Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Detección de tránsitos de exoplanetas mediante técnicas de *deep learning*

Autor: Alejandro Casal Argüelles

Tutor: Laura Ruiz Dern

Profesor: Jordi Casas Roma

Barcelona, 4 de junio de 2021

FICHA DEL TRABAJO FINAL

Título del trabajo:	Detección de tránsitos de exoplanetas mediante técnicas de *deep learning*
Nombre del autor:	Alejandro Casal Argüelles
Nombre del colaborador/a docente:	Laura Ruiz Dern
Nombre del PRA:	Jordi Casas Roma
Fecha de entrega:	Junio de 2021
Titulación o programa:	Máster Universitario en Ciencia de Datos *(Data Science)*
Área del Trabajo Final:	Campos especializados
Idioma del trabajo:	Español
Palabras clave	Detección de exoplanetas, *deep learning*, misión Kepler-K2

–Las estrellas son bellas por la flor que no se ve...

Antoine de Saint - Exupéry, El Principito

Agradecimientos

Tiempo, tiempo, tiempo... A mi mujer, Goretti, por todo el tiempo que me ha regalado; y a mis peques, Laura y Diego, por todo el tiempo que les he robado. Sin su apoyo no hubiera podido siquiera plantearme empezar este máster que ahora culmina.

Mil gracias también a mi tutora Laura, por su disponibilidad y sus siempre amables y acertados consejos.

Y como no, mi sincero agradecimiento a todas las instituciones que ofrecen tanta información y conocimiento abierto a la comunidad, sin los cuales este trabajo no hubiera sido posible.

Resumen

El presente trabajo tiene como finalidad la preparación de algoritmos automáticos, con técnicas de *deep learning*, para detectar exoplanetas a partir de los datos recogidos en la misión K2 de la NASA.

Esta misión, heredera de la misión Kepler, recuperó, de multitud de estrellas, datos de su luminosidad a lo largo del tiempo, en lo que se denomina curvas de luz. Disminuciones en la luminosidad aparente podrían indicar un tránsito planetario frente a la estrella, a la cual ocultaría parcialmente. Este es uno de los métodos actuales más exitosos para la detección de exoplanetas.
Debido a problemas en el equipamiento, la misión K2 sólo recogió información de periodos de unos 80 días, en diferentes sectores del cielo, a diferencia de la misión original Kepler, que recogió, de un único sector, datos de varios años. Así, el análisis de datos de la misión K2 presenta una dificultad importante al ser extremadamente difícil registrar un mismo tránsito planetario varias veces.

Los tratamientos clásicos para la determinación de presencia de exoplanetas, se basan en un preprocesado inicial de las curvas de luz para detectar disminuciones temporales en la luminosidad, es decir, posibles tránsitos, y su posterior análisis, como entrada a modelos predictivos, para determinar si esas disminuciones de la luminosidad están asociadas a un exoplaneta o no.

En este trabajo se plantea la utilización de la totalidad de los datos de cada curva de luz, sin el mencionado preprocesado y extracción previo de posibles tránsitos, en un enfoque *end-to-end*. Ello implica una dificultad añadida pero por otro lado debería permitir un análisis más temprano de la información que pueda estar disponible, por ejemplo, extrapolando los métodos a datos de misiones actualmente en curso, como TESS.

Palabras clave: Detección de exoplanetas, *deep learning*, misión Kepler-K2

Abstract

The aim of this work is to prepare automatic algorithms, with deep learning techniques, to detect exoplanets from the data collected by NASA's K2 mission.

This mission, heir to the Kepler mission, retrieved data on the luminosity of a multitude of stars over time, in what are called light curves. Decreases in apparent luminosity could indicate a planetary transit in front of the star, which it would partially obscure. This is one of the most successful current methods for exoplanet detection.
Due to equipment problems, the K2 mission only collected data for periods of about 80 days, in different sectors of the sky, unlike the original Kepler mission, which collected data for several years from a single sector. Thus, the analysis of the K2 mission data presents a major difficulty as it is practically impossible to record the same planetary transit several times.

The classical treatments for determining the presence of exoplanets are based on an initial preprocessing of the light curves to detect temporary decreases in luminosity, i.e. possible transits, and their subsequent analysis, as input to predictive models, to determine whether these decreases in luminosity are associated with an exoplanet or not.

In this work we propose to use all the data from each light curve, without the aforementioned pre-processing and extraction of possible transits, in an end-to-end approach. This implies an added difficulty, but on the other hand it should allow an earlier analysis of the information that may be available, for example by extrapolating the methods to data from ongoing missions, such as TESS.

Keywords: Exoplanet detection, *deep learning*, Kepler-K2 mission

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La detección de planetas más allá de nuestro sistema solar, es decir, los llamados exoplanetas, es un ámbito de creciente interés, tanto científico como popular, desde que fue descubierto el primero en 1995. Gracias a ellos podemos comprender mejor el Sistema Solar, y el universo en general, y como no, poder detectar la posible existencia de planetas de características similares a la Tierra, con condiciones propicias para la aparición de vida.

1.1. Descripción general del problema

En 2009 la Nasa lanzó al espacio el satélite Kepler con el objetivo principal de la detección de planetas extrasolares (exoplanetas), principalmente mediante la técnica de tránsito. Esta técnica consiste en la detección de variaciones en la luminosidad observada en una estrella. Un planeta, al orbitar sobre una estrella puede provocar una mínima ocultación (disminución del brillo de la estrella), que podría ser detectado, tal como se muestra en la figura 1.1, siempre y cuando esto ocurra en el mismo plano de la observación.

Así, la misión Kepler estuvo varios años apuntando a una región muy concreta del cielo, en la constelación de Cygnus, monitorizando cambios en el brillo de miles y miles de estrellas. Todas estas observaciones fueron capturadas en forma de curvas de luz que podían ser analizadas (medidas del brillo de la estrella a lo largo del tiempo).
Sobre esta información se aplicaron procesos automáticos para identificar eventos de señales periódicas de disminución de luminosidad que pudieran ser consistentes con un exoplaneta. Estas señales son conocidas como eventos de cruce de umbral o TCEs (*Threshold Crossing Events*), generándose, un catálogo de TCEs, puesto a la disposición de la comunidad científica (Thompson et al. 2018). [7]

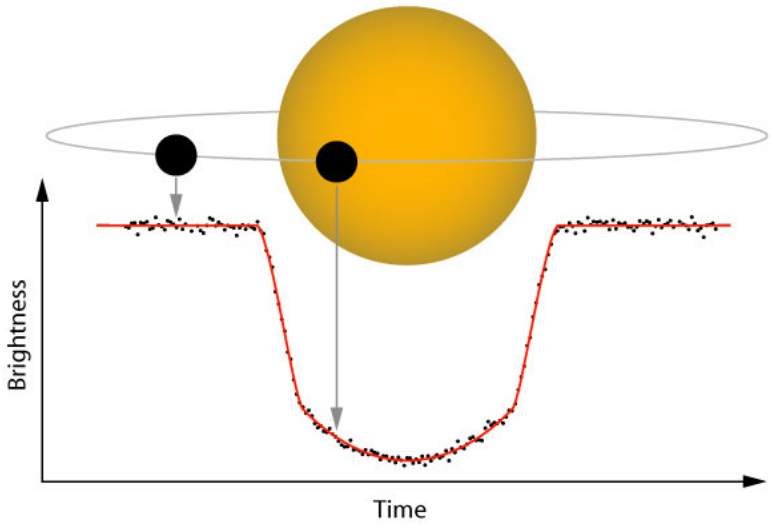


Figura 1.1: Tránsito planetario - Curva de luz
[25]

La gran mayoría de procesos de *machine learning* existentes para la detección de exoplanetas mediante la técnica de tránsito se aplican directamente sobre esta información, es decir, TCEs ya preseleccionados, con el objetivo de identificar cuáles de ellos son efectivamente exoplanetas o cuáles no (Shallue and Vanderburg, 2018) [32].

La duración de la misión original Kepler permitió identificar miles de TCEs y, especialmente, la periodicidad de dichas señales, pues ofrecen una información muy relevante para la caracterización de los exoplanetas. No obstante, en 2013 el telescopio Kepler empezó a fallar en varios de sus componentes, lo que le impedía mantener su orientación con precisión. En este punto surgió la misión K2, para aprovechar las capacidades aún operativas del equipo. K2 permitió apuntar a diferentes sectores del cielo, siguiendo el plano de la eclíptica, a lo largo del tiempo, pero por periodos mucho más cortos, de tan solo 80 días. En la figura 1.2 se muestra el funcionamiento de la misión K2.

De esta misión, como en la misión original, se liberaron a la comunidad científica las curvas de luz capturadas de cada estrella analizada en diferentes campañas, cada una de un sector específico. No obstante, se descartó la aplicación de procesos automáticos de detección de TCEs para generar un catálogo equivalente al de la misión Kepler inicial, pues con exploraciones de tan solo 80 días es extremadamente difícil detectar repeticiones de TCEs (por contextualizar, indicar que Mercurio tiene un año de 88 días).

Sobre los datos facilitados por la misión K2 se han realizado diversos estudios, también uti-

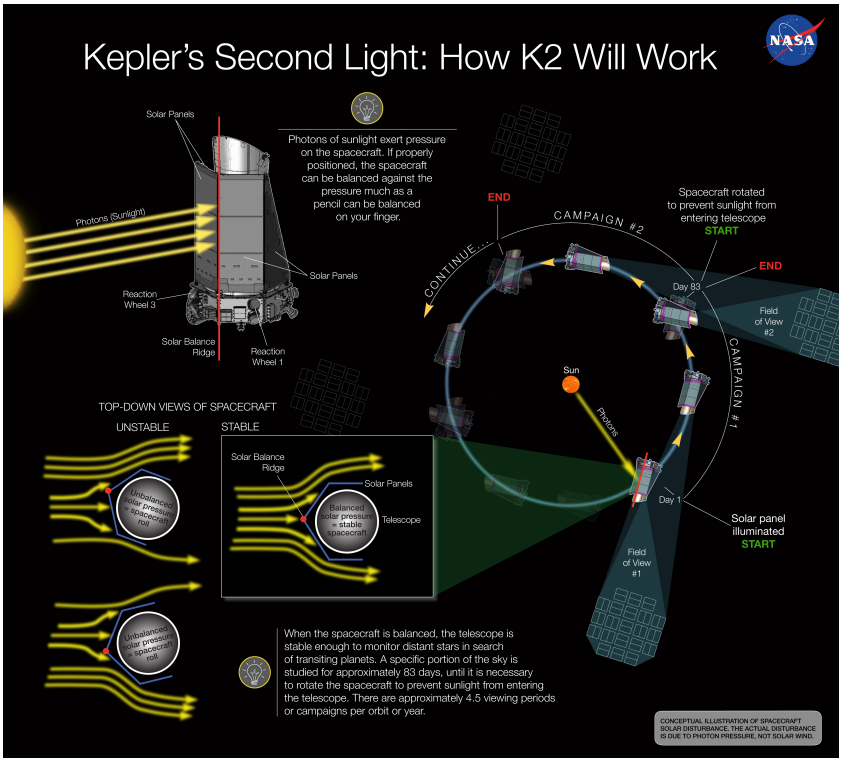


Figura 1.2: Funcionamiento Kepler - K2
[23]

lizando procesos de *machine learning* pero emulando los procesos de la misión Kepler original, es decir, detectando primero en las curvas de luz posibles TCEs, y luego, sólo con estos (no toda la curva sino sólo la parte correspondiente al posible tránsito del exoplaneta), aplicar las técnicas de *machine learning* (Dattilo et al. 2019) [6].

El objetivo del presente TFM es aplicar técnicas de *deep learning* para detectar exoplanetas directamente a partir de los datos de curvas de luz disponibles de las campañas de la misión K2, sin identificación previa de TCEs, y utilizando exclusivamente datos de curvas de luz disponibles para la comunidad a través Mikulsky Archive for Space Telescopes (MAST [19]).
Ello debería poder facilitar el análisis de curvas de luz de periodos temporales más limitados de un modo más ágil.

1.2. Motivación

Informático de formación y profesión, siempre me ha gustado la astronomía, tanto la observacional (aunque poco más de reconocer constelaciones a simple vista), como en un ámbito más formal, en base a lectura de libros de astronomía de diversa índole y siguiendo algún curso

en plataformas MOOC como Coursera, MiriadaX, etc. Así, me pareció muy interesante poder combinar la ciencia de datos con la astronomía y más en un ámbito donde la ciencia de datos se ha mostrado muy relevante en el descubrimiento de nuevos exoplanetas.

Este TFM me permitirá en primer lugar el conocer qué avances se han realizado en este ámbito de detección de exoplanetas, así como el intentar plantear métodos complementarios y/o alternativos a los actualmente existentes.
Me interesaba también el plantear un proyecto TFM que recogiera todo el ciclo de vida de los datos, desde la preparación de los *datasets* para el análisis, hasta la obtención de resultados finales.

1.3. Objetivo

1.3.1. Objetivo principal

Poder determinar si a partir de una curva de luz de una estrella, es decir, la luminosidad aparente de una estrella a lo largo de un periodo de tiempo, y únicamente con esa información, si existe un exoplaneta orbitando a la misma.

1.3.2. Objetivos parciales

- Analizar los datos de la misión K2 disponibles para la comunidad, entenderlos y con ellos preparar un conjunto de datos adecuado para la realización del estudio.
- Analizar los métodos normalmente aplicados en la detección de exoplanetas, aunque utilicen sólo una parte de los datos (únicamente los eventos de disminución de luminosidad), como pueden ser redes CNN (redes convolucionales), etc, al tiempo que analizar qué métodos podrían aplicarse, tanto considerando los datos como de forma global o como una serie temporal, donde por ejemplo podrían encajar redes RNN (redes recurrentes).
- Aplicar las técnicas de *deep learning* identificadas, compararlas entre ellas para identificar los mejores resultados, y con los datos obtenidos validar la hipótesis principal, contrastando en este caso los resultados con estudios previos que hayan aplicado otras técnicas, en lo relativo a la precisión alcanzada en la identificación de exoplanetas.
- Identificar cómo extrapolar/reutilizar los métodos con mejores resultados a otros datos de curvas de luz de otras misiones distintas a K2, que puedan tener, por ejemplo, otros periodos o frecuencias en las muestras temporales disponibles.

1.4. Metodología

Como la gran mayoría de los proyectos de ciencia de datos la investigación a realizar en el TFM tiene una triple vertiente: obtención y preparación de los datos, métodos a aplicar sobre los mismos, y finalmente, comparación y puesta en contexto de los resultados obtenidos.

En relación a los datos se utilizarán los disponibles en el archivo MAST (Mikulsky Archive for Space Telescopes [19]). En dicho repositorio se encuentran datos de las diferentes curvas de luz de todas las estrellas monitorizadas en la misión K2. Señalar que varios equipos de la comunidad internacional, en base a los datos originales han preparado versiones alternativas de dichos datos, aplicando diversos filtros, como por ejemplo para reducir el ruido presente.
Se plantea utilizar los datos originales de la misión y al menos otros dos conjuntos de datos equivalentes filtrados, para en fase de análisis de resultados poder analizar también qué conjunto de datos presenta mejores resultados. Con todos estos datos se construirá un *dataset* de curvas de luz, y con información disponible de otras fuentes (como el archivo de exoplanetas de Caltech [4]), etiquetar cada curva de luz indicando si presenta, o no, exoplanetas.
Este *dataset* se creará de manera automatizada, es decir, a partir de catálogos que indican si estrellas monitorizadas en la misión K2 tienen o no exoplanetas, recuperar las curvas de luz asociadas, y complementarlas con otras curvas de luz de estrellas sin exoplanetas, para así componer un *dataset* consistente. Este será el punto de partida para las etapas posteriores, siendo la primera de ellas el preprocesado de los datos (tratamiento de *outliers*, valores nulos, normalización, etc.).

En cuanto a los métodos se aplicarán redes convolucionales CNN sobre las señales (indicar que cada curva de luz se compone de unos 4.000 muestras de la luminosidad de una estrella, tomada a intervalos de 30 minutos, para tener así un periodo de unos 80 días). También, y dada la naturaleza de series temporales que tienen los datos se evaluarán redes RNN, o incluso la aplicación de métodos no supervisados, como Autoencoders, para posteriormente combinarlos con otros métodos de clasificación.

En relación al análisis de resultados y comparativas, dado que casi todos los estudios utilizan un preprocesado previo y ya aplican métodos sobre datos de eventos de disminución de luminosidad, no es factible comparar directamente resultados, si bien ofrecerán una orientación sobre la bondad de los obtenidos. Sí podrá realizarse una comparativa sobre qué datos de origen ofrecen unos mejores resultados con los métodos más óptimos obtenidos.

Para el TFM se utilizará el lenguaje Python, con el entorno Anaconda/Windows y/o Google

Colab, empleando notebooks con Jupyter, utilizando el entorno `Scikit-learn`, y principalmente las librerías `Keras` y `Tensorflow`.

1.5. Planificación

En el siguiente diagrama de Gantt se muestra una propuesta de planificación de las actividades del proyecto:

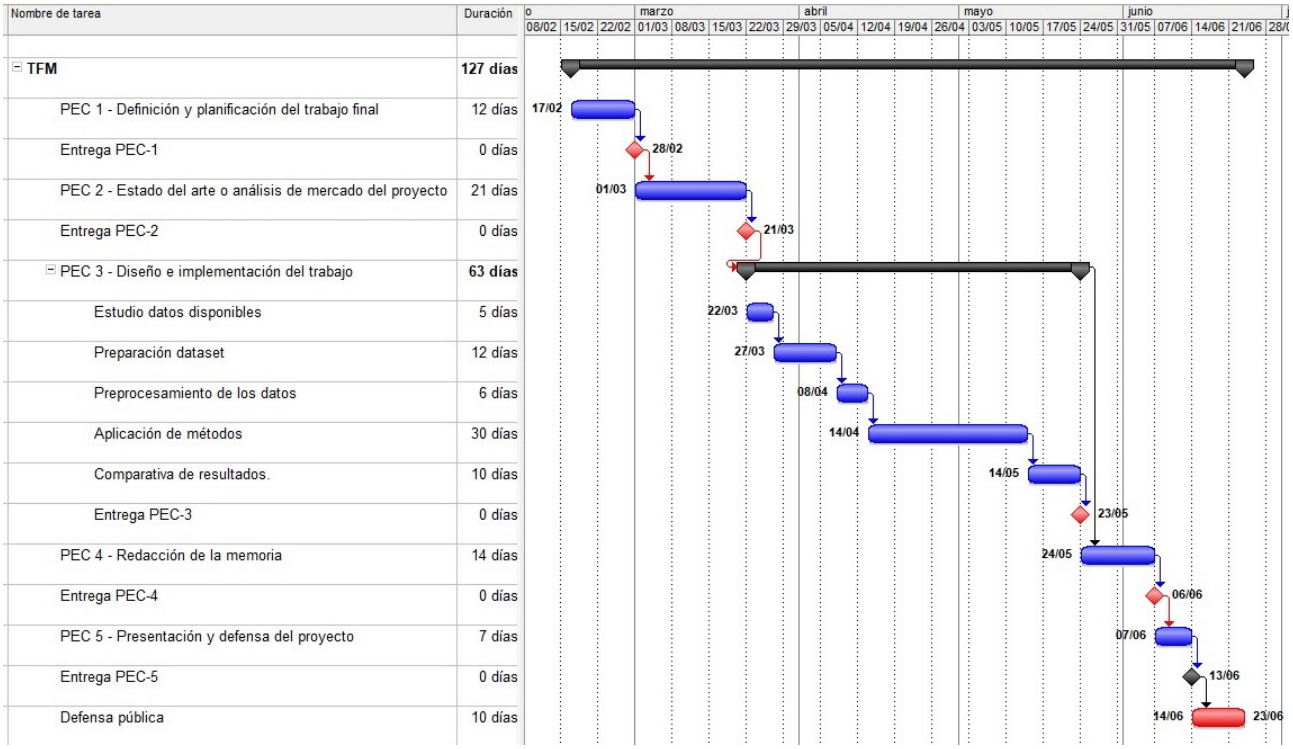


Figura 1.3: Planificación

Capítulo 2

Estado del arte

En este capítulo se presenta el estado del arte en relación a la detección de exoplanetas mediante la técnica de tránsito, donde se suele seguir un proceso bastante estandarizado consistente en las siguiente etapas:

1. Obtención de datos, en forma de curvas de luz.
2. Tratamiento de las curvas de luz
3. Detección de señales de posibles tránsitos
4. Identificación de las señales de tránsitos para determinar si se corresponden con un exoplaneta o no.

En los siguientes apartados se muestran con detalle qué tareas y prácticas más habituales comprenden cada uno de ellos, pero inicialmente, y por contextualizar el trabajo, se mencionan otros tipos de técnicas de detección de exoplanetas existentes, al margen de la técnica de tránsito, si bien éste es el más común y exitoso hasta la actualidad.

2.1. Métodos de detección

Técnica de tránsito

La descripción de la técnica de tránsito puede verse en el apartado 1.1.

Velocidad radial

Consiste en analizar el efecto de desplazamiento Doppler observado en la estrella que pueda ser causado por el efecto de la gravedad entre la misma y un planeta que la orbita. Así, se produce una oscilación o bamboleo de la estrella que se puede medir mediante los desplazamientos

al rojo de la luz (la estrella se aleja) o al azul (la estrella se acerca).
Es un buen método para la detección de planetas gigantes (por la atracción gravitacional que ejercen).

Microlentes gravitacionales

Las trayectorias de la luz están distorsionadas por objetos masivos como estrellas o planetas generando un efecto de lente gravitacional que puede cambiar la dirección de la luz de una estrella. Así, este método se basa en el hecho de que la gravedad de un exoplaneta puede enfocar la luz de estrellas distantes para hacerlas parecer temporalmente más brillantes. El principal inconveniente es la poca probabilidad de alineación requerida para el efecto, y sobre todo el no poder repetir las mediciones, lo cual siempre requerirá de otros métodos de confirmación.

Detección visual directa

Esta técnica consiste en resolver espacialmente el exoplaneta y su estrella anfitriona para obtener imágenes directas. Es la técnica más compleja de todas, pero por otro lado, es la que puede llegar a ofrecer información más relevante de un exoplaneta, como su composición química, temperatura, etc.

2.2. Obtención de datos

Existen, o han existido, diversas misiones para recuperar información, en forma de curvas de luz, de la luminosidad de las estrellas a lo largo de un periodo temporal, destacando entre ellas la misión Kepler (tanto la original como la misión K2), o las misiones TESS (*Transiting Exoplanet Survey Satelliute* [20]), CHEOPS, SuperWASP, entre otras.
No obstante, la gran mayoría de estudios y aplicación de técnicas de *machine learning* y *deep learning* para detección de exoplanetas se basa en datos de la misión Kepler, al ser en esta donde existe mayor número de exoplanetas confirmados, y por lo tanto permite, con mayor facilidad, el entrenamiento de modelos automáticos de detección. Pese a que ya no está operativo, los datos facilitados por el satélite Kepler se siguen analizando y aún hoy en día se descubren nuevos planetas en base a los mismos.

