UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DETECCIÓN DE NOVEDADES EN CURVAS DE LUZ BASADO EN APRENDIZAJE DE MÁQUINAS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

MAURICIO JAVIER ROMERO JOFRÉ

PROFESOR GUÍA:
PABLO ESTÉVEZ VALENCIA

MIEMBROS DE LA COMISIÓN:
FELIPE TOBAR HENRÍQUEZ
FRANCISCO FÖRSTER BURÓN

Este trabajo ha sido parcialmente financiado por el proyecto ANID IC12009, Instituto Milenio de Astrofísica y el proyecto Fondecyt regular 1220829.

SANTIAGO DE CHILE
2022

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: MAURICIO JAVIER ROMERO JOFRÉ
FECHA: 2022
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

DETECCIÓN DE NOVEDADES EN CURVAS DE LUZ BASADO EN APRENDIZAJE DE MÁQUINAS

Se propone un método de detección de outliers para curvas de luz (CL), basado en transformaciones de series de tiempo usadas como outliers auxiliares o aumento de datos. Se asume que los *outliers* son desconocidos y que solo se tiene acceso a un conjunto de *inliers*. Cada CL es codificada a un vector de tamaño fijo a través de una red neuronal. Un puntaje de anomalía es calculado en base a la cercanía al clúster más cercano en el espacio de las representaciones. El modelo es aplicado a los conjuntos de datos de los surveys ZTF, ASAS, LINEAR y ASAS-SN. Para la selección de modelo, se estiman métricas sustitutas con el conjunto de validación. Los resultados muestran que el método propuesto supera a los del estado del arte alcanzando un AUCPR promedio de 0.89 en la detección de outliers en los cuatro conjuntos de datos. La relevancia del trabajo radica en que es posible determinar qué tan novedosa es una curva de luz desconocida a partir de la señal misma en vez de calcular de características predefinidas, pudiendo ahorrar tiempo de cómputo y almacenamiento, en el contexto del procesamiento masivo de datos astronómicos que se espera en el futuro cercano.

Para Dios, mis padres, hermanas y amigos.

Agradecimientos

En primer lugar quiero agradecer a Dios, todo es a través de Él y para Él. Gracias por mantenerme en rumbo. Agradezco a mis padres Claudia Jofré y Mauricio Romero por ser los pilares de apoyo en todo lo que ha sido mi vida hasta el momento. A mis hermanas Carolina y Pamela por darme ánimos y el apañe.

Agradezco también a mis amigos y a la Radio Integral por haber hecho más amena la estadía en la universidad. En especial a Felipe Alarcón y Diego Aichele que ya llevamos harto camino recorrido juntos.

Quiero agradecer al profesor Pablo Estévez por la guía durante el magíster y la crítica constructiva dada al trabajo aquí mostrado. En especial a la dedicación para sacar en conjunto el *paper* relacionado a la tesis. También quiero agradecer a los compañeros del Laboratorio de Inteligencia Computacional, en especial a Óscar Pimentel y Nicolás Astorga por todas las veces que discutimos sobre algoritmos, procesamiento de datos y códigos para hacer ciertas cosas.

Finalmente, quiero agradecer al Departamento de Ingeniería Eléctrica de la Universidad de Chile por haber financiado el costo del magíster a través de la Beca de Excelencia Académica, al proyecto ANID Instituto Milenio de Astrofísica IC12009 que me mantuvo financiado durante los estudios, y al proyecto Fondecyt regular 1220829.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

La investigación astronómica moderna se lleva a cabo a partir del análisis de datos adquiridos de observatorios astronómicos. La cantidad de datos que los observatorios astronómicos generan por noche ha crecido con el tiempo. Por ejemplo, el *survey Zwicky Transient Facility* (ZTF) [43] genera del orden de 1.4 TB de datos por noche mientras que el *Vera C. Rubin Observatory* [30], en construcción en Chile, generará del orden de 20 TB de datos por noche a través del *survey Legacy Survey of Space and Time* (LSST). En ese sentido, los astrónomos han automatizado el procesamiento y análisis de grandes cantidades de datos para hacer ciencia. Un problema desafiante y llamativo es la detección de objetos nunca antes vistos por el ser humano. En la literatura este tipo de problema es llamado detección de novedades, anomalías u *outliers*. En el contexto bajo el cual se centra la presente tesis lo más correcto es llamar al problema a resolver como de detección de novedades pero se usarán los tres términos de forma indistinta [1].

Los datos generados en los observatorios astronómicos ópticos son imágenes del cielo nocturno que se pueden transformar a series de tiempo de luminosidad, llamadas curvas de luz. Las curvas de luz se caracterizan por ser muestreadas de forma irregular, tener largo variable y ser multidimensionales. Pueden ser obtenidas desde diferentes regiones del espectro electromagnético (multibanda). Las curvas de luz usadas en este trabajo tienen solo una banda y poseen tres componentes: tiempo, magnitud y error de la magnitud. Los algoritmos de detección de *outliers* deberían ser capaces de detectar objetos anómalos con las restricciones mencionadas, sin importar el largo de la curva de luz. Este problema es naturalmente no supervisado ya que los objetos desconocidos no se pueden etiquetar de antemano.

La detección de *outliers* en series de tiempo ha sido estudiada desde distintos puntos de vista: anomalías puntuales, anomalías en subsecuencias y series de tiempo completas anómalas [1]. Este trabajo se enfoca en la detección de *outliers* en la curva de luz completa. Algunos trabajos previos han intentado determinar el estado anómalo de una serie de tiempo a partir de la definición de métricas para puntos anómalos en series de tiempo [34]. Sin embargo, en el escenario astronómico se asume que los puntos de la serie de tiempo tienen un comportamiento normal sin importar si pertenecen a curvas de luz *inlier* u *outlier*. Entonces,

es necesario darle un puntaje a la curva de luz completa independiente del puntaje de una sola observación.

Los algoritmos basados en aprendizaje profundo se enfocan principalmente en detectar anomalías puntuales o en subsecuencias. Algunos métodos han sido desarrollados para la detección de *outliers* para series de tiempo completas pero no pueden manejar series de tiempo de largo variable [53] [29] [3] [37]. En [4] se manejan datos de largo variable usando deformación dinámica del tiempo (DTW, del inglés *dynamic time warping*), pero que no trata con series de tiempo con muestreo irregular directamente. En [52] se proponen transformaciones aprendibles, en donde series de tiempo y sus transformaciones son codificadas como vectores. A través de aprendizaje contrastivo se fomenta que esos vectores queden cercanos en el espacio de representaciones mientras que al mismo tiempo la distancia se maximice entre series de tiempo transformadas con otras transformaciones. Sin embargo, no es claro como utilizar ese método en curvas de luz puesto que los conjuntos de datos probados corresponden a series de tiempo muestreadas de forma regular lo cual no se cumple para los datos usados en este trabajo. Otros métodos para series de tiempo son descritos en los *surveys* [5] [69] [23].

Algunos métodos de detección de *outliers* fueron desarrollados para imágenes pero pueden ser modificados para series de tiempo. En [21] transformaciones geométricas de imágenes son usadas para entrenar clasificadores que discriminen entre transformaciones. Aunque las transformaciones pueden ser seleccionadas [54], la desventaja de este método es que definir transformaciones geométricas para series de tiempo no es una tarea sencilla como también se señala en [52]. En [28] se propone una tarea de aprendizaje extra al entrenar clasificadores. Una de las tareas auxiliares corresponde a la maximización de la entropía de la distribución de probabilidad de salida del clasificador, con la adición de ejemplos fuera de la distribución (OOD del inglés *out-of-distribution*). En ese enfoque, otros conjuntos de datos de series de tiempo son usados como datos OOD, pero eventualmente estos pueden ser reemplazados por transformaciones de series de tiempo.

Los enfoques de detección de *outliers* en astronomía son muy diversos, entre ellos se han desarrollado métodos supervisados [46] y semi-supervisados [47]. Un método para anomalías puntuales de objetos transientes es descrito en [71]. En [41] [70] son usadas características en conjunto con el algoritmo *isolation forest* (IF) para detectar *outliers* en catálogos de objetos transientes. Específicamente, [70] usa algoritmos de *clustering* y de visualización antes de detectar anomalías. En [63] se estima la distribución de cantidades derivadas de las curvas de luz de objetos periódicos utilizándose la verosimilitud como puntaje de anomalía.

Recientemente, métodos basados en aprendizaje profundo han sido desarrollados. En [66] un autocodificador (AE del inglés *autoencoder*) es extendido con un modelo de mezcla de gaussianas (GMM del inglés *gaussian mixture model*) en el espacio latente. Elementos *outliers* son detectados a partir de vectores en el espacio latente a través de la medición de la energía del GMM. Este método fue desarrollado particularmente para curvas de luz periódicas. En [68] [58] AE variacionales recurrentes y AE recurrentes variacionales [17] son usados para detección de anomalías para curvas de luz transientes y estocásticas, respectivamente. En ambos casos un IF es entrenado en el espacio latente para detectar *outliers*.

La propuesta de tesis se centra en dos puntos principales. El primero consta en utilizar transformaciones en series de tiempo que transforman una señal de clase conocida a una

serie *outlier* auxiliar o que genere un nuevo dato manteniendo la clase de origen (análogo a hacer aumento de datos). No es fácil determinar que transformaciones tienen esa cualidad. El segundo punto es aprovechar las series de tiempo transformadas como datos auxiliares desconocidos o de aumento de datos, y utilizarlas en un algoritmo de aprendizaje de similitud [42]. La idea es aprender la similitud en un espacio de representaciones inducido por una red neuronal. La red neuronal recibe una curva de luz y la codifica en una representación vectorial. Luego, la distancia entre representaciones de la misma clase es minimizada mientras que en caso contrario, la distancia entre representaciones es maximizada.

1.2. Hipótesis

Las hipótesis de este trabajo son:

- Es posible aprender un puntaje de *outlier* (*outlier score*) directamente desde la curva de luz sin el cálculo explícito de características mediante aprendizaje contrastivo.
- Existen transformaciones de curvas de luz que construyendo un conjunto auxiliar de *outliers* o un conjunto de datos aumentados, ayudan a resolver el problema de detección de novedades en curvas de luz.
- Es posible desarrollar métricas sustitutas que estén correlacionadas con el área bajo la curva *precision-recall* (AUCPR) que se pueden estimar con el conjunto de validación.

1.3. Objetivos Generales

El objetivo general de la tesis es desarrollar un algoritmo de detección de novedades utilizando directamente la curva de luz basado en redes neuronales no-supervisado.

1.4. Objetivos Específicos

- Desarrollar e implementar una red neuronal no-supervisada que entregue un puntaje de anomalía para una curva de luz.
- Determinar hiperparámetros tales como la función de distancia y función de costos dentro de un conjunto de valores posibles.
- Proponer una métrica de desempeño que tome en cuenta el desbalance de los elementos *outliers*.
- Desarrollar un esquema de selección de transformaciones y determinar la(s) transformación(es) a utilizar.
- Determinar la correlación de métricas sustitutas evaluadas en el conjunto de validación, con el AUCPR evaluado en el conjunto de prueba.
- Comparar el desempeño del algoritmo propuesto con el estado del arte en detección de *outlier* basado en curvas de luz.

1.5. Contribuciones

La contribución de la presente tesis radica en cuatro puntos:

- Resolver el problema de detección de *outliers* en curvas de luz con un enfoque no supervisado usando aprendizaje contrastivo.
- Proponer transformaciones para la detección de *outliers* en series de tiempo.
- Analizar dos escenarios para las transformaciones: aumento de datos o *outliers* auxiliares.
- Proponer métricas sustitutas para la selección de modelo bajo las condiciones de evaluación difíciles como la detección de *outliers*.

1.6. Estructura de la Tesis

La tesis se divide en 4 capítulos además de la introducción. El capítulo 2 consiste en el marco teórico que sustenta el trabajo desarrollado. En este se introducen los términos astronómicos utilizados, los algoritmos basados en redes neuronales y los distintos enfoques de resolución y definición del problema de detección de *outliers*. El capítulo 3 consiste en la metodología seguida en los distintos experimentos realizados en la tesis. El capítulo 4 se refiere a los resultados y análisis de estos. Finalmente, en el capítulo 5 se concluye el trabajo de tesis desarrollado y se contrasta lo obtenido con las hipótesis y objetivos de este trabajo, además se incluye el trabajo futuro.

Capítulo 2

Marco Teórico

2.1. Astronomía

2.1.1. Curvas de Luz

Las curvas de luz corresponden a la medición del brillo en función del tiempo de un objeto astronómico. El brillo que se mide de forma local en la Tierra, también llamado brillo aparente o flujo, corresponde a la densidad de potencia visto por el observador [10]:

$$F = \frac{E}{t \cdot A}, \tag{2.1}$$

donde E es la energía recibida por el observador, t el tiempo de observación y A el área de la superficie del medidor.

Sin embargo, las mediciones de flujo varían en órdenes de magnitud por lo que comúnmente se realiza una transformación logarítmica sobre el flujo para obtener una cantidad llamada magnitud:

$$m = -2{,}5 \cdot \log_{10}(F) + K, \tag{2.2}$$

donde m es la magnitud aparente, F el flujo y K una constante llamada *zero point*. El *zero point* es utilizado para calibrar la medición de una observación a partir de la magnitud de un objeto astronómico conocido. Es posible determinar la diferencia de magnitud entre dos objetos como:

$$\mathrm{d}m = m_1 - m_2 = -2{,}5 \cdot \log_{10}(\frac{F_1}{F_2}), \tag{2.3}$$

donde dm es la diferencia de magnitud, m_i es la magnitud del objeto i y F_i es el flujo del objeto i. Esta última ecuación es útil para estimar la magnitud aparente de un objeto a

partir de otro objeto cuya magnitud aparente es conocida.

Para estimar el flujo de un objeto astronómico se utilizan dos tipos de mediciones: fotometría y espectroscopía. La fotometría es la medición de la energía recibida a partir de una fuente, mientras que la espectroscopía mide la distribución de energía respecto a la longitud de onda o frecuencia de la onda electromagnética. Los datos utilizados en la tesis están basados en mediciones fotométricas.

A partir de un sensor de imágenes de tecnología dispositivo de carga acoplada (CCD del inglés *charged-coupled device*), es posible generar una imagen de la energía recibida por el dispositivo. El valor de cada pixel del CCD corresponde a la cantidad de fotones recibidos por el detector, y es proporcional a la energía recibida. La imagen generada es una representación distorsionada del flujo del objeto astronómico debido al poder finito de resolución del lente, la dispersión de la luz en la atmósfera o la dispersión de los fotones entre pixeles del CCD, por lo que las fuentes puntuales son dispersadas en el CCD.

Con los valores de los pixeles del CCD es posible estimar el flujo aparente a partir de la suma de la energía de cada pixel del detector:

$$F = \frac{\sum_{\mathrm{i},j} E_{\mathrm{i},j}}{t \cdot A_l}, \tag{2.4}$$

donde t es el tiempo de exposición, A_l es el área del lente de la cámara y $E_{\mathrm{i},j}$ la energía recibida del objeto de interés en el pixel (i, j). Sin embargo, esta medición no corresponde directamente al brillo de un objeto astronómico debido a la superposición de luz a partir de otras fuentes. Entonces, la energía del objeto en cada pixel del detector corresponde a:

$$E_{\mathrm{i},j} = S_{\mathrm{i},j} - B_{\mathrm{i},j}, \tag{2.5}$$

que corresponde a la resta entre el flujo total $S_{\mathrm{i},j}$ y el flujo de fondo $B_{\mathrm{i},j}$ de otras fuentes, el cual debe ser estimado. Luego, al reemplazar esta última expresión en (2.3), es posible medir la magnitud aparente de un objeto de la siguiente forma:

$$m_1 = m_2 - 2{,}5 \cdot \log 10(\frac{\sum_{\mathrm{i},j}(S_{\mathrm{i},j} - B)_1}{\sum_{\mathrm{i},j}(S_{\mathrm{i},j} - B)_2}), \tag{2.6}$$

donde m_1 es la magnitud aparente del objeto a medir, m_2 es la magnitud aparente del objeto conocido, S es la energía de cada pixel y B es la estimación del fondo por pixel. El procedimiento anterior es llamado fotometría diferencial. Es posible estimar la curva de luz para distintas bandas de frecuencia. Entonces, es posible determinar la magnitud aparente de forma separada por banda de interés.

Una posible clasificación de curvas de luz viene dada respecto a como se caracteriza temporalmente, en donde es posible distinguir entre objetos aperiódicos como transientes y estocásticos, y periódicos. Los objetos transientes corresponden a fenómenos que de forma transitoria

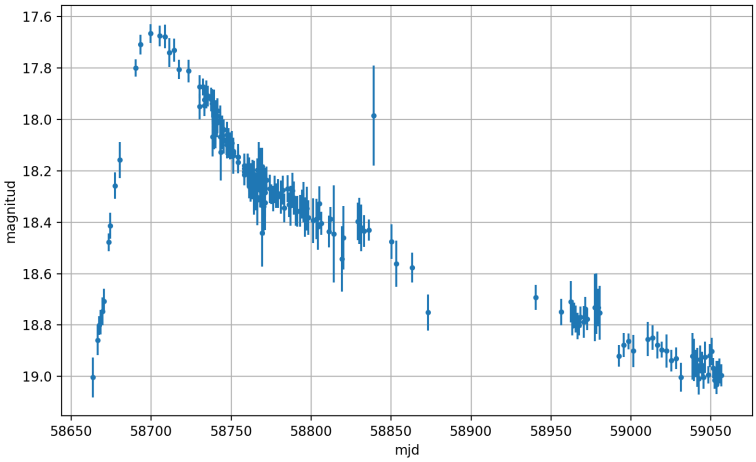

Figura 2.1: Ejemplo de curva de luz transiente.

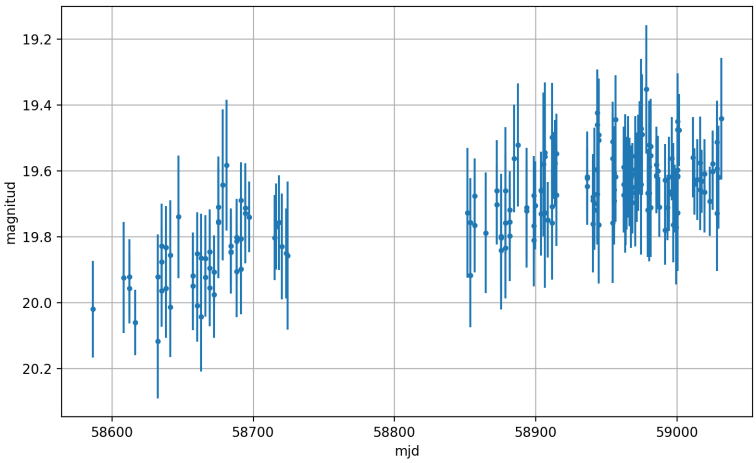

Figura 2.2: Ejemplo de curva de luz estocástica.

presentan un aumento de flujo en la curva de luz. Los objetos estocásticos son aquellos cuya formación está basada en procesos estocásticos. Los objetos periódicos presentan un patrón repetitivo en el tiempo [59]. En particular, es posible cambiar la representación temporal de las curvas de luz periódicas en una representación en fase, en un proceso llamado minimización de la dispersión de fase [64]. El tiempo viene dado por el número de días desde la medianoche del 17 de noviembre de 1858, llamado fecha juliana modificada (MJD del inglés *modified julian date*). La MJD está basada en la fecha juliana que considera el número de días desde el mediodía del 1 de enero de 4713 a.c. En las Figuras 2.1 y 2.2 se ven ejemplos de curvas de luz transiente y estocástica respectivamente. En las Figuras 2.3 y 2.4 se ven ejemplos de curvas de luz periódicas en tiempo y en fase, respectivamente.

