



UNIVERSIDAD CEU SAN PABLO
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Sistemas de Información

TRABAJO DE FIN DE GRADO

Aplicación de Técnicas de Machine Learning a la Búsqueda de Materia Oscura con Datos del Satélite Fermi-LAT

*Application of Machine Learning Techniques to Dark Matter Search
using Fermi-LAT Satellite Data*

Alumna:

Marta Canino Romero

Tutora:

Dra. Viviana Gammaldi

Madrid, junio de 2025

Calificación del Trabajo Fin de Grado

Nombre del alumno

Título del proyecto

DATOS DEL TRIBUNAL

Presidente

Secretario

Vocal

Reunido este tribunal el ____ / ____/_____, acuerda otorgar al Trabajo Fin de Grado
presentado por D./Dña. _____ la calificación de _____

Resumen

Este Trabajo de Fin de Grado aborda la detección de anomalías en fuentes de rayos gamma no identificadas del catálogo Fourth Fermi Gamma-ray LAT catalog (4FGL) del telescopio espacial Fermi Large Area Telescope (Fermi-LAT), mediante técnicas de Machine Learning (ML) semi-supervisado. El objetivo es identificar fuentes que se desvén del comportamiento típico de las fuentes astrofísicas conocidas, detectando posibles candidatos a fenómenos no convencionales como señales indirectas de materia oscura.

Se ha desarrollado un modelo basado en One-Class Support Vector Machine (OCSVM), entrenado exclusivamente con datos de fuentes astrofísicas identificadas, capaz de clasificar como anómalas aquellas muestras que se alejen de la distribución normal. El modelo ha sido evaluado en configuraciones de 2 y 4 características (2F y 4F), aplicando optimización de hiperparámetros mediante grid search y validación cruzada.

Como análisis comparativo, se han replicado y ampliado los resultados de un estudio previo basado en Artificial Neural Networks (ANNs), estableciendo una línea base para evaluar el enfoque semi-supervisado. La comparación permite identificar coincidencias y discrepancias en la detección de fuentes no identificadas relevantes.

El desarrollo siguió una metodología ágil Scrum (Scrum), implementando un pipeline completo de ciencia de datos con preprocessamiento, transformaciones logarítmicas, escalado y evaluación sistemática. El sistema fue desarrollado en Python, utilizando scikit-learn, NumPy, Pandas y Matplotlib.

Los resultados demuestran la viabilidad del enfoque no supervisado para detectar anomalías en datos astrofísicos, proporcionando una herramienta complementaria a métodos supervisados. El trabajo contribuye a validar técnicas de ML en análisis de datos espaciales, demostrando el potencial interdisciplinar de la Ingeniería en Sistemas de Información (ISI).

Palabras clave: detección de anomalías, aprendizaje no supervisado, OCSVM, Fermi-LAT, materia oscura, rayos gamma, astrofísica computacional, ML

Abstract

This Bachelor's Thesis addresses the detection of anomalies in unidentified gamma-ray sources from the 4FGL catalog of the Fermi-LAT space telescope, through the application of semi-supervised ML techniques. The main objective is to identify sources that deviate significantly from the typical behavior of known astrophysical sources, in order to detect possible candidates for unconventional phenomena, such as indirect Dark Matter (DM) signals.

A model based on OCSVM has been developed, trained exclusively with data from identified astrophysical sources, capable of classifying as anomalous those samples that deviate from the learned normal distribution. The model has been implemented and evaluated in configurations of 2 and 4 features (2F and 4F), applying hyperparameter optimization techniques through grid search and repeated Cross Validation (CV).

As part of the comparative analysis, the results of a previous study based on ANNs have been replicated and extended, establishing a baseline to evaluate the performance of the semi-supervised approach. The comparison between both models allows identifying coincidences and discrepancies in the detection of potentially relevant unidentified sources.

