito. Dicha variación es inexistente en la curva de luz original, por lo que es consecuencia de aplicar la transformada wavelet. Entonces, si todas las curvas de luz conservan esta anomalía, los resultados del análisis pueden resultar muy imprecisos.

Sin embargo, la solución a este problema es relativamente fácil. Una vez aplicada la transformada wavelet en los distintos niveles, se puede realizar un truncamiento en las curvas de luz de cada nivel excluyendo los dos extremos. De esta manera se puede eliminar de manera sencilla este tipo de anomalías.

2. Diferencias en el número de puntos de cada nivel para las distintas curvas

En el apartado 4.4.4 se ha visto que, cada una de las curvas posee un número diferente de puntos. Por lo que la dimensión de cada curva de datos es distinta y, como consecuencia, las descomposiciones wavelet de cada nivel, también lo es. Y, como para entrenar la red neuronal, se necesita que todas las entradas tengan la misma dimensión, entonces se necesita unificar las dimensiones para solucionar este problema.

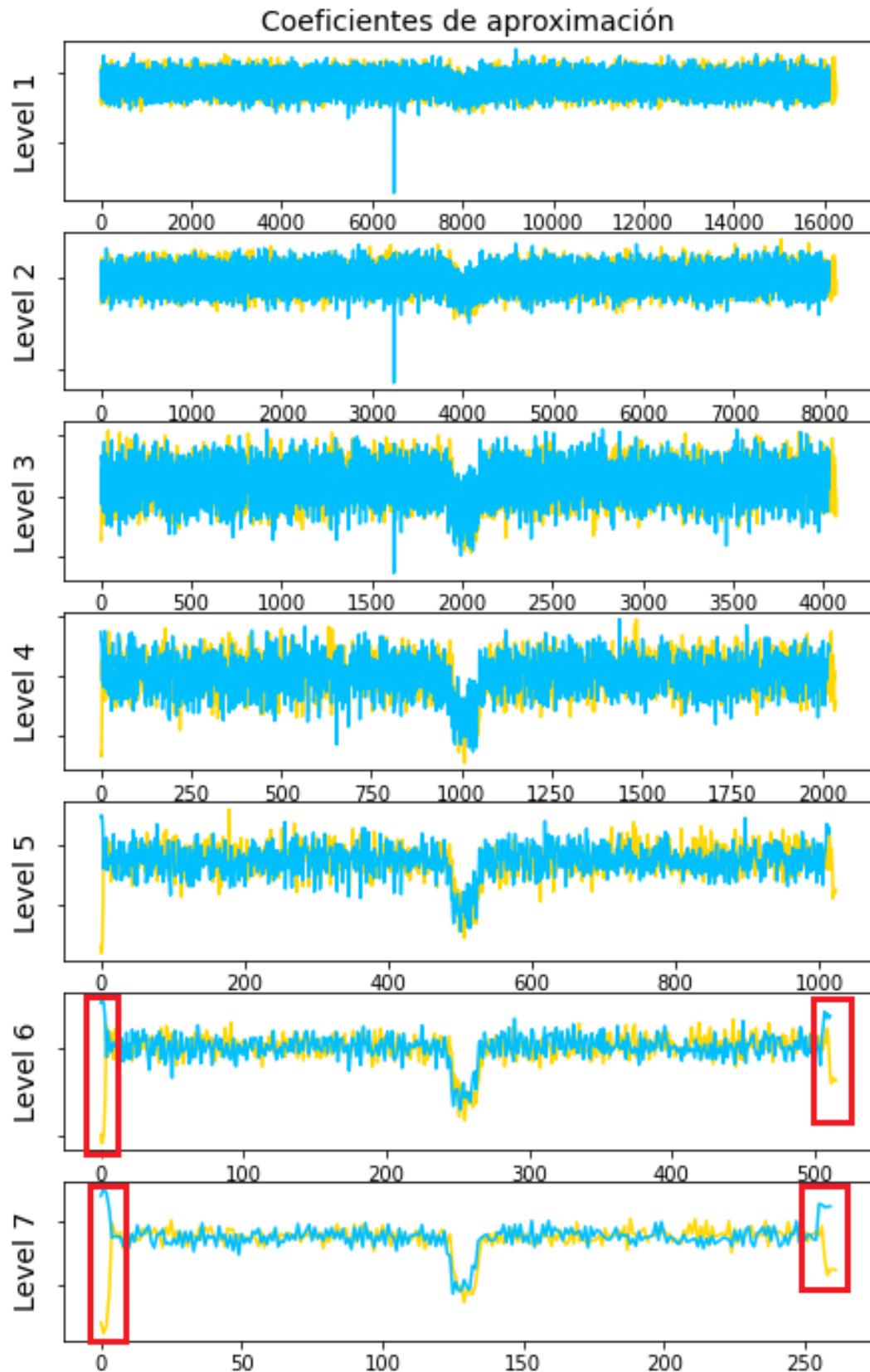


Figura 4.12: Los 7 niveles de descomposición correspondientes a las curvas de luz del exoplaneta confirmado con Kepler ID 2713049. *Fuente: Elaboración propia.*

Hay que tener en cuenta que, después del filtro realizado en el apartado 4.4.4, el número de puntos totales de cada curva pertenece al intervalo [25000, 35000]. Por lo que, mediante una observación de la longitud del tránsito de las primeras 100 curvas, se ha decidido fijar un número de puntos para cada nivel de descomposición wavelet como se muestra a continuación:

| Nivel | Nº de puntos |
|-------|--------------|
| 1 | 1001 |
| 2 | 501 |
| 3 | 251 |
| 4 | 125 |
| 5 | 65 |
| 6 | 33 |
| 7 | 17 |

Figura 4.13: Los 7 niveles de descomposición wavelet y el número de puntos por nivel.
Fuente: *Elaboración propia.*

El número de puntos elegido para cada nivel siempre es impar. El objetivo de esto es, para cada nivel, coger un número determinado de puntos a la derecha y a la izquierda del punto central. Se distinguen dos casos dependiendo de si la curva de dicho nivel tiene un número de puntos par o impar:

- Por ejemplo, para una determinada curva de luz, si el número de puntos de curva de nivel 1 es par, entonces, se toman los dos puntos centrales de dicha curva, sean M y N . Y, en este caso, se toma como el punto central el menor de ambos, es decir:

$$C = \min(M, N)$$

Y el intervalo escogido para la curva wavelet de nivel 1 es:

$$I = [C - 500, C + 500]$$

- En el otro caso, si el número de puntos de la curva de nivel 1 es impar, entonces, siendo C el punto central de dicha curva, el intervalo correspondiente es:

$$I = [C - 500, C + 500]$$

De esta manera, se obtiene un número fijo de puntos para cada nivel, todas ellas con el punto central, el valor central del tránsito.

