



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

**CLASIFICACIÓN DE OBJETOS ASTRONÓMICOS UTILIZANDO  
SECUENCIAS CORTAS DE IMÁGENES Y METADATOS**

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

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

DANIEL ALEJANDRO NEIRA ORTEGA

PROFESOR GUÍA:  
PABLO ESTÉVEZ VALENCIA

MIEMBROS DE LA COMISIÓN:  
FELIPE TOBAR HENRÍQUEZ  
PABLO HUIJSE HEISE

Este trabajo ha sido parcialmente financiado por:  
ANID-Chile, Proyecto FONDECYT 1220829  
y el Instituto Milenio de Astrofísica ICN12\_009

SANTIAGO DE CHILE

2024

RESUMEN DE LA TESIS PARA OPTAR AL  
GRADO DE MAGÍSTER EN CIENCIAS DE LA  
INGENIERÍA, MENCIÓN ELÉCTRICA  
Y MEMORIA PARA OPTAR AL  
TÍTULO DE INGENIERO CIVIL ELÉCTRICO  
POR: DANIEL ALEJANDRO NEIRA ORTEGA  
FECHA: 2024  
PROF. GUÍA: PABLO ESTÉVEZ VALENCIA

## CLASIFICACIÓN DE OBJETOS ASTRONÓMICOS UTILIZANDO SECUENCIAS CORTAS DE IMÁGENES Y METADATOS

En esta tesis se aborda la clasificación de objetos astronómicos utilizando alertas generadas por el sondeo Zwicky Transient Facility (ZTF). El modelo utiliza imágenes y metadatos contenidos en cada una de las detecciones de los objetos astronómicos, como también metadatos provenientes del catálogo ALLWISE, con el cual se entrena una red neuronal del tipo recurrente convolucional. El clasificador propuesto, denominado *temporal stamp classifier*, es capaz de clasificar entre las clases Núcleo Galáctico Activo (AGN), SuperNova (SN) y Estrellas Variables (VS), obteniendo un accuracy de 98.4% para 2 o más detecciones. Los resultados del modelo muestran que el desempeño mejora al agregar más detecciones, como también que modelos simples de recurrencia obtienen resultados competitivos a los que obtienen modelos complejos como LSTM. Como objetivo secundario, se trabaja aplicando nuevas operaciones y características al modelo stamp classifier original para mejorar su desempeño, lo cual se basa en cambios en la arquitectura y adición de rotaciones aleatorias, con lo cual se obtiene un incremento del 1.5% de accuracy en contraste con el modelo original.

*A MIS PADRES, PAREJA  
Y MIS 9 GATOS.*

***Muchas gracias***

# Agradecimientos

Agradecer en primer lugar a mis padres por su incondicional apoyo durante el proceso de realización de mi tesis. Sin su ayuda para poder llevar el trabajo y la tesis, no podría haberlo logrado. Sus palabras de aliento en los momentos difíciles, un simple "Tu puedes" me permitieron llegar a este punto.

También agradecer a Vale, mi polola, que ha estado ahí durante todo el proceso dandome su apoyo, cariño y alentandome para seguir adelante.

Agradecer a mis amigos que hice en la universidad, por todos los momentos de risa, estudio, conversaciones, compartir en los pastos, etc. Gracias a Bastián, Daniel, Henrich, Taco, Luciano, Yalian. También a mis amigos que he hecho en la vida y que me siguen acompañando Pepe, Nacho, Palma, Javur, Victor, Negro, Agustín, Lurd y Mario. Agregar también a todos los gatos que han estado durante estos años, entre los que se cuentan oso, coso, manchas, peque, sofia, lita, nina, dinamita, carbon, lana, rucindo, pascualito, moises, cormenta, silvestrina, negris, robusto, grafa, mustaine, junior, silvestre y maximo.

En especial, agradecer a mi profesor guía Pablo Estevez, su ayuda, consejos y guianza durante estos años fué una gran ayuda para mi desarrollo como persona y la realización de este trabajo. Agradecer también a la gente del laboratorio por sus comentarios durante mi trabajo, que fué un gran aporte para probar nuevas ideas de mejora, en especial a Jhon por su gran disposición para ayudar.

Mencionar también a los profesores Domenico Sapone y Jeremy Barbay por todo el tiempo que pasé siendo ayudante y auxiliar de sus respectivos ramos, aportando en mi desarrollo profesional, de mis habilidades blandas y su gran disposición por la educación y el desarrollo de uno como persona.

Finalmente, agradecer a todos los que no mencioné anteriormente y tuve la oportunidad de conocer en la universidad, con los cuales compartí una risa, muchas gracias a ti.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	4
1.2. Hipótesis . . . . .	4
1.3. Objetivo general . . . . .	5
1.4. Objetivos específicos . . . . .	5
1.5. Estructura de la tesis . . . . .	5
1.5.1. Acrónimos . . . . .	6
<b>2. Marco teórico</b>	<b>8</b>
2.1. Astronomía . . . . .	8
2.1.1. Estudios astronómicos . . . . .	9
2.1.2. Zwicky Transient Facility . . . . .	10
2.1.3. Objetos astronómicos . . . . .	12
2.2. Aprendizaje de máquinas . . . . .	15
2.3. Problema de clasificación . . . . .	15
2.4. Redes neuronales artificiales . . . . .	16
2.4.1. Modelo del perceptrón . . . . .	16
2.4.2. Capa Fully-connected . . . . .	16
2.4.3. Perceptrón Multicapa (MLP) . . . . .	17
2.4.4. Redes convolucionales . . . . .	18
2.5. Redes neuronales recurrentes . . . . .	19
2.5.1. Modelo de recurrencia simple . . . . .	19
2.5.2. Long Short-Term Memory Model (LSTM) . . . . .	20
2.5.3. Modelo Gated Recurrent Unit (GRU) . . . . .	22
2.6. Time Delay Neural Network (TDNN) . . . . .	24
2.7. Gamma memory . . . . .	24
2.7.1. Filtro de memoria . . . . .	25
2.7.2. Leaky Integrator . . . . .	26
2.7.3. Gamma memory . . . . .	26
2.8. Transferencia de aprendizaje . . . . .	26
2.9. Rotaciones Aleatorias . . . . .	27
<b>3. Metodología</b>	<b>29</b>
3.1. Mejora a la clasificación temprana de objetos astronómicos . . . . .	29
3.1.1. Base de datos . . . . .	29
3.1.2. Pre-procesamiento . . . . .	30
3.1.3. Partición de datos . . . . .	31

3.1.4.	Arquitectura del clasificador . . . . .	32
3.1.5.	Regularización de la entropía . . . . .	34
3.1.6.	Esquema de entrenamiento . . . . .	34
3.1.7.	Experimentos . . . . .	34
3.1.7.1.	Cantidad de rotaciones, rotaciones aleatorias y tamaño de la imagen . . . . .	35
3.1.8.	Experimentos extras . . . . .	35
3.2.	Clasificación temporal de objetos astronómicos . . . . .	36
3.2.1.	Base de datos . . . . .	36
3.2.2.	ALLWISE . . . . .	36
3.2.3.	Pre-procesamiento . . . . .	37
3.2.4.	Partición de datos . . . . .	41
3.2.5.	Arquitectura del clasificador . . . . .	42
3.2.6.	Esquema de entrenamiento . . . . .	45
3.2.7.	Experimentos . . . . .	46
3.2.7.1.	Pre entrenamiento de capas convolucionales . . . . .	46
3.2.7.2.	Hiperparámetros procesamiento recurrente . . . . .	46
3.2.7.3.	Inclusión de metadatos del catálogo ALLWISE . . . . .	47
3.2.7.4.	Experimentos extras . . . . .	47
<b>4.</b>	<b>Resultados</b>	<b>51</b>
4.1.	Mejora al modelo <i>Stamp Classifier</i> . . . . .	51
4.1.1.	Variación del tamaño de recorte de las imágenes de entrada, cantidad de rotaciones y grado de libertad . . . . .	51
4.1.2.	Mejor modelo Stamp Classifier . . . . .	53
4.2.	Clasificación temprana temporal de objetos astronómicos . . . . .	53
4.2.1.	Pre entrenamiento de capas convolucionales . . . . .	53
4.2.2.	Comparación entre modelos recurrentes . . . . .	54
4.2.3.	Cruce con catálogo ALLWISE . . . . .	56
4.2.4.	Modelo final recurrente . . . . .	59
4.2.4.1.	Matrices de confusión . . . . .	59
4.2.4.2.	Evaluación de predicciones . . . . .	60
4.2.4.3.	Comparación con otros modelos . . . . .	62
<b>5.</b>	<b>Análisis y discusión de resultados</b>	<b>64</b>
5.1.	Mejoras en el modelo <i>Stamp Classifier</i> . . . . .	64
5.2.	Clasificación temprana temporal de objetos astronómicos . . . . .	66
<b>6.</b>	<b>Conclusiones</b>	<b>70</b>
6.1.	Mejora al modelo Stamp Classifier . . . . .	70
6.2.	Clasificación temprana temporal de objetos astronómicos . . . . .	70
6.3.	Propuestas de trabajo futuro . . . . .	71
6.3.1.	Clasificación temprana temporal de objetos astronómicos . . . . .	71
6.3.2.	Mejoras al modelo Stamp Classifier . . . . .	73
6.4.	Comentarios finales . . . . .	74
<b>Bibliografía</b>		<b>75</b>
Bibliografía . . . . .		75

<b>Anexos</b>	<b>83</b>
A. Metadatos . . . . .	83
A.1. Documentación de los metadatos . . . . .	83
A.2. Rango de metadatos . . . . .	84
A.3. Distribución de ALL-WISE . . . . .	85
A.4. Comparación de la distribución de metadatos en forma temporal . . .	85
B. Resultados detallados del modelo Temporal Stamp Clasifier . . . . .	87
B.1. Transfer Learning . . . . .	87
B.2. Entrenamiento de capas convolucionales . . . . .	88
B.3. Probabilidades asignadas por los modelos actuales a la base de datos temporal Stamp Classifier . . . . .	88
B.4. Importancia de metadatos primera detección . . . . .	89
C. Variaciones Stamp Classifier . . . . .	90
C.1. Validación de resultados con datos extra . . . . .	90
C.2. Cruce con catálogo ALLWISE . . . . .	91

# Índice de Tablas

3.1.	Arquitectura del <i>Stamp Classifier</i> propuesto. . . . .	33
3.2.	Búsqueda hiperparámetros para el modelo <i>Stamp Classifier</i> propuesto. . . . .	35
3.3.	Distribución de ejemplos por clase en el conjunto de entrenamiento para cada número de detecciones. . . . .	42
3.4.	Arquitectura del temporal Stamp Classifier para cualquier número de detecciones $t$ . La capa <i>RNN</i> depende del número de detecciones considerado y resumido en la tabla 4.4. . . . .	44
3.5.	Búsqueda de hiperparámetros para capas recurrentes. Cada configuración es entrenada para cada número de detecciones considerada en esta tesis. D corresponde a la cantidad de detecciones considerada. . . . .	47
3.6.	Arquitectura del modelo <i>Temporal Stamp Classifier</i> que utiliza solo metadatos y no considera imágenes. . . . .	49
4.1.	Top 5 modelos con el mayor accuracy en el conjunto de validación obtenido en la búsqueda de hiperparámetros para el modelo Stamp Classifier mejorado. Los modelos van rankeados desde $M_1$ hasta $M_5$ . $M_r$ corresponde al modelo con una rotación total de $180^\circ$ y $M_b$ corresponde al modelo base Stamp Classifier. . . . .	52
4.2.	Desempeño de los mejores modelos por tipo de modelo recurrente. Cada modelo recurrente presenta la mejor configuración obtenida para cada número de detecciones. Se muestra el valor p del test de hipótesis de permutación entre el modelo y el modelo LSTM para ese número de detecciones y la arquitectura del modelo en su abreviación. . . . .	55
4.3.	Resultados en términos de <i>accuracy</i> entre modelos de solo metadatos de la tabla 3.6 que incluyen y no incluyen el cruce con el catálogo ALLWISE, en conjunto con el valor del test de hipótesis de permutación [99] entre ambos modelos para cada número de detecciones. . . . .	59
4.4.	Mejor configuración de red recurrente encontrado para cada número de detecciones según tiempo de inferencia obtenidos en la sección 4.1.2. . . . .	59
A.1.	Descripción de los metadatos utilizados en esta tesis. . . . .	83
A.2.	Descripción de los metadatos utilizados en este trabajo . . . . .	84
A.3.	Limite de valores de los metadatos. Los valores <b>max</b> y <b>min</b> significa que se mantiene el valor máximo y mínimo del conjunto de entrenamiento para ese metadato. . . . .	84



# Índice de Ilustraciones

2.1.	Ejemplo de curva de luz de un objeto donde se muestran las curvas formadas en la banda G y R. Figura tomada de [24]. . . . .	9
2.2.	Campo de visión v/s área de recolección de luz para distintos <i>surveys</i> . Figura tomada de [9]. . . . .	11
2.3.	Taxonomía utilizada en ALeRCE. Figura tomada de [9]. . . . .	13
2.4.	Ejemplo de las imágenes <i>stamps</i> de las detecciones en las cinco clases. Figura de creación propia. . . . .	15
2.5.	Modelo del perceptrón. . . . .	17
2.6.	Modelo del perceptrón multicapa. . . . .	17
2.7.	Ejemplo del proceso de convolución usando stride, y con un solo canal de entrada y salida, usando un filtro de $2 \times 2$ . . . . .	18
2.8.	Modelo de red recurrente simple. . . . .	19
2.9.	Modelo LSTM. Figura tomada de [26]. . . . .	21
2.10.	Modelo GRU. Figura tomada de [26]. . . . .	23
2.11.	Modelo TDNN. . . . .	25
2.12.	Modelo de Gamma Memory. . . . .	26
2.13.	Ejemplo de aplicación de rotación aleatoria sobre una imagen en el rango $0^\circ \pm 30^\circ$ . Cada imagen representa una rotación aleatoria aplicada sobre la imagen original. . . . .	28
3.1.	Ejemplo de estampillas de la primera detección de las 5 clases consideradas para el modelo <i>Stamp Classifier</i> . Figura tomada de [10]. . . . .	30
3.2.	Modelo <i>Stamp Classifier</i> propuesto, donde $a$ significa el tamaño del recorte utilizado, $\theta$ el ángulo de libertad para la rotación aleatoria y considerando 5 rotaciones. . . . .	33
3.3.	Proporciones y distribución de la base de datos utilizada en el modelo temporal stamp classifier en formato crudo. Figura de creación propia. . . . .	37
3.4.	Proporciones y distribución de la base de datos utilizada en el modelo temporal stamp classifier después del pre procesamiento. Figura de autoría propia. . . . .	40
3.5.	Distribución de las diferencias temporales en escala logarítmica para las 3 clases trabajadas. . . . .	41
3.6.	Modelo <i>Temporal Stamp Classifier</i> . Ejemplo para el caso de 3 detecciones, con 5 rotaciones y con un ángulo de libertad $\theta$ . <i>RNN</i> simboliza la red recurrente aplicada y detallado por número de detecciones en la tabla 4.2. . . . .	45
3.7.	Separación de metadatos estáticos y dinámicos en el entrenamiento. . . . .	48
4.1.	Tiempo de inferencia en contraste con el accuracy en test logrado por los modelos para cada número de recorte utilizado y para cada número de cantidad de rotaciones. . . . .	52

4.2.	Matrices de confusión del modelo original de Stamp Classifier presentado en [10], el cual incluye metadatos e imágenes, en el lado izquierdo (a) y del mejor modelo obtenido con las modificaciones propuestas en esta tesis en el lado derecho (b) con metadatos e imágenes. Entre ambos modelos se obtiene un valor $p = 0.01$ .	53
4.3.	Matriz de confusión para el pre entrenamiento de los filtros convolucionales utilizando la primera detección de los objetos. . . . .	54
4.4.	Desempeño de los modelos recurrentes comparados para cada número de detecciones utilizado en términos de accuracy y tiempo de inferencia. Los modelos marcados con un diamante corresponden al mejor modelo obtenido para ese tipo de red recurrente. . . . .	56
4.5.	Matrices de confusión del modelo usando solo metadatos de la tabla 3.4 comparando el efecto del cruce con el catálogo ALLWISE para 2 y 3 detecciones de un objeto. Las matrices del lado izquierdo (a y c) muestran los resultados de 2 y 3 detecciones respectivamente sin la utilización de ALLWISE, mientras que las matrices del lado derecho (b y d), muestran los resultados de 2 y 3 detecciones respectivamente considerando ALLWISE en el entrenamiento. El modelo no utiliza imágenes. . . . .	57
4.6.	Matrices de confusión del modelo usando solo metadatos de la tabla 3.4 comparando el efecto del cruce con el catálogo ALLWISE para 4 y 5 detecciones de un objeto. Las matrices del lado izquierdo (a y c) muestran los resultados de 4 y 5 detecciones respectivamente sin la utilización de ALLWISE, mientras que las matrices del lado derecho (b y d), muestran los resultados de 4 y 5 detecciones respectivamente considerando ALLWISE en el entrenamiento. El modelo no utiliza imágenes. . . . .	58
4.7.	Matrices de confusión del mejor modelo obtenido para cada número de detecciones en el conjunto de test. El modelo utiliza imágenes de las detecciones y los metadatos considerando ALLWISE. Los modelos siguen la estructura de la tabla 3.1. . . . .	60
4.8.	Ejemplo de muestras que fueron mal clasificadas por alguno de los modelos completos entrenados. Se muestran las imágenes de las 5 detecciones, donde S corresponde a las imágenes de Science, T a las imágenes de Template y D a las de Difference. . . . .	61
4.9.	Gráfico de dispersión de las 3 clases consideradas en coordenadas galácticas. Los puntos rojos corresponden a ejemplos mal clasificados en al menos una de las inicializaciones de los modelos finales recurrentes en cualquier número de detecciones. . . . .	62
4.10.	Matriz de confusión de Stamp Classifier y Light Curve Classifier de producción de ALerCE utilizando el conjunto de test de base de datos del <b>Temporal Stamp Classifier</b> . . . . .	63
A.1.	Distribución de metadatos ALL-WISE. . . . .	85
A.2.	Distribución de metadatos VS. . . . .	85
A.3.	Distribución de metadatos SN. . . . .	86
A.4.	Distribución de metadatos AGN. . . . .	86
B.1.	Evolución de la función de pérdida cuando se utiliza los pesos de la primera detección como pesos iniciales para los filtros convolucionales y cuando se inicializan de forma aleatoria. . . . .	87

B.2.	Matriz de confusión modelo de solo imágenes para caso con solo datos de la primera detección y utilizando las imágenes de todas las detecciones. . . . .	88
B.3.	Distribución de probabilidades Stamp Classifier. . . . .	88
B.4.	Distribución de probabilidades Top Light Curve Classifier. . . . .	89
B.5.	Importancia de los metadatos en la primera detección modelo Temporal Stamp Classifier. . . . .	89
C.1.	Matriz de confusión mejor modelo Stamp Classifier con nuevos datos en su primera detección. . . . .	90
C.2.	Matriz de confusión modelo Stamp Classifier incluyendo ALLWISE sobre base de datos ALeRCE-ZTFV5. . . . .	91
C.3.	Matriz de confusión modelo Stamp Classifier incluyendo ALLWISE sobre nuevos datos en su primera detección. . . . .	92

# Capítulo 1

## Introducción

El avance de la tecnología ha permitido la generación de súper telescopios capaces de recopilar una gran cantidad de información de forma continua, realizando sondeos, en el cual se analizan zonas del cielo constantemente en espera a que se detecte algún cambio, con lo cual se recolectan grandes cantidades de datos. Dentro de estos estudios se tienen los *surveys*, los cuales permiten la captura y distribución de datos astronómicos de gran volumen, entre los que se encuentran Asteroid Terrestrial-Impact Last Alert (ATLAS) [1], Hyper Suprime-Cam Subaru Strategic Program (HSC-SSP) [2], Zwicky Transient Facility (ZTF) [3], etc. Algunos de estos *surveys* procesan y distribuyen datos en tiempo real, por lo cual se hace necesario el tratamiento de ellos en la misma velocidad para no perder datos. Esto produce que los métodos manuales de análisis sean ineficientes debido a la necesidad de un experto para analizar la gran cantidad de datos, lo cual tiene un alto tiempo de procesamiento. Debido a esto, se ha hecho necesaria la inclusión de métodos automáticos que permitan procesar los datos, con una eficacia comparable a la de un experto, y con una velocidad de procesamiento mucho superior, lo cual se ha logrado gracias a la inclusión de *Machine Learning*.

Bajo esta mirada, han existido distintos acercamientos para analizar los datos astronómicos, entre los que se encuentran la predicción [4–6], la cual se basa en predecir el comportamiento futuro de un objeto a partir de su comportamiento histórico, como también la tarea de clasificación [7, 8], que se encarga de clasificar de manera automática un ejemplo entre ciertas clases pre definidas.

Estos métodos de machine learning, junto a otros métodos computacionales son aplicados por *brokers* astronómicos, los cuales son organizaciones de carácter científico que se encargan de procesar los datos provenientes de los *surveys* en tiempo real, haciendo públicos sus resultados, datos procesados y métodos a la comunidad científica para análisis posteriores. Dentro de los *brokers* existentes, se destaca Automatic Learning for the Rapid Classification of Events (ALeRCE) [9], el cual se encuentra liderado por investigadores chilenos en cooperación con la comunidad internacional. ALeRCE trabaja principalmente con los *surveys* ZTF y ATLAS, procesando sus alertas, utilizando técnicas de *machine learning* para la clasificación de objetos astronómicos.

ALeRCE, dentro de su investigación dirigida principalmente a la clasificación, ha desarrollado un *pipeline* que contiene dos métodos para la clasificación de objetos astronómicos, los cuales están divididos en dos fases dependiendo de la cantidad de alertas para un objeto.

- Clasificación temprana de una alerta: Al momento de obtener la primera detección de un objeto, ésta es clasificada utilizando la imagen generada en el lugar de observación, junto a datos provenientes del *survey* y una imagen de referencia [10].

- Clasificador tardío de series de tiempo: Este segundo clasificador utiliza la serie de tiempo generada por un objeto astronómico para realizar la clasificación, utilizando un modelo Random Forest. Para realizar esto, es necesario tener un mínimo de 6 muestras temporales para generar la serie de tiempo [11]

Ambos trabajos se diferencian principalmente en la utilización de distintos tipos de datos para realizar la clasificación, donde el correspondiente a [10] utiliza las imágenes adquiridas, como también los metadatos del momento de la detección. Este clasificador es la base del trabajo que se realizará en esta tesis, por lo cual se realizará un enfoque de procesamiento de imágenes.

Fukushima & Miyake [12] demostraron mediante simulaciones computacionales que es posible encontrar patrones espaciales dentro de imágenes utilizando redes neuronales, trabajo que principalmente se realizaba a partir de filtros convolucionales inspirados en el funcionamiento de la visión humana y animal, tal como encontraron Hubel & Wiesel [13], quienes demostraron mediante experimentos que las neuronas responden a patrones específicos dentro del campo visual, y que patrones complejos son detectados mediante computaciones entre neuronas.

El año 2012, Krizhevsky et al. [14] implementaron la red AlexNet en ImageNet Large Scale Visual Recognition Challenge, que utilizaba capas convolucionales para procesar las imágenes. A pesar de no ser la primera utilización de este tipo de redes, AlexNet obtuvo una precisión 10% mayor que el último ganador, estableciendo un nuevo estándar a partir de la utilización de las capas convolucionales, la función de activación ReLU, el uso de las GPUs para el entrenamiento de los modelos y la utilización de múltiples capas. Posterior a AlexNet, se desarrollaron otras redes neuronales para la clasificación de imágenes en la misma competencia, con un mejor desempeño que AlexNet, donde la gran mayoría comparte la característica de utilizar redes convolucionales, entre las que se encuentran [15–17].

En vista de los buenos resultados obtenidos por este tipo de redes en la clasificación de imágenes, se empezó a estudiar su utilización en problemas astronómicos. Wright & Lintott et al. [18] fueron unos de los primeros acercamientos utilizando redes convolucionales, presentando una red para poder clasificar y encontrar objetos transientes (en particular Super Novas) entre artefactos, los cuales corresponden a asteroides, y ruido o errores de medición. El objetivo fue poder procesar los datos desde un *survey* específico minimizando la intervención humana, para lo cual combina la detección desde la base de datos Pan-STARRS1 [19] utilizando su modelo entrenado y complementado con la clasificación humana para mejorar el desempeño de clasificación, comprobando que la combinación de ambas es mejor que cada una por separado.

Posteriormente, Cabrera et al. [20] propusieron el modelo Deep-HiTS, compuesto por capas convolucionales. Este utiliza la técnica de invariancia rotacional propuesta en [21], la cual consiste en rotar las imágenes de entrada en  $90^\circ$ ,  $180^\circ$  y  $270^\circ$ , lo que permite evitar el problema de imágenes rotadas debido a la utilización de todas sus rotaciones para el procesamiento de la imagen. Este modelo utiliza datos provenientes de HiTS [22] con el principal objetivo de clasificar y encontrar Supernovas, aumentando en gran medida el desempeño obtenido hasta ese momento. Por otro lado, Carrasco et al. [23] presenta un nuevo modelo, compuesto por una combinación de redes convolucionales y redes recurrentes, utilizando como entrada series de imágenes de un mismo objeto astronómico para realizar la clasificación, donde estas imágenes son detecciones en tiempos distintos. Los datos utilizados corresponden a datos simulados de HiTS, por lo cual cumplen con ciertas restricciones que los vuelven ideales y

presentan diferencias con las imágenes reales que se pueden obtener; sin embargo, permitió demostrar que la utilización de secuencias de imágenes de un mismo objeto permite tener una clasificación más confiable del objeto e incluso tener una mayor definición de su clase (aumentar la cantidad de clases a la cual puede pertenecer).

Basándose en el modelo Deep-HiTS [20], Carrasco-Davis & Reyes et al. [10] diseñaron el *Stamp Classifier*. Este modelo utiliza los datos provenientes de ZTF, la arquitectura de Deep-HiTS para el procesamiento de las imágenes junto a la inclusión de metadatos provenientes del *survey*. La inclusión de estos metadatos permite tener el contexto sobre el cual la detección fue tomada, junto a la combinación de 3 imágenes en la detección. La imagen tomada y centrada en un objeto de interés (*science*), la imagen con un promedio de lo que había antes de la detección (*template*) y una resta entre ambas imágenes (*difference*). Este modelo permitió mejorar la detección de Supernovas en gran medida, junto con la inclusión de otras 4 clases: asteroides, objetos estocásticos, objetos periódicos y falsas detecciones (Bogus). Esta clasificación se realiza a partir de la primera detección de un objeto junto a los datos de la observación de ésta.

Finalmente, Arredondo [24] demostró en su tesis que es posible extender el trabajo realizado en el *Stamp Classifier*, trabajando con 2 detecciones del mismo objeto, esto permite tener una mejora en la clasificación lograda por el modelo. Esta mejora se logra principalmente gracias a la inclusión de la diferencia entre cada medición, esto debido principalmente a que los objetos transientes (como las Supernovas), tienden a ser procesos cortos, por lo cual la diferencia entre sus mediciones tienden a ser pequeñas en comparación al resto de las clases.

Si bien, se han desarrollado varios modelos para la clasificación de objetos astronómicos, existen varios puntos que no han sido estudiados en detalle, entre los cuales se destacan los siguientes:

1. Los modelos existentes para procesar alertas provenientes de ZTF consideran la clasificación de 1 o 2 detecciones para el caso de *Stamp Classifier* y su extensión, o consideran curvas de luz de 6 muestras para el caso del *Late Classifier*, por lo cual se puede mejorar la clasificación temprana incluyendo más detecciones del mismo objeto. Esto permitiría acercar los resultados utilizando imágenes al clasificador que utiliza curvas de luz.
2. El *Stamp Classifier* solo considera la primera muestra, mientras que el modelo [24] considera dos imágenes por separado para su procesamiento, por lo cual no hay un tratamiento de las dependencias temporales entre muestras. Por otro lado, el procesamiento de la curva de luz dada por el modelo *Late Classifier* [11] utiliza, entre otras cosas, el período de la señal, lo cual no es posible obtener con baja cantidad de muestras temporales.
3. La utilización de imágenes para la clasificación posee un alto costo computacional, por lo cual se imposibilita el uso de estas en casos donde se tenga una gran cantidad de mediciones de un objeto.
4. A medida que aumenta la cantidad de mediciones, hay ciertas clases que dejan de tener importancia debido a su carácter efímero, como son el caso de asteroides y bogus, los cuales no son considerados en [11] pero sí en [10, 24].

A partir de estos antecedentes, la presente tesis busca diseñar e implementar un clasificador de objetos astronómicos que considere información en forma de imágenes y metadatos provenientes desde el *survey* ZTF. Debido a la existencia actual de dos clasificadores que funcionan a distinta cantidad de muestras temporales, se propone que este modelo considere

desde 2 a 5 detecciones de un objeto para realizar la clasificación. Debido a la cantidad de información presente en las imágenes, la cual crece significativamente a medida que aumenta la cantidad de muestras, se propone la utilización de modelos de redes neuronales recurrentes tanto simples como más complejas, comparando el desempeño de éstas. Esto debido a que, al tener recurrencias cortas, se hace viable el hecho de que redes con pocos parámetros puedan tener un desempeño similar a modelos complejos de recurrencia.

Por otra parte, al tener muestras con características temporales, es posible utilizar el tiempo en el cual estas detecciones son capturadas, lo cual entrega información valiosa para realizar la clasificación. Tradicionalmente el tiempo es manejado de manera retrospectiva, o sea, se calcula la diferencia entre dos muestras consecutivas para el procesamiento. A pesar de que esto ha resultado bien para clasificación de objetos con muestreos irregulares [25], han habido otras propuestas para el tratamiento del tiempo y estimación del periodo[26–28]; por lo tanto se propone estudiar distintas técnicas para el tratamiento del tiempo en el caso donde existe una baja cantidad de muestras.

## 1.1. Motivación

La clasificación temprana de objetos astronómicos es relevante para la detección y seguimiento de objetos de interés. Esto permite entender de mejor manera la física que describen estos objetos. Actualmente, debido al gran flujo de información proveniente de los telescopios actuales y los que están por venir, es imposible poder ser procesados por agentes humanos, por lo cual se vuelve necesario optar por métodos de clasificación automática.

Actualmente, en ALeRCE existe un vacío entre el procesamiento de la primera detección y el procesamiento de la curva de luz de un objeto, el cual requiere de 6 puntos o más en una sola banda para poder ser procesado, por lo cual existe mucha información sobre las primeras detecciones que no se está considerando para la clasificación, lo cual permitiría mejorar el desempeño de esta misma y corregir ejemplos que puedan ser confusos. Esto permitiría tener un enlace entre el procesamiento de imágenes para las alertas astronómicas y el procesamiento de las curvas de luz.

## 1.2. Hipótesis

Las hipótesis de esta tesis son:

- La inclusión de imágenes de más de una detección temprana de un objeto astronómico permitiría aumentar el desempeño del clasificador, pudiendo estimar con mayor confianza las respectivas clases.
- Se postula que el problema de recurrencia se puede resolver utilizando modelos simples tales como modelos *Gamma memory*, *TDNN*, *SimpleRNN* y *Conv1D*, entregando un desempeño comparable en exactitud y complejidad computacional que modelos recurrentes complejos del tipo *LSTM* y *GRU*.

### 1.3. Objetivo general

Desarrollar un clasificador de objetos astronómicos que utilice información de las primeras detecciones de objetos astronómicos compuestas por imágenes, tiempos de observación, datos de las coordenadas estelares y otros metadatos provenientes del survey ZTF.

### 1.4. Objetivos específicos

- Extender el modelo *Stamp Classifier* a un modelo que pueda utilizar la información de más de una detección del mismo objeto utilizando modelos recurrentes.
- Comparar el resultado de utilizar modelos simples para el procesamiento de las recurrencias con la de utilizar modelos complejos en términos de precisión y tiempos de ejecución.
- Comparar los resultados del modelo propuesto con el *Stamp Classifier* y con el clasificador jerárquico del *Light Curve Classifier*.
- Incluir modificaciones al modelo del *Stamp Classifier* original que mejoren la detección de SuperNovas.

### 1.5. Estructura de la tesis

Esta tesis se organiza en capítulos de la siguiente manera:

- **Introducción (Capítulo 1):** Se presenta el problema abordado en esta tesis, junto a una revisión literaria de trabajos anteriores que están directamente relacionados. También se presenta la motivación, hipótesis, objetivo general y los objetivos específicos de la tesis.
- **Marco teórico (Capítulo 2):** Se detallan los antecedentes teóricos comenzando por describir el concepto de astronomía, estudios astronómicos y como estos capturan datos de objetos astronómicos. Se detalla en particular el sondeo Zwicky Transient Facility (ZTF), su funcionamiento y la taxonomía utilizada por ALerCE para clasificar los objetos astronómicos. Finalmente, se realiza una descripción del aprendizaje de máquinas, junto a los modelos utilizados en esta tesis en términos teóricos y matemáticos con ejemplos de su utilización en trabajos anteriores.
- **Metodología (Capítulo 3):** En la Sección 3.1 se explica la metodología utilizada para resolver el problema de clasificación temporal de objetos astronómicos, mientras que en la Sección 3.2 se detalla la metodología para las mejoras propuestas al modelo *Stamp Classifier*. En estas secciones se describe la base de datos a utilizar, el pre procesamiento de datos aplicado a estas bases, también se incluye el modelo que se utilizó y los experimentos realizados en esta tesis.
- **Resultados (Capítulo 4):** Se reportan los resultados de los experimentos mencionados en el Capítulo 3 en términos de *accuracy*, tiempos de inferencia y comparación a partir de test de hipótesis. También se muestran figuras que ayudan a la interpretabilidad de los ejemplos mal clasificados por los modelos.



- **Análisis y discusión de resultados (Capítulo 5):** Se realiza un análisis de los resultados obtenidos en el capítulo anterior en términos de interpretabilidad de los modelos y de los ejemplos mal clasificados, se comparan con los modelos *Stamp Classifier* y *Light Curve Classifier* y se discute respecto a las diferencias entre las combinaciones de los modelos recurrentes, cruce con el catálogo ALLWISE y mejoras propuestas al modelo *Stamp Classifier*.
- **Conclusiones (Capítulo 6):** Finalmente, se concluye respecto al trabajo realizado y los resultados obtenidos a partir de los experimentos de esta tesis. Se presentan ideas de trabajo futuro para continuar este trabajo y comentarios finales de la tesis.
- **Anexos:** En esta sección se muestran descripciones detalladas de los metadatos utilizados en esta tesis, como también los resultados completos de todos los experimentos realizados y los resultados de los experimentos extras que no son esenciales para el desarrollo de esta tesis.

### 1.5.1. Acrónimos

- [ATLAS]: *Asteroid Terrestrial-Impact Last Alert.*
- [HSC-SSP]: *Hyper Suprime-Cam Subay Strategic Program.*
- [ZTF]: *Zwicky Transient Facility.*
- [ALeRCE]: *Automatic Learning for the Rapid Classification of Events.*
- [SC]: *Stamp Classifier.*
- [LCC]: *Light Curve Classifier.*
- [RNN]: *Recurrent Neural Network.*
- [TDNN]: *Time Delay Neural Network.*
- [SimpleRNN]: *Simple Recurrent Neural Network.*
- [Conv1D]: *Convolution in One Dimention.*
- [LSTM]: *Long Short-Term Memory Model.*
- [GRU]: *Gate Recurrent Unit.*
- [LSST]: *Legacy Survey of Space and Time.*
- [CCDs]: *Charge Couple Devices.*
- [PSF]: *Point Spread Function.*
- [ZADS]: *ZTF Alert Distribution System.*
- [SN]: *SuperNova.*
- [AGN]: *Galactic Nuclei (Núcleo Galáctico Activo).*
- [VS]: *Variable Star (Estrella Variable).*

- [RF]: *Random Forest*.
- [MLP]: *Multi Layer Perceptron*. (Perceptrón Multicapa).
- [TSC]: *Temporal Stamp Classifier*.
- [NaN]: *Not A Number*.
- [JD]: *Julian Date*.
- [ReLU]: *Rectified Linear Unit*.
- [BN]: *Batch Normalization*.
- [D]: *Número de detecciones*.
- [API]: *Application Programming Interface*.
- [BS]: *Batch Size*.
- [K]: *Número de rotaciones*.
- [IS]: *Image size*.
- [KS]: *Kernel Size*.
- [C]: *Capas*.
- [U]: *Unidades*.
- [V]: *Ventana de tiempo*.
- [Ke]: *Tamaño del kernel*.

# Capítulo 2

## Marco teórico

En este capítulo se describen los antecedentes teóricos para el desarrollo de este trabajo, incluyendo trabajos relacionados. En particular, se explican los conceptos de astronomía, *surveys*, redes neuronales y los modelos de redes utilizados en este trabajo, así como algunas técnicas que mejoran el entrenamiento de modelos de aprendizaje de máquinas y aprendizaje profundo.

### 2.1. Astronomía

La astronomía corresponde a la ciencia que estudia fenómenos y objetos presentes fuera de nuestro planeta, utilizando herramientas físicas, químicas y matemáticas para explicar estos fenómenos y sus orígenes. Existen dos grandes vertientes de esta ciencia: La cosmología que se encarga de estudiar el universo como un todo (interacciones de galaxias, materia oscura, etc) y la astronomía clásica, que estudia cuerpos celestes a un nivel más detallado.

Es considerada una de las ciencias más antiguas de la humanidad, teniendo sus orígenes en observaciones metódicas del cielo por parte de egipcios, griegos, chinos, mayas, etc. El avance de la matemáticas, física e instrumentos de medición han hecho posible que las investigaciones sigan dos rumbos posibles en este ámbito: La astronomía teórica y la observacional. En el caso de la astronomía teórica, se utilizan los principios de la física para construir modelos computacionales o analíticos que describen fenómenos y procesos astronómicos; por otra parte, la astronomía observacional toma datos entregados por observaciones (principalmente obtenidas desde telescopios) para analizar estos datos y compararla con principios físicos conocidos, análisis computacionales de ésta misma y con los modelos generados mediante la astronomía teórica. Esta cooperación entre ambas ramas de la astronomía permite validar lo desarrollado teóricamente con las observaciones obtenidas y sus consecuentes análisis.