The project development has followed an agile methodology based on Scrum, implementing a complete data science pipeline that includes data preprocessing, logarithmic transformations, escalado, and systematic evaluation of results. The system has been developed entirely in Python, using specialized libraries such as scikit-learn, NumPy, Pandas, and Matplotlib.

The results obtained demonstrate the feasibility of the unsupervised approach for anomaly detection in astrophysical data, providing a complementary tool to traditional supervised methods. The work contributes to the validation of ML techniques applied to space data analysis, demonstrating the interdisciplinary potential of ISI in the field of scientific research.

Keywords: anomaly detection, unsupervised learning, OCSVM, Fermi-LAT, DM, gamma rays, computational astrophysics, ML

Índice general

Lista de Acrónimos	vi
Glosario de Términos	viii
1. Introducción	1
1.1. Contexto del problema	2
1.2. Enfoque metodológico	2
1.3. Objetivos del proyecto	3
1.4. Estructura del documento	3
1.5. Alcance y limitaciones	4
2. Marco teórico	5
2.1. Contexto del problema y fuente de datos	5
2.2. Fundamentos de ML	7
2.3. Detección de anomalías: marco teórico	7
2.4. OCSVM	8
2.5. Metodología experimental	10
2.6. Estado del arte y contribuciones	12
2.7. Relevancia para la Ingeniería en Sistemas de Información	13
3. Gestión del Proyecto	14
3.1. Metodología ágil adoptada	14
3.2. Organización de roles	14
3.3. Estructura del proyecto	15
3.4. Definición de requisitos del sistema	16
3.5. Planificación temporal	17
3.6. Implementación técnica	19
4. Modelo de referencia (ANN)	21
4.1. Arquitectura y configuración general	21
4.2. Entrenamiento y aplicación del modelo ANN 2F	22
4.3. Entrenamiento y aplicación del modelo ANN 4F	23
4.4. Análisis comparativo y línea base	24

5. Desarrollo experimental (OCSVM)	25
5.1. Datos utilizados y preprocesamiento	26
5.2. Desarrollo del modelo OCSVM	28
5.3. Aplicación a fuentes no identificadas	33
5.4. Análisis comparativo ANN vs OCSVM	40
6. Otros experimentos adicionales	42
6.1. Implementación de OCSVM con datos DR4	42
6.2. Aplicación a fuentes no asociadas	43
6.3. Conclusiones del experimento DR4	44
7. Conclusiones y líneas futuras	45
7.1. Principales contribuciones y resultados del sistema	45
7.2. Limitaciones técnicas y consideraciones del sistema	46
7.3. Líneas futuras de desarrollo del sistema	46
Apéndices	48
I. Entregables	48
II. Planificación detallada del proyecto	50
III. Replicación del Estudio con Redes Neuronales	54
IV. Análisis Exploratorio de Datos	56
V. Transformación del Dataset UNIDs	59
VI. Análisis estadístico detallado de modelos OCSVM	63
VII. Modelo OCSVM 2F (código)	66
VIII. Modelo OCSVM 4F (código)	69
IX. Análisis de Overfitting y Underfitting en OneClassSVM 2F	71
X. Visualizaciones Multidimensionales del Modelo OneClassSVM 4F	75
XI. Análisis de Overfitting y Underfitting en OneClassSVM 4F	77
XII. Predicción UNID con Modelo OCSVM 2F (código)	80
XIII. Predicción UNID con Modelo OCSVM 4F (código)	83
XIV. Visualizaciones Multidim. de Predicciones con OCSVM 4F	85
XV. Análisis detallado ANN vs OCSVM	87
XVI. Figuras del experimento OCSVM con 3F del DR4	91
Bibliografía	93

Listado de Acrónimos

4FGL Fourth Fermi Gamma-ray LAT catalog - Cuarto catálogo del telescopio Fermi.

Adam Adaptive Moment Estimation - Algoritmo de optimización.

AGN Active Galactic Nucleus - Núcleo galáctico activo.