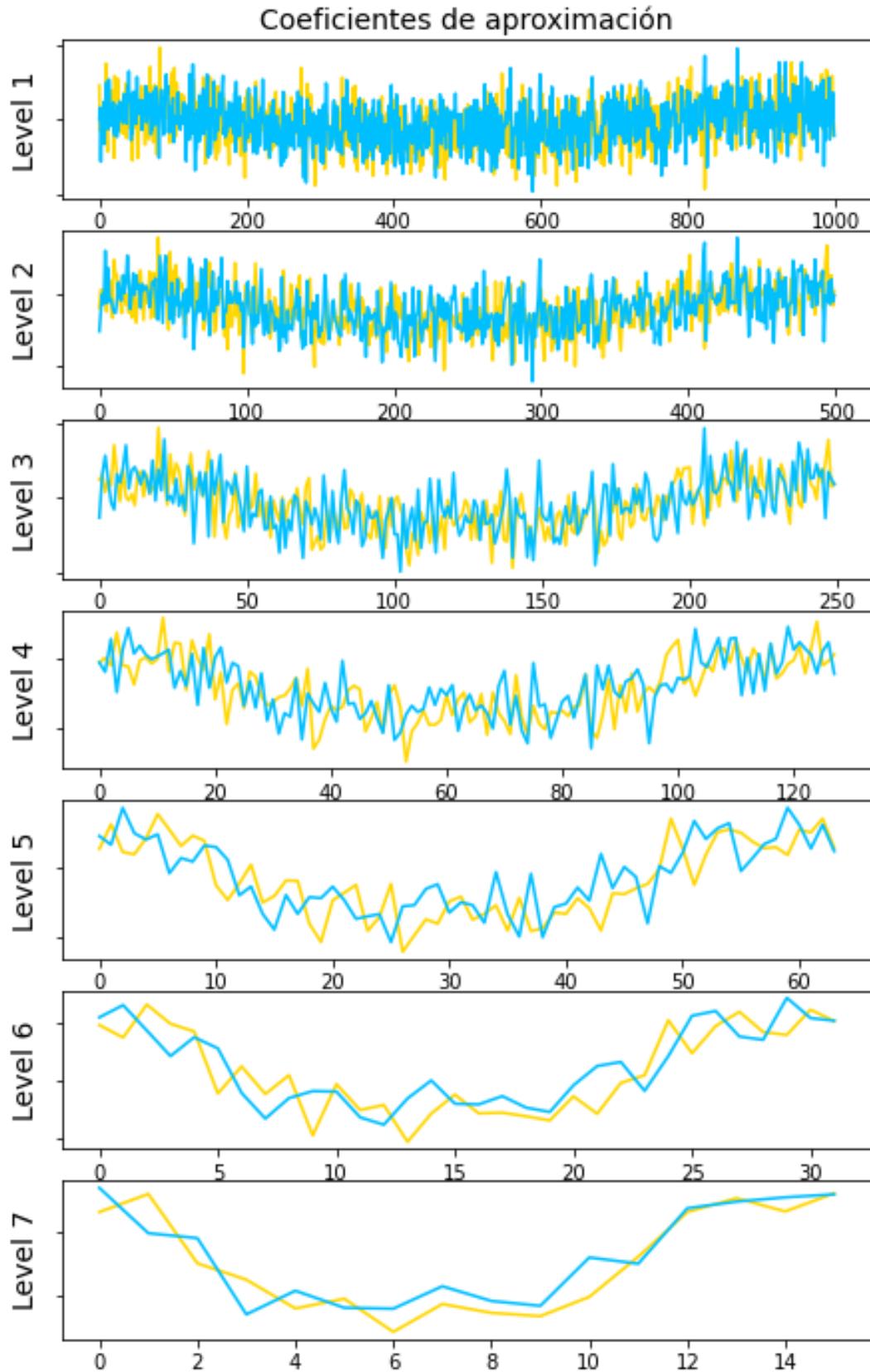


Figura 4.14: Los 7 niveles de descomposición wavelet una vez solucionado los dos problemas anteriores. *Fuente: Elaboración propia.*

En la Figura 4.14 se puede observar la misma curva de luz de la Figura 4.12 después de haber aplicado las soluciones a los dos problemas comentados. En este caso, se puede observar que se ha disminuido de forma notable el número de puntos por nivel. Manteniéndolos en unos números fijos para cada nivel. Además, tras hacer el truncamiento, ya no existe ninguna anomalía en los extremos de las curvas que pueda provocar confusión para las redes neuronales.

4.4.6 Obtención del dataframe final

Una vez concluido el análisis inicial de los datos de entrada, se ha obtenido finalmente los datos de las curvas de luz plegadas en par e impar y transformada mediante wavelet. El siguiente paso es transformar dichos datos para poder usarlo como datos de entrada en las redes neuronales.

Al principio, se ha pensado en trabajar con las propias gráficas como la mostrada en la Figura 4.14. Sin embargo, finalmente se ha optado, para aumentar la precisión de los datos de entrada, por trabajar con los propios datos numéricos de cada nivel que se ha obtenido tras aplicar la transformada wavelet.

| | koi_disposition | curva_par | curva_impar |
|---|-----------------|--|---|
| 0 | [1.0, 0.0] | [2.8293717, 2.8276892, 2.8277047, 2.8301203, 2.... | [2.828079, 2.830107, 2.8275323, 2.828484, 2.82... |
| 1 | [1.0, 0.0] | [2.8283567, 2.8284018, 2.828578, 2.8293953, 2.... | [2.8277256, 2.8282843, 2.8277252, 2.827976, 2.... |
| 2 | [0.0, 1.0] | [2.827928, 2.828656, 2.828039, 2.8286307, 2.82... | [2.828048, 2.8279974, 2.8281069, 2.8287427, 2.... |
| 3 | [1.0, 0.0] | [2.8326132, 2.8305056, 2.8253498, 2.8275583, 2.... | [2.8275824, 2.8334022, 2.8268592, 2.825933, 2.... |
| 4 | [1.0, 0.0] | [2.8292754, 2.8285542, 2.828328, 2.828544, 2.82... | [2.8295243, 2.8278937, 2.828325, 2.82869, 2.82... |
| 5 | [0.0, 1.0] | [2.8229568, 2.823623, 2.823449, 2.824882, 2.82... | [2.82922, 2.8282337, 2.827767, 2.828308, 2.825... |

Figura 4.15: Primeras entradas pertenecientes a las curvas de nivel 3 tras aplicar la transformada wavelet. *Fuente: Elaboración propia.*

En la Figura 4.15 se puede observar 6 entradas que se corresponden con las curvas de nivel 3 tras aplicar wavelet a las 6 primeras curvas de luz del dataset obtenido para el entrenamiento y test de la red neuronal. A continuación se detalla la representación de cada columna:

- **koi_disposition:** corresponde con el estado de la curva de luz. Se ha sustituido CONFIRMED y FALSE POSITIVE por vectores de dos coordenadas : [1.0, 0.0] (para los exoplanetas confirmados) y [0.0, 1.0] (para los falsos positivos). Esta transformación se debe a que, después de realizar las pruebas posteriores correspondientes, se han obtenido mejores resultados con la función softmax para la función de activación de la última capa. Y por definición de esta función, se necesita un vector de salida con, al menos, dos coordenadas pues la suma de todos sus puntos tiene que ser igual a la

unidad. Por lo que si solo se configura una coordenada de salida, este siempre sería 1 haciendo perder completamente la capacidad de predicción del modelo.

- **curva_par**: en este caso, las entradas se corresponde con la transformada wavelet (plegado par) del nivel 3 perteneciente a las curvas de luz.
- **curva_impar**: las entradas se corresponde con la transformada wavelet (plegado impar) del nivel 3 perteneciente a las curvas de luz.

Una vez terminada la transformación anterior, para cada nivel en la trasformada wavelet, se ha obtenido un dataframe correspondiente en el que se indica el estado de la curva de luz (confirmado o falso positivo expresados en vectores de dos coordenadas) y los datos correspondientes al plegado par e impar de las curvas de luz del nivel correspondiente.

4.5 Construcción de la red neuronal

4.5.1 Normalización de los datos

Dada la diversidad en los datos de las curvas de luz, la normalización resulta ser un paso esencial para que los datos numéricos sean comparables en una escala común. En este caso, se ha aplicado una normalización `min-max` para transformar cada uno de los valores de cada una de las curvas de cada nivel en valores entre 0 y 1.

Si C es la curva de longitud n , $\{p_i\}$ con $i \in \{1, \dots, n\}$ son los puntos de la curva C y q_i con $i \in \{1, \dots, n\}$ son los datos normalizados, entonces, la fórmula aplicada para la normalización es la siguiente:

$$q_i = \frac{p_i - \min(C)}{\max(C) - \min(C)} \quad (4.1)$$

Aplicando la fórmula 4.1 a todas las curvas de los 7 niveles, se obtienen todas las curvas normalizadas. En la Figura 4.16 se puede ver un ejemplo de las primeras 6 curvas de nivel 3 normalizadas.