2.2. Redes Neuronales Artificiales

Las redes neuronales artificiales son un tipo de modelo matemático inspirado en la interconexión de neuronas del cerebro. Son un modelo simplificado compuesto de unidades compu-

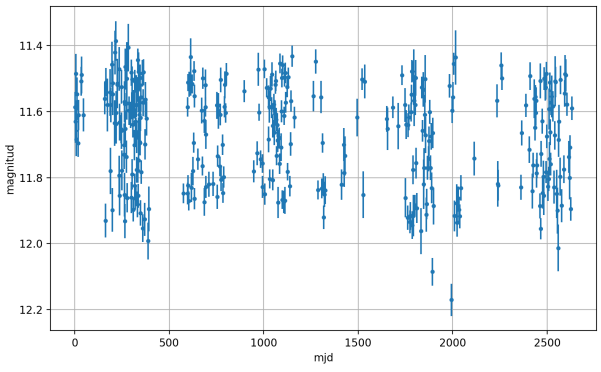

Figura 2.3: Ejemplo de curva de luz periódica en el dominio temporal.

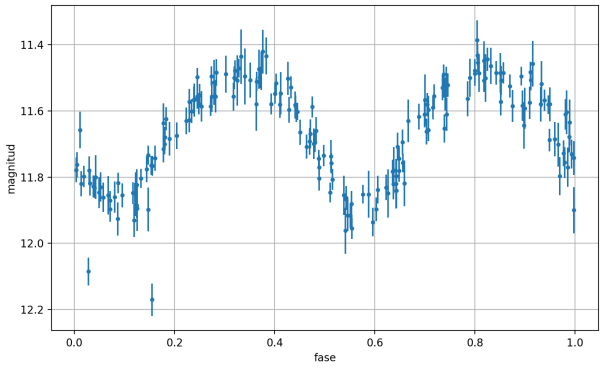

Figura 2.4: Ejemplo de curva de luz periódica en el dominio de la fase.

tacionales más pequeñas. Estas unidades computacionales representan un modelo plausible de neurona. McCulloch and Pitts [44] propusieron un modelo neuronal como una combinación lineal de alguna entrada sensorial alimentado hacia adelante hacia un saturador. Rosenblatt [55] usó el modelo anterior para clasificar entradas linealmente separables con un algoritmo de aprendizaje que permitía estimar los parámetros libres de la combinación lineal. El algoritmo de Rosenblatt es llamado Perceptrón. El Perceptrón puede clasificar datos en 2 o más clases añadiendo neuronas paralelas al cómputo.

2.2.1. Perceptrón Multicapa

La limitación del Perceptrón es la capacidad de separar solo patrones lineales. Patrones que no son linealmente separables pueden ser aprendidos a través de la composición no lineal de Perceptrones. Esta arquitectura neuronal se llama perceptrón multicapa (MLP del inglés *multilayer perceptron*) [26]. Matemáticamente, se puede describir como:

$$h_{\mathrm{i}} = f_{\mathrm{i}}(W_{\mathrm{i}} \cdot x + b_{\mathrm{i}}) \tag{2.7a}$$

$$g(x) = (h_N \circ ... \circ h_0)(x), \tag{2.7b}$$

donde W_{i} es una matriz de pesos aprendibles de la capa anterior hacia la capa siguiente, b_{i} es un vector de constantes aprendible, f_{i} es una función de activación lineal o no lineal, N es el número de capas, x la entrada a la red neuronal, g es la salida de la red neuronal MLP y $\circ$ es el operador de composición de funciones.

2.2.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN del inglés *convolutional neural networks*) [18] [38] son un tipo de red neuronal artificial diseñada para aprender características robustas a traslaciones. Las CNN fueron primeramente usadas para clasificar datos de 2 dimensiones como imágenes, pero han sido generalizadas para usar datos en 1 (secuencias), 3 ó n dimensiones. Cada capa de la CNN está compuesta de filtros aprendibles, seguido de una función de activación no lineal como en el MLP. El filtrado de las señales de entrada produce que

la red neuronal aprenda características locales que son invariantes a su localización exacta y además comparta pesos respecto al patrón de entrada. El compartir pesos hace que las CNN tengan menos parámetros que una MLP equivalente.

Las CNNs pueden ser utilizadas con datos representados por secuencias. Las secuencias no están limitadas a ser unidimensionales dado que la operación de filtrado y el tamaño de los filtros de cada capa de la CNN se ajusta al número de canales de entrada y salida en una cierta capa.

2.2.3. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNN del inglés *recurrent neural networks*) son un tipo de arquitectura de red neuronal artificial que está diseñada para el procesamiento de secuencias. En su forma basal, este tipo de redes neuronales toma como entrada el estado anterior de la red neuronal y la señal de entrada en el tiempo actual. Este tipo de red es llamado red de Elman [16].

Un problema de esta arquitectura neuronal es que es difícil aprender largas relaciones temporales debido a que el gradiente se desvanece a medida que se avanza en la secuencia. Para sortear el problema de desvanecimiento de gradiente se han diseñado distintos tipos de RNN como *long short-term memory* (LSTM) [20] y *gated recurrent unit* (GRU) [27], [15].

La LSTM utiliza el estado y una memoria externa en conjunto con compuertas que controlan la propagación de señales a medida que se avanza en la secuencia. La compuerta de olvido controla cuánto de los valores antiguos de la celda de memoria se traspasan al tiempo actual. La compuerta de entrada controla cuánto de los valores de entrada son usados en el nuevo valor de la celda de memoria. La compuerta de salida controla cuánto del nuevo valor de la celda de memoria se considera como salida de la red. El uso de las compuertas ayuda a que no se produzca desvanecimiento del gradiente.

El uso de la celda de memoria produce que la red posea una mayor cantidad de parámetros a costa de almacenar cierta información de la secuencia durante el procesamiento. Debido a lo anterior, se diseñó otro tipo de red neuronal que utiliza compuertas al igual que la red LSTM pero que no dispone de una celda de memoria externa. Esa red es denominada GRU. La compuerta de reinicio controla cuánto de los valores de salida anterior se traspasan a la salida actual mientras que la compuerta de actualización controla cuánto de la entrada actual corresponde a la salida actual de la red.

Dado que se busca codificar una serie temporal en un vector de características, por lo general se toma la última salida temporal de la red recurrente como vector, sin embargo, lo anterior limita a que toda la información contenida en la curva de luz debe estar en la última representación. Si bien la trayectoria está de forma implícita en la representación final, también se puede promediar todas las representaciones, de forma de que toda la historia temporal de la curva de luz sea utilizada como vector final, tal como se muestra en la siguiente ecuación:

$$h = \frac{1}{N} \cdot \sum_{\mathrm{i}=1}^{N} h_{\mathrm{i}}, \tag{2.8}$$

donde h_{i} es el vector salida de la red neuronal en la muestra temporal i-ésima y N el largo de la secuencia.

2.2.4. Campo Temporal

El campo temporal corresponde al tamaño efectivo de entradas en el tiempo que influyen en la salida de un cierto conjunto de operaciones matemáticas efectuadas por sobre la entrada. Por ejemplo, si se tiene una secuencia de números muestreados mensualmente, un filtro de media móvil de 3 meses posee un campo temporal de 3 meses, puesto que la salida del filtro depende de los 3 meses de entrada.

Una de las diferencias entre las redes recurrentes y redes convolucionales es el campo temporal al que se atiende. En el caso de la red recurrente, para cada instante de procesamiento se tiene alguna representación propagada hacia el futuro. Sin embargo, las redes convolucionales solo observan una parte arbitraria del pasado que depende del número de capas, tamaño de los filtros y dilatación de la operación de convolución.

El campo temporal de una cierta capa de una red CNN viene dado por la siguiente ecuación:

$$CT = k + (\mathrm{d} - 1) \cdot 2 \tag{2.9}$$

donde CT es el campo temporal, k es el tamaño de los filtros y d es la dilatación. Luego, el campo temporal efectivo en una cierta capa viene dado por:

$$CTE = CT_N + \sum_{\mathrm{i}=1}^{N-1} (CT_{\mathrm{i}} - 1) \tag{2.10}$$

donde CT_N es el campo temporal de la capa actual N y CT_{i} es el campo temporal de las capas anteriores. De las ecuaciones 2.9 y 2.10 es posible ver que el campo temporal efectivo se puede incrementar al aumentar el tamaño de los filtros, aumentando el factor de dilatación o el número de capas de la red convolucional.

2.2.5. Entrenamiento de Redes Neuronales

El entrenamiento de redes neuronales consiste en resolver un problema de optimización. El problema de optimización busca encontrar los parámetros libres del modelo que minimizan la señal de error dada por alguna función de costo. Dentro de la familia de algoritmos de optimización, lo común en el entrenamiento de redes neuronales es utilizar algoritmos basados en descenso de gradiente. Descenso de gradiente es un algoritmo iterativo que utiliza el gradiente de la superficie del error en el espacio de los parámetros para indicar la dirección

en la cual se debe avanzar, repitiéndose la operación hasta llegar a un punto de convergencia. La familia de algoritmos de descenso de gradiente no asegura que el punto de convergencia sea el mínimo global, por lo que en algunos de los casos corresponde a un mínimo local o punto silla de la superficie del error.

El mecanismo de entrenamiento de las distintas capas de una red neuronal se lleva a cabo a partir de la regla de la cadena de la derivada al momento de calcular el gradiente respecto a un cierto grupo de parámetros. Este mecanismo es denominado retropropagación [57] debido a que la señal de actualización de parámetros se propaga desde el error hacia atrás hasta todos los parámetros de la red neuronal.

Existen al menos dos filosofías de entrenamiento, entrenamiento por épocas y entrenamiento *online* [26]. En el entrenamiento por lotes se hace una pasada completa de todos los datos de entrenamiento por la red neuronal y se calcula el error promedio obtenido. Con el error promedio se realiza un paso en la actualización de parámetros utilizando gradiente descendente. La pasada de todo el conjunto de entrenamiento corresponde a una época de entrenamiento.

En el aprendizaje *online* cada muestra del conjunto de entrenamiento es pasada por la red neuronal calculándose el error para esa instancia. Con esa señal de error se realiza un paso en la actualización de parámetros y se repite hasta que todo el conjunto de entrenamiento ha pasado por la red, completando una época de entrenamiento.

Una de las ventajas del método por lotes es que la estimación del gradiente es más robusta respecto al método *online*. Además, el entrenamiento de la red neuronal es paralelizable. Sin embargo, una de las desventajas es que el método por lotes puede quedar estancado en un mínimo local, lo que en el entrenamiento *online* es menos probable dado que la actualización de parámetros es más ruidosa debido a la presentación unitaria de ejemplos. A su vez, el método por lotes necesita de mucha memoria para realizar la actualización de parámetros ya que mantiene la información de todo el conjunto de entrenamiento en la actualización.

Un método híbrido consiste en utilizar mini lotes (*mini-batches*) durante el entrenamiento. Estos mini lotes corresponden a selecciones aleatorias del conjunto de entrenamiento de un cierto tamaño mayor a 1 pero que utiliza el esquema de entrenamiento *online* para la optimización del modelo. El tamaño de los mini lotes es un hiperparámetro del modelo. En este trabajo se utiliza el esquema híbrido.

Existe una gran cantidad de algoritmos de optimización basados en gradiente descendente. El algoritmo de optimización base para el entrenamiento *online* o híbrido corresponde a gradiente descendente estocástico (SGD del inglés *stochastic gradient descent*) [22].

El parámetro más importante en los métodos basado en gradiente descendente es la tasa de aprendizaje. La tasa de aprendizaje es un parámetro que escala el valor del gradiente, el cual generalmente es reducido respecto a su valor original, de forma que la trayectoria en el espacio de parámetros sea más suave que una actualización más pronunciada. Una actualización más pronunciada puede producir que en vez de minimizar la función de costos se comience a subir en la dirección de aumento del error de entrenamiento.

Sin embargo, se ha investigado y diseñado a partir de SGD una serie de algoritmos de optimización que en cierta forma mejoran la trayectoria de descenso de gradiente en el complejo paisaje de la función de costos. Estos algoritmos evitan que el entrenamiento colapse en un mínimo local o punto silla que sea de difícil escape para SGD. En esta tesis se utiliza el algoritmo de estimación de momentos adaptativo (Adam) [35]. El algoritmo Adam corresponde a un algoritmo de la familia de optimización con tasa de aprendizaje adaptiva, el uso de momentum y rescalamiento en el paso de actualización de parámetros. La preferencia por su utilización es que posee cierta robustez a la tasa de aprendizaje elegida.

2.3. Detección de Outliers

Una de las definiciones del término *outlier* es la que sugiere Hawkings [25] "un *outlier* es una observación que se desvía tanto de otras observaciones como para despertar sospechas de que fue generada por un mecanismo diferente". De la definición anterior, se puede desprender que objetos *outliers* pueden contener información valiosa respecto al modo de generación de ciertos datos que se desvía del modo normal de creación o bien corresponder a ruido. Lo anterior calza con la definición del problema a resolver, en donde curvas de luz anómalas son consideradas como las que fueron creadas a partir de un mecanismo de generación distinto al de las curvas ya conocidas. Sin embargo, en este trabajo no se discutirá acerca de los mecanismo de generación de curvas de luz sino que la detección de los objetos novedosos.

Una posible división respecto a los métodos de detección de *outliers* se puede realizar bajo los siguiente tópicos: análisis de valores extremos, modelos de detección de *outliers* y detección en series de tiempo. Este último punto es importante puesto que es el tema principal abordado en esta tesis.

2.3.1. Análisis de valores extremos

El análisis de valores extremos sugiere la búsqueda de *outliers* en zonas correspondiente a las colas de la distribución de probabilidad de los datos. Dado que dichas zonas son de una baja probabilidad, elementos en esas partes podrían ser considerados *outliers*. Uno de los casos más simples de construir es cuando la distribución de datos sigue una distribución gaussiana. En dicho caso es posible construir un puntaje llamado *z-score* que cuantifica la desviación de un punto respecto a la media normalizado según la varianza de la distribución. Se define como:

$$z = \frac{x - \mu}{\sigma}, \tag{2.11}$$

donde z es el puntaje relacionado a la muestra x, μ es el promedio de la distribución y σ la desviación estándar. A modo general, la construcción del puntaje va a depender de la distribución de probabilidad subyacente a los datos.

El *z-score* está también relacionado al test estadístico *z-test*. El *z-test* corresponde a un test donde se intenta determinar si la distribución de la estadística de un test puede ser aproximada a una distribución normal. Por ejemplo, dado el promedio y desviación estándar

de un grupo de datos se puede determinar que tan alejado está el promedio de una muestra del grupo y eventualmente testear la hipótesis nula en donde los promedios son iguales, lo cual va a depender de la medida de la distancia en desviaciones estándar, que corresponde al *z-score*, el cual se calcularía como:

$$z = \frac{\bar{x} - \mu_X}{\frac{\sigma_X}{\sqrt{n_{\bar{x}}}}}, \tag{2.12}$$

donde $\bar{x}$ es el promedio de la muestra, μ_X el promedio del grupo, σ_X la desviación estándar del grupo y $n_{\bar{x}}$ el número de muestras.

2.3.2. Modelos de Detección de Outliers

Modelos Probabilísticos

En los métodos probabilísticos, la distribución de datos es modelada a través de una distribución de probabilidad de forma cerrada, en donde los parámetros de la distribución son encontrados. Una ventaja de usar métodos probabilísticos es que se puede ajustar cualquier tipo de datos mientras exista la distribución adecuada para su mecanismo de generación. Sin embargo, relacionado a lo anterior, es que se obliga a los datos a seguir una cierta distribución la cual puede no estar bien ajustada dada la distribución real de datos. Por otro lado, si se agregan parámetros al ajuste puede que los elementos *outliers* tengan un buen ajuste tal como los elementos normales.

Modelos Lineales

Los métodos lineales utilizan proyecciones de los datos a menores dimensiones a través de un modelo lineal. Una posible forma de construir un puntaje de anomalía es a través del análisis de valores extremos, dado que para un elemento *outlier*, el residuo asociado al ajuste del modelo lineal puede ser usado como puntaje de anomalía, en donde mayores valores indicarían una tendencia más anómala. Otra técnica es el análisis de componentes principales (PCA) [32] donde se utiliza el error de reconstrucción como puntaje de anomalía. Una de las desventajas de los métodos de reducción de dimensionalidad es que las características construidas a partir de combinaciones lineales no necesariamente conservan los atributos originales y puede ser de difícil interpretación.

Modelos Basados en Proximidad

Los métodos basados en proximidad definen a los elementos anómalos como muestras que están aisladas en termino de la similaridad o a partir de funciones de distancia. Los métodos basados en proximidad se pueden dividir en métodos basados en *clustering*, métodos basados en densidad y métodos basados en vecinos más cercanos. En los métodos basados en *clustering* primero se realiza el ajuste de los *clusters* para determinar las zonas de mayor densidad. Luego, se utiliza alguna medida del ajuste a cada *cluster* como puntaje de anomalía. En los métodos basados en densidad, es posible dividir el espacio de los datos y crear un puntaje de anomalía como el número de puntos que tiene una cierta zona del espacio. Los métodos basados en vecinos más cercanos utilizan la distancia a los k vecinos más cercanos. De esa

forma, zonas en donde existe una alta concentración de elementos tienen una distancia a los k vecinos más cercanos menor que la que podría tener un punto aislado del conjunto.

Modelos Basados en Teoría de la Información

En métodos basados en teoría de la información, elementos *outliers* tienden a aumentar el largo de código mínimo que es posible construir a partir de los datos . Están relacionados con los otros métodos en cuánto a que todos utilizan una descripción reducida de los datos para realizar la comparación de los elementos.

Hay veces en que en vez de construir el código se utiliza la entropía de segmentos de datos para medir el grado de desorden. En estos casos, segmentos cuya entropía es mayor tendería a tener objetos *outliers* entre sus elementos. Algunos métodos desarrollados en base a Teoría de la Información se pueden encontrar en [39] [2] [33] [9] [12].

2.3.3. Detección de Outliers en Series de Tiempo

La detección de *outliers* en series de tiempo tiene algunas particularidades. Tal como se indica en [5], el problema de detección de *outliers* se puede dividir de acuerdo a tres características: el tipo de dato de entrada, el tipo de *outlier* a encontrar y la naturaleza del método.

De acuerdo al tipo de dato de entrada, los métodos se diferencian de acuerdo a si pueden manejar series unidimensionales o multidimensionales. El tipo de *outlier* está referido a qué se considera *outlier* dentro de la serie de tiempo. En primer lugar, están los *outliers* de tipo puntuales, es decir, en donde un punto de la serie de tiempo puede o no ser considerado como *outlier*. Puede ser de tipo unidimensional o multidimensional. En segundo lugar, se pueden considerar subsecuencias para determinar si son *outliers*. En este caso, subsecuencias con algún tamaño de ventana son consideradas en el análisis. Nuevamente, las subsecuencias pueden ser unidimensionales o multidimensionales. Finalmente, es posible considerar una serie de tiempo completa en el análisis. En este caso, [5] plantea que solo pueden ser detectadas en el caso multidimensional, cuando alguna de las series individuales (una sola dimensión de la serie multidimensional) se comporta de forma anómala respecto a las demás.