Dentro de los objetos de estudio de la astronomía se encuentran las estrellas, las cuales son una acumulación de gas y polvo unidas debido a su gravedad. Dentro de estos objetos ocurren reacciones nucleares como la fusión nuclear, las cuales liberan una gran cantidad de energía, y mucha de esta energía es liberada en forma de luz, lo cual las hace detectables por parte de los telescopios que poseemos. En general, las estrellas que visualizamos se encuentran en la Vía Láctea, y dependiendo de su tamaño es posible definir su masa y también sus ciclos de vida, donde las estrellas masivas pasarán a ser SuperNovas, mientras que las frías

no alcanzan a romper su equilibrio y quedan como enanas cafés.

Para poder capturar información proveniente de objetos de interés se utilizan filtros de banda, los cuales centran la captura de luminosidad por parte de los telescopios a un rango de frecuencias muy específico. Esto permite filtrar la luminosidad proveniente de fuentes cercanas, lo cual permite ver objetos que no se encuentra cercanos o que son ofuscados por las fuentes cercanas, como también permite determinar que componentes tiene el objeto gracias a la frecuencia con la que emiten luz. Otras formas de capturar información proveniente de objetos astronómicos es mediante la medición de ondas gravitacionales [29] o detección de neutrinos [30, 31].

La información de estos objetos es capturada en distintos momentos, gatillada por el cambio en su luminosidad, con lo cual es posible construir una serie de tiempo utilizando los datos obtenidos en cada una de las detecciones. La serie de tiempo construida con la magnitud de la intensidad de luz es conocida como Curva de Luz, y tiene la particularidad de permitir discriminar el tipo de objeto que se está observando a partir de la amplitud, tiempo de decaimiento, período, etc. Cabe mencionar que la curva de luz se construye solo con detecciones en la misma banda de frecuencia, por lo cual tiende a ser más difícil construir una curva de luz para objetos con periodos muy largos, u objetos transientes que se encuentren en el final del evento de interés. Un ejemplo de curva de luz producida por un objeto en las bandas G y R se puede apreciar en la figura 2.1

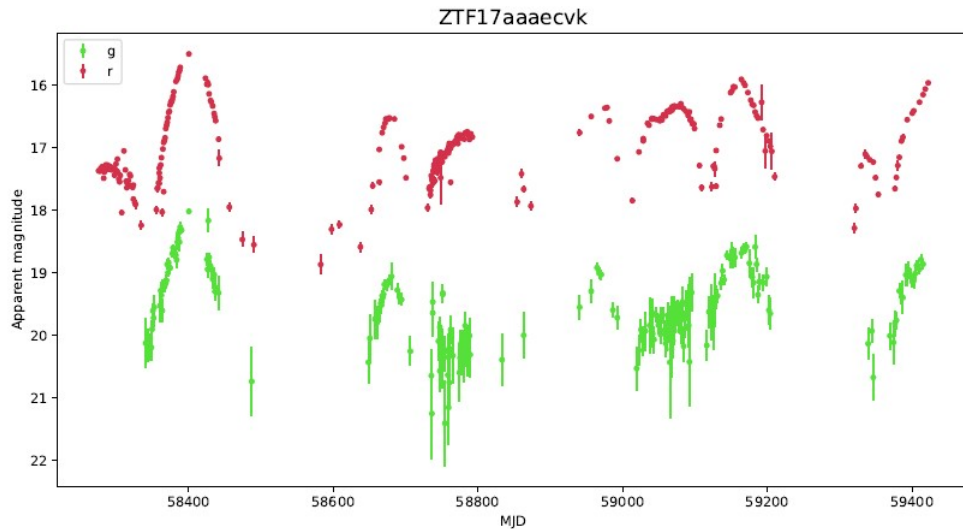


Figura 2.1: Ejemplo de curva de luz de un objeto donde se muestran las curvas formadas en la banda G y R. Figura tomada de [24].

### 2.1.1. Estudios astronómicos

Los sondeos astronómicos o *surveys* basan su trabajo en realizar un mapeo de una región del universo observable con el objetivo de obtener datos de la mayor parte de objetos que se encuentren en su área de trabajo. Estos datos son obtenidos con distintos aparatos

o técnicas de medición, tal como telescopios de última generación que toman muestras en distintas bandas, detectores de cambios en luminosidad, etc. Actualmente, estos son los mayores generadores de datos astronómicos gracias al gran avance de los sondeos, lo cual ha permitido un aumento en la velocidad de procesamiento, predicciones cada vez más precisas sin la intervención humana, gran capacidad de almacenamiento de datos, etc.

A modo de ejemplo, el estudio HiTS [22] realiza una medición sobre todo el espacio 5 veces cada dos o tres días en búsqueda de SuperNovas en el inicio de su proceso, poniendo especial atención en esas áreas para medir su evolución y encontrar evidencia del momento de su explosión [32, 33]. Otros ejemplos de estudios astronómicos son MACHO [34], ASAS [35], EROS [36] y LSST [37]. Este último es especialmente interesante debido a que aumenta considerablemente la región de observación que comenzará en 2025 y su cadencia en comparación a los otros estudios.

Se deben realizar algunas definiciones respecto del área de trabajo que van a abarcar, partiendo por el lugar del universo en el que van a trabajar, las bandas sobre las cuales van a realizar detecciones (frecuencias), instrumentos que utilizarán para realizar las mediciones, tiempo que se realizarán las mediciones de cada zona y la cadencia de la medición o frecuencia. Notar que tienen una gran importancia para la detección de objetos nuevos y de interés, los cuales permiten realizar estudios respecto a las dinámicas ante los cuales están sometidos, sobre todo para objetos que tienen comportamientos transientes como SuperNovas, donde se desea detectarlos en el inicio de su ciclo; por otro lado, también permiten realizar estimaciones de las distancias en el universo al ubicar un objeto. En el caso de las estrellas variables, estas mediciones son con respecto a la Vía Láctea, donde son comúnmente detectados [38, 39]. En el caso de las SuperNovas, que en su gran mayoría se encuentran fuera de nuestra galaxia, permiten realizar mediciones cosmológicas para explicar y entender la expansión del universo [40, 41].

Por lo general, tienden a ubicarse en lugares con un buen ambiente para la observación astronómica, los cuales requieren de cielos limpios y buenas condiciones climáticas que permitan un funcionamiento relativamente constante. Como el funcionamiento de estos es constante, generalmente se trabaja con una transmisión en tiempo real de la información que estos recaban, o sea, al momento de detectar algún evento de interés, este es alertado de forma automática a los destinatarios con la información que se recabó en el momento, que generalmente son las imágenes del lugar de interés, y los datos de observación, tales como los datos del telescopio, posición dentro del espacio, condiciones de medición, etc.

### 2.1.2. Zwicky Transient Facility

*Survey* que se encuentra actualmente en funcionamiento, y mediante el cual se obtuvieron los datos utilizados en esta tesis. Se ubica en el Observatorio Palomar en San Diego, California, Estados Unidos. Es un telescopio óptico que captura datos del cielo observable en el hemisferio norte. Posee uno de los campos de visión más grandes que se tiene actualmente, pudiendo observar en 3 bandas distintas: R, G y B (rojo, verde y azul). Esto lo hace propenso a encontrar una gran cantidad de candidatos a objetos. Las observaciones tomadas tienen una cadencia que va desde minutos hasta meses. Al poseer un campo de visión tan amplio, puede significar una menor resolución de los lugares observados, traduciéndose en una confusión entre objetos muy cercanos. Actualmente se está construyendo un nuevo *survey* que

generaría 10 veces más información por noche y con una mayor resolución que ZTF llamado LSST (Legacy Survey of Space and Time), el cual se ubica en Chile. Una comparación entre distintos *surveys* se puede apreciar en la figura 2.2

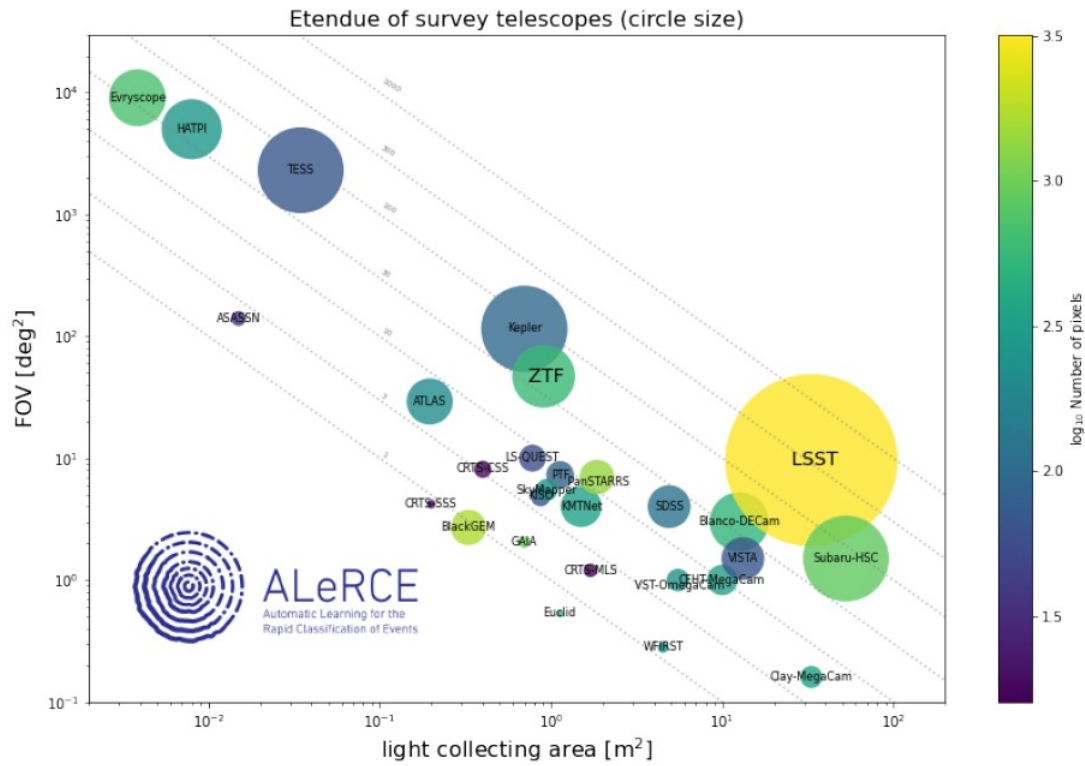


Figura 2.2: Campo de visión v/s área de recolección de luz para distintos *surveys*. Figura tomada de [9].

ZTF, al ser un telescopio óptico, está compuesto por un mosaico de 16 CCDs: Los CCDs (charge coupled devices) son circuitos integrados que transforman señales luminosas en eléctricas en un rango de espectro visible, infrarrojo y ultravioleta. El funcionamiento de ZTF se describe en los siguientes pasos [3]:

- Se abre el obturador del telescopio por 30 minutos, capturando toda la luz proveniente de las fuentes desde el espacio.
- La luz pasa por la atmósfera llegando a la cámara del ZTF. Se pueden presentar ciertas distorsiones debido a efectos físicos de la luz.
- Se llena con carga los sensores de cada píxel de los CCDs de manera proporcional a la intensidad recibida.
- Se cierra el obturador, las cargas se traducen en una matriz de números enteros.
- Se apunta a otra región del espacio y se repite el proceso.

La luz proveniente de los objetos espaciales se distribuye en los pixeles siguiendo una distribución llamada Point Spread Function (PSF), la cual describe la respuesta del impulso

entre la fuente y los datos capturados del sensor. Dependiendo de las condiciones de captura de los datos, esta distribución puede ser más extendida a condiciones peores, por lo cual ZTF se encarga de revertir este efecto antes de la entrega de los datos en tiempo real.

Los eventos capturados por ZTF son traducidos en alertas en tiempo real, las cuales están compuestas por tres imágenes del lugar de observación y condiciones sobre las cuales se tomó el evento. Las imágenes obtenidas son las siguientes:

- *Science*: Corresponde a la imagen tomada en el momento del evento.
- *Reference*: Stack de exposiciones representativas que se tomó con anterioridad. Da una estimación de cómo era la zona del evento antes de que se detectara.
- *Difference*: Diferencia entre las imágenes anteriores. Muestra solamente los cambios que hay entre ambas imágenes, o sea, el objeto de interés.

Las imágenes obtenidas son recortadas con su centro en el punto de interés, y también se filtran ciertas alertas que no pasan el filtro de alerta de ZTF: La diferencia en la detección debe ser mayor o igual a  $5\sigma$  en las coordenadas ecuatoriales o en el flujo respecto al cielo de referencia, donde  $\sigma$  corresponde a la desviación estándar de la respectiva medición.

Las muestras que pasan el filtro, son recortadas en  $63 \times 63$  píxeles centradas en el objeto, los cuales pasan a llamarse *stamps* debido a su tamaño reducido; se le asocia posteriormente un identificador único del objeto, lo cual servirá posteriormente para identificar más detecciones del mismo objeto. A las imágenes se agregan metadatos sobre las condiciones de observación, la cual incluye la posición de observación, la magnitud detectada, el error de magnitud, etc. para luego ser enviado mediante ZTF Alert Distribution System (ZADS) a los distintos centros para su análisis.

### 2.1.3. Objetos astronómicos

Los objetos astronómicos corresponden a entidades físicas o fenómenos naturales que se presentan en el universo. De estos objetos, existen algunos que pueden emitir luz o reflejarla de otra fuente cercana; esta característica permite que puedan ser detectados mediante telescopios, los cuales están generalmente localizados en la tierra, con algunas excepciones dentro del espacio. Estos objetos detectados pueden ser de varios tipos, como son estrellas, agujeros negros, asteroides, galaxias, planetas, etc. La detección de estos permite conocer sus características más allá del tipo al que pertenecen, entre los que se cuentan la composición del objeto observado, tamaño de este, posición dentro del universo, etc.

Los distintos tipos de objetos astronómicos que existen se puede agrupar en grupos que permitan su clasificación de forma más sencilla y no tan granulada. La taxionomía utilizada por ALERCE se divide en la de clasificación temprana (utilizada en este trabajo) y mostrada en la figura 2.4, y en la clasificación tardía mostrada en la figura 2.3. La clasificación temprana define 5 clases generales para la clasificación de los objetos: Supernova (SN), AGN (del inglés Active Galactic Nuclei), VS (de sus siglas Variable Star), asteroides y bogus. Se presenta un ejemplo de las imágenes obtenidas de cada clase en la figura 2.4. El motivo principal de trabajar con estas clases es debido a que se toma como base el trabajo presentado en [10],

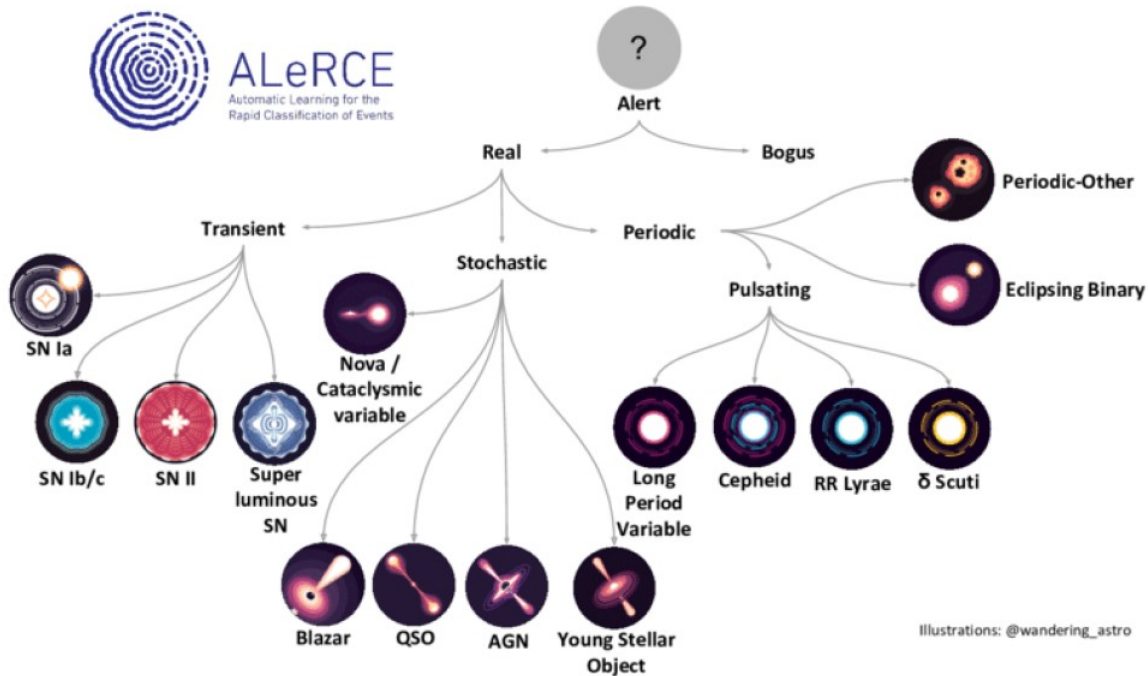


Figura 2.3: Taxonomía utilizada en ALeRCE. Figura tomada de [9].

cuya hipótesis plantea que es posible clasificar objetos astronómicos antes de poder utilizar la curva de luz, la cual requiere de 6 puntos en la misma banda para poder formarse. Bajo esto, se puede describir de forma más detallada cada clase:

- **AGN:** Las galaxias tienden a girar en torno a un agujero negro central supermasivo. En ciertas ocasiones, ocurre que la materia de la galaxia cae en el agujero negro, provocando el consumo de la materia, desgarrado por la gravedad y por lo consiguiente, una emisión de luz. Al no ser uniforme estos consumos, el brillo emitido por estos objetos tiene un comportamiento estocástico, por lo cual las alertas generadas por estos mismos presentan distintos valores de brillos distintos y sin patrón en sus alertas, como también una diferencia en tiempo entre sus alertas estocásticas. Por otro lado, estos objetos tienden a ubicarse en el centro de sus galaxias anfitrionas; en el caso de no observarse una galaxia en las imágenes, tienden a tener una baja densidad de estrellas cercanas.
- **SN:** Corresponde a una de las etapas finales de vida de una estrella supermasiva, en la cual la estrella pierde su equilibrio entre su fuerza gravitatoria y la energía interna, produciendo la explosión de la misma y liberando una gran cantidad de energía, mucha de la cual se transforma en luz. Debido a esto, estos objetos son de carácter transitorio y espontáneos, por lo cual se espera que en su imagen de referencia no se encuentre luminosidad y en su imagen de diferencia y ciencia se encuentre una luminosidad progresiva en sus detecciones; esta puede aumentar en sus detecciones, disminuir (si se detectó en el momento posterior a su explosión) o aumentar para luego disminuir. Al ser estrellas que explotan, tienden a encontrarse cerca de otras estrellas y cercanas a galaxias, que



se puede encontrar tanto en sus imágenes como también en sus metadatos referentes a estas. Al ser una clasificación amplia, existen algunas subclasificaciones de las Supernovas que se han estudiado a profundidad, como el tipo Ia [42, 43] y el tipo II [43, 44], desarrollando modelos astrofísicos que explican su comportamiento.

- **VS:** Tal como su nombre lo indica, corresponden a estrellas activas que poseen un comportamiento variable en cuanto a su luminosidad en el tiempo, pero que poseen un periodo asociado a este cambio. Los ejemplos más comunes de este tipo de objetos son las estrellas pulsantes, donde la luminosidad de esta, y por lo tanto la luz detectada, es cambiante debido a que su diámetro varía de forma periódica; también se tiene el caso de las estrellas binarias, las cuales giran en torno a un eje común, observándose un cambio de luminosidad visto desde nuestra posición. Debido a esto, se espera que las diferencias temporales entre las detecciones sean relativamente parecidas entre ellas, como también que se encuentre luminosidad en la imagen de ciencia y de referencia. Al ser estrellas, tienden a encontrarse en densidades de estrellas, como también en posiciones cercanas a la Vía Láctea o galaxias cercanas, por lo cual deberían tener una distancia pequeña con respecto al plano de la galaxia. Existen actualmente estudios que se han enfocado en algunos tipos de estas estrellas, entre las que se cuentan [45, 46]
- **Asteroides:** Corresponden a objetos que se encuentran en movimiento en el sistema solar compuestos generalmente de materiales rocosos y con tamaños muy distintos. Debido a que se encuentran en movimiento, la posición de estos varía en cada instante, por lo cual solo se debería tener una detección de estos objetos. Al analizar sus imágenes, estas deberían presentar cuerpos extendidos debido al movimiento de estos cuerpos, como también presencia de estos objetos solo en las imágenes de ciencia y diferencia. Al presentarse en el sistema solar únicamente, las posiciones de estos objetos deberían rondar en torno a la latitud 0 en coordenadas elípticas.
- **Bogus:** Nombre asignado a alertas falsas, las cuales pueden ser alertas producidas por el telescopio pero que no tienen un carácter físico interesante, como por ejemplo píxeles saturados, errores astrométricos, formación defectuosa de la imagen, columnas de píxeles malos, restas desalineadas en cálculo de la imagen de diferencia, etc. También se pueden incluir en este grupo objetos que no son de interés astrofísico, como son rayos cósmicos o satélites artificiales.

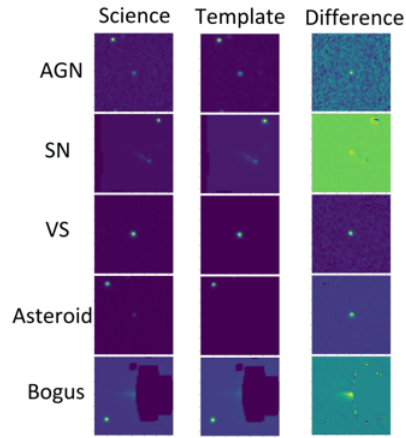


Figura 2.4: Ejemplo de las imágenes *stamps* de las detecciones en las cinco clases. Figura de creación propia.

## 2.2. Aprendizaje de máquinas

Corresponde al área de la computación que describen una serie de algoritmos que permiten detectar estructuras y relaciones entre datos a partir de estos mismos dentro de un conjunto de datos. Una de las formas de describir esta área es modelos que son ajustados a partir de los datos con los cuales se entrena. En esta sección se mencionarán las técnicas y modelos del aprendizaje de máquinas que se utilizaron en el desarrollo de este trabajo.

## 2.3. Problema de clasificación

Corresponde a una de las tareas más recurrentes dentro del aprendizaje de máquinas. Se busca encontrar un mapeo entre las variables de entrada  $X$  y las variables de salida  $Y$ , que corresponden a variables categóricas usualmente enteras  $y_i \in 1, \dots, C$ , donde  $C$  corresponde al número de categorías dentro del conjunto de datos  $\mathbb{D}$ . Este problema se puede describir utilizando la siguiente ecuación:

$$\hat{y}_i = \hat{f}(x_i) = \underset{c \in 1, \dots, C}{\operatorname{argmax}} p(y_i = c | x_i; \mathbb{D}, \theta), \quad (2.1)$$

donde la función a ser aproximada  $\hat{f}()$  corresponde a una función de densidad de probabilidad  $p$  que describe el comportamiento de la variable de salida  $y_i$  condicionada al vector de entrada  $x_i$ , al conjunto de datos  $\mathbb{D}$  con los cuales se ajustó y aproximó el modelo  $\theta$ . El modelo  $\theta = [\theta_1, \dots, \theta_M]^T$  corresponde a un conjunto de  $M$  parámetros que definen el modelo, los cuales son ajustados por un proceso de optimización hasta un valor óptimo  $\theta^*$ . La forma de ajustar este modelo es actualizando sus parámetros a partir de los datos con los cuales se ajusta, este proceso se conoce como aprendizaje a partir de los datos, por lo cual se acuña el término de aprendizaje de máquinas.

Los parámetros de  $\theta$  dependen del tipo de modelo con el cual se esté trabajando, el cual puede ser una red neuronal, como se explicará en los próximos puntos, como también métodos clásicos y estadísticos, entre los que se cuentan Support Vector Machine, Bayes Naive, Procesos gaussianos y, en particular, Random Forest [47]. Este último modelo tiene una especial

importancia debido a que ha sido altamente utilizado en el contexto de astronomía debido a su facilidad de uso, excelentes resultados y la capacidad de poder estimar las características más importantes para realizar la clasificación. Algunos ejemplos de estos trabajos son [48–53].

## 2.4. Redes neuronales artificiales

La idea de desarrollar modelos de inteligencia artificial está principalmente inspirada en la inteligencia de los seres vivos, siendo ésta una forma de simular lo que un animal o humano razona. La idea base para poder simular la inteligencia es usar modelos que se inspiren en cómo funciona el cerebro animal, el cual está compuesto principalmente de neuronas que, mediante sinapsis, producen los pensamientos y razonamiento de los seres vivos. Las redes neuronales artificiales son un modelo matemático basado en esta idea y que, sumando una serie de operaciones simples, permiten modelar operaciones no lineales y con una gran adaptabilidad.

### 2.4.1. Modelo del perceptrón

Corresponde al modelo básico que modela matemáticamente las células neuronales y sus interacciones, propuesto inicialmente por Rosenblatt en el año 1958. Este modelo se define con las siguientes ecuaciones:

$$y = \phi(x_1\omega_1 + \dots + x_N\omega_N), \quad (2.2)$$

$$y = \phi(\omega^T x), \quad (2.3)$$

donde  $y$  es la salida del modelo,  $x = [x_1, \dots, x_N]^T$  es el vector de entrada,  $\omega = [\omega_1, \dots, \omega_N]^T$  son los pesos del modelo, cada uno asociado con una componente del vector de entrada y  $\phi$  corresponde a una función de activación, que se aplica posteriormente a la multiplicación de los vectores y permite llevar la salida lineal a un espacio no lineal [54, 55]. Además de esto, es posible agregar un sesgo al modelo, con lo cual se desplaza la salida de este mismo en alguna proporción y es favorable para datos con sesgos reconocidos, con lo cual la ecuación (2.3) se transforma a la siguiente expresión:

$$y = \phi(\omega^T x + b). \quad (2.4)$$

Los pesos  $\omega$  y el sesgo  $b$  son parámetros ajustables, que se van adaptando a los datos en el entrenamiento. Un diagrama que resume el funcionamiento del perceptrón se puede apreciar en la figura 2.5.

### 2.4.2. Capa Fully-connected

El modelo *Fully-connected* corresponde a una extensión del modelo del perceptrón, donde se trabaja con una colección de  $M$  perceptrones, de tal manera que todos los perceptrones  $M$  se conectan con la capa fully connected anterior de  $N$  perceptrones mediante sus respectivos pesos. De esta forma, la salida en la posición  $i$  en un modelo *Fully-connected* se define como:

$$y_i = \phi(\omega_i^T x), \forall i \in 1, \dots, M; \quad (2.5)$$

y de forma general, para todas las posiciones, queda:

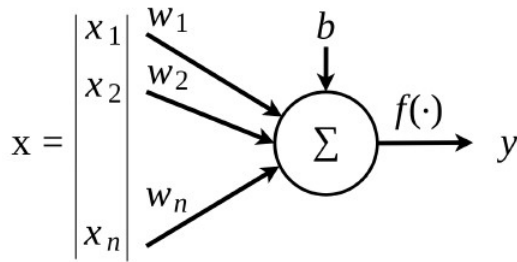


Figura 2.5: Modelo del perceptrón.

$$y = \phi(Wx), \quad (2.6)$$

donde  $W = [\omega_1, \dots, \omega_M]^T \in \mathbb{R}^{M \times N}$  es la matriz de pesos del modelo,  $x \in \mathbb{R}^M$  es la entrada del modelo,  $y \in \mathbb{R}^M$  es la salida del sistema y  $\phi$  corresponde a una función de activación, que generalmente toma una estructura no lineal. Al igual que el caso del perceptrón, es posible agregar un sesgo a este modelo  $b \in \mathbb{R}^M$ , quedando la notación de la siguiente manera:

$$y = \phi(Wx + b). \quad (2.7)$$

### 2.4.3. Perceptrón Multicapa (MLP)

La conexión causal y secuencial de varias capas *fully-connected* es lo que se denomina Perceptrón Multicapa (MLP; Rumelhart et al., 1986), donde estas capas se conectan en forma de pila, o sea, la salida de la capa  $i$  y es la entrada de la capa  $i + 1$ . Considerando un vector de entrada  $x \in \mathbb{R}$ , el vector de salida  $y \in \mathbb{R}^M$  se expresa de la siguiente manera:

$$y = f_{\theta_{NL}} \circ \dots \circ f_{\theta_1}(x), \quad (2.8)$$

donde  $f_{\theta_1}, \dots, f_{\theta_{NL}}$  son  $N$  capas *fully-connected* distintas que componen un modelo MLP. También se puede denotar este modelo de forma parametrizada, o sea,  $y = f_{\theta}(x)$ , donde  $\theta = \text{cat}[\theta_1, \dots, \theta_{NL}]$  es el vector que contiene todos los parámetros actualizables del modelo [56], con *cat* denotando la función concatenación. La conexión entre las capas *Fully-connected*, formando el modelo MLP se muestra de manera gráfica en la figura 2.6.

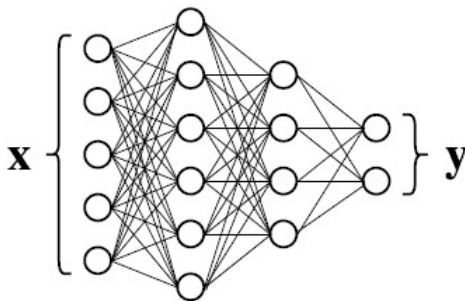


Figura 2.6: Modelo del perceptrón multicapa.

## 2.4.4. Redes convolucionales

El modelo del perceptrón multicapa ha demostrado tener una gran capacidad para encontrar patrones y relaciones entre las características de los datos con los que se entrena, pero tiene sus limitaciones dadas principalmente por su entrada vectorial; esto produce que no considere relaciones temporales o espaciales entre los datos. Bajo este problema, se diseñaron las redes convolucionales [12], las cuales están fuertemente basadas en la visión humana, la cual realiza convoluciones de lo que vemos para detectar patrones como esquinas, cambios de relieve, etc.

Tal como su nombre lo indica, estas redes realizan convoluciones en los datos de entrada, los cuales son generalmente imágenes; una de las técnicas más utilizadas es aplicar un filtro adaptativo, que es ajustado durante el entrenamiento, a una imagen de entrada. Considerando una imagen  $x$ , un filtro  $W$ , un sesgo  $b$  y una imagen de salida  $y$ , el proceso de convolución se puede escribir como:

$$y_{i,j,k} = \sum_{m,n,p} x_{i-m,j-n,p} \cdot W_{m,n,p,k} + b_k, \quad (2.9)$$

donde  $x$  e  $y$  son tensores de 3 dimensiones,  $i, j$  son coordenadas dentro de las imágenes,  $m, n$  se mueven dentro del filtro a aplicar y  $p$  es el índice que se mueve en la profundidad (o canales) de la imagen de profundidad  $k$ . El filtro se mueve sobre la imagen saltándose una cantidad definida de pixeles, que es definida por el usuario, llamado *stride*. El filtro aplicado se define por 4 dimensiones: altura, ancho, número de canales de entrada y número de canales de salida (también llamada cantidad de filtros), donde generalmente se tiende a tomar un pequeño tamaño para la altura y el ancho, y una mayor cantidad para la cantidad de filtros. Una vez aplicada la convolución, se suele agregar una función de activación no lineal, con el objetivo de trasladar el espacio de salida de la capa permitiendo una mejor representación de las características de la imagen. Un ejemplo de convolución sobre una imagen de  $4 \times 4$  de un filtro de  $2 \times 2$  se puede apreciar en la figura 2.7.

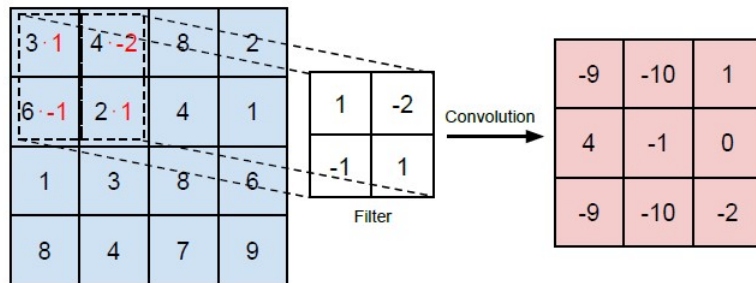


Figura 2.7: Ejemplo del proceso de convolución usando stride, y con un solo canal de entrada y salida, usando un filtro de  $2 \times 2$ .

El modelo convolucional ha sido aplicado en una serie de tareas, entre otras a datos con dependencias espaciales como son las imágenes [14, 15], como también a datos con dependencias temporales como procesamiento de audio [57, 58]. En astronomía, ha sido aplicado para la separación entre datos reales y errores (real/bogus) [20, 59], computación de fotometría [60], cálculo de la imagen de diferencia [61], detección de ondas gravitacionales [62] y detección de exoplanetas [63], entre otros.

## 2.5. Redes neuronales recurrentes

Corresponde a un tipo de redes neuronales diseñadas específicamente para procesar secuencias temporales de datos o series de tiempo [64]. Su funcionamiento o fundamento se basa en una propiedad de las cadenas de Markov, donde el estado  $h_{i,j}$  de un sistema es posible inferirlo a partir del estado anterior a esta  $h_{i,j-1}$ , y por consiguiente, a partir de todos los estados anteriores, con lo cual es posible describir la relación de los modelos RNN como:

$$h_{i,j} = f(x_{i,j}, h_{i,j-1}) \quad y_{i,j} = g(h_{i,j}), \quad (2.10)$$

donde  $h_{i,j}$  es el vector de estado que representa el sistema dinámico, modelado mediante la función  $f(\cdot)$  que toma como entrada la entrada del sistema en el tiempo actual  $x_{i,j}$  y el estado en el tiempo inmediatamente anterior  $h_{i,j-1}$ . Finalmente, la salida del sistema es modelado mediante la función  $g(\cdot)$ .

### 2.5.1. Modelo de recurrencia simple

Definir una red recurrente se basa en encontrar funciones  $f(\cdot)$  y  $g(\cdot)$  adecuadas. La aproximación mas simple de estas funciones es usar la misma idea de las capas MLP:

$$h_{i,j} = f(x_{i,j}, h_{i,j-1}) = \phi(W_x^T x_{i,j} + b_x + W_h^T h_{i,j-1} + b_h), \quad (2.11)$$

$$y_{i,j} = g(h_{i,j}) = h_{i,j}. \quad (2.12)$$

Esta formulación implica que la salida del sistema corresponde al estado del sistema, como también que el estado depende directamente de la entrada al sistema y del estado anterior. Tanto la entrada como el estado anterior son ponderados por vectores de peso  $W_x^T$  y  $W_h^T$  respectivamente, además de poseer un sesgo entregado por los vectores  $b_x$  y  $b_h$ . A la combinación de estos valores se le aplica una función no lineal  $\phi$  para poder modelar las interacciones no lineales entre los vectores. En la figura 2.8 se presenta esta dependencia de estados y formulación matemática.

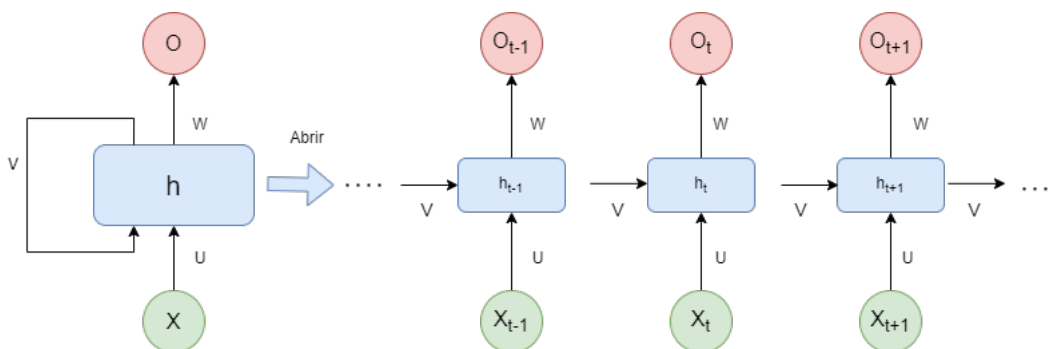


Figura 2.8: Modelo de red recurrente simple.

La formulación planteada posee un buen comportamiento en secuencias temporales relativamente cortas, pero sufre de dos problemas al momento de aplicarlo en secuencias pro-

longadas: desvanecimiento del gradiente y explosión del gradiente. El primer fenómeno se produce cuando los valores que se obtienen al computar el gradiente disminuyen drásticamente, acercándose a 0, con lo cual los cambios en los pesos del modelo cambian en un factor muy pequeño, por lo cual se cae en un mínimo local en la función de pérdida. Caso contrario se produce con la explosión del gradiente, donde el valor de este crece tanto debido a la derivación por regla de la cadena, que los pesos no pueden adaptarse, el valor de la función de costo crece, llegando hasta el overflow computacional, con lo cual el entrenamiento deja de funcionar.