ANN Artificial Neural Network - Red neuronal artificial que simula el funcionamiento de neuronas biológicas.

API Application Programming Interface - Interfaz de programación de aplicaciones.

ASTRO Astrophysical source - Fuente de origen astrofísico conocido.

AUC Area Under the Curve - Área bajo la curva.

CSV Comma-Separated Values - Formato de archivo de valores separados por comas.

CV Cross Validation - Técnica de validación cruzada.

DL Deep Learning - Subconjunto de ML que utiliza redes neuronales profundas.

DM Dark Matter - Materia hipotética que no emite radiación electromagnética pero tiene efectos gravitacionales.

DR3 Data Release 3 - Tercera versión de publicación de datos del catálogo Fermi.

DR4 Data Release 4 - Cuarta versión de publicación de datos del catálogo Fermi.

ETL Extract, Transform, Load - Proceso de extracción, transformación y carga de datos.

Fermi-LAT Fermi Large Area Telescope - Telescopio espacial para detección de rayos gamma.

Git Sistema de control de versiones distribuido.

IDE Integrated Development Environment - Entorno de desarrollo integrado.

ISI Ingeniería en Sistemas de Información - Carrera universitaria.

JSON JavaScript Object Notation - Formato de intercambio de datos.

LAT Large Area Telescope - Telescopio de área grande utilizado para detectar rayos gamma.

Lista de Acrónimos

ML Machine Learning - Conjunto de técnicas computacionales que permiten a las máquinas aprender patrones a partir de datos.

MLP Multi-Layer Perceptron - Perceptrón multicapa.

OCSVM One-Class Support Vector Machine - Algoritmo de ML para detección de anomalías.

RBFF Radial Basis Function - Función de base radial usada en kernels.

ReLU Rectified Linear Unit - Función de activación.

Scrum Metodología ágil de gestión de proyectos que enfatiza la colaboración, flexibilidad y entrega iterativa.

Sprint Período de trabajo de duración fija en metodologías ágiles como Scrum.

TFG Trabajo de Fin de Grado - Proyecto académico final de carrera universitaria.

TS Test Statistic - Estadístico de prueba.

UNAS Unassociated source - Fuente no asociada con objetos conocidos.

UNID Unidentified source - Fuente no identificada en catálogos astronómicos.

WIMP Weakly Interacting Massive Particle - Candidato teórico para partícula de materia oscura.

XP eXtreme Programming - Metodología ágil con énfasis en prácticas técnicas.

Glosario de Términos

anomalía Observación que se desvía significativamente de otras observaciones, sugiriendo que fue generada por un mecanismo diferente. Sinónimo de outlier en el contexto de ML.

característica Variable individual medible de un fenómeno que está siendo observado, también conocida como feature o atributo. En astronomía incluye propiedades como flujo, índice espectral, variabilidad, etc..

clasificación Tarea de ML que consiste en asignar etiquetas o categorías a instancias basándose en sus características. En este proyecto se usa para distinguir entre fuentes Astrophysical source (ASTRO) y potenciales candidatos a materia oscura.

clustering Técnica de aprendizaje no supervisado que agrupa datos similares en conjuntos o clusters sin conocimiento previo de las categorías.

dataset Conjunto estructurado de datos utilizados para entrenamiento, validación o prueba de algoritmos de ML. En este proyecto incluye datos del catálogo 4FGL.

escalado Proceso de normalización de características para que tengan rangos similares, mejorando el rendimiento de algoritmos de ML como y OCSVM.

framework Estructura conceptual y tecnológica de soporte definida que sirve de base para la organización y desarrollo de software. Ejemplos incluyen scikit-learn para ML y TensorFlow para Deep Learning (DL).

grid search Técnica de optimización que evalúa sistemáticamente todas las combinaciones posibles de hiperparámetros en un espacio discreto definido para encontrar la configuración óptima.