Tanto para la misión Kepler y K2 se obtienen curvas de luz que se almacenan en ficheros (uno por estrella), y que se ponen a disposición de la comunidad en el MAST - *Mikulski Archive for Space Telescopes* [19]. Para cada objeto estelar se disponen de dos curvas, unas con cadencias de observación largas (una muestra cada 29,4 minutos), y otra de cadencia corta (cada 58,8

segundos), si bien en la práctica sólo se utilizan las primeras. Y las muestras se organizan por trimestres, de modo que en un fichero de datos se tienen los flujos de luz de unos 80 a 90 días. En el caso de la misión Kepler se repiten para cada trimestre las mismas estrellas (apuntaba siempre a la misma región del espacio, de modo que se pueden obtener flujos continuos de unos 70.000 puntos), mientras para K2 cada trimestre apuntaba a una región diferente, con lo que los ficheros contienen unas 4.000 muestras.

Estos archivos están formateados como archivos FITS (*Flexible Image Transport System*), el formato de archivo digital más comúnmente utilizado en astronomía, que contienen encabezados que describen el entorno de observación y la calidad de los datos y una tabla de las mediciones de flujo a lo largo del tiempo.
Hay dos valores informados para las mediciones de flujo: flujo de fotometría de apertura simple (SAP) y flujo de SAP de acondicionamiento de datos de búsqueda previa (PDC). Las versiones de PDC SAP de las curvas de luz eliminan parte de las variaciones instrumentales y el ruido en los datos, al tiempo que preservan los posibles tránsitos de los exoplanetas estelares. Por lo tanto, esta es la medida de flujo utilizada para construir las curvas de luz a ser analizadas.

En el caso de la misión K2, debido a la inestabilidad del satélite en su modo de funcionamiento K2, esas curvas de luz en bruto exhiben grandes características sistemáticas que podrían llegar a impedir la detección de tránsitos planetarios, siendo corregidas por algunos equipos descorrelacionando la sistemática variabilidad del movimiento de la nave espacial.
Este sería el caso de las curvas de luz K2SFF (*K2 Extracted Lightcurves*), también disponibles en MAST[19] y generadas por el equipo Dattilo et al. (2019) [6] en base a lo desarrollado por Vanderbug and Johnson (2014) [34], o de las curvas EVEREST (*EPIC Variability Extraction and Removal for Exoplanet Science Targets*) (Luger et al., 2016) [18], K2SC, y otras.

Estas y otras curvas de luz que intentan mejorar las originales pueden obtenerse en el MAST en K2 High Level Science Products[22].

2.3. Tratamiento de las curvas de luz

Como hemos visto en la sección anterior todas las curvas de luz pueden contener anomalías inducidas por los propios elementos de medición, variaciones estelares, ruido debido a valores atípicos, discontinuidades dentro de las curvas de luz, señales débiles, etc. (Jara-Maldonado et al, 2020) [15] que requieren un preprocesado previo, entre los cuales se incluyen los descritos a continuación.

Aplanamiento de la curva

En la mayoría de los casos suele aplicarse inicialmente un aplanamiento de las curvas de luz, siguiendo lo indicado por Shallue and Vanderburg (2018) [32], que consiste en ajustar una *spline* polinomial a los puntos de la curva de luz para posteriormente dividir la curva de luz original por la *spline*. Este procedimiento da como resultado una curva de luz aplanada sin ruido de variabilidad estelar.

Eliminación de valores extremos

Igualmente se eliminan valores extremos (*outliers*), eliminando en un proceso iterativo aquellos puntos que difieren de los adyacentes un número superior a N σ veces (N veces la desviación estándar).

Tratamiento de valores nulos

En cuanto a los valores nulos, no conviene eliminarlos sin más y compactar la curva de luz pues ello podría distorsionar la figura de un posible tránsito. Se aplican entonces interpolaciones entre los valores de los extremos, bien de tipo lineal, o como proponen Hinners et al. (2018) [8], asignando valores aleatorios comprendidos entre dichos valores.

Normalización de datos

Este mismo autor también propone la normalización de los datos, pues los valores de flujo brutos contienen relativamente poca información por sí mismos. Además, con datos de la misión Kepler original, al combinar datos de una misma estrella pero de trimestres distintos, las variaciones, sin normalizar los datos, podrían ser muy significativas y distorsionar completamente cualquier análisis posterior de la curva de luz.
Uno de los métodos comúnmente utilizados es el aplicado por Dattilo et al. (2019) [6], donde escalan los valores para que su mediana sea cero y el valor mínimo sea -1.

Agrupamiento de los datos

Otro tratamiento que es habitual ver en la mayoría de trabajos, especialmente cuando los flujos de luz contienen muchos puntos, es realizar agrupamiento de los datos, bien por ejemplo seleccionando sólo un punto de cada N, o bien agrupando los puntos en grupos de N y quedándose con la mediana de los mismos. Las curvas de luz resultantes mantienen las mismas propiedades y forma que la original.

Data augmentation

No siempre es posible disponer de conjuntos de datos para entrenar los modelos. Así, en diferentes estudios se aplican técnicas de generación de datos sintéticos. Suelen consistir en utilizar curvas de luz que no contienen ningún TCE e inyectar en las mismas TCEs de tránsitos confirmados presentes en otras curvas de luz.
Estas técnicas se aplican por ejemplo en el entrenamiento de métodos que utilizan datos de TESS, donde aún no hay un número significativo de TCEs y sus posibles exoplanetas confirmados. Esto permite trabajar con datos de entrenamiento más balanceados, pues de otro modo, a la complejidad del tratamiento de las curvas de luz se añade el entrenamiento con datos altamente desbalanceados.

2.4. Detección de señales de posibles tránsitos

Un elemento común en los principales trabajos de detección de exoplanetas mediante la técnica de tránsito es el realizar una identificación previa en las curvas de luz de los denominados TCEs (*Threshold Crossing Events*), es decir, eventos de la curva de luz que suponen una disminución de la luminosidad y por lo tanto, señalan la presencia de un posible tránsito planetario.

En el caso de la misión Kepler la obtención de los TCEs está incorporada en el propio tratamiento de los datos recibidos, mediante la *pipeline* Autovetter, y todos los TCEs detectados son catalogados y puestos a la disposición de la comunidad científica en el *Autovetter Planet Candidate Catalog* disponible en el MAST [19]. A los objetos con TCEs se les denominan KOI (*Kepler Object of Interest*).

En el caso de la misión K2, al tener menos datos, la misión no incorpora en el tratamiento de datos una *pipeline* equivalente, pero todos los trabajos replican los procesos de identificación para obtener igualmente TCEs de las curvas de luz.
Para esta tarea de identificación se aplica el algoritmo BLS (*Box-fitting Least Squares*) (Kovács et al., 2002) [16] (ver detalles en el anexo A.1), si bien posteriormente se han desarrollado otros algoritmos como TLS (*Transit Least Squares*) (Hippke and Heller, 2019) [11] que presentan mejores resultados.

Una vez identificados los TCEs, y como paso previo a la identificación de señales, estos TCEs se etiquetan como *planet candidate*, *astrophysical false positive* (AFP), y *nontransiting phenomenon* (NTP).

Es interesante señalar que en el caso de la misión Kepler original, dada la larga duración de las curvas de luz disponibles es posible identificar varios TCEs, correspondientes a un mismo tránsito planetario, y cada uno de ellos aparece en el catálogo. Así, los conjuntos de datos disponibles para análisis contienen un número muy superior de TCEs que de exoplanetas.
En el caso de los datos de K2 ello no siempre es posible, al sólo disponer de periodos temporales de 80 días. Ello dificulta sobremanera el disponer de curvas de luz donde observar la periodicidad de un tránsito.

Como veremos en el siguiente apartado, los métodos de identificación de exoplanetas con datos de la misión Kepler sí pueden aprovechar, y utilizan activamente, la presencia de los mismos tránsitos planetarios a lo largo del tiempo.

2.5. Identificación de las señales de tránsitos

A lo largo del tiempo se han realizado distintas aproximaciones a la identificación de señales de tránsito. Alguna de ellas utilizan por ejemplo métodos clásicos de *machine learning* como árboles de decisión, pero no utilizando las curvas de luz (o la parte correspondiente de las mismas de los TCEs), sino en características extraídas de los propios TCEs (periodicidad del tránsito, profundidad, etc.), combinándolos con parámetros estelares extraídos de otros catálogos. Otros métodos también aplican máquinas de soporte vectorial (SVM) (Pearson et al., 2017) [29], utilizan técnicas de reducción de dimensionalidad, como PCA (*Principal Component Analysis*) o métodos supervisados como k-NN (Thompson et al., 2018) [33], Random Forests (Armstrong et al, 2018) [1] y un largo etcétera.

Los mencionados métodos se han visto superados en los últimos años por la aplicación de métodos basados en *deep learning*, siendo uno de los primeros referentes de aplicación, y posteriormente más replicados y ampliados, el trabajo Shallue and Vanderburg (2018) [32] que utilizó datos de la misión Kepler original.
Partiendo de los TCEs, y aplicando diversos de los métodos de preprocesamiento de datos mencionados anteriormente (normalización, gestión de valores nulos, aplanamiento de la curva mediante *splines*, etc.), definen una red convolucional 1D doble: por un lado utilizando una visión global de un TCE y por otro una visión local del mismo (ver figura 2.1).
En ambos casos se trata del segmento de la curva de luz que contiene el TCE, centrado en el mismo, uno con una gran amplitud a ambos extremos, y la otra, mucho más focalizada en el TCE, a modo de ampliación o vista de detalle. Ambos fragmentos se agrupan en menos puntos

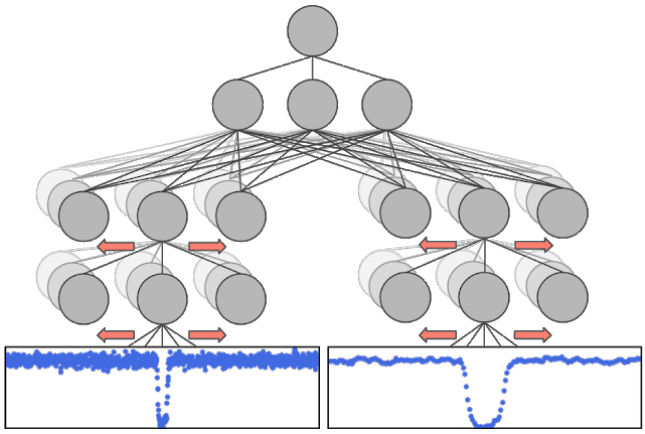


Figura 2.1: Esquema Astronet - Vista global y local [32]

(en *bins*), para así ajustarlos a las dimensiones de entrada de las redes CNN-1D, con 2001 y 201 entradas, respectivamente.
Ambas redes convergen finalmente en una red FCN *(Fully Connected Network)*, que combina ambas entradas y determina si el TCE se corresponde o no con un exoplaneta.
A este modelo lo denominan AstroNet.

En 2019, Dattilo et al. [6] retoman el trabajo de Shallue and Vanderburg (2018) [32] y aplican los mismos métodos pero utilizando en este caso curvas de luz de la misión K2. Ello les obliga a preprocesar las curvas de luz para la obtención de TCEs, al no existir un catálogo como en el caso de la misión Kepler original. Del mismo modo, para cada TCEs identificado utilizan por un lado la visión global del evento de tránsito sobre la totalidad del periodo orbital (agrupando los datos en 701 agrupamientos, tomando la mediana de los puntos de cada grupo), y la vista local (con un agrupamiento en 51 puntos).
Como elemento novedoso, las dos redes CNN 1D (visión global y local) convergen en una FCN a la que, adicionalmente, introducen atributos escalares de la estrella de la cual se analiza sus TCEs. A su modelo lo denominan AstroNet-K2.

Un modelo muy parecido a AstroNet-K2 es el propuesto por Osborn et al. (2020) [28], pero aplicado sobre datos TESS. En este caso, y debido a los pocos datos de entrenamiento que disponen, aplican técnicas interesantes de *data augmentation* como introducir ruido gausiano a las curvas de luz, desplazar ligeramente la fase de las curvas o simplemente invertir los datos (a modo de espejo) y así conseguir un conjunto de datos de entrenamiento mayor.

Un trabajo anterior también destacable es el de Pearson (2017) [29]. En este caso generan un

dataset totalmente sintético; en vez de utilizar un único TCE doblan la curva de luz, basándose en el periodo del TCE, y utilizan el valor medio de los distintos TCE, obteniendo así un TCE mucho más consistente. Este método de doblado de las curvas de luz por el periodo del tránsito, y calculando posteriormente la media de todos los bloques, de modo que la curva resultante contiene un único tránsito que representa todos los tránsitos de la curva original, también es utilizado por Jara-Maldonado et al. (2020) [15].

Chintarungruangchai and Jiang (2019) [5] proponen una mejora en la aproximación de Pearson, pues si el periodo del doblado de la curva de luz no es exacto, entonces al aplicar el valor medio al doblado de la curva tampoco será exacto, y el TCE puede acabar totalmente desdibujado. Aplican entonces lo que denominan un modelo 2D-CNN-folding-2, que no es más que sobreponer los diferentes TCEs, obteniendo así una imagen 2D donde el tránsito repetido se aprecia como una banda en una imagen 2D.

2.6. Tratamientos alternativos

Hasta ahora hemos repasado cómo se suelen identificar exoplanetas, y como hemos visto, siempre se utiliza un proceso en dos partes: identificando primero TCEs (tránsitos) de las curvas de luz, para posteriormente analizar si dichos TCEs se corresponden o no con exoplanetas, utilizándose principalmente redes convolucionales CNN 1D que ofrecen buenos resultados. No obstante, existen muchas otras técnicas de procesado y de *machine learning* sobre las curvas de luz para otras funcionalidades, ya no ligadas a la presencia de tránsitos. En estos casos una curva de luz puede verse, y de hecho lo es, como una serie temporal, y sobre la misma pueden aplicarse otro tipo de arquitecturas que no requieran identificar previamente a los TCEs. Este sería el caso, por ejemplo, de clasificar el tipo de una estrella a partir de su curva de luz.

Así, en esta sección analizaremos otros métodos más genéricos que pueden aplicarse sobre curvas de luz que podrían llegar a ser de utilidad en la determinación de tránsitos de exoplanetas, principalmente considerando a las curvas de luz como series temporales, y tratando a las mismas como un todo. Tal como indica Fawaz (2019) [13] el abanico es amplio.

Como puede observarse en la figura 2.2, los modelos vistos hasta ahora son principalmente modelos discriminativos, *end-to-end*, aplicando redes convolucionales CNN o modelos híbridos.

Como parte de los modelos generativos, uno de los primeros trabajos registrados para el tratamiento de curvas de luz es del de Naul et al. (2017) [26]. Presentan un *autoencoder* RNN (red

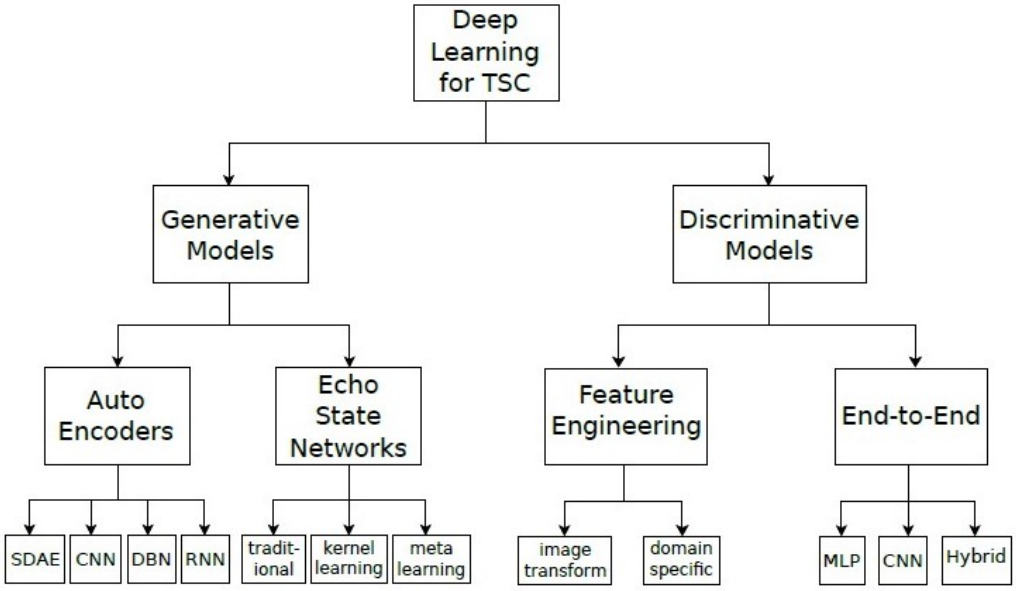


Figura 2.2: Enfoques deep learning para la clasificación de series temporales [14]

neuronal recurrente), que se compone de un codificador (*encoder*) que toma una serie temporal (una curva de luz) como entrada y produce un vector de características de longitud fija como salida, y un decodificador (*decoder*), que traduce la representación del vector de características de nuevo en una serie temporal de salida. En el caso de un *autoencoder*, el modelo se entrena utilizando como salidas las mismas entradas. Así, el vector de características intermedio obtenido puede verse como una representación de la serie temporal de entrada, y utilizarse como entrada de un clasificador, siendo esta una de las principales ventajas de este tipo de aproximaciones.

En el caso de Naul et al. (2017) [26] alimentan a un clasificador basado en *Random Forests* para clasificar estrellas variables. Su modelo de *autoencoder* utiliza dos capas GRU de tamaño 64 para codificación y dos para decodificación, con un tamaño de vector de características de 8.

Más adelante, Hinners et al. (2018) [8], partiendo de la misma premisa, utilizan una red RNN pero con dos capas ocultas LSTM con 16 nodos cada una, también para clasificación de estrellas a partir de sus curvas de luz, pero sin buenos resultados, lo cual consideran es debido al tratar con clases muy poco balanceadas. Los mismos autores, no obstante, utilizando el mecanismo de extracción de características planteado por Naul et al. [26] sí consiguen buenos resultados, utilizando la librería python `FATS` (*Feature Analysis for Time Series*) que incluye un conjunto de rutinas para el análisis de series temporales) [27].

Por último, citar el trabajo de Brunel et al. (2019) [3], donde se exploran nuevos métodos de clasificación, en este caso no de exoplanetas, sino de supernovas, a partir de curvas de luz pero con variantes de redes CNN distintas a las descritas previamente.
Una de las alternativas que proponen está basada en redes *Inception*, con la idea de realizar un proceso de clasificación unificado, a diferencia de otros métodos en los que primero se realiza una extracción de características de las curvas de luz y posteriormente otro de clasificación. La red *Inception* es similar a una red CNN con la diferencia de que las capas convolucionales y las capas de agrupación se reemplazan por módulos *Inception*, en los que se realizan distintas combinaciones de convoluciones dejando que sea el modelo el que escoja la mejor opción. Ello permite al modelo recuperar tanto características locales mediante convoluciones pequeñas como características de alto nivel mediante convoluciones más grandes. Indicar que más recientemente, Fawaz et al. (2020) [14] también profundizan en el uso de redes *Inception* para la clasificación de series temporales.

* * *

Como hemos visto, prácticamente la totalidad de las aproximaciones a la identificación de exoplanetas siguen un mismo patrón: identificación de TCEs, o candidatos a tránsitos y posterior clasificación de los mismos en exoplanetas o no exoplanetas siendo en estos TCEs donde se aplican los métodos predictivos (en algunos casos con ayuda de fuentes de datos complementarias, ajenas a las propias curvas de luz).
Recordemos que el objetivo principal del presente trabajo es el poder determinar si a partir de una curva de luz de una estrella, es decir, la luminosidad aparente de la misma a lo largo de un periodo de tiempo, y únicamente con esa información, si existe un exoplaneta orbitando a la misma. Es decir, un enfoque global de la problemática, si bien, obviamente, se utilizarán ideas y métodos descritos en este apartado, especialmente en relación al preprocesado de las curvas de luz, quizás la parte más exigente de todas.

Capítulo 3

Recuperación y procesado de los datos

En este capítulo se presentan los distintos pasos realizados para obtener un conjunto de datos adecuado (*dataset*) sobre el cual poder aplicar posteriormente los métodos predictivos.

Dicho *dataset* contendrá información de diferentes objetos EPIC (*K2 Ecliptic Plane Input Catalog*), es decir, estrellas que tienen asociado un identificador, y para cada una de ellas su curva de luz, la campaña K2 en la que fue obtenida y finalmente un indicador de si dicho objeto EPIC contiene o no exoplanetas.
Así pues precisamos dos fuentes de información. Por un lado, el catálogo de objetos EPIC que nos indique la presencia o no de exoplanetas y por otro lado, las curvas de luz, que recordemos, recogen unos 80 días de datos, con muestras cada 30 minutos aproximadamente, lo que da unas 4.000 medidas de intensidad del brillo de la correspondiente estrella.

3.1. Identificación de elementos del *dataset*

Para recuperar los datos del catálogo K2 accedemos a NASA Exoplanet Archive [24] donde pueden obtenerse:

- Datos de objetivos K2 con datos (*k2_targets*).
- Planetas confirmados (*k2_names*)
- Candidatos, que recoge tanto candidatos como falsos positivos (*k2_candidates*).

Analizando dicha información puede constatarse que se dispone de unos 400.000 objetos EPIC, en los cuales se han confirmado 449 planetas, e identificado otros 2.500 candidatos, descartándose de estos últimos unos 240 al comprobarse que no se correspondían con exoplanetas. No obstante, se dan casos en que, para un mismo objeto EPIC, se han identificado varios

exoplanetas orbitando a la estrella, es decir, sistemas planetarios, como puede ser el conocido *Trappist-1* [21]. De este modo, dado que una misma estrella (objeto EPIC), puede contener diversos exoplanetas, si nos ceñimos a cuántos objetos EPIC disponen de algún exoplaneta, la volumetría anterior se reduce, con lo que, finalmente, de los 400.000 objetos EPIC tenemos 324 conteniendo exoplanetas, 783 conteniendo candidatos y 239 conteniendo falsos positivos.

Para el presente estudio consideraremos, al igual que la mayoría de los trabajos referenciados, todos los candidatos como pertenecientes a la clase positiva (como indican Dattilo et al. (2019) [6], *"un candidato se considera como exoplaneta mientras no se demuestre lo contrario"*).

Obviamente, con unos 1.100 elementos como clase positiva de 400.000 elementos, el conjunto de datos está totalmente desbalanceado, pero como nuestro objetivo es obtener un clasificador binario que nos indique si, a partir de una curva de luz, hay presencia de exoplanetas o no, optamos por construir un *dataset* completamente balanceado, y por ello, a los 1.100 elementos de clase positiva añadimos la misma cantidad de elementos de clase negativa.
La distribución de estos elementos por campañas K2 es la mostrada en la tabla 3.1.

