Vol. 1. Núm. 2E (2022)
Número especial: VII Congreso de Astronomía y Astrofísica

Ciencia en Desarrollo
Revista de la Facultad de Ciencias
Vicerrectoría en Investigación y Extensión - VIE
Centro de Investigaciones y Extensión de la Facultad de Ciencias - CIEC

https://doi.org/10.19053/01217488.v1.n2E.2022.15244

Optimización de un sistema difuso para la detección automática de tránsitos planetarios en curvas de luz de estrellas individuales

Optimized Fuzzy System for automatic detection of planetary transits in light curves of individual stars

Christian Leonardo Muñoz Cárdenas[1], David Santiago Gómez Lozano[1], Cristian Marquez[2], Edilberto Suarez Torres[3], Camilo Delgado Correal[3]
[1]Facultad de ingeniería, Universidad Distrital Francisco José de Caldas
[2]Elemy Care
[3]Observatorio Astronómico LatitUD, Universidad Distrital Francisco José de Caldas
[*]Correo clmunozc@correo.udistrital.edu.co

Resumen

El método de tránsito es un método efectivo para identificar planetas extrasolares, que se basa en la disminución poco profunda que provoca un planeta en el brillo aparente de su estrella anfitriona. Sin embargo, los eventos de tránsito están muy cerca del límite de la sensibilidad de detección de los telescopios y se necesitan al menos tres (3) eventos de tránsito producidos por el mismo planeta para confirmar el descubrimiento de éste, lo que hace que se requieran observaciones por tiempos prolongados de una estrella para detectar planetas extrasolares que puedan estar orbitandola, lo que resulta en grandes cantidades de datos que deben ser analizados. En este trabajo se desarrolló una nueva tuberia de software (pipeline) para la detección autónoma de rastros de tránsitos planetarios analizando características extraídas de curvas de luz estelares utilizando un clasificador de lógica difusa, evadiendo la tarea de buscar tránsitos en cada sección de las curvas de luz. Para el desarrollo de esta tuberia de software se implementó la metodologia llamada: Knowledge Discovery in Databases (KDD) la cual presenta una forma de extraer conocimiento de grandes conjuntos de datos.

***Palabras clave*:** Curvas de luz, Inteligencia computacional, Lógica Difusa, Método de tránsito, Optimización global, Planetas extrasolares.

Abstract

The transit method is an effective way to find extrasolar planets. This method is based on the shallow decrease that a planet causes in its host star's apparent brightness: when the planet passes through our line of sight, it affects the brightness we receive from the star with our telescopes. However, these transit events are very close to the telescopes' detection sensitivity limit. To confirm a planet observation, it takes at least three (3) transit events, making long-time observations of a star necessary to detect extrasolar planets that may orbit it, which results in large amounts of data to be analized. In this work we did a new software pipeline

for autonomous detection of transit traces by analyzing extracted features from stellar light curves using a fuzzy logic classifier, avoiding the task of searching for transit events in each section of the light curves. During the development process of the software pipeline, the Knowledge Discovery in Databases (KDD) methodology is implemented, which presents a way to extract knowledge from large datasets.

***Key words*:** Computational intelligence, Extrasolar planets, Fuzzy logic, Global Optimization, Light curves, Transit method.

1. Introducción

La búsqueda de planetas que orbitan estrellas diferentes al Sol empezó con el descubrimiento del primer sistema exoplanetario en 1992 por Wolszczan mientras estudiaba el pulsar PSR B1257+12, con el pasar del tiempo, el campo se fue ampliando, por lo que se realizaron misiones para el lanzamiento de telescópios espaciales dedicados a la busqueda de estos cuerpos celestes [1], una de las más importantes fue la misión Kepler, cuyo objetivo era determinar la frecuencia de planetas similares a la Tierra, obteniendo datos fotométricos de estrellas en busca de disminuciones en su brillo que puedan indicar la presencia de un planeta, este método se conoce como «el método de tránsito» [2].

Al día de hoy (14 de enero de 2023), se ha confirmado el descubrimiento de 5,241 exoplanetas, de los cuales 3,914 (75.2 %) han sido descubiertos por el método de tránsito [3], lo que evidencia que este método es el de mayor seguimiento y utilización por parte de la comunidad cientifica [4]. Existen una gran cantidad de misiones dedicadas a la busqueda de exoplanetas utilizando este método (COROT, TESS, SuperWASP) [1], lo que provoca que el tamaño de las extensas bases de datos donde se almacenan curvas de luz estelares se incrementen progresivamente, provocando asimismo la necesidad crear nuevos sistemas automatizados para la detección efectiva de estos cuerpos que permitan con ello hacer el análisis automático de una gran cantidad de datos.

Para el análisis de curvas de luz de estrellas, se utiliza comúnmente el algoritmo BLS (Box-fitting Least Squares) con el fin de hacer más visible perturbaciones periódicas en la serie de tiempo, sin embargo, este algoritmo trabaja únicamente con tránsitos periodicos [5], dejando a un lado las observaciones que pudieran presentar un único tránsito planetario. Mislis et al. [6] presentó un nuevo enfoque para el análisis fotométrico de curvas de luz que permite la detección de señales de eventos de interés astronómicos utilizando un algoritmo de detección ciega de señales, en donde se busca determinar la factibilidad del uso de este tipo de detección de señales para encontrar firmas de tránsitos planetarios, utilizando datos de curvas de luz estelares obtenidos por el telescopio espacial Kepler (o similares) y algoritmos de clasificación automática.

Este trabajo se enfoca en el desarrollo de una nueva solución automátizada a la necesidad de analizar extensas bases de datos fotométricos en búsqueda de tránsitos planetarios, no solo porque el método de tránsito sea el más popular debido a su eficiencia, lo que provoca que haya grandes cantidades de datos disponibles para su estudio, sino también porque se tiene en cuenta la alta tasa de falsas detecciones que implica el método, pues los transitos planetarios son fácilmente confundibles con otros fenómenos, como estrellas binarias eclipsantes [7], signos de alta actividad estelar, entre otros.

Asimismo, se busca contribuir a la investigación de metodologías alternativas a las derivadas del algoritmo BLS, extendiendo el trabajo de Mislis et al. [6], en el cual se desarrolló un método estadístico para la detección de eventos de interés astronómicos en curvas de luz, investigando la efectividad de su método para la detección exclusiva de tránsitos planetarios. Es por lo anterior que en este artículo se propone un nuevo desarrollo de un sistema de software de detección automática de curvas de luz que presenten rastros de tránsitos planetarios utilizando el método estadístico de detección de señales presentado en Mislis et al. [6] y clasificadores automáticos de inteligencia artificial optimizados para estudiar la factibilidad de su uso en escenarios reales de busqueda de exoplanetas.

1.1. Antecedentes

En la literatura relacionada a la utilización de técnicas de inteligencia artificial en la búsqueda de exoplanetas se hace uso de gran variedad de técnicas de la ciencia de datos para aportar a la automatización del análisis de curvas de luz, como métodos de aprendizaje profundo con redes neuronales convolucionales, maquinas de soporte vectorial, entre otros [9]. Por ejemplo, Shallue y Vanderburg [10] usa la clasificación de fragmentos de curvas de luz utilizando una red neuronal convolucional y el algoritmo BLS, en donde el sistema evalúa eventos de interés astronómico, obteniendo para cada evento dos curvas de luz con distinta cantidad de puntos de datos, una curva de luz de «vista global», que contempla 2,001 puntos, y una curva de «vista local», que contempla 201 puntos, del evento periódico. Posteriormente, ambas vistas se introducen a la arquitectura de red neuronal convolucional para su clasificación, obteniendo una efectividad de clasificación de 96 %. Este método fue capaz de hallar dos exoplanetas para el momento desconocidos, Kepler-80g y Kepler-90i. Este trabajo fue seguido por la investigación de Ansdell et al. [11] que se basa en la introducción de conocimiento científico del dominio por parte de los investigadores, utilizando las series de tiempo de la posición del pixel centro de luz (centroide), obtenidas de las mismas imágenes Target Pixel File (TPF) de las que se obtuvieron las curvas de luz, como variable de entrada para mejorar la clasificación de la arquitectura de red neuronal diseñada por Shallue y Vanderburg, mejorando su rendimiento a un 97 %.