## 2.5.2. Long Short-Term Memory Model (LSTM)

El modelo LSTM corresponde a un tipo de RNN que intenta evitar el problema del desvanecimiento del gradiente y la explosión del gradiente [65–67]. Este modelo asume que existe una celda de memoria que aprende como manejar la información temporal, aprendiendo qué información debe ser leída y guardada en cada paso temporal, en un estado interno  $h_{i,j}$ . También aprende qué información debe olvidarse de  $h_{i,j}$  y qué parte de esta información debe mostrarse en la salida  $y_{i,j}$ . Todo esto se encuentra descrito por las siguientes ecuaciones, siendo resumidas en la figura 2.9:

$$i_{i,j} = \sigma(W_{xi}^T x_{i,j} + b_{xi} + W_{hi}^T h_{i,j-1} + b_{hi}), \quad (2.13)$$

$$f_{i,j} = \sigma(W_{xf}^T x_{i,j} + b_{xf} + W_{hf}^T h_{i,j-1} + b_{hf}), \quad (2.14)$$

$$o_{i,j} = \sigma(W_{xo}^T x_{i,j} + b_{xo} + W_{ho}^T h_{i,j-1} + b_{ho}), \quad (2.15)$$

$$\hat{c}_{i,j} = \phi_{\tanh}(W_{x\hat{c}}^T x_{i,j} + b_{x\hat{c}} + W_{h\hat{c}}^T h_{i,j-1} + b_{h\hat{c}}), \quad (2.16)$$

$$c_{i,j} = f_{i,j} \odot c_{i,j-1} + i_{i,j} \odot \hat{c}_{i,j}, \quad (2.17)$$

$$y_{i,j} = h_{i,j} = o_{i,j} \odot \phi_{\tanh}(c_{i,j}). \quad (2.18)$$

donde  $\phi_{\tanh}$  y  $\sigma$  corresponden a las funciones tangente hiperbólica y sigmoide,  $\odot$  al producto punto. De forma más detallada, la capa LSTM y su notación se explica de la siguiente manera:

- Compuerta de entrada (2.13): La compuerta entrada de la red, representada por  $i_{i,j}$ , es computada utilizando la entrada a la red  $x_{i,j}$  y el estado anterior  $h_{i,j-1}$ , donde cada uno de estos es operado mediante una capa *fully-connected*, con sus respectivas matrices de pesos  $W_{xi}$  y  $W_{hi}$ , como también sus vectores de sesgo  $b_{xi}$  y  $b_{hi}$ . A la salida de esta operación se le aplica la función  $\sigma$ , la cual tiene como resultado que los valores  $i_{i,j}$  se encuentren en el rango  $[0, 1]$
- Capa de olvido (2.14): Es representada por  $f_{i,j}$ , y utiliza la misma lógica para su computación que la compuerta de entrada, utilizando los mismos vectores de entrada, pero con

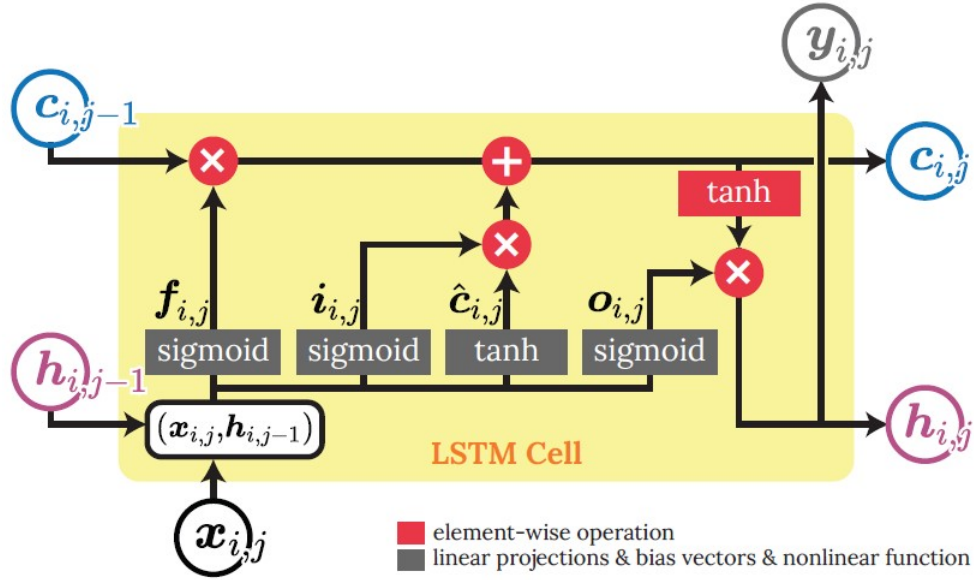


Figura 2.9: Modelo LSTM. Figura tomada de [26].

sus propias matrices de pesos  $W_{xf}$  y  $W_{hf}$  y sus vectores de sesgo  $b_{xf}$  y  $b_{hf}$ . Su salida también se encuentra en el rango  $[0, 1]$  debido a la aplicación de la función  $\sigma$ .

- Compuerta de salida (2.15): Es representada por  $o_{i,j}$ , y utiliza la misma lógica para su computación que la compuerta de entrada y de olvido, utilizando los mismos vectores de entrada, pero con sus propias matrices de pesos  $W_{xo}$  y  $W_{ho}$  y sus vectores de sesgo  $b_{xo}$  y  $b_{ho}$ . Su salida también se encuentra en el rango  $[0, 1]$  debido a la aplicación de la función  $\sigma$ .
- Vector de memoria candidato (2.16): Este vector es computado a partir de la entrada actual a la red  $x_{i,j}$  y el estado anterior  $h_{i,j-1}$ . Cada entrada es computada mediante capas *fully connected*, con sus respectivas matrices de pesos  $W_{x\hat{c}}$  y  $W_{h\hat{c}}$  y sus vectores de sesgo  $b_{x\hat{c}}$  y  $b_{h\hat{c}}$ . A diferencia de las compuertas anteriores, la función de activación para este vector es una tangente hiperbólica  $\phi_{\tanh}$ , lo cual hace que los valores de salida estén en el rango  $(-1, 1)$ .
- Vector de memoria (2.17): Vector que contiene información pasada y actual de las entradas al sistema. Es computado utilizando el vector de memoria candidato  $\hat{c}_{i,j}$ , el cual es operado punto a punto con la compuerta de entrada; como la compuerta de entrada toma valores entre  $[0, 1]$ , un valor más cercano a 0 implica una poca importancia del valor actual, mientras que más cercano a 1 implica una alta importancia de este mismo. Por otro lado, se agrega la suma del término  $f_{i,j} \odot c_{i,j-1}$ , siendo  $f_{i,j}$  valores entre  $[0, 1]$ , por lo cual un valor cercano a 0 también significa una menor importancia de la historia de la entrada actual (entradas pasadas), y un valor cercano a 1 implica una mayor importancia de los valores anteriores de la entrada.



- Vector de estado y salida (2.18): La salida de la red LSTM corresponde al estado del sistema, el cual es computado a partir de la memoria calculada en el punto anterior, teniendo una activación  $\phi_{\tanh}$ , lo cual mueve los valores a un rango  $(-1, 1)$ . Este último es atenuado o resaltado a partir de la operación punto a punto con  $o_{i,j}$ , la cual se comporta de igual manera que para el caso del vector de memoria.

Como consideración final, en el modelo LSTM se utilizan distintas inicializaciones para los pesos existentes. En el caso del vector de memoria  $c_{i,j}$  y del vector de estado  $h_{i,j}$  se utiliza una inicialización nula, o sea,  $c_o = h_o = \vec{c}$ . Por otra parte, los vectores de sesgo  $b_x$  se inicializan con un vector de  $\vec{1}$ , de forma de que en las primeras inicializaciones del modelo, no se olvide toda la información asociada a las muestras.

Este modelo ha sido utilizado en diversas aplicaciones con dependencias temporales con muy buenos resultados. En particular, han sido utilizadas para el procesamiento del lenguaje natural [68], traducción entre idiomas [69], reconocimiento de voz [70]. En astronomía, ha sido utilizado para la computación fotométrica [60] y la clasificación de secuencias de imágenes de objetos astronómicos [23]. Por otro lado, es posible combinar este modelo con las capas convolucionales, de manera de procesar información temporal en conjunto con espacial, lo cual ha tenido aplicaciones en el procesamiento de videos [71–73] y reconocimiento de voz [74].

### 2.5.3. Modelo Gated Recurrent Unit (GRU)

Este modelo fue presentado como una alternativa para el procesamiento recurrente [75, 76], la cual representa una simplificación al modelo LSTM al reducir el número de operaciones y, por lo tanto, el número de parámetros a ajustar durante el entrenamiento, lo cual lo vuelve menos compleja y más liviana computacionalmente.

Las ecuaciones que definen el modelo GRU se definen a continuación, las cuales son resumidas en la figura 2.10:

$$z_{i,j} = \sigma(W_{xz}^T x_{i,j} + b_{xz} + W_{hz}^T h_{i,j-1} + b_{hz}), \quad (2.19)$$

$$r_{i,j} = \sigma(W_{xr}^T x_{i,j} + b_{xr} + W_{hr}^T h_{i,j-1} + b_{hr}), \quad (2.20)$$

$$g_{i,j} = \phi_{\tanh}(W_{xg}^T x_{i,j} + b_{xg} + r_{i,j} \odot (W_{hg}^T h_{i,j-1} + b_{hg})), \quad (2.21)$$

$$y_{i,j} = h_{i,j} = z_{i,j} \odot h_{i,j-1} + (1 - z_{i,j}) \odot g_{i,j}, \quad (2.22)$$

donde  $\phi_{\tanh}$  corresponde a la función tangente hiperbólica y  $\sigma$  a la función sigmoide; el operador  $\odot$  corresponde al producto punto. El modelo GRU, junto a sus ecuaciones, puede ser explicado de la siguiente manera:

- Compuerta de actualización (2.19): Utiliza el vector de entrada  $x_{i,j}$ , como también el estado anterior  $h_{i,j-1}$  para ser computado. Utiliza una capa *fully connected* para operar,

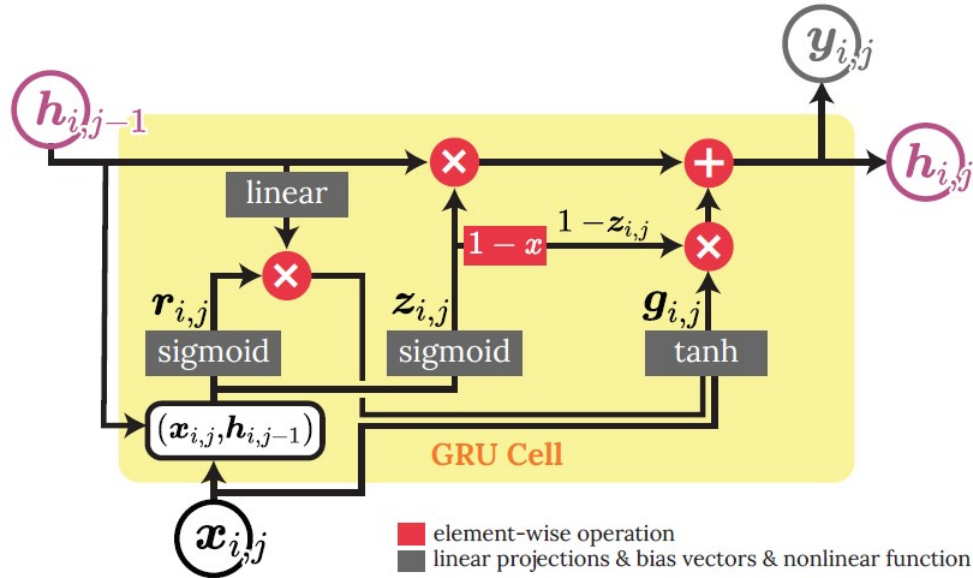


Figura 2.10: Modelo GRU. Figura tomada de [26].

con sus matrices  $W_{xz}$  y  $W_{hz}$ , y sus sesgos  $b_{xz}$  y  $b_{hz}$ . Debido a la activación  $\sigma$ , los valores de salida se encuentran en el rango  $[0, 1]$ .

- Compuerta de reseteo (2.20): Utiliza el vector de entrada  $x_{i,j}$ , como también el estado anterior  $h_{i,j-1}$  para ser computado. Utiliza una capa *fully connected* para operar, con sus matrices  $W_{xr}$  y  $W_{hr}$ , y sus sesgos  $b_{xr}$  y  $b_{hr}$ . Debido a la activación  $\sigma$ , los valores de salida se encuentran en el rango  $[0, 1]$ .
- Vector de memoria candidato (2.21): Utiliza el vector de entrada  $x_{i,j}$ , como también el estado anterior  $h_{i,j-1}$  para ser computado. Utiliza una capa *fully connected* para operar, con sus matrices  $W_{xg}$  y  $W_{hg}$ , y sus sesgos  $b_{xg}$  y  $b_{hg}$ . Debido a la activación  $\phi_{\tanh}$ , los valores de salida se encuentran en el rango  $(-1, 1)$ . El vector  $r_{i,j}$  controla cuanto del vector  $h_{i,j-1}$  debe conservarse para la computación de  $g_{i,j}$ .
- Vector de estado y salida (2.22): En el caso del modelo GRU, la salida del sistema  $y_{i,j}$  es igual al estado actual del sistema  $h_{i,j}$ . Este estado se computa a partir de dos operaciones: La primera operación pondera el estado anterior del sistema, atenuando si  $z_{i,j}$  es cercano a cero. El segundo término atenúa o ignora completamente valores provenientes de  $g_{i,j}$  a partir del vector  $(1 - z_{i,j})$ . La lógica detrás de esto se puede describir a continuación para una componente  $k$ :
  1. Cuando el valor de  $z_{i,j,k}$  es cercano a 1, el valor de  $h_{i,j,k}$  se encuentra mayormente dominado por el valor de su estado anterior, o sea, por  $h_{i,j-1,k}$ .
  2. Cuando el valor de  $z_{i,j,k}$  es cercano a 0, el valor de  $h_{i,j,k}$  se encuentra mayormente dominado por el vector de memoria  $g_{i,j,k}$ .

A partir de lo mencionado anteriormente, es posible afirmar que el modelo GRU mezcla de forma simultánea las compuertas de entrada y olvido utilizando la compuerta de reseteo, lo cual la vuelve más ligera computacionalmente, alcanzando resultados comparables a los obtenidos con LSTM [64]. Este modelo ha sido aplicado en datos con dependencias temporales, tales como la clasificación a partir de audio [77, 78], procesamiento del lenguaje natural [79, 80] y en astronomía, ha sido aplicado en el problema de clasificación de objetos transientes y SuperNovas [81].

## 2.6. Time Delay Neural Network (TDNN)

Modelo propuesto en [82] representando una alternativa basada totalmente en procesamiento *feed-forward* para el procesamiento de muestras con características temporales. La entrada al sistema corresponde a un vector  $x \in R^{m*T}$ , donde  $m$  corresponde al tamaño de cada tiempo, y  $T$  a la cantidad de registros temporales, se procede de la siguiente forma:

- Se define una ventana de tamaño  $h$ .
- Se toma los registros  $x_{i,0}$  hasta  $x_{i,h}$ , los cuales son procesados por una capa *fully connected*, obteniéndose una salida  $y_{1,i,0}$ , la cual se toma como la capa 2.
- Se mueve la ventana un espacio, tomando ahora desde  $x_{i,1}$  hasta  $x_{i,h+1}$ , y esta se procesa por una capa *fully connected*, obteniéndose la salida  $y_{1,i,1}$ , siendo el segundo registro de la capa 2. Se continúa moviendo un espacio hasta alcanzar por el último registro temporal inicial.
- Se vuelve a repetir el proceso anterior, pero tomando ahora la capa 2 como valores de entrada. Este mismo proceso se repite las veces que sean definidas en el diseño.

Notar que la explicación anterior asume un paso entre ventanas de una muestra temporal, lo cual puede ser modificado. También es posible agregar paddings alrededor de la señal de entrada, de manera de poder abarcar la mayor cantidad de información posible. El proceso explicado se resume de manera gráfica en la figura 2.11. Este modelo ha sido aplicado en procesamiento del lenguaje natural [83], conversión de texto a audio [84] y en reconocimiento de voz [85].

## 2.7. Gamma memory

Red neuronal del tipo recurrente presentada en [86], y que se caracteriza por ser una colección de *Leaky Integrators*. Para definir su comportamiento, se deben definir otros puntos anteriormente:

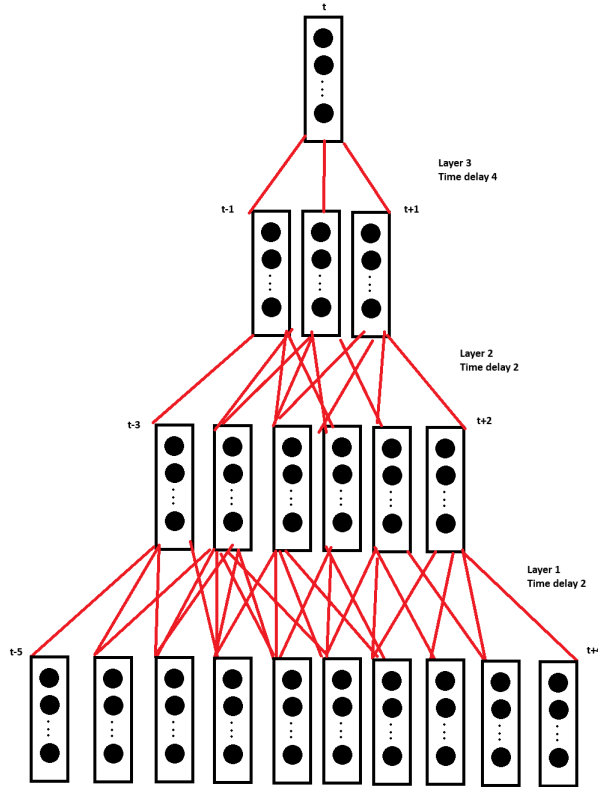


Figura 2.11: Modelo TDNN.

### 2.7.1. Filtro de memoria

Corresponde a un sistema lineal con una respuesta al impulso  $g(t)$  con las siguientes condiciones:

- $g(t)$  es causal, o sea  $g(t) = 0 \quad \forall t < 0$ .
- La suma de todos los valores  $g(t) = 1$ , o sea, se encuentra normalizado

$$\sum_{t=0}^{\infty} |g(t)| = 1. \quad (2.23)$$

De forma consecuente, un filtro de memoria también es un sistema BIBO<sup>1</sup> debido a que la suma de todos sus valores se encuentra acotada. Esto permite definir un estado de memoria  $v(t)$  de la entrada  $u(t)$  como:

$$v(t) = \sum_{k \leq t} g(t - k)u(k). \quad (2.24)$$

<sup>1</sup> Sistema BIBO: Sistema donde si la entrada del sistema se encuentra acotada, entonces la salida de este también se encontrará acotada

## 2.7.2. Leaky Integrator

Aplicación del estado de memoria, donde se cambia la función  $g(t)$  por la siguiente expresión:

$$g(t) = (1 - \mu)\mu^{t-1}u(t - 1), \quad (2.25)$$

donde  $\mu$  es un valor adaptativo, que se ajusta junto al entrenamiento del modelo.

## 2.7.3. Gamma memory

Corresponde a una extensión de la capa *Leaky Integrator*, el cual consiste en una cascada de  $K$  *Leaky integrator*, donde  $K$  es un parámetro elegido arbitrariamente por el diseñador. La principal característica es que todos los *Leaky Integrators* de *Gamma Memory* comparten el mismo parámetro  $\mu$  que se va ajustando en el entrenamiento [86, 87]. Este modelo ha sido utilizado en identificación de sistemas dinámicos para control de sistemas [88] y también para el pronóstico de la calidad del agua [89]. El modelo *Gamma memory*, junto a la representación de una celda de *Leaky integrator* se puede apreciar en la figura 2.12.

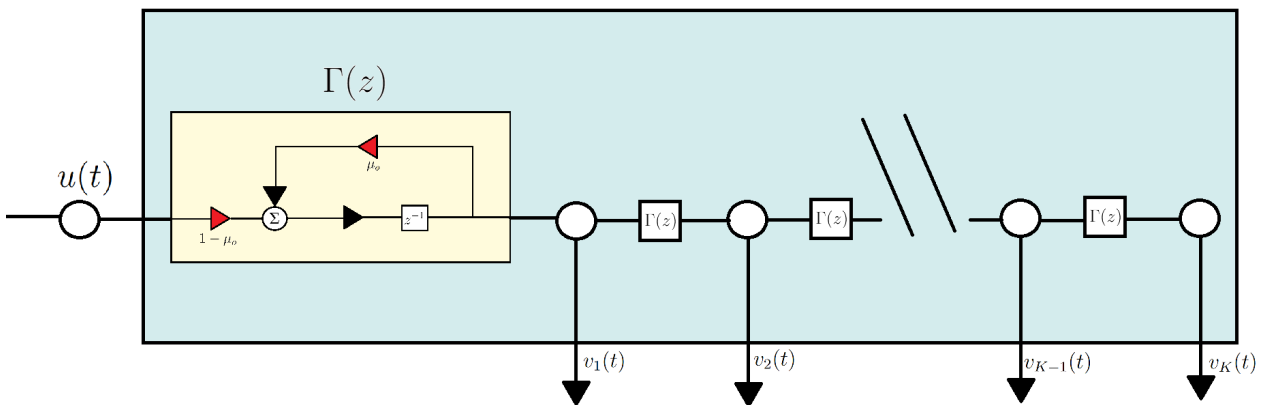


Figura 2.12: Modelo de Gamma Memory.

## 2.8. Transferencia de aprendizaje

La transferencia de aprendizaje (del inglés Transfer Learning) corresponde a una técnica del aprendizaje de máquinas en donde el conocimiento aprendido para resolver un problema es reutilizado como punto de partida para mejorar el desempeño en la resolución de otro problema [90]. La forma más común de utilizar esta técnica es tomar un modelo que ha sido entrenado en una base de datos que sea similar a la base de datos objetivo. A modo de ejemplo, tomar un modelo que usó una base con distintos objetos para luego utilizarlo en un problema de identificación de animales. Se carga el modelo entrenado para luego re entrenar el modelo a partir de los pesos pre cargados pero utilizando la nueva base de datos objetivo. Esta técnica ha demostrado ser muy útil para mejorar el desempeño de modelos y acelerar los tiempos de entrenamiento [91, 92].

## 2.9. Rotaciones Aleatorias

El uso de rotaciones de las imágenes de entrenamiento ha sido altamente utilizado como una manera de aumentar la cantidad de datos de entrenamiento. En el contexto de la astronomía, las rotaciones son utilizadas como una forma de incluir invariancia rotacional, esto debido principalmente a que un objeto detectado en una imagen astronómica no cambia su identificación dependiendo de la orientación que esta tenga. A manera de ejemplo, si se detecta una galaxia en una imagen, esta puede tener cualquier orientación o posición dentro de la imagen. La invariancia rotacional ha sido incluida anteriormente utilizando rotaciones fijas de las imágenes de entrada, para posteriormente obtener un promedio de las características detectadas por las capas convolucionales.

*Random rotation* propone agregar una característica aleatoria a la rotación de las imágenes en cada iteración o batch de entrenamiento. Sea  $X$  una imagen de entrada,  $X_r$  la imagen rotada,  $T(A, \Theta)$  una transformación que rota la imagen  $A$  en un ángulo  $n$ ,  $\theta$  un ángulo máximo sobre el cual variar,  $p$  un número aleatorio entre  $-1$  y  $1$ ,  $w$  el ancho de la imagen y  $h$  la altura de la imagen. La capa de *Random Rotation* se puede definir por la siguiente expresión:

$$\begin{aligned}
 X_r(i, j) &= T(X(i, j), n, \theta) = X(i', j') \\
 i' &= \left(i - \frac{w}{2}\right) \cos(n + \theta \cdot p) + \left(j - \frac{h}{2}\right) \sin(n + \theta \cdot p) \\
 j' &= \left(\frac{w}{2} - i\right) \sin(n + \theta \cdot p) + \left(j - \frac{h}{2}\right) \cos(n + \theta \cdot p)
 \end{aligned} \tag{2.26}$$

En términos simples, se tendrá que al pasar por la capa de *Random Rotation*, la imagen será rotada en un ángulo aleatorio entre  $n - \theta$  y  $n + \theta$ . Esta característica permite que cada vez que la imagen pasa por la capa, sea rotada en un ángulo distinto, pero en torno a un valor definido y con un grado de libertad también definido. En la figura 2.13 se puede apreciar un ejemplo de distintas rotaciones aplicadas a una imagen con un  $\theta = 30^\circ$  y  $n = 0^\circ$ . Cada posición  $i, j$  es una iteración distinta de la imagen original sobre la capa *Random Rotation*. Cabe mencionar que esta técnica solo se aplica en el conjunto de entrenamiento, ya que su característica aleatoria perjudica las métricas de evaluación al utilizarse en los conjuntos de validación y prueba.



Figura 2.13: Ejemplo de aplicación de rotación aleatoria sobre una imagen en el rango  $0^\circ \pm 30^\circ$ . Cada imagen representa una rotación aleatoria aplicada sobre la imagen original.

# Capítulo 3

## Metodología

### 3.1. Mejora a la clasificación temprana de objetos astronómicos

En este punto se explica el trabajo realizado para mejorar el modelo presentado en [10] de clasificación de alertas astronómicas utilizando la primera detección en 5 clases: AGN, SN, VS, Asteroides y Bogus. El modelo presentado en esta sección toma como base el *Stamp Classifier*, agregando rotaciones aplicadas a las imágenes de entrada en ángulos no cuadrados, la inclusión de rotaciones aleatorias en el entrenamiento de los modelos, la modificación en el recorte de la imagen de entrada. Esta modificación del *Stamp Classifier* utiliza datos generados por el sondeo astronómico ZTF. El objetivo de este modelo es poder detectar nuevos objetos utilizando los datos de su primera detección, en especial SNe, y estimar también las probabilidades de clase.

La metodología presentada en esta sección sigue la utilizada en [10] en gran parte de los pasos, siendo la principal diferencia el recorte de las imágenes de entrada, la cual se recorta inicialmente en  $45 \times 45$ , para luego ser recortada en cada iteración en cada una de las rotaciones por separado.

#### 3.1.1. Base de datos

La base de datos utilizada en la mejora de la clasificación temprana corresponde a la base ALeRCE-ZTFv5, siendo esta la misma base utilizada para entrenar el *Stamp Classifier* original. Esta base se compone de la primera detección de distintos objetos astronómicos de las 5 clases nombradas anteriormente. Cada detección posee 3 imágenes (*stamps*) con una resolución de  $63 \times 63$ , donde cada pixel equivale a 1 arco-segundo de declinación del cielo; también posee metadatos que entregan información no visible en la imagen, como la posición del objeto detectado, la intensidad de la luminosidad observada, el error de medición de esta, si existe alguna galaxia cercana, etc.

La cantidad de alertas en la base de datos es de 52.244, divididas entre las clases como: 14.966 (29%) AGN, 1.620 (3%) SN, 14.996 (29%) VS, 9.899 (19%) asteroides y 10.763 (20%) de bogus. Ejemplos de *stamps* de las 5 clases mencionadas se muestran en la figura 3.1.



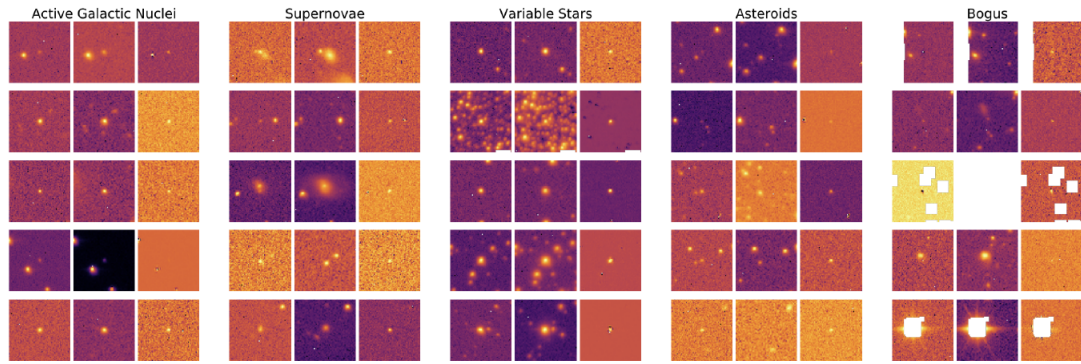


Figura 3.1: Ejemplo de estampillas de la primera detección de las 5 clases consideradas para el modelo *Stamp Classifier*. Figura tomada de [10].

### 3.1.2. Pre-procesamiento

El pre-procesamiento aplicado a la base de datos ALeRCE-ZTFv5 se resume en los siguientes puntos, los cuales están ordenados de forma secuencial.

- **Imágenes:**

1. Descartar alertas no cuadradas.
2. Recortar los stamps a  $45 \times 45$  píxeles.
3. Normalizar cada imagen de cada objeto entre 0 y 1, sin considerar los valores que no sean numéricos (NaN).
4. Reemplazar valores NaN con el valor 0.

- **Metadatos:**

1. Se utilizan 23 metadatos detallados en la tabla A.1 del Anexo A.
2. Se limitan los valores máximos y mínimos de los metadatos. Los límites se encuentran en la tabla A.3 para algunos metadatos; los metadatos que no se encuentran en la tabla, son limitados por el máximo y el mínimo presente en el conjunto de entrenamiento.
3. Se normalizan los datos para que distribuyan como una distribución  $(0, 1)$ , calculando la media ( $\mu$ ) y desviación estándar ( $\sigma$ ) sobre el conjunto de entrenamiento.

De forma más detallada, los motivos de este preprocesamiento se explican a continuación:

- **Imágenes:** Las imágenes provenientes de ZTF tienen una dimensión estándar de  $63 \times 63$ , pero en ciertas ocasiones ocurre que se generan imágenes con dimensiones distintas debido a que una alerta se genera en el borde de la imagen generada por la CCD del telescopio, resultando en imágenes no cuadradas. Debido a que estos casos son excepcionales, se decidió descartarlos de la base de datos si alguna de las imágenes presenta este problema, resultando en la eliminación de 650 ejemplos. Posterior a este filtro, la distribución de las clases queda en 14,742 (29%) AGN, 14723 (29%) VS, 1596 (3%) SNe, 9799 (19%) asteroides y 10734 (20%) Bogus.

Se realiza una normalización de los valores presentes en cada imagen para que sigan una distribución (0,1) sin considerar los valores NaNs de las imágenes. Esto se realiza principalmente para tener valores similares en la entrada del modelo independiente del objeto que se tenga. Debido a saturaciones presentes en los pixeles al momento de capturar el stamp, algunos valores quedan dañados, lo cual se traduce en valores NaNs dentro de la imagen, los cuales son más frecuentes en la clase Bogus; para poder trabajar con ellos en el modelo y que no indeterminen la función de pérdida, se cambian estos valores por 0 con posterioridad a la normalización de la imagen; al otorgarle un valor a estos valores indeterminados, se facilita el procesamiento de la imagen en el sentido que es un valor estándar para todas las muestras.

Imágenes pequeñas tienden a tener mejor rendimiento en modelos de clasificación junto al hecho de que los stamps se encuentran centrados en el punto de interés; por lo cual, se procede a recortar las imágenes en el centro a una dimensión de  $45 \times 45$ . Este tamaño se selecciona de manera de poder realizar rotaciones dentro del modelo que no pierdan ni incluyan información falsa en rotaciones que no son múltiplos de  $90^\circ$ , para posteriormente realizar un recorte en el tamaño a entrenar.

Finalmente, utilizando las imágenes de la primera detección, se arma un cubo con las 3 imágenes de ciencia, referencia y diferencia, con lo cual se tiene una dimensión de entrada de imágenes de  $(45, 45, 3)$ .

- **Metadatos:** Como segunda entrada al modelo se tienen los metadatos de las detecciones, los cuales, corresponden a 23 características (vector) asociadas a cada ejemplo. Para cada metadato se limita su valor como se muestra en la tabla A.3, con la excepción de los datos que no se encuentran en esa tabla, los cuales son limitados por los valores máximos y mínimos presentes en el conjunto de entrenamiento. Cada metadato es normalizado tomando el promedio  $\mu$  y la desviación  $\sigma$ , donde estos estadísticos se calculan utilizando solo el conjunto de entrenamiento.

### 3.1.3. Partición de datos

Se tomaron 100 muestras de cada clase para el conjunto de validación, y 100 muestras por clase para el conjunto de test, ambos de forma aleatoria. Posteriormente, se realiza un sobre muestreo en el conjunto de entrenamiento, con el objetivo de tener igual cantidad de ejemplo para todas las clases en este conjunto; esto permite tener batches balanceados durante el entrenamiento y sobrellevar de mejor manera el problema de desbalanceo entre las clases; obteniendo 14.542 muestras por clase, completando 72.710 muestras en total en el conjunto de entrenamiento. Cabe mencionar que la clase SN se encuentra sobre muestreada, teniendo que repetirse cerca de 7 veces sus muestras. Esto se realiza según lo realizado en [10] el cual se argumenta en evitar el sobreajuste sobre las clases mayoritarias.

### 3.1.4. Arquitectura del clasificador

El modelo de clasificación utilizado en esta sección corresponde al modelo presentado en [10] con modificaciones sustanciales, las cuales consideran un tamaño de entrada superior al original ( $45 \times 45$ ), teniendo un recorte posterior a las rotaciones, donde estas rotaciones pueden ser no ortogonales (no múltiplos de  $\pi/2$ ) y con una componente aleatoria que cambia para cada batch.

Para describir el modelo de rotaciones aleatorias, es necesario definir 3 cantidades que varían según las pruebas realizadas y descritas posteriormente. En primer lugar, se tiene el número de rotaciones  $k$ , las cuales son equiespaciadas, a modo de ejemplo, si se define  $k = 5$ , se tendrá que las imágenes de entrada serán rotadas en  $0, 72^\circ, 144^\circ, 216^\circ$  y  $288^\circ$ . En segundo lugar el grado de libertad para rotar  $\theta$  comentado en la sección 2.6, el cual define cuanta libertad tiene la imagen para rotar; si se tiene un  $k = 5$  y  $\theta = 10^\circ$ , entonces la segunda rotación de la imagen puede ser cualquier valor entre  $62^\circ$  y  $82^\circ$ , este parámetro cambia en cada iteración sobre el modelo y en tercer lugar, el tamaño de la imagen posterior al recorte  $a$ .

Las entradas al modelo *Stamp Classifier* propuesto están compuestas por un cubo con las imágenes provenientes de ZTF de la primera detección: imágenes de ciencia, referencia y diferencia, las cuales son pre apiladas formando el cubo de dimensiones  $45 \times 45 \times 3$  y un vector de 24 metadatos. La parte convolucional del modelo recibe un *batch* de imágenes de la primera detección. Una vez ingresada a la capa convolucional, esta es aumentada con versiones rotadas de sí mismo  $k$  veces; considerando la función  $r = [360/k]^\circ$  que rota la imagen en  $r$  grados, obteniéndose un batch aumentado  $B(x) = [x, rx, \dots, r^{(k-1)}x, r^kx]$ . Cada una de estas rotaciones son recortadas posteriormente a un tamaño  $a$ , quedando para cada rotación una imagen de dimensión  $a \times a \times 3$ , donde estas imágenes recortadas pasan por las capas convolucionales. Se utilizan los parámetros mostrados en la tabla 3.1, donde se detalla el número de filtros, tamaño de la convolución, padding utilizado, dimensiones de entrada y salida. Todas las convoluciones, a excepción de la primera y la última, poseen *zero padding*, con el objetivo de mantener la dimensión de entrada en la salida de la capa, como también poseen funciones de activación *ReLU*, al igual que las capas *Fully Connected*. La salida de la última capa de *Max Pooling* es aplanada, formando los vectores que entran a las capas *Fully Connected*, siendo esta la última capa que trabaja de forma independiente con la rotación. Tomando la salida de esta capa para cada rotación, se apilan cada vector de cada rotación obteniéndose un tensor de  $k \times 64$ , donde se promedia los valores en la dimensión apilada, obteniéndose finalmente un vector de 64 dimensiones, siendo esto el proceso de *Cyclic Pooling*. Posterior al procesamiento de las imágenes, se concatenan los metadatos al resultado aplicando un Batch Normalization a esta nueva entrada, lo cual resulta en un vector de dimensión 87, luego se aplican las capas *Fully Connected*, obteniendo finalmente, gracias a la función de activación *softmax*, las probabilidades estimadas para las 5 clases.

En la figura 3.2 se muestra una representación gráfica de la arquitectura detallada, la cual se ajustó a  $k = 5$ , donde se puede apreciar las rotaciones utilizadas, la entrada de 3 imágenes, metadatos y el ángulo  $\theta$  de libertad para rotar. En la tabla 3.1 se detalla el modelo capa por capa, con los parámetros que definen cada una de ellas, y las dimensiones de salida de cada capa.

Tabla 3.1: Arquitectura del *Stamp Classifier* propuesto.

Capa	Parámetros de la capa	Tamaño de salida
Entrada	-	$45 \times 45 \times 3$
Aumentación por rotaciones	-	$45 \times 45 \times 3$
Recorte de imágenes	-	$a \times a \times 3$
Convolución	$4 \times 4, 32$	$24 \times 24 \times 32$
Convolución	$3 \times 3, 32$	$24 \times 24 \times 32$
Max pooling	$2 \times 2, \text{stride } 2$	$12 \times 12 \times 32$
Convolución	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolución	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolución	$3 \times 3, 64$	$12 \times 12 \times 64$
Max pooling	$2 \times 2, \text{stride } 2$	$6 \times 6 \times 64$
Aplanado	-	2304
Fully connected	$2304 \times 64$	64
Concatenación de rotaciones	-	$k \times 64$
Cyclic pooling	-	64
Concatenación con metadatos tras BN	-	$64 + 21$
Fully connected con dropout [36]	$90 \times 64$	64
Fully connected	$64 \times 64$	64
Fully connected con softmax	$64 \times 5$	5 (nº clases)

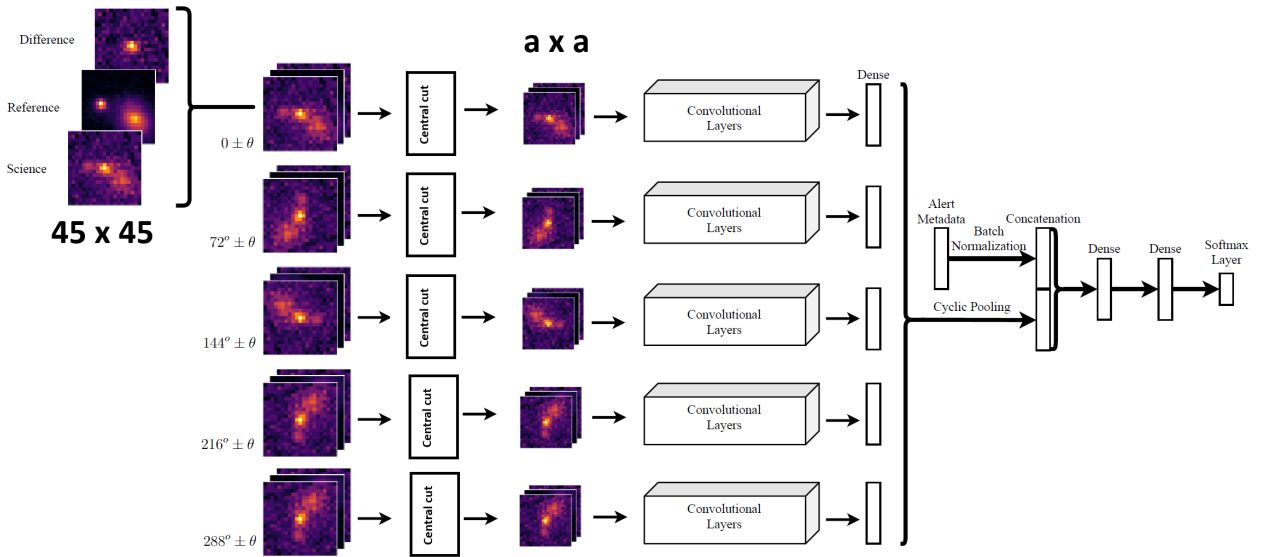


Figura 3.2: Modelo *Stamp Classifier* propuesto, donde  $a$  significa el tamaño del recorte utilizado,  $\theta$  el ángulo de libertad para la rotación aleatoria y considerando 5 rotaciones.