Campaña	Confirmados	Candidatos	Total
0	2	43	45
1	37	67	104
2	9	81	90
3	38	63	101
4	27	54	81
5	43	123	166
6	29	91	120
7	13	85	98
8	32	80	112
10	46	31	77
11	1		1
12	12	7	19
13	9	5	14
14	11		11
15	10		10
16	3	53	56
17	2		2
Total	324	783	1107

Tabla 3.1: Objetos EPIC con exoplanetas confirmados o candidatos

Podríamos obtener simplemente una muestra aleatoria de otros elementos sin exoplanetas

ni candidatos, pero ello podría acabar distorsionando los análisis posteriores, ya que existe una gran variabilidad en las distintas campañas de la misión K2 (desde la 0 a la 17) en relación a la calidad y cantidad de datos disponibles.

Como ya se ha comentado en capítulos anteriores, la misión K2 es una extensión de la misión Kepler original cuando el equipamiento empezó a fallar (incapacidad de orientar bien el satélite), lo cual se traduce en una calidad de datos inferior, unido a un volumen muy inferior de muestras para cada estrella (80 días en vez de unos 4 años de datos continuos). Todo ello afecta directamente a la selección de los elementos de clase negativa a incorporar al *dataset*. Los factores de calidad y número de datos por objeto EPIC son muy dependientes de cada campaña, por lo que los elementos de clase negativa los escogeremos de modo proporcional, por campañas, a los de la clase positiva. Recordemos que en cada campaña el satélite apuntaba a una región fija del espacio, recogiendo datos de las estrellas presentes en el campo de observación. Así, cualquier problema en el satélite (orientación, fallos temporales en la obtención de medidas, etc), afecta a todos los objetos EPIC de la campaña. De ahí la importancia de la elección proporcional de objetos EPIC, por campañas, en el *dataset*.

Igualmente, y en previsión de utilizar no sólo modelos predictivos, donde conviene utilizar clases balanceadas, sino otros métodos, no limitamos la identificación de objetos EPIC de clase negativa a los 1.100 de clase positiva sino que incrementamos hasta obtener un total de 6.000 objetos EPIC, manteniendo las proporciones por campañas y clase positiva.

3.2. Obtención de las curvas de luz

Una vez hemos definido el catálogo de objetos EPIC que utilizaremos, el siguiente paso es la descarga de curvas de luz del MAST [19] para los mismos.

Aquí, a diferencia de la misión Kepler original, donde sólo se ofrece una versión de las curvas de luz, para K2 en el propio MAST existen publicadas diferentes versiones de las mismas, que han ido aportando diversos grupos de trabajo, y que recogen mejoras en dichas curvas de luz para solucionar los problemas con el equipamiento del satélite. Todas estas versiones pueden encontrarse en el portal MAST- HLSP [22] donde aparte de las curvas originales (que identificaremos como K2), hay otras disponibles, como K2SFF, K2SC, EVEREST, K2VARCAT, etc. Para este estudio se han utilizado y evaluado diversos conjuntos, en concreto K2 (las curvas originales de la misión), K2SFF [34], EVEREST [18], pues uno de los objetivos buscados es también de tipo comparativo, para evaluar qué procesamiento correctivo de las curvas de luz es

más eficiente para posteriormente identificar presencia de exoplanetas (como se detallará más adelante, la versión K2SFF es la que parece ofrecer mejores resultados).
Así, para estos 3 tipos de datos se han descargado las curvas de luz de los 6.000 objetos EPIC preseleccionados, en forma de ficheros FITS, de los cuales se extraen las curvas de luz.

Todos los datos de la misión Kepler-K2 se dividen en objetivos de cadencia larga (LC) con muestreos cada 29,4 minutos, y objetivos de cadencia corta (SC) con muestreos cada 58,85 segundos. Utilizaremos los de larga cadencia (LC), dado que los de cadencia corta no están presentes en todos los casos, y para la identificación de exoplanetas no son necesarios unos muestreos con tanto nivel de detalle.

Debe indicarse que la recuperación de datos se realiza de modo directo de los distintos ficheros FITS disponibles en el MAST (recuperación mediante la URL asociada a cada fichero FITS), aunque existen librerías Python, como `Astropy` o `Lightkurve`, que incluyen rutinas para la recuperación de dichas curvas de modo programático.

Los ficheros FITS de la misión K2 contienen mucha información, pero la relevante para el estudio se centra en los flujos de brillo observados, es decir, las propias curvas de luz. Estos se dividen en dos tipos diferentes de flujos observados: la fotometría de apertura simple (*Simple Aperture Photometry* o SAP) que contiene los datos de flujo con correcciones para flujo de fondo, y las corregidas, o *Presearch Data Conditioning* (PDC) que son las comúnmente utilizadas, al ya incorporar correcciones en las curvas de luz derivadas de las variaciones del instrumental del satélite.
En la figura 3.1 se muestra un ejemplo de curva de luz en formato SAP, mientras que en la figura 3.2 se muestra la misma curva de luz pero en formato PDCSAP. Este flujo ya corregido, en el caso de los datos K2 se corresponde con el campo PDCSAP_FLUX del fichero FITS, mientras que para el formato K2SFF se utiliza el campo FCOR y para EVEREST el campo FLUX. Igualmente, para cada punto de la curva de luz está disponible el instante temporal de la muestra, pero es un dato que no recogemos, pues no es necesario para los análisis posteriores, pues con conocer la frecuencia de las muestras es suficiente.

Indicar también que la extracción de datos de los ficheros FITS se realiza mediante el uso de funciones de la librería `Astropy` [2], y que la información detallada del contenido de los ficheros FITS puede consultarse en el MAST-HLSP-K2 [22].

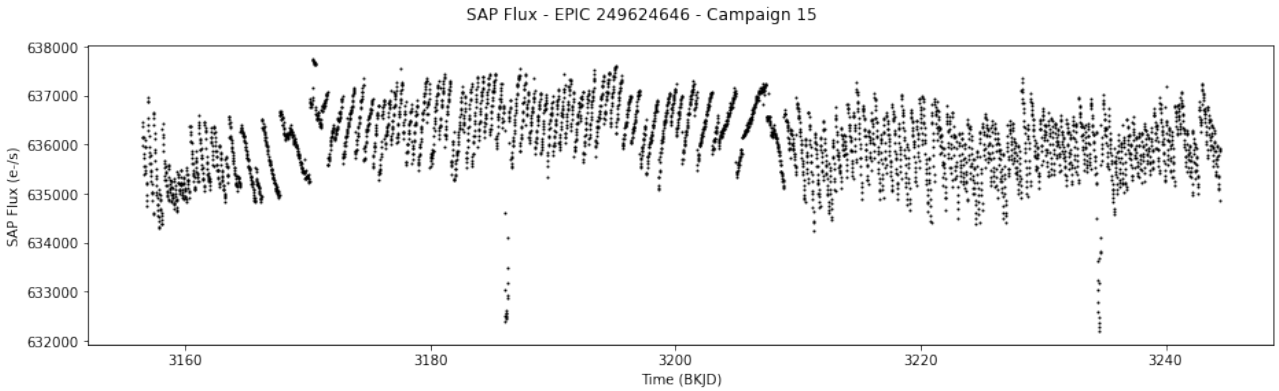


Figura 3.1: Ejemplo curva de luz en formato SAP

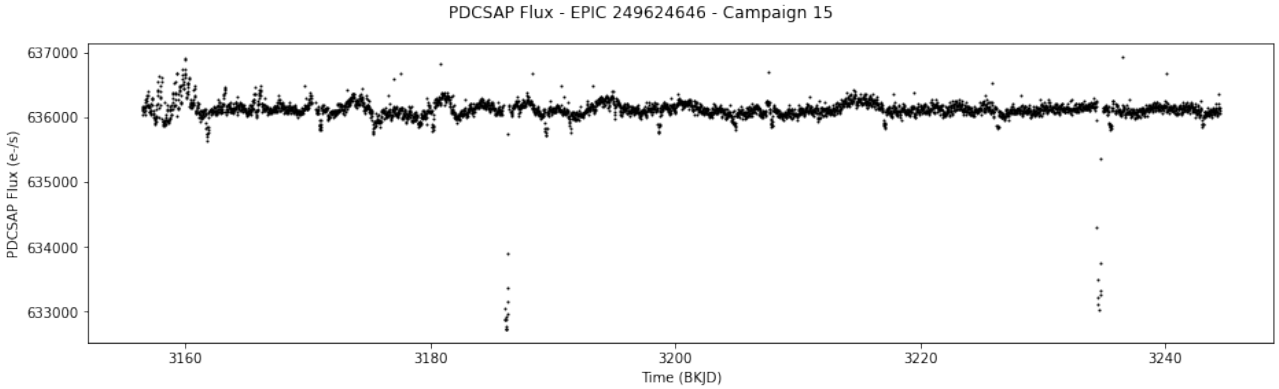


Figura 3.2: Ejemplo curva de luz en formato PDC-SAP

3.3. Preprocesado de curvas de luz

Una vez recuperadas las curvas de luz ya pasamos al preprocesado de las mismas.

Se analiza en primer lugar la presencia de valores nulos en las curvas de luz y su distribución, obteniéndose los datos, agrupados por campañas.
En la tabla 3.2 se muestra, para cada campaña, el número de elementos considerados, y promedios de las curvas de luz correspondientes: la longitud (número de medidas por curva), valores nulos, huecos en la curva (número medio intervalos sin datos), y finalmente, el promedio del máximo intervalo sin datos presente en las curvas. Como puede apreciarse, existe mucha variabilidad entre longitudes, muy dependientes de cada campaña, y multitud de huecos en los datos, siendo algunos de ellos muy significativos. Por ejemplo, en la campaña 10 aparece un gran hueco en los datos uniforme, fruto de problemas temporales en el satélite durante esa campaña.

Uno de los trabajos en el procesamiento de las curvas de luz será el uniformizar las longitudes

Campaña	Registros	Longitud	Valores Nulos	Num.Huecos	Max.Hueco
0	227	3.604	153	125	8
1	499	3.915	372	205	141
2	508	3.805	277	221	11
3	473	3.385	213	176	6
4	419	3.469	204	168	6
5	990	3.662	228	192	4
6	557	3.862	223	179	14
7	441	3.982	552	243	9
8	499	3.852	369	238	61
10	348	3.236	1.218	163	652
11	4	1.138	94	73	6
12	85	3.861	543	217	257
13	62	3.943	321	227	31
14	49	3.900	221	188	4
15	45	4.307	280	230	15
16	785	3.893	274	212	7
17	9	3.282	212	171	6
Total	6.000	3.731	343	198	64

Tabla 3.2: Valores en curvas de luz (promedios por campaña)

de las mismas para tener una entrada fija como entrada al entrenamiento de los modelos. Así, los trabajos realizados en las curvas de luz son los siguientes: eliminar la tendencia en las curvas de luz, tratar los valores outliers, normalizar los datos, compactar huecos, interpolar valores nulos y finalmente, uniformizar longitudes curvas de luz. Veamos cada uno de ellos a continuación.

3.3.1. Eliminar tendencia en las curvas de luz

Como se aprecia en la figura 1.1 es esperable que una curva de luz destinada a identificar exoplanetas sea una curva lo más plana posible, para así poder identificar más fácilmente los descensos que podrían indicar la presencia de los exoplanetas.

Existen diversas técnicas para eliminar la tendencia de una curva de luz, es decir, aplanar la curva, siendo la más habitual la obtención de una *spline* que se ajuste a la curva y dividiendo posteriormente la curva por dicho *spline*. Esta técnica es efectiva pero tiene el inconveniente que cualquier tránsito planetario presente en la curva puede distorsionar el *spline* obtenido. Así, cuando el análisis utiliza TCEs, se calcula el *spline* de la curva sin tener en cuenta el intervalo correspondiente a los TCEs.
En nuestro caso ello no es posible pues únicamente se utilizarán las curvas de luz, sin infor-

mación adicional. Por dicho motivo se ha optado por la utilización de la librería `Wotan` [9], creada por Hippke et al. (2019) [11], y utilizando el método que los autores consideran el más adecuado: un control deslizante de ventana de tiempo con un estimador de ubicación robusto iterativo basado en *Tukey's biweight*.
Se trata de un método robusto y como tal no está afectado por la presencia de valores nulos o de outliers en los datos. Esto permite tanto el aplanamiento de la curva (eliminación o modelado del ruido instrumental y estelar) así como la eliminación de la tendencia de la misma.

La rutina `flatten` que ofrece dicha librería aplica el método indicado, y permite un ajuste fino del filtro de ventana de tiempo utilizada. Este filtro, equivaldría, en cierto modo, a eliminar en los cálculos para aplanar la curva, a los posibles tránsitos que ésta pudiera contener (similar a la eliminación mencionada de los TCEs antes de calcular una *spline* mencionada anteriormente). Un valor pequeño de la ventana eliminaría la variabilidad estelar más efectivamente, pero con el riesgo de eliminar también los posibles tránsitos planetarios.

Los autores indican que la ventana debería ser 2 o 3 veces mayor que la duración de un tránsito. Pero de tránsitos puede haber de distintas duraciones. Así, por ejemplo, un tránsito de la Tierra frente al Sol es de unas 13 horas, lo que 3x implicaría utilizar al menos una ventana de 1.62 días. Se ha optado por utilizar un valor conservador, de 2 días, con lo apenas deberían eliminarse posibles tránsitos, a costa, eso sí, de reducir menos la variabilidad estelar presente en la curva de luz.

3.3.2. Tratamiento de valores outliers

Es habitual considerar como *outliers* aquellos puntos que superen en N veces la desviación estándar, si bien aplican valores distintos para la parte superior de la curva, con un valor de 2σ, como de la parte inferior, donde en función del tipo de tránsito (en este caso la profundidad del mismo), pueden aplicar unos valores u otros.
Siguiendo la recomendación de Hippke et al. (2019) [10] aplicamos de nuevo un valor muy conservador de 20 σ. Para esta eliminación utilizaremos la rutina `sigma_clip` de la librería `Astropy` [2].

3.3.3. Normalización de los datos

Aplicamos normalización max-min para tener los valores resultantes en el rango [0,1]

3.3.4. Compactación de la curva

Como hemos visto, las curvas presentan numerosos huecos en los datos, algunos de ellos de gran tamaño. Frente a estos huecos, que no son más que valores nulos, tenemos dos opciones: interpolar los datos o bien directamente eliminarlos compactando la curva. Si interpolamos pero el hueco es grande, podríamos distorsionar completamente la curva de luz, pero podemos compactar la curva sin problema pues en este estudio buscamos tránsitos no la caracterización de los mismos (frecuencia de paso, profundidad, ...)

Se ha optado por fijar un umbral de 10 muestras consecutivas sin datos como límite de compactación. Así, todo hueco superior a 10 muestras es eliminado, es decir, se compacta la curva.

3.3.5. Interpolación de valores nulos

Todos los valores nulos aún presentes en la curva, que como máximo será de 10 muestras consecutivas, son substituidos por interpolación lineal, aplicando ruido gaussiano a los valores.

3.3.6. Uniformizar longitudes curvas de luz

Una vez tenemos la curva procesada ya podemos igualar las longitudes de las curvas de luz.

Fijamos en primer lugar una longitud deseada, de 3.500 puntos, ya que se trata de un punto aproximado que recoge la media de valores presentes en las curvas de luz. Para aquellas curvas que superen estos 3.500 puntos eliminaremos de modo equidistante y repartido por toda la curva valores hasta conseguir la longitud adecuada. En el resto de casos, una primera opción considerada fue aplicar el mecanismo inverso al anterior, es decir, insertar puntos equidistantes a lo largo de la curva de luz, tomando como valor la interpolación de los valores adyacentes. El problema de esta aproximación es que en algunos casos la diferencia de puntos con respecto a la longitud deseada es tan grande que la curva quedaría totalmente distorsionada.
Finalmente se ha optado por aplicar una replicación de la curva por simetría por su parte final: si la curva va de A$\rightarrow$ B, crear curva A$\rightarrow$ B$\rightarrow$ A (y repetir el proceso mientras la longitud sea inferior a la deseada), para finalmente truncar el resultado a la longitud final.
Con este mecanismo no se producen discontinuidades ni saltos en las curvas, si bien podrían llegar a duplicarse tránsitos, pero de nuevo, esto no es problema, pues no buscamos ni número de tránsitos ni su caracterización detallada, sino tan solo si éstos están presentes o no.

3.4. Visualización de curvas de luz

En esta sección mostramos algunos ejemplos de curvas de luz de la misión K2, así como el impacto del preprocesado aplicado a las mismas, previo a la aplicación de métodos predictivos.

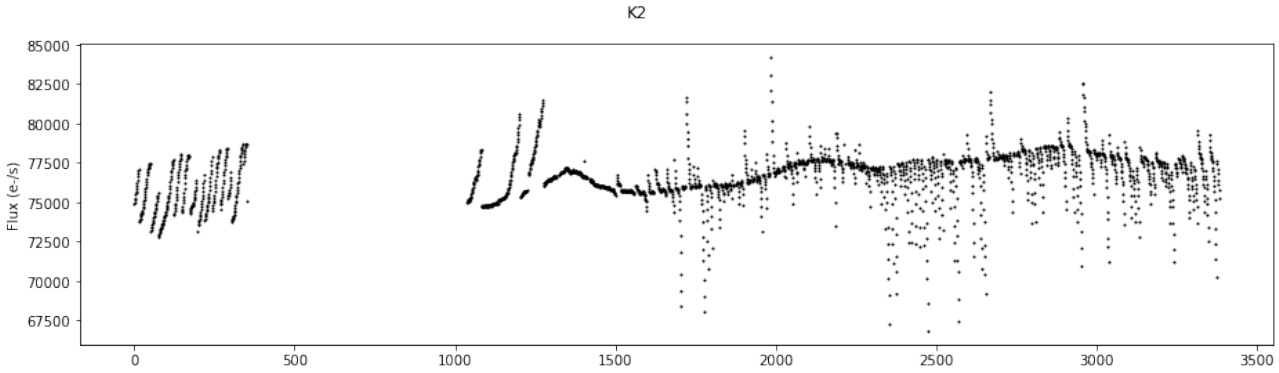
En la figura 3.3 puede observarse visualmente el tratamiento de datos, mostrándose la curva original K2, la curva procesada con K2SFF y finalmente la curva postprocesada con los mecanismos descritos. En la curva original K2 se aprecia un gran hueco, que en la curva K2SFF desaparece, quedando la curva en unos 2.500 puntos. En la curva final, tras aplicar el procesado, y aplicando simetría, tenemos ya una curva preparada, con longitud fija de 3.500 puntos, para poder ser utilizada en los procesos analíticos.

Para contextualizar la dificultad en el análisis de las curvas de luz de la misión K2 se muestran otros ejemplos, pues ciertamente es una tarea compleja ya que no siempre las curvas de luz permiten siquiera una identificación visual de los tránsitos.
En la figura 3.4 se muestra la curva de luz del EPIC 228735255, conocido como K2-140b, que contiene tránsitos de un exoplaneta tipo "Hot Jupiter". "Hot Jupiter" es un tipo de exoplaneta de masa similar a Júpiter pero que orbita muy próximo a su estrella, en este caso con un periodo de tan solo 6,75 días, lo que permite visualizar la periodicidad de los tránsitos, claramente marcados en la curva.
Pero también otros casos más complejos como el de la figura 3.5, también una curva de luz con exoplaneta, siendo este caso el más habitual presente en los datos disponibles.

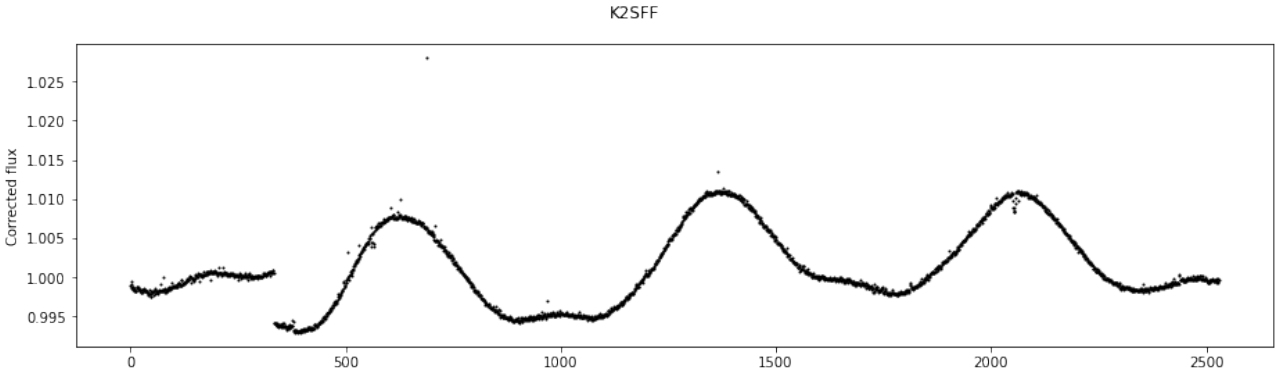
3.5. Preparación del *dataset* final

Una vez recuperadas y procesadas las curvas de luz para los objetos EPIC de interés y de las diferentes fuentes de datos (K2, K2SFF y EVEREST), preparamos el *dataset* final:

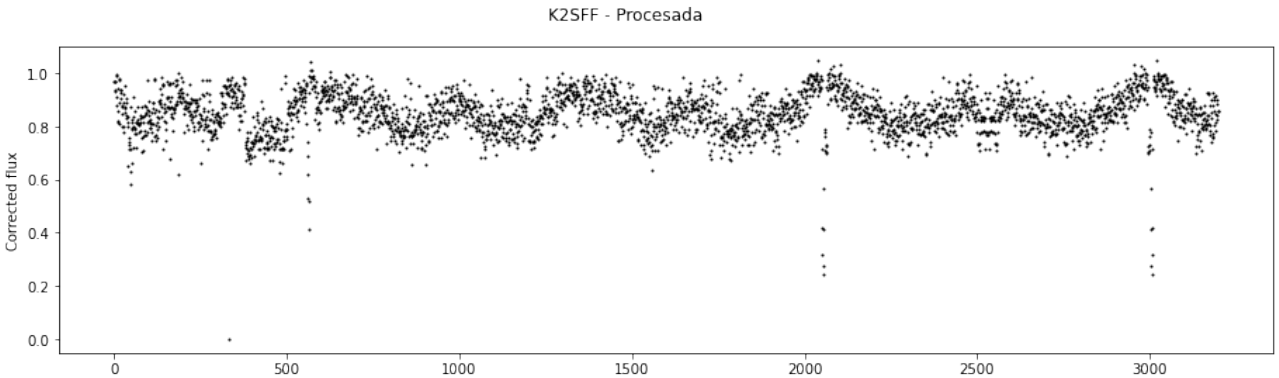
- Buscamos en primer lugar el factor común, es decir, aquellos objetos EPIC que aparecen en los tres subconjuntos de datos, con lo que se eliminan algunas curvas.
- Del conjunto resultante, separamos un 10 % de los datos para test, y el resto, para entrenamiento. Posteriormente, estos últimos se utilizarán para entrenamiento y validación.
- Obtenemos para cada fuente de información (K2, K2SFF y EVEREST) dos ficheros, el primero para entrenamiento, con 986 curvas de luz con exoplanetas y el mismo número pero sin exoplanetas, y 109 curvas de luz para test con exoplanetas, y el mismo número, 109, sin exoplanetas.