hiperparámetro Parámetro de configuración de un algoritmo de ML que debe establecerse antes del entrenamiento y que controla el proceso de aprendizaje. Su optimización se realiza mediante técnicas como grid search.

inlier Punto de datos que se encuentra dentro del patrón normal esperado del dataset, opuesto a outlier. Los algoritmos como OCSVM aprenden a distinguir entre inliers y outliers.

kernel Función matemática que permite transformar datos a un espacio de mayor dimensión para facilitar la separación de clases en algoritmos de ML. Tipos comunes incluyen Radial Basis Function (RBF), lineal y polinomial.

materia oscura Forma hipotética de materia que no emite, absorbe ni refleja radiación electromagnética, pero cuya existencia se infiere por sus efectos gravitacionales. Se estima que constituye aproximadamente el 27% del universo. Los candidatos teóricos incluyen Weakly Interacting Massive Particles (WIMPs) y otras partículas exóticas.

metodología ágil Enfoque de desarrollo de software que enfatiza la colaboración, la adaptabilidad y la entrega iterativa de valor al cliente. Incluye frameworks como Scrum, Kanban y eXtreme Programming (XP).

outlier Punto de datos que se desvía significativamente del patrón general del conjunto de datos, también conocido como valor atípico o anomalía. Su detección es fundamental en técnicas de ML como OCSVM.

overfitting Fenómeno que ocurre cuando un modelo de ML se ajusta excesivamente a los datos de entrenamiento, perdiendo capacidad de generalización a datos nuevos. Se previene mediante validación cruzada y regularización. Es lo opuesto al underfitting.

pipeline Secuencia automatizada de pasos de procesamiento de datos que incluye preprocesamiento, transformación, entrenamiento y evaluación de modelos de ML. Permite crear flujos de trabajo reproducibles y eficientes.

Product Owner Rol en Scrum responsable de definir y priorizar los requisitos del producto, gestionar el backlog y maximizar el valor del trabajo realizado por el equipo de desarrollo.

púlsar Estrella de neutrones altamente magnetizada que rota rápidamente y emite haces de radiación electromagnética desde sus polos magnéticos. Son fuentes importantes de rayos gamma y pueden ser detectadas por el telescopio Fermi-LAT.

rayos gamma Radiación electromagnética de alta energía emitida por procesos nucleares y astrofísicos, con energías superiores a 100 keV. Representan la forma más energética de radiación electromagnética y pueden originarse en púlsars, explosiones de supernovas, y aniquilación de materia oscura.

regresión Tarea de ML que predice valores numéricos continuos basándose en las características de entrada, complementaria a la clasificación.

Scrum Master Facilitador del proceso Scrum que ayuda al equipo a seguir las prácticas ágiles, elimina impedimentos y protege al equipo de distracciones externas.

supernova Explosión estelar que marca el final violento de la vida de una estrella masiva, liberando enormes cantidades de energía en forma de radiación electromagnética, incluyendo rayos gamma.

underfitting Fenómeno que ocurre cuando un modelo de ML es demasiado simple para capturar los patrones subyacentes en los datos, resultando en un rendimiento pobre tanto en entrenamiento como en validación. Es lo opuesto al overfitting.

validación cruzada Técnica de evaluación que divide los datos en múltiples subconjuntos para entrenar y validar el modelo repetidamente, proporcionando una estimación más robusta del rendimiento del algoritmo de ML.

Capítulo 1

Introducción

Este Trabajo de Fin de Grado aborda la aplicación de técnicas avanzadas de Aprendizaje Automático (ML) a uno de los problemas más desafiantes de la física contemporánea: la búsqueda indirecta de materia oscura. El proyecto demuestra el impacto de la ISI en investigación científica interdisciplinaria, utilizando datos reales del telescopio espacial Fermi-LAT de la NASA.