### 3.1.5. Regularización de la entropía

El aumento de la complejidad de las redes neuronales actuales, junto a la inclusión de modelos más complejos que incluyen convoluciones, ha resultado en una pérdida de confianza en las probabilidades entregadas por el modelo. Esto implica que los modelos entrenados tienden a sobreajustarse a las clases que predicen, entregando una probabilidad muy cercana a 1 incluso en las muestras donde el modelo se equivoca. Esto provoca que no se pueda realizar análisis probabilísticos sobre muestras que pueden ser confundidas. Esto es sumamente necesario en el caso del *Stamp Classifier*, debido a que se debe realizar una validación manual de las muestras que son clasificadas como SNe por parte de un experto y la probabilidad predicha ayuda a poder discernir una muestra real.

Para poder mitigar este fenómeno, se pueden utilizar varias técnicas posteriores al entrenamiento, como la regulación de temperatura mostrada en [24], como también operaciones durante el entrenamiento, que consideran la modificación de la función de pérdidas, que es lo ocupado en [10] y retomado en esta tesis. La idea detrás de esta modificación es tomar la función de pérdida típica para la clasificación *Entropía cruzada* y sumarle un término que penalice las probabilidades muy cercanas a 1, obteniéndose la ecuación 3.1:

$$L = - \underbrace{\sum_{c=1}^N y_c \log(\hat{y}_c)}_{\text{Entropía cruzada}} + \underbrace{\beta \sum_{c=1}^N \hat{y}_c \log(\hat{y}_c)}_{\text{Regularización de la entropía}}, \quad (3.1)$$

donde  $N$  es el número de clases,  $y_c$  es la clase en codificación *one-hot*,  $\hat{y}_c$  es la clase predicha en la misma codificación y  $\beta$  corresponde a la constante de regularización, la cual es ajustada como parámetro antes de comenzar el entrenamiento por parte del usuario. Para todos los experimentos de esta tesis, se utilizó  $\beta = 0.5$ , el cual fué escogido para el modelo final de [10] como una elección de expertos. Las diferencias que presenta en los resultados variar este valor no tienen una relevancia estadística [10], por lo cual se mantiene el valor entregado por los expertos y no se explora en este trabajo.

### 3.1.6. Esquema de entrenamiento

Los pesos de toda la red se inicializan de forma aleatoria. Una vez generados los pesos iniciales, se entrena el modelo durante 20.000 iteraciones. En cada iteración se considera el procesamiento de un *batch* de datos, se evaluó el resultado en el conjunto de validación cada 10 iteraciones, y se guardó el menor valor registrado en este conjunto. Posterior a las 20.000 iteraciones, se inicia un contador en 0, donde se va sumando 1 cada vez que se avanza una iteración. Al evaluar el conjunto de validación, si se encuentra un valor menor en la función de pérdida, se reinicia el contador en 0. Si el contador llega a 20.000, se termina el entrenamiento.

Al momento de entrenar el modelo, se utiliza el optimizador Adam [34] para ajustar los pesos del modelo, con los parámetros  $\beta_1 = 0.5$  y  $\beta_2 = 0.9$

### 3.1.7. Experimentos

### 3.1.7.1. Cantidad de rotaciones, rotaciones aleatorias y tamaño de la imagen

Una de las mejoras propuestas para el modelo del *Stamp Classifier* es cambiar la cantidad de rotaciones aplicada a las imágenes de entrada y probar distintas configuraciones de ésta, variando desde 4 rotaciones hasta 8 rotaciones en ángulos equidistantes entre las rotaciones. En conjunto con esto, se propone cambiar la rotación normal sobre un ángulo fijo para todas las imágenes por la rotación aleatoria explicada en la sección 2.6, explorando distintos valores para  $\theta$  en torno a los ángulos equidistantes; a modo de ejemplo, si se tiene el caso de 4 rotaciones en  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  y  $270^\circ$ , y se elige un  $\theta = 10^\circ$ , entonces cada batch rota en un ángulo  $0^\circ \pm n_o 10^\circ$ ,  $90^\circ \pm n_1 10^\circ$ ,  $180^\circ \pm n_2 10^\circ$  y  $270^\circ \pm n_3 10^\circ$ ,  $n_i$  con  $i \in 1, 2, 3, 4$  son números aleatorios entre 0 y 1 que cambia en cada batch, determinando una rotación distinta en torno al ángulo central. Por otro lado, se busca explorar también el tamaño de la imagen de entrada posterior a la rotación de las imágenes  $a$ . Para cada combinación de valores, se entrenan 5 modelos distintos, obteniendo el promedio de estas iteraciones, con el objetivo de verificar la robustez del modelo (dado por una baja variación de sus resultados) y que no dependa de una inicialización particular. Los parámetros explorados se muestran en la tabla 3.2.

Tabla 3.2: Búsqueda hiperparámetros para el modelo *Stamp Classifier* propuesto.

Hiperparámetro	Valores de búsqueda
Rotaciones k	4; 5; 6; 7; 8
Libertad $\theta$	$0^\circ$ ; $3.6^\circ$ ; $7.2^\circ$ ; $10.8^\circ$ ; $14.4^\circ$ ; $18^\circ$ ; $180^\circ$
Tamaño del recorte de la imagen $a$	21, 27, 33, 39

### 3.1.8. Experimentos extras

- **Validación del entrenamiento con datos extra:** En la sección 3.2 de esta tesis se ocupa una base de datos análoga a los datos utilizados en el *Stamp Classifier*, pero con características temporales, la que fue obtenida tiempo después. Por este motivo, existen ejemplos que están en esta nueva base de datos y no se encuentran en **ALeRCE-ZTFv5**, por lo cual pueden ser utilizadas como muestras para probar el funcionamiento del clasificador. Notar que estas muestras solo están repartidas en las clases *AGN*, *SN*, *VS* y *bogus*, más no existen muestras de *asteroides*, y solo se tomó la primera detección de cada objeto para este experimento. El resultado de esto se puede ver en la figura C.1 del anexo C.3.
- **Cruce con catálogo ALL-WISE:** Como se mencionará en la siguiente sección, el cruce con el catálogo ALLWISE entrega información relevante para la clasificación entre objetos reales. Debido a esto, se propuso la idea de incluirlo dentro de la clasificación de la primera alerta. El resultado de la inclusión utilizando la base de datos original para el entrenamiento y la validación se muestra en la figura C.2 del Anexo C.3. Ante estos resultados, se optó por realizar una validación utilizando los datos extras del primer experimento mencionado anteriormente, con lo cual se obtuvo la figura C.2.

## 3.2. Clasificación temporal de objetos astronómicos

En este capítulo se detalla el trabajo realizado para solucionar el problema de clasificación temprana de 3 clases de objetos astronómicos: AGN, VS y SN. Esta tarea se resuelve a través de un clasificador basado en CNNs con rotaciones aleatorias y capas recurrentes RNNs, llamado *Temporal Stamp Classifier* (TSC). Este modelo utiliza como entrada información de alertas astronómicas, compuestas por imágenes y metadatos de las detecciones tempranas del flujo de alertas generadas por el sondeo astronómico ZTF. El principal objetivo del modelo es poder detectar SN lo antes posible, lo cual conlleva a que se utilicen las primeras 5 alertas en vez de tomar aleatoriamente detecciones del objeto, pudiendo analizar de esta manera como aumenta la confianza en las predicciones a medida que se obtienen más detecciones del mismo.

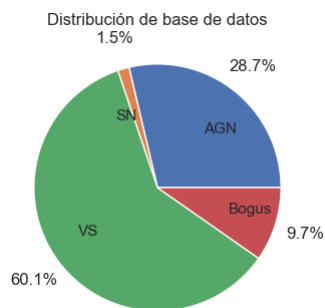
### 3.2.1. Base de datos

Para el entrenamiento del modelo TSC se utiliza una base de datos de alertas de ZTF. Esta base de datos contiene alertas ordenadas cronológicamente de detecciones de objetos, compuesta por imágenes, día y hora de medición y metadatos, con un máximo de 10 detecciones por objeto. Las imágenes de la alerta corresponden a estampillas, que se llamarán *stamps* por simplicidad, correspondiente al momento de la alerta. Cada uno de estos stamps corresponden a un recorte de la imagen original obtenida por el telescopio, centrado en el evento registrado y con una resolución de  $63 \times 63$ , donde cada píxel del stamp equivale a 1 arco-segundo de declinación en el cielo. Los metadatos presentes en la base entregan información respecto a la posición espacial del objeto observado, condiciones sobre la cual se toma la detección, información respecto a objetos próximos, cercanía de objetos de interés (como galaxias) desde otros catálogos, tiempo en el cual se tomó la medición, etc.

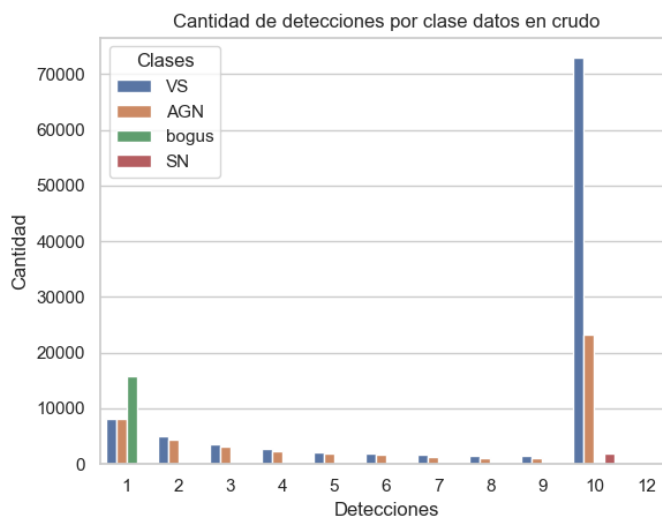
La cantidad de alertas en la base de datos es de 1.183.092, al agrupar todas estas alertas por objeto, utilizando la llave *objectId* que permite identificar un objeto único, el conteo de objetos llega a 168.156, los cuales se dividen en 48.262 (28.7%) AGN, 101.038 (60.09%) de VS, 2.524 (1.51%) de SN y 16.332 (9.71%) de bogus. Esta distribución se muestra de manera gráfica en la figura 3.3, como también la cantidad de objetos que se tiene por número de detecciones, debido a que algunas muestras no alcanzan a tener las 10 detecciones de corte de este catálogo.

### 3.2.2. ALLWISE

El catálogo ALLWISE fué utilizado en [11] para la clasificación de objetos estocásticos en sus subclases, teniendo una gran importancia para el clasificador propuesto en ese trabajo. Para poder encontrar este dato, se iteró sobre todos los objetos presentes en la base de datos, con lo cual se toma la posición en coordenadas J200 de la primera alerta; utilizando esta posición, se utiliza el código presente en [24] encontrando los valores de ALLWISE asociados a esa posición en un rango de 5 arcossegundos alrededor del punto. Los objetos que no poseen un valor de ALLWISE para alguno de los filtros, o en todos ellos, se rellenan con el valor 0. Una vez obtenidos todos los valores, se toma la resta consecutiva entre los valores de cada filtro, obteniéndose 3 metadatos nuevos llamados:  $W_1 - W_2$ ,  $W_2 - W_3$  y  $W_3 - W_4$ , donde  $W_i$  corresponde a la magnitud encontrada en el filtro  $i$  -ésimo.



(a) Proporción de datos en data-set.



(b) Distribución temporal.

Figura 3.3: Proporciones y distribución de la base de datos utilizada en el modelo temporal stamp classifier en formato crudo. Figura de creación propia.

### 3.2.3. Pre-procesamiento

El pre-procesamiento aplicado a la base de datos original + ALLWISE se resume en los siguientes puntos, los cuales están ordenados de forma secuencial.

- **Bogus:**

1. Eliminar todas las muestras de clase bogus de la base de datos.

- **Tiempos de medición:**

1. Calcular la diferencia en segundos entre cada detección consecutiva de un objeto.
2. Descartar las detecciones consecutivas que tengan una diferencia menor a 6 horas.
3. Descartar todas las detecciones posteriores a la quinta detección del objeto en orden cronológico.
4. Tomar el logaritmo de la diferencia en tiempo entre detecciones. Notar que el tiempo está medido en fecha juliana.

- **Imágenes:**

1. Descartar alertas con una dimensión distinta de  $63 \times 63$ .
2. Descartar imágenes de Difference de todas las detecciones.
3. Recortar los stamps a  $45 \times 45$  píxeles centrados.
4. Normalizar cada imagen de cada detección de cada objeto entre 0 y 1, sin considerar los valores que no sean numéricos (NaN).



5. Reemplazar valores NaN con el valor 0.

- **Metadatos:**

1. Se toman 23 metadatos detallados en la tabla A.1 del Anexo A.
2. Se limitan los valores máximos y mínimos de los metadatos. Estos valores se encuentran en la tabla A.3 del Anexo A.
3. Agregar los metadatos de ALLWISE a partir de la posición de los objetos, según lo explicado en el punto 3.1.2.
4. Se normalizan los datos para que distribuyan  $(0, 1)$ , calculando los estadísticos de media ( $\mu$ ) y desviación estándar ( $\sigma$ ) para cada número de detección de forma independiente, sobre el conjunto de entrenamiento de metadatos.

A continuación, se explica en forma más detallada el procedimiento utilizado para el preprocesamiento de los datos, junto a las justificaciones e implicancias de las decisiones tomadas.

- **Bogus:** En primera instancia se eliminan todos los ejemplos que sean de la clase Bogus. Esto se debe a que esta clase corresponde a errores de medición por parte del telescopio, pixeles quemados que se confundieron con una alerta, rayos cósmicos, etc. Al ser errores, no tienen un comportamiento temporal, o sea, no deberían volver a aparecer más de una vez en el mismo lugar o con el mismo identificador. Esto se condice con el análisis de distribución de datos, donde la gran mayoría de estos ejemplos poseen solo una detección. Al ser este modelo un análisis para ejemplos con más de una detección, y considerando que existen actualmente bastantes fuentes para descartar estos ejemplos en etapas anteriores como [10, 24], se decidió simplificar el modelo y eliminar esta tarea de clasificación.
- **Tiempos de medición:** Los tiempos de medición en la base de datos son registrados en Julian Date (JD), los cuales no son legibles fácilmente, por lo cual se transforman a fecha y hora. Teniendo este formato, se obtiene la diferencia en segundos entre cada medición consecutiva. A modo de ejemplo, se obtiene la diferencia en segundos entre la medición de la primera detección del objeto y la segunda detección del objeto, lo cual se guarda como una nueva variable.

Pruebas preliminares han demostrado que ciertos tipos de objetos presentan cambios notorios entre sus detecciones cuando la diferencia es mayor a 6 horas, lo cual también se condice con los tiempos en los cuales los procesos astronómicos suceden: No es posible identificar un cambio en la explosión de una SN cuando entre cada medición existen minutos de diferencia. Debido a esto, se eliminan las muestras que tengan una diferencia menor a 6 horas. Debido a este último punto, la diferencia entre mediciones en segundos (y también en JD) tienden a ser valores muy elevados, lo cual es contraproducente para el desempeño de un modelo de red neuronal. Debido a esto, y a experiencias anteriores con tiempos de medición [26], se aplica la función logaritmo sobre todos los datos, reduciendo su valor significativamente, pero manteniendo el comportamiento creciente. Cabe destacar que para la primera detección siempre se tiene una diferencia temporal de 0, por lo cual, al momento de aplicar el logaritmo, este valor se indefiniría. Ante esto,

se decidió mantener el valor 0 para todas las primeras detecciones.

Finalmente, debido a que el modelo trabajado busca ser una conexión entre la clasificación de la primera muestra y la clasificación tardía utilizando la curva de luz, se limita la cantidad de detecciones a utilizar a solo las primeras 5, por lo cual se descartan las siguientes detecciones de la base de datos.

- **Imágenes:** Las imágenes provenientes de ZTF tienen una dimensión estándar de  $63 \times 63$ , pero en ciertas ocasiones ocurre que se generan imágenes con dimensiones distintas debido a que una alerta se genera en el borde de la imagen generada por la CCD del telescopio, resultando en imágenes no cuadradas. Debido a que estos casos son excepcionales, se decidió descartar de la base de datos si en alguna de las imágenes se presenta este problema, resultando en la eliminación de 7793 ejemplos. Posterior a este filtro, la distribución de las clases queda en 46247 (28.8 %) AGN, 95377 (59.5 %) VS, 2416 (1.5 %) SNe y 16323 (10.2 %) Bogus.

Se realiza una normalización de los valores presentes en cada imagen para que sigan una distribución (0,1) sin considerar los valores NaNs de las imágenes. Esto se realiza principalmente para tener valores similares en la entrada del modelo independiente del objeto que se tenga. Debido a saturaciones presentes en los píxeles al momento de capturar el *stamp*, algunos valores quedan dañados, lo cual se traduce en valores NaNs dentro de la imagen, los cuales se encuentran más presentes en la clase Bogus. Para poder trabajar con ellos en el modelo y que no indetermina la función de pérdida, se cambian estos valores por 0 posterior a la normalización de la imagen; al otorgarle un valor a estos valores indeterminados, se facilita el procesamiento de la imagen en el sentido que es un valor estándar para todas las muestras.

Pruebas preliminares del modelo han demostrado que la imagen de diferencia permite al modelo identificar principalmente Bogus de las otras clases, lo cual no es una tarea asignada al clasificador construido en esta tesis; al remover esta imagen de la entrada al modelo, no cambia significativamente el resultado obtenido, pero sí disminuye considerablemente la complejidad del modelo como también el tiempo de procesamiento de cada muestra, por lo cual se procede a descartar esta imagen en cada detección. En pruebas anteriores propias y según Carrasco-Davis & Reyes et al. [10], imágenes pequeñas tienden a tener mejor rendimiento en modelos de clasificación junto al hecho de que los stamps se encuentran centrados en el punto de interés; por lo cual, se procede a recortar las imágenes en el centro a una dimensión de  $45 \times 45$ . Este tamaño se selecciona a partir de experimentos realizados para mejorar el modelo Stamp Classifier y explicados en la sección anterior 3.1.7, donde se realiza el recorte de la imagen posterior a las rotaciones de esta, preservando de esta manera valores reales en rotaciones no cuadradas.

Finalmente, utilizando las imágenes de cada detección, se arma un hipercubo dependiendo de la cantidad de detecciones que tenga el ejemplo en particular. Este hipercubo se construye armando un cubo en cada detección, compuesto por las imágenes de ciencia y referencia, para luego apilar los cubos de cada detección, con lo cual se tiene una

dimensión de entrada de imágenes de  $(n, 45, 45, 2)$ , donde  $n$  representa la cantidad de detecciones que tiene el ejemplo.

- **Metadatos:** Como tercera entrada al modelo se tienen los metadatos de las detecciones, los cuales, junto a ALLWISE agregado en esta tesis, corresponden a 26 características (vector) asociado a cada ejemplo y detección. Para cada metadato se limita su valor en todas las detecciones ocupando la tabla A.3. Debido a que algunos de los metadatos dependen de la detección misma (ej: magnitud de la detección), el rango en el cual se mueve cada detección es distinto, por lo cual la normalización de cada detección debe ser independiente. Por lo cual, cada detección es normalizada de forma independiente tomando el promedio  $\mu_i$  y la desviación  $\sigma_i$ , con  $i$  la detección, donde estos estadísticos se calculan utilizando solo el conjunto de entrenamiento para todo  $i \in \{1, 2, 3, 4, 5\}$ .

La distribución de los datos en términos de cantidad de ejemplos por clase y la separación en número de detecciones después del pre procesamiento se muestra en la figura 3.4. Junto a esto, se muestra la distribución de las diferencias temporales en escala logarítmica de las 5 detecciones consideradas, siendo 4 diferencias temporales posibles para las 3 clases consideradas en la figura 3.5.

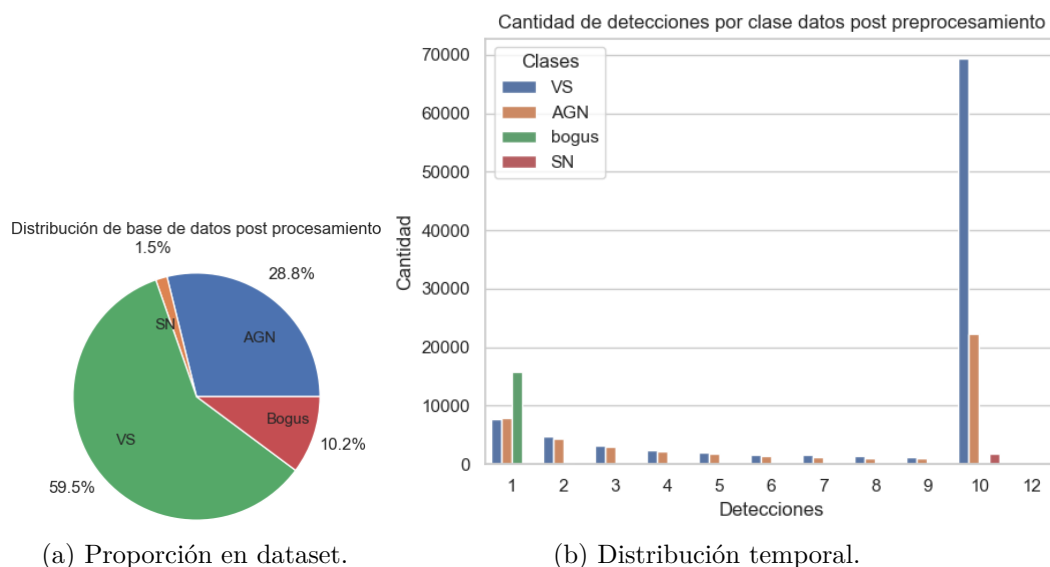


Figura 3.4: Proporciones y distribución de la base de datos utilizada en el modelo temporal stamp classifier después del pre procesamiento. Figura de autoría propia.

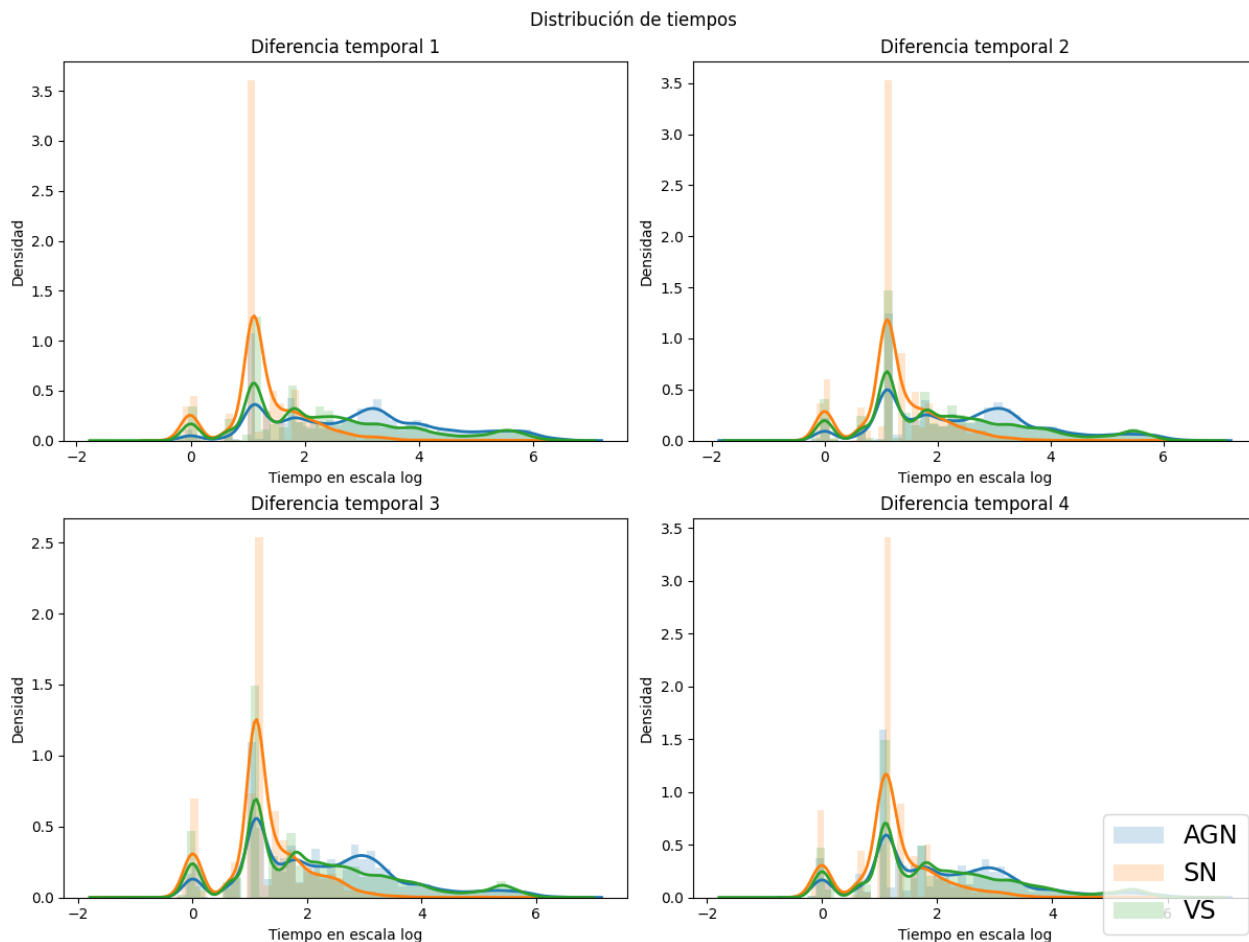


Figura 3.5: Distribución de las diferencias temporales en escala logarítmica para las 3 clases trabajadas.

### 3.2.4. Partición de datos

Tal como se explicó en el punto anterior, hay ciertas partes del pre-procesamiento de datos que eliminan detecciones de ciertos ejemplos, como también otros que eliminan ejemplos completamente, por lo cual es necesario realizar la partición de datos posterior al proceso de pre-procesamiento. Debido a que existen muestras con pocas detecciones, se tomó como base para la selección de ejemplos solo los ejemplos que tenían 5 detecciones o más; dentro de estas muestras, se tomó aleatoriamente 200 muestras por clase para el conjunto de validación y otras 200 muestras aleatorias por clase para el conjunto de test, con lo cual, tanto el conjunto de validación como de test tienen 600 muestras balanceadas, mientras que el conjunto de entrenamiento contiene el resto de las muestras, donde la cantidad de ejemplos depende directamente de la cantidad de detecciones sobre la cual se quiere entrenar, lo cual se detalla en la tabla 3.3. La decisión de tomar 200 muestras por clase se basa en tener conjuntos balanceados para la validación, con lo cual las métricas de pérdida y accuracy tengan mayor relevancia; también se busca que los porcentajes de estas mismas métricas no varíen tanto ante ejemplos que pudieran tener problemas o ser más complejas en su clasificación, como puede ser ejemplos con imágenes mal tomadas, un error en la obtención de ciertos metadatos, píxeles NaNs en zonas de interés, etc. Cabe mencionar que no se aplicó la técnica de sobre muestreo utilizada en el Stamp Classifier debido a que la diferencia de proporción de

muestras en el conjunto de entrenamiento es demasiada, teniendo un 2% de muestras de SN, por lo que se tendría una repetición constante de las mismas muestras, llevando a un sobre ajuste en pruebas preliminares de esta tesis.

Tabla 3.3: Distribución de ejemplos por clase en el conjunto de entrenamiento para cada número de detecciones.

Clase \ Detecciones	1	2	3	4	5
AGN	46247 (32.11 %)	38276 (29.85 %)	33998 (28.54 %)	31068 (27.53 %)	28834 (26.66 %)
SN	2416 (1.68 %)	2360 (1.84 %)	2313 (1.97 %)	2260 (2 %)	2183 (2.02 %)
VS	95377 (66.21 %)	87600 (68.31 %)	82802 (69.49 %)	79529 (70.47 %)	77139 (71.32 %)

### 3.2.5. Arquitectura del clasificador

El modelo de clasificación utilizado en esta tesis se basa en el modelo presentado en [10], con la introducción de modelos recurrentes para el procesamiento de las distintas detecciones de los ejemplos, tanto en sus imágenes como en sus metadatos, como también se agregaron rotaciones equiespaciadas de las imágenes de entrada y la inclusión de aleatoriedad en el ángulo de rotación tomando la mejor arquitectura encontrada en los experimentos descritos en la sección 3.1.7.

Para describir el modelo de rotaciones aleatorias, es necesario definir 3 cantidades que varían según las pruebas realizadas y descritas posteriormente. En primer lugar, se tiene el número de rotaciones  $k$ , las cuales son equiespaciadas, a modo de ejemplo, si se define  $k = 5$ , las imágenes de entrada serán rotadas en  $0^\circ$ ,  $72^\circ$ ,  $144^\circ$ ,  $216^\circ$  y  $288^\circ$ . En segundo lugar, el grado de libertad para rotar  $\theta$  comentado en 2.2, el cual define cuanta libertad tiene la imagen para rotar; si se tiene un  $k = 5$  y un  $\theta = 10^\circ$ , entonces la segunda rotación de la imagen puede ser cualquier valor entre  $62^\circ$  y  $82^\circ$ , y cambia en cada iteración sobre el modelo. Finalmente, se tiene la cantidad de detecciones a considerar, lo cual, según definición inicial del modelo, se tiene  $n \in \{1, 2, 3, 4, 5\}$ .

Las entradas al modelo *temporal stamp classifier* están compuestas por un hipercubo con las imágenes provenientes de ZTF de cada detección: imágenes de ciencia y de referencia, las cuales son pre procesadas formando el hipercubo de dimensiones  $n \times 45 \times 45 \times 2$ , como también de la diferencia entre los tiempos de medición en forma de vector de dimensión  $n$ , y un vector de 26 metadatos. El modelo convolucional recibe un *batch* de imágenes de una detección en particular, por lo cual cada detección entra al modelo convolucional de forma separada, generando una salida para cada detección. Una vez que ingresa a la capa convolucional, esta es aumentada con versiones rotadas de sí mismo  $k$  veces; considerando la función  $r = \lfloor 360/k \rfloor^\circ$  que rota la imagen en  $r$  grados, con lo que se obtiene un batch aumentado  $B(x) = [x, rx, \dots, r^{(k-1)}x, r^kx]$ . Cada una de estas rotaciones son recortadas en el centro a  $33 \times 33$  pixeles, ingresando a las capas convolucionales, las cuales poseen los parámetros mostrados en la tabla 3.4, donde se detalla el número de filtros, tamaño de la convolución, *padding* utilizado, dimensiones de entrada y salida. Todas las convoluciones, a excepción de la primera y la última, poseen *zero padding*, con el objetivo de mantener la dimensión de entrada en la salida de cada capa, como también poseen funciones de activación *ReLU*, al igual que las capas *Fully Connected*. La salida de la última capa de Max Pooling es aplanada formando los vectores que entran a las capas *Fully Connected*, siendo esta la última capa que

trabaja de forma independiente con la rotación. Tomando la salida de esta capa, se apilan cada vector de cada rotación obteniéndose un tensor de  $k \times 64$ , donde se promedia los valores en la dimensión apilada, obteniéndose finalmente un vector de 64 dimensiones, siendo esto el proceso de *Cyclic Pooling*, donde se promedia los valores en la dimensión apilada. La salida del proceso de *Cyclic Pooling* corresponde a la salida del procesamiento de cada detección, siendo esta misma la entrada a la capa recurrente de la red, por lo cual el vector de salida será de  $n \times 64$ . A este vector se le concatenan las diferencias temporales, como también los metadatos con comportamiento temporal, resultando en un vector recurrente de  $n \times 72$ , el cual es procesado por las capas recurrentes, obteniendo una salida con 72 dimensiones. Posterior a este procesamiento recurrente, se concatenan los metadatos estáticos, obteniendo una salida de 87 dimensiones, se aplican las capas *Fully Connected*, obteniendo finalmente, gracias a la capa *Fully Connected* con función de activación *softmax*, las probabilidades estimadas para las 3 clases.

En la figura 3.6 se muestra una representación gráfica de la arquitectura, la cual se ajustó a  $k = 5$ , donde se puede apreciar las rotaciones utilizadas, la entrada al procesamiento recurrente y la utilización de solo 2 imágenes por detección. Por otro lado, en la tabla 3.4 se aprecia la arquitectura capa por capa, con los parámetros de cada una de ellas y el tamaño de la salida.

Tabla 3.4: Arquitectura del temporal Stamp Classifier para cualquier número de detecciones  $t$ . La capa *RNN* depende del número de detecciones considerado y resumido en la tabla 4.4.

Capa	Parámetros de la capa	Tamaño de salida
Entrada	-	$t \times 27 \times 27 \times 3$
Aumentación por rotaciones	-	$t \times 27 \times 27 \times 3$
Convolución	$4 \times 4, 32$	$t \times 24 \times 24 \times 32$
Convolución	$5 \times 5, 32$	$t \times 24 \times 24 \times 32$
Max pooling	$2 \times 2, \text{stride } 2$	$t \times 12 \times 12 \times 32$
Convolución	$5 \times 5, 64$	$t \times 12 \times 12 \times 64$
Convolución	$5 \times 5, 64$	$t \times 12 \times 12 \times 64$
Convolución	$5 \times 5, 64$	$t \times 12 \times 12 \times 64$
Max pooling	$2 \times 2, \text{stride } 2$	$t \times 6 \times 6 \times 64$
Aplanado	-	$t \times 2304$
Fully connected	2304	$t \times 64$
Concatenación de rotaciones	-	$t \times k \times 64$
Cyclic pooling	-	$t \times 64$
Concatenación con metadatos dinámicos tras BN	-	$t \times 64 + 8$
RNN	Variable	72
Concatenación con metadatos estáticos tras BN	-	87
Fully connected con dropout [36]	$90 \times 64$	64
Fully connected	$64 \times 64$	64
Fully connected con softmax	$64 \times 3$	3 (nº clases)

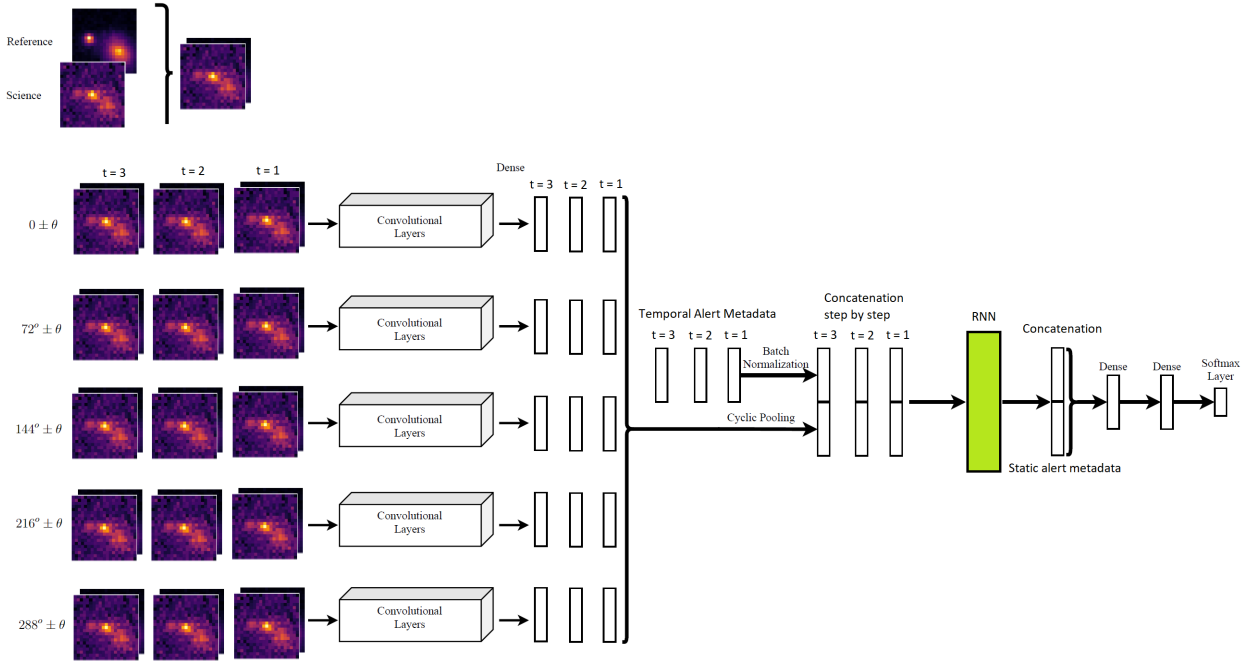


Figura 3.6: Modelo *Temporal Stamp Classifier*. Ejemplo para el caso de 3 detecciones, con 5 rotaciones y con un ángulo de libertad  $\theta$ . *RNN* simboliza la red recurrente aplicada y detallado por número de detecciones en la tabla 4.2.