(a) Curva K2

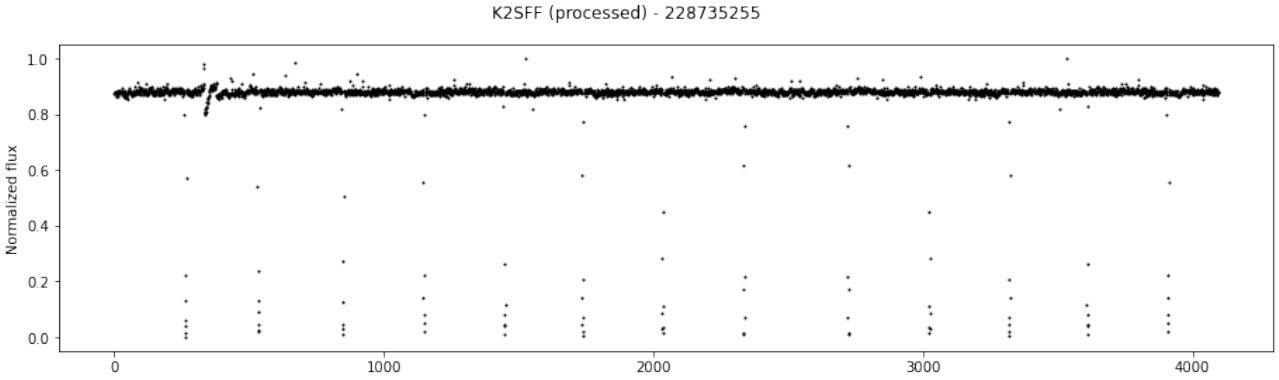


(b) Curva K2SFF

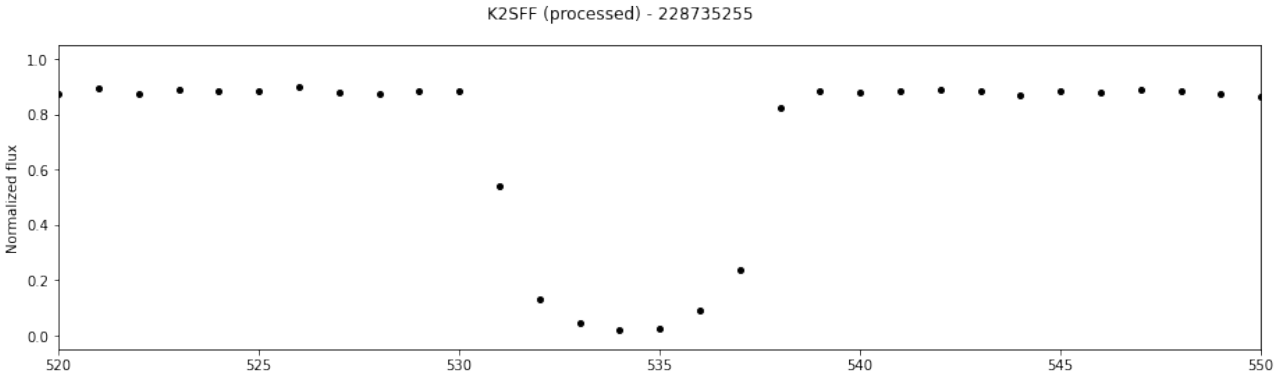


(c) Curva K2SFF post-procesada

Figura 3.3: Procesado de curvas de luz - EPIC 201128338



(a) Curva de luz completa



(b) Detalle de un tránsito

Figura 3.4: Curva de luz post-procesada - EPIC 228735255 - K2-140b

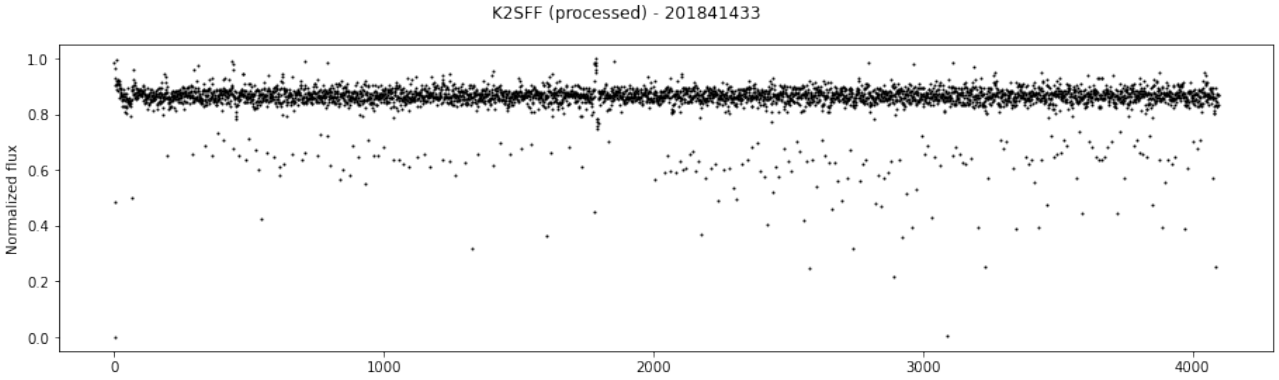


Figura 3.5: Curva de luz post-procesada- EPIC 201841433

Capítulo 4

Implementación

4.1. Introducción

En este capítulo se presentan los distintos métodos analizados para la clasificación de curvas de luz de la misión K2 para indicar la presencia, o no, de exoplanetas en las mismas.

Se han analizado únicamente métodos basados en redes neuronales profundas (*deep learning*), descartándose así de entrada otras posibles alternativas basadas en modelos más tradicionales de minería de datos: árboles de decisión, SVM (*Support Vector Machines*), vecinos más próximos (KNN), y un largo etcétera. El motivo no es otro que la naturaleza de los datos a procesar, que son de tipo temporal, es decir, series temporales. El utilizar los métodos anteriores requeriría un procesado previo de los datos, para poder hallar las características de las curvas de luz, por ejemplo, mediante análisis de componentes principales (PCA), o incluso el uso de redes convolucionales para obtener un vector de características que posteriormente fuera la entrada a otros modelos, como podrían ser los árboles de decisión.

Lo anterior podría ser de utilidad en el caso de querer caracterizar la totalidad de la curva de luz, como por ejemplo determinar el tipo de estrella que se trata. Pero en la búsqueda de posibles tránsitos planetarios buscamos características o en cierto modo, anomalías, presentes en la curva de luz. De ahí los trabajos descritos en el capítulo anterior, para justamente eliminar cualquier característica global de la curva de luz y poderse centrarse en los detalles, los posibles tránsitos.

4.2. Metodología

No existen reglas específicas para la selección de los mejores modelos que podrían ser adecuados para nuestra problemática, por lo que se ha optado por elegir aquellas opciones que previamente, o en otros entornos similares, se han mostrado útiles en el procesado de series temporales, siendo las redes convolucionales y las redes recurrentes las opciones, sobre el papel más adecuadas.
La propia selección de las arquitecturas de red ha consistido en la aplicación de procesos de ensayo y error, si bien teniendo como referencia arquitecturas ya existentes. En el presente estudio no se exponen todas las combinaciones analizadas, sino sólo aquellas más representativas o que han ofrecido mejores resultados. Así por ejemplo, en redes convolucionales existen multitud de combinaciones que podrían utilizarse, en relación al número de capas convolucionales, a los tamaños de filtros, tamaños de mapas de características, o diferente orden de aplicación, y un largo etcétera. La selección y ajuste ha sido un proceso manual.

4.2.1. Hiperparámetros

Sí existe, por otro lado, una serie de hiperparámetros, que una vez seleccionada una arquitectura de red pueden ajustarse en busca de mejores resultados. Estos hiperparámetros son:

- **Épocas** (*epochs*):
  Número de veces que se pasa todo el conjunto de datos a través de la red
- **Tasa de aprendizaje** (*learning rate*)
  Controla cuánto cambiar el modelo en respuesta al error estimado cada vez que los pesos del modelo son actualizado. Un valor muy pequeño conduce a un proceso de aprendizaje largo y tedioso y un alto valor puede conducir a un aprendizaje más rápido pero también más inestable.
- **Tamaño de lote** (*batch size*)
  Número de muestras que se pasarán por la red simultáneamente, y son utilizados para calcular el error y actualizar los coeficientes del modelo.

Existen otros hiperparámetros que han sido fijados de modo general para todos los modelos:

- **Función de pérdida** (*loss*):
  Esta función es una forma matemática de medir qué tan incorrectas son las predicciones. Además, la función de pérdida se utiliza en el proceso de optimización para encontrar

los mejores pesos de modelo para sus datos. En nuestro estudio hemos optado, de modo general, por utilizar la función de pérdida de entropía cruzada (*categorical crossentropy*) al ser esta normalmente muy adecuada en procesos de clasificación

- **Optimizador** (*optimizer*)
  El optimizador actúa, junto con la función pérdida, actualizando los pesos del modelo. En nuestro estudio hemos optado por utilizar el optimizador Adam de modo general, al ser uno de los más utilizados y con excelentes resultados, si bien se han realizado pruebas preliminares con otros optimizadores, como NAdam, mostrando un rendimiento equivalente.

- **Función de activación** (*activation function*)
  Hemos optado de nuevo por la función de activación más comúnmente utilizada: ReLU.

4.2.2. Métricas

Para poder comparar los resultados de los distintos modelos precisamos de un conjunto de métricas.

Si bien la función de pérdida es la que permite entrenar un modelo, una vez obtenido éste, deben realizarse predicciones contra los datos de validación para comprobar su efectividad.
Dada la naturaleza de nuestro estudio, que es una clasificación binaria, a la que además forzamos que los datos de entrenamiento y validación estén totalmente balanceados, el valor de la precisión (*accuracy*) es el más adecuado, juntamente con el *recall*, ya que lo que buscamos es identificar los elementos de clase positiva, es decir, curvas de luz con exoplanetas.
Estas métricas vienen definidas a partir de la matriz de confusión (4.1) que se puede obtener al realizar una predicción sobre los datos de validación (o de test).

La precisión o exactitud (*accuracy*) proporciona información general sobre el número de instancias incorrectamente clasificadas:

$$PRE = \frac{TP + TN}{FP + FN + TP + TN} \tag{4.1}$$

El *recall* se corresponde con la tasa de verdaderos positivos, o TPR (*true positive rate*):

$$TPR = \frac{TP}{FN + TP} \tag{4.2}$$

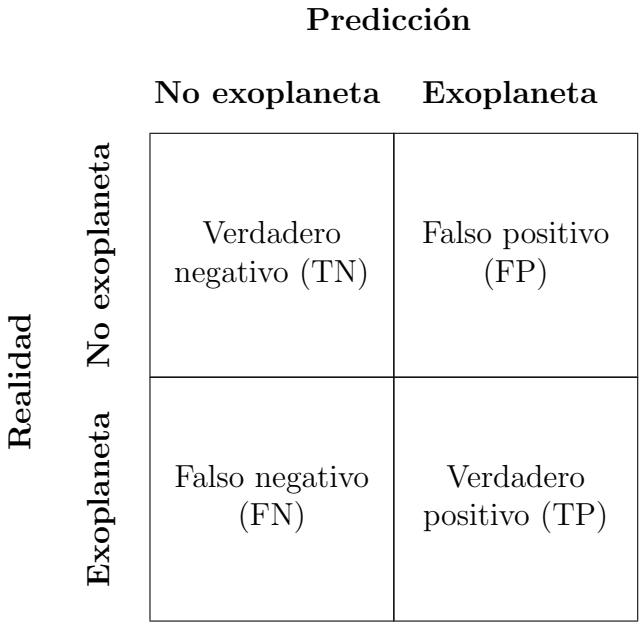


Figura 4.1: Matriz de confusión

Estas serán las dos métricas principales utilizadas para la comparación de los modelos.

Es importante señalar que en el entrenamiento de los modelos se ha optado, de modo general, por aplicar un valor de épocas elevado. Ello podría condicionar el resultado final de un modelo pues a partir de un determinado número de épocas un modelo podría volverse inestable y la precisión con los datos de validación empeorar y no mejorar. Por este motivo, se han configurado mecanismos (*callbacks*) para que registren durante cada entrenamiento el modelo que ofrezca una mejor precisión, siendo éste el modelo de referencia en cada caso.

4.2.3. Datos de entrenamiento, validación y test

Vimos en el capítulo anterior que se generaron un conjunto de datos de entrenamiento y otro de test, ambos con clases totalmente balanceadas. El conjunto de datos de test supone un 10 % del total de los datos. En concreto, el primer conjunto de datos contiene 986 curvas de luz de cada clase y el segundo, de test, 109 curvas también por cada clase. Si no se indica posteriormente lo contrario (en algún caso particular) para la aplicación de los métodos predictivos el conjunto de datos de entrenamiento se divide a su vez en dos subconjuntos: un 90 % para entrenamiento, y el 10 % restante para validación.

4.2.4. Entorno

Todos los entrenamientos de los modelos se han ejecutado en el entorno Google Colab, utilizando GPUs.
Igualmente, para la codificación se ha utilizado la librería `Keras` disponible en `Tensorflow`.
Tanto el código como los *dataset* se encuentran en una carpeta compartida en Google Drive.

4.3. Selección tipos de curvas a procesar

Los mecanismos de preprocesado de curvas de luz descritos en el apartado 3.3 han sido aplicados a las curvas K2, K2SFF y EVEREST, descargadas del MAST [22]. Finalmente, la mejor combinación resultante es la utilización de datos K2SFF, aplicando los preprocesados mencionados. Aunque cabe destacar que también se han realizado pruebas con la versión K2SC, que han ofrecido unos resultados parecidos a las curvas K2SFF. Pero se descartaron al solamente estar disponibles las curvas de luz de las campañas 3 a 8, y parte de la campaña 10, con lo que se reducía sensiblemente el número de objetos EPIC con exoplanetas disponibles.
En el anexo A.2 se ofrecen algunos detalles de los procesos aplicados para la obtención de las curvas K2SFF y EVEREST.

Para esta identificación relativa a qué formato de curvas y qué procesados son los que mejores resultados ofrecen se aplicaron los primeros modelos predictivos, que veremos en los siguientes apartados (4.4.1, 4.4.2.1 y 4.4.2.2). En el anexo A.3 se muestran los resultados al aplicar dichos modelos sobre los datos K2 y EVEREST.

En relación a la selección de curvas de luz, en la fase inicial del proyecto, se realizaron otras pruebas adicionales. La primera de ellas consistió en utilizar las curvas de luz originales K2 pero filtrando los datos de las mismas en función de la calidad de cada punto. En los ficheros FITS de las curvas de luz existe un campo denominado SAP_QUALITY que recoge una serie de indicadores (*flags*), hasta 32, que recogen, para cada punto, posibles incidencias: problemas con el equipamiento específicos, errores en la orientación, de fotometría, momentos en que el satélite se estaba reorientando con sus motores, etc. Así, una de las pruebas consistió en utilizar los datos originales K2 pero eliminando (marcando con valores nulos) todas las muestras que tuvieran algún flag activo, pero se observó que los resultados apenas mejoraban. Cabe decir que esta información de calidad es utilizada, entre otras, en K2SFF para mejorar las curvas de luz [34].

Otra de las pruebas realizadas consistió en recoger datos de curvas de luz no directamente

del MAST sino mediante la librería `Lightckurve` [17] y que permite, adicionalmente, aplicar mecanismos de mejora (como el aplanamiento de la curva, etc), si bien éstos son demasiado agresivos. Por ejemplo, a nivel de *outliers*, por defecto, eliminan todo lo que supera 6 σ por la parte inferior de las curvas de luz, lo que podía llegar a eliminar tránsitos en las curvas. Como en el caso anterior, los resultados tampoco fueron satisfactorios.

Así, todos los modelos que se describen a partir de ahora han sido aplicados sobre el conjunto de datos K2SFF al cual se le ha aplicado el preprocesado detallado en el apartado 3.3.

4.4. Modelos

4.4.1. Redes totalmente conectadas

Un primer paso en la evaluación de modelos consiste en disponer de uno o varios modelos que sean sencillos y básicos y que actúen como línea base o punto de partida para evaluar así qué modelos cumplen mejor con los objetivos buscados.

Como modelo base se plantea una red MLP (*Multi-Layer Perceptron*) sencilla, tal como se muestra en la figura 4.2.

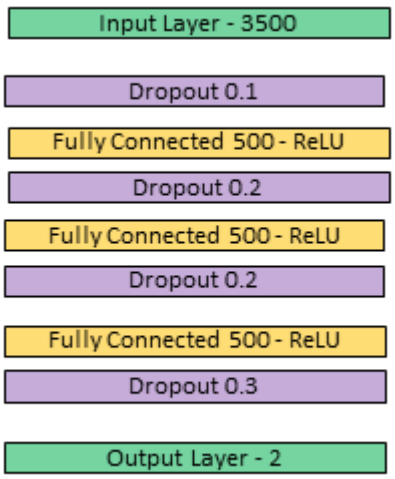


Figura 4.2: Red MLP básica

Realizando 2 ejecuciones, para obtener así el promedio de los resultados, se obtienen los resultados mostrados en la tabla 4.1, que muestran diversas combinaciones de hiperparámetros, con 500 *epochs* (nota: los tiempos de entrenamiento se expresan en segundos).
Como era en cierto modo esperable dada la naturaleza de los datos de entrada, ninguno de los

MLP - Multi Layer Perceptron

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,01	16	29	49,83 %	55,56 %	0,00 %	55,56 %
	32	18	49,50 %	44,44 %	100,00 %	55,56 %
	64	13	50,62 %	44,44 %	100,00 %	55,56 %
0,001	16	36	50,62 %	44,44 %	100,00 %	55,56 %
	32	18	50,62 %	44,44 %	100,00 %	55,56 %
	64	12	50,56 %	44,44 %	100,00 %	55,56 %
0,0001	16	34	50,62 %	44,44 %	100,00 %	56,07 %
	32	24	50,62 %	44,44 %	100,00 %	57,07 %
	64	11	50,45 %	44,44 %	100,00 %	58,08 %

Tabla 4.1: Resultados MLP

modelos ha podido converger, siendo el resultado mayoritario, tal como se aprecia en el valor de *recall* obtenido en validación, una predicción conforme todas las curvas contienen exoplanetas.

Indicar que, en el anexo A.3 se muestran los resultados al aplicar este modelo sobre los datos K2 y EVEREST.

4.4.2. Redes convolucionales

Para intentar clasificar las curvas de luz conforme presentan o no exoplanetas, podemos utilizar las redes convolucionales o CNN (*Convolutional Neural Network*), pues pese a que no son imágenes sino series temporales pueden aplicarse todos los mecanismos propios de las redes convolucionales para imágenes 2D, pero para series temporales, donde únicamente tenemos una dimensión (1D).

En este apartado analizaremos redes convolucionales puras, es decir, sin añadir después de las capas convolucionales de la red nuevas capas totalmente conectadas para poder realizar la clasificación. Indicar que los modelos combinados los veremos posteriomente, en el apartado 4.4.4.

4.4.2.1. Red FCN - *Fully Convolutional Network*

En la figura 4.3 se muestra una red totalmente convolucional estándar, con 3 capas, donde puede observarse que, de modo tradicional, a medida que avanzan las capas aumenta el tamaño de las dimensiones de la salida (*feature maps*) (64, 128, 256), y disminuye en tamaño de los filtros (8, 5, 3), para finalizar en una capa de agrupación.

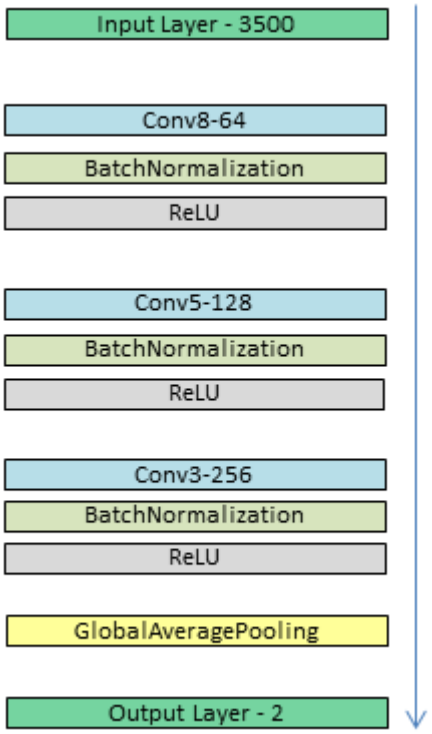


Figura 4.3: Red FCN básica

Durante la fase de pruebas se realizaron diversas combinaciones en cuanto al número de capas convolucionales, así como tamaño de dimensiones de salida de las capas y filtros aplicados, siendo la red mostrada una de las que presenta mejores resultados.

FCN - Full Convolutional Network

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,01	16	385	89,57 %	70,20 %	40,91 %	73,74 %
	32	400	89,94 %	65,41 %	73,87 %	75,26 %
	64	377	87,80 %	62,88 %	55,68 %	72,98 %
0,001	16	413	81,97 %	55,06 %	34,66 %	77,53 %
	32	255	76,13 %	58,08 %	86,36 %	77,02 %
	64	305	76,50 %	62,38 %	67,05 %	76,52 %
0,0001	16	597	75,45 %	74,75 %	50,57 %	78,29 %
	32	521	74,64 %	72,73 %	53,41 %	78,79 %
	64	631	74,41 %	75,76 %	68,18 %	78,28 %

Tabla 4.2: Resultados red FCN - Fully Convolutional Network

En la tabla 4.2 se muestran los resultados obtenidos con la red FCN analizada (red que tiene 142.530 parámetros) con promedios para 2 ejecuciones por cada combinación de hiperparámetros mostrada, y un total de 500 *epochs*; igualmente, en el anexo A.3 se muestran los

resultados al aplicar este modelo FCN sobre los datos K2 y EVEREST.