Sin embargo, en el marco bajo el cual se plantea en esta tesis, las series de tiempo a considerar son multidimensionales pero no se indica si la serie de tiempo, magnitud o error de la magnitud es anómala, sino que se marca la serie completa como normal o anómala. Por lo que el enfoque mostrado en [5] no se ajusta a todos los posibles usos y problemas relacionados a la detección de anomalías.

Capítulo 3

Metodología

3.1. Método Propuesto

El algoritmo propuesto para la detección de *outliers* astronómicos (ODT del inglés *Outlier Detection based on Transformations for Astronomical Time Series* pero sin la A de *Astronomical*, T de *Time* y S de *Series*) está basado en aprendizaje de métrica (*metric learning*) [42]. A modo general el algoritmo codifica series de tiempo de largo variable en vectores de tamaño fijo a través de un criterio de similitud. Mediante clústering se determinan grupos de representaciones las cuales calzan en mayor medida con las clases originales para luego determinar un valor numérico para cada representación dependiendo de la distancia al clúster más cercano. Este valor numérico corresponde a un puntaje de *outlier* que mientras más alto, mayor tendencia a ser una curva de luz novedosa.

Si bien el algoritmo original está diseñado para el procesamiento de imágenes, se modificó el modelo para que sea capaz de procesar series de tiempo multidimensionales, muestreadas de forma irregular y de largo variable. Un modelo paramétrico (red neuronal) es usado para codificar un par de curvas de luz a un par de vectores de largo fijo, respectivamente. La distancia entre ellos es optimizada dependiendo de sus etiquetas. Si tienen la misma etiqueta, la distancia entre ellos es minimizada. En cualquier otro caso, la distancia es maximizada. En [42] se propone usar en la optimización ejemplos fuera de la distribución, es decir, provenientes de conjuntos de datos distintos con respecto a los datos de la distribución original (ID del inglés *in-distribution*). En este caso, el paso de optimización se modifica. Si el par corresponde a un ID y un OOD, la distancia es maximizada porque el par proviene de dos distribuciones de datos distintas. Si el par corresponde a dos ejemplos OOD, la distancia también es maximizada porque en caso contrario, los ejemplos OOD se podrían aglomerar como una clase nueva en el espacio latente. Esto no es deseable porque los ejemplos OOD son datos auxiliares para el problema de detección de anomalías, y no pueden ser confundidos con una nueva clase de datos ID.

El criterio de optimización se llama función de costo contrastiva [24]. La función contrastiva busca que objetos similares entre si sean acercados en el espacio de representaciones mientras que objetos disimiles sean alejados. Formalmente se define como:

$$L(x_1, x_2, y, \theta) = \frac{1}{2} \cdot (1 - y) \cdot D_\theta^2 + \frac{1}{2} \cdot y \cdot (\text{máx}(0, m - D_\theta))^2, \tag{3.1}$$

donde $D_\theta = ||f_\theta(x_1) - f_\theta(x_2)||$, f_θ es la función que codifica series de tiempo a vectores de tamaño fijo, que depende de los parámetros θ (definido en la sección 3.1.1), m el margen y x_1, x_2 dos series de tiempo. La etiqueta y toma el valor 0 cuando x_1 y x_2 son de la misma clase y 1 en caso contrario. El margen es la máxima distancia en la que dos objetos son considerados muy lejanos para ser añadidos a la suma de la función de costos ya que el término correspondiente satura a 0.

Una alternativa a considerar es la función de costos de tripletas [61] para aprender representaciones. Si x_a es un objeto ancla, x_p un objeto positivo respecto al ancla (misma clase) y x_n un objeto negativo respecto al ancla (clase distinta), la descripción matemática de la función de costos es la siguiente:

$$L(x_a, x_p, x_n) = D_{\text{ap}}^2 - D_{\text{an}}^2 + m, \tag{3.2}$$

donde m es el margen, D_{ap} es la distancia entre los objetos ancla y positivo y D_{an} es la distancia entre los objetos ancla y negativo. Si $D_{\text{ap}} > D_{\text{an}}$ el objeto negativo es llamado negativo difícil. Cuando $0 < D_{\text{an}} - D_{\text{ap}} < m$ el objeto negativo es llamado semi-difícil, y si $D_{\text{an}} - D_{\text{ap}} > m$ es llamado negativo fácil. El minado de objetos negativos difíciles no es considerado en este trabajo ya que no es el foco principal del método propuesto. Pares y tripletas son siempre seleccionadas de forma aleatoria en cada lote de entrenamiento.

3.1.1. Codificador

El codificador corresponde a una red neuronal que toma la curva de luz normalizada y retorna un vector de tamaño fijo. La arquitectura de la red neuronal se eligió del tipo red recurrente o red convolucional. En ambos casos la red neuronal produce para cada tiempo de la curva de luz una representación. Las representaciones pueden ser promediadas a través del tiempo o considerar solo la última representación para su posterior uso. En ambos casos, se produce una única representación por curva de luz.

Luego, la representación es transformada a través de una transformación lineal a un espacio de igual, mayor o menor dimensionalidad respecto a la representación de la red neuronal. El tamaño va a depender de las combinaciones de hiperparámetros (HP) consideradas. En la Tabla 3.1 se puede ver el espacio de búsqueda de los hiperparámetros considerados, mientras que en la Figura 3.1 muestra la arquitectura general del codificador.

Para mantener una cota sobre el espacio de búsqueda del margen, las representaciones de las curvas de luz se escalan por el factor $\sqrt{\text{d}}$, donde d es la dimensionalidad de la representación. Si se considera dos variables aleatorias gaussianas independientes de dimensionalidad d, $X \sim \mathcal{N}(\mu_x, \Sigma_x)$ e $Y \sim \mathcal{N}(\mu_y, \Sigma_y)$, es posible calcular el valor de esperado de la distancia entre muestras de ambas distribuciones a partir de la estimación de $E[||X - Y||^2]$. Dado que X y Y son independientes, el término $||X - Y||$ es gaussiano de parámetros $\mu = \mu_x - \mu_y$ y $\Sigma = \Sigma_x + \Sigma_y$. Luego, se define $Z = X - Y$ y se estima $E[||Z||^2]$ a continuación:

Tabla 3.1: Hiperparámetros y su espacio de búsqueda.

Parámetro	**Espacio de búsqueda**
Tamaño de lotes (BS)	[32, 64, 128, 256, 512]
Bidireccional[a]	[Falso, Verdadero]
Función de distancia (DF)	[Euclideana, L1]
Dropout	$u \sim U(0{,}2, 0{,}8)$
Representación	[AVG, LAST]
Función de costo (CF)	[CON, TRI]
Tasa de aprendizaje (LR)	10^u, $u \sim U(-4, -2)$
Márgen	U(0, 2)
Arquitectura	[GRU, LSTM, CNN]
Tamaño oculto	[32, 64, 128]
Tamaño de representaciones	[16, 32, 64]
Número de capas ocultas (NHL)	[1, 2]
Decaimiento de pesos (WD)	10^u, $u \sim U(-4, -1)$
Número de filtros[b]	[32, 64, 128]
Tamaño del kernel[b]	[3, 5]

[a]Solo para redes neuronales recurrentes.
[b]Solo para redes neuronales convolucionales.

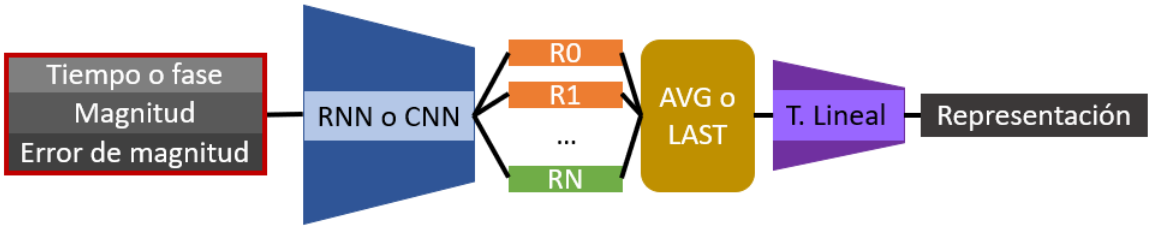

Figura 3.1: Arquitectura general del codificador. R_i indica la representación en el tiempo i hasta la representación final N.

$$E[||Z||^2] = E[Z_1^2] + ... + E[Z_\mathrm{d}^2], \tag{3.3}$$

restando $\sum_{\mathrm{i}=1}^{\mathrm{i}=\mathrm{d}} E[Z_\mathrm{i}]^2$ a ambos lados de la ecuación (3.3) se obtiene que:

$$\begin{aligned} E[||Z||^2] - E[Z_1]^2 + ... - E[Z_\mathrm{d}]^2 &= \mathrm{Var}(Z_1) + ... + \mathrm{Var}(Z_\mathrm{d}) \\ &= \mathrm{tr}(\Sigma), \end{aligned} \tag{3.4}$$

mientras que $\sum_{\mathrm{i}=1}^{\mathrm{i}=\mathrm{d}} E[Z_\mathrm{i}]^2 = \sum_{\mathrm{i}=1}^{\mathrm{i}=\mathrm{d}} \mu_\mathrm{i}^2 = ||\mu||^2$, por lo que:

$$E[||Z||^2] = ||\mu_x - \mu_y||^2 + \mathrm{tr}(\Sigma_x + \Sigma_y), \tag{3.5}$$

bajo los supuestos que $\mu_x = \mu_y = 0$, $\Sigma_x = \Sigma_y = \Lambda$, y que Λ es diagonal con varianza $\lambda_\mathrm{i}^2 = \lambda^2 \,, \forall \lambda_\mathrm{i}^2 \in \mathrm{diag}(\Lambda)$, se llega a que:

$$\begin{aligned} E[||X - Y||^2] &= \mathrm{tr}(2 \cdot \Lambda) \\ &= 2 \cdot \mathrm{d} \cdot \lambda^2 \,. \end{aligned} \tag{3.6}$$

Finalmente, se puede apreciar que el valor de esperado de la distancia cuadrática entre ambas distribuciones crece linealmente con la dimensionalidad, por lo que al normalizar la distancia por d el valor de esperado de la distancia cuadrática es $2 \cdot \lambda^2$. Luego, normalizar la distancia cuadrática por d es equivalente a normalizar los vectores en $\sqrt{\mathrm{d}}$. Si $\lambda = 1$, la cota máxima en el espacio de búsqueda para el margen queda fija en 2, dados los supuestos mencionados.

3.1.2. Asignación de Puntaje

El modelo de asignación de puntaje corresponde a determinar distancias en el espacio generado a partir de la red neuronal, tal como se indica en [42]. Si se tienen C clases dentro del conjunto *inlier*, se estiman C centroides de clase a partir de la representación promedio de cada una de las clases. De igual forma, se estima la matriz de covarianza por clase. Cuando se procesa una curva de luz, se obtiene la representación a través del codificador y luego se calcula la distancia de la representación a cada uno de los centroides estimados. Un puntaje de anomalía viene dado por la distancia al centroide más cercano. La justificación es que el centroide más cercano corresponde a la clase más representativa para una curva de luz desconocida, por lo que si el objeto queda a una distancia muy grande respecto a la clase más cercana, debería considerarse como un candidato a objeto *outlier*. Para calcular la distancia al centroide más cercano existen algunas alternativas. La más directa es utilizar la distancia Euclideana al centroide, la cual es isotrópica en el espacio de características. Sin embargo, no es cierto que la distribución de representaciones de una cierta clase se distribuya de forma isotrópica respecto al centroide, por lo que una medida más flexible es la distancia de

Mahalanobis. Esta última hace la suposición que los elementos relacionados a algún centroide se distribuyen de forma multidimensional con distintas varianzas. Si bien lo anterior puede no ser cierto, es menos restrictiva que la distancia Euclideana. La expresión para un puntaje de anomalía es la siguiente:

$$\text{OS} = \min_{\text{i}} (f(x) - z_{\text{i}})^t \cdot \Sigma_{\text{i}}^{-1} \cdot (f(x) - z_{\text{i}}),\ \text{i} \in [1, C] \tag{3.7}$$

donde Σ_{i} corresponde a la matriz de covarianza de la nube de puntos relacionada al centroide i y C es el número de clases del conjunto de datos.

3.1.3. Preprocesamiento de la Curva de Luz

El preprocesamiento comienza con la normalización de la curva de luz de forma que su promedio sea cero y su variancia unitaria. Matemáticamante queda expresado como:

$$m_{norm}(t) = \frac{m(t) - \mu_t}{\sigma_t} \tag{3.8}$$

$$\sigma^m_{norm}(t) = \frac{\sigma^m(t)}{\sigma_t}, \tag{3.9}$$

donde $m_{norm}(t)$ es la serie de magnitud normalizada, $m(t)$ la serie de magnitud original, μ_t el promedio temporal de la magnitud, σ_t la desviación estándar de la magnitud, $\sigma^m_{norm}(t)$ la serie de error de magnitud normalizada y $\sigma^m(t)$ la serie de error de magnitud original. Al canal de tiempo de muestreo se le resta el tiempo inicial y posteriormente se usa una escala logarítmica dado que algunos tiempos de muestreo pueden presentar un gran orden de magnitud. Dado que la transformación logarítmica retorna $-\infty$ para el tiempo de la primera muestra (ya restada respecto al tiempo inicial t_0), se fuerza a que su valor sea 0. En las siguientes ecuaciones se puede ver el procesamiento del tiempo:

$$\begin{aligned} t &\leftarrow t - t_0 \\ t &\leftarrow \log(t) \\ t_0 &\leftarrow 0\,. \end{aligned} \tag{3.10}$$

3.1.4. Transformaciones

Si bien el modelo es capaz de aprender el espacio de características producido solamente a partir objetos *inliers*, una de las modificaciones propuestas por [42] es utilizar objetos que no correspondan a los elementos *inliers*, generados o adquiridos de alguna otra forma. Estos objetos son llamados *outliers* auxiliares (OA). En [42] los OA son adquiridos a partir de otros conjuntos de datos, es decir, datos fuera de la distribución original. En esta tesis, los OA se generan a partir de transformaciones de series de tiempo en vez de datos externos OOD. La idea es generar series de tiempo transformadas que sirvan como *outliers* plausibles.

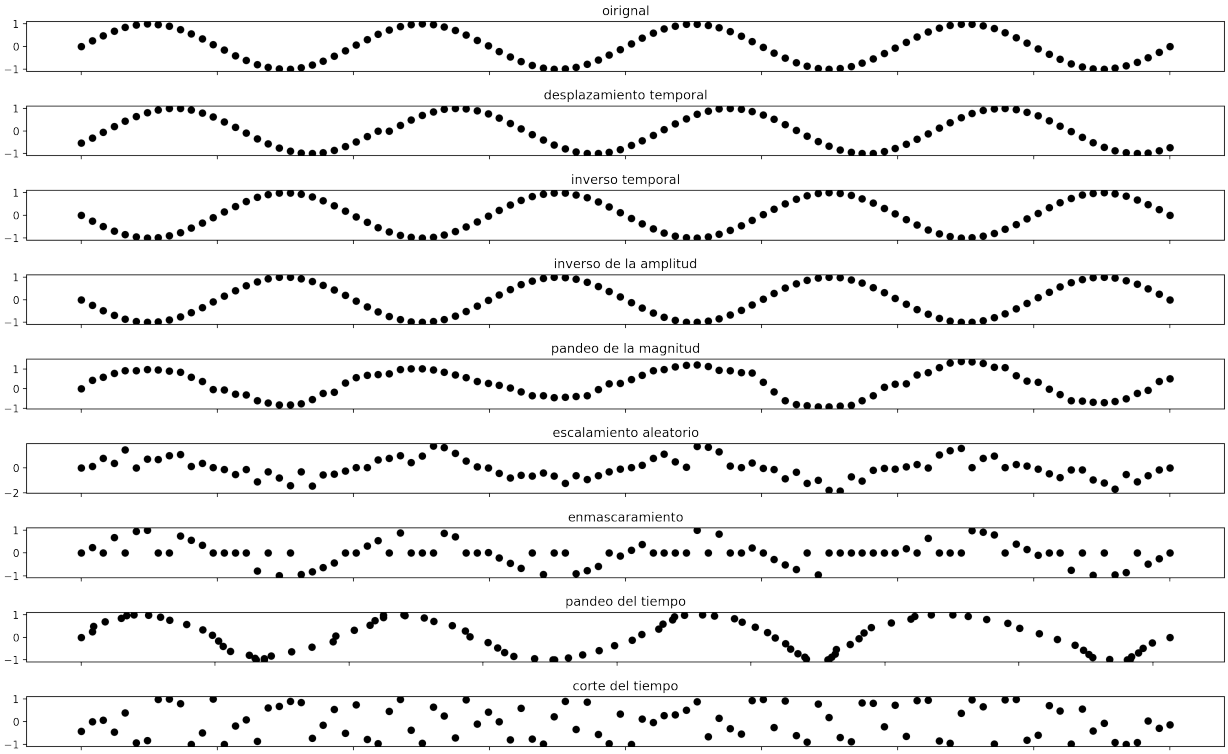

Figura 3.2: Ejemplo de transformaciones aplicadas sobre una señal sinusoidal.

Por un lado este enfoque podría ser útil si un objeto astronómico novedoso es detectado como *inlier* u *outlier* con respecto a los datos de entrenamiento ID. Por otro, este enfoque podría confundir al codificador si las curvas de luz transformadas se ven como datos ID. Para relajar la restricción sobre los datos OOD, se propone utilizar las transformaciones como datos *outliers* auxiliares (AUX) o como aumento de datos (AUG). En el caso AUX se utiliza el paso de optimización modificado, mientras que en el caso AUG se usa el paso original.

Las transformaciones utilizadas se pueden dividir en dos grupos si se considera la conservación del periodo de las señales originales. En la Figura 3.2 se muestra un ejemplo de las transformaciones propuestas aplicadas a una serie de tiempo sinusoidal.

Transformaciones que Conservan la Periodicidad

Dentro del grupo de transformaciones que conservan la periodicidad de la curva de luz original se encuentran: desplazamiento temporal, inverso temporal, inverso de la amplitud, pandeo de la magnitud, escalamiento aleatorio y enmascaramiento. A continuación se detalla cada una de ellas. En la Tabla 3.2 se muestran las distribuciones de los parámetros aleatorios de las transformaciones.

- Desplazamiento temporal: el desplazamiento temporal de una curva de luz consiste en realizar un desplazamiento periódico de ésta, es decir, al desplazar la curva hacia el futuro, los elementos que salen de ella son ingresados al comienzo de la secuencia. Se selecciona una muestra al azar y la curva de luz es desplazada hacia el futuro hasta que la muestra seleccionada llegue a la posición de inicio de la secuencia. Los canales

que son desplazados son el de magnitud y error de magnitud. Matemáticamente queda definida como $\bar{x}(\mathrm{i}) = x(a + \mathrm{i}) \,(\mathrm{mod}\, N)$ donde a es un número aleatorio y N el largo de la curva de luz.