### 3.2.6. Esquema de entrenamiento

El entrenamiento del modelo completo se divide en dos pasos:

- El primer paso se basa en el entrenamiento del modelo base para obtener un punto de partida de los filtros convolucionales que procesan las imágenes. Para esto, se toma solo las imágenes de la primera detección de cada objeto, con las cuales se entrena el modelo mostrado en la figura 3.6, con la salvedad de que no incluye la capa de *RNN* ya que solo se está considerando la primera detección, como también que la estructura es la mejor arquitectura encontrada para el Stamp Classifier dado por un recorte de imagen  $a = 33$ , 6 rotaciones y una libertad de  $14.4^\circ$ . El entrenamiento de este modelo se realiza evaluando en el conjunto de validación cada 10 iteraciones de entrenamiento, donde se guardan los pesos de las capas convolucionales en caso de que sea el menor valor de la función de pérdida encontrada hasta el momento. Este proceso se realiza durante 20000 iteraciones, luego se realizan 10000 iteraciones más, si en esas 10000 iteraciones se encuentra un menor valor de la función de pérdida, se agregan 10000 iteraciones más. El entrenamiento se detiene cuando se ha pasado por 10000 iteraciones sin encontrar un nuevo mínimo en la función de pérdida.
- El entrenamiento del modelo completo se inicia con la carga de los pesos de los filtros convolucionales obtenidos en el paso anterior, lo cual da un punto de partida para los pesos compartidos de todas las detecciones. Una vez cargados los pesos, se entrena el modelo de la figura 3.6 incluyendo la capa *RNN* y los metadatos estáticos y dinámicos durante 30000 iteraciones, donde una iteración se considera el procesamiento de un *batch* de datos, evaluando en el conjunto de validación cada 10 iteraciones, guardando el



menor valor registrado en este conjunto. Durante cada iteración, se actualizan los pesos tanto de las capas convolucionales como las del resto del modelo. Posterior a las 30000 iteraciones, se inicia un contador en 0, donde se va sumando 1 cada vez que se avanza una iteración; cada 10 iteraciones se analiza la función de pérdida en el conjunto de validación, si es un valor menor a lo obtenido hasta el momento, se reinicia el contador en 0. Si el contador llega a 20000, se termina el entrenamiento.

Los conjuntos de validación y test se determinan como primer paso, por lo cual se mantienen constantes durante todos los entrenamientos, mientras que el resto de alertas son utilizadas en el conjunto de entrenamiento. Notar que los ejemplos para el paso 1 y 2 corresponden son los mismos, diferenciándose solo en la cantidad de detecciones consideradas.

Al momento de entrenar el modelo, se utiliza el optimizador Adam [54] para ajustar los pesos del modelo, con los valores de sus hiperparámetros dados por defecto, y la función de pérdida que se utiliza es la entropía cruzada.

### 3.2.7. Experimentos

A continuación se describe la metodología utilizada en los experimentos para evaluar el *Temporal Stamp Classifier*. Cabe mencionar que los primeros experimentos se basan en encontrar el mejor modelo base para el entrenamiento, o sea, entrenar con una sola detección para las 3 clases existentes, lo cual se utiliza como base para el entrenamiento del modelo final. En la sección 3.2.7.3 se mencionan otros experimentos extra que no son esenciales para el desarrollo y análisis de este trabajo.

#### 3.2.7.1. Pre entrenamiento de capas convolucionales

El pre entrenamiento de las capas convolucionales se realiza utilizando la primera alerta de todas las muestras presentes en la base de datos. La arquitectura de las capas convolucionales corresponde a la mejor estructura encontrada para el modelo *Stamp Classifier*, compuesto por imágenes de entrada de dimensiones  $45 \times 45$ , 6 rotaciones, una libertad de rotación de  $14.4^\circ$  y un recorte posterior a la rotación de  $a = 33 \times 33$ . El modelo se entrena utilizando solo las imágenes de ciencia y referencia. El resultado de este entrenamiento se muestra en la figura 4.3.

#### 3.2.7.2. Hiperparámetros procesamiento recurrente

El procesamiento de las características recurrentes de las detecciones se realiza mediante redes RNN, las cuales tienen distintas variaciones. Debido a que el objetivo de procesamiento son secuencias de pocas detecciones (máximo 5), redes muy complejas podrían no ser las óptimas debido a su gran cantidad de parámetros a ajustar. En consecuencia, se probaron distintas combinaciones de redes utilizando 6 modelos: *LSTM*, *GRU*, *SimpleRNN*, *Gamma memory*, *TDNN* y *Conv1D* con distintas combinaciones para cada tipo; estos entrenamientos fueron realizados utilizando solo las imágenes. Para cada número de detecciones  $n \in 2, 3, 4, 5$  se entrenó todas las configuraciones mostradas en la tabla 3.5, seleccionando el mejor modelo para cada  $n$  de forma independiente.

Tabla 3.5: Búsqueda de hiperparámetros para capas recurrentes. Cada configuración es entrenada para cada número de detecciones considerada en esta tesis. D corresponde a la cantidad de detecciones considerada.

Red	Celdas	Capas
LSTM	32, 64, 128	1, 2, 3
GRU	32, 64, 128	1, 2, 3
SimpleRNN	32, 64, 128	1, 2, 3
Red	Filtros $\gamma$	
Gamma Memory	1, 2, 3, 4, 5, 6, 7, 8	
Red	Tamaño del filtro	
TDNN	2	
Conv1D	2,...,D	

### 3.2.7.3. Inclusión de metadatos del catálogo ALLWISE

Los experimentos asociados a entrenar el modelo presentado con los metadatos utilizados en el trabajo original presentan un buen desempeño para la clasificación de las 3 clases del caso recurrente, pero posee ciertos problemas para poder identificar entre las clases VS y AGN. Debido a esto, y siguiendo los resultados obtenidos en [11], la información entregada por el cruce con el catálogo ALLWISE permite diferenciar de mejor manera estas dos clases, por lo cual se incluyó esta nueva característica como un metadato estático, debido a que es dependiente de la posición del objeto observado.

### 3.2.7.4. Experimentos extras

- **Efecto de utilizar Transfer Learning:** La metodología para entrenar el modelo Temporal Stamp Classifier comienza con el entrenamiento de las capas convolucionales utilizando la primera alerta de cada objeto, para luego cargar estos pesos pre entrenados en cada modelo a entrenar. Este procedimiento se diseñó con la idea de tener un punto de partida para los pesos y que posteriormente solo se vayan ajustando para cada número de detección y que el entrenamiento sea más rápido en converger, como también para evitar posibles mínimos locales cuando la cantidad de parámetros es muy alta. Para probar su efectividad, se entrenó 5 veces cada modelo de la tabla 4.4 con transfer learning y sin esto, obteniendo la evolución de la función de pérdida. Esta evolución se muestra en la figura B.1 del anexo B.1. Las curvas muestran que el uso de transfer learning permite tener una convergencia a baja cantidad de iteraciones, teniendo un bajo valor de función de pérdida a temprana época de entrenamiento en contraste con el caso aleatorio, donde la convergencia toma más tiempo. Para el caso de 5 detecciones, la diferencia es más notoria, teniendo un valor menor en la función de pérdida para el caso con transfer learning.
- **Entrenamiento de los filtros convolucionales con la primera alerta:** Se compara la utilización de las imágenes de todas las alertas hasta la quinta en el pre entrenamiento de los filtros convolucionales en contraste con solo utilizar la primera alerta. Esto se realiza con el modelo base que considera 4 rotaciones sin aleatoriedad y un recorte  $a = 21$ . Este resultado se muestra en la figura B.2. La decisión de utilizar solo la primera detección de cada muestra para el entrenamiento inicial de los filtros convolucionales se

toma debido a que se quiere evitar entregar información a priori de como serían las alertas posteriores para el caso de bajas detecciones, donde esta tendría información respecto a las detecciones posteriores; por otra parte, este entrenamiento se realiza con el objetivo de dar un punto de partida para los demás modelos, por lo cual no se hace imperativo que el ajuste sea óptimo ya que se ajustarán al momento de entrenar los demás modelos. Por otra parte, la figura B.2 muestra que la diferencia entre ambas no es considerable, por lo cual se prefiere el uso de solo la primera alerta debido a tener un menor tiempo de entrenamiento.

- **Modelo recurrente de solo metadatos:** El modelo presentado en este trabajo considera el uso tanto de las imágenes de cada detección, como también de los metadatos asociados a esta detección. La combinación de ambos tipos ha demostrado mejorar de forma considerable la clasificación de objetos astronómicos, diluyendo de cierta manera el aporte de ambos datos al ser mezclados. Debido a esto, se analizó cuál es el desempeño de un clasificador usando solo uno de estos tipos de datos; para el caso de las imágenes, este se obtuvo en el experimento mostrado en la sección 3.2.7.2, mientras que para el caso de los metadatos, esta fue analizada de forma separada para cada número de detecciones entre 1 y 5. El resultado de esto puede ser apreciado en la figura 4.5 y 4.6, como los resultados del modelo sin utilizar ALLWISE.
- **Separación de metadatos estáticos:** Debido a que los metadatos entregan información respecto a las condiciones sobre las cuales se tomó la detección, algunos de ellos no presentan variación en su valor en las distintas detecciones, salvo por errores de medición; entre ellos se cuentan principalmente los que dependen directamente de la posición donde se encuentra el objeto observado como también los que dependen de la imagen de referencia. El experimento se basa en separar los metadatos entre estáticos y dinámicos; los dinámicos se concatenarán al vector entregado por las capas convolucionales previo a ingresar a las capas recurrentes, mientras que para los estáticos, se tomará el valor de la primera detección, para ser concatenados a la salida de la capa recurrente. Esto reduce la cantidad de características a procesar por la capa recurrente, simplificando el modelo y posiblemente mejorando los resultados. En la figura 3.7 se representa de manera gráfica como esta separación se agrega en el entrenamiento, donde *headers* representa los metadatos. Por otro lado, la arquitectura en detalle se muestra en la tabla 3.6

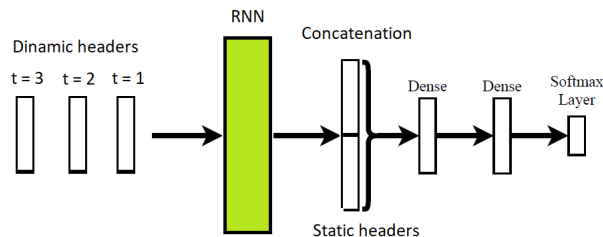


Figura 3.7: Separación de metadatos estáticos y dinámicos en el entrenamiento.

Tabla 3.6: Arquitectura del modelo *Temporal Stamp Classifier* que utiliza solo metadatos y no considera imágenes.

Capa	Parámetros de la capa	Tamaño de salida
Entrada	-	$t \times 8$
Batch Normalization	-	$t \times 8$
GRU	$2 \times 32$	32
Concatenación con metadatos estáticos tras BN	-	47
Fully connected con dropout [36]	$47 \times 64$	64
Fully connected	$64 \times 64$	64
Fully connected con softmax	$64 \times 3$	3 (n° clases)

- **Importancia de los metadatos en primera detección:** Los metadatos más importantes para detectar los objetos fueron estudiados en [10]. Sin embargo, ciertos metadatos fueron incluidos debido a su importancia para la clasificación de bogus y asteroides en contraste con las demás clases. Estas categorías no fueron incluidas en el trabajo actual, por lo cual es esperable que la importancia de los metadatos cambie en este nuevo enfoque. Este análisis se realizó solo con la primera detección ya que el modelo *Random Forest* no considera las dependencias temporales entre los datos, por lo cual no entrega una buena interpretación de la importancia de estos mismos. El resultado de este análisis se muestra en el anexo B.4, figura B.5.
- **Comparación de probabilidades entre modelos:** Una de las interpretaciones más relevantes que se obtuvo del modelo *Stamp Classifier* original es la posibilidad de estimar la probabilidad con las que un objeto se puede catalogar en cierta clase gracias a la función de pérdidas que penaliza los sobreajustes en las clases. Esto es sumamente útil en muestras que son fácilmente confundibles debido a problemas de captura. Con este objetivo se analizó si la inclusión de imágenes mejora la probabilidad del modelo en la clase predicha, y como esta se compara con el modelo *Stamp Classifier* y el modelo *Light Curve Classifier*. Esto se realizó haciendo llamadas a la API de ALerCE<sup>2</sup>, retornando todas las probabilidades predichas por ambos modelos, y tomando la probabilidad asignada a la clase real. El resultado de esta comparación se puede ver en el anexo B.3, figuras B.3 y B.4.
- **Bogus con más de una detección:** La eliminación de todas las muestras de clase bogus viene dado a que típicamente estos ejemplos no aparecen dos veces en la misma posición, como por ejemplo los rayos cósmicos. En el caso de muestran que si tengan más de una detección, se propone utilizar el modelo del Stamp Classifier mejorado para su detección utilizando la primera y la segunda detección de estas y si el modelo predice que se trata de un bogus en la primera y segunda detección, puede ser considerada como de esta clase. En el dataset utilizado, existen 16332 ejemplos bogus, donde solo 253 poseen más de una detección, con lo cual se obtuvo un clasificación correcta del 86 % para la primera detección y de 84 % para la segunda detección utilizando el modelo Stamp Classifier mejorado. Esto representa solo 32 muestras mal clasificadas de las 16332,

<sup>2</sup> <https://alerce.readthedocs.io/en/latest/>

siendo menos del 0.2% del total de bogus. Cabe mencionar que con esta baja cantidad de muestras, no es posible entrenar un modelo de clasificación tipo Aprendizaje profundo, por lo cual se debe utilizar otras técnicas para su clasificación como la propuesta.

# Capítulo 4

## Resultados

En este capítulo se detallan los resultados obtenidos de los experimentos expuestos en el capítulo 3. En la sección 4.1 se presentan los resultados de las mejoras propuestas al modelo del Stamp Classifier, junto con el resultado del mejor modelo en contraste con el modelo base. En la sección 4.2 se muestran los resultados para el modelo Temporal Stamp Classifier. En el punto 4.2.2 se presentan los resultados para todas las configuraciones de los modelos recurrentes probados y para cada número de detecciones utilizado, mostrando los mejores resultados para cada tipo de modelo recurrente. En el punto 4.2.3, se muestra la comparación entre el modelo base y el modelo que integra el cruce con el catálogo ALLWISE y, finalmente, en el punto 4.2.4 se muestra el modelo final considerando la mejor configuración de rotaciones, ángulo, modelo recurrente y ALLWISE, que utiliza tanto las imágenes como los metadatos.

### 4.1. Mejora al modelo *Stamp Classifier*

En esta sección se describen los resultados obtenidos con el modelo *Stamp Classifier* propuesto, con las modificaciones explicadas en la sección 3.1.7, así como también las comparaciones realizadas para evaluar la significancia de las mejoras propuestas.

#### 4.1.1. Variación del tamaño de recorte de las imágenes de entrada, cantidad de rotaciones y grado de libertad

En la tabla 4.1 se muestran los 5 mejores modelos de las combinaciones de hiperparámetros entrenadas junto a la configuración, donde  $M_1$  corresponde al mejor modelo hasta  $M_5$ .  $M_r$  corresponde al modelo con una libertad de rotación completa y  $M_b$  al modelo base con arquitectura igual al Stamp Classifier. Se muestra el accuracy logrado en test como también el valor p del test de hipótesis de permutación entre el modelo base y el modelo probado. Por otra parte, para medir el efecto de agregar más rotaciones al modelo, se grafica el accuracy en contraste con el tiempo de inferencia para los valores de  $k$  utilizados y en cada valor de recorte de la imagen  $a$ , mostrándose en la figura 4.1.

Tabla 4.1: Top 5 modelos con el mayor accuracy en el conjunto de validación obtenido en la búsqueda de hiperparámetros para el modelo Stamp Classifier mejorado. Los modelos van rankeados desde  $M_1$  hasta  $M_5$ .  $M_r$  corresponde al modelo con una rotación total de  $180^\circ$  y  $M_b$  corresponde al modelo base Stamp Classifier.

Modelo	Hiperparámetros	Accuracy en test	p-value
$M_1$	$IS : 33 ; K : 6 ; \theta : 14.4^\circ$	$93.36 \pm 0.4$	0.007
$M_2$	$IS : 27 ; K : 6 ; \theta : 18^\circ$	$92.96 \pm 0.5$	0.008
$M_3$	$IS : 27 ; K : 5 ; \theta : 18^\circ$	$92.84 \pm 0.4$	0.008
$M_4$	$IS : 39 ; K : 8 ; \theta : 14.4^\circ$	$92.84 \pm 0.3$	0.007
$M_5$	$IS : 27 ; K : 7 ; \theta : 14.4^\circ$	$92.72 \pm 0.3$	0.007
$M_r$	$IS : 33 ; K : 5 ; \theta : 180^\circ$	$92.22 \pm 0.8$	0.3
$M_b$	$IS : 21 ; K : 4 ; \theta : 0^\circ$	$91.44 \pm 0.4$	Referencia

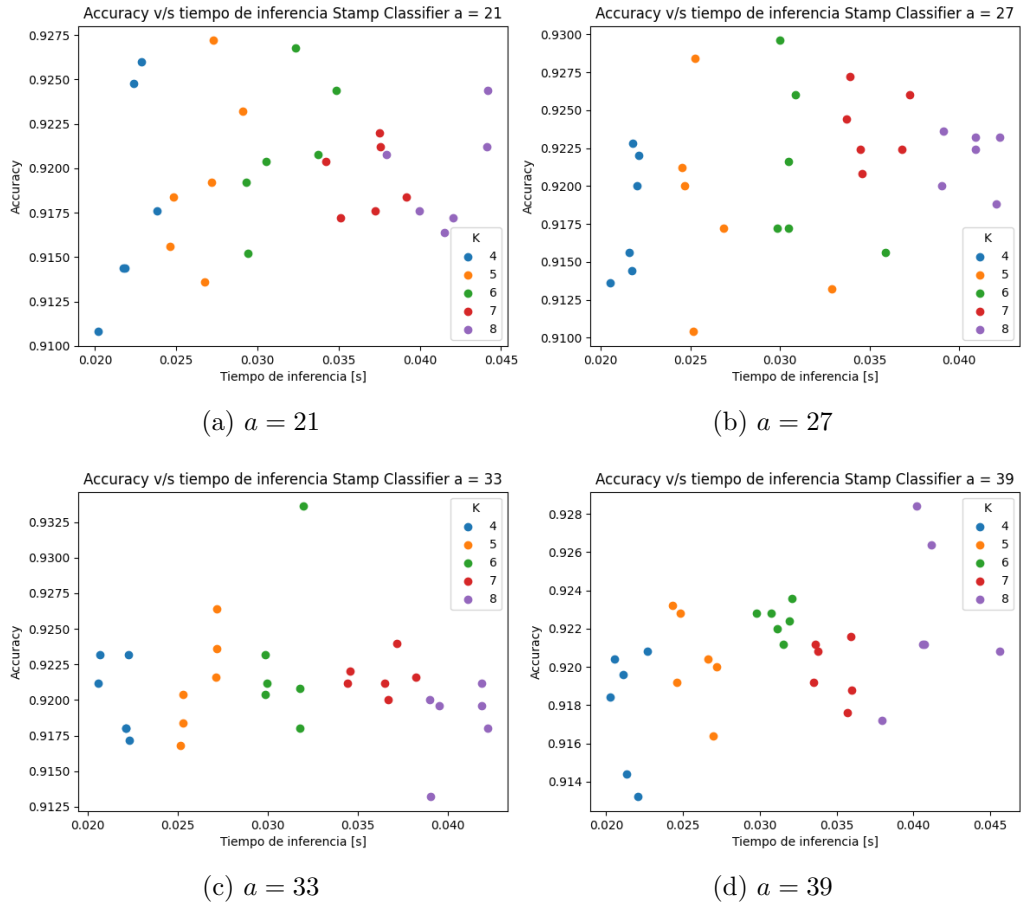
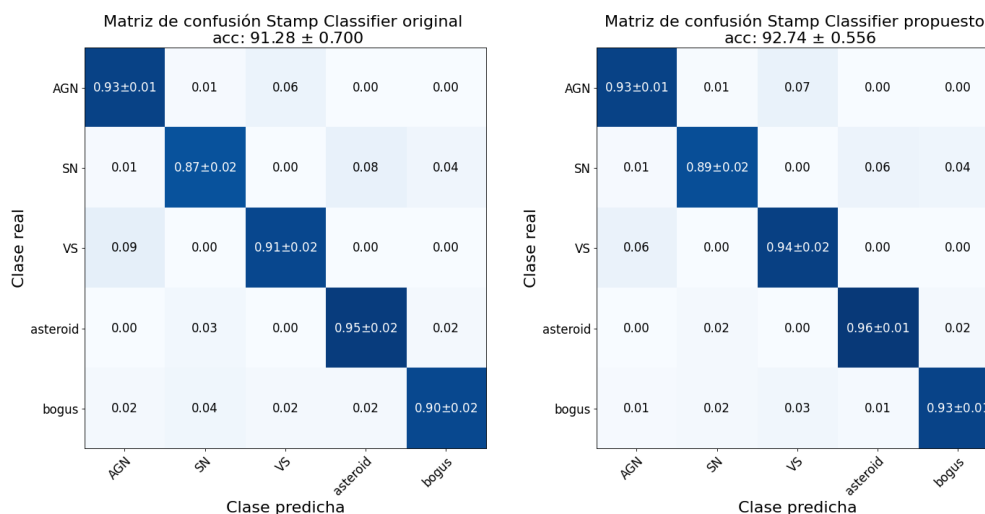


Figura 4.1: Tiempo de inferencia en contraste con el accuracy en test logrado por los modelos para cada número de recorte utilizado y para cada número de cantidad de rotaciones.

## 4.1.2. Mejor modelo Stamp Classifier

Para el mejor modelo obtenido, correspondiente al modelo con 6 rotaciones, un grado de libertad de  $14.4^\circ$ , un tamaño de batch de 128 y un tamaño de recorte de  $33 \times 33$ , se realiza una comparación con el modelo original en términos de sus matrices de confusión, donde se realizan 20 iteraciones del experimento original y el modelo propuesto con el objetivo de tener un resultado con mayor validez estadística. Con los resultados de los experimentos, se computa el valor del test de hipótesis de permutación [99] comparando ambos modelos, obteniendo un valor  $p = 0.01$ . Las matrices de confusión de ambos modelos se muestran en la figura 4.2.



(a) Modelo Stamp Classifier original. (b) Modelo Stamp Classifier propuesto.

Figura 4.2: Matrices de confusión del modelo original de Stamp Classifier presentado en [10], el cual incluye metadatos e imágenes, en el lado izquierdo (a) y del mejor modelo obtenido con las modificaciones propuestas en esta tesis en el lado derecho (b) con metadatos e imágenes. Entre ambos modelos se obtiene un valor  $p = 0.01$ .

## 4.2. Clasificación temprana temporal de objetos astronómicos

En esta sección se describen los resultados obtenidos en el entrenamiento del modelo *Temporal Stamp Classifier* en sus distintas etapas para la selección del mejor modelo para cada número de detecciones.

### 4.2.1. Pre entrenamiento de capas convolucionales

La estructura de los filtros convolucionales se toma a partir de los resultados de la mejora al modelo Stamp Classifier, o sea, las imágenes de entrada son de dimensiones  $45 \times 45$ , con un recorte de  $a = 33$ , 6 rotaciones y un grado de libertad en las rotaciones de  $14.4^\circ$ . Con este modelo, se entrena utilizando solo las imágenes, obteniéndose la matriz de confusión de la figura 4.3.



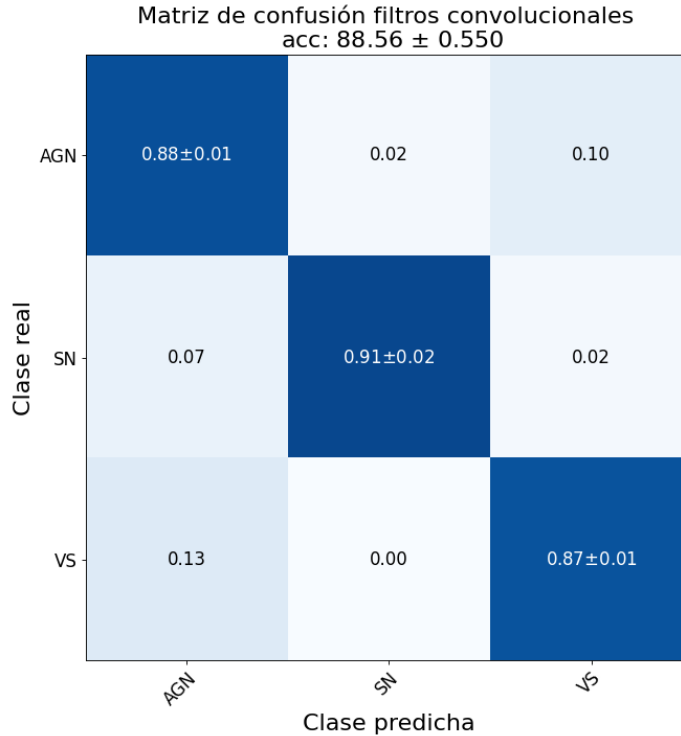


Figura 4.3: Matriz de confusión para el pre entrenamiento de los filtros convolucionales utilizando la primera detección de los objetos.

#### 4.2.2. Comparación entre modelos recurrentes

Utilizando como base para la estructura de las capas convolucionales el mejor modelo obtenido en el punto anterior, se entrenaron 37 configuraciones distintas para el procesamiento de más de una detección de ejemplos. La configuración entrenada depende del tipo de modelo recurrente a utilizar, donde LSTM, GRU y SimpleRNN tienen 9 configuraciones cada uno, variando la cantidad de celdas y capas recurrentes; Gamma Memory posee 9 configuraciones distintas siendo la cantidad de celdas  $\gamma$  el factor a variar entre los modelos, TDNN posee solo una configuración dada por el tamaño de la ventana a analizar, siendo de tamaño 2 y Conv1D posee  $D - 1$  configuraciones, donde  $D$  corresponde al número de detecciones considerado. Cada configuración detallada comienza a partir de *transfer learning*, o sea, es entrenada utilizando como punto de partida los pesos del mejor modelo obtenido en la primera detección, para luego continuar el entrenamiento con las nuevas entradas. Cada modelo es entrenado 5 veces distintas para cada número de detecciones, por lo cual se entrenó un total de  $37 \times 5 \times 4 = 740 + 10 * 5 = 790$  modelos en total, siendo  $185 + D - 1$  modelos para el procesamiento de cada número de detecciones, siendo  $D$  2, 3, 4 y 5 detecciones.

En cada configuración se calcula el promedio de *accuracy* en las 10 iteraciones realizadas en el conjunto de validación y el conjunto de prueba, como también la desviación estándar de esta métrica y el valor p entre el modelo de red recurrente y el modelo LSTM, el cual fue tomado como base de comparación; Todo esto se realizó para cada cantidad de detecciones por separado. En la tabla 4.2 se muestran los mejores modelos recurrentes para cada número de detecciones con las respectivas métricas mencionadas y la configuración correspondiente a cada modelo, donde  $C$  corresponde a la cantidad de capas,  $U$  la cantidad de unidades

por cada,  $\gamma$  la cantidad de filtros,  $V$  corresponde a la ventana de tiempo,  $Ke$  al tamaño del *kernel* y  $D$  al número de detecciones, como también se muestra el tiempo de inferencia para el modelo TI. Por otro lado, en la figura 4.4 se muestra un mapa de dispersión de todas las configuraciones entrenadas en contraste con el tiempo de inferencia obtenido, donde se remarca el mejor modelo obtenido con un cuadrado para cada tipo de arquitectura probada.

Tabla 4.2: Desempeño de los mejores modelos por tipo de modelo recurrente. Cada modelo recurrente presenta la mejor configuración obtenida para cada número de detecciones. Se muestra el valor p del test de hipótesis de permutación entre el modelo y el modelo LSTM para ese número de detecciones y la arquitectura del modelo en su abreviación.

	Modelo	GRU	LSTM	G. M.	TDNN	RNN	Conv1D
2 D	Acc.	$91.3 \pm 0.4$	$91.7 \pm 0.3$	$91.3 \pm 0.2$	$91 \pm 0.1$	$91.2 \pm 0.7$	$91.3 \pm 0.6$
	p-v.	0.35	Ref.	0.1	0.1	0.68	0.3
	Arq.	1C 256U	1C 256U	$\gamma = 7$	$V = 2$	1C 128U	$Ke = 2$
	TI	21.04	20.57	5.47	22.68	24.63	5.75
3 D	Acc.	$91.8 \pm 0.4$	$92 \pm 0.3$	$91.4 \pm 0.5$	$91.6 \pm 0.1$	$91.8 \pm 0.4$	$91.6 \pm 0.6$
	p-v.	0.28	Ref.	0.06	0.07	0.08	0.1
	Arq.	1C 256U	1C 64U	$\gamma = 7$	$V = 2$	1C 128U	$Ke = 3$
	TI	30.42	29.63	7.84	32.88	30.42	8.33
4 D	Acc.	$92.5 \pm 0.4$	$92.2 \pm 0.6$	$92.2 \pm 0.5$	$92.4 \pm 0.5$	$92.4 \pm 0.4$	$92.5 \pm 0.6$
	p-v.	0.31	Ref.	0.81	0.68	0.69	0.48
	Arq.	1C 256U	3C 256U	$\gamma = 6$	$V = 2$	1C 64U	$Ke = 2$
	TI	39.31	46.06	10.86	40.8	49.02	11.13
5 D	Acc.	$93.2 \pm 0.5$	$92.9 \pm 0.3$	$92.6 \pm 0.4$	$92.9 \pm 0.3$	$92.6 \pm 0.6$	$93.2 \pm 0.6$
	p-v.	0.23	Ref.	0.24	0.98	0.2	0.29
	Arq.	1C 64U	1C 128U	$\gamma = 8$	$V = 2$	1C 64U	$Ke = 4$
	TI	55	50.54	55.06	57.16	57.44	71.48

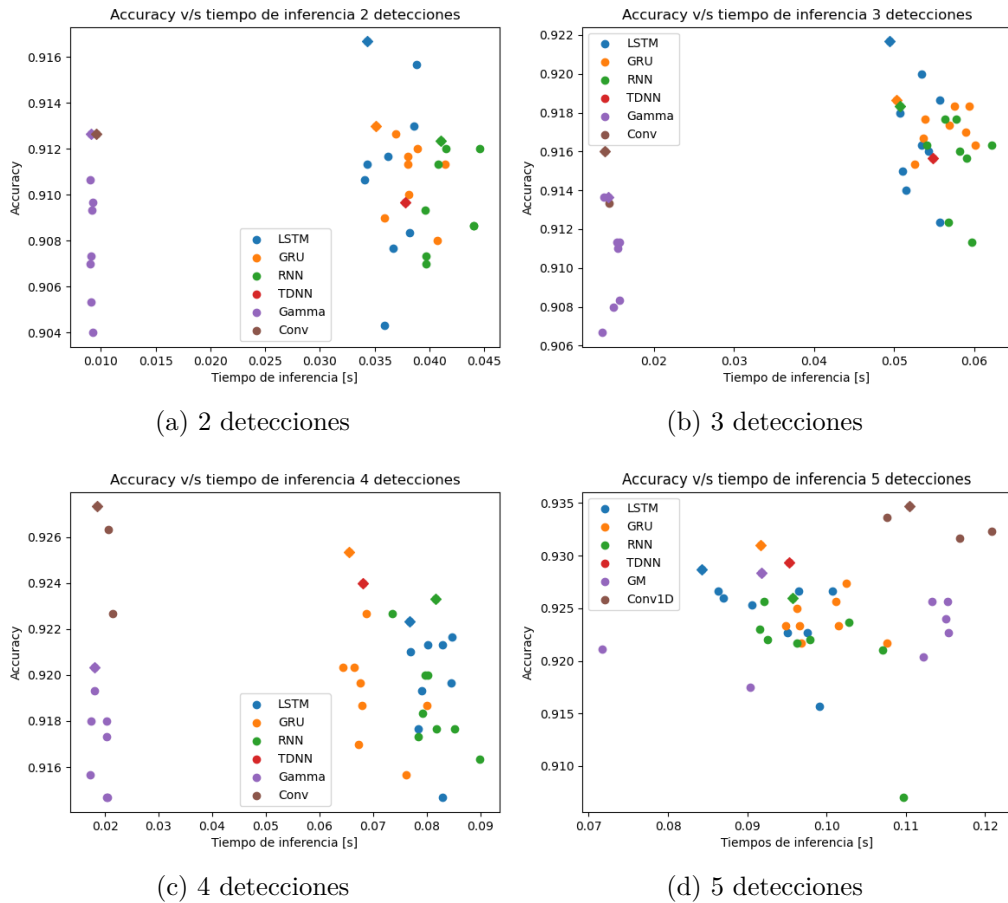


Figura 4.4: Desempeño de los modelos recurrentes comparados para cada número de detecciones utilizado en términos de accuracy y tiempo de inferencia. Los modelos marcados con un diamante corresponden al mejor modelo obtenido para ese tipo de red recurrente.

Cabe mencionar que el gráfico de dispersión de los modelos muestra el desempeño del modelo en contraste con el tiempo de inferencia, donde el modelo deseable se encontraría lo más cercano a la esquina superior izquierda, mostrando un alto desempeño de clasificación y un bajo tiempo de inferencia. El mejor modelo tiende a ser una competencia entre el modelo LSTM y Conv1D para todo número de detecciones. En el caso del modelo Gamma, este posee el mismo tiempo de inferencia para todas sus configuraciones, mientras que para el caso TDNN, al ser solo una configuración probada debido al bajo número de detecciones utilizadas en este trabajo, el mejor modelo corresponde al único modelo entrenado. A medida que aumenta la cantidad de detecciones, el tiempo de inferencia va aumentando para todos los modelos, acercándose cada vez más al tiempo que toma Gamma memory.

### 4.2.3. Cruce con catálogo ALLWISE

El cruce entre la base de datos original ZTFV5 con el catálogo ALLWISE permite incluir 3 metadatos nuevos dados correspondiente a la resta de las magnitudes en la posición del objeto. Esto permite diferenciar la base de datos en 2 tipos: La primera incluyó solo los datos originales obtenidos de ZTF y la segunda incluye los nuevos metadatos entregados por el cruce entre ALLWISE y los datos originales. Para poder medir la efectividad de incluir estos

datos, se entrenó el modelo detallado en la tabla 3.6 con cada una de las bases de datos mencionadas; notar que este modelo solo utiliza metadatos y no imágenes, como también separa los metadatos en estáticos y dinámicos, considerando los nuevos metadatos de ALLWISE como metadatos estáticos. Cada modelo es entrenado 5 veces utilizando los mismos parámetros tanto para el optimizador, como para los hiperparámetros del modelo, con lo cual se obtienen las matrices de confusión mostradas en las figuras 4.5 y 4.6, donde se muestra el accuracy logrado por ambos modelos y su varianza. Un resumen del *accuracy* logrado por los modelos, en conjunto con el valor del test de hipótesis de permutación [99] entre ellos se muestra en la tabla 4.3.