4.4.2.2. Red ResNet - *Residual Network*

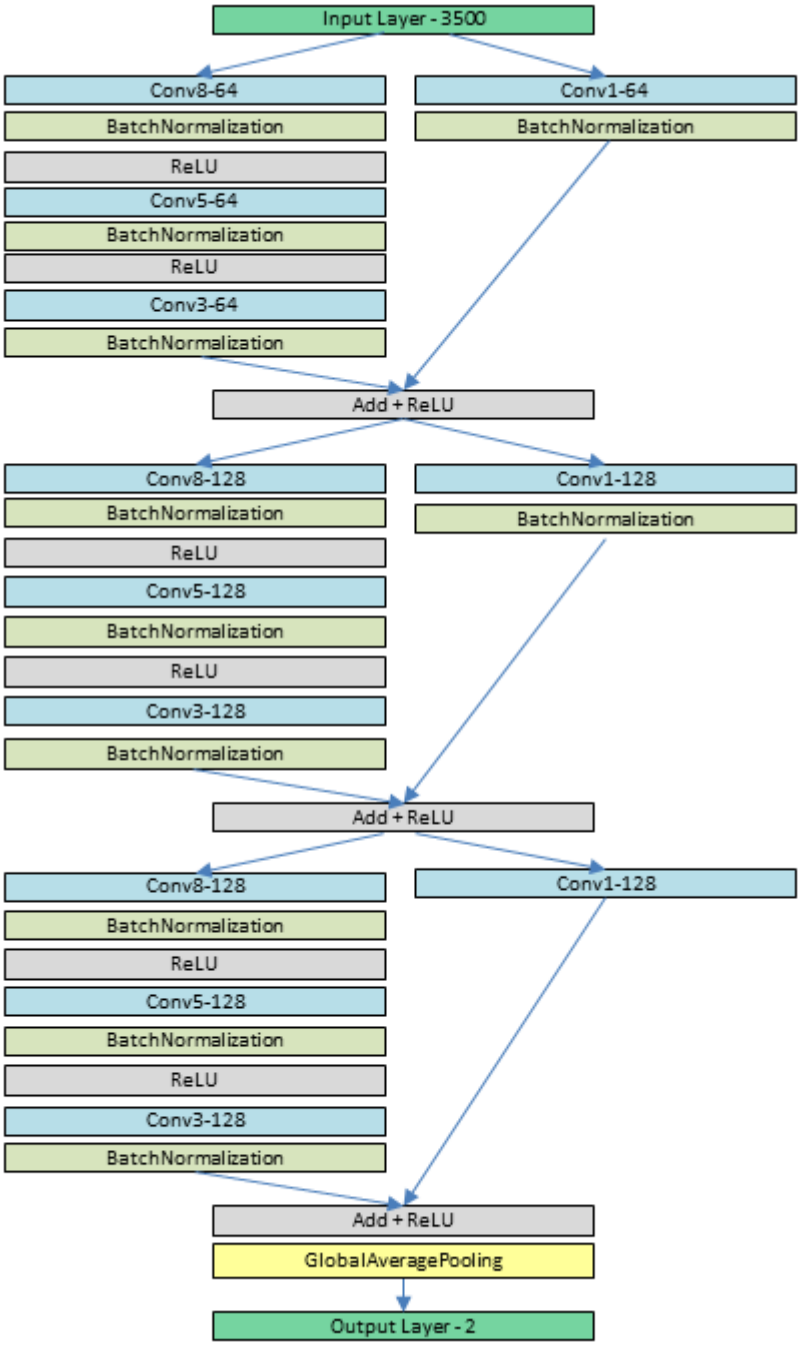


Figura 4.4: Red estructura ResNet

Las redes ResNet (*Residual Network*) nacieron de la comprobación que una red convolucional, y en contra de lo que pudiera parecer, no siempre mejora por tener mayor profundidad. De

hecho, a mayor profundidad pueden aparecer problemas como el desvanecimiento del gradiente (*vanishing gradient*) o de la "maldición de la dimensionalidad" (*curse of dimensionality*), dejando la red de aprender, ya que su aprendizaje se estanca y empieza a degradarse. Así, redes convolucionales menos profundas pueden tener, en ocasiones, un mejor comportamiento que redes más profundas.
Para solucionar estos problemas, en las redes ResNet aparecen los denominados bloques residuales. Estos se saltan unas cuantas capas, tal como se aprecia en la figura 4.4, donde se introducen tres capas adicionales de identidad, es decir, replican la entrada en la salida (capas del lado derecho de la imagen). Se ha utilizado una versión de ResNet adaptada a datos de entrada lineales (series temporales), según lo propuesto por Wang et al. (2017) [35].

En la tabla 4.3 se muestran los resultados de la red con estructura ResNet (red con 506.818 parámetros) con promedios para 2 ejecuciones por cada combinación de hiperparámetros mostrada y un total de 500 *epochs*.

Red con estructura ResNet - Residual Network

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,01	16	878	96,50 %	65,91 %	70,46 %	76,77 %
	32	899	98,26 %	69,95 %	60,80 %	75,76 %
	64	705	97,13 %	66,17 %	32,96 %	71,72 %
0,001	16	816	98,96 %	67,68 %	55,12 %	77,28 %
	32	840	99,72 %	70,96 %	55,69 %	75,51 %
	64	764	92,76 %	56,06 %	81,25 %	76,77 %
0,0001	16	774	99,27 %	56,06 %	92,05 %	80,30 %
	32	872	100,00 %	74,25 %	61,36 %	79,30 %
	64	745	99,83 %	66,92 %	28,41 %	78,79 %

Tabla 4.3: Resultados red con estructura ResNet

Podemos observar que los mejores resultados se obtienen con una tasa de aprendizaje (*learning rate*) de 0.0001, y que el tamaño de lote para el entrenamiento (*batch size*) apenas influye en los resultados, obteniéndose una mejor precisión (*accuracy*) cercana al 80 %, considerando el mejor resultado obtenido durante cada entrenamiento, valor que no tiene porqué coincidir con el correspondiente a la última iteración (*epoch*).

Indicar que, en el anexo A.3 se muestran los resultados al aplicar este modelo con arquitectura ResNet sobre los datos K2 y EVEREST.

4.4.2.3. Red Inception

Este tipo de redes pretenden resolver un tema recurrente en el diseño de redes convolucionales: qué tipos de convoluciones utilizar (tamaño de los *feature maps*), qué tamaño de filtros, etc. Así la idea es usar todos los que se consideren adecuados y dejar que sea el propio modelo el que decida. De este modo se realizan todas las convoluciones de modo paralelo, y se concatenan los resultados en mapas de características antes de ir a la siguiente capa.
Una de las ventajas adicionales de este tipo de redes es que permite recuperar tanto características locales mediante convoluciones pequeñas como características de alto nivel mediante convoluciones más grandes.

En la figura 4.5 se muestra una red de este tipo donde, como puede observarse, se prueban de modo paralelo varias convoluciones con distinta parametrización. Por otro lado en la tabla 4.4 se muestran los resultados la red *Inception*, con 220.802 parámetros, con promedios para 2 ejecuciones por cada combinación de hiperparámetros mostrada, y un total de 500 *epochs*.

Red Inception

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,0100	32	861	86,19 %	61,62 %	57,95 %	69,70 %
	64	867	80,38 %	64,14 %	56,82 %	70,71 %
0,0010	32	871	79,43 %	59,09 %	62,50 %	70,20 %
	64	1212	79,31 %	62,63 %	61,36 %	70,20 %
0,0001	32	337	63,64 %	55,05 %	67,05 %	65,15 %
	64	351	63,19 %	57,07 %	60,23 %	66,67 %

Tabla 4.4: Resultados red Inception

Podemos observar que los mejores resultados se obtienen con una tasa de aprendizaje (*learning rate*) de 0.001, y que el tamaño de lote para el entrenamiento (*batch size*) apenas influye en los resultados, obteniéndose una mejor precisión (*accuracy*) cercana al 70 %, considerando el mejor resultado obtenido durante cada entrenamiento.

4.4.3. Redes recurrentes

Las redes recurrentes, por su propia naturaleza, deberían ser adecuadas para el análisis de curvas de luz, pues éstas no son más que series temporales que recogen la luminosidad de una

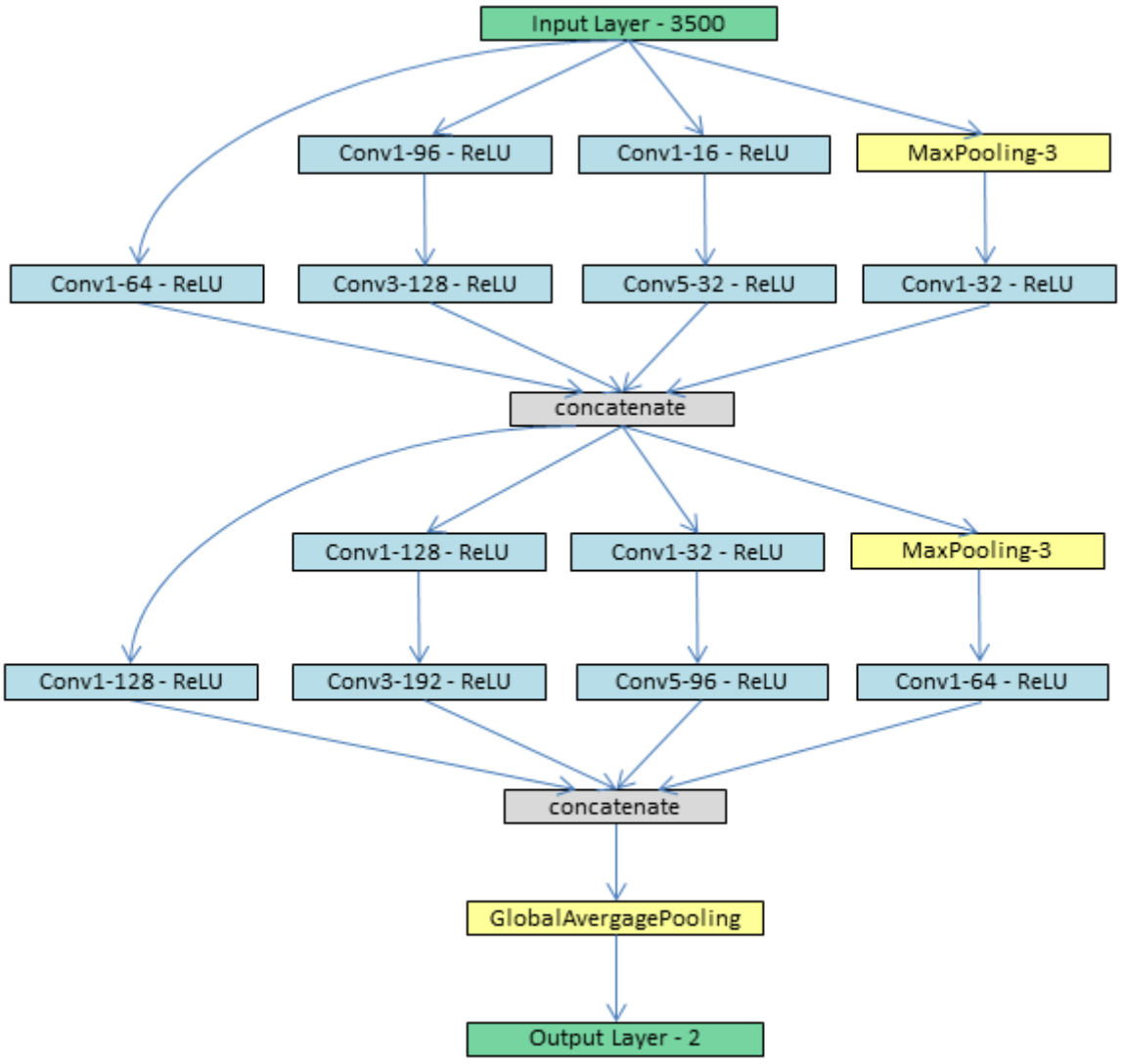


Figura 4.5: Red Inception

estrella.

Se han testeado redes recurrentes con 1, 2 y 3 niveles de profundidad, utilizando tanto celdas LSTM (*Long Short Term Memory*) como celdas GRU (*Gated Recurrent Unit*), y con distintos números de unidades en cada capa (8, 16, 32, 64, 128, 256 y 512).

En la figura 4.6 se muestran las redes con mejores resultados obtenidos.

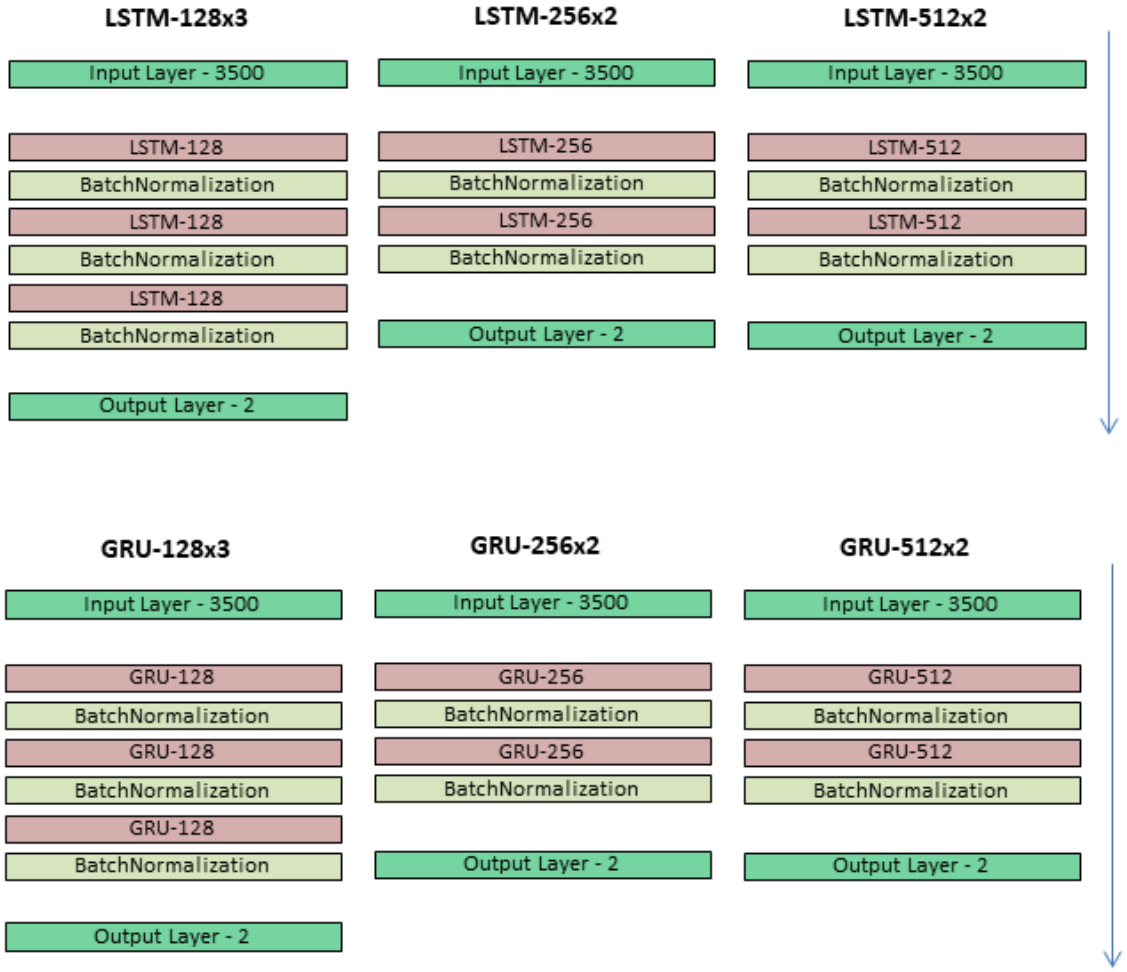


Figura 4.6: Redes recurrentes: LSTM - GRU

Tras una exploración inicial para identificar los mejores hiperparámetros (100 épocas, con un tamaño de lote de 32 y una tasa de aprendizaje de 0.005), se han conseguido los resultados mostrados en la tabla 4.5 con los que sólo se consigue una precisión del 73 %, y a costa de tener que entrenar modelos con un número de parámetros muy elevado que conlleva tiempos de entrenamiento acordes.

Redes recurrentes

modelo	parámetros	train_time	acc_train	acc_val	recall_val	best_acc_val
LSTM-128x3	331.522	2.293	82,47 %	57,58 %	81,82 %	68,69 %
LSTM-256x2	792.066	2.474	76,44 %	52,02 %	82,95 %	70,71 %
LSTM-512x2	3.156.994	5.563	72,83 %	62,63 %	15,91 %	72,22 %
GRU-128x3	250.242	2.022	99,72 %	67,17 %	67,05 %	72,73 %
GRU-256x2	596.226	2.028	98,99 %	61,11 %	77,27 %	73,74 %
GRU-512x2	2.372.098	4.229	100,00 %	65,15 %	64,77 %	71,72 %

Tabla 4.5: Resultados redes recurrentes LSTM y GRU

4.4.4. Redes mixtas

Hemos visto en los apartados anteriores distintos tipos de redes, bien totalmente conectadas, bien totalmente convolucionales o totalmente recurrentes, pero lo habitual es utilizar redes mixtas que recojan lo mejor de cada una de ellas, como por ejemplo, capas convolucionales para extracción de características seguidas de capas totalmente conectadas para realizar la clasificación final.

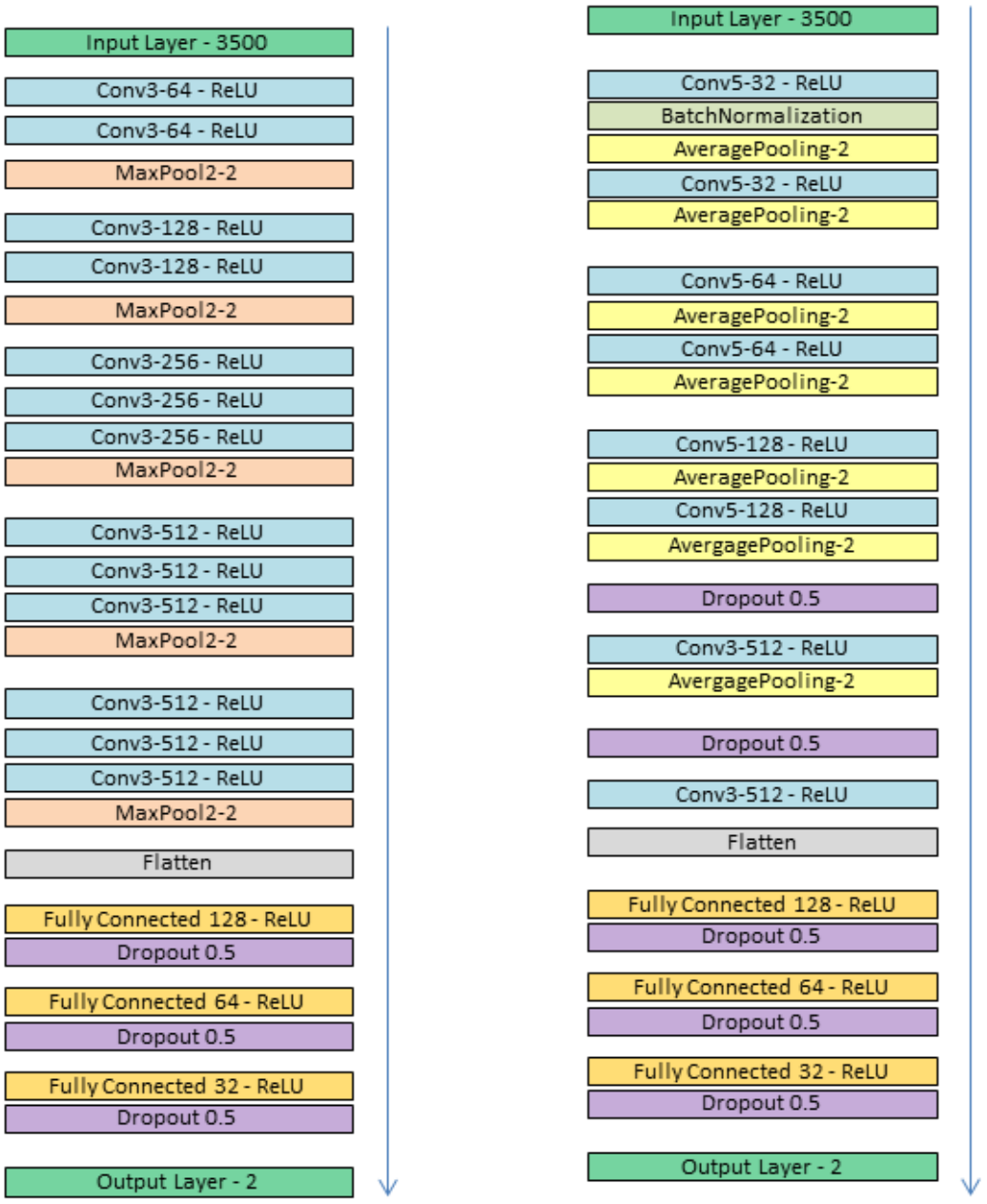
Se presentan a continuación diversas pruebas realizadas al respecto, todas ellas orientadas a mejorar los resultados de las redes anteriormente descritas.

4.4.4.1. VGG

Una primera aproximación ha consistido en utilizar alguna arquitectura conocida de red y reutilizarla para el tratamiento de las curvas de luz. Así, se ha procedido a adaptar una arquitectura clásica VGG (creada por el *Visual Geometry Group*, de la Universidad de Oxford), tal como se muestra en la figura 4.7 (a).

A destacar que, a diferencia de la arquitectura original, se ha optado por un número sensiblemente menor de neuronas en las capas totalmente conectadas. El motivo principal es el gran volumen de parámetros resultantes en el modelo original, 235.506.938. Se ha intentado entrenar el modelo con la red VGG original pero no ha sido posible por falta de recursos hardware que lo soporten. Indicar que este tipo de redes tan profundas suelen utilizarse partiendo de modelos preentrenados, lo cual no es posible en nuestro caso al trabajar con una versión 1D de la misma.

Con una red final con 11.470.386 parámetros no se ha conseguido una precisión superior al



(a) Red VGG

(b) Red inspirada en análisis de ECG

Figura 4.7: Redes mixtas

60 %, pese a ser testeada con diferentes hiperparámetros (tasas de aprendizaje de 0.01, 0.001 y 0.0001 y lotes con tamaño 16, 32 y 64), tal como se muestra en la tabla 4.6.

Red VGG

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,01	16	516	49,38 %	55,56 %	0,00 %	55,56 %
	32	460	50,51 %	50,00 %	50,00 %	55,56 %
	64	659	49,89 %	44,44 %	100,00 %	55,56 %
0,001	16	566	50,62 %	44,44 %	100,00 %	55,56 %
	32	432	50,62 %	44,44 %	100,00 %	50,00 %
	64	391	50,62 %	44,44 %	100,00 %	55,56 %
0,0001	16	524	74,69 %	53,54 %	81,25 %	58,84 %
	32	462	50,65 %	44,44 %	100,00 %	55,56 %
	64	491	50,45 %	44,44 %	100,00 %	56,07 %

Tabla 4.6: Resultados red VGG

4.4.4.2. Red mixta inspirada en modelos ECG - Electrocardiogramas

Otras combinaciones de redes mixtas analizadas combinan las descritas en las secciones anteriores, tanto las redes CNN como las redes RNN, añadiendo al final de las mismas 2 o 3 capas totalmente conectadas, siendo también los resultados de las mismas no satisfactorios.

Por analogía de las curvas de luz con otro tipo de datos relacionados con series temporales se han analizado modelos que tienen buenos resultados en entornos como en análisis de electrocardiogramas (ECG). Las similitudes son evidentes, pues en ese tipo de análisis se buscan precisamente anomalías, al igual que en las curvas de luz buscamos anomalías que serían los tránsitos planetarios. No menos cierto es que los datos en los electrocardiogramas son flujos continuos, mientras que en las curvas de luz, como hemos visto, la presencia de ruido o variabilidad en los datos es muy acuciada.

En esta línea, se ha adaptado la red propuesta por Chair-Heh Hsieh et al. (2020) [12], en la forma mostrada en la figura 4.7 (b), con la que se han obtenido los resultados mostrados en la tabla 4.7. Con promedios de 2 ejecuciones, los mejores resultados se acercan al 80 % siendo el hiperparámetro correspondiente a la tasa de aprendizaje el más influyente en los resultados.