El trabajo desarrollado por Mislis et al [6] brinda un enfoque distinto al desarrollo de sistemas de software para el análisis de estos datos, utilizando para la detección fotométrica de eventos de ínteres astronómicos el cálculo de propiedades estadísticas de las curvas de luz, en donde se calculó 4 propiedades de las curvas de luz: curtosis, la cual se refiere al achatamiento o elevación de una distribución con respecto a la distribución normal; sesgo, el índice de asimetría de una distribución con respecto a su media; autocorrelación integral, la suma de los valores de autocorrelación para todos los posibles valores de retraso; y entropía de información modificada, medida de incertidumbre para cada punto de datos de una curva de luz. Se utilizaron los valores de estas características para clasificar las curvas de luz en cinco (5) categorías distintas: Constante, denota una estrella con brillo constante; tránsito, una estrella con uno o más tránsitos planetarios presentes; variable, una estrella con brillo variable; binaria eclipsante, una estrella que orbita y asímismo es orbitada por otra estrella; microlente gravitacional, una curva de luz que presenta un efecto de microlente gravitacional producido por un cuerpo en el rango de visión entre la estrella y el planeta Tierra. Usando el algoritmo de Bosques Aleatorios, el cual refleja las propiedades intrinsecas de las distintas categorías consideradas de un conjunto de datos, también se analizó la significancia de cada una de estas características, obteniendo una efectividad de clasificación mayor al 90 % para todas las clases y especificamente para la clase de curvas de luz con tránsitos, se obtuvo una efectividad del 92 %.

En el trabajo de Ofman et al. [12] se utilizó algoritmos de inteligencia artificial semi-supervisados y no supervisados, cuyo «entrenamiento» no depende directamente de la clasificación de los ejemplares sino que es más orientado al agrupamiento de ejemplares similares para la posterior detección de datos anómalos, implementados originalmente para la detección de delitos financieros, problemas de seguridad cibernética e Internet de las cosas (IoT) en la industria Fintech por la compañía ThetaRay y se aplicaron para la búsqueda de EPC (ExoPlanet Candidates) en curvas de luz. con el sistema de ThetaRay, utilizando las fortalezas de sus algoritmos se pudo identificar varias decenas de candidatos a exoplanetas esperando a ser confirmados.

Asif et al. [13], en su trabajo desarrollado para la Conferencia Internacional de Sistemas Inteligentes (IS) de 2018, investigaron el uso de un sistema adaptativo de inferencia neuro-difuso (ANFIS) para la clasificación autónoma de curvas de luz en las que se presentan tránsitos, esto lo hacia extrayendo características de la curva de luz como la media, la desviación estándar y midiendo la deformación dinámica de tiempo entre cada par de curvas de luz. Introduciendo estas variables de entrada en el sistema neuro-difuso, se alcanzó un 81 % de efectividad, pero, a diferencia de los demás trabajos, el sistema

de clasificación utilizado permite ver la significancia de cada variable de entrada, pudiendo visualizar la «concepción» del sistema con respecto a qué considera una curva de luz con tránsitos presentes y qué no, en este caso, pudo evidenciarse que la poca dimensionalidad de la entrada fue la que impidió una mejor eficiencia de clasificación.

Otro ejemplo de algoritmo para la clasificación de estos posibles exoplanetas lo podemos encontrar en el trabajo de Valizadegan et al. [14] donde se utiliza el aprendizaje automático para analizar estos datos en busca de nuevos exoplanetas. A diferencia de los trabajos de aprendizaje automático existentes, ExoMiner, el clasificador de aprendizaje profundo propuesto en este trabajo, imita cómo los expertos de dominio examinan las pruebas de diagnóstico para examinar una señal de tránsito. ExoMiner es un clasificador comprensible (es decir, proporciona una retroalimentación de sus resultados a los expertos), robusto y altamente preciso, pues permitió validar 301 nuevos exoplanetas del MAST Kepler Archive, además, el modelo es lo suficientemente general como para aplicarse en misiones actuales, como lo es la misión TESS.

1.2. Método de Transito

El método de tránsito consiste en observar por periodos prolongados una estrella con el objetivo de detectar sutiles caídas en su brillo aparente provocadas por algún planeta que orbita a través del ángulo de visión de la estrella desde el planeta Tierra. Este método proporciona información precisa sobre la órbita y el tamaño del planeta: la duración del tránsito depende de la distancia del planeta a la estrella y de la masa estelar, puesto que la masa y el tamaño de la estrella pueden ser determinados mediante otras observaciones, se puede determinar la longitud de la órbita del planeta a partir de la duración del tránsito; el tamaño del planeta puede ser calculado midiendo la disminución máxima del brillo durante el tránsito, debido a que cuanto más grande es el planeta, más marcada es la reducción del brillo aparente de la estrella [15].

Sin embargo, el método del tránsito sólo puede revelar los planetas que pasan exactamente entre su estrella y la Tierra, y esta limitado por la resolución máxima del fotómetro, es decir, la capacidad de este de detectar caídas muy pequeñas en el brillo, por lo que los planetas deben ser lo suficientemente grandes para causar una reducción de brillo detectable, esto hace que se puedan presentar complicaciones para confirmar exoplanetas por medio de este método, a veces requiriendo una confirmación de la existencia del posible planeta por medio de otros métodos, como el método de velocidad radial. Además, el paso de un planeta no es el único fenómeno que puede provocar caídas en el brillo aparente de la estrella, pues puede ser una binaria eclipsante, producto de la variabilidad de la estrella, fallas técnicas de los aparatos utilizados para la observación, entre otros [15].

1.3. Metodología KDD (Knowledge Discovery in Databases)

El descubrimiento de conocimiento se refiere a la extracción de información implícita, previamente desconocida y potencialmente útil a partir de datos. Dados un conjunto de hechos F, un lenguaje L y una medida de certeza c, se define un patrón como una declaración S en L que describe relaciones entre un subconjunto F_S de F con cierta certeza c, de modo que S sea más simple que una enumeración de los hechos de F_s. Un patrón que sea interesante y lo suficientemente certero en términos definidos por el usuario se denomina «conocimiento» [16].

La metodología KDD plantea una serie iterativa de pasos para la extracción de información útil de bases de datos de gran tamaño, estos pasos pueden ser realizados nuevamente en caso de requerirlo (Ver Figura, estos son [17]:

- Identificación de la meta: Este es el primer paso del proceso KDD y tiene que ver con el resultado esperado con respecto al usuario final del conocimiento obtenido del proceso,
- Selección de datos: Luego de tener claro lo que queremos de la información que se tiene disponible, se procede a seleccionar los datos que son necesarios para realizar la extracción del conocimiento.
- Preprocesamiento: Los datos seleccionados pasan a ser limpiados y preprocesados para faci-

litar su interpretación y el descubrimiento de patrones para los modelos de minería de datos a utilizar.

- Transformación: En este paso del proceso se obtiene un nuevo conjunto de datos a partir de los datos preprocesados anteriormente, utilizando técnicas de reducción de dimensionalidad, ingeniería de características, entre otras.
- Minería de datos: Durante este paso del proceso KDD, se deben seleccionar las tareas y algoritmos de minería de datos apropiados para lograr la meta esperada con el conjunto de datos obtenido y ponerlos a prueba.
- Evaluación de resultados: Luego de la ejecución de los algoritmos de minería de datos seleccionados, procedemos a evaluar su desempeño y a realizar la identificación del conocimiento que pudo obtenerse de la minería de datos.

1.4. Redes Neuronales

Las redes neuronales son un modelo inspirado en el funcionamiento de los sistemas vivos, la biologia, en especial del cerebro humano; Estas redes neuronales forman un conjunto de nodos conocidos como neuronas artificiales que están conectadas y transmiten señales entre sí con el objetivo de «aprender» a realizar tareas complejas que no podrían ser realizadas mediante la clásica programación basada en reglas [10].

1.5. Lógica difusa

La lógica difusa [18] es una extensión de la lógica booleana tradicional basada en la teoría matemática de conjuntos difusos, esta introduce la noción gradual en la verificación de una condición, permitiendo que el resultado de una condición pueda estar en un valor entre falso y verdadero. Esta lógica provee una muy valiosa flexibilidad para el razonamiento, pues permite tener en cuenta incertidumbres.

Uno de los beneficios más importantes de la lógica difusa es que las reglas están formuladas en lenguaje natural [18], por ejemplo, se pueden definir ciertas reglas para calcular la cantidad de propina a pagar por un servicio en un restaurante según la calidad del servicio y la comida:

1. Si $Servicio = Malo$ o $Comida = Malo$ entonces $Propina = Baja$
2. Si $Servicio = Bueno$ entonces $Propina = Media$
3. Si $Servicio = Excelente$ o $Comida = Rica$ entonces $Propina = Alta$

Teniendo en cuenta estas reglas, es posible definir las variables lingüisticas de entrada (*Servicio* y *Comida*) y la variable lingüisticas de salida (*Propina*) con el objetivo de definir conjuntos difusos para atribuir un valor cuantificable a los conceptos inexactos de un mal, buen y excelente servicio; de una mala y rica comida; y de una baja, media y alta propina. Una variable lingüistica corresponde a la tupla de tres (3) elementos: V, nombre de la variable; X, rango de la variable; y T_V, conjuntos difusos de cada valor posible de la variable [18].