| | koi_disposition | curva_par | curva_impar |
|----------|------------------------|--|---|
| 0 | [1.0, 0.0] | [0.7268536, 0.9983077, 0.9983151, 0.99946296, ...] | [0.4792021, 0.99983793, 0.99874294, 0.9991477, ...] |
| 1 | [1.0, 0.0] | [0.66703254, 0.99928516, 0.99936664, 0.9997445...] | [0.5317571, 0.9992015, 0.9989582, 0.9990673, 0...] |
| 2 | [0.0, 1.0] | [0.95026535, 0.99898255, 0.9986544, 0.99896914...] | [0.9445001, 0.9984047, 0.9984627, 0.99879974, ...] |
| 3 | [1.0, 0.0] | [0.8636907, 0.997951, 0.9953349, 0.9964555, 0....] | [0.57266307, 0.99965906, 0.9967659, 0.9963563, ...] |
| 4 | [1.0, 0.0] | [0.9496195, 0.9994971, 0.9993768, 0.9994917, 0...] | [1.0, 0.9992073, 0.99944335, 0.9996428, 0.9987...] |
| 5 | [0.0, 1.0] | [0.0, 0.99621624, 0.99615484, 0.9966605, 0.997...] | [0.4585271, 0.99735314, 0.9971566, 0.9973844, ...] |

Figura 4.16: Curvas de luz par e impar de nivel 3 normalizados. *Fuente: Elaboración propia.*

4.5.2 Formateo en array

Una vez normalizado los datos, el siguiente paso es transformar los datos de `list` a `array`. Esto es, como se ha comentado en el apartado 4.3, un formato especial de datos sobre el que se construye algunas de las bibliotecas utilizadas en el presente trabajo. Para ello, se procede a concatenar las curva par e impar de cada curva de luz en un mismo `array` usando la siguiente función:

```
def formateo(df_val):
    #Formateamos los datos para la red neuronal para que sean array
    DatosIniciales = []
    for i in range(len(df_val)):
        a = df_val['curva_par'][i]
        b = df_val['curva_impar'][i]
        a = a.tolist()
        b = b.tolist()
        c = a + b
        DatosIniciales.append(c)
    Datos = np.array(DatosIniciales)
    return Datos
```

Figura 4.17: Función en Python para concatenar la curva par e impar de una curva de luz en un mismo `array`. *Fuente: Elaboración propia.*

En la siguiente figura se puede visualizar un ejemplo de los datos una vez concatenados y transformados en `array`:

```
array([[0.72685361, 0.9983077 , 0.9983151 , ..., 0.99972183, 0.99969983,
       1.        ],
      [0.66703254, 0.99928516, 0.99936664, ..., 0.99972802, 1.        ,
       1.        ],
      [0.95026535, 0.99898255, 0.99865443, ..., 0.99988967, 1.        ,
       1.        ],
      ...,
      [0.        , 0.99621624, 0.99615484, ..., 0.99949074, 0.99953979,
       1.        ],
      [0.69524127, 0.99932802, 0.9996177 , ..., 0.99972266, 1.        ,
       1.        ],
      [0.58371204, 0.99913484, 0.99949616, ..., 0.99970335, 1.        ,
       1.        ]])
```

Figura 4.18: Visualización de los datos en formato `array`. *Fuente: Elaboración propia.*

Es decir, para cada nivel de descomposición wavelet, tenemos un `array` que está formado por otros 1000 `arrays` pertenecientes a las 1000 curvas de luz disponibles. Y cada uno de estos 1000 `arrays`, a su vez, está compuesto por la concatenación de dos `arrays`: la primera perteneciente a la descomposición par de la curva y la segunda a la descomposición impar como se muestra a continuación:

`[[curva_par1, curva_impar1], [curva_par2, curva_impar2],...]`

4.5.3 Separación de datos

Con los datos de entrada ya en el formato correcto, el siguiente paso consiste en separar los datos de entrada en dos conjuntos: uno de entrenamiento y otro de test. Para cada uno de estos dos conjuntos, se separan en dos grupos: los datos de entrada y las etiquetas.

Como ya se ha comentado en el punto 4.4.4, se dispone de un total de 1000 curvas de luz para entrenar la red neuronal, 500 pertenecientes a exoplanetas confirmados y 500 pertenecientes a falsos positivos. Para cada una de estas 1000 curvas, se han obtenido sus plegados en cada uno de los 7 niveles con la transformada wavelet. Por lo que en total se dispone de 7000 datos de entrada, 1000 para cada nivel.

Con la observación de las primeras gráficas de fase como la mostrada en la Figura 4.14, se ha comprobado que las curvas de los niveles del 3 al 7 proporcionan, a priori, mejores resultados para la red neuronal que las curvas de nivel 1 y 2 (éstas tienen demasiadas variaciones en sus gráficas como se observa en la Figura 4.14). Por lo que finalmente se ha optado trabajar con las curvas de esos 5 niveles.

Se van a entrenar 5 redes neuronales por separado, una red por cada uno de los niveles del 3 al 7. Por lo que las 1000 curvas de entrenamiento y validación para cada nivel se han distribuido de la siguiente manera:

- 350 curvas de exoplanetas confirmados y 350 curvas de falsos positivos para el conjunto de entrenamiento.
- 150 curvas de exoplanetas confirmados y 150 curvas de falsos positivos para el conjunto de validación.
- 50 confirmados y 50 falsos positivos para el conjunto de test.

Adicionalmente, como ya se ha comentado, analizaremos los resultados de predicción de los modelos usando el conjunto de las 100 curvas de luz con la etiqueta de CANDIDATO.

Por otro lado, en cada dataframe como la mostrada en la Figura 4.16, se ha separado la columna de `koi_disposition` para ser el conjunto de etiqueta de resultados ([1.0, 0.0] corresponde a exoplaneta confirmado y [0.0, 1.0] corresponde a falso positivo) mientras que las columnas `curva_par` y `curva_impar` se han concatenado para formar el conjunto de datos de entrada como se ha descrito anteriormente.

4.5.4 Formato de los resultados

Como se ha comentado en el apartado anterior, se ha elaborado una red neuronal por cada uno de los 5 niveles de interés. De esta manera, se obtienen 5 redes neuronales independientes cuyos resultados dependen únicamente del entrenamiento de las curvas del

nivel correspondiente. Una vez finalizado el entrenamiento de las 5 redes neuronales, se combinan los resultados obtenidos para predecir las 100 curvas de test mencionadas en el punto 4.5.3, devolviendo la probabilidad de que sea un exoplaneta confirmado. Por ejemplo, si el resultado de la predicción de una curva imaginaria C usando los 5 modelos es:

| | |
|----------|----------------|
| Modelo 3 | [0.612, 0.388] |
| Modelo 4 | [0.102, 0.898] |
| Modelo 5 | [0.453, 0.547] |
| Modelo 6 | [0.232, 0.768] |
| Modelo 7 | [0.955, 0.045] |

donde la primera coordenada representa la probabilidad de que sea un exoplaneta confirmado y la segunda corresponde a la probabilidad de que sea un falso positivo.

Por un análisis posterior de los resultados de las 100 curvas del conjunto de test, se ha observado que, las curvas correspondientes a los exoplanetas confirmados obtienen, en general, una probabilidad mayor del 80% (en el caso de que se haya predicho correctamente). Por lo que, para aumentar la precisión de dicha predicción en el modelo final, se ha establecido los siguientes criterios:

- Si para un modelo, el valor de la probabilidad de que sea un exoplaneta es mayor o igual que 0.80, entonces, para el cálculo final del porcentaje, se sustituye este valor por 1.
- En cambio, si el valor de dicha probabilidad para un modelo es menor o igual que 0.20, entonces, se sustituye este valor por 0.
- Si el valor pertenece al intervalo (0.20, 0.80), entonces se conserva dicha probabilidad para el cálculo final.

Una vez transformados los valores iniciales de probabilidad siguiendo las reglas anteriores, se aplica la siguiente fórmula para determinar la probabilidad final:

$$P_{\text{final}}(x) = \sum_{i=3}^7 \frac{P_i(x_i)}{5}, \quad (4.2)$$

siendo x la curva analizada, x_i la curva del nivel correspondiente en la descomposición wavelet, $P_{\text{final}}(x)$ la probabilidad final y $P_i(x_i)$ la probabilidad transformada del nivel i correspondiente a la curva x_i .