- Inverso temporal: el inverso temporal consiste en invertir temporalmente los elementos de la secuencia original. Solo se invierte los canales de magnitud y error de magnitud. Matemáticamente queda definida como $\bar{x}(\mathrm{i}) = x(-\mathrm{i})$.
- Inverso de la amplitud: corresponde a modificar el canal de magnitud de la curva de luz normalizada invirtiendo todo el canal. Matemáticamente queda definida como $\bar{x}(\mathrm{i}) = -x(\mathrm{i})$.
- Pandeo de la magnitud: el pandeo de la magnitud (del inglés *magnitude warping* viene dado por el reescalamiento de la diferencia de la magnitud de dos muestras contiguas. Matemáticamente viene dada por $\bar{x}(\mathrm{i}) = a \cdot \delta x + x_{\mathrm{i}-1}$ donde x_{i} es la magnitud i-ésima, $\delta x = x_{\mathrm{i}} - x_{\mathrm{i}-1}$ y a es un número aleatorio.
- Escalamiento aleatorio: el escalamiento aleatorio corresponde a reescalar de forma aleatoria la señal de magnitud y error de una serie de tiempo. Cada muestra es reescalada de forma aleatoria. Matemáticamente queda definida como $\bar{x}(\mathrm{i}) = a \cdot x_{\mathrm{i}}$ donde a es un número aleatorio.
- Enmascaramiento: el enmascaramiento consiste en saturar a valor 0 algunas muestras de la serie temporal. Se puede realizar por bloques o a nivel de muestra. En este trabajo, el enmascaramiento se realizó de forma aleatoria por muestra. Matemáticamente se define como $\bar{x}(\mathrm{i}) = a \cdot x_{\mathrm{i}}$ donde a es un valor aleatorio binario.

Tabla 3.2: Transformaciones base y su espacio de parámetros.

Transformación	**Parámetros**
Desplazamiento temporal (TS)	índice del desplazamiento $u \sim \mathcal{U}(0, \text{sequence length})$
Inverso temporal (TI)	Ninguno
Inverso de la amplitud (AI)	Ninguno
Pandeo de la magnitud (MW)	escalamiento de la diferencia de magnitud $u \sim U(0, 2)$
Escalamiento aleatorio (RS)	factor de escalamiento $u \sim U(0, 2)$
Enmascaramiento (ZM)	muestra de valor cero $u > 0{,}5$, $u \sim U(0, 1)$
Pandeo del tiempo (TW)	escalamiento de la diferencia temporal $u \sim U(0, 2)$
Corte del tiempo (TSL)	Ninguno

Transformaciones que no Conservan la Periodicidad

Dentro del grupo de transformaciones que no conservan la periodicidad de la curva de luz original se encuentran: pandeo del tiempo y corte del tiempo. A continuación se detalla cada una de ellas. En la Tabla 3.2 se muestran las distribuciones de los parámetros aleatorios de las transformaciones.

- Pandeo del tiempo: el pandeo del tiempo (del inglés *time warping* corresponde a modificar el intervalo de tiempo entre dos muestras sucesivas. La transformación se lleva a cabo a través de un escalamiento. La modificación puede ser una contracción o elongación del intervalo. El escalamiento se aplica sobre cada uno de los intervalos de tiempo pertenecientes a una curva de luz. El escalamiento se selecciona de forma aleatoria, de forma que se obtienen transformaciones de pandeo del tiempo aleatorias que pueden

contraer o elongar. $\bar{t_{\mathrm{i}}} = \delta t_{\mathrm{i}} \cdot a + t_{\mathrm{i}-1}$ donde t_{i} es el tiempo i-ésimo de la serie temporal, a es un número aleatorio y $\delta t_{\mathrm{i}} = t_{\mathrm{i}} - t_{\mathrm{i}-1}$.

- Corte del tiempo: el corte del tiempo (del inglés *time slicing*) corresponde a desordenar la secuencia a lo largo el tiempo, de forma que la nueva secuencia generada posea los mismos valores que la secuencia anterior pero con un distinto orden temporal, es decir, se desordena la serie de magnitudes y errores respecto al tiempo. Esta transformación es de las más destructivas porque puede romper completamente la forma de un suceso transiente o romper con la periodicidad de un elemento periódico.

3.2. Conjuntos de Datos

En esta tesis se consideran 4 conjuntos de datos de curvas de luz: *All Sky Automated Survey* (ASAS) [51], *Lincoln Near-Earth Asteroid Research* (LINEAR) [62] [49], *All Sky Automated Survey for Supernovae* (ASAS-SN) [36] [31] y *Zwicky Transient Facility* (ZTF), recopilado y curado por el *broker* astronómico ALeRCE [19] [8] [59]. Las Tablas 3.3-3.8 muestran el tamaño relativo de cada clase para darle énfasis al desbalance de clases. El valor 0.00 no es exactamente cero pero está fuera de la precisión mostrada en las tablas.

3.2.1. ASAS-SN

El conjunto de datos ASAS-SN está compuesto por curvas de luz periódicas (88.919) de 9 clases tal como se observa en la Tabla 3.3. Siguiendo la metodología de [66], la clase VAR es dejada como clase *outlier* (OC).

- CEPH: delta cefeida, conocida como cefeida clásica o cefeidas tipo I. Estrellas pulsantes gigantes tipo G, K y supergigantes.
- DSCT: estrellas pulsantes delta scuti, donde los modos radial y no-radial pueden estar presentes.
- ECL: sistema de estrellas binarias en donde el plano de órbita está casi en la línea de vista del observador por lo que se observa que una estrella eclipsa a la otra.
- M: las estrellas variables mira son gigantes rojas con temperaturas de aproximadamente 3000 K, con un tamaño de 200 a 300 veces el radio solar y entre 3000 a 4000 veces más luminosas que el sol.
- ROT: la luminosidad de las estrellas variables rotacionales varían porque tienen un brillo superficial no uniforme y/o formas elipsoidales.
- RRAB: son estrellas variables pulsantes RR Lyrae de tipo a y b, con alrededor de la mitad de la masa del sol. El periodo de pulsación depende de la masa, luminosidad y temperatura. Son conocidas como candelas estándar para la medición de distancias.
- RRCD: son estrellas variables pulsantes RR Lyrae de tipo c y d que se diferencian de las RRAB por tener periodos más cortos y mayor variación sinusoidal y, por ser pulsantes de doble modo, respectivamente.
- SR: son estrellas gigantes o supergigantes semi regulares cuyos periodos van desde 20 a más de 2000 días, acompañadas de irregularidades en sus periodos.
- VAR: son objetos de la base de datos ASAS-SN que no cumplen los criterios de clasificación (estrellas variables de tipo no especificado). Esta categoría homogénea contiene

objetos de distintos tipos de variabilidad que son distintos a los de las clases anteriormente nombradas.

Tabla 3.3: Composición del conjunto de estrellas periódicas de ASAS-SN.

Clase	**Tamaño relativo**
CEPH	0,02
DSCT	0,01
ECL	0,42
M	0,04
ROT	0,03
RRAB	0,14
RRCD	0,05
SR	0,27
VAR	0,02

3.2.2. ZTF

El conjunto ZTF está compuesto de curvas de luz de 22 clases, subdivididas en objetos transientes (ZTF-TRA), estocásticos (ZTF-EST) y periódicos (ZTF-PER). Este conjunto posee dos bandas de observación (g y r), en donde la banda g es la utilizada por este trabajo.

ZTF Transiente

El conjunto transiente está compuesto por 525 curvas de luz de 7 clases. La Tabla 3.4 muestra el tamaño relativo de cada clase.

- SLSN: supernova super lumínica. Clase de explosiones cerca de 10 veces más brillantes que una supernova estándar.
- SNII: supernova tipo II. Colapso de núcleo de estrella roja supergigante.
- SNIIb: supernova tipo 2b, en donde la emisión de hidrógeno se vuelve indetectable.
- SNIIn: supernova tipo IIn. Supernova en medio circunestelar denso.
- SNIa: supernova de tipo Ia. Explosión termonuclear de carbono-oxígeno con una estrella enana blanca.
- SNIbc: supernova tipo Ib o Ic. Colapso de núcleo de estrellas masivas despojada de su envoltura.
- TDE: Evento de disrupción de marea. Disrupción estelar debido a proximidad a agujero negro.

ZTF Estocástico

El conjunto estocástico está compuesto por 16.552 curvas de luz de 7 clases. La Tabla 3.5 muestra el tamaño relativo de cada clase.

- AGN: núcleo galáctico activo. Agujero negro supermasivo con acreción central donde la galaxia domina la luz total. La variabilidad es debida posiblemente a inestabilidades en el disco de acreción.

Tabla 3.4: Composición del conjunto de objetos transientes ZTF-TRA.

Clase	**Tamaño relativo**
SLSN*	0,02
SNII	0,19
SNIIb*	0,01
SNIIn*	0,02
SNIa	0,72
SNIbc*	0,03
TDE*	0,00

*Clases dejadas como *outliers*.

- Blazar: AGN de acreción central con un jet relativista direccionado hacia el observador. La variabilidad está determinada por haces relativistas de sincrotrón y compton inverso.
- CV/Nova: estrella variable cataclísmica (incluye novas clásicas). Sistema binario de transferencia de masa en donde una estrella de secuencia principal transfiere masa a una enana blanca a través del desbordamiento del lóbulo de Roche. En el caso de novas clásicas, explosiones termonucleares ocurren en la superficie acretante de masa de la enana blanca, seguida por un estado inactivo.
- NLAGN: AGN de región de línea estrecha. La región de línea estrecha es la región de gas interestelar ionizado y caliente extendido por el núcleo galáctico activo.
- QSO: objeto cuasi estelar. AGN de acreción central que domina por sobre la galaxia receptora en la luminosidad total. La variabilidad es debida posiblemente a inestabilidades en el disco de acreción.
- NLQSO: QSO de línea de emisión estrecha.
- YSO: objeto estelar joven, en las etapas iniciales de evolución. Se dividen en protoestrella y estrella de presecuencia inicial.

Tabla 3.5: Composición del conjunto de objetos estocásticos ZTF-EST.

Clase	**Tamaño relativo**
AGN	0,12
Blazar	0,04
CV/Nova	0,04
NLAGN*	0,00
QSO	0,75
NLQSO*	0,00
YSO	0,04

*Clases dejadas como *outliers*.

ZTF Periódico

El conjunto periódico está compuesto por 53.514 curvas de luz de 8 clases. La Tabla 3.6 muestra el tamaño relativo de cada clase.

- CEPH: ver definición en sección 3.2.1.

- DSCT: ver definición en sección 3.2.1.
- EA: estrella eclipsante clasificada fenomenológicamente de acuerdo a la forma de la curva de luz como β Persei.
- EB/EW: estrellas eclipsantes clasificadas fenomenológicamente de acuerdo a la forma de la curva de luz como β Lyrae y W Ursea Majoris, respectivamente.
- LPV: estrellas variables de periodo largo. Estrellas gigantes o supergigantes frías.
- Periodic-Other: otros tipos de estrellas variables.
- RRL: RR Lyrae. Ver definición en sección 3.2.1.
- RSCVN: estrella variable RS Canum Venaticorum. Sistemas binarios en donde la estrella primaria es típicamente gigante, caracterizados por curvas de luz semi periódicas debido a cromosferas activas y la presencia de manchas estelares.

Tabla 3.6: Composición del conjunto de objetos periódicos ZTF-PER.

Clase	**Tamaño relativo**
CEPH*	0,01
DSCT*	0,01
EA	0,03
EB/EW	0,28
LPV	0,17
Periodic-Other*	0,00
RRL	0,50
RSCVN*	0,01

*Clases dejadas como *outliers*.

3.2.3. ASAS

El conjunto de datos ASAS está compuesto de 3.000 curvas de luz periódicas de 5 clases. El tamaño relativo de cada clase se muestra en la Tabla 3.7.

- Beta persei: también llamada EA, ver definición en sección 3.2.2.
- CEPH: ver definición en sección 3.2.1.
- RRFM: RR Lyrae (ver definición en sección 3.2.1.) en donde las pulsaciones están moduladas en frecuencia.
- SR: ver definición en sección 3.2.1.
- W ursae majoris: también llamada EW, ver definición en sección 3.2.2.

3.2.4. LINEAR

El conjunto de datos LINEAR está compuesto de 3.690 curvas de luz periódicas de 5 clases. El tamaño relativo de cada clase se puede ver en la Tabla 3.8.

- Beta persei: también llamada EA, ver definición en sección 3.2.2.
- DSCT: ver definición en sección 3.2.1.
- RRFM: ver definición en sección 3.2.3.

Tabla 3.7: Composición del conjunto de datos ASAS.

Clase	**Tamaño relativo**
EA	0,11
CEPH	0,04
RRFM	0,24
SR	0,06
EW	0,54

- RRFO: tipo de RR Lyrae que vibra en el primer modo armónico (del inglés *first overtone*).
- W ursae majoris: ver definición en sección 3.2.3.

Tabla 3.8: Composición del conjunto de datos LINEAR.

Clase	**Tamaño relativo**
EA	0,05
DSCT	0,01
RRFM	0,43
RRFO	0,13
EW	0,38

Para el conjunto ZTF, se seleccionaron las clases minoritarias por familia y se apartaron como conjunto de *outliers*. En las Tablas 3.4, 3.5 y 3.6 están marcados con * las clases *outliers*. Cada partición tiene una cantidad n fija de clases *outliers* (n-vs-resto). Eso significa que las clases marcadas con * quedan fijas como *outliers* y no se cambia su estado durante el entrenamiento y evaluación de los algoritmos, i.e., se usan solo al momento de evaluar. Por otro lado, para los conjuntos ASAS y LINEAR, cada clase es dejada como *outlier* en corridas diferentes (uno-vs-resto). Entonces, los resultados muestran 5 instancias de los conjuntos ASAS y LINEAR, uno por cada clase dejada como *outlier*. Finalmente, en el caso de ASAS-SN, una clase (VAR) es dejada como *outlier* (uno-vs-resto).

3.2.5. División de Datos

Los datos son divididos de dos formas dependiendo del conjunto. Para los conjuntos ASAS-SN, ZTF-TRA, ZTF-EST y ZTF-PER los datos son primeramente divididos en conjuntos *inlier* y *outlier* de acuerdo a las clases dejadas como *outliers* en la sección 3.2. Los datos *inlier* son divididos en tres conjuntos (en paréntesis se muestra el tamaño relativo): entrenamiento (0,8), validación (0,1) y prueba (0,1). El método de división sigue la distribución de clases de los respectivos conjuntos de datos. Finalmente, se mezclan los datos de prueba *inlier* y los datos *outlier* formando un solo conjunto de prueba. De esta forma los datos *outlier* solo están presentes en el conjunto de prueba.

Los conjuntos ASAS y LINEAR son divididos usando el mismo criterio: entrenamiento (0,8), validación (0,1) y prueba (0,1), siguiendo la distribución de clases respectiva. Los datos *outlier* son removidos de los conjuntos de entrenamiento y validación de acuerdo a la clase *outlier* bajo evaluación y mezclados temporalmente con el resto del conjunto de prueba como objetos *outlier*.

3.3. Modelos Base

Se utilizan 7 modelos base a comparar con el modelo propuesto. Se dividen en dos tipos: los basados en características (4) y los basados en redes neuronales (3). Los algoritmos basados en características son usualmente más rápidos de entrenar y evaluar que los métodos basados en redes neuronales. Sin embargo, las características deben ser calculadas cada vez que una nueva muestra es recibida, considerando el escenario de flujo de datos. Por otro lado, los métodos basados en redes neuronales son más lentos de entrenar pero necesitan solo una pasada hacia adelante para dar puntaje a una nueva curva de luz. En algunos casos, basta con utilizar el último estado de la red a medida que nuevos datos llegan al detector.

3.3.1. Modelos base basados en características

En este tipo de modelos es necesaria una etapa previa de extracción de características. La extracción se lleva a cabo con TurboFats [59] el cual es una extensión de FATS [48] desarrollado por el *broker* ALeRCE. De las 153 características disponibles de la librería, solo 60 pueden ser usadas, ya que las curvas de luz de este trabajo corresponden a series de tiempo en una sola banda, de modo que características multibanda quedan descartadas.

Para los 4 modelos, se realiza un muestreo aleatorio de 100 repeticiones del espacio de búsqueda de hiperparámetros. En cada una de esas configuraciones el algoritmo es ajustado y los hiperparámetros seleccionados son los que maximizan el AUCPR.

Isolation Forest

El método *isolation forest* (IF) [40] funciona bajo la premisa de modelar datos anómalos en vez de modelar el conjunto normal. Están basados en dos suposiciones. La primera consiste en que las anomalías son pocas respecto al conjunto normal. La segunda a que las características de los objetos anómalos tienen atributos que difieren del comportamiento normal. Entonces, elementos anómalos deberían ser más fácil de aislar que elementos normales. Para aislar una muestra, se particiona de forma recursiva mediante la elección aleatoria de características y un umbral respectivo hasta que la partición tiene un solo elemento. Una vez que el *isolation tree* (IT) está construido, se toma un *ensemble* de ITs y aquellas muestras que en promedio tienen un largo menor corresponden a datos anómalos. El espacio de búsqueda se muestra en la Tabla 3.9.

Tabla 3.9: Espacio de búsqueda del método *isolation forest*.

Parámetro	**Espacio de búsqueda**
Num. de estimadores	10^u, $u \sim U(0, 2)$
Tamaño de muestra max.	`int`(num. de muestras $\cdot\, u$), $u \sim U(0, 1)$
Contaminación	$U(0, 0{,}5)$
Max. num. de características	`int`(num. of features $\cdot\, u$), $u \sim U(0, 1)$
Bootstrap	[Verdadero, Falso]

Local Outlier Factor

El método *local outlier factor* (LOF) [6] está basado en densidades. Se estima la densidad local de un punto respecto a sus k vecinos más cercanos a través de la distancia de éste hacia

ellos. Luego, se estima la densidad local de cada uno de los k vecinos más cercanos y se compara respecto a la densidad local del punto de consulta. De esa forma se pueden estimar zonas de baja densidad las cuales corresponderían a objetos *outliers*. El espacio de búsqueda se muestra en la Tabla 3.10.

Tabla 3.10: Espacio de búsqueda del método *local outlier factor*.

Parámetro	**Espacio de búsqueda**
Num. de vecinos	10^u, $u \sim U(0,2)$
Algoritmo	[auto, ball tree, kd tree, brute]
Tamaño de hoja	10^u, $u \sim U(0,2)$
Contaminación	$U(0,0{,}5)$

One-class Support Vector Machine

El método de *one-class support vector machine* (OCSVM) [60] tiene un funcionamiento similar a una máquina de soporte vectorial (SVM) en cuánto a que ambos construyen un hiperplano separador. La diferencia radica en que mientras la SVM contruye el hiperplano separador entre dos clases, la OCSVM contruye un hiperplano separador entre el origen y los datos correspondientes a las clases utilizadas. La idea es encontrar el plano que maximice la distancia con respecto al origen. Un parámetro de importancia en la OCSVM es el parámetro ν el cual controla la fracción en el error de entrenamiento y a su vez, la fracción de vectores de soporte utilizados. Al igual que la SVM, es posible utilizar un espacio de características generado a partir de un kernel. El espacio de búsqueda se muestra en la Tabla 3.11.