Ya con estas reglas y variables lingúisticas es posible crear un modelo de lógica difusa que nos permita implicar qué cantidad de propina proporcionar según los datos de calidad del servicio y comida que se le introduzcan [18], por ejemplo, supóngase una calidad de servicio de 7.83 y una calidad de la comida de 7.32, teniendo en cuenta únicamente la regla «si $Servicio = Excelente$ o $Comida = Rica$ entonces $Propina = Alta$», la proposición de que la propina sea alta tiene mayor grado de verdad que las proposiciones de que la propina sea media o baja.

Al tener en cuenta todas las reglas, se construye una matriz de decisión para la implicación difusa[18] de toda la lógica construida anteriormente, los grados de verdad para este ejemplo se calculan así: la regla 1 no aporta a la solución, pues ni la calidad de la comida ni la calidad del servicio fueron en algún grado de certeza «malas»; por la regla 2, ya que el servicio es, en cierto grado de certeza, «bueno», esta aporta cierto grado de verdad a la proposición de que la cantidad de propina debe ser «media»; y la regla 3, como se dijo en la explicación anterior, aporta un grado de verdad alto, más que la regla 2, debido a que la calidad de la comida es en un grado alto de certeza, «rica» y la calidad del servicio es en un grado alto de certeza «excelente». Esto da como

resultado que la propina debe ser «alta» en un grado alto de certeza.

Este modelo puede ser construido como un agente basado en reglas que utilice los conjuntos difusos como base de conocimiento para que, a partir de los datos de calidad del servicio y la comida que le sean brindados, pueda sugerir qué cantidad de propina pagar a cambio siguiendo las reglas definidas. Estos tipos de sistema son denominados «sistemas difusos»[18].

1.6. Detección de bordes

En visión computacional, para poder identificar objetos individuales en una imagen es necesario poder identificar las discontinuidades repentinas entre pixeles provocadas por la presencia de los distintos objetos en la imagen, estas discontinuidades se denominan «bordes», estos son importantes debido a que puede deducirse de estos bordes gran cantidad de información semántica y geométrica de los objetos en la imagen, siendo al mismo tiempo una representación más compacta que la imagen en bruto [19].

Se pueden identificar estos bordes haciendo uso de la derivada dimensional de una imagen, la cual e denomina 'gradiente' y es definida como la razón de cambio de luminosidad de los pixeles que componen la imagen, buscando cambios drásticos entre pixeles cercanos de la imagen, para calcular el gradiente existen varios operadores o 'kernels' por los que se procesa la imagen mediante un proceso de fitrado de imágenes [19], el en cual se realiza una convolución en dos (2) dimensiones de dos imágenes, siendo una la imagen a procesar y la otra, el 'kernel' que actua como filtro.

Luego de realizar las convoluciones con los 'kernels' para la detección de bordes y de operar los resultados de estas, se obtiene una imagen donde solo pueden visualizarse los bordes [19], es decir, los cambios repentinos en el color de la imagen, permitiendo obtener una silueta de los objetos en la imagen original.

1.7. Optimización por Enjambre de Partículas

Optimización por Enjambre de partículas o Particle Swarm Optimization (PSO) es uno de los métodos de optimización global más efectivos para la resolución de problemas no lineales y complejos de alta dimensión. Dado que el rendimiento de PSO depende en gran medida de la elección de su entorno (por ejemplo, inercia, factores cognitivos y sociales, velocidad mínima y máxima)[20].

Se inspira en el comportamiento de los enjambres de insectos en la naturaleza. En concreto, podemos pensar en un enjambre de abejas, ya que éstas a la hora de buscar polen buscan la región del espacio en la que existe más densidad de flores, porque la probabilidad de que haya polen es mayor. La misma idea fue trasladada al campo de la computación en forma de algoritmo y se emplea en la actualidad en la optimización de distintos tipos de sistemas [21].

2. Metodología

Para el desarrollo de este trabajo se utilizó la metodología KDD (Knowledge Discovery in Databases) con el objetivo de descubrir patrones que permitan la identificación de tránsitos planetarios a partir de propiedades estadísticas calculadas de curvas de luz analizadas en estudios previos haciendo uso de los sistemas difusos para obtener un nuevo modelo basado en reglas formuladas en lenguaje natural.

2.1. Selección de datos

Los datos seleccionados consisten en dos conjuntos diferentes de curvas de luz, las cuales son gráficas de la variación del brillo aparente de estrellas individuales con respecto al tiempo, cada una de estas curvas de luz está etiquetada con un 0, en caso de que no presente ningún tránsito planetario confirmado (caso negativo), y con un 1, en caso contrario (caso positivo).

El primer conjunto de datos corresponde a curvas de luz que fueron capturados por el telescópio espacial Kepler, en su mayoría, durante la Campaña 3 de la misión y el resto son curvas de luz con tránsitos planetarios confirmados para disminuir el desequilibrio entre las categorías, estas curvas de luz fueron preprocesadas para hacer que sean gráficas contínuas

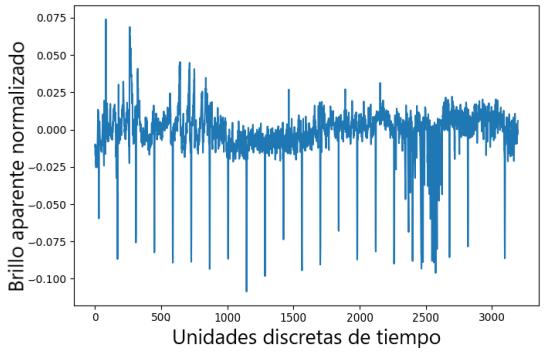


Figura 1. Curva de luz normalizada del conjunto de datos obtenido desde Kaggle[22].

y de la misma longitud. Este conjunto de datos fue obtenido a través de la plataforma Kaggle[22], un portal comunitario en línea de ciencia de datos y Machine Learning.

Este conjunto de datos corresponde a curvas de luz de estrellas individuales compuestas de 3,197 puntos de datos cada una, el conjunto está dividido en dos subconjuntos de entrenamiento y prueba: el subconjunto de entrenamiento, que contiene 5,087 curvas de luz, de estas solo 37 corresponden a casos positivos y 5,050 a casos negativos (Ver un ejemplo de estas curvas de luz en Figura 1); y el subconjunto de prueba, que contiene 570 curvas de luz de los cuales solo 5 son casos positivos y 565 casos negativos. Se evidencia el gran desequlibro de categorías del conjunto de datos, pues solo el 0.7 % de los casos presentes en el conjunto son positivos.

La selección del uso de este conjunto de datos se basa en 2 principales razones: la antigüedad de estos datos hacen que sea menos probable que existan falsos positivos, es decir, casos mal etiquetados en el conjunto de datos; y la limpieza realizada previamente en el conjunto de datos facilita el proceso de pre-procesamiento a realizar en este desarrollo.

Sin embargo, debido a dificultades presentadas en el proceso de análisis con el conjunto de datos considerado inicialmente (Ver 3. Resultados y discusión), se optó por reemplazarlo por el conjunto de datos utilizado en el trabajo de Shallue y Vanderburg[10] para la creación de un nuevo modelo de red neuronal convolucional para la detección de exoplanetas sobre datos del telescópio Kepler, estos datos fueron extraídos del catalogo de candidatos planetarios Autovetter para Q1-Q17 DR24 del proyecto Kepler de la NASA[23]. Son curvas de luz preprocesadas para eliminar variaciones de luz producto de fenómenos no relacionados con tránsitos planetarios, como la variabilidad intrinseca de las estrellas, y que han sido 'dobladas' por el periodo de cada posible exoplaneta con el objetivo de superponer los tránsitos similares entre sí, obteniendo una gráfica con la información de estos en un único tránsito que es más fácil de generalizar para un sistema de inteligencia computacional. Este conjunto de datos, por cada ejemplo, posee dos curvas de luz con distinta cantidad de puntos, una curva de luz de «vista global», la cual se compone de 2,001 puntos de datos, y una curva de luz de «vista local», compuesta por 201 puntos de datos, para el desarrollo experimental de este trabajo, se utilizaron únicamente las curvas de luz de «vista local» de cada ejemplo, debido a que estas contienen menos puntos de datos y son más rápidas de procesar para la extracción de características (Ver ejemplos de curvas de luz de «vista global» y «vista local» en Figura 2).