Por ejemplo, para el caso visto al inicio de este apartado, la probabilidad final para la curva imaginaria C que se ha analizado es:

$$P_{\text{final}}(C) = \frac{0.612 + 0 + 0.453 + 0.232 + 1}{5} = 0.659$$

4.5.5 Capas de las redes neuronales

Para construir un modelo de redes neuronales adecuado, se necesita combinar los diferentes tipos de capas existentes con los valores adecuados para sus variables y en el orden correcto. Por lo que es esencial conocer la función de cada una de las capas y su influencia en la red neuronal. A continuación, veamos los diferentes tipos de capas que hemos empleado en este trabajo:

Capa convolucional

Como su nombre indica, esta capa se caracteriza por la operación matemática de convolución. Esto consiste en tomar conjuntos de valores cercanos de la matriz de entrada, siempre del mismo tamaño, y realizar el producto escalar de ese conjunto con un kernel predefinido. Al recorrer toda la matriz de entrada en orden, se obtiene una nueva matriz con diferentes parámetros como se muestra en la siguiente figura:

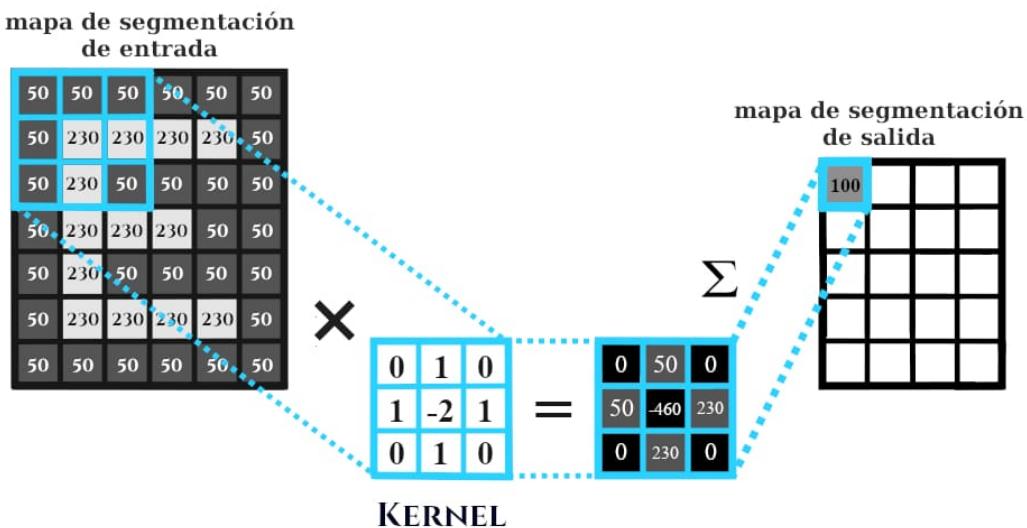


Figura 4.19: Ejemplo de la operación de convolución de dos dimensiones con un kernel predefinido de tamaño 3x3 aplicado a una imagen de tamaño 7x6 y con `stride = 1` para obtener una matriz final de dimensiones 4x5. *Fuente: Dive into Deep Learning, 2020.*

Este tipo de capas tiene como objetivo extraer características propias de cada matriz a la vez que lo comprime para reducir su tamaño inicial. En el caso en que no se desea una reducción de la dimensión, se suele realizar la operación de Padding. Esta operación consiste simplemente en agregar valores nulos alrededor de la matriz original, los cuales serán eliminados nuevamente después de la convolución. Además, el Padding tiene una función adicional. Y es que también se aplica en aquellos casos en los que se presenta información relevante en los extremos y no se quiere perder dicha información después de la convolución.

La otra variable a tener en cuenta es el **stride**. Al realizar la operación de **convolución**, el valor de **stride** indica el desplazamiento que se realiza entre una convolución y la siguiente. Siendo 1 o 2 los valores habituales dependiendo de las dimensiones del kernel empleado.

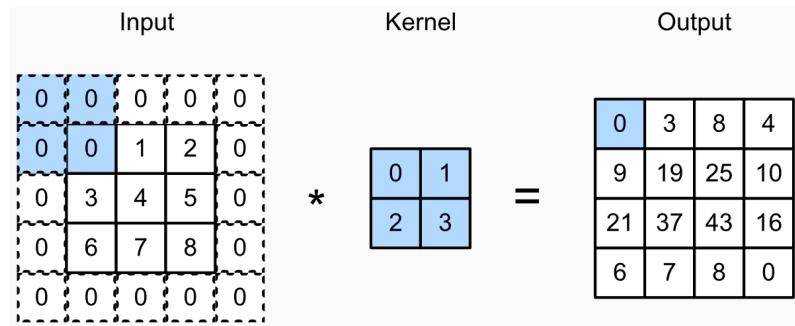


Figura 4.20: Ejemplo de convolución después de aplicar padding. Fuente: Rubén Rodríguez Abril, LMO, 2022.

Capa Densa o Fully-connected

Son las capas básicas de cálculo de las redes neuronales. Consiste en que cada neurona de la capa se conecta con todas las neuronas de la capa anterior como se muestra en la siguiente figura:

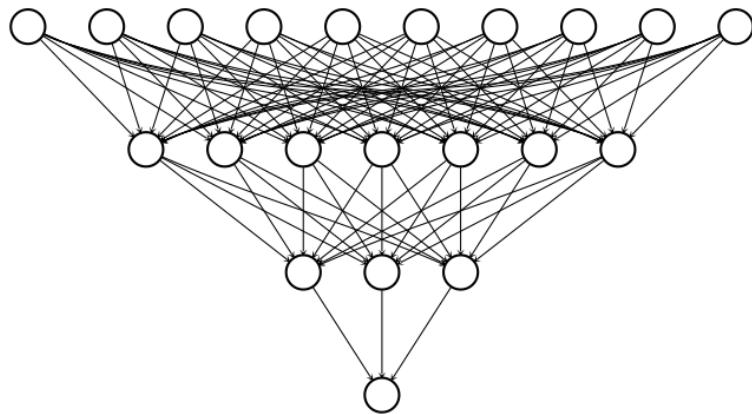


Figura 4.21: Ejemplo de una red neuronal con capas **densas** en el que todas las neuronas de una capa están conectadas con todas las neuronas de la capa anterior. Fuente: Datacamp, 2021.

Capa MaxPooling

El **pooling** es una operación que se aplica por bloques en una matriz y permite extraer la información más representativa de cada uno de los bloques. Por lo general se aplica entre dos capas de convolución. El **max-pooling** consiste en dividir la matriz en bloques del mismo tamaño y, en cada bloque, se extrae el máximo de dicho bloque.

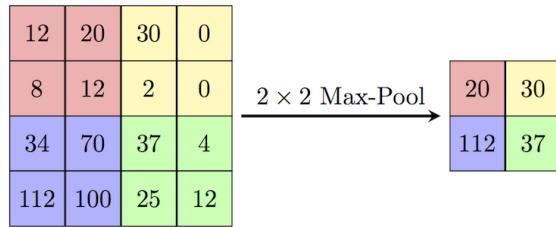


Figura 4.22: Ejemplo de **maxpooling** de tamaño 2×2 aplicado a una matriz de 4×4 padding.
Fuente: *Computer Science Wiki, 2018.*

Capa Dropout

Dropout es un método que consiste en desactivar un número determinado de neuronas de forma aleatoria. En cada iteración, **dropout** desactiva diferentes neuronas, de manera que obliga a las neuronas cercanas buscar nuevos patrones ayudando a reducir el overfitting.

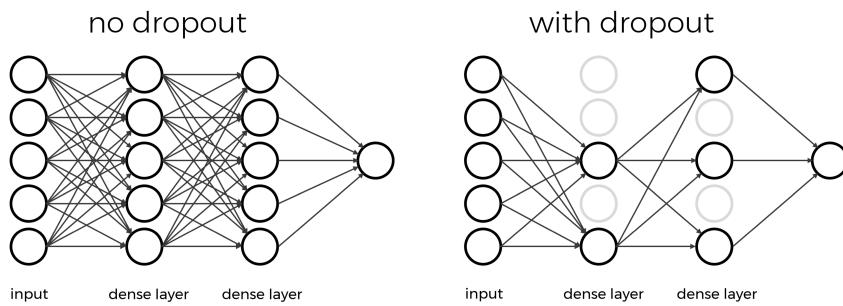


Figura 4.23: Una red neuronal sin **dropout** (izquierda) comparado con una red neuronal con **dropout** (derecha). Fuente: *Github, 2022.*

Capa Flatten

La capa **Flatten** se usa para “aplanar” la matriz de entrada, es decir, reducir a una sola dimensión la matriz de entrada multidimensional. Esta operación se emplea normalmente en la transición de la capa convolucional a la capa **Fully-connected**.