Tabla 3.11: Espacio de búsqueda del método *one-class support vector machine*.

Parámetro	**Espacio de búsqueda**
Kernel	[linear, poly, rbf, sigmoid]
Gamma	[scale, auto]
Nu	$U(0,1)$
Contracción	[Verdadero, Falso]

Análisis de Componentes Principales

La utilización del análisis de componentes principales (PCA) [32] en la búsqueda de *outliers* sigue la lógica de la reconstrucción del vector de características original. En ese sentido, se busca reducir el espacio original y luego calcular el error de reconstrucción entre la variable original y la predicha. Si se lograra capturar las características del conjunto de entrenamiento normal, sería posible discriminar entre objetos *inlier* y *outlier* a partir del error de reconstrucción, donde un error más grande implicaría una tendencia más grande a ser *outlier*. El espacio de búsqueda se muestra en la Tabla 3.12.

Tabla 3.12: Espacio de búsqueda del método análisis de componentes principales.

Parámetro	**Espacio de búsqueda**
Num. de componentes	`int`(tamaño original $\cdot\, u$), $u \sim U(0,1)$

3.3.2. Modelos Base Basados en Redes Neuronales

Los métodos basados en redes neuronales comparten el mismo preprocesamiento de las curvas de luz con respecto al método propuesto.

Autoencoder

La detección de *outliers* en base a *autoencoders* (AE) [45] sigue el camino opuesto al del IF. En este caso, se busca modelar los datos normales en el conjunto de datos y tomar como *outliers* elementos que se salen de la normalidad. El modelamiento se hace a partir de la reconstrucción de datos. El AE aprende a reconstruir los datos normales los cuales deberían tener un menor error de reconstrucción, mientras que los datos anómalos deberían tener uno mayor. Entonces un puntaje de anomalía se construye a partir del error de reconstrucción de los datos. De igual forma, es posible utilizar las representaciones que genera el autoencoder y utilizar un algoritmo como IF o LOF. Sin embargo, dados los alcances de esta tesis, se utiliza el error de reconstrucción como puntaje de anomalía.

Modelo de Mezcla de Gaussianas

El modelo de mezcla de gaussianas (AEGMM) [66] [72] modela los datos normales a través de la suposición de que las muestras se distribuyen de forma Gaussiana. En particular, se puede considerar que si el conjunto de datos contiene C clases, se puede construir una GMM de C componentes. En este caso, para cada conjunto de datos correspondiente a cada clase, se asume que la generación de datos proviene a partir de una Gaussiana centrada en media muestral y de covarianza correspondiente a la covarianza de la nube de puntos respectiva. A partir de GMM se puede construir una función de energía, la cual entregará un bajo nivel para muestras normales y alta energía para muestras anómalas. La particularidad de este GMM es que utiliza como columna vertebral el AE de la sección anterior. Entonces, los vectores con los cuales se optimiza el GMM corresponden a los del espacio latente generado por el AE. De esa forma, el criterio de optimización del algoritmo no es solo el GMM sino que también clasifica las curvas de luz (entropía cruzada) y las reconstruye (error cuadrático medio).

Clasificador Binario

La idea de utilizar un clasificador binario (CB) viene de aprovechar los *outliers* auxiliares generados a partir de transformaciones y utilizarlos como una clase *outlier* falsa, lo cual está inspirado en [65]. De esa forma, es posible afrontar el problema de detección de *outliers* a través del entrenamiento de un clasificador de dos clases. El clasificador binario viene dado con una arquitectura similar al del codificador del AE, pero en vez de proyectar la representación hacia menor dimensionalidad, la proyección se lleva hacia una salida binaria. El criterio de entrenamiento del clasificador binario es la entropía cruzada.

La Tabla 3.13 muestra los HP de la red neuronal para los tres modelos, los cuales están seleccionados a partir de [45]. Retoques menores fueron hechos a AEGMM para hacer el método estable como contraer el tamaño del espacio latente a 8 y en vez de estimar la covarianza completa solo la diagonal es utilizada en la optimización.

Tabla 3.13: Hiperparámetros para los métodos basados en redes neuronales. El espacio latente del método AEGMM tiene un tamaño de 8 dimensiones debido a problemas de estabilidad.

Parámetro	**Valor**
Arquitectura	GRU
Tamaño de lotes	512
Dropout	0.25
Tasa de aprendizaje	$5 \cdot 10^{-4}$
Tamaño oculto	96
Tamaño latente	64 (8)
Número de capas	2

3.4. Criterios de Evaluación

Todos los métodos son evaluados con el mismo criterio: AUCPR. Esta métrica es un promedio del rendimiento del detector sobre todos los puntos de operación con respecto a las métricas de *precision* y *recall* del modelo. Si se considera el ejemplo de clasificación binaria (ver Figura 3.3), el *precision* se puede entender como la tasa entre verdaderos positivos y las instancias detectadas como positivas, mientras que el *recall* se puede entender como la tasa de verdaderos positivos y todos los elementos cuya condición sea positiva. Entonces, dependiendo del punto de operación, el detector puede tener un alto *precision* a costa de un bajo *recall* (zona A), es decir, el detector detecta poco pero cuando lo hace es certero, como también un bajo *precision* y alto *recall* (zona B), es decir, detecta muchos de los elementos de condición positiva pero equivocándose mucho también. Lo ideal es un detector en un punto de operación de alto *precision* y alto *recall*, o que el *precision* promedio medida como AUCPR sea cercana a 1.

Como los algoritmos bajo prueba retornan puntajes de anomalía en vez de clases predichas de clasificación, el rendimiento del modelo es estimado en cada umbral del puntaje. Como el AUCPR se calcula con el conjunto de prueba, se recomienda [1] medir el AUCPR en un *ensemble* de detectores del mismo tipo, en vez de medir el rendimiento del modelo con una sola configuración de HP. Sin embargo, este método evaluativo es lento en modelos basados en redes neuronales debido al largo tiempo de entrenamiento. Dado que los conjuntos de entrenamiento usados en este trabajo están altamente desbalanceados con respecto al número de ejemplos *outlier*, es recomendado usar la curva *precision-recall* en vez de la curva ROC (acrónimo del inglés *receiver operator characteristic*) [14].

Para hacer todos los resultados comparables entre conjuntos de datos, los puntajes de anomalía son remuestreados para computar el AUCPR con conjuntos balanceados de puntajes de anomalía con respecto a las etiquetas *inlier-outlier* en el conjunto de prueba. De esa forma, el rendimiento aleatorio siempre tiene un valor de 0.5 AUCPR mientras que también se puede obtener una incertidumbre asociada a la métrica al mismo tiempo. El remuestreo se realiza 100 veces por cada evaluación del modelo.

Se realiza un test estadístico de permutación [11] para comparar el método propuesto ODT con la situación base (sin transformaciones) y con los métodos base, basados en características y en redes neuronales. La hipótesis nula (H_0) corresponde a que el rendimiento promedio de

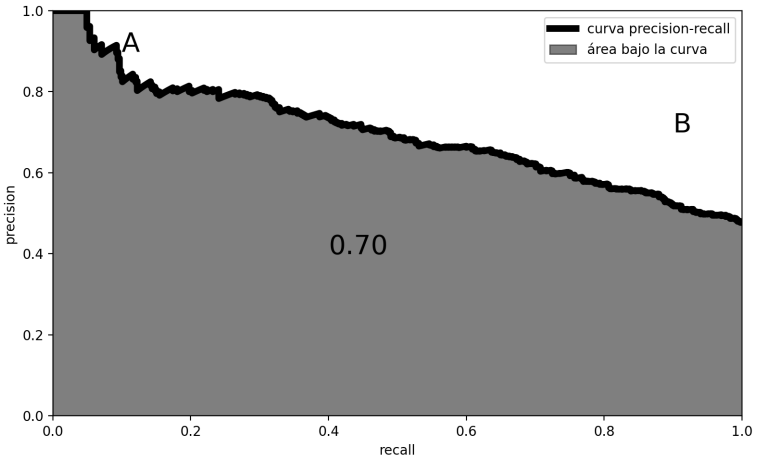

Figura 3.3: Curva *precision-recall* (PR) del ejemplo de clasificación binaria. La zona A indica un alto *precision* y bajo *recall* mientras que la zona B indica un bajo *precision* y alto *recall*. En el punto *recall* = 1 se puede apreciar el rendimiento trivial al detectar todos los objetos como condición positiva. El número dentro del área gris corresponde al área bajo la curva (AUC) de la curva PR.

ODT y los otros modelos son el mismo mientras que la hipótesis alternativa (H_1) es que los rendimientos promedios son distintos. La estadística del test es la diferencia absoluta entre el AUCPR promedio. El número de permutaciones fue fijado en 50.000.

3.5. Métricas Sustitutas

Las métricas sustitutas son utilizadas para estimar la correlación entre ellas y el AUCPR. La idea es computar las métricas con el conjunto de validación y por lo tanto seleccionar un modelo usando ese puntaje. El AUCPR, como siempre, es medido en el conjunto de prueba.

3.5.1. Métrica kNN

La métrica de los k vecinos más cercanos (kNN del inglés *k-nearest neighbors*) es calculada como la tasa de aciertos promedio de los kNN con respecto a una muestra en el espacio latente. La tasa de aciertos está definida como el número de vecinos cercanos (NN) que tiene la misma etiqueta de la muestra bajo evaluación, dividida por k, el número total de NN. Por ejemplo, si una muestra del conjunto de validación es de la clase 0, y los 3 NN tienen clases 0, 1, 1, la tasa de aciertos para esa muestra es 2/3. Luego, la tasa de aciertos es calculada para todas las muestras en el conjunto de validación y promediada para obtener la métrica kNN. Representa la concordancia promedio con respecto a la clase en la vecindad de la muestra.

3.5.2. Coeficiente de Silueta (SC)

El coeficiente de silueta [56] para una muestra se calcula usando la distancia dentro del *cluster* y la distancia media del *cluster* más cercano, para cada muestra. El puntaje de silueta es el promedio de los SC de todas las muestras. El SC estima que tan bien una muestra pertenece a su propio *cluster* con respecto a los *clusters* vecinos.

3.5.3. Puntaje de Calinski-Harabasz (C-H)

El puntaje de Calinski-Harabasz [7] está definido como la tasa entre la dispersión dentro del *cluster* y la dispersión entre *clusters*. Para *clusters* bien definidos se desea una baja dispersión dentro del *cluster* y una alta dispersión entre *clusters*.

3.5.4. Puntaje de Davies-Bouldin (D-B)

El puntaje de Davies-Bouldin [13] está definido como la medida de similaridad promedio de cada *cluster* respecto a su *cluster* más similar, donde la similaridad es la tasa de distancias dentro del *cluster* y entre *clusters*. El índice mide la similitud de un *cluster* con respecto a su *cluster* más cercano. Es deseable un bajo valor para este índice.

El cómputo de SC, C-H y D-B es llevado a cabo con [50]. Debido a la naturaleza del algoritmo y la estimación de centroides para el cálculo del puntaje de anomalía, encajan como candidatas a métricas sustitutas las cuales serán correlacionadas con el AUCPR. La ventaja de estas métricas es que pueden ser calculadas con el conjunto de validación. Finalmente, dependiendo del resultado de la correlación, es posible calcular el frente de Pareto de las soluciones del método entrenado con transformaciones.

3.6. Evaluación de Algoritmos

3.6.1. Selección de Red Neuronal y Parámetros de Entrenamiento

El primer paso es seleccionar la mejor arquitectura de red neuronal para cada conjunto de datos bajo estudio. Los HP bajo evaluación y su respectivo espacio de búsqueda se muestra en la Tabla 3.1. Los HP se pueden dividir en dos familias: relacionados a la arquitectura de red neuronal y criterio de optimización. Los HP relacionados a la arquitectura red neuronal son:

- Bidireccional: el parámetro bidireccional de las redes recurrentes considera si existen capas de procesamiento que operan en la dirección inversa en el tiempo en vez de solo operar hacia adelante.
- *Dropout*: corresponde a un método de regularización en donde se apagan neuronas, i.e., su salida se vuelve cero. Cada neurona tiene una probabilidad de ser apagada en cada iteración de entrenamiento.
- Representación: es la representación generada por la red neuronal. Puede ser la última (LAST) o el promedio a través del tiempo (AVG).
- Arquitectura: es el tipo de red neuronal a utilizar. Pueden ser del tipo recurrente como GRU, LSTM, o convolucionales CNN.
- Tamaño oculto: es el número de neuronas en cada capa de la red neuronal recurrente.
- Tamaño de representaciones: es el tamaño del espacio latente.
- Número de capas ocultas: es el número de capas ocultas de las redes recurrentes o el número de capas convolucionales.
- Número de filtros: es el número de filtros en cada capa de convolución de la red convolucional.

- Tamaño del kernel: es el tamaño de los filtros de cada capa de convolución.

Los HP relacionados a la optimización son:

- Tamaño de lotes: es el número de ejemplos que se utilizan para realizar una iteración del entrenamiento.
- Función de distancia: es la distancia a ser minimizada durante entrenamiento.
- Función de costo: es el criterio de optimización a utilizar.
- Tasa de aprendizaje: corresponde a un escalar que escala el gradiente del error. Dependiendo de su valor, este puede acelerar, frenar o desestabilizar el entrenamiento.
- Margen: corresponde a una cota para la cual la distancia entre dos objetos no es considerada en la optimización dado que se consideran suficientemente lejanos.
- Decaimiento de pesos: corresponde a un método de regularización en donde se añade un término a la función de costos, el cual minimiza el tamaño de los parámetros de acuerdo a la norma Euclidiana, aunque es posible utilizar otros tipos de norma también.

El criterio de optimización sigue la función de costos base, es decir, no se utilizan transformaciones de curvas de luz en esta etapa y el entrenamiento solo se desarrolla con las funciones de costo CON o TRI para los conjuntos ASAS-SN, ZTF-TRA, ZTF-EST, ZTF-PER, ASAS y LINEAR. El criterio de selección es la métrica kNN más alta medida en el conjunto de validación.

3.6.2. Selección de Transformaciones

Ocho transformaciones con sus respectivos parámetros son usadas, tal como se muestra en la Tabla 3.2. Durante el entrenamiento, se usan transformaciones y composiciones de pares de transformaciones. Aparte de eso, las transformaciones son usadas tanto en los modos de aumento datos como de *outlier* auxiliares. En total, se prueban 72 transformaciones diferentes.

Algunos parámetros de las transformaciones son seleccionados de forma aleatoria para fomentar diversidad en las curvas de luz transformadas. Sin embargo, puede ser una elección subóptima ya que algunos valores de parámetros podrían tener un mayor rendimiento. Esta etapa es evaluada con el AUCPR medido en el conjunto de prueba. El cálculo del AUCPR es descrito en la sección 3.4. Se asume que la transformación óptima es la que maximiza el AUCPR. También, el rendimiento promedio del conjunto de transformaciones es calculado para determinar si la mejor configuración es cercana o lejana al desempeño promedio.

3.6.3. Características Agregadas y Ajuste Fino

Si bien el método descrito utiliza la información de forma de las curvas de luz, es posible aumentar las características del espacio latente generado por la red neuronal con el periodo (si está disponible) y la desviación estándar de la serie de magnitud de la curva de luz. La concatenación y mezcla de características está inspirada en el algoritmo [66]. En el caso del periodo, se utiliza el logaritmo en base 10 del valor. No se recomienda la utilización de la magnitud promedio dado que existe cierta correlación entre la distancia al objeto y la magnitud medida en la curva, lo cual podría confundir a los algoritmos si dos objetos de la misma clase tienen magnitudes muy diferentes solo por estar distantes entre si.

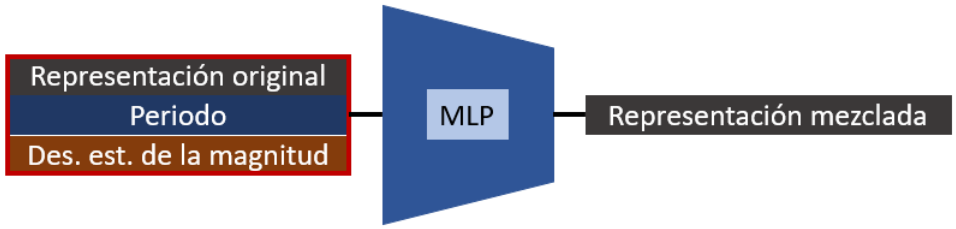

Figura 3.4: Modelo del ajuste fino. Las características concatenadas periodo y desviación estándar de la magnitud pueden estar juntas o por separado dependiendo si el conjunto de datos tiene los periodos disponibles.

El método de optimización de ajuste fino sigue el ya descrito en las secciones 3.1 y 3.1.2 pero se utiliza una red neuronal MLP para mezclar y reagrupar a los vectores. Se busca mezclar características debido a que en el espacio latente se utilizan distancias para medir similitud, por lo que utilizar dos características ajenas al espacio latente, sin mezclar de antemano, puede influir negativamente en el valor de las distancias. Por eso que se genera un espacio latente que fusiona las características agregadas al espacio latente original de las codificaciones de la curva de luz, el cual es entrenado manteniendo los criterios descritos en las secciones 3.1 y 3.1.2. Dado lo anterior, la red neuronal tiene tamaño de entrada $\mathrm{d} + \mathrm{d}_f$ donde d es el tamaño del espacio latente, d_f es el tamaño de las características concatenadas el cual puede ser 1 ó 2, mientras que la salida es de tamaño d. El número de capas ocultas es 2 con tamaños ocultos de valor d y activación tangente hiperbólica. La Figura 3.4 muestra como se generan las nuevas codificaciones de curvas de luz. Al agregar las características antes mencionadas, se busca ver si mejora o no el desempeño en el problema de detección de *outliers*.

Capítulo 4

Resultados y Análisis

4.1. Selección de Modelo

El primer paso es seleccionar los HP del modelo y los parámetros de optimización. La Tabla 4.1 muestra los HP seleccionados. Algunos valores de los parámetros son compartidos entre conjuntos de datos como la función de distancia, el orden de magnitud de la tasa de aprendizaje, arquitectura de red neuronal, tamaño de las capas ocultas, tamaño del espacio latente y el número de capas ocultas. Una explicación posible es que todos los conjuntos de datos probados son curvas de luz. Una situación distinta podría suceder si en vez de curvas de luz, series de tiempo de distinto tipo fuesen usadas como consumo eléctrico o registros de electroencefalograma. Por lo que la capacidad de la red neuronal representada a partir de los hiperparámetros seleccionados es similar ante los conjuntos de datos de curvas de luz.

Tabla 4.1: Mejores combinaciones de hiperparámetros para los conjuntos ASAS-SN, ZTF-TRA, ZTF-STO, ZTF-PER, ASAS y LINEAR basado en el mejor valor de la métrica kNN.