Este nuevo conjunto está dividido en tres subconjuntos: el conjunto de entrenamiento, que contiene 12,589 curvas de luz, de las cuales 2,880 corresponden a casos positivos y los 9,709 restantes a casos negativos; el conjunto de validación y el conjunto de pruebas, ambos con 1,574 curvas de luz, de los cuales 360 son casos positivos y 1,214 casos negativos. Se puede evidenciar un desequilibrio menos pronunciado con respecto al primer conjunto de datos considerado, pero que igualmente puede afectar al análisis de resultados.

2.2. Preprocesamiento de datos

Como se resaltó en la etapa anterior de la metodología, existe un desequilibrio importante en ambos conjuntos de datos con respecto a los casos positivos y negativos del conjunto, por lo que para trabajar con esto se utilizó la libreria Imbalanced-learn [24] para aplicar el algoritmo de sobremuestreo SMOTE (Synthetic Minority Oversampling Technique, Técnica de sobremuestreo de minorías sintéticas) de manera que la interpretación de los resultados de clasificación no se vea afectada por el desequlibrio.

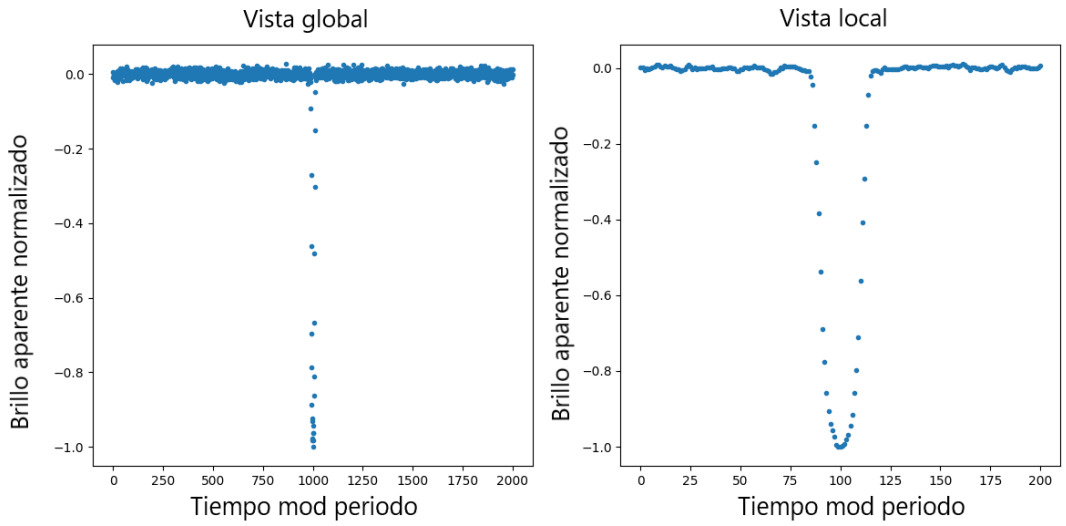


Figura 2. Curvas de luz normalizadas y 'dobladas' de «vista global» (izquierda) y «vista local» (derecha) del conjunto de datos obtenido desde el trabajo de Shallue y Vanderburg[10].

Esta técnica consiste en generar casos de la clase menos numerosa a partir de los casos existentes, de modo que las clases coincidan en el número de casos, consiguiendo un conjunto de datos totalmente balanceado.

También se utilizó la librería Scikit-learn [25] para normalizar los datos, esto con el objetivo de tener en cuenta las escalas relativas de cada curva de luz.

2.3. Ingeniería de características

2.3.1. Selección de variables

Para seleccionar las características a extraer de las curvas de luz se tuvo en cuenta las reglas propuestas de Mislis et al.[6] para la selección de características para su sistema automático de detección ciega, las reglas son descritas a continuación:

- Las características deben ser lo más generales posibles, de modo que el sistema pueda calcularlas para cualquier curva de luz.
- El cálculo de estas características debe ser lo más rápido posible para que el sistema sea una solución viable en escenarios en tiempo real y con grandes cantidades de casos a analizar.
- Las características deben tener muy poca correlación para obtener la mayor cantidad de información, lo que no ocurre con características muy correlacionadas entre sí.

Siguiendo estas reglas y usando como base el trabajo realizado por Mislis et al.[6], se seleccionaron 5 características principales a extraer de las curvas de luz recibidas inicialmente.

- Curtosis (C): Calcula el grado de «achatamiento» de una distribución, es definida por

$$C = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^4}{\sigma^4}$$

  Donde $\bar{x}$ es la media, σ es la desviación estandar y n es el número de elementos en la distribución, en este caso, el número de puntos de cada curva de luz.

- Sesgo (S): Calcula la asimetría de una distribución, es definida por

$$S = \frac{1}{n} \sum_{i=1}^{n} \frac{(x_i - \bar{x})^3}{\sigma^3}$$

  Donde $\bar{x}$ es la media, σ es la desviación estandar y n es el número de elementos en la distribución, en este caso, el número de puntos de cada curva de luz.

- Autocorrelación integral (Ai): Es la suma de todos los valores de autocorrelación para todos los valores de retraso, puede brindar información acerca de patrones periódicos en las curvas de luz, es descrita por

$$C = \left| \sum_{\tau=1}^{n} \left(\frac{1}{(n-\tau) rms^2} \sum_{i=1}^{n=\tau} (x_i - \bar{x})(x_{i+\tau} - \bar{x}) \right) \right|$$

Donde τ es el retraso, *rms* es la raíz cuadrática media.

- Entropía espectral (E): Es una forma normalizada de la Entropia de Shannon, esta mide la complejidad espectral de una señal [26], es definida por

$$E = \sum_{f} p_f \log \left(\frac{1}{p_f} \right)$$

Donde f es la frecuencia de cada elemento de la distribución y p_f es el poder de cada frecuencia, esta variable solo se utilizó con el primer conjunto de datos.

- Longitud Wavelet (Wl): Representa la longitud de onda acumulada de una señal con respecto al tiempo [27], es descrita por

$$WL = \sum_{i=1}^{n} |x_{i+1} - x_i|$$

Donde x_i es el i-ésimo punto de la señal y n es la cantidad de puntos de la señal.

Las siguientes variables se agregaron despues del reemplazo de conjunto de datos considerado durante la selección por el conjunto de curvas de luz obtenido del trabajo de Shallue y Vanderburg [10].

- Máximo y mínimo (M, m): Representan el punto máximo y mínimo de una serie de tiempo, son descritas respectivamente por

$$M = \max X,$$

$$m = \min X$$

Donde X es conjunto de todos los puntos de datos de la serie de tiempo, esta variable solo se utilizó con el segundo conjunto de datos.

- Coeficientes de mínimos cuadrados (a, b): Representan los coeficientes de la recta resultante de la realización de una regresión lineal a una serie de tiempo, son descritas respectivamente por

$$b = \frac{n \sum_{i=1}^{n} t_i x_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} t_i\right)}{n \sum_{i=1}^{n} (t_i^2) - \left(\sum_{i=1}^{n} t_i\right)^2}$$

$$a = \frac{\sum_{i=1}^{n} x_i}{n} - b \frac{\sum_{i=1}^{n} t_i}{n}$$

Donde x_i es el i-ésimo punto de la señal, t_i es el i-ésimo punto del tiempo y n es la cantidad de puntos de la señal, esta variable solo se utilizó con el segundo conjunto de datos.

2.3.2. Extracción de características

El cálculo de las características seleccionadas en las curvas de luz se basó principalmente en el uso de librerías especializadas para este tipo de cálculos: Para calcular la curtosis (C) y el sesgo (S) se utilizó el módulo de estadística (stats) de SciPy [28]; se usó la libreria NumPy[29] para calcular los máximos y mínimos y las autocorrelaciones a sumar de las curvas de luz cuyo resultado corresponde a la Autocorrelación Integral; la entropía espectral fue calculada utilizando la librería PyEEG [30]; y el cálculo de la longitud Wavelet y los coeficientes de mínimos cuadrados fueron codificados directamente.

Antes de computar las características de las curvas de luz, se generaron otras series de tiempo a partir de estas: el gradiente de la curva de luz, basado en técnicas para la detección de bordes en visión computacional (Ver Figura 3 y Figura 4); y la transformada de Fourier de la curva de luz (Ver Figura 5 y Figura 6), esto, con el objetivo de brindar al problema más dimensionalidad que permita una generalización mayor y, por ende, una mayor efectividad de clasificación.