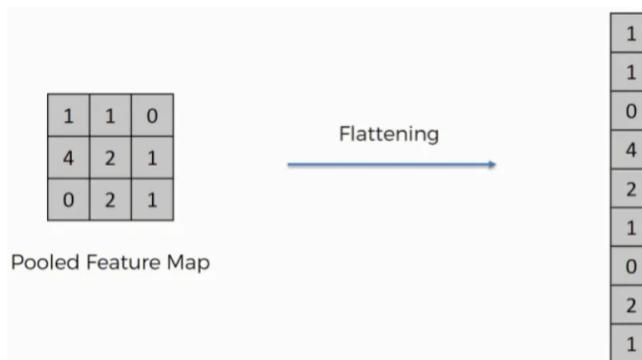


Figura 4.24: Ejemplo de la operación de **Flatten**. Fuente: *Bootcamp AI, 2019.*

4.5.6 Modelos de las redes neuronales

Una vez vistas las capas que se van a emplear, se procede a construir las 5 redes neuronales correspondientes a las 5 niveles de descomposición wavelet. Hay que tener en cuenta, siguiendo el 4.5.2, que las dimensiones de los input son el doble de la longitud de las descomposiciones par e impar de cada nivel. Por ejemplo, para una curva de nivel 3, la dimensión de entrada es 502 porque se han concatenado dos curvas de 251 datos de entrada. Finalmente se han construido las redes neuronales siguiendo las siguientes estructuras:

NIVEL 3 input size: (502,) Epoch: 100 Batch size: 32

```
Conv1D(251, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=3, strides=1, padding='valid')
Conv1D(502, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=3, strides=1, padding='valid')
Dropout(0.5)
Dense(502, activation='relu')
Dense(251, activation='relu')
Dense(2, activation='softmax')
```

NIVEL 4 input size (250,) Epoch: 100 Batch size: 32

```
Conv1D(125, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Conv1D(250, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Dropout(0.5)
Dense(250, activation='relu')
Dense(125, activation='relu')
Dense(2, activation='softmax')
```

NIVEL 5 input size (130,) epoch: 100 batch size: 32

```
Conv1D(65, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Conv1D(130, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Dropout(0.5)
Dense(130, activation='relu')
Dense(65, activation='relu')
Dense(2, activation='softmax')
```

NIVEL 6 input size (66,) epoch: 100 batch size: 32

```
Conv1D(33, kernel size=2, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Conv1D(66, kernel size=2, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Dropout(0.25)
Dense(66, activation='relu')
Dense(33, activation='relu')
Dense(2, activation='softmax')
```

NIVEL 7 input size (34,) epoch: 100 batch size: 32

```
Conv1D(17, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Conv1D(34, kernel size=3, activation='relu', padding='same')
MaxPooling1D(pool size=2, strides=1, padding='valid')
Dropout(0.25)
Dense(34, activation='relu')
Dense(17, activation='relu')
Dense(2, activation='softmax')
```

Capítulo 5

Resultados

5.1 Descripción de los resultados

Una vez construidas las 5 redes neuronales correspondientes con los 5 niveles de las curvas de luz después de aplicar la transformada wavelet, vamos a analizar los resultados obtenidos por estos modelos. Para ello, vamos a utilizar los datos de 3 curvas distintas. Cada una es un ejemplo bastante representativo para cada tipo de etiqueta (**CONFIRMADO**, **FALSO POSITIVO** y **CANDIDATO**). En la Figura 5.1 podemos ver las 3 curvas de luz por nivel de descomposición wavelet. Y en la Figura 5.2 observamos los resultados de las predicciones de las 3 curvas anteriores realizadas por cada uno de los 5 modelos.

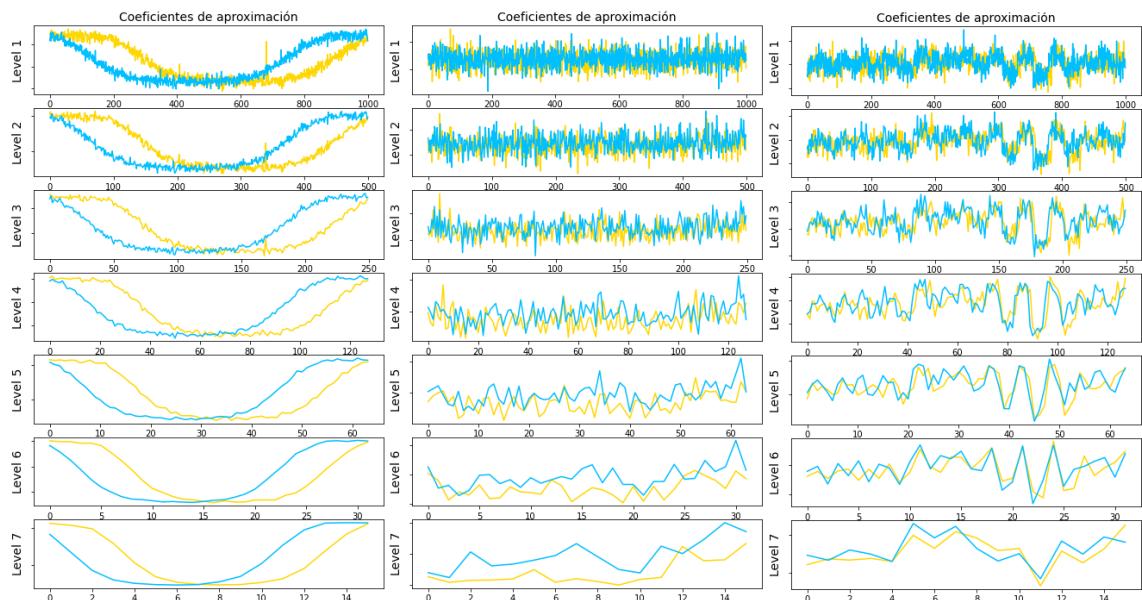


Figura 5.1: Curvas de luz con la descomposición wavelet. A la izquierda exoplaneta confirmado (Kepler ID 3935914), en el centro falso positivo (Kepler ID 11913073) y a la derecha candidato por determinar (Kepler ID 10028127). *Fuente: Elaboración propia.*

| Red neuronal | CONFIRMADO | FALSO POSITIVO | CANDIDATO |
|-----------------------|----------------|----------------|----------------|
| Modelo nivel 3 | [0.854, 0.146] | [0.236, 0.764] | [0.534, 0.466] |
| Modelo nivel 4 | [0.899, 0.101] | [0.332, 0.668] | [0.478, 0.522] |
| Modelo nivel 5 | [0.926, 0.074] | [0.375, 0.625] | [0.573, 0.427] |
| Modelo nivel 6 | [0.954, 0.046] | [0.423, 0.577] | [0.503, 0.497] |
| Modelo nivel 7 | [0.934, 0.066] | [0.452, 0.548] | [0.765, 0.235] |

Figura 5.2: Resultado de la predicción de las 3 curva anteriores. *Fuente: Elaboración propia.*

Para la curva de la derecha, perteneciente al exoplaneta confirmado con Kepler ID 393591, podemos ver un patrón claro de disminución de flujo perteneciente al tránsito del exoplaneta. Los niveles 1 y 2 parecen tener cierto ruido debido a las pequeñas variaciones constantes que se presentan a lo largo de la curva. Pero, a partir del nivel 3, parece que este tipo de ruidos se va disminuyendo hasta desaparecer casi por completo en el nivel 5. En cuanto a los resultados de la predicción, parece que la red neuronal ha identificado, de manera muy satisfactoria las características de este tipo de curvas por lo que ha realizado una predicción correcta en los 5 modelos.