4.4.4.3. Redes mixtas CNN - RNN

Otros tipos de arquitecturas de red evaluadas han sido combinaciones de capas convolucionales, para extracción inicial de características, seguidas de capas recurrentes, para finalizar

Red mixta CNN + FC

learning_rate	batch_size	train_time	acc_train	acc_val	recall_val	best_acc_val
0,01	16	90	49,27 %	44,44 %	100,00 %	55,56 %
	32	114	50,51 %	50,00 %	50,00 %	55,56 %
	64	66	49,04 %	44,44 %	100,00 %	55,56 %
0,001	16	116	50,51 %	44,44 %	100,00 %	55,56 %
	32	70	74,97 %	56,82 %	82,39 %	65,91 %
	64	64	99,89 %	71,47 %	62,50 %	78,03 %
0,0001	16	118	99,21 %	73,99 %	55,12 %	79,55 %
	32	95	99,33 %	71,72 %	66,48 %	78,03 %
	64	89	98,62 %	77,27 %	59,09 %	78,79 %

Tabla 4.7: Resultados red mixta inspirada en ECG

con capas totalmente conectadas.

En la figura 4.8 se muestran un par de estas redes, y en la figura 4.9 se muestra otro tipo de red, en este caso una red dual: por un lado una red recurrente de 2 capas, con celdas LSTM y por otro lado, una red convolucional, que convergen en unas capas completamente conectadas, con el objetivo de obtener lo mejor de ambas aproximaciones.

Los resultados de estas redes mixtas CNN-RNN, con capas completamente conectadas en su parte final son los mostrados en la tabla 4.8, donde puede observarse que los resultados para las distintas redes son bastante estables, todos ellos con precisiones entre el 75 % y el 80 %.

4.4.5. Autoencoders

Los autoencoders son un tipo especial de redes neuronales que funcionan intentando reproducir los datos de entrada en la salida de la red, es decir:

- Aceptan como entrada un conjunto de datos.
- Comprimen la información en una representación interna (*latent space*)
- Reconstruyen la entrada a partir de la representación interna.

Son varias las aplicaciones de este tipo de redes, como: la reducción de la dimensionalidad de los datos, la generación de datos sintéticos, etc. Uno de los posibles usos de estas redes es la detección de anomalías, donde tenemos una clase mayoritaria y otra apenas residual, es decir, como en nuestro caso, clases muy desbalanceadas, y queremos justamente detectar los casos

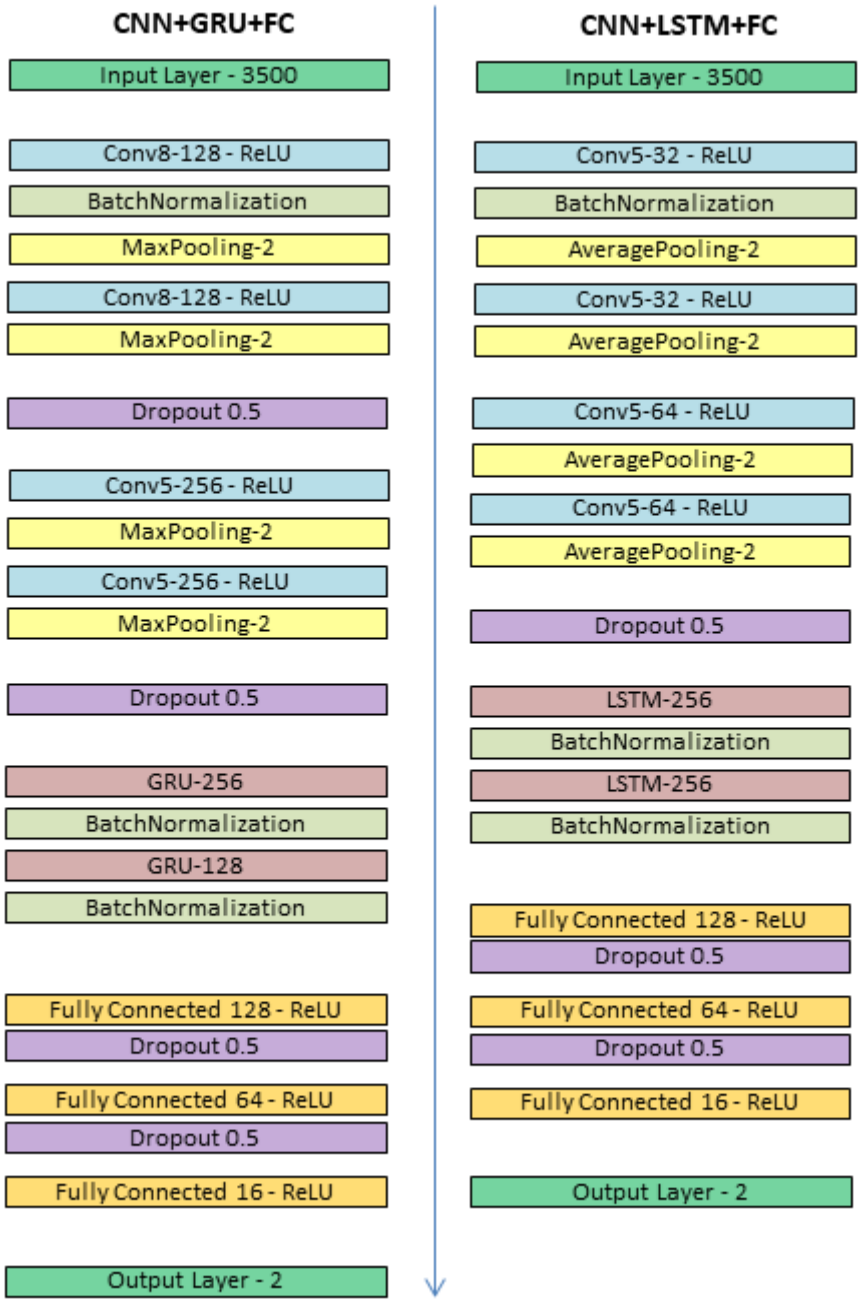


Figura 4.8: Redes mixtas CNN+RNN+FC

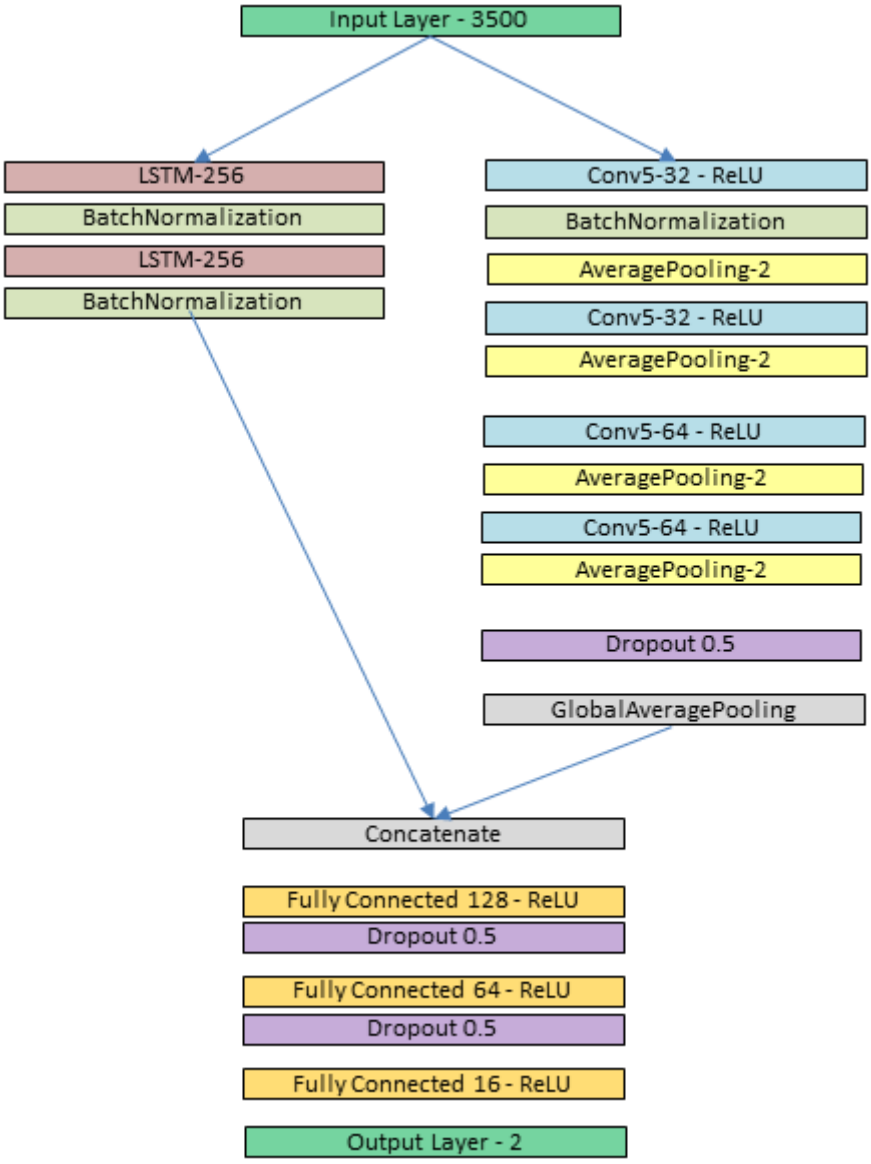


Figura 4.9: Red dual CNN+RNN

Redes mixtas CNN+RNN+FC

modelo	learning rate	batch size	train_time	acc_train	acc_val	recall_val	best_acc_val
CNN+GRU	0,0001	32	284	86,91 %	69,36 %	75,00 %	78,45 %
		64	216	82,43 %	59,43 %	89,01 %	76,60 %
	0,0005	32	285	95,88 %	74,75 %	62,88 %	79,29 %
		64	216	95,98 %	54,55 %	87,50 %	77,27 %
	0,001	32	286	92,13 %	72,06 %	52,65 %	78,45 %
		64	216	95,09 %	67,85 %	45,46 %	78,45 %
CNN+LSTM	0,0001	32	215	83,73 %	71,89 %	40,53 %	80,81 %
		64	148	82,45 %	78,28 %	56,82 %	79,46 %
	0,0005	32	217	97,11 %	69,70 %	70,83 %	76,10 %
		64	148	95,81 %	66,16 %	24,62 %	75,93 %
	0,001	32	217	90,04 %	70,03 %	57,20 %	78,95 %
		64	148	94,51 %	70,54 %	64,39 %	77,27 %
DUAL CNN+LSTM	0,0001	32	1.189	71,63 %	72,73 %	50,38 %	75,59 %
		64	758	72,81 %	72,73 %	48,87 %	73,90 %
	0,0005	32	1.193	81,12 %	77,44 %	63,26 %	80,47 %
		64	756	79,23 %	75,42 %	51,89 %	78,45 %
	0,001	32	1.204	85,32 %	64,48 %	81,44 %	79,12 %
		64	757	82,68 %	73,74 %	45,83 %	79,12 %

Tabla 4.8: Resultados redes mixtas CNN+RNN+FC

positivos, que son muy escasos.

La idea es la siguiente: si entrenamos el modelo únicamente con datos de la clase negativa y si posteriormente aplicamos al modelo un elemento de la clase positiva, al intentar reconstruir los datos a partir de la representación interna debería dar un error de reconstrucción elevado. Para calcular este error normalmente se utiliza el error cuadrático medio (MSE - *Mean Square Error* 4.3).

$$MSE_{\text{LOSS}} = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 \tag{4.3}$$

Así, fijando un umbral de error se podrían discriminar, en base al error de reconstrucción, los elementos de clase negativa (error bajo), y los de la clase positiva (error alto).

En definitiva, el proceso a seguir será el siguiente:

1. Entrenar el modelo únicamente con casos de la clase negativa.

2. Calcular el error máximo de reconstrucción de los valores de entrenamiento (diferencia

entre la curva de luz de entrada y la curva reconstruida). Esto determinará el umbral de error de reconstrucción.

3. Aplicar el modelo a los datos a validar: si el error de reconstrucción supera el umbral anterior, anotarlo como clase positiva (exoplaneta).

Los modelos con Autoencoders pueden ser implementados con cualquier tipo de red. Desde redes totalmente conectadas, redes convolucionales, redes recurrentes, o cualquier combinación de ellas. Entre las pruebas realizadas mencionar algunas de ellas:

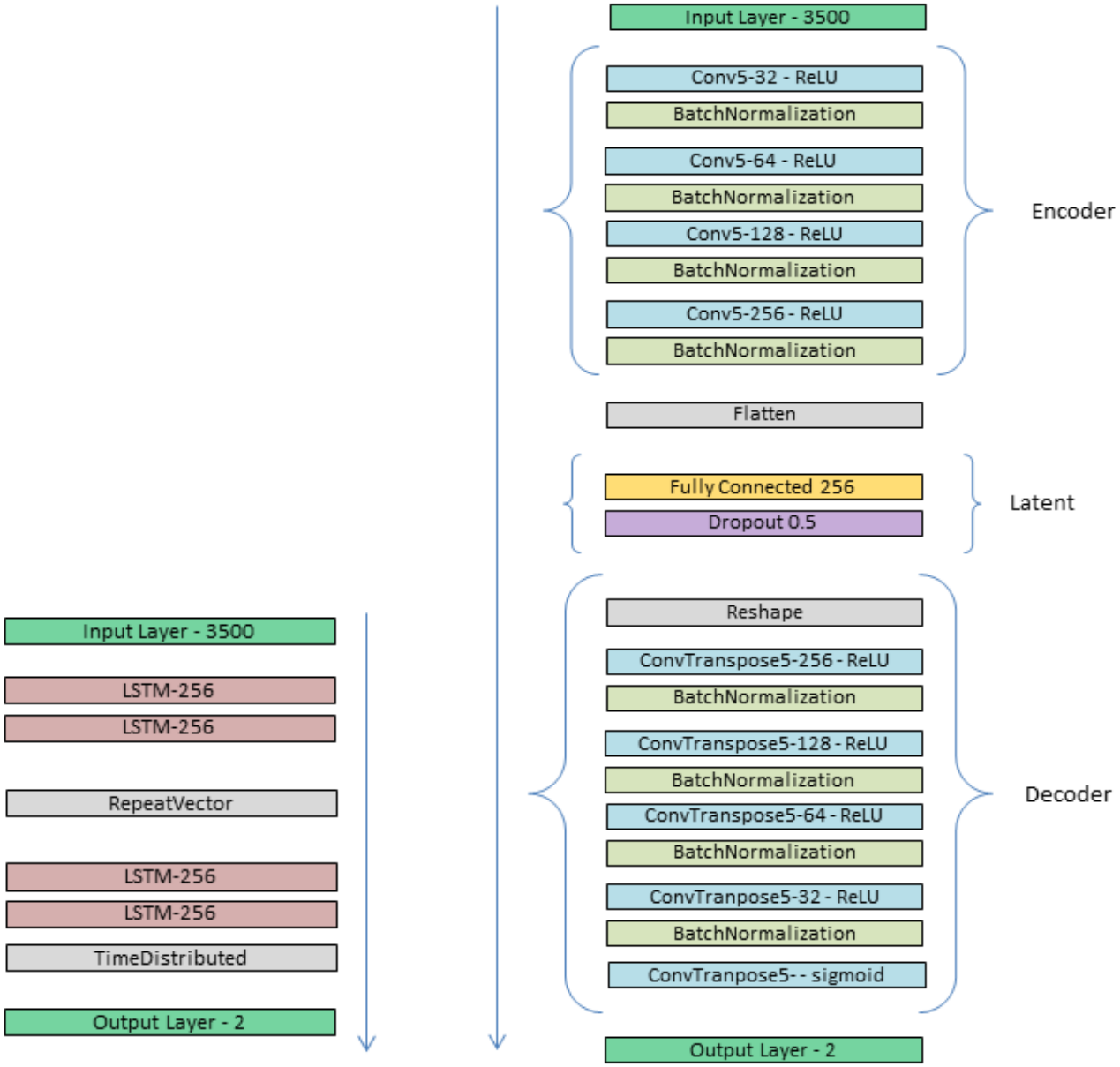
- Redes completamente conectadas: 1, 2 o 3 capas, con 32, 64, 128, 256 neuronas por capa.
- Redes completamente convolucionales: número de capas convolucionales (1,2,3,4), con o sin capas de agrupamiento (*pooling*), con diferentes combinaciones de valores de filtros (*kernel*): 32, 64, 128, 256, diferentes pasos (*stride*), ...
- Redes completamente recurrentes, con celdas LSTM: 1 o 2 capas para la capa de codificación (*encoder*), y lo mismo para la capa de decodificación (*decoder*), número de celdas: 64, 128 y 256.
- Redes mixtas convolucionales con capas intermedias totalmente conectadas, y diversas combinaciones como las recién mencionadas.

Y todo lo anterior con diversas combinaciones de hiperparámetros aplicados al entrenar los correspondientes modelos: *batch_size*, *learning_rate*, *epochs*.

En la figura 4.10 se muestran un par de arquitecturas Autoencoders probadas: una basada en una red recurrente y otra con una combinación de capas convolucionales además de una capa interna totalmente conectada.

Para esta última red, una de las que ofrece una menor pérdida en fase de entrenamiento, el error de reconstrucción para las curvas de luz con exoplanetas o sin exoplanetas es prácticamente el mismo. Ello apunta a que la variación o ruido en las propias curvas de luz, tomadas en su totalidad, anula el error producido por los posibles tránsitos planetarios. Por ello, se probó, para intentar focalizar el error de reconstrucción en los puntos más dispares entre la curva original y la reconstruida (los posibles tránsitos), calcular también el error MSE para los 50 puntos con mayor error, pero incluso con esta limitación adicional apenas pueden diferenciarse los elementos de la clase positiva de los de la clase negativa.

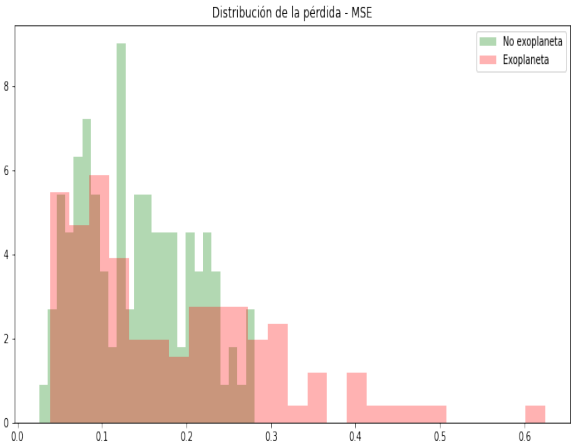
En la figura 4.11 (a) se muestra el error MSE de los datos de validación tras aplicar el autoencoder y calcular el error entre la curva original y la reconstruida.



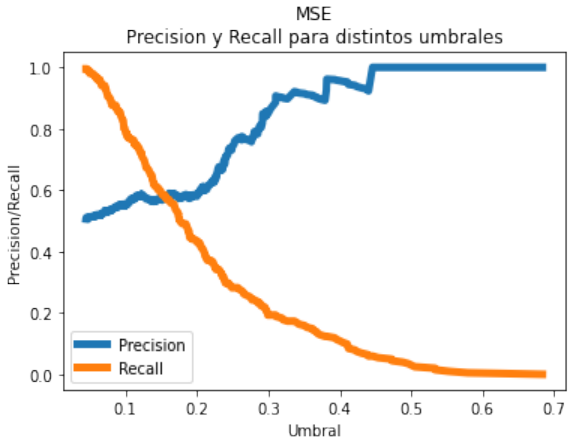
(a) Autoencoder LSTM

(b) Autoencoder CNN + FCN

Figura 4.10: Autoencoders



(a) Pérdida MSE entre clases



(b) Discriminación Precision-Recall con umbral pérdida

Figura 4.11: Análisis datos autoencoders

Sería esperable encontrar un umbral de error que separara ambas clases. Tal y como se comentó inicialmente, ese umbral debería fijarse en base al error máximo encontrado al reconstruir los datos de entrenamiento a partir de los originales, utilizando exclusivamente elementos de la clase negativa.

Una alternativa para elegir el umbral de error para separar la clase positiva y negativa podría consistir en visualizar gráficamente los valores de precisión (*accuracy*) y *recall* en función de la aplicación de distintos umbrales de error, tal como se muestra en la figura 4.11 (b). Ajustando el umbral a la intersección de ambas curvas puede obtenerse una precisión (*accuracy*) del 58 %, con un *recall* del 57 %.

En definitiva, el uso de autoencoders, dado el ruido intrínseco presente en las curvas de luz impide su uso para la identificación de curvas de luz que contengan tránsitos planetarios.

4.5. Resultados

En la sección anterior hemos visto diversas arquitecturas de redes neuronales para intentar identificar qué curvas de luz contienen exoplanetas, con los resultados parciales obtenidos durante la búsqueda de los mejores hiperparámetros.
Para los mejores modelos obtenidos se ha procedido, para tener unos resultados más fiables, a aplicar una validación cruzada (*K-Fold validation*), con 10 *folds*, para posteriormente, reentrenar cada mejor modelo y, por último, realizar la validación final contra los datos de test, no utilizados en ningún momento previamente.

La tabla 4.9 muestra los resultados finales, mostrando el promedio de la validación cruzada en entrenamiento, y el resultado final sobre los datos de test.