Para el cálculo del gradiente de las curvas de luz se aplica un operador Sobel, utilizado para la detección de bordes en procesamiento de imágenes, que permita segmentar la curva de luz a partir de sus caidas de brillo aparente; para el cálculo de la transformada de Fourier de la curva de luz se utilizó el módulo Fast Fourier transform (FFT) de la librería SciPy[28].

Luego, se computan y agrupan las características por cada serie de tiempo disponible, aplicando

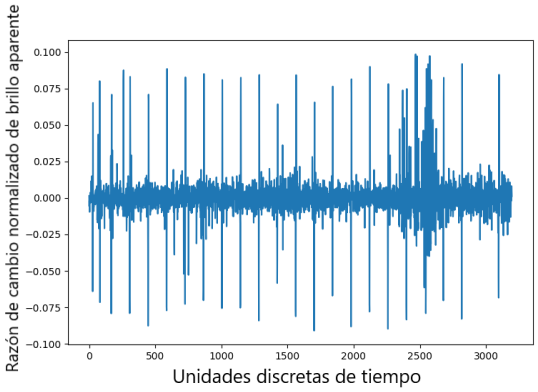


Figura 3. Gradiente de una curva de luz del conjunto de datos obtenido desde Kaggle[22].

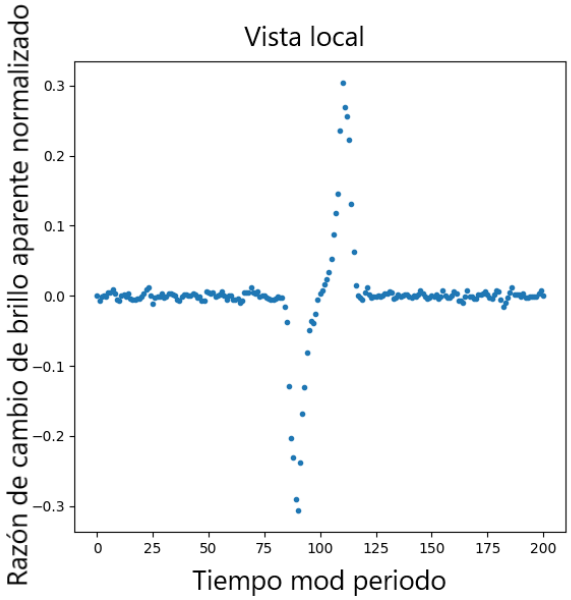


Figura 4. Gradiente de una curva de luz de «vista local» del conjunto de datos obtenido desde el trabajo de Shallue y Vanderburg[10].

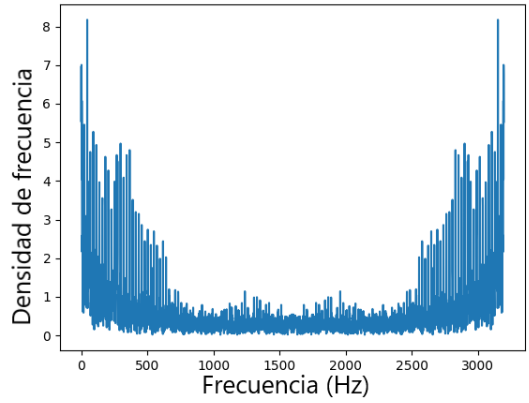


Figura 5. Transformada de Fourier de una curva de luz del conjunto de datos obtenido desde Kaggle[22].

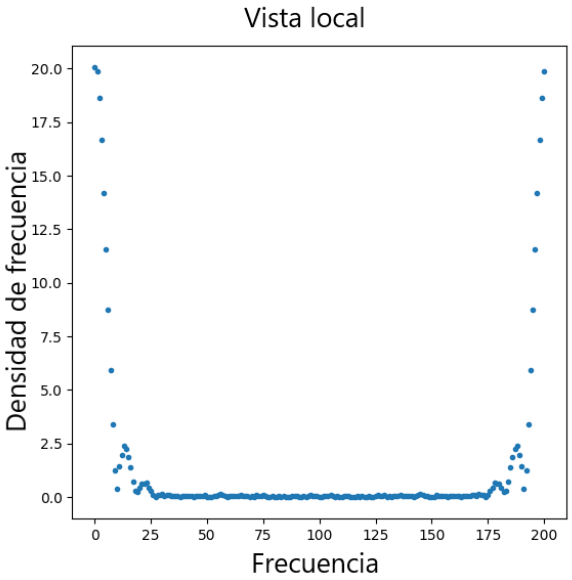


Figura 6. Transformada de Fourier de una curva de luz de «vista local» del conjunto de datos obtenido desde el trabajo de Shallue y Vanderburg[10].

métodos de normalización y estandarización implementados en la librería Scikit-learn[25] para formar un nuevo conjunto de datos, se pasa de tener dos conjuntos de datos, uno con 3.197 características, otro con dos vistas de 2,001 y 201 características, con un fuerte desequilibrio de datos a dos conjuntos de datos, uno con 15 características, otro con 36 características, con sus clases totalmente balanceadas.

2.4. Minería de datos

Para esta etapa del desarrollo realizado, con el uso del primer conjunto de datos, se tuvieron en cuenta dos algoritmos de inteligencia computacional para la clasificación automática de curvas de luz con firmas de tránsitos planetarios: Las redes neuronales, debido a su precisión; los sistemas difusos, debido a su significancia. Para el segundo conjunto de datos, se utilizaron únicamente los sistemas difusos y comparar estos resultados con los obtenidos por el sistema Astronet, evaluado con el mismo conjunto de datos.

En el desarrollo de los sistemas difusos, se tuvieron en cuenta un número de variables de entrada igual a cinco (5), una única variable de salida y dos (2) conjuntos difusos que definen los valores posi-

bles de cada una de estas variables: «planeta», que contiene los valores en los que una variable pueda encontrarse de haberse extraído de una curva de luz con tránsitos planetarios presentes; y «no planeta», que contiene los valores en los que se encontraría una variable de haberse extraído de una curva de luz sin tránsitos presentes. Para cada sistema, se consideró un conjunto de reglas que permitiera comparar todas las posibles combinaciones de las variables de entrada del sistema, de la siguiente manera.

1. Si (*Variable*1 = *planeta*) y
   (*Variable*2 = *planeta*) y
   (*Variable*3 = *planeta*) y
   (*Variable*4 = *planeta*) y
   (*Variable*5 = *planeta*)
   entonces (*Salida* = *planeta*),
2. Si (*Variable*1 = *planeta*) y
   (*Variable*2 = *planeta*) y
   (*Variable*3 = *planeta*) y
   (*Variable*4 = *planeta*) y
   (*Variable*5 = *noplaneta*)
   entonces (*Salida* = *planeta*),
3. Si (*Variable*1 = *planeta*) y
   (*Variable*2 = *planeta*) y
   (*Variable*3 = *planeta*) y
   (*Variable*4 = *noplaneta*) y
   (*Variable*5 = *noplaneta*)
   entonces (*Salida* = *planeta*),
4. Si (*Variable*1 = *planeta*) y
   (*Variable*2 = *planeta*) y
   (*Variable*3 = *noplaneta*) y
   (*Variable*4 = *noplaneta*) y
   (*Variable*5 = *noplaneta*)
   entonces (*Salida* = *noplaneta*),
   ...

De esta forma, todas las variables de entrada aportan en la misma magnitud al valor de la variable de salida.

Los sistemas difusos, durante el uso del primer conjunto de datos, fueron implementados haciendo uso de la librería Scikit-Fuzzy[31], se decidió que cada conjunto difuso tuviera dos funciones de pertenencia, y que estas fueran de forma Gaussiana y se tuvo en cuenta un conjunto de reglas que permitiera que todas las características tuvieran la misma influencia en la decisión del sistema al momento de la clasificación de los casos. De manera similar a como se hizo con las redes neuronales, se categorizaron tres tipos distintos de sistemas difusos, diferentes únicamente en los parámentros que reciben: un tipo que recibe las características extraidas de las curvas de luz, otro tipo que recibe las características extraidas de los gradientes de las curvas de luz y otro tipo que recibe las características extraidas de las transformadas de Fourier. Debido a la baja dimensionalidad que permiten manejar los sistemas difusos, se limitó a cinco (5) el número de características de un tipo de sistema difuso pudiera recibir, por lo que no se consideró ningún sistema difuso que recibiera más de cinco características.