Parámetro	**ASAS-SN**	**ZTF-TRA**	**ZTF-STO**	**ZTF-PER**	**ASAS**	**LINEAR**
Batch size	512	256	512	256	128	128
Bidireccional	Sí	No	Sí	Sí	No	Sí
Función de distancia	EUC	EUC	EUC	EUC	EUC	L1
Dropout	0.39	0.60	0.39	0.33	0.49	0.45
Representación	LAST	AVG	LAST	LAST	AVG	AVG
Función de costos	TRI	TRI	TRI	CON	CON	TRI
Tasa de aprendizaje	$2{,}5 \cdot 10^{-3}$	$9{,}7 \cdot 10^{-3}$	$2{,}5 \cdot 10^{-3}$	$3{,}1 \cdot 10^{-3}$	$1{,}5 \cdot 10^{-3}$	$4{,}1 \cdot 10^{-3}$
Margen	0.12	1.72	0.12	0.69	0.51	1.37
Arquitectura	LSTM	GRU	LSTM	LSTM	LSTM	LSTM
Tamaño oculto	128	128	128	128	128	64
Tamaño latente	32	32	32	32	32	64
Número de capas	2	1	2	2	2	2
Weight decay	$1{,}2 \cdot 10^{-4}$	$1{,}2 \cdot 10^{-2}$	$1{,}4 \cdot 10^{-4}$	$4{,}4 \cdot 10^{-2}$	$1{,}0 \cdot 10^{-3}$	$1{,}5 \cdot 10^{-2}$

La mejor configuración de kNN se encuentra cercana al rango de variabilidad del desempeño promedio de la métrica kNN como se observa en la Tabla 4.2. Esto quiere decir que el

paso de aprendizaje contrastivo es robusto en cierto grado con respecto al espacio de búsqueda, ya que las mejores combinaciones están levemente por sobre el rango de variabilidad del desempeño promedio. Una explicación plausible es que los datos de entrada en entrenamiento son seleccionados de forma aleatoria a medida que el algoritmo compara pares o tripletas de series de tiempo. Como el número de combinaciones posibles es alto, la red neuronal recibe bastante variedad de datos de entrenamiento lo cual ayuda con la convergencia hacia un buen espacio latente.

Tabla 4.2: Mejor valor y promedio de la métrica kNN calculada con los datos de validación para los conjuntos ASAS-SN, ZTF-TRA, ZTF-STO, ZTF-PER, ASAS y LINEAR.

Dataset	**best kNN**	**mean kNN**
ASAS-SN	0,88	$0{,}74 \pm 0{,}15$
ZTF-TRA	0,76	$0{,}66 \pm 0{,}07$
ZTF-STO	0,78	$0{,}74 \pm 0{,}03$
ZTF-PER	0,70	$0{,}59 \pm 0{,}10$
ASAS	0,98	$0{,}87 \pm 0{,}10$
LINEAR	0,94	$0{,}76 \pm 0{,}15$

4.2. Selección de Transformaciones

Las mejores transformaciones son dependientes del conjunto de datos. Las transformaciones pueden estar tomando ventaja de algunos detalles en la forma o tiempos de muestreo que las hacen más útiles en el problema de detección de *outliers*. La Tabla 4.3 muestra que el rendimiento de acuerdo al AUCPR cuando no se aplican transformaciones (AUCPR base) es estadísticamente peor que usando la mejor transformación (entendida como aislada o composición de transformaciones). En algunos casos, es mejor considerar datos transformados como *outlier* auxiliares y en otros como aumento de datos. Las transformaciones de inverso de la amplitud (AI), desplazamiento temporal (TS) y corte del tiempo (TSL) son las transformaciones mayormente seleccionadas ya sea de forma aislada o en composición. El modo de la transformación depende del conjunto de datos sin mostrar una clara ventaja en entre los modos AUX y AUG.

La transformación de los datos originales durante entrenamiento produce un AUCPR más alto la mayoría de las veces, tal como se ve en la Tabla 4.3. En algunos casos (LINEAR RRFO) hay un gran aumento de rendimiento ($+0.31$) mientras que en otros (ZTF-PER y ASAS OC EW) es bastante pequeño ($+0.02$). En todos los casos, los incrementos de desempeño son estadísticamente significativos como se observa en el valor p.

Por otro lado, es posible advertir una diferencia en desempeño entre los conjuntos de datos que utilizan curvas de luz dobladas (ASAS-SN, ASAS y LINEAR) respecto a los que no. La diferencia se puede ver tanto en el caso base como con transformaciones. Una de las razones de la diferencia de desempeño puede estar relacionada en como se representa el tiempo en las curvas de luz. En el caso de curvas periódicas la representación en fase es natural, sin embargo, en los casos transiente y estocástico, en donde es más difícil encontrar una representación natural se obtiene un desempeño peor, al igual que en el caso ZTF-PER cuando se utiliza la representación original en el tiempo.

Tabla 4.3: Resultados del AUCPR para el caso base sin transformar y con la mejor transformación. El valor p es mostrado en la columna $p_{\text{ODT-base}}$. La columna Transf. indica la transformación o el par de transformaciones usado. La columna Modo indica si las transformaciones si las transformaciones se utilizan como aumentación o como *outlier* auxiliar. En la Tabla 3.2 se ve la notación de las transformaciones.

Dataset	**AUCPR base**	**AUCPR ODT**	$\mathbf{p_{ODT\text{-}base}}$	**Transf.**	**Modo**
ASAS-SN	$0{,}56 \pm 0{,}01$	$0{,}63 \pm 0{,}01$	$2 \cdot 10^{-5}$	AI-TS	AUX
ZTF-TRA	$0{,}63 \pm 0{,}02$	$0{,}81 \pm 0{,}01$	$2 \cdot 10^{-5}$	TSL-MW	AUG
ZTF-STO	$0{,}61 \pm 0{,}05$	$0{,}66 \pm 0{,}05$	$3 \cdot 10^{-4}$	TSL	AUG
ZTF-PER	$0{,}57 \pm 0{,}01$	$0{,}59 \pm 0{,}01$	$8 \cdot 10^{-5}$	AI-RS	AUX
ASAS EA	$0{,}93 \pm 0{,}02$	$1{,}00 \pm 0{,}00$	$1 \cdot 10^{-4}$	AI-MW	AUX
ASAS CEPH	$0{,}79 \pm 0{,}10$	$0{,}94 \pm 0{,}05$	$2 \cdot 10^{-4}$	TW-TSL	AUG
ASAS RRFM	$0{,}91 \pm 0{,}02$	$0{,}97 \pm 0{,}01$	$4 \cdot 10^{-5}$	TI-TS	AUG
ASAS SR	$0{,}93 \pm 0{,}05$	$0{,}99 \pm 0{,}02$	$2 \cdot 10^{-4}$	TW	AUX
ASAS EW	$0{,}96 \pm 0{,}00$	$0{,}98 \pm 0{,}00$	$5 \cdot 10^{-5}$	TI-TW	AUX
LINEAR EW	$0{,}82 \pm 0{,}02$	$0{,}97 \pm 0{,}00$	$2 \cdot 10^{-5}$	TS-TSL	AUX
LINEAR EA	$0{,}84 \pm 0{,}06$	$0{,}92 \pm 0{,}03$	$2 \cdot 10^{-5}$	TI-TS	AUG
LINEAR RRFM	$0{,}83 \pm 0{,}01$	$0{,}95 \pm 0{,}01$	$2 \cdot 10^{-5}$	TS-TSL	AUX
LINEAR RRFO	$0{,}63 \pm 0{,}04$	$0{,}94 \pm 0{,}02$	$2 \cdot 10^{-5}$	AI-RS	AUG
LINEAR DSCT	$0{,}78 \pm 0{,}14$	$0{,}87 \pm 0{,}09$	$2 \cdot 10^{-4}$	AI-TW	AUX

4.3. Métodos Base

El método ODT propuesto tiene un mejor rendimiento que los métodos basados en características excepto para los datos de ASAS-SN, ASAS SR, LINEAR EA y LINEAR DSCT, tal como se muestra en la Tabla 4.4, en donde el valor p indica que si existen diferencias significativas entre los resultados. Se puede pensar que las diferencias en desempeño pueden estar relacionadas a las características extraídas desde las curvas de luz. Algunas de ellas podrían no ser óptimas para el problema de detección de *outliers* y se podría enfocar el trabajo en seleccionar el conjunto de características óptimo. Vale la pena mencionar que el método propuesto solo tiene acceso a la forma de la curva de luz por lo que al agregar más características tales como la desviación estándar de la magnitud de la curva de luz o el periodo (mientras esté disponible) podría mejorar el rendimiento obtenido.

La Tabla 4.5 muestra el desempeño del modelo propuesto con respecto a los algoritmos basados en redes neuronales. El modelo propuesto tiene un mejor rendimiento que los detectores basados en redes entrenados en otras tareas de aprendizaje, excepto en los datos de ASAS-SN, ZTF-TRA, ZTF-EST, y LINEAR DSCT en donde si hay una diferencia significativa dado el valor p obtenido. En el caso de ASAS SR no es posible determinar si los puntajes obtenidos para ODT y el clasificador binario son distintos entre si.

El resultado de la Tabla 4.5 confirma que las transformaciones junto a la tarea de aprendizaje contrastivo son útiles para discriminar datos *outlier* respecto a los datos *inlier*, exceptuando los casos en los que ocurre lo contrario sin embargo no son los mayoritarios. Dependiendo del modo de la transformación, los datos artificiales pueden aumentar los datos *inlier* para producir mejores representaciones mientras que en el caso contrario, los datos

Tabla 4.4: Comparación del modelo propuesto (ODT) con respecto a los métodos basados en características en términos del AUCPR. Los p valores están en las columnas $p_{\text{ODT-modelo}}$.

Dataset	$\mathbf{ODT_{AUCPR}}$	$\mathbf{IF_{AUCPR}}$	$\mathbf{p_{ODT\text{-}IF}}$	$\mathbf{LOF_{AUCPR}}$	$\mathbf{p_{ODT\text{-}LOF}}$	$\mathbf{SVM_{AUCPR}}$	$\mathbf{p_{ODT\text{-}SVM}}$	$\mathbf{PCA_{AUCPR}}$	$\mathbf{p_{ODT\text{-}PCA}}$
ASAS-SN	$0{,}63 \pm 0{,}01$	$\mathbf{0{,}68 \pm 0{,}01}$	$2 \cdot 10^{-5}$	$0{,}58 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}66 \pm 0{,}01$	$3 \cdot 10^{-4}$	$\mathbf{0{,}68 \pm 0{,}01}$	$4 \cdot 10^{-5}$
ZTF-TRA	$\mathbf{0{,}81 \pm 0{,}01}$	$0{,}70 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}60 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}58 \pm 0{,}02$	$2 \cdot 10^{-5}$	$0{,}63 \pm 0{,}03$	$2 \cdot 10^{-5}$
ZTF-EST	$\mathbf{0{,}66 \pm 0{,}05}$	$0{,}59 \pm 0{,}04$	$4 \cdot 10^{-4}$	$0{,}60 \pm 0{,}05$	$9 \cdot 10^{-4}$	$0{,}61 \pm 0{,}06$	$2 \cdot 10^{-3}$	$0{,}57 \pm 0{,}05$	$4 \cdot 10^{-4}$
ZTF-PER	$\mathbf{0{,}59 \pm 0{,}01}$	$0{,}51 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}50 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}54 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}48 \pm 0{,}01$	$2 \cdot 10^{-5}$
ASAS EA	$\mathbf{1{,}00 \pm 0{,}00}$	$0{,}92 \pm 0{,}04$	$1 \cdot 10^{-4}$	$0{,}69 \pm 0{,}06$	$2 \cdot 10^{-5}$	$0{,}81 \pm 0{,}07$	$4 \cdot 10^{-5}$	$0{,}80 \pm 0{,}07$	$2 \cdot 10^{-5}$
ASAS CEPH	$\mathbf{0{,}94 \pm 0{,}05}$	$0{,}82 \pm 0{,}10$	$2 \cdot 10^{-4}$	$0{,}75 \pm 0{,}11$	$2 \cdot 10^{-4}$	$0{,}87 \pm 0{,}04$	$5 \cdot 10^{-4}$	$0{,}81 \pm 0{,}11$	$2 \cdot 10^{-4}$
ASAS RRFM	$\mathbf{0{,}97 \pm 0{,}01}$	$0{,}64 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}55 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}81 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}70 \pm 0{,}04$	$2 \cdot 10^{-5}$
ASAS SR	$0{,}99 \pm 0{,}02$	$\mathbf{1{,}00 \pm 0{,}01}$	$1 \cdot 10^{-3}$	$0{,}88 \pm 0{,}05$	$1 \cdot 10^{-4}$	$0{,}96 \pm 0{,}04$	$9 \cdot 10^{-4}$	$0{,}95 \pm 0{,}04$	$2 \cdot 10^{-4}$
ASAS EW	$\mathbf{0{,}98 \pm 0{,}00}$	$0{,}50 \pm 0{,}00$	$2 \cdot 10^{-5}$	$0{,}47 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}92 \pm 0{,}00$	$2 \cdot 10^{-5}$	$0{,}37 \pm 0{,}00$	$2 \cdot 10^{-5}$
LINEAR EW	$\mathbf{0{,}97 \pm 0{,}00}$	$0{,}70 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}47 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}71 \pm 0{,}02$	$2 \cdot 10^{-5}$	$0{,}44 \pm 0{,}01$	$2 \cdot 10^{-5}$
LINEAR EA	$0{,}92 \pm 0{,}03$	$\mathbf{0{,}97 \pm 0{,}04}$	$6 \cdot 10^{-4}$	$0{,}81 \pm 0{,}08$	$1 \cdot 10^{-4}$	$0{,}89 \pm 0{,}06$	$2 \cdot 10^{-3}$	$0{,}88 \pm 0{,}06$	$8 \cdot 10^{-4}$
LINEAR RRFM	$\mathbf{0{,}95 \pm 0{,}01}$	$0{,}78 \pm 0{,}02$	$2 \cdot 10^{-5}$	$0{,}59 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}70 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}74 \pm 0{,}02$	$2 \cdot 10^{-5}$
LINEAR RRFO	$\mathbf{0{,}94 \pm 0{,}02}$	$0{,}55 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}52 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}67 \pm 0{,}05$	$2 \cdot 10^{-5}$	$0{,}51 \pm 0{,}03$	$2 \cdot 10^{-5}$
LINEAR DSCT	$0{,}87 \pm 0{,}09$	$0{,}83 \pm 0{,}14$	$7 \cdot 10^{-3}$	$0{,}58 \pm 0{,}10$	$4 \cdot 10^{-5}$	$\mathbf{0{,}94 \pm 0{,}09}$	$2 \cdot 10^{-3}$	$0{,}68 \pm 0{,}16$	$2 \cdot 10^{-4}$

Tabla 4.5: Comparación del modelo propuesto (ODT) con respecto a métodos basados en redes neuronales en términos del AUCPR. Los valores p se muestran en las columnas $p_{\text{ODT-modelo}}$.

Dataset	$\mathbf{ODT_{AUCPR}}$	$\mathbf{AE_{AUCPR}}$	$\mathbf{p_{ODT\text{-}AE}}$	$\mathbf{AEGMM_{AUCPR}}$	$\mathbf{p_{ODT\text{-}AEGMM}}$	$\mathbf{CB_{AUCPR}}$	$\mathbf{p_{ODT\text{-}CB}}$
ASAS-SN	$0{,}63 \pm 0{,}01$	$0{,}37 \pm 0{,}00$	$2 \cdot 10^{-5}$	$0{,}64 \pm 0{,}01$	$1 \cdot 10^{-3}$	$\mathbf{0{,}69 \pm 0{,}01}$	$2 \cdot 10^{-5}$
ZTF-TRA	$0{,}81 \pm 0{,}01$	$\mathbf{0{,}83 \pm 0{,}02}$	$5 \cdot 10^{-4}$	$0{,}69 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}69 \pm 0{,}02$	$2 \cdot 10^{-5}$
ZTF-EST	$0{,}66 \pm 0{,}05$	$0{,}42 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}42 \pm 0{,}03$	$2 \cdot 10^{-5}$	$\mathbf{0{,}69 \pm 0{,}05}$	$3 \cdot 10^{-3}$
ZTF-PER	$\mathbf{0{,}59 \pm 0{,}01}$	$0{,}47 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}49 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}57 \pm 0{,}01$	$4 \cdot 10^{-5}$
ASAS EA	$\mathbf{1{,}00 \pm 0{,}00}$	$0{,}58 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}46 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}96 \pm 0{,}04$	$3 \cdot 10^{-4}$
ASAS CEPH	$\mathbf{0{,}94 \pm 0{,}05}$	$0{,}32 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}53 \pm 0{,}07$	$2 \cdot 10^{-5}$	$0{,}80 \pm 0{,}11$	$2 \cdot 10^{-4}$
ASAS RRFM	$\mathbf{0{,}97 \pm 0{,}01}$	$0{,}34 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}62 \pm 0{,}02$	$2 \cdot 10^{-5}$	$0{,}81 \pm 0{,}02$	$2 \cdot 10^{-5}$
ASAS SR	$\mathbf{0{,}99 \pm 0{,}02}$	$0{,}93 \pm 0{,}07$	$5 \cdot 10^{-4}$	$0{,}48 \pm 0{,}05$	$2 \cdot 10^{-5}$	$\mathbf{0{,}99 \pm 0{,}02}$	$8 \cdot 10^{-2}$
ASAS EW	$\mathbf{0{,}98 \pm 0{,}00}$	$0{,}55 \pm 0{,}00$	$2 \cdot 10^{-5}$	$0{,}34 \pm 0{,}00$	$2 \cdot 10^{-5}$	$0{,}57 \pm 0{,}01$	$2 \cdot 10^{-5}$
LINEAR EW	$\mathbf{0{,}97 \pm 0{,}00}$	$0{,}53 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}43 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}88 \pm 0{,}03$	$2 \cdot 10^{-5}$
LINEAR EA	$\mathbf{0{,}92 \pm 0{,}03}$	$0{,}68 \pm 0{,}08$	$4 \cdot 10^{-5}$	$0{,}39 \pm 0{,}03$	$2 \cdot 10^{-5}$	$0{,}86 \pm 0{,}07$	$3 \cdot 10^{-4}$
LINEAR RRFM	$\mathbf{0{,}95 \pm 0{,}01}$	$0{,}40 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}63 \pm 0{,}01$	$2 \cdot 10^{-5}$	$0{,}67 \pm 0{,}02$	$2 \cdot 10^{-5}$
LINEAR RRFO	$\mathbf{0{,}94 \pm 0{,}02}$	$0{,}58 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}57 \pm 0{,}04$	$2 \cdot 10^{-5}$	$0{,}80 \pm 0{,}03$	$2 \cdot 10^{-5}$
LINEAR DSCT	$0{,}87 \pm 0{,}09$	$0{,}89 \pm 0{,}09$	$1 \cdot 10^{-2}$	$0{,}53 \pm 0{,}12$	$2 \cdot 10^{-5}$	$\mathbf{0{,}95 \pm 0{,}09}$	$1 \cdot 10^{-3}$

artificiales pueden ayudar a diferenciar datos *inlier* respecto a *outlier* reales en el espacio latente.

En la Tabla 4.5 el clasificador binario, el cual se entrena mediante la clasificación de *inliers* y datos aumentados en modo *outlier auxiliar*, es la mejor opción con los datos de ASAS-SN, ZTF-EST y LINEAR DSCT. Lo anterior pareciera mostrar que las transformaciones no solo son útiles para la tarea de aprendizaje contrastivo sino que también con otros objetivos de optimización como la entropía cruzada en clasificación binaria. Incluso algunos métodos relacionados [28] podrían beneficiarse del uso de las transformaciones en modo AUX como muestras OOD en el término OOD de la función de costo. Además, sería posible entrenar clasificadores de curvas de luz con transformaciones en modo aumento de datos.