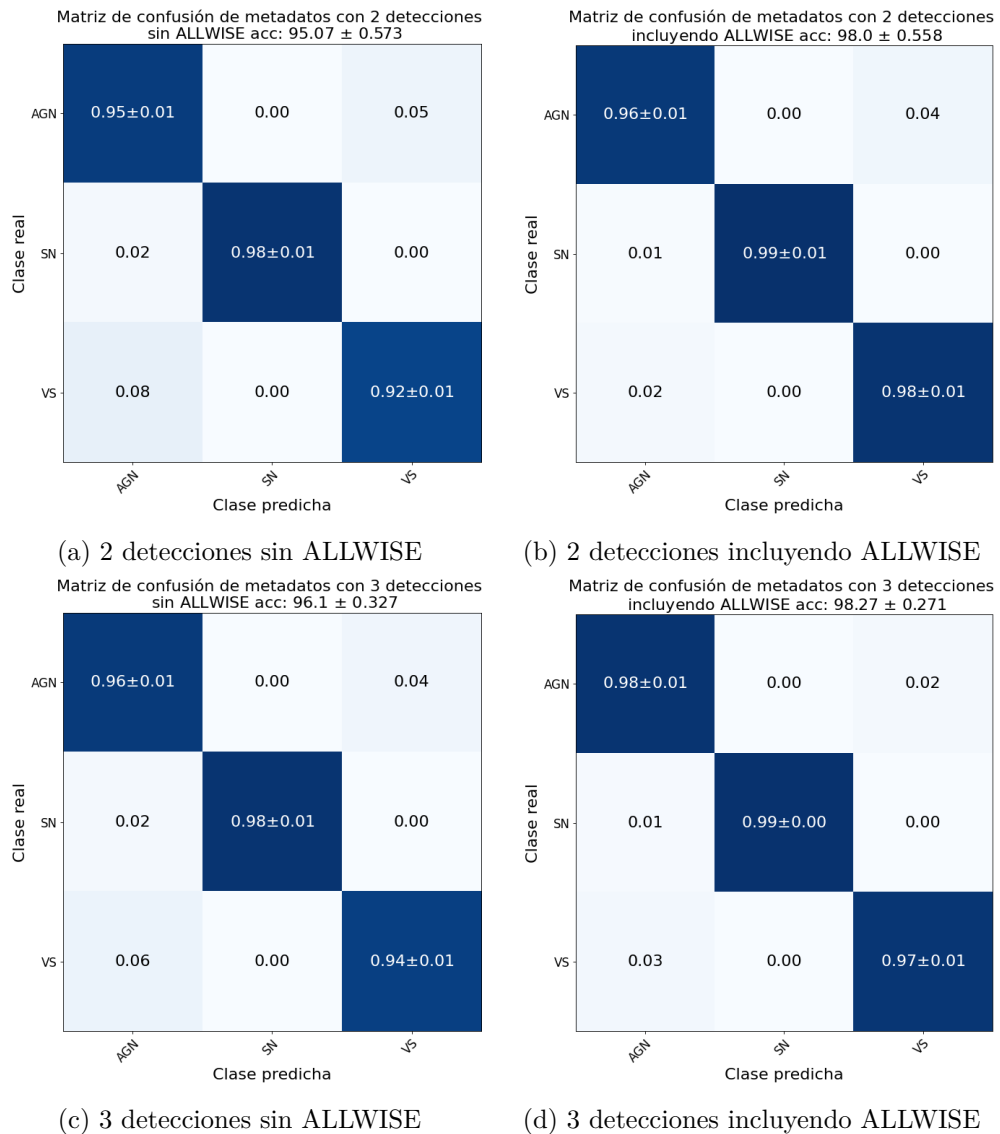
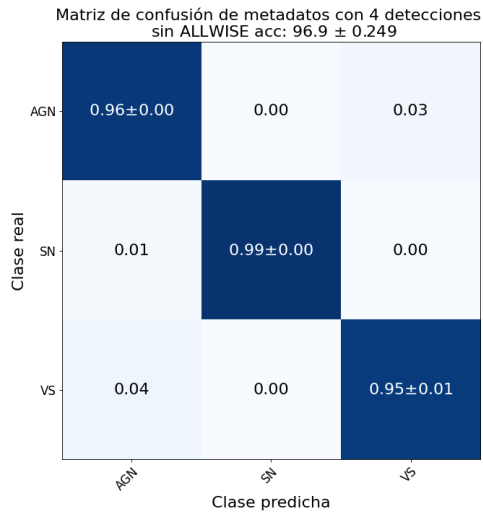
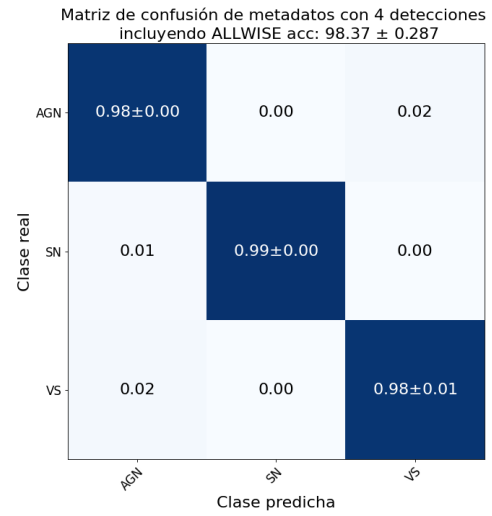


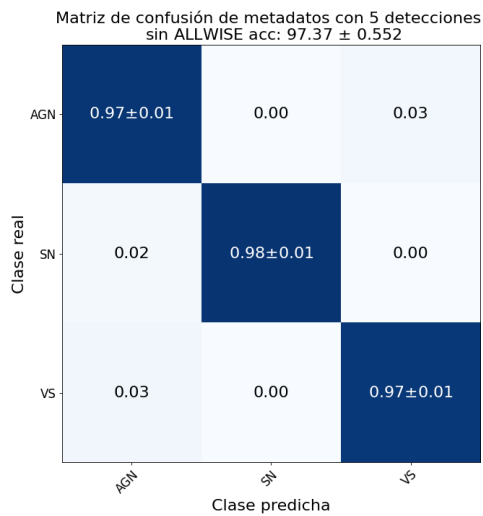
Figura 4.5: Matrices de confusión del modelo usando solo metadatos de la tabla 3.4 comparando el efecto del cruce con el catálogo ALLWISE para 2 y 3 detecciones de un objeto. Las matrices del lado izquierdo (a y c) muestran los resultados de 2 y 3 detecciones respectivamente sin la utilización de ALLWISE, mientras que las matrices del lado derecho (b y d), muestran los resultados de 2 y 3 detecciones respectivamente considerando ALLWISE en el entrenamiento. El modelo no utiliza imágenes.



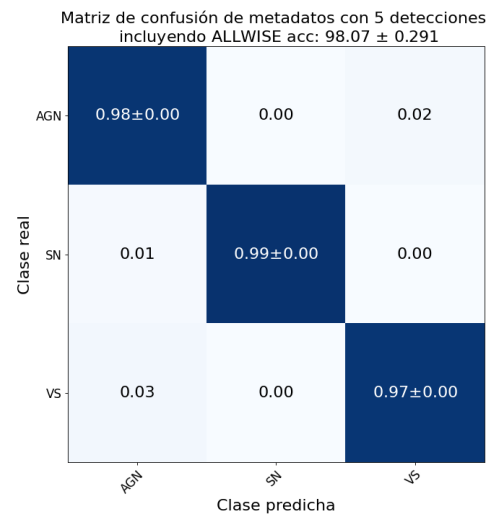
(a) 4 detecciones sin ALLWISE



(b) 4 detecciones incluyendo ALLWISE



(c) 5 detecciones sin ALLWISE



(d) 5 detecciones incluyendo ALLWISE

Figura 4.6: Matrices de confusión del modelo usando solo metadatos de la tabla 3.4 comparando el efecto del cruce con el catálogo ALLWISE para 4 y 5 detecciones de un objeto. Las matrices del lado izquierdo (a y c) muestran los resultados de 4 y 5 detecciones respectivamente sin la utilización de ALLWISE, mientras que las matrices del lado derecho (b y d), muestran los resultados de 4 y 5 detecciones respectivamente considerando ALLWISE en el entrenamiento. El modelo no utiliza imágenes.

Tabla 4.3: Resultados en términos de *accuracy* entre modelos de solo metadatos de la tabla 3.6 que incluyen y no incluyen el cruce con el catálogo ALLWISE, en conjunto con el valor del test de hipótesis de permutación [99] entre ambos modelos para cada número de detecciones.

Batch/ Recorte	2 detecciones	3 detecciones	4 detecciones	5 detecciones
Sin ALLWISE	$95.07 \pm 0.57$	$96.1 \pm 0.32$	$96.9 \pm 0.25$	$97.37 \pm 0.55$
Con ALLWISE	$98 \pm 0.56$	$98.27 \pm 0.27$	$98.37 \pm 0.29$	$98.07 \pm 0.29$
p-value	0.01	0.01	0.008	0.08

## 4.2.4. Modelo final recurrente

### 4.2.4.1. Matrices de confusión

Se utilizó la mejor configuración obtenida en los puntos anteriores, correspondiente a utilizar los metadatos entregados por ALLWISE y 6 rotaciones aleatorias con un grado de libertad de  $14.4^\circ$ . La estructura de la red corresponde a la mostrada en la tabla 3.6, donde la capa *RNN* utilizada depende del número de detecciones considerado, con lo cual se tendrá un modelo por cada número de detecciones, siendo el escogido el mejor modelo obtenido en términos de tiempo de inferencia en la tabla 4.2 y detallado en la tabla 4.4; esto debido a que no existen diferencias estadísticas relevantes entre los modelos, por lo cual se optó por el modelo más rápido para predecir. Cada modelo es entrenado 5 veces, obteniendo las métricas promedio y desviación estándar, mostrados en las matrices de confusión de la figura 4.7. Estos modelos obtienen los menores valores de la función de costos cercanos a la iteración 20.000. Cada uno de los modelos entrenados demora cerca de 3 horas de entrenamiento en una GPU *GTX1080Ti*.

Tabla 4.4: Mejor configuración de red recurrente encontrado para cada número de detecciones según tiempo de inferencia obtenidos en la sección 4.1.2.

	Model	Accuracy en Test	Arquitectura
2 detecciones	Conv1D	$91.3 \pm 0.6$	$Ke = 2$
3 detecciones	Conv1D	$91.6 \pm 0.6$	$Ke = 3$
4 detecciones	Conv1D	$92.5 \pm 0.6$	$Ke = 2$
5 detecciones	LSTM	$93.2 \pm 0.5$	1C 128U

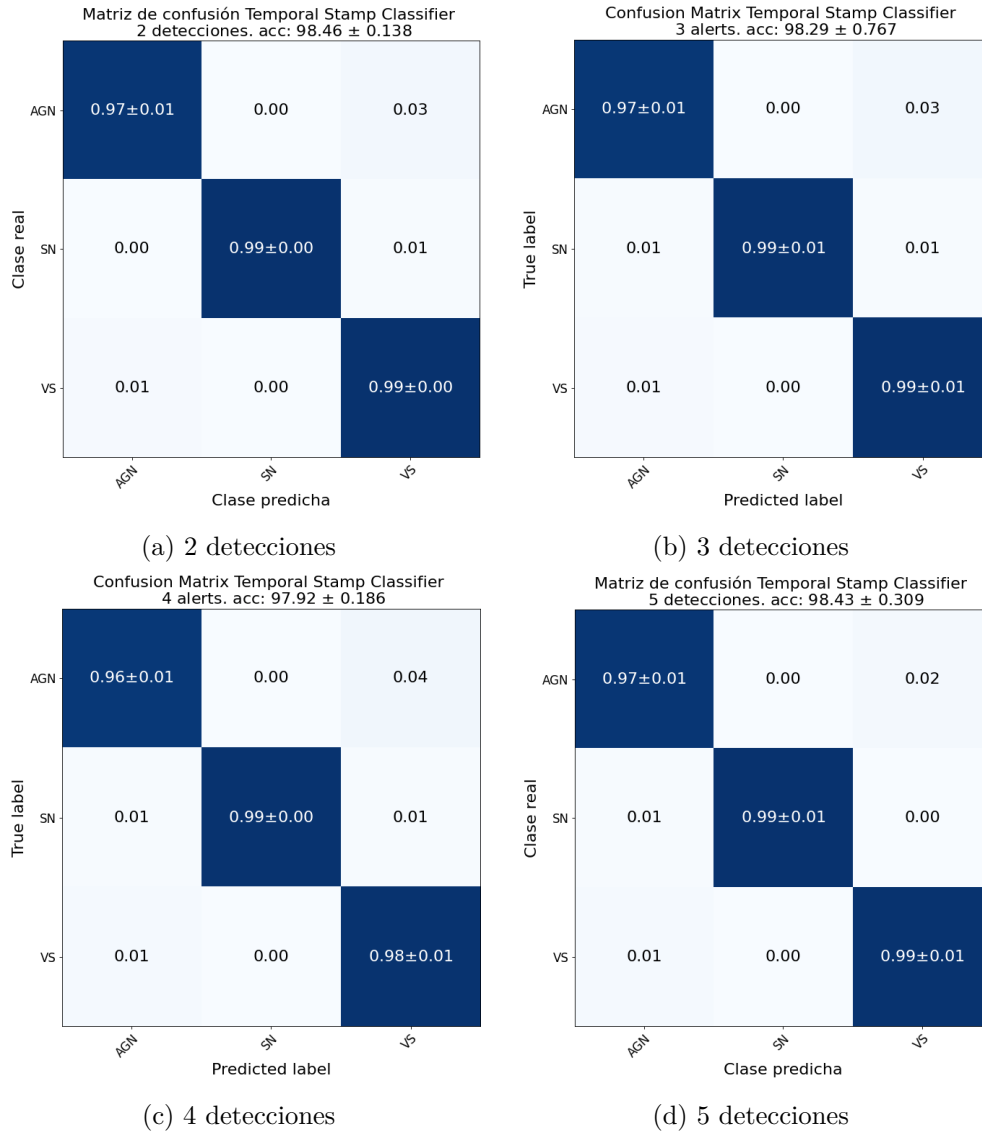


Figura 4.7: Matrices de confusión del mejor modelo obtenido para cada número de detecciones en el conjunto de test. El modelo utiliza imágenes de las detecciones y los metadatos considerando ALLWISE. Los modelos siguen la estructura de la tabla 3.1.

#### 4.2.4.2. Evaluación de predicciones

Una de las características principales de los datos utilizados son las estampillas de cada detección del objeto, donde las características esenciales descritas en el marco teórico deberían presentarse para la clasificación. A forma de poder evaluar que las clasificaciones fallidas de los modelos sean efectivamente casos con una gran dificultad en su clasificación, se grafican algunos ejemplos de estas muestras. Una muestra es candidata a ser errónea si en al menos uno de los modelos entrenados tuvo una mala clasificación. Estos ejemplos se muestran en la figura 4.8.

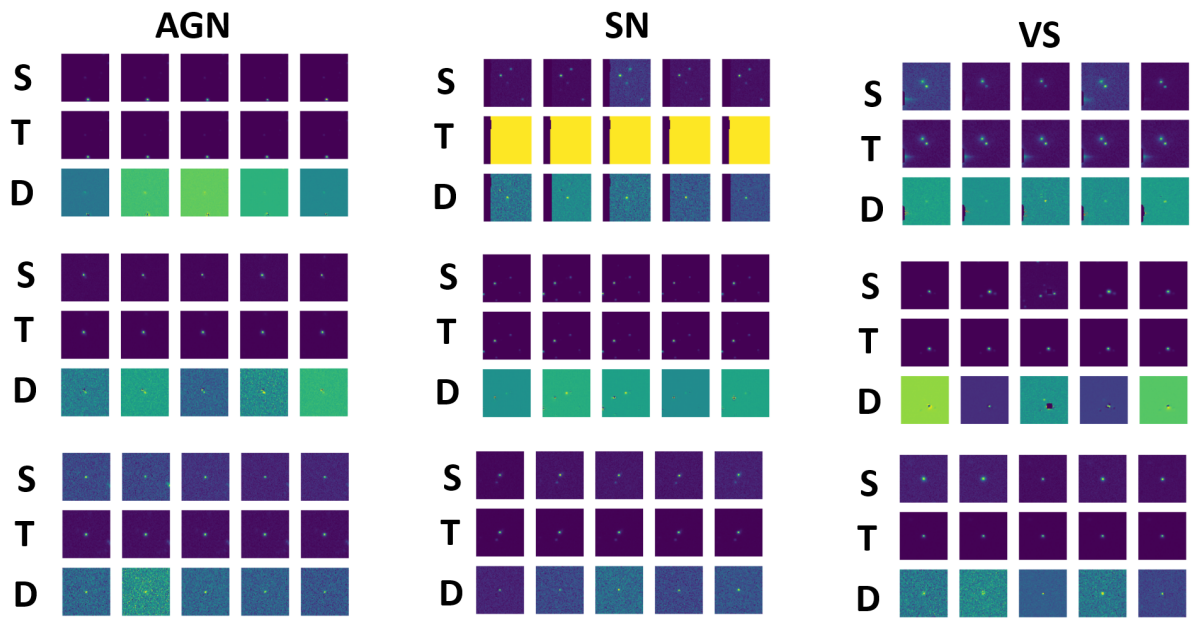


Figura 4.8: Ejemplo de muestras que fueron mal clasificadas por alguno de los modelos completos entrenados. Se muestran las imágenes de las 5 detecciones, donde S corresponde a las imágenes de Science, T a las imágenes de Template y D a las de Difference.

Por otro lado, la posición de los objetos astronómicos es característica para cada uno de los objetos considerados, sobre todo en coordenadas galácticas. A modo de revisar la calidad de las detecciones, se grafican las muestras de toda la base de datos, resaltando los objetos que tuvieron una mala clasificación en alguno de los modelos entrenados. Estas posiciones se muestran en la figura 4.9.



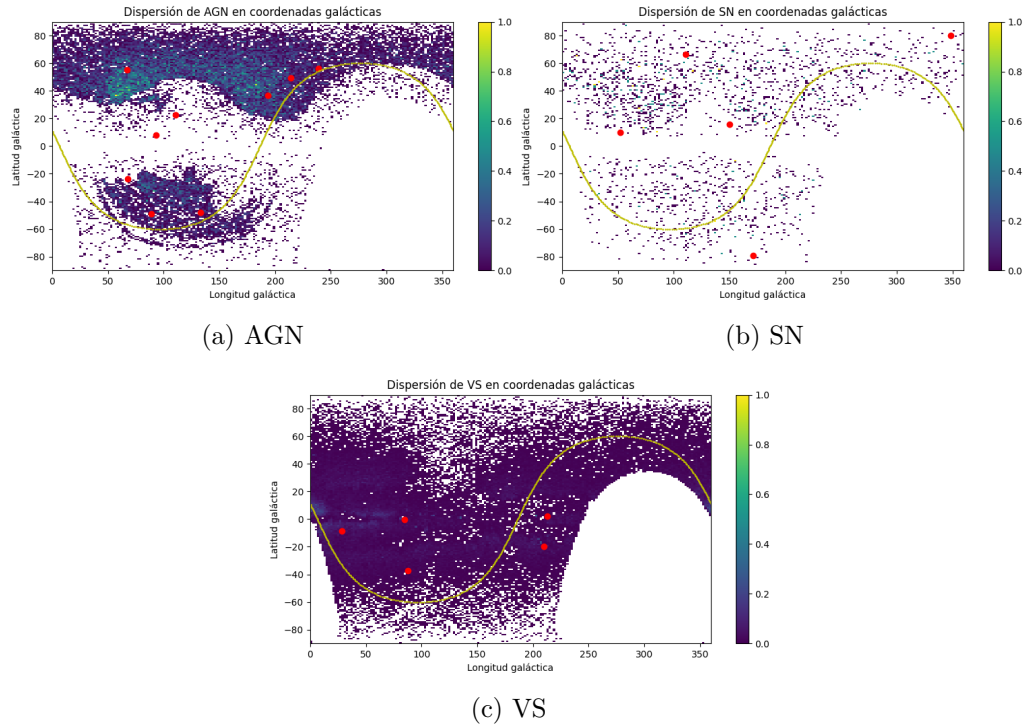
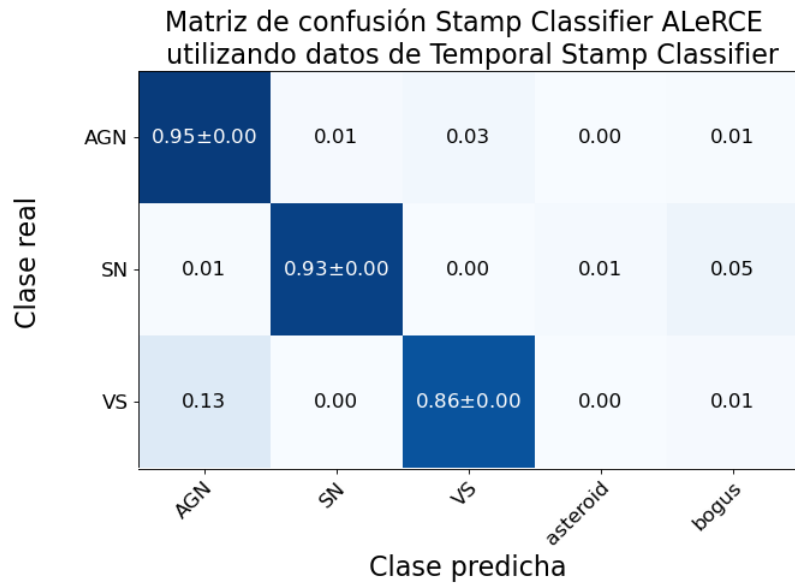


Figura 4.9: Gráfico de dispersión de las 3 clases consideradas en coordenadas galácticas. Los puntos rojos corresponden a ejemplos mal clasificados en al menos una de las inicializaciones de los modelos finales recurrentes en cualquier número de detecciones.

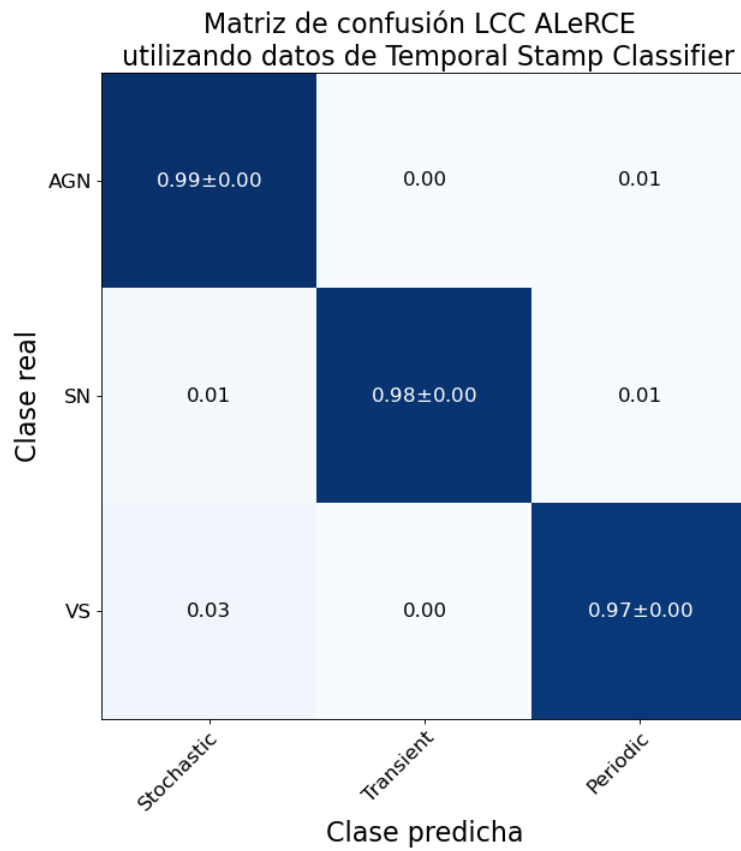
#### 4.2.4.3. Comparación con otros modelos

Utilizando la API de ALerCE<sup>3</sup>, se pueden obtener las predicciones realizadas por el modelo Stamp Classifier y el Light Curve Classifier en su clasificador de las clases superiores para los modelos que se encuentran actualmente en producción. Con estas predicciones, es posible analizar la eficacia del modelo entrenado comparando el desempeño con los modelos que ha sido puesto en producción tanto para el caso del *Stamp Classifier* como para el *Light Curve Classifier*. Utilizando las 600 muestras de la base de datos del *Temporal Stamp Classifier* que fueron ocupadas en el conjunto de test, se consulta la clasificación entregada por ambos modelos para estas muestras, junto a las probabilidades asignadas por los modelos, y se grafica la matriz de confusión para cada modelo, las cuales se presentan en la figura 4.10.

<sup>3</sup> <https://alerce.readthedocs.io/en/latest/>



(a) Stamp Classifier.



(b) Light Curve Classifier.

Figura 4.10: Matriz de confusión de Stamp Classifier y Light Curve Classifier de producción de ALerCE utilizando el conjunto de test de base de datos del Temporal Stamp Classifier.

# Capítulo 5

## Análisis y discusión de resultados

En concordancia con los capítulos 3 y 4, este capítulo se divide en el trabajo realizado para la mejora a la clasificación de la primera detección y el procesamiento y clasificación de secuencias de detecciones. Finalmente, se hablará de las dificultades encontradas y el trabajo futuro para continuar lo expuesto en este trabajo.

### 5.1. Mejoras en el modelo *Stamp Classifier*

Respecto a la clasificación de la primera alerta de objetos astronómicos, en este trabajo se logró proponer mejoras al modelo que permitieran aumentar la cantidad de muestras bien clasificadas, en particular, para la clase SN. Estas mejoras consistieron en modificar el tamaño de la imagen de entrada, aumentar la cantidad de rotaciones utilizadas dejando de lado las transformaciones cuadradas e incluir aleatoriedad en el ángulo de rotación de las imágenes.

Las mejoras posibles al modelo *Stamp Classifier* pueden ser divididas en mejoras en el procesamiento de las imágenes y en los metadatos. Para el caso del procesamiento de las imágenes, es posible evidenciar una gran cantidad de formas para aumentar la cantidad de datos como transformaciones a las imágenes, traslaciones de esta misma, difusión de la imagen, etc. La problemática con estas técnicas viene en que la imagen de la base de datos se encuentra centrada en el objeto de interés, por lo cual al mover este objeto del centro, se pierde la característica principal que distingue el tipo de dato; de la mano con esto, existen ciertos objetos que no se encuentran en el centro que pueden ser de interés para la clasificación de la muestra, como es el caso de galaxias o estrellas cercanas al objeto, por lo cual al momento de aplicar difumisión en la imagen, es posible “cambiar” el objeto de una estrella a una galaxia, por lo cual tiende a confundir más el modelo. En cuanto a la inclusión de mejoras para los metadatos, el camino natural sería incluir información de otros catálogos como metadatos extras; el problema con este acercamiento, es que la mayoría del cruce entre la base de datos ALeRCE-ZTFV5 y otro catálogo tendría que hacerse mediante la posición del objeto. Al realizar esto, objetos móviles como los asteroides y errores de medición como bogus tendrían un metadato asociado a su posición, lo cual pierde sentido considerando que la posición de un asteroide es variable en el tiempo, y la posición de un error puede darse en cualquier momento; esto produce que el modelo tienda a funcionar mejor si se realiza un cruce con otra base de datos, pero al momento de probar el modelo con datos que no ha utilizado tienda a confundirse y fallar en una mayor medida que si no se hubiera agregado el dato. Esto se hizo visible con la inclusión de ALLWISE, dado que al agregar el metadato y

probarlo con más datos, se obtuvo la figura C.3 del Anexo C.2.

El tamaño de la entrada es un parámetro importante para el desempeño del modelo debido a que un mayor tamaño de imagen implica una mayor cantidad de parámetros a ajustar, lo que se traduce en la necesidad de una mayor cantidad de datos, lo cual es problemático para el caso de SN donde se tiene una baja cantidad de muestras, al disminuir el tamaño, baja la cantidad de parámetros y se hace más fácil el entrenamiento, pero se pierde posible información importante que puede estar en las periferias de la imagen. A este problema se le conoce como *maldición de dimensionalidad*. En el trabajo original [10], se probaron 3 valores de recorte, los cuales presentan una gran distancia entre ellos, por lo cual, en esta tesis se probaron valores intermedios que pudieran entregar mejorar el resultado, aunque sea en una baja proporción. Los resultados expuestos en la tabla 4.1 muestran que los mejores modelos tienden a ser los que tienen un recorte mayor al original, o sea, de  $27 \times 27$  y  $33 \times 33$ , como también que incluyen rotaciones no ortogonales de 5 o más rotaciones y con un ángulo de libertad de  $14.4^\circ$  o mayor. Estos resultados indican que agregar estas 3 modificaciones al modelo contribuye a mejorar el desempeño de este mismo, teniendo un valor del valor p del test de hipótesis de permutación menor a 0.01 para los 5 modelos expuestos, siendo de esta manera estadísticamente relevantes.

Al analizar el caso de una rotación aleatoria libre, dado por  $\theta = 180^\circ$ , se tiene que el resultado obtenido no es estadísticamente superior al caso base. Esto está directamente ligado a que este caso las imágenes en el entrenamiento pueden tomar cualquier valor para rotar, por lo cual es probable que 2 rotaciones estén muy cercanas unas de otras, lo cual no aporta información al entrenamiento, perdiendo de esta manera la invarianza rotacional que se quiere lograr con la técnica del Cyclic Pooling.

En cuanto al análisis de los tiempos de inferencia, es posible notar que en todos los tamaños de recorte de la imagen, el tiempo de inferencia no se ve modificado de forma significativa, teniendo los mismos límites de tiempo entre los modelos que va desde  $0.02[s]$  a  $0.04[s]$ , por lo cual no tiene un mayor costo computacional cambiar el tamaño del recorte. Esto también puede ser una consecuencia directa del dataset, el cual parte con una imagen de tamaño  $45 \times 45$ , para luego realizar el recorte, por lo cual el tiempo está directamente ligado a la imagen original del dataset. Por otra parte, algo que se podría esperar es que al aumentar la cantidad de rotaciones aumente el tiempo de inferencia, lo cual se evidencia de la figura 4.1. Por otra parte, algo significativo que se puede extraer de la figura corresponde a que el modelo con 7 rotaciones tiende a ser el más estable dentro de todos los tamaños de recorte, formando una agrupación en cada uno de ellos, mientras que los otros tamaños tienen comportamientos donde a veces se agrupan, y en otros se dispersan considerablemente.

Al analizar los resultados obtenidos y mostrados en la figura 4.2, donde se compara el modelo original con el modelo que considera 6 rotaciones con  $\theta = 14.4^\circ$  y tamaño del recorte en  $33 \times 33$  realizando 20 realizaciones de cada uno. La comparación entre ambos resalta que las modificaciones realizadas aumentan el desempeño del modelo en un 1.5% y con un valor del test de hipótesis de 0.01, lo cual es bastante positivo considerando que se está en un alto valor de *accuracy* logrado con el modelo base, por lo cual subir el porcentaje aún en 1% es un buen aporte y considerando el test de hipótesis obtenido, estos cambios realizados son estadísticamente relevantes. Por otro lado, la precisión de la predicción de SN disminuye en

sus confusiones con la clase asteroide.

Finalmente, se puede declarar que las modificaciones introducidas en este trabajo al modelo *Stamp Classifier* permiten tener un mejor desempeño del modelo, aun cuando el desempeño de SNe no mejore con las modificaciones, pero si las otras clases y los falsos positivos que se pueden generar. Por otro lado, estas modificaciones no complejizan el modelo de manera significativa, por lo cual es posible ser utilizado en producción sin afectar el flujo de los datos.

## 5.2. Clasificación temprana temporal de objetos astronómicos

El modelo *temporal stamp classifier* se presenta como un modelo con arquitectura simple basada en capas convolucionales y recurrentes con buen desempeño, capaz de procesar y clasificar muestras de hasta 5 detecciones provenientes del survey ZTF.

Uno de los aspectos más importantes de este trabajo corresponde al procesamiento de las imágenes de las detecciones, lo cual se realiza utilizando la configuración mejor modelo del Stamp Classifier, entrenado utilizando solo las imágenes de ciencia y referencia. El resultado obtenido con un accuracy de 88.56 % permite tener un buen punto de partida para los pasos posteriores, mostrado también en las funciones de pérdida de la figura B.1.

Un factor importante para analizar las dependencias temporales es la diferencia temporal entre las mediciones de la figura 3.5, donde las 3 clases poseen diferencias muy sutiles pero que pueden ser importantes. Inicialmente, todas tienen un alto valor en 1, esto debido a que muchas detecciones progresivas son muy cercanas unas de otras. En segunda instancia, AGN y VS tienen un comportamiento similar para todas las diferencias temporales, dado por sus características físicas (estocásticas y periódicas respectivamente) que agrupan objetos con comportamientos o periodos muy lejanos entre ellos. Dentro de las VS, se encuentran objetos con periodos muy pequeños (de horas) y otros con períodos muy largos. En el caso de SN, se puede apreciar como la densidad disminuye considerablemente cuando aumenta la diferencia temporal. Esto se debe principalmente a que este tipo de objetos corresponden a explosiones de estrellas, lo cual libera mucha energía y un cambio de luminosidad variante en un corto período de tiempo, por lo cual se detectan muy seguido cuando se encuentran en su período de evento, para luego dejar de emitir luz. Esto produce que las detecciones sean cercanas entre ellas en términos de tiempo. Este análisis permite a priori estimar que la diferencia temporal es un factor importante para la clasificación de estos objetos.

La segunda parte del modelo propuesto se encarga del procesamiento de la dependencia temporal de los datos a través de capas recurrentes. Se exploraron distintas configuraciones para procesar secuencias cortas de imágenes. Los resultados de la tabla 4.2 muestran que los mejores modelos para llevar a cabo la tarea son los tipo LSTM y Conv1D, pero la diferencia estadística entre estos modelos en comparación a los demás no es significativa, teniendo un valor p del test de hipótesis significativamente alto. Al analizar en más detalle entre modelos, se puede apreciar como el comportamiento de GRU y LSTM tiende a ser similar para todas las detecciones, teniendo del test de hipótesis siempre cercano a 1, haciendo que la decisión entre ellos venga de la complejidad que poseen. Con respecto al modelo Conv1D, este pre-

senta un comportamiento competitivo para todas las detecciones trabajadas aún no siendo un modelo recurrente y que asume implícitamente que todas las muestras están regularmente muestreadas, siendo cercano al máximo (obtenido por LSTM) para el caso de 2 y 3 detecciones, y siendo el mejor modelo para 4 y 5 detecciones. Esto implica que la irregularidad del muestreo no es una característica esencial para la separación de las clases, lo cual se puede deber a la baja cantidad de detecciones trabajadas.

Los modelos RNN y TDNN tienden a tener un comportamiento bastante similar para todas las detecciones, teniendo un resultado más bajo que los otros modelos para el caso de 2 detecciones. En el caso del modelo *Gamma Memory*, el resultado aumenta progresivamente a medida que se aumenta la cantidad de detecciones, teniendo un resultado similar y competitivo a los demás modelos en el caso de 5 detecciones. Esto se puede deber directamente a que la recurrencia no se puede procesar de forma correcta con tan pocas detecciones, considerando que el factor  $\gamma$  de este modelo es compartido por todos los filtros que la componen. Estos resultados permiten aprobar la hipótesis planteada en esta tesis que los modelos simples pueden tener resultados competitivos con los modelos complejos en bases de datos de secuencias cortas y corroborado por los valores alcanzados del test de hipótesis de permutación.

Otro factor que resalta es que para los modelos complejos y *RNN* se tiene que la mejor configuración encontrada tiende a ser la más sencilla, con solo una capa del modelo recurrente, pero variando la cantidad de unidades presentes en la capa; en cambio, para el modelo de *Gamma Memory*, se obtienen con un alto valor de  $\gamma$ , lo cual indica una alta cantidad de filtros para poder procesar la recurrencia, haciendo el modelo significativamente más complejo y para el caso de *Conv1D*, se tienen con una baja cantidad de capas para el procesamiento, siendo de  $D - 1$  para el caso de 3 y 5 detecciones en cuanto al número de filtros.

Este análisis se complementa con la figura 4.4, la cual relaciona el *accuracy* logrado por todos los modelos entrenados con su respectivo tiempo de inferencia para una muestra, donde para la mayoría de los casos se tiene que la diferencia entre los modelos tiende a ser mínima tanto en *accuracy* como en tiempo de inferencia, con excepción de los modelos *Gamma* y *Conv1D*, que tienen un tiempo de inferencia considerablemente menor a los otros modelos para el caso de 2, 3 y 4 detecciones. La diferencia de *accuracy* entre los modelos recurrentes puede ser explicado directamente por el carácter aleatorio del entrenamiento, lo cual hace que los modelos lleguen a soluciones cercanas pero distintas en sus parámetros; en cambio la diferencia en el tiempo de inferencia se puede deber simplemente a consecuencias de las lecturas de los datos, debido principalmente a que los datos se encuentran serializados en *TFRecords*, lo cual hace que su lectura no sea en conjunto, si no unitaria, conllevando a una dependencia del estado actual del disco duro donde se encuentra almacenada la información. Este análisis no aplica para el caso de *Gamma Memory*, debido a que el tiempo de ejecución es siempre constante producto de la implementación realizada, que calcula todos los filtros independiente de la cantidad elegida. Para el caso de *Conv1D*, se puede apreciar como este tiene un tiempo de inferencia mucho menor a los demás modelos para los casos de 2, 3 y 4 detecciones, pero siendo similar o incluso superior en el caso de 5 detecciones. Esta diferencia en el caso de 5 detecciones se analizó en más detalle, donde se encontró que la implementación utilizada depende de una capa llamada *Sequential* que produce la lentitud en el cálculo debido a que no permite optimizar las llamadas a las capas convolucionales. Se propone de esta forma cambiar esta implementación en una futura revisión para poder acortar este tiempo de procesamiento.

Si se analiza de forma unitaria, se puede apreciar como a medida que aumenta la cantidad de detecciones utilizadas, los tiempos de inferencia de los modelos tienden a acercarse al de *TDNN*, *Gamma memory* y *Conv1D*, y la resolución, sobre todo en el eje x de tiempo de inferencia, tiende a ser mucho mayor, haciendo que las muestras parezcan más separadas, pero la diferencia sigue siendo de 0.005 segundos. Esto permite reafirmar lo dicho con la tabla 4.2 de que los modelos simples poseen un resultado competitivo e incluso superior para *Conv1D* con un menor tiempo de procesamiento, mostrándose en la misma agrupación que los modelos complejos, similar a un cluster. Estos resultados permitirían elegir los modelos más simples en contraste con los complejos sin pérdida significativa de desempeño debido a que el tiempo de procesamiento entregado por los modelos simples es menor para baja cantidad de detecciones y los *accuracy* logrados por estos tiende a ser muy cercano a los de modelos complejos.

La inclusión de ALLWISE como metadato estático en el entrenamiento del modelo con solo metadatos, permite mejorar el resultado de la clasificación en cerca del 2% para el caso de 2 y 3 detecciones, cerca del 1.2% para el caso de 4 detecciones y en menor medida para 5 detecciones, como se muestra en la figura 4.5 y 4.6. El principal aporte que genera este metadato es arreglar la confusión presente entre VS y AGN, las cuales tienden a confundirse entre ellas con los datos originales; esto se debe principalmente a que esta característica es sumamente utilizada para la detección y separación de AGN [11], también conocido como *colores*, reduciendo en un 3% la cantidad de muestras mal clasificadas de VS y en un 2% de muestras mal clasificadas de AGN en el caso de 2 detecciones, siendo menor el impacto a medida que se aumenta la cantidad de detecciones. Gracias a esto, la clasificación de las muestras tiende a ser constante en sus resultados a partir de las 2 detecciones solo considerando metadatos, y con un alto valor de *accuracy* (mayor a 98%), lo cual permite tener una buena separación de las clases en una etapa temprana de las detecciones y sin necesidad de esperar las 6 detecciones necesarias para poder utilizar la curva de luz en el clasificador tardío [11]. Cabe mencionar que la separación de metadatos entre estáticos y dinámicos viene de los resultados obtenidos en los experimentos extras del punto 3.1.7.4, donde se encontró que esta separación mejora el desempeño del modelo al entregarle información a priori al modelo de que ciertos metadatos no deben tener dependencia temporal, mientras que la decisión de catalogar las características ALLWISE como estáticas viene a que su valor se obtiene a partir de la posición del objeto, por lo cual es constante para todas sus detecciones debido al supuesto de que la posición de los objetos es constante (o tiene una varianza muy pequeña dada solo por error de medición).