Al utilizar el segundo conjunto de datos, la implementación de los sistemas difusos se diferencia en las características designadas como variables de entrada en los distintos sistemas a evaluar, para la selección de estas variables de entrada se hizo uso de la herramienta de análisis de datos Orange Data Mining, desarrollada por Demsar et al.[32], para hacer uso de una funcionalidad que la aplicación llamada «Sugerir características», la cual calcula un puntaje de las permutaciones de n variables, este puntaje computa la efectividad de clasificación de un clasificador K-Nearest Neighbor de los datos bidimensional proyectada linealmente y este refleja que tan bien se separan las clases en la proyección.

Al realizar este ejercicio, los tres (3) grupos de cinco (5) variables con mejor puntaje de categorización son: máximo de transformada de Fourier, sesgo de curva de luz, máximo de gradiente, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, mínimo de gradiente; máximo de transformada de Fourier, sesgo de curva de luz, máximo de gradiente, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, curtosis de curva de luz; y máximo de transformada de Fourier, Coeficiente B de ecuación de minimos cuadrados de transformada de Fourier, máximo de curva de luz, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, sesgo de curva de luz. Estos grupos de variables fueron los grupos de variables de entrada de los tres (3) sistemas difusos a optimizar y

evaluar. En la proyección lineal de cada uno de los grupos de variables encontrados con la herramienta de 'Sugerir características' de Orange (Ver Figura 7), puede verse que las variables seleccionadas realizan un agrupamiento adecuado de los casos positivos y negativos, sin embargo, no puede identificarse claramente mucha diferencia entre los valores de estas variables para ambas clases de ejemplos.

Para la calibración de los sistemas de inteligencia computacional considerados en este trabajo, se utilizó un algoritmo de Optimización por Enjambre de Particulas con un auto-calibrador de lógica difusa, implementado en la librería FST-PSO[33] para realizar una optimización global en rangos definidos de cada una de las propiedades de los sistemas y obtener soluciones de calibración factibles de manera automática.

3. Resultados y discusión

Para brindar un entendimiento mayor de los resultados obtenidos, se decidió dividir la discusión de resultados en dos (2) etapas, la primera etapa, en la que se utilizó el conjunto de datos obtenido de la plataforma Kaggle[22], y la segunda etapa, que viene después de optar por un cambio en el conjunto de datos considerado, en la que se utiliza el conjunto utilizado en el trabajo de Shallue y Vanderburg[10].

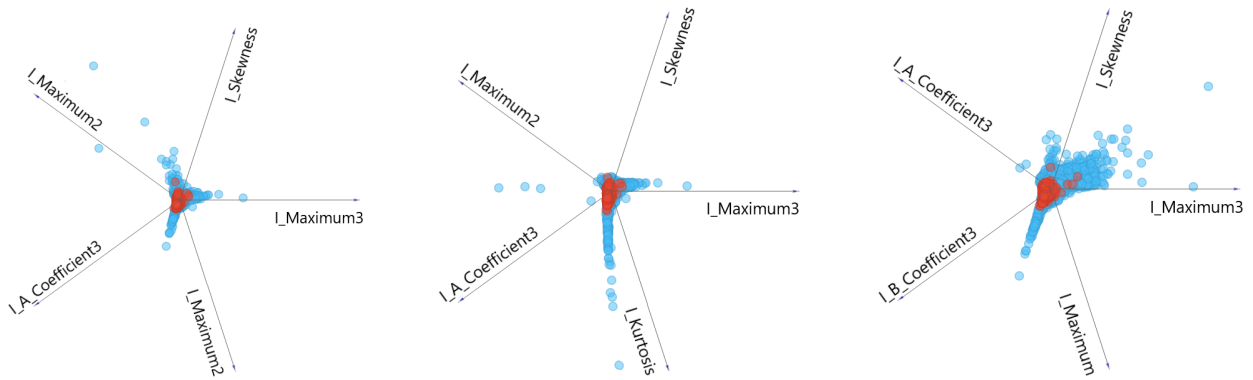
3.1. Etapa 1

Del proceso de minería de datos de la etapa 1 se pude identificar un claro efecto de sobreajuste en la mayoría de los sistemas difusos implem entados, lo que denota la alta varianza del problema de detectar tránsitos planetarios en curvas de luz continuas y sin técnicas de 'doblado', sin embargo, con este trabajo se pudo obtener un sistema difuso con un rendimiento remarcable, pues no solo obtuvo un buen rendimiento con los datos de entrenamiento (99.14 %) sino tambíen obtuvo un buen rendimiento con los datos de prueba (86.02 %), lo cual nos permite tener un nuevo sistema difuso del cual podemos analizar qué variables considero más significativas para la resolución del problema.

La Tabla 1 se refiere a los valores de efectividad de clasificación calculados de los resultados de clasificación de los conjuntos de datos de entrenamiento y prueba para cada sistema considerado en la etapa 1, de esta puede evidenciarse un rendimiento promedio mayor de parte de los sistemas difusos que consideraron las variables extraídas de la transformada de Fourier de las curvas de luz, lo cual nos indica que transformar las curvas de luz al dominio de la frecuencia puede contener información importante a tener en cuenta para la detección automática de tránsitos planetarios en curvas de luz sin mucho preprocesamiento.

Más allá del rendimiento promedio de cada tipo de clasificador, cabe resaltar el sistema difuso optimizado por Enjambre de Partículas que obtuvo el mejor rendimiento cuando a efectividad de clasificación se refiere, el cual, teniendo en cuenta las características extraidas de la transformada de Fourier de la curva de luz, obtuvo una efectividad de clasificación en los datos de entrenamiento de 99.1 % y de 86.0 % en los datos de prueba. Aprovechando la interpretabilidad que brinda la lógica difusa, podemos visualizar que las características más significantes para el sistema difuso son las mediciones de curtosis y entropía espectral de la transformada de Fourier de la curva de luz, mientras que el resto de características (el sesgo, la autocorrelación integral y la entropía espectral de la transformada de Fourier), no son significantes por lo que no contribuyen y hasta pueden afectar negativamente el proceso de clasificación del sistema experto (Ver Figura 8).

La matríz de confusión correspondiente al sistema difuso que obtuvo mayor rendimiento general en el proceso de clasificación realizado (Ver Tabla 2), 'FS3-R2', el cual obtuvo una efectividad de clasificación de 86.0 %, tuvo una efectividad de clasificación de 100 % en los casos positivos, sin cometer errores con falsos negativos, sin embargo, clasificó el 27.4 % de los casos negativos como positivos, lo que demuestra que el sistema es permisivo con los casos positivos, pasando incorrectamente varios casos negativos que pueden parecerle positivos, buscando evitar la clasificación errónea de los casos positivos. Este sistema realizado tambíen demostró el mayor rendimiento en la correcta clasificación de casos negativos, es decir, la identificación de falsos positivos, con 87 falsos positivos obtenidos durante el proce-



(a) (b) (c)

Figura 7. Proyección lineal de los tres (3) grupos de variables encontrados mediante la busqueda de los grupos con mayor distinción entre los casos positivos y negativos realizada con la herramienta Orange. (a) máximo de transformada de Fourier, sesgo de curva de luz, máximo de gradiente, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, mínimo de gradiente (b) máximo de transformada de Fourier, sesgo de curva de luz, máximo de gradiente, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, curtosis de curva de luz (c) máximo de transformada de Fourier, Coeficiente B de ecuación de minimos cuadrados de transformada de Fourier, máximo de curva de luz, coeficiente A de ecuación de minimos cuadrados de transformada de Fourier, sesgo de curva de luz.

Tabla 1. Efectividad de clasificación de sistemas considerados en etapa 1

Efectividad de clasificación (%)							
Sistema difuso 1 (FS_1)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
Entrenamiento	Prueba	Entrenamiento	Prueba	Entrenamiento	Prueba	Entrenamiento	Prueba
86,20	42,10	89,81	45,63	81,82	46,71	85,94	44.81
Sistema difuso 2 (FS_2)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
95,61	31,97	98,93	49,10	88,19	46,79	94,43	42.62
Sistema difuso 3 (FS_3)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
99,42	52,25	99,14	86,02	99,07	45,52	99,21	61,26

so de clasificación con los datos de entrenamiento, un 1.7 % de los casos negativos, y 155 casos negativos confundidos por positivos por el sistema con los datos de prueba, un 27.4 % de los casos negativos.