4.4. Métricas Sustitutas al AUCPR

El problema principal de la evaluación de algoritmos de detección de *outliers* es la falta de objetos *outlier* disponibles durante el entrenamiento, en un escenario del mundo real. Con ese problema en mente, es posible correlacionar métricas de desempeño estimadas con el conjunto de validación con respecto al AUCPR calculado después de entrenamiento con el conjunto de prueba.

Dado que en el problema de detección de anomalías los elementos *outliers* no son conocidos de antemano, no se puede generar un conjunto el cual pueda ser ingresado a los datos de entrenamiento. Si así fuera, el problema se reduce a uno de clasificación binaria. Entonces, el ajuste de los modelos se debe hacer sin el conocimiento de los elementos anómalos, lo cual entra en conflicto con la evaluación puesto que no se tiene un control exacto del criterio de optimización a seguir para resolver el problema. En otras palabras, el criterio de optimización de entrenamiento puede que no sea el adecuado para resolver el problema de detección de *outliers*. Sin embargo, los elementos anómalos pueden ser definidos y utilizados en el conjunto de prueba para medir el desempeño promedio de los detectores, con el peligro de que bajo esa metodología la elección de modelo se sobre-ajuste al conjunto prueba y perder la capacidad de generalización.

Entonces una hipótesis en el desarrollo de esta tesis es la proposición de métricas sustitutas las cuales se puedan calcular con un conjunto de validación sin el uso de *outliers*. Si una métrica sustituta se correlaciona con el AUCPR medido, hay una clara chance que monitorear esa métrica durante entrenamiento podría ayudar a seleccionar los HP, i.e., usándolo como un indicador sustituto de un buen desempeño con respecto al AUCPR, sin la necesidad de consultar con el conjunto de prueba para corroborarlo.

La Tabla 4.6 muestra la correlación promedio entre 72 casos (un caso por transformación o composición de transformaciones) de diferentes métricas sustitutas con respecto al AUCPR para los distintos conjuntos de datos bajo prueba. Las métricas kNN y el coeficiente de silueta (SC) muestran correlaciones medias y altas con respecto al AUCPR, mientras que el índice Davies-Bouldin (D-B) muestra una correlación inversa. El índice Calinski-Harabasz (C-H) no muestra una clara tendencia de correlación directa o inversa ya que el valor depende altamente en el conjunto de datos. Debería ser posible determinar el frente de Pareto de las métricas kNN, SC y D-B de diferentes configuraciones de HP y elegir los valores óptimos dependiendo

de las restricciones o necesidades del usuario. Y también esto supone que las métricas kNN, SC y D-B son importantes y candidatas a ser monitoreadas durante el entrenamiento.

Tabla 4.6: Correlación entre el AUCPR y las métricas sustitutas calculadas con el conjunto de validación.

Dataset	**kNN**	**SC**	**Índice C-H**	**Índice D-B**
ASAS-SN	0,09	0,06	−0,31	−0,28
ZTF-TRA	0,23	0,37	0,26	−0,20
ZTF-EST	0,05	−0,05	−0,31	0,23
ZTF-PER	0,40	0,19	0,36	−0,27
ASAS EA	0,63	0,49	0,29	−0,47
ASAS CEPH	0,50	0,44	0,03	−0,48
ASAS RRFM	0,06	−0,12	−0,13	−0,06
ASAS SR	0,82	0,74	0,47	−0,75
ASAS EW	0,84	0,76	0,55	−0,81
LINEAR EW	0,17	0,22	0,37	−0,13
LINEAR EA	0,15	0,02	−0,05	−0,11
LINEAR RRFM	0,33	0,44	−0,04	−0,28
LINEAR RRFO	0,44	0,16	−0,22	−0,11
LINEAR DSCT	0,00	0,07	−0,09	0,04

En la Figura 4.1 se muestra que la mejor solución del frente de Pareto, entendida como la que tiene el mayor AUCPR, es subóptima respecto a la mejor de las transformaciones, lo cual es un resultado esperado. En algunos casos la diferencia no es tan marcada sin embargo en otros baja incluso por la línea del desempeño basal. Por otro lado, el peor desempeño de las distintas soluciones del frente de Pareto en general bajan bastante con respecto a la mejor solución de transformaciones y la de la mejor solución del frente. Los peores casos de Pareto presentan menores desempeños que los basales.

Es normal que el desempeño de las distintas soluciones del frente de Pareto difieran en el AUCPR respecto al mejor encontrado con las transformaciones. La idea de usar las métricas sustitutas es no usar el conjunto de prueba para medir el desempeño, pero este resultado deja algunas dudas respecto a la efectividad de esta idea, ya que eventualmente la elección de la solución no debería estar supeditada a la estimación del AUCPR.

Una de las formas de interpretar la correlación de estas cantidades es abstrayendo a qué es lo que pasa con las características y centroides de los vectores. La relación respecto a kNN y AUCPR es que a medida que el algoritmo evoluciona el espacio generado por el entrenamiento contrastivo, los objetos de la misma clase tienden a estar cercanos entre si, que es lo que se busca con el entrenamiento. Una pequeña diferencia es que kNN muestra que al menos en la localidad los objetos mantienen cercanía con los de su misma clase. Luego, sería posible asumir que entre mejor se agrupan estos objetos, tienen una mayor chance de discriminar objetos no pertenecientes a la clase local. Sin embargo, dado el tratamiento de las curvas de luz, la cercanía entre distintos objetos viene dada solo por la forma de la curva de luz, por lo que si dos curvas de luz son altamente similares morfológicamene, pero de distinta clase, el valor kNN bajará y por ende, la relación con el AUCPR que es lo que se puede ver en la Tabla 4.6. Eventualmente puede que haya que agregar más características para que el

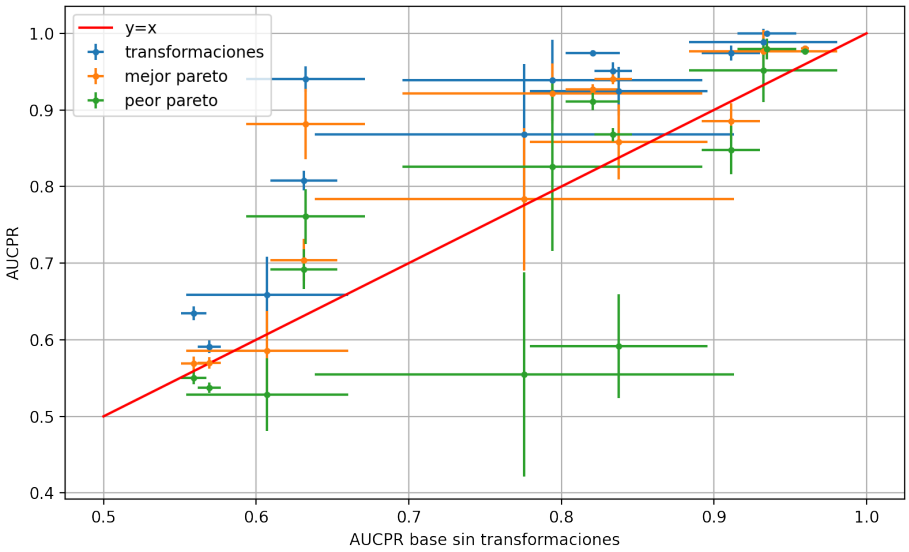

Figura 4.1: Comparación del desempeño al seleccionar el modelo propuesto ODT a través del frente de Pareto de las métricas sustitutas con respecto al modelo base sin transformaciones y al entrenamiento con transformaciones. Las barras corresponden a ± la desviación estándar de cada conjunto de modelos.

algoritmo no se base solo en la parte morfológica.

En la Tabla 4.7 se puede apreciar los AUCPR medidos al seleccionar el valor óptimo de tres métricas, kNN, SC y D-B. Si bien los tres indicadores son más o menos parejos en el número de veces en que son un óptimo, el indicador SC (8) en el que más veces es óptimo respecto a los demás indicadores. Luego lo siguen kNN (7) y D-B (6). Una razón de que la métrica SC sea levemente mejor que las otras es que mide el nivel de pertenencia de una muestra a su *cluster* respecto a los *clusters* vecinos, lo cual guarda relación en la construcción de un puntaje de anomalía, en donde el puntaje es la distancia al centroide más cercano, pero sin considerar la distancia a los centroides vecinos.

4.5. Características Agregadas y Ajuste Fino

El ajuste fino tuvo un resultado que depende principalmente de la característica agregada. Las características logaritmo del periodo y la desviación estándar de la magnitud de la curva de luz se concatenan a la representación en el espacio latente de cada curva de luz (la dimensión de cada espacio latente se ve en la fila *Tamaño latente* de la Tabla 4.1) y luego se comprime esa representación al tamaño original del espacio latente a través de una red neuronal MLP mediante algunas iteraciones más del aprendizaje contrastivo.

En la Tabla 4.8 se puede ver que en el caso en donde se agrega el periodo como característica la diferencia de desempeño es muy volátil. En algunos casos de mejor desempeño anterior, éste empeora y en otros se mantiene. Una explicación al fenómeno es que puede existir periodos de valores similares o sobrepuestos entre las distintas clases, por lo que al agregar esa información al ajuste fino puede acercar objetos que anteriormente estaban más

Tabla 4.7: AUCPR medido en la configuración que maximiza los indicadores kNN, SC o que minimiza D-B. Se destaca está la métrica de mayor valor respecto a los indicadores.

Dataset	**ODT**	**kNN**	**SC**	**Índice D-B**
ASAS-SN	$0{,}63 \pm 0{,}01$	$\mathbf{0{,}57 \pm 0{,}01}$	$\mathbf{0{,}57 \pm 0{,}01}$	$0{,}55 \pm 0{,}01$
ZTF-TRA	$0{,}81 \pm 0{,}01$	$0{,}69 \pm 0{,}03$	$\mathbf{0{,}70 \pm 0{,}03}$	$0{,}69 \pm 0{,}03$
ZTF-EST	$0{,}66 \pm 0{,}05$	$0{,}54 \pm 0{,}04$	$\mathbf{0{,}57 \pm 0{,}05}$	$\mathbf{0{,}57 \pm 0{,}05}$
ZTF-PER	$0{,}59 \pm 0{,}01$	$0{,}54 \pm 0{,}01$	$\mathbf{0{,}56 \pm 0{,}01}$	$0{,}55 \pm 0{,}01$
ASAS EA	$1{,}00 \pm 0{,}00$	$\mathbf{0{,}98 \pm 0{,}01}$	$\mathbf{0{,}98 \pm 0{,}01}$	$\mathbf{0{,}98 \pm 0{,}01}$
ASAS CEPH	$0{,}94 \pm 0{,}05$	$0{,}83 \pm 0{,}11$	$0{,}84 \pm 0{,}11$	$\mathbf{0{,}92 \pm 0{,}04}$
ASAS RRFM	$0{,}97 \pm 0{,}01$	$\mathbf{0{,}89 \pm 0{,}02}$	$0{,}85 \pm 0{,}03$	$0{,}85 \pm 0{,}04$
ASAS SR	$0{,}99 \pm 0{,}02$	$\mathbf{0{,}98 \pm 0{,}03}$	$0{,}97 \pm 0{,}04$	$\mathbf{0{,}98 \pm 0{,}03}$
ASAS EW	$0{,}98 \pm 0{,}00$	$\mathbf{0{,}98 \pm 0{,}00}$	$\mathbf{0{,}98 \pm 0{,}00}$	$\mathbf{0{,}98 \pm 0{,}00}$
LINEAR EW	$0{,}97 \pm 0{,}00$	$0{,}91 \pm 0{,}01$	$\mathbf{0{,}93 \pm 0{,}01}$	$0{,}91 \pm 0{,}01$
LINEAR EA	$0{,}92 \pm 0{,}03$	$\mathbf{0{,}86 \pm 0{,}05}$	$0{,}64 \pm 0{,}08$	$0{,}59 \pm 0{,}07$
LINEAR RRFM	$0{,}95 \pm 0{,}01$	$0{,}87 \pm 0{,}01$	$\mathbf{0{,}90 \pm 0{,}01}$	$0{,}88 \pm 0{,}02$
LINEAR RRFO	$0{,}94 \pm 0{,}02$	$\mathbf{0{,}80 \pm 0{,}04}$	$0{,}76 \pm 0{,}04$	$0{,}76 \pm 0{,}04$
LINEAR DSCT	$0{,}87 \pm 0{,}09$	$0{,}65 \pm 0{,}13$	$0{,}71 \pm 0{,}17$	$\mathbf{0{,}78 \pm 0{,}09}$

separados.

En las Figuras 4.2 y 4.3 se muestra el histograma de la distribución del logaritmo del periodo para los conjuntos ASAS y LINEAR respectivamente. Se puede considerar que en el caso donde los histogramas están sobrepuestos, como ocurre entre las clases EW y RRFM para ASAS y, EW, RRFO, RRFM y EA para LINEAR, agregar la información del periodo puede ser contraproducente en la tarea de detección de *outliers* debido a que comparten rangos en los casos nombrados. Al contrastar esa información con el resultado de la Tabla 4.8 se verifica que para los casos EW de ASAS y, EW y EA de LINEAR el desempeño baja respecto a la situación sin agregar el periodo. Por otro lado, la idea anterior se refuta para los casos RRFM de ASAS y, RRFO y RRFM de LINEAR ya que el desempeño aumenta. Por otra parte, el caso SR de ASAS debería aumentar el desempeño ya que el rango del periodo difiere bastante respecto a las otras clases, pero que sin embargo baja al utilizarlo en el ajuste. El desempeño sí aumenta en el caso DSCT de LINEAR cuyo rango de periodo también es claramente diferenciado respecto al resto de clases.

Además, en las Figuras 4.2 y 4.3 son visibles algunas inconsistencias en la estimación de los periodos para las clases EW y EA de ambos conjuntos de datos. En el caso de EW se puede apreciar que el histograma de LINEAR está ligeramente desplazado hacia la derecha respecto al de ASAS mientras que para EA el histograma de LINEAR está ligeramente desplazado hacia la izquierda respecto a ASAS. Dada la información de los párrafos presente y actual, se propone como trabajo futuro profundizar en la estimación, concatenación y mezcla de la información del periodo dentro del método ODT propuesto, ya que el periodo es una variable valiosa en problemas de clasificación de curvas de luz y debería mantener cierta preponderancia en el problema de detección de *outliers*.

La desviación estándar de la serie de la magnitud presenta en general por si sola (excepto en ZTF-TRA, ZTF-EST y ZTF-PER) un comportamiento en el cual aumenta el desempeño

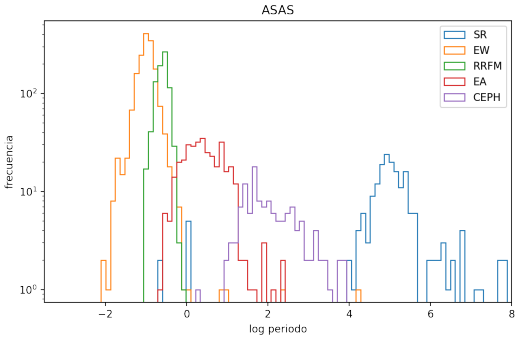

Figura 4.2: Histograma del logaritmo del periodo para el conjunto ASAS separado por clase.

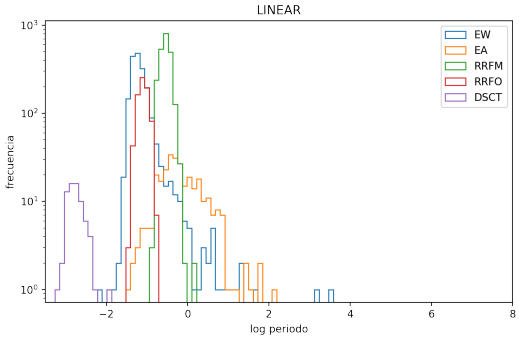

Figura 4.3: Histograma del logaritmo del periodo para el conjunto LINEAR separado por clase.

en términos del AUCPR al ser agregada como característica extra, en donde agregar esa información mejora el desempeño al detectar *outliers*. Cabe destacar que en los casos de ZTF la diferencia es prácticamente 0.

Al agregar ambas características a las representaciones de las curvas de luz en el espacio latente, la variabilidad del resultado con características periódicas se propaga, es decir, el resultado depende del conjunto de datos bajo prueba y en general baja el desempeño en términos del AUCPR respecto a utilizar solo la desviación estándar.

Tabla 4.8: Diferencia de desempeño en términos de AUCPR promedio entre el resultado del modelo ODT mostrado en la columna ODT proveniente de la Tabla 4.4 y el desempeño del mismo modelo ODT pero ajustado de forma fina agregando como característica extra a las representaciones latentes de cada curva de luz el logaritmo del periodo (columna Periodo), la desviación estándar de la magnitud de la curva de luz (columna Desviación Estándar) o ambas (columna Per. y Des. Est.) de forma respectiva a cada curva de luz.

Dataset	**ODT**	**Periodo**	**Desviación Estándar**	**Per. y Des. Est.**
ASAS-SN	$0{,}63 \pm 0{,}01$	$0{,}00 \pm 0{,}00$	$0{,}14 \pm 0{,}00$	$-0{,}01 \pm 0{,}00$
ZTF-TRA	$0{,}81 \pm 0{,}01$	-	$-0{,}02 \pm 0{,}00$	-
ZTF-EST	$0{,}66 \pm 0{,}05$	-	$0{,}00 \pm 0{,}01$	-
ZTF-PER	$0{,}59 \pm 0{,}01$	-	$0{,}00 \pm 0{,}00$	-
ASAS EA	$1{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$	$0{,}02 \pm 0{,}00$	$0{,}00 \pm 0{,}01$
ASAS CEPH	$0{,}94 \pm 0{,}05$	$0{,}10 \pm 0{,}01$	$0{,}17 \pm 0{,}01$	$0{,}09 \pm 0{,}01$
ASAS RRFM	$0{,}97 \pm 0{,}01$	$0{,}03 \pm 0{,}01$	$0{,}09 \pm 0{,}01$	$0{,}05 \pm 0{,}01$
ASAS SR	$0{,}99 \pm 0{,}02$	$-0{,}02 \pm 0{,}01$	$0{,}09 \pm 0{,}01$	$-0{,}02 \pm 0{,}01$
ASAS EW	$0{,}98 \pm 0{,}00$	$-0{,}04 \pm 0{,}00$	$0{,}11 \pm 0{,}00$	$-0{,}06 \pm 0{,}00$
LINEAR EW	$0{,}97 \pm 0{,}00$	$-0{,}14 \pm 0{,}00$	$0{,}06 \pm 0{,}00$	$-0{,}15 \pm 0{,}00$
LINEAR EA	$0{,}92 \pm 0{,}03$	$-0{,}13 \pm 0{,}01$	$0{,}21 \pm 0{,}01$	$-0{,}12 \pm 0{,}01$
LINEAR RRFM	$0{,}95 \pm 0{,}01$	$0{,}07 \pm 0{,}00$	$0{,}14 \pm 0{,}00$	$0{,}05 \pm 0{,}00$
LINEAR RRFO	$0{,}94 \pm 0{,}02$	$0{,}05 \pm 0{,}01$	$0{,}11 \pm 0{,}01$	$0{,}05 \pm 0{,}01$
LINEAR DSCT	$0{,}87 \pm 0{,}09$	$0{,}11 \pm 0{,}02$	$0{,}31 \pm 0{,}02$	$0{,}11 \pm 0{,}02$

Capítulo 5

Conclusión

El método propuesto es una mejora significativa para el problema de detección de curvas de luz que sean *outliers*. Las curvas de luz son primeramente codificadas a un espacio latente donde la distancia de pares o tripletas de curvas de luz es optimizada. El modelo propuesto supera, excepto en algunos casos, a otros métodos basados en características de curvas de luz y redes neuronales. El método usa transformaciones de series de tiempo en dos modos de operación, aumento de datos o *outlier* auxiliares.