Al analizar el modelo final, el cual incluye tanto las imágenes de todas las detecciones como los metadatos de ALLWISE, es posible apreciar como el *accuracy* logrado por este modelo aumenta al obtenido solo con imágenes como el que considera solo metadatos. La mejora del modelo al incluir ambos tipos de datos ya había sido postulado en [10], por lo cual esto solo es una corroboración de este postulado. En el caso de 2, 3 y 5 detecciones, el aumento de *accuracy* es de tan solo unas décimas, mientras que para el caso de 4 detecciones, se tiene una pequeña disminución de este valor, como también se tiene que los resultados para los 4 casos son bastante similares, lo cual puede significar que se ha alcanzado un tope en cuanto al *accuracy* que se puede lograr con un modelo en la clasificación de estas muestras, lo cual se corrobora con la comparación de este modelo con el clasificador de curvas de luz, obteniendo

resultados muy similares para las 3 clases.

La pequeña disminución de *accuracy* tanto del modelo final como con el modelo que incluye solo metadatos en el caso de 5 detecciones se puede deber principalmente a una casualidad estadística, significando que no es un evento aislado que requiera mayor preocupación. Cabe mencionar además que si se compara el modelo final con el que incluye solo los metadatos, los resultados son bastante similares, por lo cual se puede deducir que el mayor aporte en términos de clasificación viene de los metadatos, lo cual se condice con la tabla 4.2, donde los resultados con imágenes no superan el 93 % de *accuracy* para todas las configuraciones probadas; de todas formas, la inclusión de las imágenes permite reducir la varianza de los modelos, por lo cual se puede deducir que, a pesar de no tener un aporte mayor en el *accuracy* total, permite tener resultados más consistentes.

En cuanto a la validación de los resultados obtenidos, se puede apreciar en los errores de predicción que se muestran en la figura 4.8 que existen algunas imágenes que poseen ciertas secciones de color púrpura constante, los cuales corresponden a secciones de la imagen con valores NaNs que fueron llevados a 0 en el preprocesamiento, siendo estos valores errores en la captura de la imagen. Estas muestras fueron justamente las muestras donde la mayoría de las inicializaciones realizadas tuvieron problemas para clasificar, teniendo una baja cantidad de modelos donde se pudo clasificar en su clase correcta; por otro lado, otros errores corresponden a muestras que fueron mal clasificadas por al menos una inicialización de todos los modelos finales, por lo cual su mala clasificación puede deberse a un tema estadístico o a simplemente ser muestras con algún comportamiento extraño que favorece la confusión, como por ejemplo una posición extraña dentro del espacio, recordando que las 3 clases trabajadas tienen un comportamiento bastante particular en cuanto a la posición que ocupan, sobre todo en coordenadas galácticas. La figura 4.9, la cual muestra las posiciones de los objetos en coordenadas galácticas, donde los puntos rojos corresponden a estas muestras mal catalogadas por algún modelo, donde se puede ver que para los gráficos de AGN y SN, existen ciertas muestras que escapan al comportamiento regular de los objetos, estando en puntos con baja densidad de muestras. Respecto a muestras que no tienen ninguna de estas dos características, el análisis de por qué el modelo pudo confundirse depende ya de características astronómicas, lo cual requiere un análisis profundo por parte de un experto.

Finalmente, al analizar la comparación entre el modelo *Temporal Stamp Classifier* en contraste con el *Stamp Classifier* y el *Light Curve Classifier*, es notorio ver como el modelo propuesto tiene resultados comparativamente similares a los obtenidos por este segundo modelo y en consecuencia, superiores a los obtenidos con el modelo *Stamp Classifier* original para todas las cantidades de detecciones consideradas en este trabajo. Esto permite afirmar que el objetivo de esta tesis de construir un modelo que pudiera llenar el espacio entre la primera detección y la curva de luz pudo ser logrado, pudiendo tener resultados similares al mejor modelo que se tiene actualmente para la clasificación en primera instancia de las muestras astronómicas.



# Capítulo 6

## Conclusiones

Este capítulo se divide en 2 secciones principales: clasificación temprana temporal de objetos astronómicos y mejoras al modelo *Stamp Classifier*. En cada una de las secciones se menciona si se cumplieron las hipótesis planteadas en esta tesis, como también los objetivos generales y específicos. Se agrega además una sección de trabajo futuro y últimos comentarios.

### 6.1. Mejora al modelo *Stamp Classifier*

Se logró agregar modificaciones al modelo *Stamp Classifier* original para la predicción de la primera detección de un objeto astronómico mejorando los resultados expuestos en el trabajo original. Las modificaciones propuestas comprenden el cambio en el tamaño de la imagen de entrada, aumento del tamaño del batch, inclusión de una mayor cantidad de rotaciones y añadir un grado de libertad para la rotación de las imágenes. Cada uno de estos cambios fue analizado de forma independiente para analizar su impacto en el modelo, donde los que tuvieron mayor relevancia en términos estadísticos fueron la inclusión de más rotaciones y la libertad de rotación de la imagen.

Se logró aumentar el *accuracy* de  $91.28 \pm 0.77$  a  $92.74 \pm 0.556$ , siendo un aumento de un poco más de 1% en *accuracy* y una reducción significativa en la desviación del resultado, lo cual entrega un valor del test de hipótesis  $p = 0.01$ , por lo cual cumple con ser estadísticamente relevante al cumplir con el criterio del 5%. También se pudo mostrar que la inclusión de nuevas formas para el tratamiento de las imágenes de entrada permiten tener un mejor procesamiento de éstas y, por lo tanto, un mejor resultado.

### 6.2. Clasificación temprana temporal de objetos astronómicos

Se logró extender el modelo *Stamp Classifier* a un nuevo modelo llamado *Temporal Stamp Classifier*. Utilizando información de distintas alertas de objetos astronómicos compuestos por imágenes y metadatos provenientes del sondeo ZTF se logró entrenar el modelo para clasificar entre las clases AGN, SN y VS. El modelo está diseñado en base a una CNN para el procesamiento de las imágenes de las detecciones, pudiendo realizar la extracción de características de forma automática y rápida, mientras que el problema de recurrencia es resuelto mediante redes RNN, procesando tanto los metadatos recurrentes como las características

obtenidas de las imágenes de todas las detecciones consideradas. Las entradas del modelo son las imágenes de ciencia de todas las detecciones, la imagen de referencia del objeto, los 23 metadatos de todas las detecciones y los 3 metadatos obtenidos con el cruce con el catálogo ALLWISE. La arquitectura convolucional del modelo posee 6 rotaciones equiespaciadas, con un grado de libertad de rotación de  $14.4^\circ$ , lo cual agrega invarianza a la rotación de las muestras.

Para las 3 clases que identifica el modelo *Temporal Stamp Classifier* se logra un valor de *accuracy* en el conjunto de test de  $0.984 \pm 0.3$  en promedio entre todas las detecciones, utilizando un conjunto de test balanceado de la base de datos de *Temporal Stamp*. Este valor obtenido es suficiente para declarar que se cumple la hipótesis de que agregar más detecciones al modelo mejora el desempeño de clasificación para las clases AGN, SN y VS. Esto permite concluir que se logra el primer objetivo de esta tesis. Por otra parte, la comparación entre los distintos modelos de redes recurrentes en 2, 3, 4 y 5 detecciones mostró que la diferencia entre los resultados obtenidos no tiene significancia estadística entre ellos, por lo cual los modelos de recurrencia simple pueden ser aplicados en el caso de baja cantidad de detecciones con resultados competitivos a los obtenidos con modelos más complejos de recurrencia, logrando el segundo objetivo de esta tesis.

## 6.3. Propuestas de trabajo futuro

### 6.3.1. Clasificación temprana temporal de objetos astronómicos

La arquitectura utilizada en este trabajo se basó en gran medida en la arquitectura para el procesamiento de la primera detección [10], lo cual considera la utilización de varios filtros convolucionales para el procesamiento de las imágenes. Estos filtros conllevan una gran cantidad de parámetros, lo cual se traduce en varias multiplicaciones de matrices, las cuales pueden llevar un tiempo considerable de procesamiento. Uno de los experimentos extras de este trabajo consistió en reducir la cantidad de multiplicaciones utilizando filtros de  $1 \times 1$  entre los filtros convolucionales, lo cual demostró tener un desempeño competitivo al caso de no incluirlos. Por otro lado, los resultados de utilizar los metadatos dinámicos y estáticos para la clasificación entrega un resultado bastante cercano al del modelo final, por lo cual se puede interpretar que el mayor aporte para la clasificación está dado por los metadatos mismos y el procesamiento de la imagen viene a dar un apoyo extra para conseguir el resultado final, además de poder visualizar la muestra. En base a esta hipótesis, se propone como trabajo futuro cambiar la arquitectura del procesamiento de la imagen por una configuración más simple, con menor cantidad de filtros y capas, pero que obtenga un resultado cercano al original, con lo cual se reduciría considerablemente el tiempo de procesamiento de las muestras.

Por otro lado, considerando que los mecanismos de atención han tenido buenos resultados en el procesamiento de las imágenes, se propone cambiar la configuración de filtros convolucionales con mecanismos de atención tipo Multi Head [100]. Este tipo de red tiene la particularidad de tener buen desempeño para el procesamiento tanto de relaciones espaciales como temporales, por lo cual sería una buena alternativa para este problema que presenta ambas problemáticas. El principal desafío de este acercamiento viene por la cantidad de muestras, debido a que este tipo de arquitectura suele necesitar una gran cantidad de datos

para ser ajustados, lo cual no es un problema para las clases VS y AGN, pero si para las SN, por lo cual se hace necesario aumentar la cantidad de muestras mediante nuevas detecciones o utilizar enfoques generativos para la inclusión de muestras generadas. Si se opta por este camino, una forma de enfrentar el problema vendría dado por entrenamiento por separado, donde inicialmente se entrenaría solo la parte para el procesamiento de las imágenes del modelo, posteriormente, y de forma independiente, se entrenaría la arquitectura para el procesamiento de los metadatos, y finalmente se juntarían ambos modelos entrenados, realizando una sintonización final de los pesos para trabajar en conjunto.

Estudios de la clasificación de sondeos astronómicos han demostrado que la inclusión del tiempo de medición es esencial para el buen desempeño de los modelos [24, 25], pero en [26] se mostró que un pre procesamiento de este dato permite mejorar sustancialmente los resultados obtenidos. Siguiendo estos resultados, se propone modificar la forma utilizada en este trabajo, donde se tomó el logaritmo de la diferencia de tiempos de medición, con lo cual se propone la utilización de algún método de modulación del tiempo como la utilizada en [26], donde esta modulación se aplica a curvas de luz de SN. La principal dificultad de utilizar esta modulación en el trabajo de detección temprana viene dado por la cantidad de detecciones debido a que si solo se cuenta con 2 o 3 muestras, es complejo realizar una modulación que incluya la estimación o uso de un periodo. De todas formas, se propone basarse en la modulación temporal de [26] para realizar un procesamiento distinto del tiempo como trabajo futuro.

La inclusión de modelos simples de recurrencias permitió encontrar que este tipo de redes tienen desempeños competitivos a los obtenidos con redes complejas que han sido estudiados en la literatura. El desempeño de estos modelos simples cumple con el desempeño, pero quedan atrás en términos de tiempo de inferencia debido a que fueron programadas de forma manual utilizando la librería `Tensorflow`, sobre todo el caso de Gamma Memory. Una forma de mejorar este desempeño es mejorar la eficiencia del código confeccionado, sacando provecho a técnicas de optimización para el manejo de los datos recurrentes.

Uno de los objetivos más importantes de este trabajo es construir un modelo que permita clasificar entre las 3 clases de más alto nivel de los objetos astronómicos, acercando el resultado obtenido con la primera alerta al obtenido con la curva de luz del objeto. El desempeño del modelo permitió encontrar que con tan solo 3 alertas es posible tener resultados similares a los obtenidos con la curva de luz de 6 detecciones en la misma banda, pero utilizando imágenes y metadatos de las alertas, por lo cual, es posible aumentar la complejidad del problema en cuanto a la cantidad de clases. Una posible mejora sería expandir las clases utilizadas de forma progresiva; un ejemplo de esto sería separar SN en 2 sub clases y proceder con el entrenamiento, o separar las estrellas variables en sus subcategorías; todo esto dejando las otras 2 clases sin modificaciones. El objetivo de solo expandir una clase viene a que si el modelo se equivoca en clasificar la SNIa en SNII, esta seguiría siendo una clasificación relativamente correcta, ya que sigue considerando que la muestra corresponde a una SN. Para poder llevar a cabo esto, que fue considerado para realizarse en este trabajo, es necesario realizar un cruce entre la base de datos que se posee con la base de datos utilizada en [11] u alguna otra similar, de manera de poder obtener las subclases de los objetos que se utilizaron en este trabajo; esto conllevaría a que algunas muestras no tendrían sus subclases, lo cual reduciría el tamaño de la base de datos. De todas formas, este acercamiento podría abrir una posibilidad de mejora en la clasificación fina de objetos astronómicos.

### 6.3.2. Mejoras al modelo Stamp Classifier

El desempeño del Stamp Classifier se ve fuertemente diezmado debido a su menor desempeño en la clasificación de SN. Este comportamiento se debe principalmente a la baja cantidad de muestras de esta clase en la base de datos utilizada, lo cual produce una baja representatividad de estas en el conjunto de entrenamiento. El aumento progresivo de la cantidad de estas muestras podría ser una oportunidad de mejora radical, considerando que actualmente, en el equipo de ALERCE, se trabaja activamente en la identificación de SN utilizando el SN Hunter, siendo incluidas en nuevas versiones del dataset utilizado. Por otro lado, gracias al avance de técnicas generativas como las GANs o GPTs, se podrían generar nuevas muestras de imágenes de SN a partir de estos modelos, los cuales permitirían tener imágenes sintéticas similares a las muestras reales de la base de datos, teniendo una mejor forma de balancear la base de datos que la simple replicación de muestras utilizada tanto en este trabajo, como en el trabajo original. El problema con los modelos generativos es la gran cantidad de muestras que se requieren para ser ajustados, como también la necesidad de incluir metadatos en la generación, los cuales poseen condiciones físicas específicas que no pueden ser ignoradas, y no necesariamente se relacionan con lo generado en las imágenes, por lo cual se requeriría de algún modelo que modele de forma adecuada las características de una SN, como es el caso del modelo SPM [26].

Una forma sencilla de explorar mejoras al modelo propuesto consiste en mejorar la búsqueda de hiperparámetros utilizada, incluyendo un mayor tamaño del batch utilizado, lo cual requiere una mayor capacidad de procesamiento que el utilizado. Esto permitiría tener una mejor generalización de los resultados y, en consecuencia, un mejor desempeño en los conjuntos de validación y test. En cuanto a la búsqueda de hiperparámetros, actualmente existen librerías dentro de `python` que permite realizar esta búsqueda de una manera más eficiente como *RayTune* [101] y *hyperopt* [102].

Otra forma de mejorar este modelo viene de la inclusión de ALLWISE como metadato, el cual es dependiente de la posición del objeto. La inclusión de este metadato fué removida en este punto del trabajo debido a que las clases Asteroides y bogus no poseen una relación con la posición del objeto al ser objetos en movimiento y errores respectivamente, por lo cual solo ensucian los resultados, como se puede apreciar en la figura C3 del Anexo C.4. De todas formas, se encontró en este trabajo que si son muy útiles para poder identificar las otras 3 clases, por lo cual podría realizarse un modelo en cascada, donde inicialmente se tenga un clasificador que permita separar entre 3 clases: Bogus, asteroides y Objetos. Teniendo esta clasificación, se tomaría solo los ejemplos dentro de la clase Objetos, y estos serían clasificados dentro de las 3 clases restantes: SN, AGN y VS, con la salvedad que en el clasificador secundario se utilizarían también las características ALLWISE. Utilizando este acercamiento, se podría sacar provecho de las ventajas de ALLWISE, y se evitaría el problema de los objetos falsos y móviles dentro de la primera detección.

De los gráficos presentados sobre posiciones de objetos en coordenadas galácticas en la figura 4.7 y en [10] se encuentran ciertos objetos que son catalogados como su clase, pero no presentan el comportamiento regular de su clase, los cuales son conocidos como objetos *outliers*. Actualmente, existen métodos automáticos para la detección de estos objetos

[103, 104], los cuales dependen de la curva de luz para su detección, siendo una complicación debido a la necesidad de la construcción de esta curva, la cual necesita un mínimo de 6 detecciones en la misma banda. Bajo esto, se postula como trabajo futuro adaptar el modelo *Stamp Classifier* o el modelo *Temporal Stamp Classifier* para la detección de estos objetos utilizando técnicas de cuantificación de la incerteza, los cuales han tenido un nuevo acercamiento utilizando operadores de RKHS<sup>4</sup> [105], teniendo resultados más rápidos que los acercamientos bayesianos [106] y permitiendo analizar cuanto se debe confiar en la predicción realizada por el modelo.

Finalmente, tal como se mencionó para el caso de detección recurrente, una forma de explorar nuevas formas de procesar las detecciones sería utilizar estructuras de atención para el procesamiento de las imágenes, las cuales han demostrado tener un buen desempeño para obtener las características de las imágenes de entrada. El desempeño de estos modelos está directamente relacionado con la cantidad de muestras que se utilizan para entrenar, por lo cual sería necesario tener una base de datos más actual para esta idea, la cual incluya un mayor número de muestras sobre todo de la clase SN.

## 6.4. Comentarios finales

El trabajo realizado y expuesto en esta tesis permitió llenar el vacío existente entre la clasificación de la primera detección de un objeto y la curva de luz de este, obteniendo resultados competitivos con los modelos actualmente desarrollados. Por otro lado, se pudo incluir mejoras al modelo de la primera detección, aumentando la capacidad del modelo para la detección de objetos y, por sobre todo, de SN.

En general, esta tesis abordó un problema como objetivo principal correspondiente al procesamiento de las detecciones temporales, pero debido a su gran parecido con el procesamiento de la primera detección, se trabajó también en esta dirección, abriendo posibilidades para la mejora en varios aspectos lo cual beneficia a la investigación astronómica para el procesamiento de grandes cantidades de datos, los cuales solo tienden a aumentar en volumen en los últimos años.

---

<sup>4</sup> RKHS: Kernels en Espacios de Hilbert

# Bibliografía

- [1] Tonry, J. L. , Denneau, L., Heinze, A. N., Stalder, B., Smith, K. W., Smartt, S. J., Stubbs, C. W., Weiland, H. J., Rest, A., et al “ATLAS: A High-Cadence All-Sky Survey System”, *Publications of the Astronomical Society of the Pacific*, Vol. 130, no. 988, p. 064505, 2018.
- [2] Aihara, H., Arimoto, N., Armstrong, R., Arnouts, S., Bahcall, N. A., Bickerton, S., Bosch, J., Bundy, K., Capak, P. L., Chan, J. H., et al. “The hyper supprime-cam ssp survey: overview and survey design”, *Publications of the Astronomical Society of Japan*, Vol. 70, 2018.
- [3] Bellm, E. C., Kulkarni, S. R., Graham, M. J., Dekany, R., Smith, R. M., Riddle, R., Masci, F. J., Helou, G., Prince, T. A., Adams, S. M., et al. “The Zwicky Transient Facility: System Overview, Performance, and First Results”, *Publications of the Astronomical Society of the Pacific*, Vol. 131, No. 995, P. 018002, 2018.
- [4] Grigorievskiy, A., Mantere, M., Akusok, A., Eirola, E. & Lendasse, A. “Forecasting the outbursts of the photometry light curve of star v363 lyr”, *ITISE*, 2014.
- [5] Zhang, R. & Zou, Q., “Time series prediction and anomaly detection of light curve using lstm neural network”, *Journal of Physics: Conference Series*, Vol. 1061, 2018.
- [6] Tachibana, Y., Graham, M. J., Kawai, N., Djorgovski, S. G., Drake, A. J., Mahabal, A. A. & Stern, D., “Deep modeling of quasar variability”, *The Astrophysical Journal*, Vol. 903, No. 1, 2020.
- [7] Boone, K., “Avocado: Photometric classification of astronomical transients with gaussian process augmentation”, *The Astrophysical Journal*, Vol. 158, No. 6, 2019.
- [8] Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R. & Hlo, R., “Rapid: Early classification of explosive transients using deep learning”, *Publications of the Astronomical Society of the Pacific*, Vol. 131, No. 1005, P. 118002, 2019.
- [9] Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., Estévez, P. A., Sánchez-Sáez, P., Arredondo, J., Bauer, F. E., Carrasco-Davis, R., Catelan, M., Elorrieta, F., Eyheramendy, S., Huijse, P., Pignata, G., Reyes, E., Reyes, I., Rodríguez-Mancini, D., Ruz-Mieres, D., Valenzuela, C., Alvarez-Maldonado, I., Astorga, N., Borissova, J., Clocchiatti, A., Cicco, D. D., Donoso-Oliva, C., Graham, M. J., Kurtev, R., Mahabal, A., Maureira, J. C., Molina-Ferreiro, R., Moya, A., Palma, W., Pérez-Carrasco, M., Protopapas, P., Romero, M., Sabatini-Gacitúa, L., Sánchez, A., Martín, J. S., Sepúlveda-Cobo, C., Vera, E., & Vergara, J. R., “The automatic learning for the rapid classification of events (alerce) alert broker”, *The Astronomical Journal*, Vol. 161, No. 5, 2020.
- [10] Carrasco-Davis, R., Reyes, E., Valenzuela, C., Förster, F., Estévez, P. A., Pignata, G., Bauer, F. E., Reyes, I., Sánchez-Sáez, P., Cabrera-Vives, G., et al, “Alert Classification

- for the ALeRCE Broker System: The Realtime Stamp Classifier”, *The Astronomical Journal*, Vol. 162, No. 6, 2020.
- [11] Sánchez-Sáez, P., Reyes, I., Valenzuela, C., Förster, F., Eyheramendy, S., Elorrieta, F., Bauer, F. E., Cabrera-Vives, G., Estévez, P. A., Catelan, M., Pignata, G., Huijse, P., Cicco, D. D., Arévalo, P., CarrascoDavis, R., Abril, J., Kurtev, R., Borissova, J., Arredondo, J., Castillo-Navarrete, E., Rodriguez, D., Ruz-Mieres, D., Moya, A., Sabatini-Gacitúa, L., Sepúlveda-Cobo, C., & Camacho-Iñiguez, E., “Alert classification for the ALeRCE broker system: The light curve classifier”, *The Astronomical Journal*, Vol. 161, No. 3, 2021.
- [12] Fukushima, K., & Miyake, S. “Neocognitron: A selforganizing neural network model for a mechanism of visual pattern recognition”, *Competition and cooperation in neural nets*, (pp. 267–285). Springer, 1982.
- [13] Hubel, D. H. & Wiesel, T. N., “Receptive fields of single neurones in the cat’s striate cortex”, *The Journal of Physiology*, Vol. 148, 1959.
- [14] Krizhevsky, A.; Sutskever, I., Hinton, Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks”, *Communications of the ACM*, Vol. 60, pp. 84-90, 2017.
- [15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A., “Going Deeper with Convolutions”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015.
- [16] Simonyan, K. & Zisserman, A., “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *ICLR*, 2015.
- [17] Kaiming, H.; Xiangyu, Z., Shaoqing, R.; Jian, S., “Deep Residual Learning for Image Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE*, pp. 770–778, 2016.
- [18] Wright, D. E., Lintott, C. J., Smartt, S. J., Smith, K. W., Fortson, L., Trouille, L., Allen, C. R., Beck, M., Bouslog, M. C., Boyer, A., et al. “A transient search using combined human and machine classifications”, *Monthly Notices of the Royal Astronomical Society*, Vol 472, No. 2, pp. 1315–1323., 2017.
- [19] Chambers, K. C. & Magnier, E. A., et al “The PAN-STARRS1 surveys”, *ArXiv e-prints*, 2016.
- [20] Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.C. “DeepHiTS: Rotation invariant convolutional neural network for transient detection”, *The Astrophysical Journal*, Vol. 836, No. 1, 2017.
- [21] Sander, D., Willett, K & Dambre, J., “Rotation-invariant convolutional neural networks for galaxy morphology prediction”, *ArXiv preprint ArXiv:1503.07077*, 2015.
- [22] Förster, F., Maureira, J. C., San Martín, J., Hamuy, M., Martínez, J., Huijse, P., Cabrera, G., Galbany, L., De Jaeger, Th., González–Gaitán, S., Anderson, J. P., Kunkarayakti, H., Pignata, G., Bufano, F., Littín, J., Olivares, F., Medina, G., Smith, R. C., Vivas, A. K., Estévez, P. A., Muñoz, R., & Vera, E., “The high cadence transient survey (HiTS). I. Survey design and supernova shock breakout constraints”, *The Astrophysical Journal*, Vol. 832, No. 2, 2018.
- [23] Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., Estévez, P. A., Huijse, P., Protopapas, P., Reyes, I., Martínez-Palomera, J., & Donoso, C. “Deep learning for image

- sequence classification of astronomical events”, *Publications of the Astronomical Society of the Pacific*, Vol. 131, No. 1004, pp. 108006, 2019.
- [24] Arredondo, J., Román, P., Förster, F., “Clasificación de objetos astronómicos mediante sus alertas utilizando aprendizaje profundo”, *Tesis de Magister, Universidad de Santiago de Chile*, 2021.
- [25] Naul, B., Bloom, J. S., Pérez, F., Van der Walt, S., et al, “A recurrent neural network for classification of unevenly sampled variable stars”, *Nature Astronomy*, Vol. 2, pp. 151–155, 2018.
- [26] Pimentel, Ó., Estévez, P. A., Förster, F., “Deep Attention-Based Supernovae Classification of Multi-Band Light-Curves”, *The Astronomical Journal*, Vol. 165, No. 1, 2022.
- [27] Huijse, P., Estévez, P. A., Zegers, P., Príncipe, J., Protopapas, P., “Period Estimation in Astronomical Time Series Using Slotted Correntropy”, *IEEE Signal Processing Letters*, Vol. 18, No. 6, pp. 371–374, 2011.
- [28] Huijse, P., Estevez, P.A., Forster, F., Daniel, S.F., Connolly, A.J., Protopapas, P., Carrasco, R., Principe, J.C., “Robust Period Estimation using Mutual Information for Multi-band Light Curves in the Synoptic Survey Era”, *Astrophysical Journal Supplement Series*, Vol. 236, No. 1, 2018.
- [29] LIGO Scientific Collaboration and Virgo Collaboration, Abbott, B., Abbott, R., Abbott, M., et al “Observation of Gravitational Waves from a Binary Black Hole Merger”, *Physical Review Letters*, Vol. 116, No. 6, p. 061102, 2016.
- [30] Padovani, P., Giommi, P., Resconi, E., Glauch, T., Arsioli, B., Sahakyan, N. & Huber, M. “Neutrino emission from the direction of the blazar TXS 0506+056 prior to the IceCube-170922a alert”, *Science*, Vol. 361, No. 6398, pp. 147–151, 2018.
- [31] Keivani, A., Murase, K., Petropoulou, M., Fox, D., Cenko, S., Chaty, S., Coleiro, A., DeLaunay, J., Dimitrakoudis, S., Evans, P., Kennea, J., Marshall, F., Mastichiadis, A., Osborne, J., Santander, M., Tohuvavohu, A., Turley, C. “Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922a”, *Science*, Vol. 361, No. 6398, p. eaat1378, 2018.
- [32] Thompson, T. A., Burrows, A., Pinto, P. A., “Shock Breakout in Core-Collapse Supernovae and Its Neutrino Signature”, *The Astrophysical Journal*, Vol. 592, pp. 434–456, 2003.
- [33] Forster, F, Moriya, T. J., Maureira, J. C., Anderson, J. P., Blinnikov, S. et al, “The delay of shock breakout due to circumstellar material evident in most type II supernovae”, *Nature Astronomy*, Vol. 2, p. 808, 2018.
- [34] Alcock, C., Allsman, R. A., Alves, D. R., Axelrod, T. S., Becker, A. C. et al, “The MACHO Project: Microlensing Detection Efficiency”, *The Astrophysical Journal Supplement Series*, Vol. 136, No. 2, p. 439, 2001.
- [35] Pojmánski G., “The All Sky Automated Survey”, *Astronomische Nachrichten*, Vol. 325, No. 6-8, pp. 553–555, 2004.
- [36] Szymanski, M. K., “The optical gravitational lensing experiment. internet access to the ogle photometry data set: ogle-II bvi maps and i-band data”, *Acta Astron.*, Vol. 55, pp. 43–57, 2005.



- [37] Izevic, Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E. et al, “LSST: from Science Drivers to Reference Design and Anticipated Data Products”, *Astrophysical Journal*, Vol. 873, No. 2, 2008.
- [38] Ngeow C. C., Gieren, W. & Klein, C., “Distance determination from the Cepheid and RR Lyrae period-luminosity relations”, *Proceedings of the International Astronomical Union*, Vol. 9, No. S301, pp. 123–128, 2013.
- [39] Feast, M. W., Menzies, J. W., Matsunaga, N. & Whitelock, P. A., “Cepheid variables in the flared outer disk of our galaxy”, *Nature*, Vol. 509, No. 7500, pp. 342–344, 2014.
- [40] Schmidt, B. P., Suntzeff, N. B., Phillips, M. M., Schommer, R. A., Clocchiatti, A. et al., “The High-Z Supernova Search: Measuring Cosmic Deceleration and Global Curvature of the Universe Using Type IA Supernovae”, *The Astrophysical Journal*, Vol. 507, pp. 46–63, 1998.
- [41] Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A. et al., “Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant”, *The Astronomical Journal*, Vol. 116, pp. 1009–1038, 1998.
- [42] Mazzali, P. A., Ropke, F. K., Benetti, S., Hillebrant, W., “A Common Explosion Mechanism for Type Ia Supernovae”, *Science*, Vol. 315, No. 5813, pp. 825–828, 2007.
- [43] Hsiao, E. Y., Conley, A., Howell, D. A., Sullivan, M., Pritchett, C. J. et al, “K-Corrections and Spectral Templates of Type Ia Supernovae”, *The Astrophysical Journal*, Vol. 663, pp. 1187–1200, 2007.
- [44] Moriya, T. J., Mazzali, P. A., Tominaga, N., Hachinger, S., Blinnikov, S. I. et al, “Light-curve and spectral properties of ultrastripped core-collapse supernovae leading to binary neutron stars”, *Monthly Notices of the Royal Astronomical Society*, Vol. 466, pp. 2085–2098, 2017.
- [45] Catelan, M & Smith, H. A., “Pulsating Stars”, *Wiley-VCH*, 2015.
- [46] Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S. et al, “Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey”, *Monthly Notices of the Royal Astronomical Society*, Vol. 389, pp. 1179–1189, 2008.
- [47] Breiman, L. “Random Forests”, *Machine Learning*, Vol. 45, No.1, pp. 5–32, 2001.
- [48] Belokurov, V., Evans, N. W. & Du, Y. L., “Light-curve classification in massive variability surveys - I. Microlensing”, *Monthly Notices of the Royal Astronomical Society*, Vol. 341, No. 4, pp. 1373–1384, 2003.
- [49] Belokurov, V., Evans, N. W. & Du, Y. L., “Light-curve classification in massive variability surveys – II. Transients towards the Large Magellanic Cloud”, *Monthly Notices of the Royal Astronomical Society*, Vol. 352, No. 1, pp. 233–242, 2004.
- [50] Brinks, H., Richards, J. W., Poznanski, D., Bloom, J. S., Rice, J. et al, “Using machine learning for discovery in synoptic survey imaging data”, *Monthly Notices of the Royal Astronomical Society*, Vol. 435, pp. 1047–1060, 2013.
- [51] Nun, I., Protopapas, P., Sim, B., Zhu, M., Dave, R. et al, “FATS: Feature Analysis for Time Series”, *Astrophysics Source Code Library*, ascl:1711.017., 2015.
- [52] Benavente, P., Protopapas, P. & Pichara, K., “Automatic Survey-invariant Classification of Variable Stars”, *The Astrophysical Journal*, Vol. 845, No. 2, p. 147, 2017.

- [53] Castro, N., Protopapas, P. & Pichara, K., “Uncertain Classification of Variable Stars: Handling Observational GAPS and Noise”, *The Astronomical Journal*, Vol. 155, No. 1, p. 16, 2018.
- [54] Nair, V., Hinton, G. “Rectified Linear Units Improve Restricted Boltzmann Machines”, *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10. USA: Omnipress, pp. 807-814, 2010.
- [55] Cybenko, G.. “Approximation by superpositions of a sigmoidal function”, *Mathematics of Contro, Signals and Systems*, Vol.2, No. 4, pp. 303-314, 1989.
- [56] Hornik, K. “Approximation capabilities of multilayer feedforward networks”, *Neural Networks*, Vol. 4, No. 2, pp. 251-257, 1991.
- [57] Lee, H., Pham, P., Largman, Y. & Ng, A. Y., “Unsupervised feature learning for audio classification using convolutional deep belief networks”, *NIPS.*, pp. 1096-1104, 2009.
- [58] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G. et al, “Convolutional Neural Networks for Speech Recognition”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 10, pp. 1533–1545, 2014.
- [59] Cabrera-Vives, G., Reyes, I., Forster, F., Estévez, P. A. & Maureira, J. C., “Supernovae detection by using convolutional neural networks”, *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 251-258, 2016.
- [60] Kimura, A., Takahashi, I., Tanaka, M., Yasuda, N., Ueda, N. et al., “Single-epoch supernova classification with deep convolutional neural networks”, *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Atlanta, GA, USA, 2017, pp. 354-359,, 2017.
- [61] Sedaghat, N. & Mahabal, A., “Effective image differencing with convolutional neural networks for real-time transient hunting”, *Monthly Notices of the Royal Astronomical Society*, Vol. 476, pp. 5365–5376, 2018.
- [62] George, D. & Huerta, E. A., “Deep Learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data”, *Physics Letters B*, Vol. 778, pp. 64–70, 2018.
- [63] Shallue, C. J. & Vanderbug, A., “Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90”, *The Astronomical Journal*, Vol. 155, No. 2, p. 94, 2018.
- [64] Lipton, Z. C., Berkowitz, J. & Elkan, C., “A Critical Review of Recurrent Neural Networks for Sequence Learning”, *arXiv:1506.00019 [cs]*, *arXiv: 1506.00019*, 2015.
- [65] Sepp J., “Long short-term Memory”, *Neural Computation*, Vol 9. pp. 1735-80, 1997.
- [66] Gers, F. A., Schmidhuber, J. & Cummins, F., “Learning to forget: continual prediction with LSTM”, *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, Vol. 2, 1999.
- [67] Sak, H., Senior, A. W. & Beaufays, F., “Long short-term memory recurrent neural network architectures for large scale acoustic modeling”, *INTERSPEECH*, 2014.
- [68] Goldberg, Y., “Neural Network Methods for Natural Language Processing”, *Synthesis Lectures on Human Language Technologies*, Vol. 10, No. 1, pp. 1–309, 2017.
- [69] Sutskever, I., Vinyals, O. & Le, Q. V., “Sequence to Sequence Learning with Neural

- Networks”, *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014.
- [70] Graves, A., Mohamed, A. R. & Hinton, G., “Speech recognition with deep recurrent neural networks”, *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 38. 6645-6649, 2013.
- [71] Sainath, T. N., Vinyals, O., Senior, A. & Sak, H., “Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks”, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580-4584, 2015.
- [72] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S. et al., “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 4, pp. 677-691, 2017.
- [73] Zhao, R., Ali, H. & Smagt, P. V. D., “Two-stream RNN/CNN for action recognition in 3d videos”, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [74] Zhao, Y., Jin, X. & Hu, X., “Recurrent convolutional neural network for speech processing”, *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5300-5304, 2017.
- [75] Cho, K., van Merriënboer, B., Bahdanau, DZ., Bougares, F., Schwenk, H., Bengio, Y., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734, 2014.
- [76] Bonassi, F., Farina, M. & Scattolini, R., “On the stability properties of Gated Recurrent Units neural networks”, *Systems & Control Letters*, Vol. 157, pp. 0167-6911, 2020.
- [77] Xu, Y., Kong, Q., Huang, Q., Wang, W. & Plumbley, M. D., “Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging”, *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3461-3466, 2020.
- [78] Rana, R., “Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech”, *arXiv:1612.07778*, 2016.
- [79] Kitagawa, Y. & Komachi, M., “Long Short-Term Memory for Japanese Word Segmentation”, *arXiv:1709.08011*, 2017.
- [80] Li, Ch., Ma, X., Jiang, B., Li, X., Zhang, X., Liu, X., Cao, Y., Kannan, A. & Zhu, Zh., “Deep Speaker: an End-to-End Neural Speaker Embedding System”, *arXiv:1705.02304*, 2017.
- [81] Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R. & Hlozek, R., “RAPID: Early Classification of Explosive Transients Using Deep Learning” *Publications of the Astronomical Society of the Pacific*, Vol. 131, No. 1005, p. 118002, 2019.
- [82] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. Lang, K., “Phoneme Recognition Using Time-Delay Neural Networks”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 3, pp. 328. - 339, 1989.
- [83] Karaali, O., Corrigan, G. & Gerson, I., “Speech Synthesis with Neural Networks”, *World*

*Congress on Neural Networks (1996) 45-50, 1996.*

- [84] Karaali, O., Corrigan, G., Gerson, I. & Massey, N., “Text-To-Speech Conversion with Neural Networks: A Recurrent TDNN Approach”, *Proceedings of Eurospeech (1997) 561-564. Rhodes, Greece, 1997.*
- [85] Chen, L., Liang, Y., Shi, X., Zhou, Y. & Wu, Ch., “Crossed-Time Delay Neural Network for Speaker Recognition”, *MultiMedia Modeling. MMM. Lecture Notes in Computer Science, Vol 12572, 2020.*
- [86] deVriest, B. & Principe, J. C., “Short Term Memory Structures for Dynamic Neural Network”, *Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers, Pacific Grove, CA, USA, Vol.2, pp. 766-770, 1992.*
- [87] Principe, J., Kuo, J. & Celebi, S., “An analysis of the Gamma Memory in Dynamic Neural Networks”, *IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 331-337, 1994.*
- [88] Motter, M. & Principe, J., “A Gamma Memory Neural Network for system identification”, *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, Vol. 5, pp. 3232-3237, 1994.*
- [89] West, D. & Dellana, S., “An empirical analysis of neural network memory structures for basin water quality forecasting”, *International Journal of Forecasting, Vol. 27, pp. 777-803, 2011.*
- [90] Bozinovski, S. & Fulgosi, A., “The influence of pattern similarity and transfer learning upon the training of a base perceptron B2”, *Proceedings of Symposium Informatica 3-121-5, Bled, 1976.*
- [91] Yosinski, J., Clune, J., Bengio, Y. & Lipson, H., “How Transferable Are Features in Deep Neural Networks?”, *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014.*
- [92] Oquab, M., Bottou, L., Laptev, I. & Sivic, J., “Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks”, *2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1717-1724, 2014.*
- [93] Kingma, D. P., Ba, J., “Adam: A Method for Stochastic Optimization”, *International Conference on Learning Representations, 2014.*
- [94] Chambers, K., Magnier, E., Metcalfe, N., Flewelling, H., Huber, M. et al “The Pan-STARRS1 Surveys”, *Arxiv arXiv:1612.05560, 2016.*
- [95] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *J. Mach. Learn. Res., Vol. 15, No. 1, pp. 1929-158, 2014.*
- [96] Wright, E. “The Wide-field Infrared Survey Explorer (WISE): Mission Description and Initial On-orbit Performance”, *The Astronomical Journal, Vol. 140, No. 6, pp. 1868-1881, 2010.*
- [97] Cutri, R. M. “Explanatory Supplement to the AllWISE Data Release Products”, *Explanatory Supplement to the AllWISE Data Release Products, 2013.*
- [98] Efron, B. & Tibshirani, R. J. “An introduction to the bootstrap”, *Chapman & Hall/CRC*

*Monographs on Statistics & Applied Probability*, 1994.