3.2. Etapa 2

En la segunda etapa del trabajo desarrollado, se pudo evidenciar que las redes neuronales llevan la supremacía con respecto a precisión se refiere, pues el sistema Astronet en su momento tuvo mejores resultados correspondientes a la efectividad de clasificación, con un 96,0 % evaluando con datos de prueba[10], mientras que el sistema difuso con mayor efectividad de clasificación obtenido llegó hasta el 90.4 % evaluando los datos de prueba, sin embargo, cabe resaltar, que los resultados obtenidos con

Tabla 2. Matriz de Confusión de Sistema difuso 'FS3-R2'

		Valores predichos (%)		
		p	**n**	**total**
Valores Reales (%)	**p′**	100.00	27.43	P′
	n′	0.00	72.57	N′
	total	P	N	

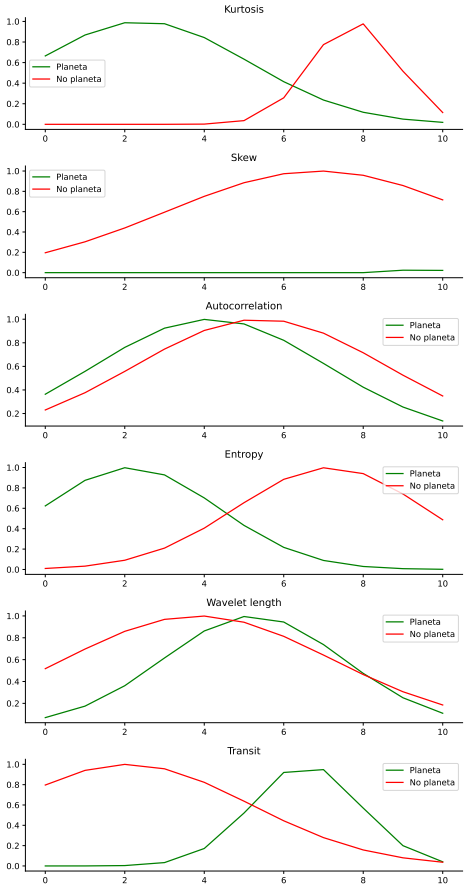


Figura 8. Funciones de pertenencia para los conjuntos difusos considerados en el sistema difuso con mayor efectividad de clasificación de la etapa 1

la implementación de sistemas difusos son teniendo en cuenta solo cinco (5) variables de entrada computadas a partir de los datos de la curva de luz de cada ejemplo, mientras que Astronet recibe la curva de luz de «vista global» del evento, compuesta de 2,001 variables de entrada, y la curva de luz de «vista local» del evento, compuesta por 201 variables de entrada, lo cual deja ver la diferencia de dimensionalidad entre ambos modelos y la ventaja que podría otorgar la implementación de modelos de clasificación basados en sistemas difusos con respecto a la cantidad de variables de entrada que se debe analizar para su posterior uso.

De los resultados obtenidos con este trabajo acerca de la efectividad de clasificación de cada uno de los clasificadores considerados (Ver 3 y su comparación con los resultados de los sistemas difusos anteriores se puede evidenciar una mejora en la generalización de los sistemas difusos, pues se resolvió el problema de sobreajuste de la etapa anterior al tratar las curvas de luz ya procesadas y «dobladas» en el fenómeno a estudiar y al realizar una selección más objetiva de las características que cada sistema difuso recibe como variables de entrada.

La matriz de confusión del sistema difuso con mayor rendimiento es FS2-R2 (Ver Tabla 4), el cual obtuvo una efectividad de clasificación en los casos positivos de un 91,76 %, solo un 8,23 % de falsos positivos, evidencia una sensibilidad de un 89,33 % y una especifidad de un 91,53 %, lo que muestra la capacidad superior del modelo para la discriminación de casos negativos a la capacidad del mismo para detectar los casos relevantes.

Del análisis cualitativo de los conjuntos difusos resultantes de la optimización de este sistema (Ver Figura 9) se puede identificar que en la mayoría de las variables de entrada y en la variable de salida existe una función de pertenencia que contiene a la otra, lo que podría indicar la similitud de los valores de las variables de entrada para los casos positivos y negativos, lo que dificultó el objetivo de la optimización del modelo para encontrar conjuntos de dos funciones de pertenencia que aporten de manera significativa a la distinción entre los casos negativos y los casos positivos, debido a esto, el sistema fue «obligado» a realizar una categorización de las casos de entrenamiento considerando que los valores de una clase se encuentran en el rango de los valores de la otra clase, que le permitio al sistema obtener un rendimiento considerable, el cual podría mejorar considerando otras variables de entrada más significativas para la distinción de las clases de los casos o un número mayor de funciones de pertenencia consideradas por cada variable. En este sistema, las variables de entrada más significativas (aunque no por mucho) para la distinción de casos negativos y positivos son *l_Maximum*3, valor correspondiente al valor máximo de la transformada de Fourier de la vista local de la curva de luz, y *l_Kurtosis*, que corresponde al valor de la curtosis de la vista local de la curva de luz.

4. Conclusiones

La significancia de los sistemas difusos permite un análisis detallado de la solución encontrada durante

Tabla 3. Efectividad de clasificación de sistemas considerados en la etapa 2

Efectividad de clasificación (%)							
Sistema difuso 1 (FS_1)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
Entrenamiento	Prueba	Entrenamiento	Prueba	Entrenamiento	Prueba	Entrenamiento	Prueba
87,81	87,23	62,81	62,93	83,38	83,2	78.00	77,78
Sistema difuso 2 (FS_2)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
76,60	76,07	90,45	90,40	86,35	86,24	84,47	84,24
Sistema difuso 3 (FS_3)							
Primera ejecución ($R1$)		Segunda ejecución ($R2$)		Tercera ejecución ($R3$)		Promedio	
80,02	79,70	86,62	86,45	79,70	78,71	82,11	81,62

Tabla 4. Matriz de Confusión de Sistema difuso 'FS2-R2'

		Valores predichos (%)		
		p	**n**	**total**
Valores Reales (%)	**p′**	91.76	10.95	P′
	n′	8.23	89.04	N′
	total	P	N	

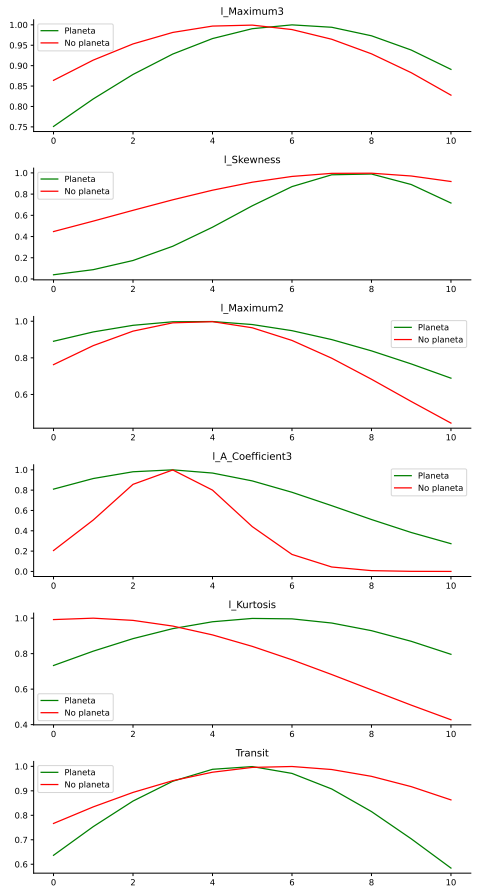


Figura 9. Funciones de pertenencia para los conjuntos difusos considerados en el sistema difuso con mejor rendimiento promedio en la etapa de análisis con el conjunto de datos de [10]

el proceso de optimización global de los sistemas implementados, de este análisis se puede evidenciar la importancia de cada variable de entrada utilizada en los distintos sistemas difusos, entre las que resaltan la curtosis de la curva de luz, la entropía de la transformada de Fourier de la curva de luz y el valor máximo de la transformada de Fourier de la curva de luz, pues fueron las variables con mayor significancia durante el desarrollo del trabajo propuesto.

Teniendo en cuenta la curtosis, durante la etapa 1 del desarrollo de este trabajo, se puede evidenciar que un valor bajo o medio de curtosis está relacionado con una probabilidad mayor de que la curva de luz analizada posea algún tránsito planetario, sin embargo, durante la etapa 2, un valor muy bajo de curtosis está relacionado con una probabilidad mayor de que la curva de luz analizada no presente ningún tránsito planetario. Con respecto a la entropía, durante la etapa 1, un valor bajo de entropía se asocia con una probabilidad mayor de que la curva de luz presente un tránsito planetario; y al analizar los conjuntos difusos del valor máximo de la transformada de Fourier de la curva de luz, un valor alto está relacionado a una probabilidad mayor de que

exista un tránsito planetario en la curva de luz.