Para el caso de la curva del centro perteneciente al falso positivo, podemos observar que las curvas de los niveles 1 y 2 parecen totalmente caóticas, sin ningún patrón fácilmente entendible a primera vista. Por lo que parece correcta la decisión de descartar estos dos niveles para la elaboración de los modelos. Los niveles 3, 4 y 5 todavía presentan variaciones irregulares constantes, sin características destacables evidentes. Ya los niveles 6 y 7 presentan variaciones menos abruptas, presentando alguna disminución del flujo destacable que se podría interpretar como la presencia de un posible tránsito. En cuanto a los resultados podemos ver que parecen acordes a lo observado en las curvas de luz, las predicciones de los 3 primeros modelos parecen bastante concluyentes. Sin embargo, los resultados de los modelos de los niveles 6 y 7 parecen algo más equilibrado, acercándose al valor de 0.5 en ambos casos. Esto significa que las dos redes neuronales no tienen claro si los patrones que presentan estas dos curvas pertenece al tránsito de exoplaneta o no. Lo cual coincide con lo que hemos observado en la Figura 5.1. La ventaja que se presenta en este caso es que, al considerar más niveles de descomposición wavelet, como ya hemos comentado, en los niveles 3, 4 y 5 de esta curva, no se presentan características que pueden confundirse con el tránsito de un exoplaneta. Por lo que sirven, en algunos casos, para evitar realizar predicciones erróneas de falsos tránsitos de exoplanetas.

Y por último, la curva de la derecha, que corresponde con la etiqueta de **CANDIDATO**. En los niveles 3, 4, 5 y 6, a parte de las pequeñas variaciones de flujo que también se han observado en los dos casos anteriores, podemos ver que existen varios intervalos en cada nivel en los que parece indicar una disminución del flujo que pueda corresponder al tránsito de un exoplaneta. Mientras que en el nivel 7, solo queda una disminución notable

del flujo en el punto 11 del eje X. Si nos fijamos en los resultados de la predicción, esto no parece arrojar claridad, los 4 primeros modelos parecen poco concluyentes, situándose, en los 4 casos, sobre el punto medio. La única red que parece tener un resultado claro es la correspondiente al nivel 7, con un 76,5% de posibilidad de que pertenezca a un exoplaneta confirmado. Por lo que finalmente el modelo de redes neuronales concluye, con un 80% de probabilidad de que sea un falso positivo.

Usando el criterio del cálculo de la probabilidad final comentado en el apartado 4.5.4, las probabilidades que se obtienen son:

$$P_{\text{final}}(C_{\text{confirmado}}) = \frac{1 + 1 + 1 + 1 + 1}{5} = 1$$

$$P_{\text{final}}(C_{\text{falso_positivo}}) = \frac{0.236 + 0.332 + 0.375 + 0.423 + 0.452}{5} = 0.364$$

$$P_{\text{final}}(C_{\text{candidato}}) = \frac{0.534 + 0.478 + 0.573 + 0.503 + 0.765}{5} = 0.570$$

Se puede confirmar que las probabilidades finales obtenidas coinciden con las observaciones que hemos realizado anteriormente. Por lo que, a priori, parece ser que los modelos construidos funcionan satisfactoriamente para estas 3 curvas de luz. Sin embargo, no podemos generalizar este resultado pues estos 3 casos están elegidos estratégicamente para poder destacar sus características propias. Por lo que, para evaluar mejor los modelos construidos, veamos las métricas correspondientes.

5.2 Métricas de evaluación

5.2.1 Métricas empleadas

Las dos métricas que vamos a emplear para evaluar los modelos son:

- **accuracy:** El accuracy o la exactitud es una métrica que se emplea en los modelos de clasificación. Mide la relación entre los casos en los que el modelo ha acertado en la predicción en relación con los casos totales como se indica a continuación:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

siendo TP (true positive) el número de casos confirmados acertados, TN (true negative) el número de falsos positivos acertados, FN (false negative) el número de casos en los que se han etiquetado los confirmados como falsos negativos y FP (false positive) el número de casos de falsos positivos etiquetados como confirmados.

Esta métrica, a pesar de ser una de las más usadas para la evaluación, tiene sus desventajas. Concretamente, esta metrica puede ser engañosa si los datos de entrada que tenemos de las distintas clases está desbalanceada. Por ejemplo, si de las 1000 entradas que tenemos, 999 pertenecen a falsos positivos y 1 a exoplaneta confirmado. Entonces, si el modelo devuelve, como resultado de predicción, que la curva analizada es un falso positivo, entonces el accuracy va a ser casi un 1, pero esto no indica que el modelo sea bueno. Es por esto que, en la sección 4.4.4, hemos elegido los datos de entrada de manera equilibrada (500 confirmados y 500 falsos positivos).

- **loss function:** Una loss function o función de pérdida evalúa la desviación que existe entre los valores reales de las observaciones y los valores estimados de las predicciones realizadas por la red neuronal. Por lo que cuanto menor valor de **loss**, más eficiente es la red neuronal.

La función de pérdida usada en las redes neuronales de este trabajo es la **Cross-Entropy Loss**. Esta función, usada en las tareas de clasificación multiclases, se define como:

$$CE = - \sum_{i=1}^n t_i \log(p_i) \quad (5.2)$$

donde n es el número de las clases que hay, t_i corresponde al *groundtruth* de la clase $i \in C$ y p_i representa la probabilidad Softmax de la clase i en la CNN.

Para este caso particular de clasificación binaria, la **Binary Cross-Entropy Loss** para $n = 2$ se define como:

$$CE = -t_1 \log(s_1) - t_2 \log(s_2) \quad (5.3)$$

Como tenemos únicamente dos clases, podemos tomar $t_1 = t$ y $t_2 = 1 - t$. Y sabemos que en la función Softmax se tiene que cumplir que la suma de las probabilidades sea 1. Entonces, $p_1 + p_2 = 1 \Rightarrow p_1 = 1 - p_2$. Por lo que si tomamos $p_1 = p$ tenemos $p_2 = 1 - p$ y:

$$CE = -t \log(p) - (1 - t) \log(1 - p) \quad (5.4)$$

5.2.2 Métricas obtenidas

Después de observar varios entrenamientos, los resultados obtenidos con las métricas detalladas en el apartado anterior han sido más o menos regulares. En la Figura 5.3 podemos ver representadas las curvas de **accuracy** de cada uno de los modelos representados con el color rojo y las curvas de **loss** representadas en color azul.