- [99] Pitman, E. J. G. “Significance tests which may be applied to samples from any population”, *Royal Statistical Society Supplement*, 1937, 4: 119-30 and 225-32, 1937.
- [100] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., “Attention is all you need”, *Advances in Neural Information Processing Systems*, pp. 5998–6008., 2017.
- [101] Liaw, R., Liang, E., Nishihara, R., Moritza, P., Gonzalez, J. E. & Stoica, I., “Tune: A Research Platform for Distributed Model Selection and Training”, *arXiv preprint arXiv:1807.05118*, 2018.
- [102] Bergstra, J., Yamins, D., Cox & D. D., “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures”, *TProc. of the 30th International Conference on Machine Learning (ICML 2013)*, June 2013, pp. I-115 - I-23., 2013.
- [103] Sanchez-Sáez, P., Lira, H., Martí, L., Sanchez-Pi, N., Arredondo, J., Bauer, F.E. 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*, Vol. 162, No. 5, 206, 2021.
- [104] Perez-Carrasco, M., Cabrera-Vives, G., Hernández-García, L., Förster, F., et al., “Alert Classification for the ALerCE Broker System: The Anomaly Detector”, *The Astrophysical Journal*, Vol. 166, pp. 151, 2023.
- [105] Singh, R., Principe, J. C., “A Physics inspired Functional Operator for Model Uncertainty Quantification in the RKHS”, *arXiv preprint arXiv:2211.01999*, 2022.
- [106] MacKay, D. J., “A practical bayesian framework for backpropagation networks”, *Neural computation*, Vol. 4, No. 3, pp. 448–472, 1992.

# Anexos

## Anexo A. Metadatos

### A.1. Documentación de los metadatos

Tabla A.1: Descripción de los metadatos utilizados en esta tesis.

Metadato	Descripción
sgscore{1, 2, 3}	Puntaje de estrella/galaxia de la {primera, segunda, tercera} fuente más cercana del catálogo PanSTARRS1, $0 \leq \text{sgscore} \leq 1$ , donde un valor de 1 significa mayor probabilidad de ser una estrella, -999 cuando no hay fuente.
distpsnr {1, 2, 3}	Distancia de la {primera, segunda, tercera} fuente más cercana del catálogo PanSTARRS1, si existe una dentro de 30 arcosegundos, -999 si es que no hay fuente [arcosegundos].
isdiffpos	t (convertido a 1) si el candidato es de una substracción positiva (ciencia menos referencia); f (convertido a 0) si el candidato es una substracción negativa (referencia menos ciencia).
fwhm	Full Width Half Max (propiedad de distribución Gaussiana) asumiendo un centro gaussiano de la alerta del candidato en la imagen de ciencia de SExtractor [pixels].
magpsf	Magnitud del ajuste de PSF fotométrico de la alerta en la imagen de diferencia [mag].
sigmapsf	Incertidumbre de 1-sigma en magpsf [mag].
ra, dec	Coordenadas Right ascension y declinación del candidato; J2000 [deg].
diffmaglim	Mag. lím. de 5-sigma en imagen de diferencia, basado en ajuste fotométrico de PSF [mag].
classtar	Puntaje de clasificación estrella/galaxia de la alerta en imagen de diferencia, de SExtractor.
ndethist	Número de detecciones espacialmente coincidentes que caen dentro de 1.5 arcosegundos al ir hacia el comienzo del estudio astronómico; sólo son contadas las detecciones que caen sobre el mismo campo e identificador (ID) de canal de lectura donde el candidato de entrada es observado. Todas las detecciones crudas hasta un nivel fotométrico de señal/ruido 3 son incluidas.
ncovhist	Número de veces que la posición del candidato de entrada cae en cualquiera de las regiones y canal de lectura, en retrospectiva al comienzo del estudio astronómico.
chinr, sharpnr	DAOPhot parámetros chi, sharp de la fuente más cercana en la imagen de referencia catálogo-PSF dentro de 30 arcosegundos.
Coordenadas elípticas	latitud y longitud eclíptica calculadas desde los metadatos ra, dec del candidato [grados].
Coordenadas galácticas	latitud y longitud galáctica calculadas desde los metadatos ra, dec del candidato [grados].
no detecciones aprox.	ncovhist menos ndethist. Número aproximado de observaciones en la posición del candidato, con una señal más baja que la razón señal/ruido 3.

Tabla A.2: Descripción de los metadatos utilizados en este trabajo

Metadato	Descripción
isdiffpos	t (convertido a 1) si el candidato es de una substracción positiva (ciencia menos referencia); f (convertido a 0) si el candidato es una substracción negativa (referencia menos ciencia).
fwhm	Full Width Half Max (propiedad de distribución Gaussiana) asumiendo un centro gaussiano de la alerta del candidato en la imagen de ciencia de SExtractor [pixels].
magpsf	Magnitud del ajuste de PSF fotométrico de la alerta en la imagen de diferencia [mag].
sigmapsf	Incertidumbre de 1-sigma en magpsf [mag].
diffmaglim	Mag. lím. de 5-sigma en imagen de diferencia, basado en ajuste fotométrico de PSF [mag].
classtar	Puntaje de clasificación estrella/galaxia de la alerta en imagen de diferencia, de SExtractor.
ndethist	Número de detecciones espacialmente coincidentes que caen dentro de 1.5 arcossegundos al ir hacia el comienzo del estudio astronómico; sólo son contadas las detecciones que caen sobre el mismo campo e identificador (ID) de canal de lectura donde el candidato de entrada es observado. Todas las detecciones crudas hasta un nivel fotométrico de señal/ruido 3 son incluidas.
ncovhist	Número de veces que la posición del candidato de entrada cae en cualquiera de las regiones y canal de lectura, en retrospectiva al comienzo del estudio astronómico.
no detecciones aprox.	ncovhist menos ndethist. Número aproximado de observaciones en la posición del candidato, con una señal más baja que la razón señal/ruido 3.

## A.2. Rango de metadatos

Tabla A.3: Limite de valores de los metadatos. Los valores **max** y **min** significa que se mantiene el valor máximo y mínimo del conjunto de entrenamiento para ese metadato.

Metadato	[valor min, valor max]
sgscore 1	[-1, max]
distpsnr1	[-1, max]
sgscore2	[-1, max]
distpsnr2	[-1, max]
sgscore3	[-1, max]
distpsnr3	[-1, max]
ifwhm	[min, 10]
ndethist	[min, 20]
ncovhist	[min, 3000]
chinr	[-1, 15]
sharpnr	[-1, 1.5]
no-detecciones	[min, 2000]

### A.3. Distribución de ALL-WISE

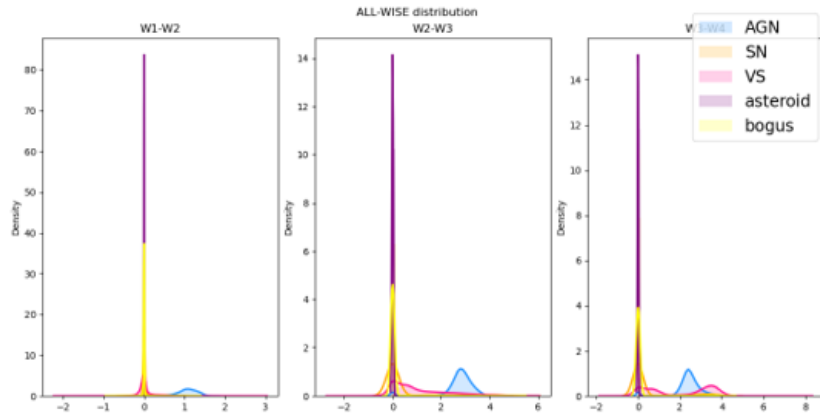


Figura A.1: Distribución de metadatos ALL-WISE.

### A.4. Comparación de la distribución de metadatos en forma temporal

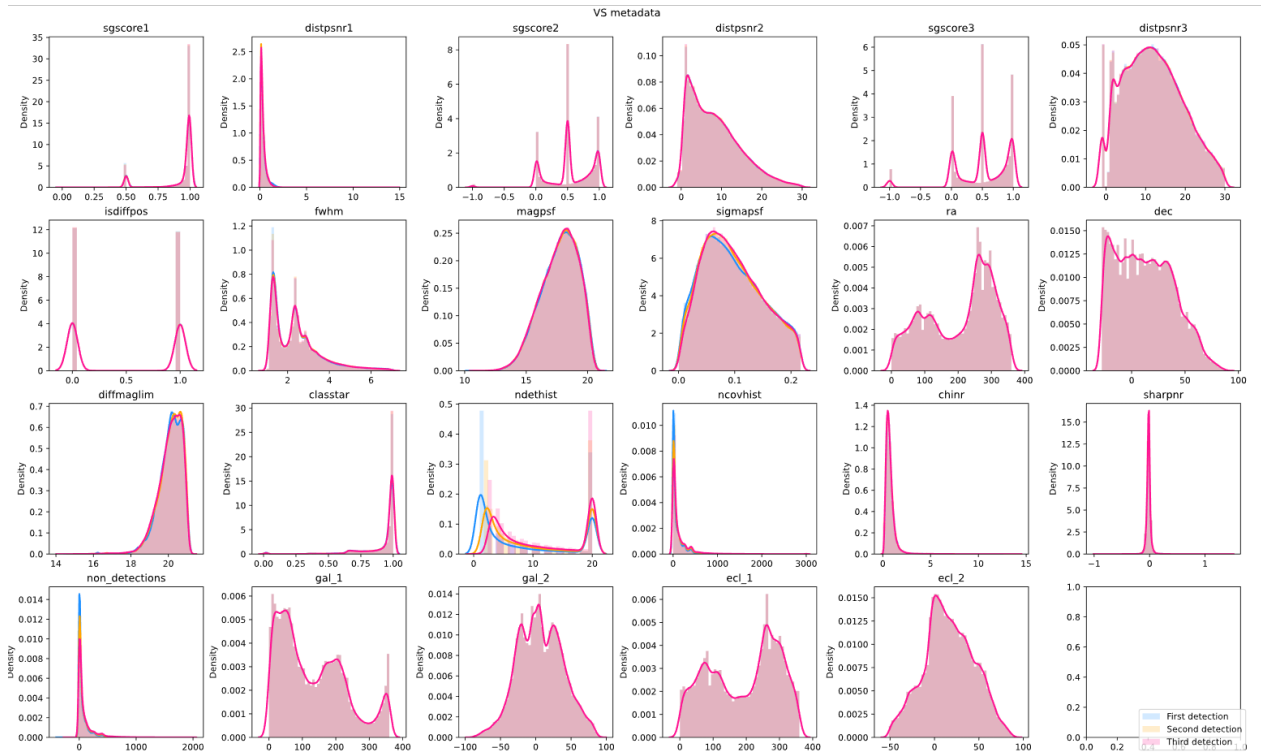


Figura A.2: Distribución de metadatos VS.



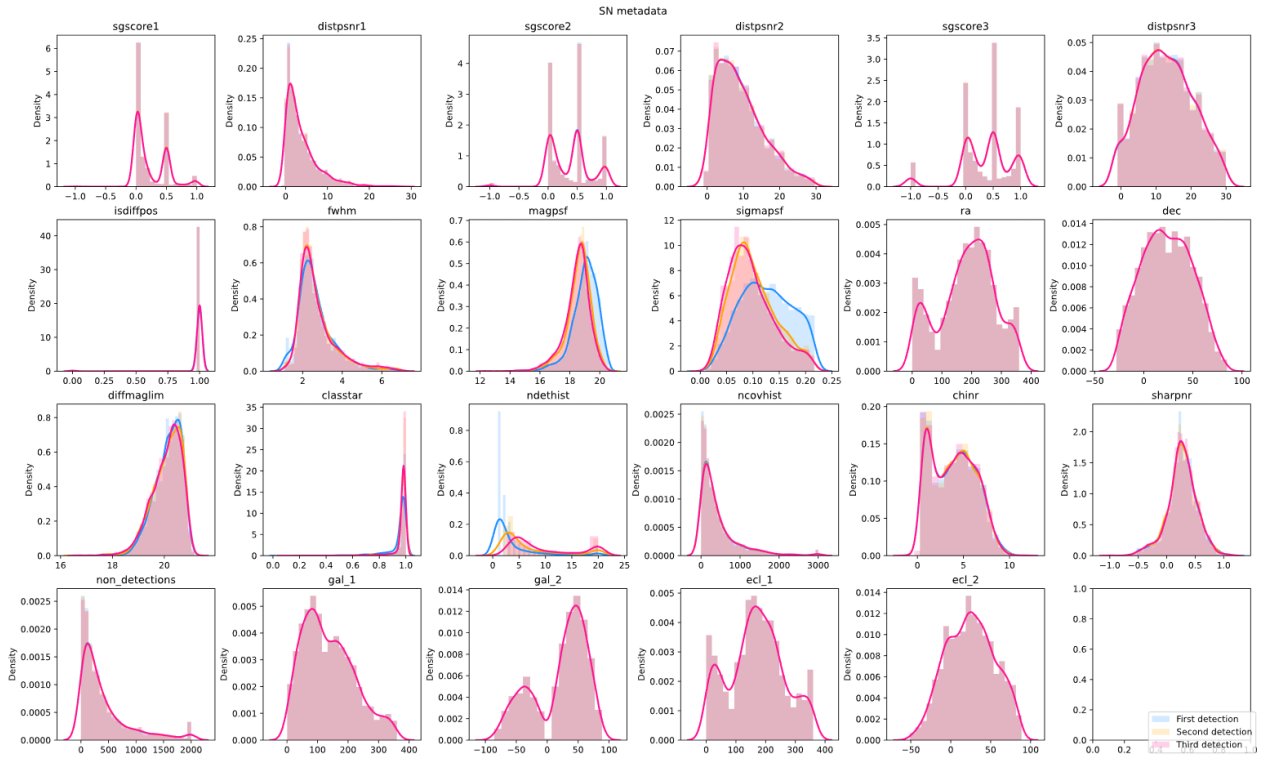


Figura A.3: Distribución de metadatos SN.

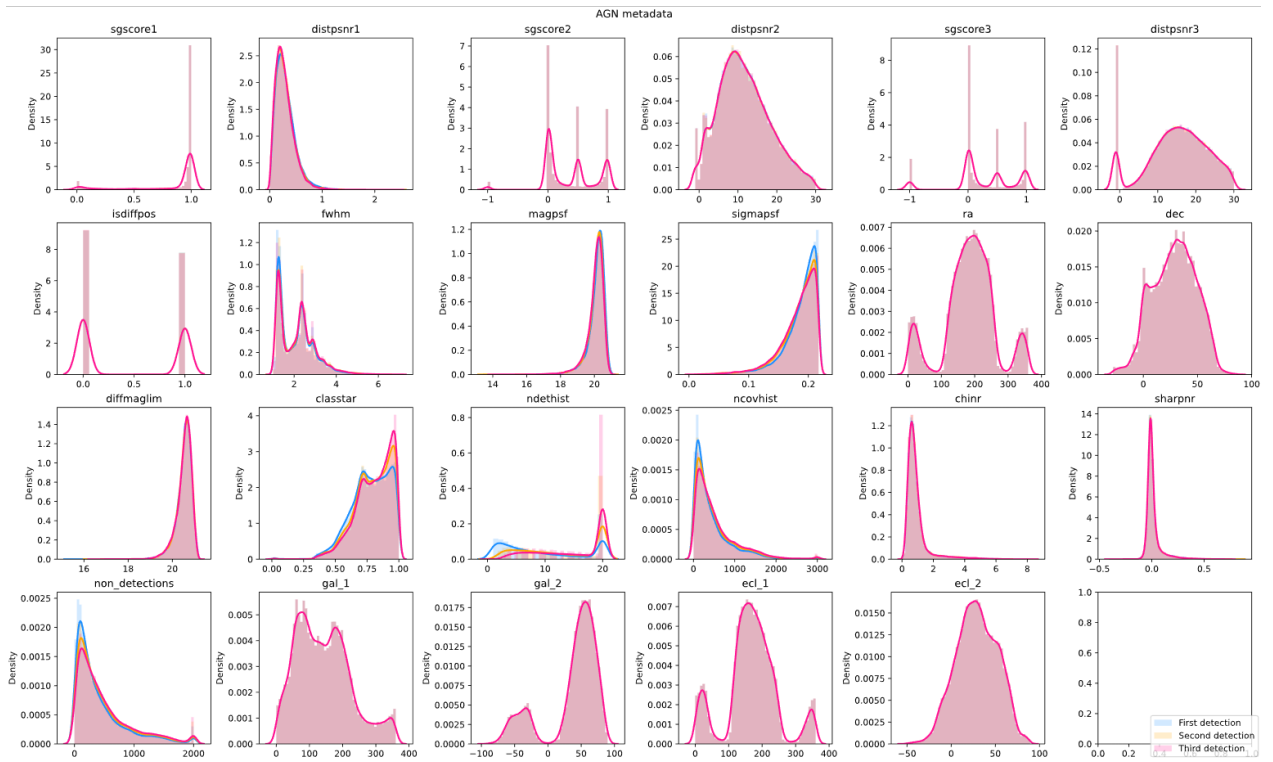
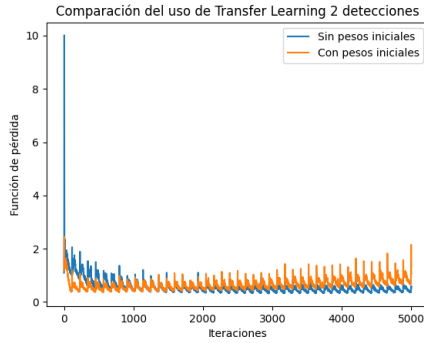


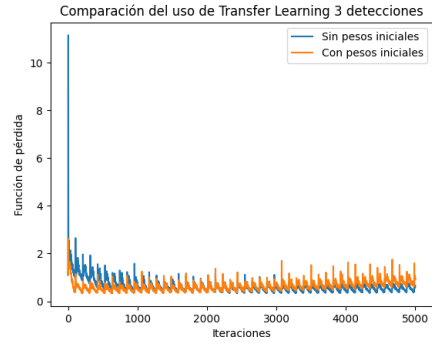
Figura A.4: Distribución de metadatos AGN.

# Anexo B. Resultados detallados del modelo Temporal Stamp Classifier

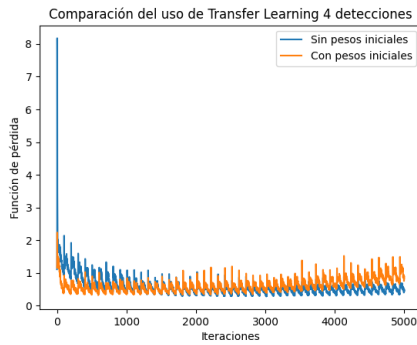
## B.1. Transfer Learning



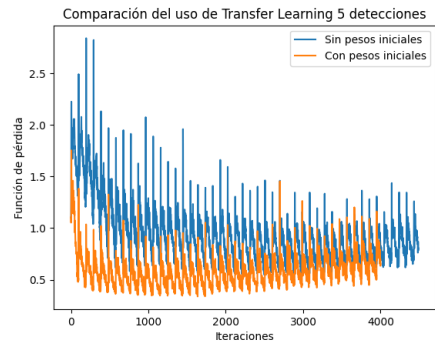
(a) 2 detecciones



(b) 3 detecciones



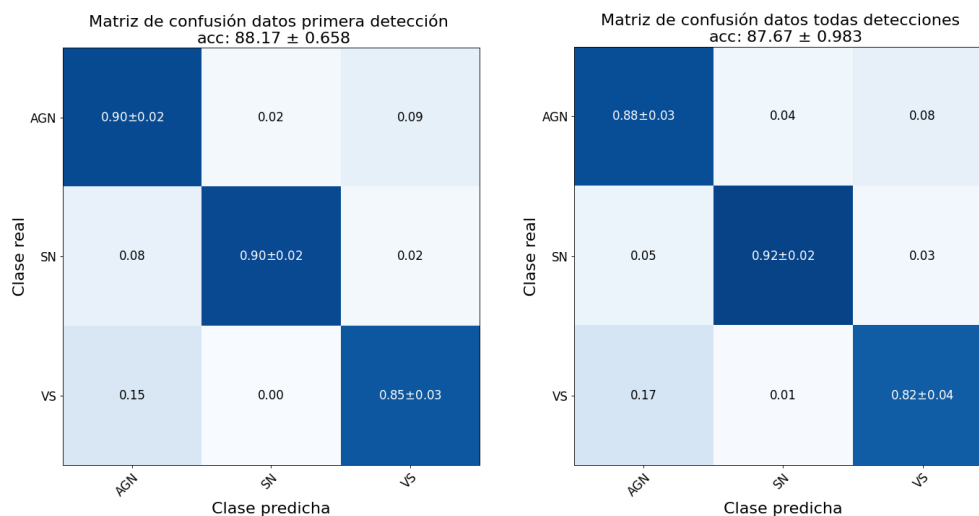
(c) 4 detecciones



(d) 5 detecciones

Figura B.1: Evolución de la función de pérdida cuando se utiliza los pesos de la primera detección como pesos iniciales para los filtros convolucionales y cuando se inicializan de forma aleatoria.

## B.2. Entrenamiento de capas convolucionales



(a) Datos de primera detección

(b) Con datos de detecciones

Figura B.2: Matriz de confusión modelo de solo imágenes para caso con solo datos de la primera detección y utilizando las imágenes de todas las detecciones.

## B.3. Probabilidades asignadas por los modelos actuales a la base de datos temporal Stamp Classifier

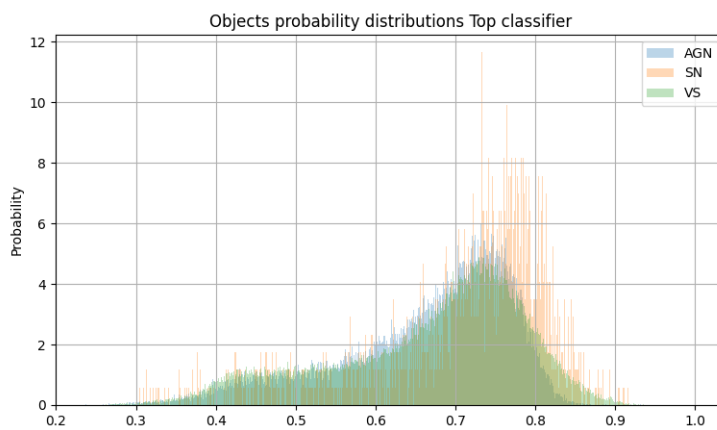


Figura B.3: Distribución de probabilidades Stamp Classifier.

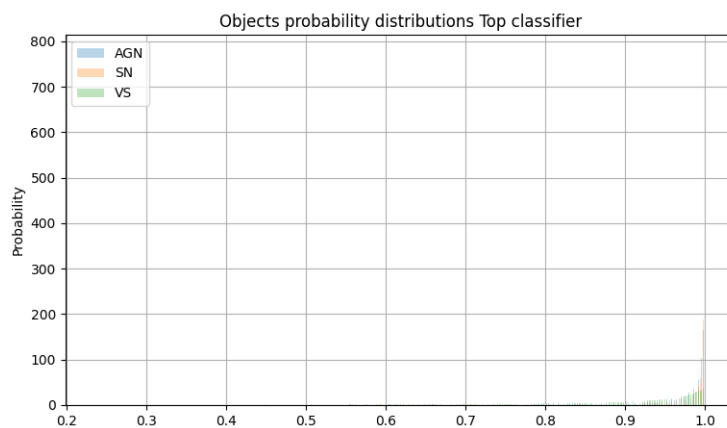


Figura B.4: Distribución de probabilidades Top Light Curve Classifier.

## B.4. Importancia de metadatos primera detección

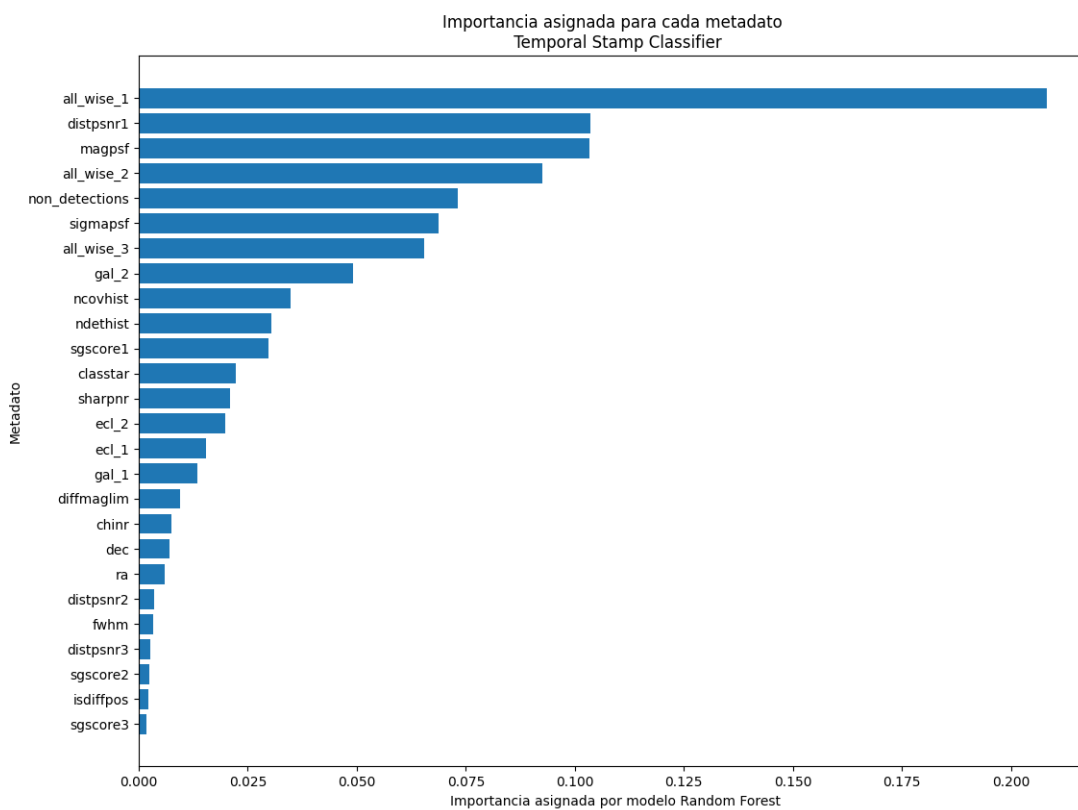


Figura B.5: Importancia de los metadatos en la primera detección modelo Temporal Stamp Classifier.

# Anexo C. Variaciones Stamp Classifier

## C.1. Validación de resultados con datos extra

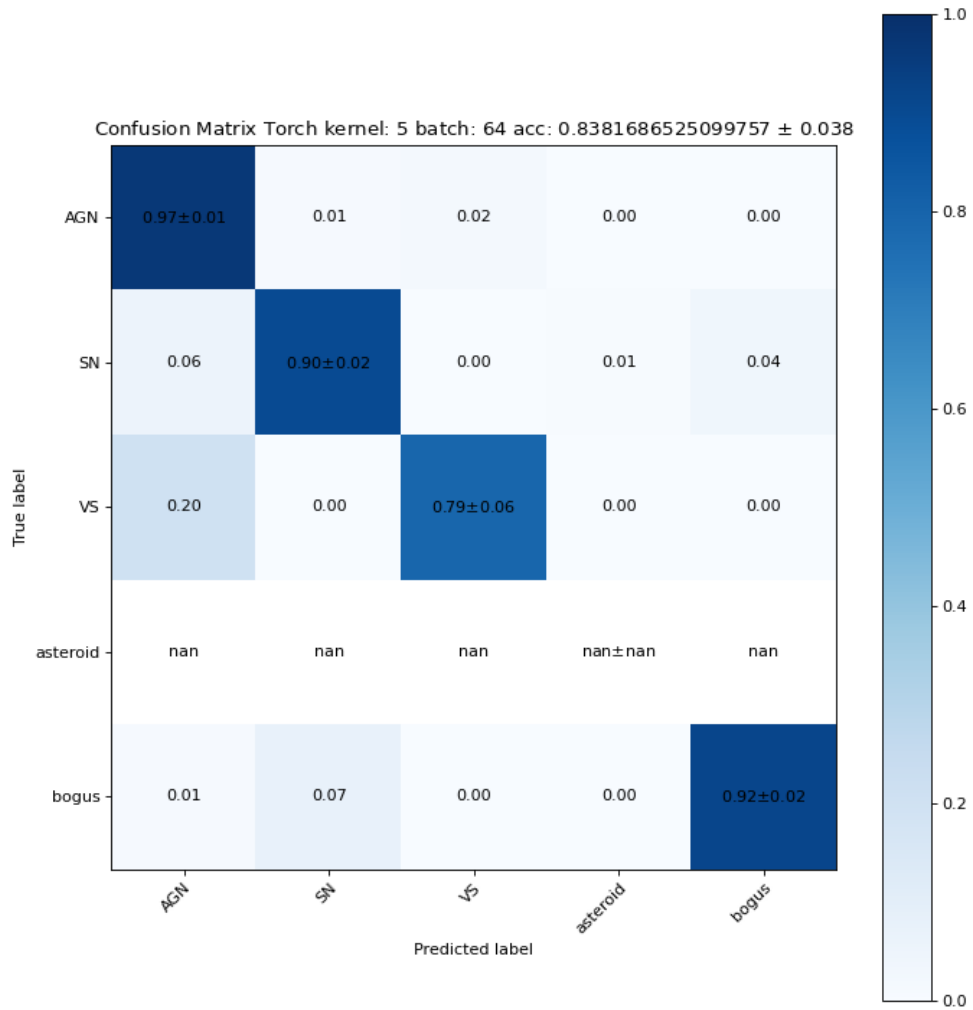


Figura C.1: Matriz de confusión mejor modelo Stamp Classifier con nuevos datos en su primera detección.

## C.2. Cruce con catálogo ALLWISE

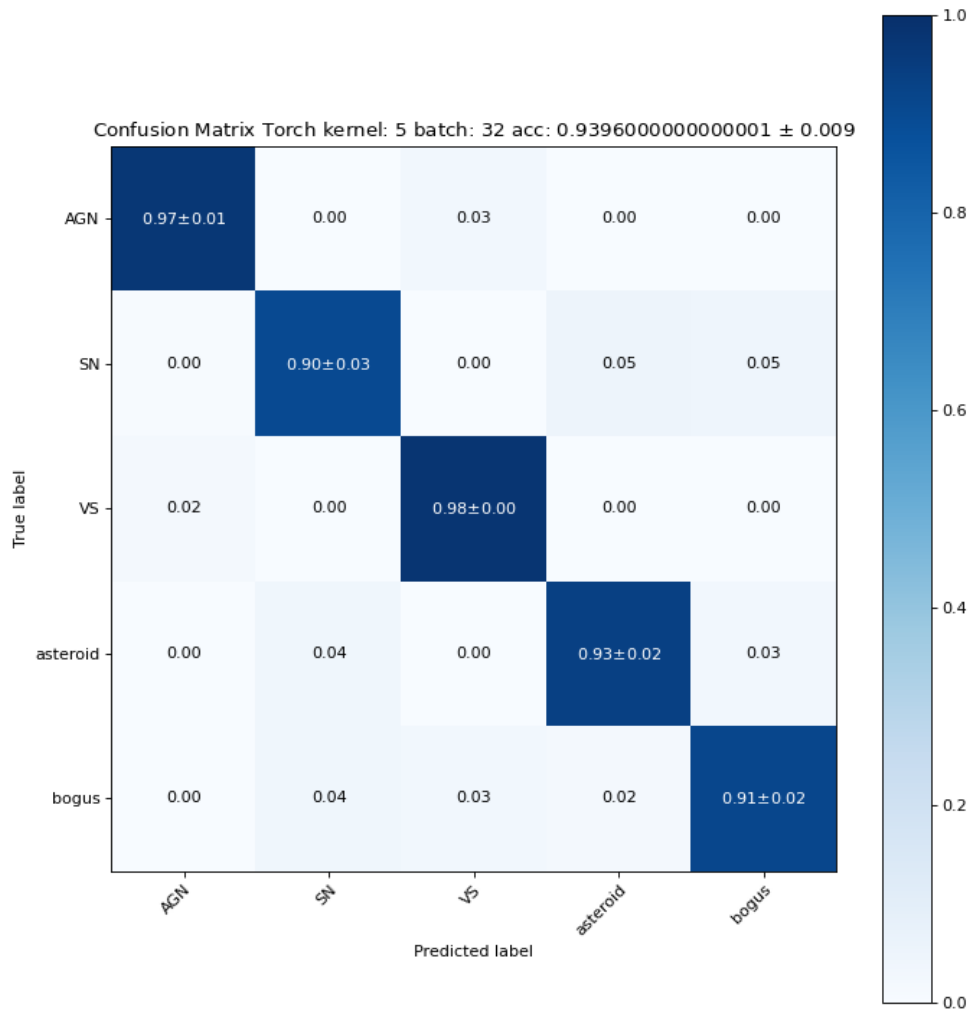


Figura C.2: Matriz de confusión modelo Stamp Classifier incluyendo ALLWISE sobre base de datos ALrCE-ZTFV5.

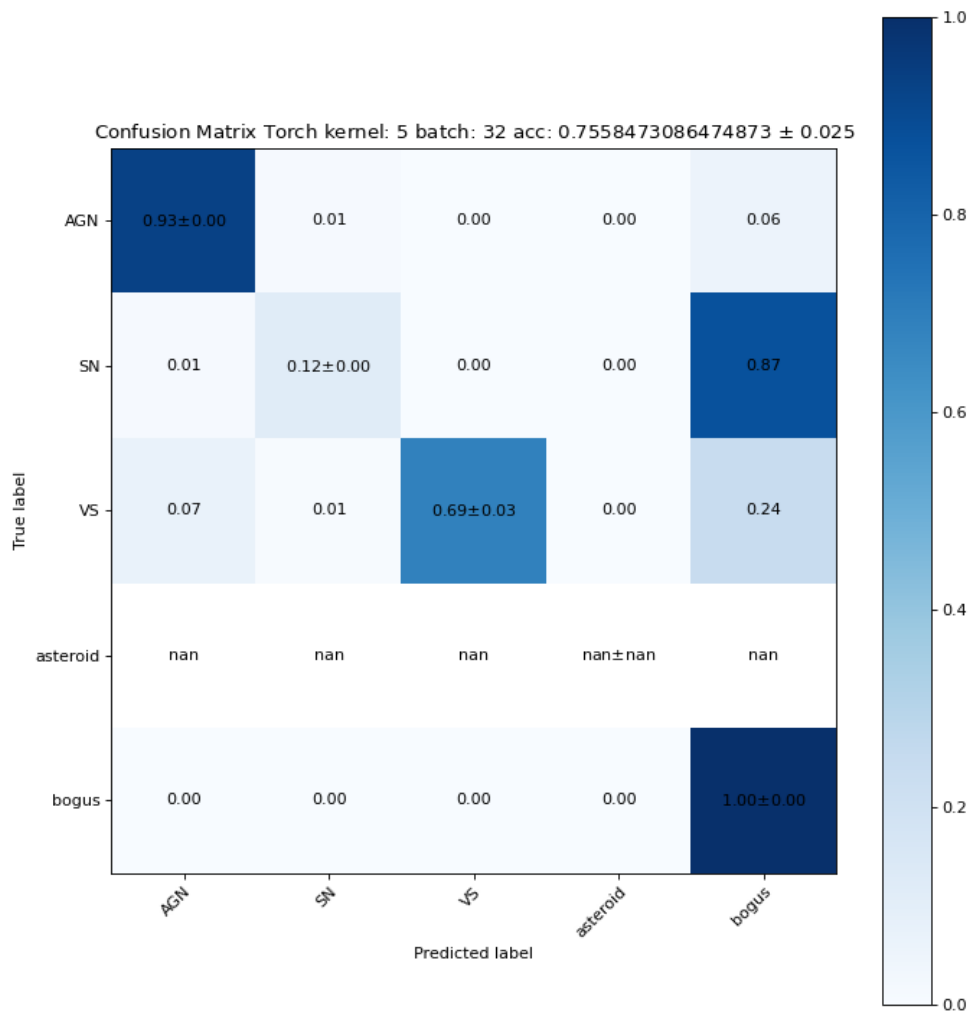


Figura C.3: Matriz de confusión modelo Stamp Classifier incluyendo ALL-WISE sobre nuevos datos en su primera detección.