Con respecto a los resultados cuantitativos del desarrollo reaizado, los sistemas difusos resultan ser un modelo de inteligencia computacional con un rendimiento aceptable para la detección autónoma de rastros de tránsitos planetarios en curvas de luz, pues en el estudio realizado se obtuvo una efectividad de clasificación superior al 90.4 % obtenida del sistema difuso optimizado con mayor rendimiento, desarrollado en este trabajo. Tambíen puede evidenciarse que los sistemas difusos son capaces de generalizar mejor el fenómeno al utilizar datos extraidos de curvas de luz 'dobladas' que utilizando datos extraidos de curvas de luz en un dominio de tiempo continuo, cuyo procesamiento previo para la eliminación de perturbaciones lumínicas producto de otros fenómenos es mucho menor.

Para trabajos futuros, se propone el estudio de la utilidad de los sistemas difusos utilizando estas u otras variables de entrada considerando más conjuntos difusos y de otras formas geométricas para cada variable, como funciones triangulares o trapezoidales, también, se puede estudiar el uso de otros algoritmos de optimización y abarcar una optimización de las reglas consideradas para los sistemas difusos implementados.

Declaración de conflicto de interés: Los autores manifiestan no tener conflictos de interés.

Referencias

[1] A. Megías, "Caracterización de sistemas exoplanetarios mediante el ajuste de series temporales fotométricas y de velocidad radial", Trabajo de fin de Máster, Departamento de Física de la Tierra y Astrofísica, Facultad de Ciencias Físicas, Universidad Complutense de Madrid, Madrid, 2019.

[2] E.A. Rojas, "Análisis de curvas de luz de exoplanetas utilizando datos de la sonda espacial Kepler", Trabajo de graduación, Departamento de Física, Universidad de San Carlos de Guatemala, Guatemala, 2016.

[3] P. Brennan, "Exoplanet Exploration: Planets Beyond our Solar System", 2022. Disponible en https://exoplanets.nasa.gov/discovery/exoplanet-catalog. Obtenido en enero 11, 2023.

[4] J.G. Ahuatzi, "Ajuste simultáneo a curvas de luz y velocidad radial para sistemas en tránsito", Instituto Nacional de Astrofísica, óptica y Electrónica, Puebla, México, 2014.

[5] G. Kovács, S. Zucker & T. Mazeh, "A box-fitting algorithm in the search for periodic transits", en *A&A*, vol. 391, no. 1, pp. 369-377, Agosto, 2002. https://doi.org/10.1051/0004-6361:20020802

[6] D. Mislis, E. Bachelet, K. A. Alsubai, et al, "sidra: a blind algorithm for signal detection in photometric surveys", en *MNRAS*, vol. 455, no. 1, pp. 626â633, Enero, 2016. https://doi.org/10.1093/mnras/stv2333

[7] D.S. Julian, "Búsqueda de tránsitos planetarios alrededor de enanas ultrafrías", Trabajo de fin de Máster, Universidad de La Laguna, 2019.

[8] NASA Exoplanet Science Institute, "Exoplanet and Candidate Statistics"[online]. NASA Exoplanet Archive, 2022. Disponible en https://exoplanetarchive.ipac.caltech.edu/docs/counts_detail.html

[9] G.C. Sturrock, B. Manry & S. Rafiqi, "Machine Learning Pipeline for Exoplanet Classification", en *SMU Data Sci. Rev.*, vol. 2, no. 1, Article 9, 2019. https://scholar.smu.edu/datasciencereview/vol2/iss1/9

[10] C. J Shallue & A. Vanderburg, "Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90", en *ApJ*, vol. 155, no. 2, pp 21, 2018. https://doi.org/10.3847/1538-3881/aa9e09

[11] M. Ansdell, Y. Ioannou, H. P. Osborn, et al, "Scientific Domain Knowledge Improves Exoplanet Transit Classification with Deep Learning", en *ApJL*, vol. 869, L7, 2018. https://doi.org/10.3847/2041-8213/aaf23b

[12] L. Ofman, A. Averbuch, A. Shliselberg, et al, "Automated identification of transiting exoplanet candidates in NASA Transiting Exoplanets Survey Satellite (TESS) data with machine learning methods", en *NewA*, vol. 91, no. 1, 2022. https://doi.org/10.1016/j.newast.2021.101693

[13] R. M. Asif Amin et al, "Detection of exoplanet systems in Kepler light curves using adaptive neuro-fuzzy system", 2018 International Conference on Intelligent Systems (IS), pp. 66â72, 2018. https://doi.org/10.1109/IS.2018.8710502

[14] H. Valizadegan et al, "ExoMiner: A Highly Accurate and Explainable Deep Learning Classifier That Validates 301 New Exoplanets", en *ApJ*, vol. 926, no. 2, pp. 120. https://doi.org/10.3847/1538-4357/ac4399

[15] C. Moutou & F. Pont, "Detection and characterization of extrasolar planets: the transit method", en *Ecole de Goutelas*, vol. 28, pp. 55â79, 2006. http://astro.u-strasbg.fr/goutelas/g2005/

[16] W. Frawley et al, "Knowledge Discovery in Databases: An Overview", en *AI Mag.*, 13(3), 57, 1992. https://doi.org/10.1609/aimag.v13i3.1011

[17] O. Maimon & L. Rokach, "Introduction to Knowledge Discovery in Databases", en *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA, pp 1-14. https://doi.org/10.1007/0-387-25465-X_1

[18] F. Dernoncourt, "Introduction to fuzzy logic", [online]. Massachusetts Institute of Technology, Cambridge, MA. 2013. Disponible en http://aisii.azc.uam.mx/mcbc/Cursos/IntCompt/Lectura15.pdf

[19] Serena Yeung, "What is Computer Vision?"[online]. Stanford Artificial Intelligence Laboratory's Outreach Summer camp (SAILORS), 2015. Disponible en https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

[20] M. Clere, "Particle Swarm Optimization ", en *Hermes Science/Lavoisier*, Paris, Francia, 2005. https://doi.org/10.1002/9780470612163

[21] Robinson, Jacob & Rahmat-Samii, Yahya. "Y.: Particle Swarm Optimization in Electromagnetics. IEEE Trans. on Antennas and Propagation *Antennas and Propagation*. https://doi.org/10.1109/TAP.2004.823969

[22] "Exoplanet Hunting in Deep Space"[online], Kaggle, 2016. Disponible en https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data

[23] J. H. Catanzarite, "Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24", NASA Ames Research Center, 2015. KSCI-19091-001

[24] G. Lemaitre et al, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning", arXiv, 2016. https://doi.org/10.48550/ARXIV.1609.06570

[25] F. Pedregosa et al, "Scikit-learn: Machine Learning in Python", arXiv, 2012. https://doi.org/10.48550/ARXIV.1201.0490

[26] U. Rajendra et al, "Application of entropies for automated diagnosis of epilepsy using EEG signals: A review", en *Knowledge-Based Syst.*, vol. 88, pp. 85-96, Noviembre, 2015. https://doi.org/10.1016/j.knosys.2015.08.004

[27] O. Ulkir et al, "Emg Signal Classification Using Fuzzy Logic", en *Balkan j. Electric. Comput. Eng.*, vol. 5, no. 2, pp. 97-101, Septiembre, 2017, https://doi.org/10.17694/bajece.337941

[28] P. Virtanen et al, "SciPy 1.0: fundamental algorithms for scientific computing in Python", en *Nat. Methods*, vol. 17, pp. 261-272, Febrero 2020. https://doi.org/10.1038/s41592-019-0686-2

[29] C. R. Harris et al, "Array programming with NumPy", en *Nat*, vol. 585, pp. 357-362, 2020. https://doi.org/10.1038/s41586-020-2649-2

[30] F. Bao et al, "PyEEG: An Open Source Python Module for EEG/MEG Feature Extraction", en *Comput. Intell. Neurosci.*, vol. 2011, 406391, 2011. https://doi.org/10.1155/2011/406391

[31] J. Warner et al, "JDWarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2 (v0.4.2)", Zenodo, 2019. https://doi.org/10.5281/zenodo.802396

[32] J. Demsar et al, "Orange: Data Mining Toolbox in Python", en *j. Mach. Learn. Res.*, vol. 14, pp. 2349-2353, 2013. http://jmlr.org/papers/v14/demsar13a.html

[33] M. S. Nobile et al, "Fuzzy Self-Tuning PSO: A Settings-Free Algorithm for Global Optimization", en *Swarm Evol. Comput.*, vol. 39, pp. 70-85, Abril, 2018. https://doi.org/10.1016/j.swevo.2017.09.001