El método ODT funciona mejor cuando la serie de tiempo es procesada previamente a ser utilizada por el algoritmo. Por ejemplo, en los conjuntos ASAS y LINEAR, parte del procesamiento previo descrito por los autores de los trabajos basados en esos datos contempla el suavizado de las series y remoción de puntos específicos de la curva de luz que puedan ser *outliers*. El desempeño en esos conjuntos fue generalmente mayor que con ZTF, en donde el preprocesamiento es mínimo. En la misma línea, el método falla mayormente con series de tiempo estocásticas y transientes mientras que las curvas de luz periódicas presentan un mejor desempeño. El método también puede fallar cuando curvas de luz *outliers* presenten similitud con los datos de entrenamiento, debido a la naturaleza del entrenamiento realizado.

El algoritmo se podría desplegar de distintas maneras en un ambiente productivo. En una primera instancia se puede diferenciar el procesamiento de datos respecto al uso del tiempo y almacenamiento. Si el tiempo de procesamiento no es un problema, se pueden procesar todos los puntos de la curva de luz cada vez que llega una nueva muestra y actualizar un puntaje de *outlier* de cada objeto. Por otro lado, si el almacenamiento no es un problema, el sistema puede almacenar la última muestra de la curva de luz y el estado interno de la red neuronal recurrente para generar un puntaje de anomalía cuando se presenta una nueva muestra, por cada objeto en el conjunto de interés. Lo anterior también se puede complementar tomando en cuenta que el procesamiento puede ser realizado por lotes, es decir, procesando un conjunto de curvas de luz al mismo tiempo o a través del procesamiento de a una serie a la vez.

Los factores más importantes están relacionados al preprocesamiento de las series de tiempo. Tal como se mencionó anteriormente, los conjuntos de datos que presentan un preprocesamiento adecuado pueden obtener un desempeño superior a las menos procesadas. El procesamiento no necesariamente es solo suavizar las series de tiempo, sino que también pue-

de ser el uso de una representación distinta como la fase de la señal periódica. Un factor importante también es el uso de características externas tal como se mostró en la etapa de ajuste fino. El uso de la desviación estándar de la serie de tiempo fue positivo debido a que esa información es removida al momento de normalizar la curva de luz. En el caso del periodo, su uso depende de la estimación que se realice. Si la estimación es incongruente con los datos, por ejemplo, que por algún error de estimación o metodológico dos clases que posean periodos distintos su estimación sea similar, puede generar confusión en el algoritmo, tal como se mostró con las clases EW y EA de ASAS y LINEAR.

Se propusieron varias métricas sustitutas: la tasa de aciertos de los k vecinos más cercanos de cada curva de luz en el espacio latente (métrica kNN), el coeficiente de silueta, el índice de Calinski-Harabasz y el índice Davies-Bouldin para evitar usar el conjunto de prueba (AUCPR) para la selección de modelo. Las métricas sustitutas se miden en el conjunto de validación donde no hay *outliers* reales. Entre estas, la métrica kNN mostró mayor correlación con respecto al AUCPR, siendo cercana coeficiente de silueta. Al elegir la transformación óptima de acuerdo al frente de Pareto de todas las métricas antes mencionadas se tiene que el desempeño baja respecto al óptimo, sin embargo es esperable debido a que la mejor solución de Pareto, estimada mediante el conjunto de validación puede ser igual o peor que la solución que maximiza el AUCPR con el conjunto de prueba. Dado que la mejor y peor soluciones de Pareto presentan gran variabilidad, no es conveniente por el momento utilizar ese criterio como selección. Si se consideran las métricas que tienen una marcada correlación directa o inversa con respecto al AUCPR, esto es, kNN, coeficiente de silueta y Davies-Bouldin, las que mantienen un mayor AUCPR con respecto al óptimo para la métrica corresponden al coeficiente de silueta y la métrica kNN, por lo que se podrían considerar las dos últimas para la estimación del frente de Pareto.

Al agregar características adicionales a los vectores de las curvas de luz codificadas y ajustar de forma fina la representaciones con aprendizaje contrastivo, se tiene que en general el periodo confunde al algoritmo mientras que la desviación estándar de la curva permite mejorar las representaciones de acuerdo al desempeño del AUCPR medido.

Se corroboran las hipótesis del trabajo: es posible aprender un puntaje de anomalía a partir de la curva de luz sin el cálculo previo de características; hay transformaciones que presentan mejor desempeño que otras en el problema de detección de *outliers* y es posible encontrarlas al evaluar el rendimiento; además existen métricas sustitutas que guardan correlación directa o inversa con el AUCPR pero que al elegir un modelo a través del frente de Pareto de los indicadores, el desempeño puede variar demasiado y por lo tanto, el método sustituto no es completamente fiable de utilizar. Lo anterior no quita que el método sustituto pueda ser sintonizable.

5.1. Trabajo Futuro

El método propuesto tiene suficiente espacio para mejoras en distintas partes y etapas. Los parámetros de las transformaciones son seleccionados en estos momentos de forma aleatoria en cada lote de entrenamiento. Es posible seleccionar los parámetros óptimos para cada transformación y tratarlos como nuevos parámetros a ser buscados. El puntaje de anomalía asume Gaussianidad de los *clusters* de cada clase debido a la estimación de centroides. Un

acercamiento menos restrictivo podría ser estimar la distribución de datos en el espacio latente a través de métodos como estimación de densidad con *kernel* y usando la verosimilitud como puntaje de anomalía.

Al nivel de curva de luz, la representación es una variable importante a considerar. Por esa razón, métodos como [45] y [66] usan la representación en fase en vez del tiempo natural. Representaciones del tiempo como codificaciones seno-coseno podrían ser investigadas para clasificación de curvas de luz y detección de *outliers*. Es de suma importancia tener una buena estimación del periodo de las curvas de luz periódicas al momento de agregar esa información a las representaciones latentes. Se propone como trabajo futuro profundizar en la estimación, concatenación y mezcla de la información del periodo dentro del método ODT propuesto. Otros tipos de arquitecturas de redes neuronales para secuencias se podrían utilizar además de las aquí usadas, como los modelos de atención y *transformers* [67].

Finalmente, es posible aumentar el uso de las métricas sustitutas a través del ajuste fino de lo modelos agregando pocas curvas de luz *outliers*. Sin embargo, una metodología de esas características escapa al acercamiento no-supervisado desarrollado en esta tesis.

Bibliografía

[1] C. C. Aggarwal. *Outlier analysis*. Springer, 2nd edition, 2017.

[2] S. Ando. Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 13–22. IEEE, 2007.

[3] L. Beggel, B. X. Kausler, M. Schiegg, M. Pfeiffer, and B. Bischl. Time series anomaly detection based on shapelet learning. *Computational Statistics*, 34(3):945–976, 2019.

[4] S.-E. Benkabou, K. Benabdeslem, and B. Canitia. Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge and Information Systems*, pages 463–486, 2018.

[5] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3), apr 2021.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.

[7] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[8] R. Carrasco-Davis, E. Reyes, C. Valenzuela, F. Förster, P A. Estévez, G. Pignata, F. E. Bauer, I. Reyes, P. Sánchez-Sáez, and G. Cabrera-Vives et al. Alert classification for the ALeRCE broker system: The real-time stamp classifier. *The Astronomical Journal*, 162(6):231, 2021.

[9] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In *VLDB*, volume 98, pages 606–617, 1998.

[10] F. R. Chromey. *To measure the sky: an introduction to observational astronomy*. Cambridge University Press, 2nd edition, 2016.

[11] G. W. Cobb. *Introduction to design and analysis of experiments*. Key College, 1998.

[12] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2006.

[13] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[14] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 233–240, New York, NY, USA, 2006. Association for Computing Machinery.

[15] R. Dey and F. M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[16] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[17] O. Fabius and J. R. Van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[18] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):826–834, 1983.

[19] F. Förster, G. Cabrera-Vives, E. Castillo-Navarrete, P. A. Estévez, P. Sánchez-Sáez, J. Arredondo, F. E. Bauer, R. Carrasco-Davis, M. Catelan, and F. Elorrieta et al. The automatic learning for the rapid classification of events (ALeRCE) alert broker. *The Astronomical Journal*, 161(5):242, apr 2021.

[20] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[21] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. *Advances in neural information processing systems*, 31, 2018.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[23] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

[24] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[25] D. Hawkings. *Identification of outliers*. Chapman and Hall, 1980.

[26] Simon Haykin. *Neural networks and learning machines*. Pearson Education India, 3rd edition, 2010.

[27] J. C. Heck and F. M. Salem. Simplified minimal gated unit variations for recurrent neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1593–1596. IEEE, 2017.

[28] D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[29] R. J. Hyndman, E. Wang, and N. Laptev. Large-scale unusual time series detection. In *2015 IEEE international conference on data mining workshop (ICDMW)*, pages 1616–1619. IEEE, 2015.

[30] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, and J. Andrew et al. Lsst: from science drivers to reference design and anticipated data products. *The Astrophysical Journal*, 873(2):111, 2019.

[31] T. Jayasinghe, C. S. Kochanek, K. Z. Stanek, B. J. Shappee, T. W.-S. Holoien, T. A. Thompson, J. L. Prieto, S. Dong, M. Pawlak, and J. V. Shields et al. The ASAS-SN catalogue of variable stars I: The Serendipitous Survey. *Monthly Notices of the Royal Astronomical Society*, 477(3):3145–3163, 04 2018.

[32] I. T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[33] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215, 2004.

[34] T. Kieu, B. Yang, C. Guo, and C. S. Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *28th international joint conference on artificial intelligence*, 2019.

[35] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] C. Kochanek, B. J. Shappee, K. Z. Stanek, T. W.-S. Holoien, T. A. Thompson, J.-L. Prieto, S. Dong, J. V. Shields, D. Will, and C. Britt et al. The all-sky automated survey for supernovae (asas-sn) light curve server v1.0. *Publications of the Astronomical Society of the Pacific*, 129, 06 2017.

[37] J. A. Lara, D. Lizcano, V. Rampérez, and J. Soriano. A method for outlier detection based on cluster analysis and visual expert criteria. *Expert Systems*, 37(5):e12473, 2020.

[38] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[39] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 130–143. IEEE, 2000.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[41] K. Malanchev, A. Volnova, M. Kornilov, M. Pruzhinskaya, E. Ishida, F. Mondon, and

V. Korolev. Use of machine learning for anomaly detection problem in large astronomical databases. In *Data Analytics and Management in Data Intensive Domains: I In-ternational Conference DADID/RCDL'2019 (October 15–18, 2019, Kazan, Russia): Conference Proceedings. Edited b Alexander Elizarov, Boris Novikov, Sergey Stupnikov.– Kazan: Kazan Federal University, 2019.–496* ., page 238, 2019.

[42] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez. Metric learning for novelty and anomaly detection. In *British Machine Vision Conference (BMVC)*, 2018.

[43] F. J. Masci, R. R. Laher, B. Rusholme, D. L. Shupe, S. Groom, J. Surace, E. Jackson, S. Monkewitz, R. Beck, and D. Flynn et al. The zwicky transient facility: data processing, products, and archive. *Publications of the Astronomical Society of the Pacific*, 131(995):018003, dec 2018.

[44] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[45] B. Naul, J. Bloom, F. Perez, and S. van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2, 02 2018.

[46] I. Nun, K. Pichara, P. Protopapas, and D.-W. Kim. Supervised detection of anomalous light curves in massive astronomical catalogs. *The Astrophysical Journal*, 793(1):23, 2014.

[47] I. Nun, P. Protopapas, B. Sim, and W. Chen. Ensemble learning method for outlier detection and its application to astronomical light curves. *The Astronomical Journal*, 152(3):71, aug 2016.

[48] I. Nun, P. Protopapas, B. Sim, M. Zhu, R. Dave, N. Castro, and K. Pichara. FATS: feature analysis for time series. *arXiv e-prints*, page arXiv:1506.00010, May 2015.

[49] L. Palaversa, Ž. Ivezić, L. Eyer, D. Ruždjak, D. Sudar, M. Galin, A. Kroflin, M. Mesarić, P. Munk, and D. Vrbanec et al. Exploring the variable sky with linear. iii. classification of periodic light curves. *The Astronomical Journal*, 146(4):101, sep 2013.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[51] G. Pojmanski. The all sky automated survey. catalog of variable stars. i. 0 h - 6 hquarter of the southern hemisphere. *Acta Astronomica*, 52:397–427, 01 2002.

[52] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pages 8703–8714. PMLR, 2021.

[53] U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock. Finding anomalous periodic time series. *Machine learning*, 74(3):281–313, 2009.

[54] E. Reyes and P. A. Estévez. Transformation based deep anomaly detection in astronomical images. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[55] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[56] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323, 1986.

[58] P. Sánchez-Sáez, H. Lira, L. Martí, N. Sanchez-Pi, J. Arredondo, F. E. Bauer, A. Bayo, G. Cabrera-Vives, C. Donoso-Oliva, and P. A. Estévez et al. Searching for changing-state agns in massive data sets. i. applying deep learning and anomaly-detection techniques to find agns with anomalous variability behaviors. *The Astronomical Journal*, 162(5):206, 2021.

[59] P. Sánchez-Sáez, I. Reyes, C. Valenzuela, F. Förster, S. Eyheramendy, F. Elorrieta, F. E. Bauer, G. Cabrera-Vives, P. A. Estévez, and M. Catelan et al. Alert classification for the alerce broker system: The light curve classifier. *The Astronomical Journal*, 161(3):141, 2021.

[60] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 582–588, Cambridge, MA, USA, 1999. MIT Press.

[61] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[62] B. Sesar, Ž. Ivezić, J. S. Stuart, D. M. Morgan, A. C. Becker, S. Sharma, L. Palaversa, M. Jurić, P. Wozniak, and H. Oluseyi. Exploring the variable sky with linear. ii. halo structure and substructure traced by rr lyrae stars to 30 kpc. *The Astronomical Journal*, 146, 05 2013.

[63] M. D. Soraisam, A. Saha, T. Matheson, C.-H. Lee, G. Narayan, A. K. Vivas, C. Scheidegger, N. Oppermann, E. W. Olszewski, and S. Sinha et al. A classification algorithm for time-domain novelties in preparation for LSST alerts. application to variable stars and transients detected with DECam in the galactic bulge. *The Astrophysical Journal*, 892(2):112, apr 2020.

[64] R. F. Stellingwerf. Period determination using phase dispersion minimization. *The Astrophysical Journal*, 224:953–960, 1978.

[65] D. M. J. Tax and R. P. W. Duin. Uniform object generation for optimizing one-class classifiers. *J. Mach. Learn. Res.*, 2:155–173, mar 2002.

[66] B. T.-H. Tsang and W. C. Schultz. Deep neural network classifier for variable stars with novelty detection capability. *The Astrophysical Journal*, 877(2):L14, 2019.

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[68] V. A. Villar, M. Cranmer, G. Contardo, S. Ho, and J. Y.-Y. Lin. Anomaly detection for multivariate time series of exotic supernovae. *arXiv preprint arXiv:2010.11194*, 2020.

[69] H. Wang, M. J. Bah, and M. Hammad. Progress in outlier detection techniques: a survey. *IEEE Access*, 7:107964–108000, 2019.

[70] S. Webb, M. Lochner, D. Muthukrishna, J. Cooke, C. Flynn, A. Mahabal, S. Goode, Y. Andreoni, T. Pritchard, and T. M. C. Abbott. Unsupervised machine learning for transient discovery in deeper, wider, faster light curves. *Monthly Notices of the Royal Astronomical Society*, 498(3):3077–3094, 2020.

[71] M. Williamson, M. Modjaz, and F. B. Bianco. Optimal classification and outlier detection for stripped-envelope core-collapse supernovae. *The Astrophysical Journal*, 880(2):L22, jul 2019.

[72] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture mmodel for unsupervised anomaly detection. In *ICLR*, 2018.

Anexo

Lista de acrónimos:

- AE: Autoencoder.
- AEGMM: Autoencoder gaussian mixture model.
- AI: Transformación inverso de la amplitud.
- ALeRCE: Automatic Learning for the Rapid Classification of Events.
- ASAS: All Sky Automated Survey.
- ASAS-SN: All Sky Automated Survey for Supernovae.
- AUC: Area under the curve.
- AUCPR: Area under the curve precision recall.
- AUG: Transformación usada como aumento de datos.
- AUX: Transformación usada como outlier auxiliar.
- AVG: Representación promedio en el tiempo.
- BS: Batch size.
- CB: Clasificador binario.
- CCD: Charged-coupled device.
- CF: Función de costo.
- C-H: Puntaje de Calinski-Harabasz.
- CL: Curva de luz.
- CNN: Convolutional neural network.
- CON: Función de costo contrastiva.
- D-B: Puntaje de Davies-Bouldin.
- DF: Función de distancia.
- DTW: Dynamic time warping.
- FATS: Feature Analysis for Time Series.
- GMM: Gaussian mixture model.
- GRU: Gated recurrent unit.
- HP: Hiperparámetro.
- ID: In-distribution.
- IF: Isolation forest.
- IT: Isolation tree.

- kNN: K nearest neighbor.
- LAST: Última representación en el tiempo.
- LINEAR: Lincoln Near-Earth Asteriod Research.
- LOF: Local outlier factor.
- LR: Learning rate.
- LSST: Legacy Survey of Space and Time.
- LSTM: Long short-term memory.
- MJD: Modified julian date.
- MLP: Multilayer perceptron.
- MW: Transformación pandeo de la magnitud.
- NHL: número de capas ocultas.
- OA: Outlier auxiliar.
- OCSVM: One-class support vector machine.
- ODT: Outlier Detection based on Transformations for Astronomical Time Series, nombre del método propuesto (no considera la A (Astronomical), T (Time) y S (Series) finales).
- OOD: Out-of-distribution.
- PCA: Principal component analysis.
- RNN: Recurrent neural network.
- ROC: Receiver operator characteristic.
- RS: Transformación escalamiento aleatorio.
- SC: Coeficiente de silueta.
- SGD: Stochastic gradient descend.
- SVM: Support vector machine.
- TI: Transformación inverso temporal.
- TRI: Función de costo por tripleta.
- TS: Transformación desplazamiento temporal.
- TSL: Transformación corte del tiempo.
- TW: Transformación pandeo del tiempo.
- WD: Weight decay.
- ZM: Transformación enmascaramiento.
- ZTF: Zwicky Transient Facility.
- ZTF-EST: Conjunto de datos ZTF estocástico.
- ZTF-PER: Conjunto de datos ZTF periódico.
- ZTF-TRA: Conjunto de datos ZTF transiente.