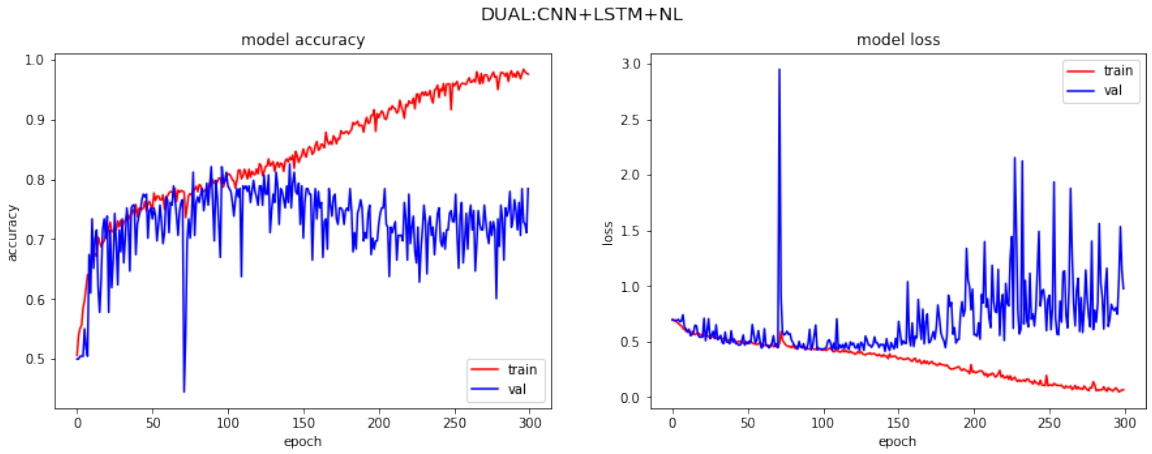
Mejores modelos - Validación cruzada y Test

modelo	learning rate	batch size	acc train	acc val	best acc val	acc test	recall test
CUSTOM ECG	0,0005	32	98,80 %	73,02 %	79,31 %	80,73 %	70,64 %
RESNET	0,0001	32	99,71 %	65,61 %	77,23 %	79,82 %	66,06 %
CNN+LSTM	0,0001	32	95,74 %	73,02 %	80,38 %	79,36 %	73,39 %
DUAL CNN+LSTM	0,0005	32	83,53 %	73,02 %	79,67 %	82,57 %	71,56 %

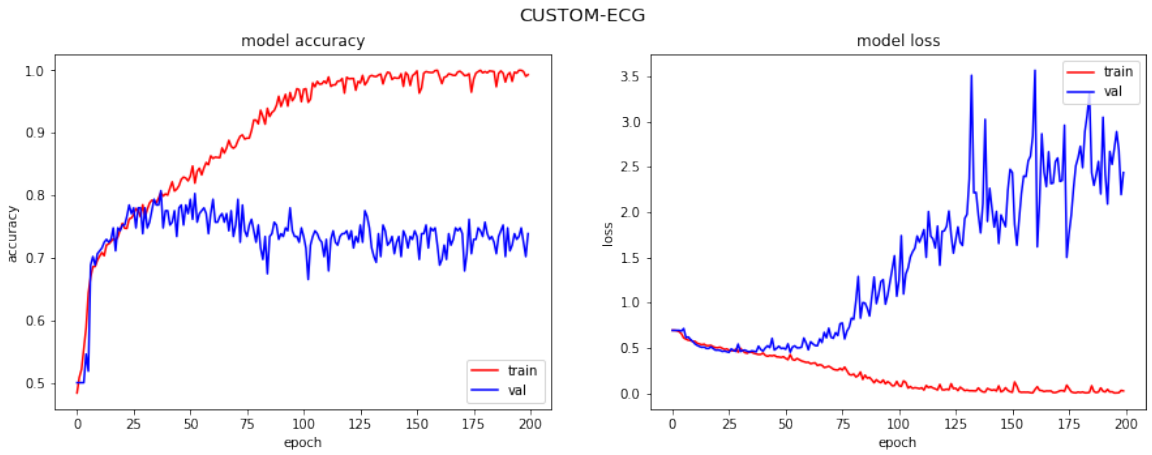
Tabla 4.9: Resumen mejores resultados

Puede comprobarse que todos los modelos presentan una precisión cercana al 80 %, el cual parece ser el límite máximo que se puede alcanzar, siendo el modelo dual mixto que combina una red con capas convolucionales, con capas recurrentes y finalizando con capas completamente conectadas, el que mejor resultado ofrece.

De los dos mejores modelos (CNN + LSTM DUAL y CUSTOM ECG) podemos observar en la figura 4.12 sus curvas de aprendizaje en cuanto a la precisión (*accuracy*) como a la pérdida (*loss*), tanto sobre el conjunto de datos de entrenamiento como el de validación.



(a) DUAL CNN+LSTM

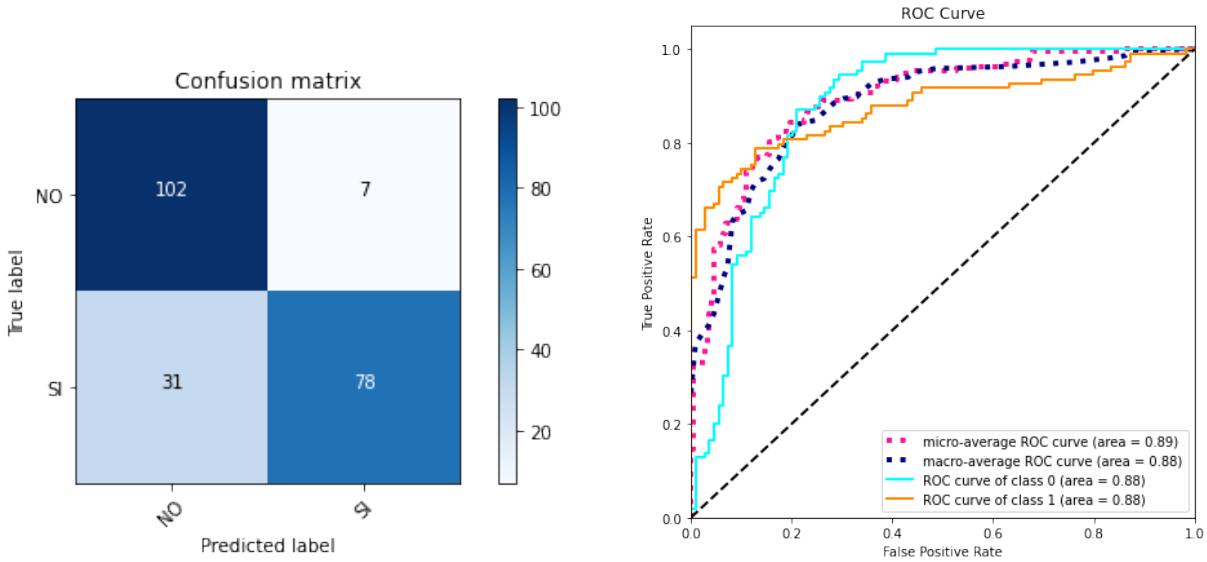


(b) CUSTOM ECG

Figura 4.12: Precisión y pérdida durante el entrenamiento

Las curvas de aprendizaje mostrarían signos de sobreajuste (*overfitting*), pues si bien en los datos de entrenamiento la precisión consigue llegar al 100 % no ocurre lo mismo con los datos de validación donde no se consigue en ningún caso superar la frontera del 83 %. Señalar igualmente que estos modelos incluyen diversas capas de normalización (*batch normalization*) y de *dropout*, para intentar evitar este aparente sobreajuste que también podría venir derivado del poco volumen de curvas de luz en el dataset.

Podemos visualizar igualmente la matriz de confusión resultante al aplicar el modelo a los datos de test, así como la curva ROC (*Receiver Operating Characteristic*)(ver figura 4.13).



(a) Matriz de confusión (b) Curva ROC

Figura 4.13: Validación final con datos de test

La matriz de confusión, como ya vimos en el apartado 4.2.2, presenta en una tabla una visión gráfica de los errores cometidos por el modelo de clasificación, donde se observa el valor del *recall* obtenido de 71,56 %, correspondiente al haber clasificado correctamente a 78 curvas de luz como clase positiva (con exoplanetas), de un total de 109. Por su lado, la curva ROC mide el rendimiento respecto a los falsos positivos y los verdaderos positivos en diferentes umbrales de clasificación y es una excelente manera de visualizar el desempeño de un clasificador binario. A partir de la curva ROC se calcula el área bajo la curva o AUC (*Area Under the Curve*) que permite caracterizar el rendimiento del modelo de clasificación, valor que en nuestro caso es de 0,88, valor que puede considerarse como "bueno", de acuerdo a la siguiente clasificación estándard:

[0,5; 0,6)	Malo
[0,6; 0,75)	Regular
[0,75; 0,9)	Bueno
[0,9; 0,97)	Muy bueno
[0,97; 1)	Excelente

Tabla 4.10: Clasificación estándar valor AUC

Dado que, obviamente, el objetivo principal de nuestro clasificador es detectar curvas de luz conteniendo exoplanetas, podría optarse por reducir el umbral de clasificación, para así tener menos falsos negativos (31 en la matriz de confusión), aún a costa de aumentar los falsos positivos (7 en la matriz de confusión).

Capítulo 5

Interpretabilidad

Uno de los problemas tradicionales de las redes neuronales es su opacidad, es decir, saber el por qué una red neuronal toma una determinada decisión. Presentan, pues, grandes problemas en la interpretabilidad de los modelos.
Para intentar mitigar esta problemática, Selvaraju et al. [31] crearon lo que denominaron *Gradient-weighted Class Activation Mapping*, o de modo abreviado, Grad-CAM, un algoritmo que permite identificar dónde está mirando el modelo para realizar su pronóstico. Así, si aplicamos este algoritmo sobre nuestros modelos podríamos identificar qué partes de la curva de luz son las que acaban influyendo en la decisión final de la red, siendo esperable que estas partes, además, se correspondan con los posibles tránsitos planetarios.

Grad-CAM utiliza los gradientes fluyendo hacia la capa convolucional final para producir un mapa de localización aproximado que resalta las regiones importantes en la imagen para predecir la decisión final. De un modo simplificado, Grad-CAM consiste en encontrar la capa convolucional final en la red y examinar la información de gradiente que fluye hacia esa capa. La salida de Grad-CAM es una visualización de mapa de calor para una etiqueta de clase determinada y podemos usar este mapa de calor para verificar visualmente en qué parte de la imagen está mirando la red convolucional. Utilizando los gradientes de la capa convolucional final, podemos tomar el promedio de sus pesos (la importancia de los gradientes) de los posibles filtros de dicha capa (es decir, multiplicando cada mapa de activación por sus pesos correspondientes). Así, si un mapa de activación se ha activado durante el pase hacia adelante y si sus gradientes son grandes, significa que la región que se activa tiene un gran impacto en la decisión final del modelo. Esta información podemos visualizarla y así ver qué partes de la entrada influyen más en la determinación sobre si la curva de luz recoge algún exoplaneta o no.

Pese a que el mejor modelo obtenido es el DUAL:CNN+LSTM, aplicaremos Grad-CAM

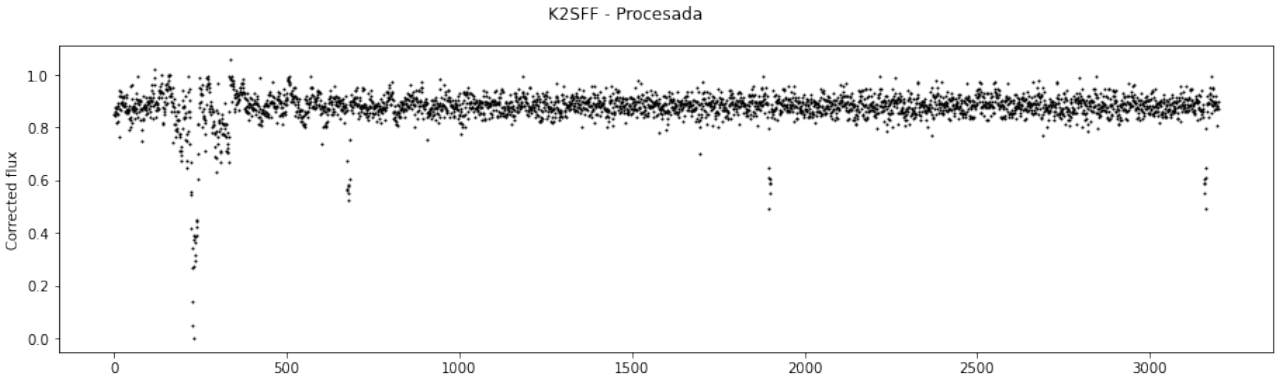
utilizando el modelo que utiliza una red con estructura ResNet.

Como se ha indicado Grad-CAM utiliza la última capa convolucional del modelo (o alguna de las últimas), y en los modelos combinados CNN+RNN (4.8 y 4.9) podríamos visualizar esta información, pero dado que las capas convolucionales posteriormente se combinan con capas recurrentes tendríamos una visión parcial de la información, es decir, observaríamos un punto ciertamente muy inicial en el proceso de decisión del modelo.

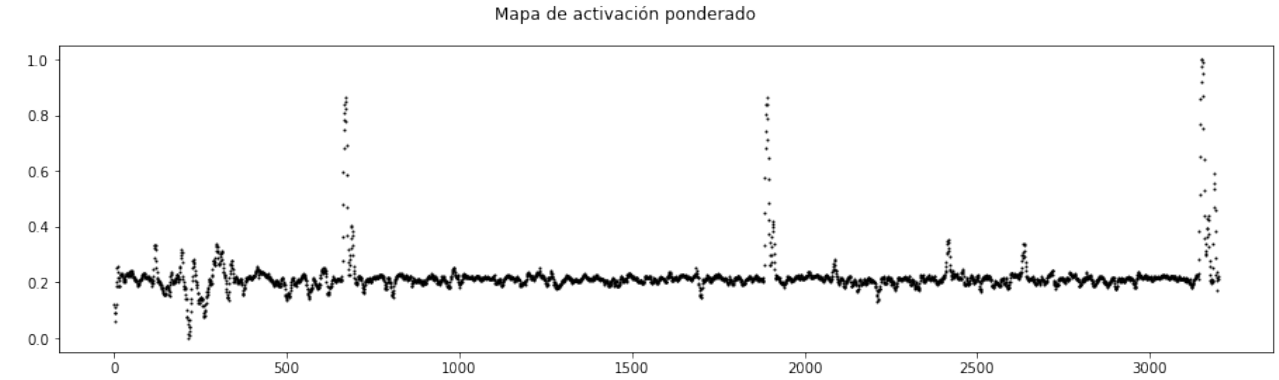
Por otro lado, en el modelo CUSTOM-ECG, como puede observarse en su diagrama (4.7 (b)), todas sus capas convolucionales van seguidas de una capa de agrupamiento (*pooling*), con *stride*=2), con lo que la dimensión del flujo de datos se va reduciendo en cada una de estas capas a la mitad, aunque aumentando a su vez los filtros aplicados: 32, 64, 128, 256 y 512. Así, observando alguna de las últimas capas convolucionales de este modelo la dimensión de dichas capas serían muy reducidas, y ofrecerían, al reescalar el mapa de activación correspondiente al tamaño de la curva de luz original, áreas muy grandes, y por lo tanto, muy poco precisas. Para obtener información adecuada debería utilizarse una de las capas convolucionales iniciales del modelo, lo que tampoco sería especialmente adecuado.

Por el contrario, la red con estructura ResNet no contiene ninguna capa de agregación, con lo cual la dimensión de las capas convolucionales coincide siempre con el tamaño de la entrada inicial del modelo, siendo por lo tanto más adecuado este modelo para aplicar Grad-CAM. En todo caso, la diferencia entre la precisión obtenida entre los distintos modelos es muy similar, y la utilización de la red con estructura ResNet puede ser, por el motivo descrito, mucho más ilustrativa.
Podemos observar, en la figura 5.1, la aplicación de Grad-CAM con este modelo para el objeto EPIC 201092629. En la parte superior de la imagen, tenemos la curva de luz de entrada al modelo. En la parte central, el mapa ponderado de los filtros de acuerdo a la importancia de los gradientes de la última capa convolucional del modelo, y en la parte inferior la superposición de la curva de luz utilizando el mapa de activación como un mapa de calor, y efectivamente, podemos comprobar que el modelo decide que la curva contiene un exoplaneta, focalizando la decisión justamente en tres aparentes tránsitos planetarios, descartando un potencial tránsito al inicio de la curva, probablemente producido por ruido.

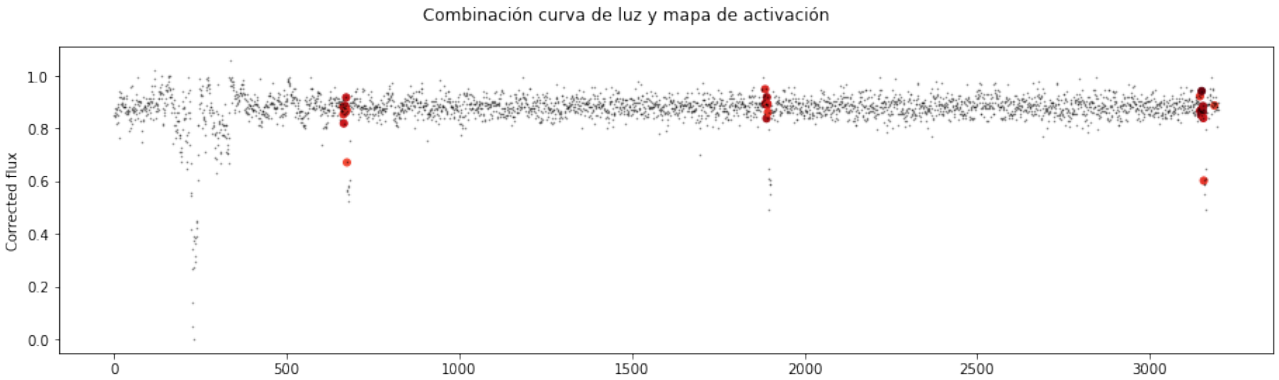
Otros casos de aplicación serían los siguientes: el primero, un exoplaneta correctamente clasificado (figura 5.2), el segundo, un falso negativo (figura 5.3), y el tercero, un falso positivo (figura 5.4).



(a) Curva de luz de entrada



(b) Mapa de activación ponderado



(c) Grad-CAM aplicado

Figura 5.1: Grad-CAM aplicado a EPIC 201092629 - Exoplaneta confirmado

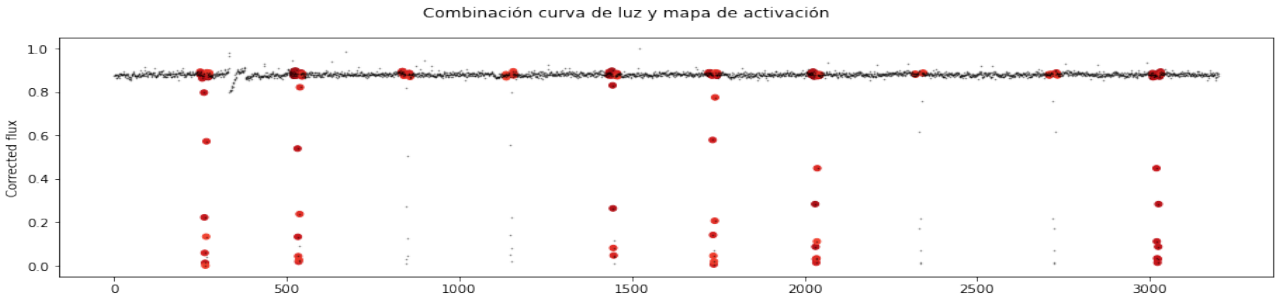


Figura 5.2: Grad-CAM aplicado a EPIC 228735255 - Exoplaneta confirmado

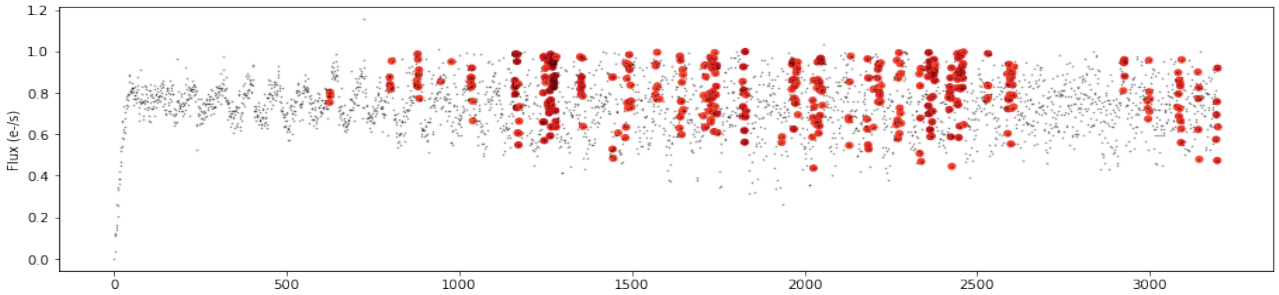


Figura 5.3: Grad-CAM aplicado a EPIC 212524671 - Falso negativo

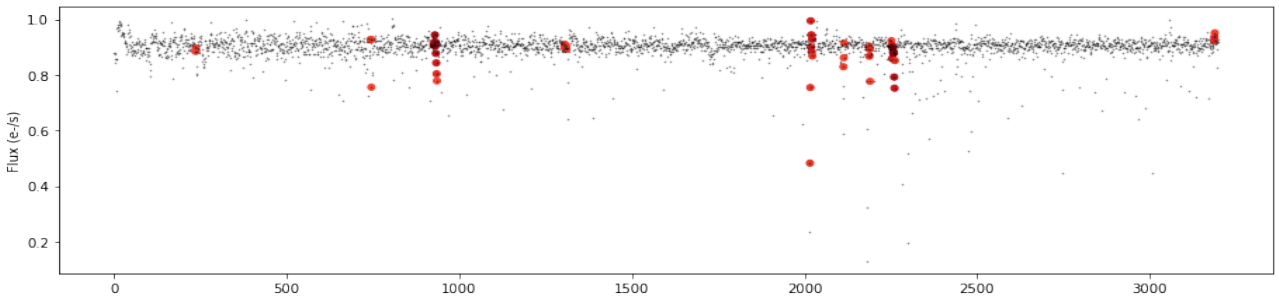


Figura 5.4: Grad-CAM aplicado a EPIC 205668963 - Falso positivo

Capítulo 6

Conclusiones y líneas de trabajo futuras

Para concluir, en este capítulo presentamos un resumen de las principales conclusiones que responden a las cuestiones planteadas al inicio del proyecto, además de incluirse potenciales líneas de trabajo futuro.

6.1. Conclusiones

Planteábamos al inicio del proyecto una pregunta: ¿es posible determinar si a partir de una curva de luz de una estrella, es decir, la luminosidad aparente de una estrella a lo largo de un periodo de tiempo, y únicamente con esa información, si existe un exoplaneta orbitando a la misma? No existe una respuesta concluyente al respecto, pues hemos podido comprobar que la precisión obtenida apenas supera el 80 %. Aplicar un modelo como los obtenidos, por ejemplo para un mecanismo totalmente automatizado, podría descartar quizás demasiados exoplanetas, y al tiempo generar un número significativo de falsos positivos, con lo que tanto en un caso como en otro se requerirían procesos adicionales.
No menos cierto es que cualquier modelo predictivo, incluido el nuestro, sólo será capaz de reconocer patrones específicos en los datos para los cuales ha sido entrenado, con lo que ningún modelo de detección de exoplanetas será completo. En definitiva, los modelos serán tan buenos como lo sean los datos de entrenamiento utilizados. Así, por ejemplo, podrían aparecer tránsitos de exoplanetas cuya presencia en las curvas de luz sean distintas a las habituales, o para los modelos que se basan fuertemente en la periodicidad de los tránsitos, encontrar exoplanetas con variaciones temporales en los mismos que podrían distorsionar las predicciones realizadas, etc.