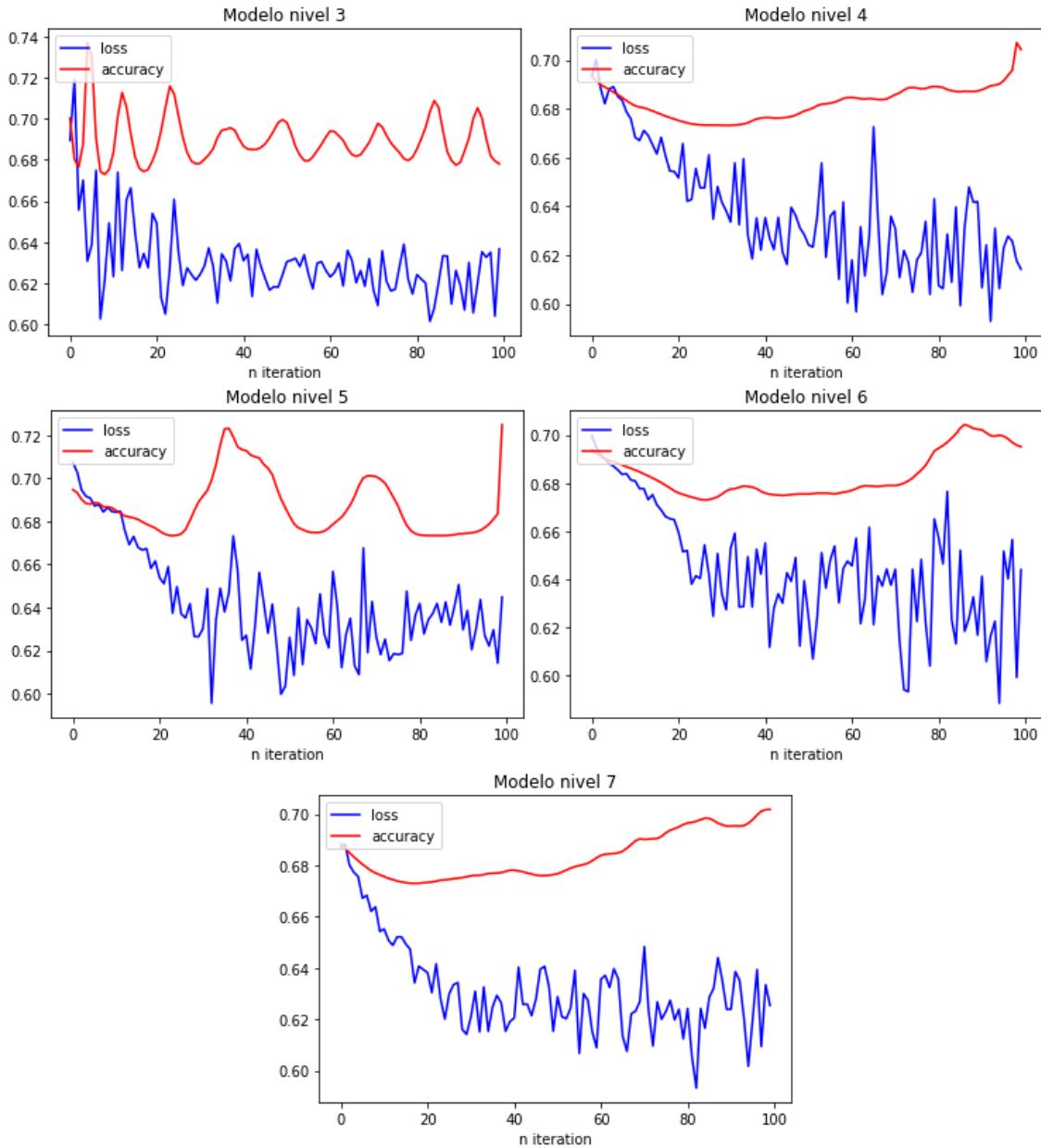


Figura 5.3: Curvas de **accuracy** (rojo) y **loss** (azul) para cada uno de los modelos. *Fuente: Elaboración propia.*

En general se observa un descenso en la **loss** a medida que avanza el entrenamiento mientras que las curvas de **accuracy** presentan una leve subida a medida que crecen las epoch. Sin embargo, dichas variaciones no son muy grandes. En la mayoría de los casos las curvas de **loss** suele presentar una disminución de 0.1 durante el entrenamiento mientras que el **accuracy** presenta únicamente un leve aumento de entre 0.1 y 0.2 puntos.

Este resultado parece jugar en contra de los resultados que hemos analizado en el apartado 5.1. Sin embargo, esto no es cierto. Como ya hemos comentado en 5.1 cuando hemos

analizado las curvas, dichas curvas de luz fueron escogidas meticulosamente para poder entender e interpretar los resultados fácilmente. Pero, en la práctica las cosas no son tan sencillas como lo comentaremos en el capítulo siguiente.

5.3 Análisis de conjuntos de curvas adicionales

5.3.1 Conjunto de curvas test

Disponemos de un total de 100 curvas para el conjunto de test. Siguiendo las reglas del apartado 4.5.4, hemos obtenido los siguientes valores:

| | | Resultados | |
|------------|----------|------------|----------|
| | | Positivo | Negativo |
| Predicción | Positivo | 33 | 12 |
| | Negativo | 17 | 38 |

Figura 5.4: Matriz de confusión de los resultados de las 100 curvas de test. *Fuente: Elaboración propia.*

En la Figura 5.4, observamos, siguiendo las normas del apartado 4.5.4, los resultados de las predicciones realizadas por el modelo final de redes neuronales. Por ejemplo, el **accuracy** obtenido es de:

$$\text{accuracy} = \frac{33 + 38}{100} = 0.71 \quad (5.5)$$

Intuitivamente, los resultados obtenidos parecen encajar con las métricas obtenidas en la Figura 5.3.

5.3.2 Predicción del conjunto de curvas candidatas

Análogamente, hemos realizado el análisis de las 100 curvas con la etiqueta **CANDIDATO** y hemos obtenido, siguiendo las reglas del apartado 4.5.4, que 38 de ellas son positivas y 62 son negativas. No se sabe con certeza las verdaderas etiquetas de estas curvas de luz. Por lo que no podemos analizar las métricas para este conjunto de curvas.

Sin embargo, si tomamos en cuenta los números de las curvas de cada tipo expuestas en el punto 4.4.1, podemos ver que la relación de curvas positivas con la del total es:

$$\% \text{positivos} = \frac{\text{Curvas positivas}}{\text{Total}} = \frac{966}{2181} * 100 = 44.3\% \quad (5.6)$$

Y en nuestro caso particular para la predicción de las curvas candidatas, hemos obtenido que el 38% son positivas. Lo cual puede ser una indicación de que no va mal encaminado.

Sin embargo, hay que tener en cuenta que dichos porcentajes que hemos visto, al ser una población muy baja (2181 elementos únicamente), solamente lo podemos tomar como una aproximación de los posibles resultados, sin indicar, bajo ningún concepto, un resultado firme sobre la relación del número de exoplanetas con el total de los objetos analizados.

Capítulo 6

Conclusiones y líneas de trabajos futuros

Cuando todo parece ir bien encaminado cuando estás trabajando en el modelo. Sin embargo, a la hora de analizar los resultados, no todo sale como uno espera. Esta frase describe muy bien la mayoría de los trabajos que se realizan en este campo, incluyendo el presente. Pero, a diferencia de algunos en los que uno no logra encontrar la explicación. Parece que, en este caso, sí lo hay y es fácil de aclararlo.

El punto que hay que tener en cuenta es que no todas las curvas etiquetadas como **CONFIRMADO** tienen el aspecto de la curva de luz del exoplaneta con Kepler ID 3935914 en el que se aprecia un claro descenso en el flujo debido al tránsito. Ni todas las curvas etiquetadas como **FALSO POSITIVO** tiene un aspecto tan irregular como la curva de luz con Kepler ID 1913073 mostrada en la misma Figura 5.1.

Por ejemplo, en la Figura 6.1 se observa a la izquierda las curvas de luz del falso positivo con Kepler ID 10419211. Sin embargo, si observamos con detalle dichas curvas, podemos observar que, a priori, cumplen justamente con las características descritas para las curvas de luz de un exoplaneta. Mientras que en la derecha, podemos observar las curvas de luz del exoplaneta confirmado con Kepler ID 7935997. En dichas curvas de luz se pueden observar una irregularidad constante en los flujos a lo largo del tiempo. Pero no se observa en ningún momento una disminución evidente del flujo que correspondería con el tránsito del exoplaneta.

Y hay que tener en cuenta que estos casos no son aislados, sino que, tras una revisión de las 100 primeras curvas de luz de cada tipo, hay numerosos casos bastante confusos que pueden llevar a una interpretación errónea por las redes neuronales.

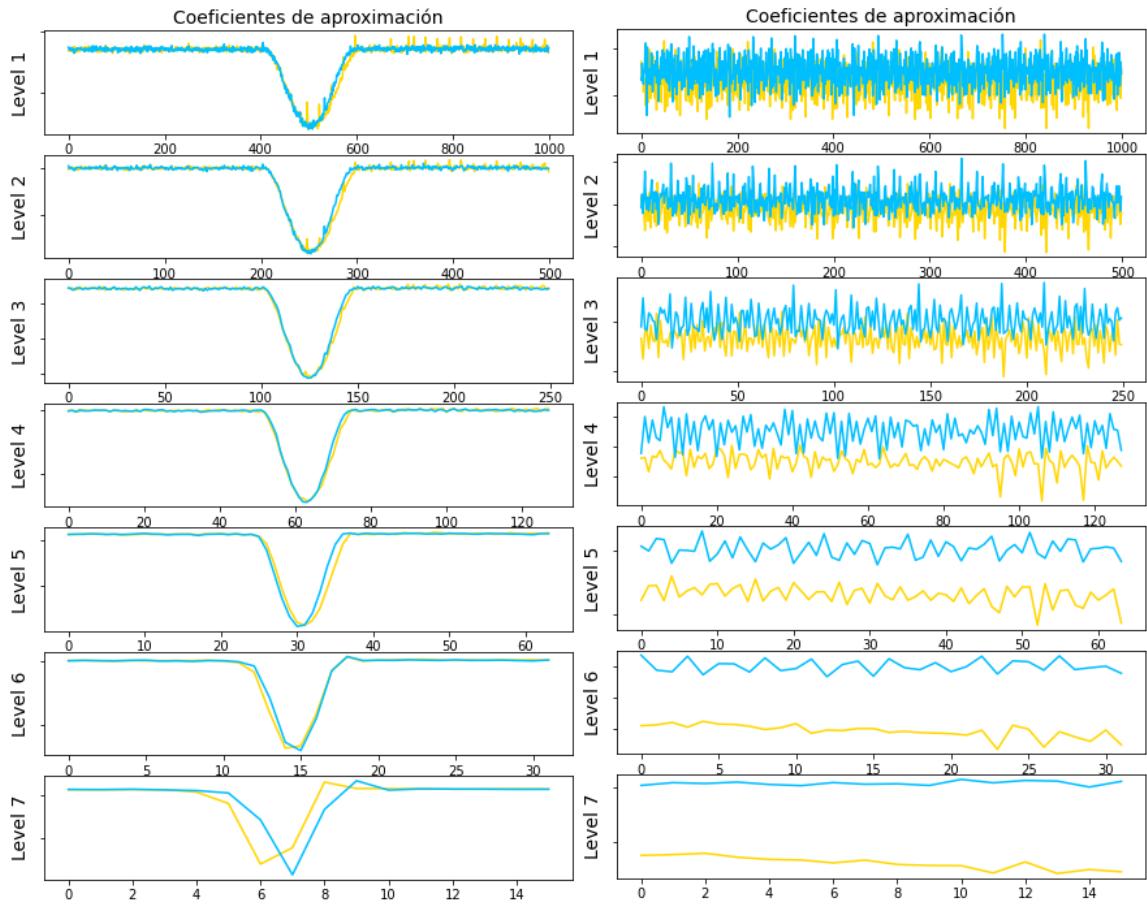


Figura 6.1: Curva de luz del falso positivo con Kepler ID 10419211 (a la izquierda) y exoplaneta confirmado con Kepler ID 7935997 (a la derecha). *Fuente: Elaboración propia.*

Dicho esto, vamos a analizar otros aspectos de este trabajo:

- Uno de los puntos que más hay que tener en cuenta es que, dada la gran complejidad que representan estos datos de entrada por la presencia de curvas como las dos de la Figura 6.1, creo que el número de entradas que se ha utilizado para entrenar estos modelos resulta ser bastante pequeño. Se podría plantear usar el dataset completo de la misión de Kepler (9564 curva de luz en total) si la capacidad de cómputo lo permite.
- El siguiente punto a comentar es la aplicación de la transformada wavelet a las curvas de luz. Hemos visto que, en algunos casos, las descomposiciones wavelet de las curvas de luz parece bastante bueno (por ejemplo, la Figura 4.14). Pero hay otros caso en los que no se aclaran mucho la situación (como la Figura 6.1). Es interesante poder separar dichos casos “problemáticos” para tratarlos aparte y poder encontrar una solución asequible con la transformada wavelet para poder interpretarlo mejor.
- Por otro lado, también hay que tener en cuenta los modelos de redes neuronales en sí. Creo que este punto va ligado a la cantidad de datos de entrada. Al no haber

cantidad de datos suficientes, los modelos de redes neuronales puede no haberse entrenado de la manera correcta (y de hecho, es más bien una afirmación). Con el aumento del dataset, se puede plantear una revisión en las estructuras de las capas de los modelos para mejorar las métricas de evaluación.

- También es interesante estudiar la eficiencia de los modelos usando otras métricas adicionales como el recall o la F1 score.
- Por último, creo que es interesante plantear métodos de normalización adicionales a la aplicada para analizar las diferentes posibilidades que se puede obtener.

Como conclusión final, comentar que se han cumplido con los objetivos fijados al inicio del trabajo. Pero los resultados obtenidos no han sido tan buenos como los esperados por diversas razones comentadas anteriormente. Pero, por la falta de recursos computacionales y la limitación del tiempo, no ha sido posible continuar con el desarrollo del presente trabajo para tratar los puntos comentados. Por lo que es de especial interés poder continuar con esta línea de investigación de cara a futuros trabajos.

Bibliografía

- [1] Alexander Poffo, D. (2012). *Determinación de la zona de habitabilidad*. Universidad Nacional de Córdoba. <https://docplayer.es/14671899-Determinacion-de-la-zona-de-habitabilidad.html>
- [2] Alonso Sobrino, R. (2006). *Detección y caracterización de exoplanetas mediante el método de los tránsitos*. Universidad de La Laguna. <https://dialnet.unirioja.es/servlet/tesis?codigo=19988>
- [3] Bonse, M. J., Quanz, S. P. y Amara, A. (2021). *Wavelet based speckle suppression for exoplanet imaging. Application of a de-noising technique in the time domain*. Institute for Particle Physics and Astrophysics. <https://doi.org/10.48550/arXiv.1804.05063>
- [4] Brennan, P. (2 de abril de 2021). *What is an exoplanet?*. <https://exoplanets.nasa.gov/what-is-an-exoplanet/>
- [5] Cofield, C., Brennan, P. y Hawkes, A. (25 de octubre de 2018). *Rocky? Habitable? Sizing up a Galaxy of Planets*. <https://www.nasa.gov/feature/jpl/rocky-habitable-sizing-up-a-galaxy-of-planets>
- [6] Cubillos, P., Harrington, J., Loredo, T. J., Lust, N. B., Bleicic, J. y Stemm, M. (2016). *On correlated-noise analyses applied to exoplanet light curves*. The Astronomical Journal 153(1), pág 1-14. <https://doi.org/10.3847/1538-3881/153/1/3>
- [7] García Crespo, R. (2021). *Búsqueda y detección de exoplanetas mediante técnicas de Machine Learning*. Universidad Politécnica de Valencia. <http://hdl.handle.net/10251/174964>
- [8] Gómez-Martínez, Y. (2016). *Configurando narrativas históricas y preguntas directrices para un abordaje sistémico sobre la Revolución Copernicana*. Universidad de São Paulo. <https://doi.org/10.11606/D.48.2016.tde-29042016-110915>
- [9] González González, R. A. (2010). *Algoritmo basado en Wavelets aplicado a la detección de incendios forestales*. Universidad de las Américas Puebla. http://catarina.udlap.mx/u_dl_a/talles/documentos/mel/gonzalez_g_ra/

- [10] Guirado Fuentes, L. (2022). *Análisis de curvas de luz Kepler mediante la transformada wavelet*. Universidad Internacional de la Rioja. <https://reunir.unir.net/handle/123456789/14113>
- [11] Li, J., Tenenbaum, P., Twicken, J. D., Burke, C. J., Jenkins, J. M., Quintana, E. V., Rowe, J. F. y Seader, S. E. (2019). *Kepler Data Validation II–Transit Model Fitting and Multiple-planet Search*. Astronomical Society of the Pacific, 131(1), pág 16-20. <https://doi.org/10.1088/1538-3873/aaf44d>
- [12] Mallonn, M., Poppenhaeger, K., Granzer, T., Weber, M. y Strassmeier, K. G. (2022). *Detection capability of ground-based meter-sized telescopes for shallow exoplanet transits*. Astronomy & Astrophysics, 657(1), pág 1-10. <https://doi.org/10.1051/0004-6361/202140599>
- [13] Martín Martín, L. (2019). *Introducción a la teoría de wavelets. Construcción y propiedades de wavelets continuas y discretas*. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/38198>
- [14] Masciadri, E. y Raga, A. (2004). *Exoplanet Recognition Using a Wavelet Analysis Technique*. The Astrophysical Journal 611(2), pág 137-140. <https://dx.doi.org/10.1086/423984>
- [15] Piñón Gaytán, F. (1997). *El pensamiento filosófico de Giordano Bruno. El hombre como el deus creatus*. Iztapalapa: Revista de Ciencias Sociales y Humanidades, 41(1), pág 155-166. <https://dialnet.unirioja.es/servlet/articulo?codigo=7646021>
- [16] Sánchez Sánchez, V. (2019). *Detección de exoplanetas en sistemas binarios*. Universidad de la Laguna. <https://riull.ull.es/xmlui/handle/915/16255>
- [17] Walbolt, K. (21 de marzo de 2022). *Historic timeline: first exoplanets discovered*. <https://exoplanets.nasa.gov/alien-worlds/historic-timeline>

Apéndice A

Artículo científico