¿Cuál sería en todo caso un resultado satisfactorio? Para Dattilo et al. (2019) [6] idealmente un sistema de identificación de exoplanetas debería tener un *recall* de al menos un 90 % y una precisión (*accuracy*) superior al 95 % para introducir errores en las tasas de ocurrencia del planeta inferiores al 5 % mientras permanecen sensibles a la mayoría de los candidatos a planetas de la muestra. Con su modelo, AstroNet-K2, un referente sobre los datos de la misión K2, obtienen una precisión del 98 %, con un AUC del 0.988, lo cual es un valor ciertamente notable.
Pero como ya hemos visto, este tipo de modelos primero deben obtener un catálogo de TCEs (para lo cual utilizan el algoritmo BLS), que posteriormente deben ser etiquetados, en su caso, manualmente, para preparar el *dataset* final que utilizan. No facilitan métricas de este proceso, con lo que el resultado final obtenido vendrá directamente condicionado por la calidad real de su *dataset*. Los mismos autores asumen que su modelo no podría ser utilizado para un modelo totalmente automatizado.
Igualmente, cualquier método basado en la identificación previa de TCEs mediante BLS o algoritmos similares, está condicionado fuertemente por la repetición de tránsitos (uno de los pilares del algoritmo BLS, ver anexo A.1). Es este quizás el principal hándicap de estas aproximaciones que parten de los TCEs: con observaciones más acotadas temporalmente limitan mucho su potencial, si bien, no es menos cierto que, justamente, la periodicidad de los tránsitos es un punto clave para la confirmación y caracterización de exoplanetas.

En todo caso, puede ser interesante una aproximación *end-to-end* como la abordada en el presente trabajo para identificar candidatos, incluso sin presencia de periodicidad, para poder ser estudiados más tarde con otros mecanismos con más detalle, en búsqueda de tránsitos repetitivos y su posible categorización.
El presente trabajo, en su conjunto, puede verse así como una prueba de concepto inicial sobre esta aproximación *end-to-end* no basada en la identificación previa de tránsitos TCEs, prueba que debería continuar su desarrollo mediante su aplicación a datos de la misión TESS, como veremos a continuación, más restrictivos si cabe temporalmente que los datos de la misión K2, que ya lo eran frente a los datos de larga duración de la misión Kepler original.

Por último señalar que el esfuerzo previsto para elaborar el presente trabajo ha superado con creces las estimaciones iniciales, si bien en contrapartida las motivaciones de elección de la temática de este TFM se han cumplido totalmente, tanto por el hecho de abordar, y sobre todo aprender, un ámbito de la astronomía de especial interés, como por el poder abarcar todo el ciclo del proyecto, desde la preparación del *dataset* hasta la obtención de los modelos predictivos finales.

6.2. Líneas de trabajo futuras

La principal línea de trabajo futuro se centraría en la aplicación de las experiencias y métodos obtenidos en el presente trabajo a otras misiones también centradas en la identificación de exoplanetas, y en concreto, a la misión TESS [20]. Esta misión no tiene las problemáticas vistas en la misión K2, en relación a la calidad de datos, por lo que es de esperar que los preprocesados necesarios sobre las curvas de luz sean mucho menos exigentes. No obstante, tiene otras particularidades que requerirían otro tipo de adaptaciones.

En relación a la preparación del *dataset*, al igual que en la misión K2 se dispone de catálogos de exoplanetas confirmados, candidatos y falsos positivos, y, como en la misión Kepler original, catálogos de TCEs, que denominan TOI (*Tess Object of Interest*). Para el estudio y preparación del *dataset* se requeriría un cruce de datos doble, ya que la misión está organizada por la observación de sectores del cielo, y a diferencia de las campañas de la misión K2, éstos no son disjuntos. Así, una misma estrella puede ser observada en diferentes momentos y sectores, pudiéndose disponer de distintas curvas de luz. Así, el saber que una estrella tiene un exoplaneta no es suficiente; para saber cuál de las curvas es la adecuada se requiere examinar el catálogo de TOI, pues ahí sí se identifica tanto la estrella como el sector donde se observó el tránsito. Y con esa información descargar el fichero FITS adecuado e incorporarlo al *dataset*.

Otra adaptación imprescindible sería adaptar el preprocesado de las curvas, y en especial su aplanamiento (eliminación de la tendencia), pues el periodo entre muestras no es de 30 minutos como en la misión K2 sino de tan solo 2 minutos, y como vimos, la frecuencia de las muestras incide directamente en la configuración del procedimiento de aplanado de las curvas.

Y un último factor a considerar es el relativo a las duraciones de las observaciones, o sea, las longitudes de las curvas de luz, que son de 27 días en la misión TESS en la gran mayoría de los casos. Es decir, tenemos una frecuencia de datos mucho mayor (cada 2 minutos en vez de 30) pero una duración total inferior (27 días frente a los 80 días de la misión K2). Ello implicaría por lo tanto ajustes en la dimensión de entrada de los modelos.

Visto lo anterior, el adaptar y aplicar los mecanismos descritos en el presente trabajo para la misión TESS, teniendo en cuenta el hándicap de calidad que presentan las curvas de luz de la misión K2 (que sobre el papel no presentan las curvas en TESS) permitiría conocer si los resultados obtenidos, que no superan una precisión del 80 %, son una limitación de los métodos planteados o más bien fruto de la calidad de los datos analizados; y en general, ver la utilidad real de una aproximación *end-to-end* como la propuesta.

Apéndice A

Anexos

A.1. BLS - *Box-fitting Least Squares*

El periodograma BLS - *Box-fitting Least Squares Squares* (Kovacs et al. 2002) [16] se desarrolló para encontrar la periodicidad de tránsitos en las curvas de luz. BLS es un método que se basa en buscar o ajustar, a través de mínimos cuadrados, una función escalonada de 2 valores con un único intervalo diferente a la serie de tiempo cuando en la serie ocurre un tránsito, es decir, cuando baja la intensidad de la luz.
Así, supongamos que tenemos una curva de luz que contiene un tránsito repetido periódicamente con el periodo P. A partir de la curva de luz podemos construir una curva de fase que puede ser representada por una función escalonada:

$$f(n) = \begin{cases} 1 & \text{si t} < t_0 \text{ - } \frac{q}{2}\text{P} \\ 1-d & \text{si t} > t_0 + \frac{q}{2}\text{p} \end{cases} \tag{A.1}$$

Donde d es la profundidad del tránsito, t_0 el centro del tránsito y q la razón de la duración del tránsito durante el período P.

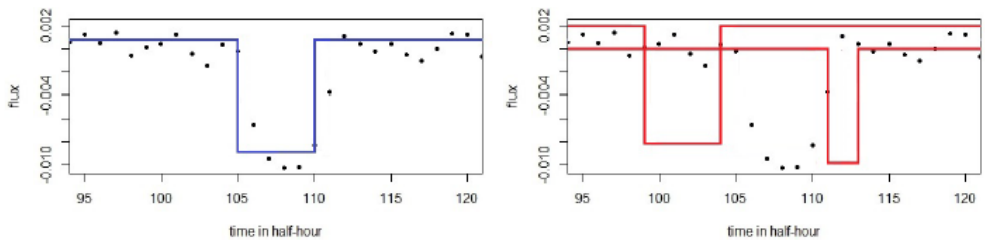


Figura A.1: BLS - Función escalonada [30]

Podemos describir BLS como un procedimiento que, para cada periodo de prueba P_0, se ajusta la función de paso anterior para diferentes valores de d, q y t_0, tal como se muestra en la figura A.1

Los distintos ajustes devuelven un valor SR (*Signal Residue*, o residuo de señal), que denota la calidad del ajuste. El periodo de prueba para el cual el factor SR es máximo es el periodo correcto del tránsito. Como puede verse en la figura A.2, BLS es un periodograma muy potente para detectar tránsitos, incluso aquellos con señal del tránsito pobre comparada al nivel de ruido. En dicha figura se muestra arriba, la curva de luz. Abajo a la izquierda, el espectro de frecuencia BLS normalizado y finalmente abajo a la derecha, la curva de luz plegada.

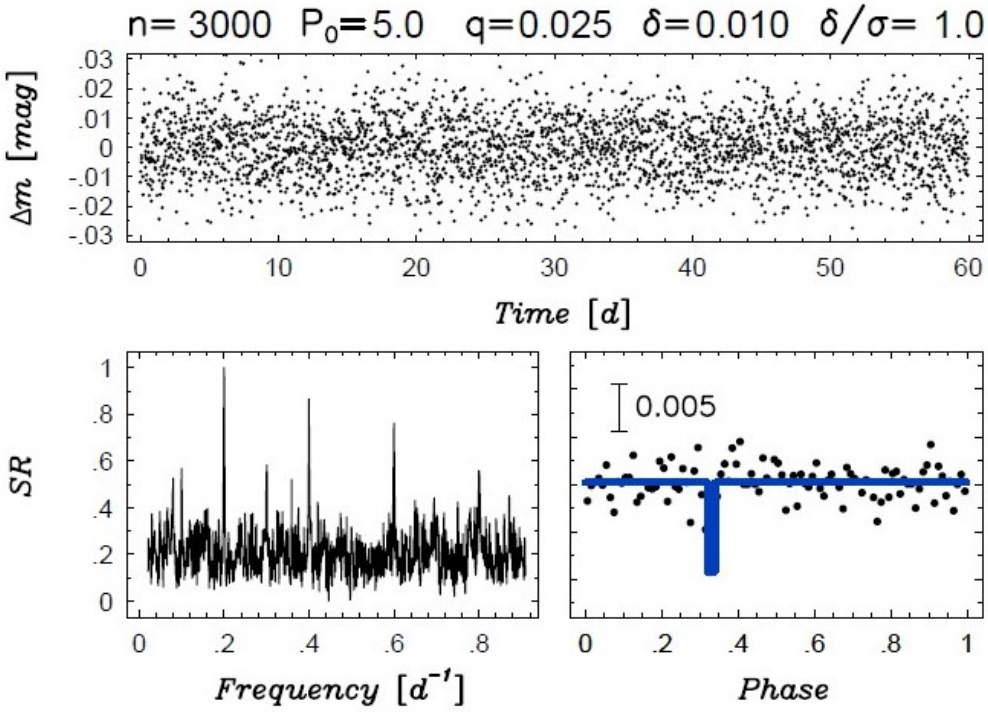


Figura A.2: Ejemplo de aplicación del algoritmo BLS [16]

Otro factor importante al aplicar el algoritmo BLS es que los datos a los que se aplique deben contener tres o más tránsitos, ya que es lo que se necesita para confirmar que se trata de un tránsito planetario, y en caso de no haberlos producirá errores o no lo detectará bien.

A.2. Fuentes de datos K2SFF y EVEREST

Como se ha comentado en el trabajo, debido a la inestabilidad del satélite en su modo de funcionamiento durante la misión K2, las curvas de luz en bruto obtenidas exhiben grandes características sistemáticas que podrían llegar a impedir la detección de tránsitos planetarios.

Tanto es así que debido a la precisión de orientación reducida, la fotometría de apertura bruta en K2 es entre 3 y 4 veces menos precisa que la de la misión Kepler original.
Así, varios equipos han preparado procesos para intentar reducir estas problemáticas y conseguir curvas de luz de precisión parecida a las facilitadas por la misión original del satélite Kepler.
Este sería el caso de las curvas de luz K2SFF (*K2 Extracted Lightcurves*), también disponibles en MAST[19] y generadas por el equipo Dattilo et al. (2019) [6] en base a lo desarrollado por Vanderbug and Johnson (2014) [34], o de las curvas EVEREST (*EPIC Variability Extraction and Removal for Exoplanet Science Targets*) (Luger et al., 2016) [18], que han sido las utilizadas en el estudio, conjuntamente con las curvas originales.

K2SFF fue el primer método de corrección publicado sobre los datos de la misión K2.
Los procesos K2SFF decorrelacionan la fotometría de apertura K2 con el centroide de la posición de las imágenes estelares. Estos centroides se determinan en función del centro de luz o mediante un ajuste gaussiano a la función de dispersión de puntos estelares. Posteriormente el movimiento de los centroides se ajusta con un polinomio y se transforma en un único parámetro que relaciona el movimiento de la nave espacial con las variaciones de flujo, que luego se utiliza para eliminar la tendencia de los datos.
Es, en definitiva, una técnica basada en métodos numéricos para identificar y eliminar las correlaciones entre la posición estelar y las fluctuaciones de intensidad, haciéndose suposiciones sobre la naturaleza de las correlaciones entre el movimiento de la nave espacial y la variabilidad instrumental.
Adicionalmente se eliminan de los flujos de luz aquellos puntos que no presentan una calidad adecuada en función de los flags de calidad que aparecen en los ficheros FITS para cada punto. Se descartan así, por ejemplo, datos con una variedad de anomalías, incluidos los rayos cósmicos, los ajustes de orientación y los fallos del detector, incluyéndose también los puntos obtenidos mientras los propulsores de la nave estaban encendidos.

En el caso de las curvas EVEREST se utiliza una combinación de decorrelaciones a nivel de píxel (PLD - *pixel-level decorrelation*) para eliminar el error de orientación de la nave espacial. El principio básico de PLD es que las señales astrofísicas son las mismas en todos los píxeles correspondientes a una estrella, mientras que los efectos instrumentales, como las desviaciones en la orientación, no lo son.
Importante señalar lo indicado en la página del MAST-HLSP [22], pues indica que dado que EVEREST realiza ajustes por mínimos cuadrados para reducir el ruido en las curvas de luz K2, las características astrofísicas como tránsitos y eclipses a veces podrían ser un poco menos

profundas en el conjunto de datos sin tendencia. Para evitar esto, EVEREST enmascara automáticamente los valores atípicos antes de calcular los ajustes. Sin embargo, es probable que en este paso se pierdan los tránsitos de baja relación señal-ruido.
Esto podría justificar las diferencias observadas en el rendimiento observado entre el utilizar las curvas de luz K2SFF y las EVEREST.

A.3. Entrenamiento con datos K2 y EVEREST

Se incluyen aquí los resultados obtenidos de los modelos CUSTOM-ECG (4.4.4.2), FCN (4.4.2.1) y ResNet (4.4.2.2) en su aplicación a los conjuntos de datos K2 y EVEREST. Como se puede observar, todos presentan peores prestaciones que los mostrados en la memoria para K2SFF, motivo por el cual ya no se procedió posteriormente a la aplicación del resto de modelos.

Datos K2

modelo	learning_rate	batch_size	acc_train	acc_val	best_acc_val
CUSTOM	0,0001	16	97,18 %	65,16 %	70,96 %
		32	98,54 %	67,93 %	70,46 %
		64	92,28 %	68,18 %	70,46 %
	0,001	16	50,62 %	44,44 %	55,56 %
		32	50,88 %	44,44 %	55,56 %
		64	50,62 %	44,44 %	55,56 %
	0,01	16	49,33 %	50,00 %	55,56 %
		32	49,61 %	44,44 %	55,81 %
		64	48,93 %	44,44 %	55,56 %
FCN	0,0001	16	68,07 %	57,33 %	66,42 %
		32	68,44 %	48,48 %	66,42 %
		64	66,69 %	62,63 %	65,66 %
	0,001	16	80,13 %	54,80 %	67,17 %
		32	72,18 %	61,87 %	66,42 %
		64	66,15 %	55,81 %	67,68 %
	0,01	16	70,66 %	56,82 %	66,16 %
		32	78,10 %	58,84 %	67,43 %
		64	74,41 %	60,36 %	67,17 %
RESNET	0,0001	16	98,53 %	55,05 %	66,42 %
		32	99,63 %	50,51 %	65,15 %
		64	98,87 %	58,34 %	64,65 %
	0,001	16	96,70 %	56,57 %	66,67 %
		32	96,71 %	61,11 %	66,17 %
		64	99,95 %	58,34 %	65,91 %
	0,01	16	91,86 %	55,56 %	67,93 %
		32	95,89 %	58,34 %	65,41 %
		64	96,59 %	60,61 %	67,68 %

Tabla A.1: Entrenamiento con datos K2

Datos EVEREST

modelo	learning_rate	batch_size	acc_train	acc_val	best_acc_val
CUSTOM	0,0001	16	78,16 %	66,92 %	68,94 %
		32	78,70 %	60,86 %	68,94 %
		64	77,74 %	63,14 %	68,19 %
	0,001	16	50,62 %	44,44 %	55,56 %
		32	50,17 %	44,44 %	55,81 %
		64	50,62 %	44,44 %	55,56 %
	0,01	16	49,38 %	55,56 %	55,56 %
		32	50,62 %	44,44 %	55,56 %
		64	49,15 %	44,44 %	55,56 %
FCN	0,0001	16	63,67 %	54,55 %	66,16 %
		32	61,47 %	58,09 %	65,66 %
		64	63,19 %	54,80 %	66,16 %
	0,001	16	68,07 %	56,32 %	66,67 %
		32	66,97 %	59,60 %	65,15 %
		64	61,73 %	51,26 %	58,59 %
	0,01	16	66,55 %	55,56 %	65,16 %
		32	67,14 %	50,26 %	62,88 %
		64	67,02 %	50,76 %	61,11 %
RESNET	0,0001	16	90,62 %	58,34 %	72,48 %
		32	88,25 %	61,62 %	69,45 %
		64	81,85 %	58,84 %	68,19 %
	0,001	16	82,56 %	59,60 %	68,69 %
		32	85,48 %	54,80 %	67,68 %
		64	82,11 %	59,60 %	68,19 %
	0,01	16	85,35 %	59,09 %	67,93 %
		32	82,41 %	59,60 %	66,67 %
		64	83,51 %	63,39 %	67,18 %

Tabla A.2: Entrenamiento con datos EVEREST

Bibliografía

[1] David J Armstrong, Maximilian N Gunther, James McCormac, Alexis M S Smith, Daniel Bayliss, François Bouchy, Matthew R Burleigh, Sarah Casewell, Philipp Eigmuller, Edward Gillen, Michael R Goad, Simon T Hodgkin, James S Jenkins, Tom Louden, Lionel Metrailler, Don Pollacco, Katja Poppenhaeger, Didier Queloz, Liam Raynard, Heike Rauer, Stephane Udry, Simon R Walker, Christopher A Watson, Richard G West, and Peter J Wheatley. Automatic vetting of planet candidates from ground-based surveys: machine learning with NGTS. *Monthly Notices of the Royal Astronomical Society*, 478(3):4225–4237, 05 2018.

[2] Astropy. Librería Astropy. https://docs.astropy.org/en/stable/stats/index.html.

[3] Anthony Brunel, Johanna Pasquet, Jérôme Pasquet, Nancy Rodriguez, Frédéric Comby, Dominique Fouchez, and Marc Chaumont. A cnn adapted to time series for the classification of supernovae, 2019.

[4] Caltech. NASA Exoplanet Archive. https://exoplanetarchive.ipac.caltech.edu/docs/.

[5] Pattana Chintarungruangchai and Ing-Guey Jiang. Detecting exoplanet transits through machine-learning techniques with convolutional neural networks. *Publications of the Astronomical Society of the Pacific*, 131(1000):064502, May 2019.

[6] Anne Dattilo, Andrew Vanderburg, Christopher J. Shallue, Andrew W. Mayo, Perry Berlind, Allyson Bieryla, Michael L. Calkins, Gilbert A. Esquerdo, Mark E. Everett, Steve B. Howell, and et al. Identifying exoplanets with deep learning. ii. two new super-earths uncovered by a neural network in k2 data. *The Astronomical Journal*, 157(5):169, Apr 2019.

[7] Susan E. Thompson et al. Planetary candidates observed by kepler . VIII. a fully automated catalog with measured completeness and reliability based on data release 25. *The Astrophysical Journal Supplement Series*, 235(2):38, apr 2018.

[8] Trisha A. Hinners, Kevin Tat, and Rachel Thorp. Machine learning techniques for stellar light curve classification. *The Astronomical Journal*, 156(1):7, jun 2018.

[9] Hippke. Librería Wotan. https://wotan.readthedocs.io/en/latest/index.html.

[10] Michael Hippke, Trevor J. David, Gijs D. Mulders, and René Heller. Wōtan: Comprehensive time-series detrending in python. *The Astronomical Journal*, 158(4):143, Sep 2019.

[11] Hippke, Michael and Heller, René. Optimized transit detection algorithm to search for periodic transits of small planets. *A&A*, 623:A39, 2019.

[12] Chaur-Heh Hsieh, Yan-Shuo Li, Bor-Jiunn Hwang, and Ching-Hua Hsiao. Detection of atrial fibrillation using 1d convolutional neural network. *Sensors*, 20(7), 2020.

[13] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, Mar 2019.

[14] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, Sep 2020.

[15] Jara-Maldonado, Alarcon-Aquino, and Rosas-Romero et al. Transiting exoplanet discovery using machine learning techniques: A survey. *Earth Sci Inform*, 13:573–600, sep 2020.

[16] Kovács, G., Zucker, S., and Mazeh, T. A box-fitting algorithm in the search for periodic transits. *A&A*, 391(1):369–377, 2002.

[17] Lightkurve. Librería Lightkurve. https://docs.lightkurve.org/.

[18] Rodrigo Luger, Eric Agol, Ethan Kruse, Rory Barnes, Andrew Becker, Daniel Foreman-Mackey, and Drake Deming. Everest: Pixel level decorrelation ofk2 light curves. *The Astronomical Journal*, 152(4):100, Oct 2016.

[19] MAST. Mikulsky Archive for Space Telescopes. https://archive.stsci.edu/k2/.

[20] MIT. TESS - Transiting Exoplanet Survey Satellite. https://tess.mit.edu//.

[21] NASA. 10 Things: All About TRAPPIST-1 . https://solarsystem.nasa.gov/news/335/10-things-all-about-trappist-1/.

[22] NASA. K2 - High Level Science Products. https://archive.stsci.edu/k2/hlsps.html.

[23] NASA. Kepler's Second Light: How K2 Will Work. https://www.nasa.gov/kepler/keplers-second-light-how-k2-will-work.

[24] NASA. NASA Exoplanet Archive - K2. https://exoplanetarchive.ipac.caltech.edu/docs/program_interfaces.html#k2.

[25] NASA. TESS - Primary science. https://heasarc.gsfc.nasa.gov/docs/tess/primary-science.html.

[26] Brett Naul, Joshua S. Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2):151–155, Nov 2017.

[27] Isadora Nun, Pavlos Protopapas, Brandon Sim, Ming Zhu, Rahul Dave, Nicolas Castro, and Karim Pichara. Fats: Feature analysis for time series, 2015.

[28] H. P. Osborn, M. Ansdell, Y. Ioannou, M. Sasdelli, D. Angerhausen, D. Caldwell, J. M. Jenkins, C. Räissi, and J. C. Smith. Rapid classification of tess planet candidates with convolutional neural networks. *Astronomy & Astrophysics*, 633:A53, Jan 2020.

[29] Kyle A. Pearson, Leon Palafox, and Caitlin A. Griffith. Searching for exoplanets using artificial intelligence. *Monthly Notices of the Royal Astronomical Society*, 474(1):478–491, Oct 2017.

[30] @ridlo. BLS (box-fitting least squares) algorithm. https://es.slideshare.net/ridlocepheid/boxfitting-algorithm-presentation.

[31] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.

[32] Christopher J. Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94, jan 2018.

[33] Susan E. Thompson, Fergal Mullally, Jeff Coughlin, Jessie L. Christiansen, Christopher E. Henze, Michael R. Haas, and Christopher J. Burke. A MACHINE LEARNING TECHNIQUE TO IDENTIFY TRANSIT SHAPED SIGNALS. *The Astrophysical Journal*, 812(1):46, oct 2015.

[34] Andrew Vanderburg and John Asher Johnson. A technique for extracting highly precise photometry for the two-wheeledkeplermission. *Publications of the Astronomical Society of the Pacific*, 126(944):948–958, Oct 2014.

[35] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1578–1585, 2017.