La materia oscura constituye aproximadamente el 27% del contenido del universo, superando significativamente a la materia ordinaria visible ($\approx 5\%$), como ilustra la figura 1.1 (Planck Collaboration, 2013). Su detección indirecta mediante observaciones de rayos gammas representa una oportunidad única para combinar astrofísica experimental con técnicas computacionales avanzadas.

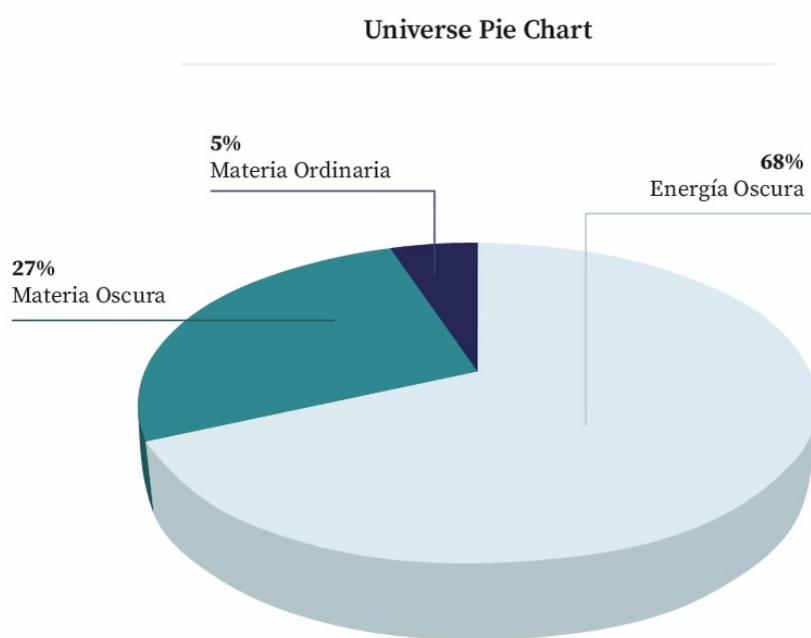


Figura 1.1. Composición del universo según mediciones del satélite Planck. La materia oscura (27%) supera considerablemente a la materia ordinaria (5%). Fuente: Planck Collaboration et al., *A&A*, 641, A6, 2020.

1.1. Contexto del problema

El catálogo 4FGL de Fermi-LAT constituye uno de los datasets astronómicos más completos disponibles, conteniendo información detallada de más de 5000 fuentes gamma detectadas - objetos o fenómenos celestes que emiten rayos gammas, que son la forma más energética de radiación electromagnética - (Abdollahi et al., 2020). Sin embargo, aproximadamente un tercio de estas fuentes permanece sin identificar (Unidentified sources (UNIDs)), representando tanto un desafío científico como una oportunidad computacional.

Desde una perspectiva de ingeniería informática, este escenario plantea un problema clásico de **detección de anomalías en datasets desbalanceados**: identificar patrones inusuales (posibles señales de materia oscura) dentro de una población mayoritariamente conocida (ASTROs).

Los enfoques tradicionales han utilizado principalmente métodos supervisados basados en ANNs (Gammaldi et al., 2023; Saz Parkinson et al., 2017). Sin embargo, estos requieren ejemplos etiquetados de todas las clases de interés, limitación crítica cuando se buscan fenómenos completamente nuevos o estadísticamente raros.

1.2. Enfoque metodológico

Este proyecto supera las limitaciones mencionadas mediante técnicas de **detección de anomalías semi-supervisadas**, específicamente OCSVM. Esta elección metodológica se justifica por:

- **Disponibilidad de datos:** Abundantes fuentes astrofísicas identificadas (clase normal) vs. ausencia de ejemplos confirmados de materia oscura (anomalías)
- **Naturaleza del problema:** Búsqueda de patrones raros en un espacio de características multidimensional
- **Robustez:** Capacidad de detectar anomalías sin conocimiento previo de su naturaleza específica

OCSVM aprende exclusivamente del comportamiento "normal" (fuentes identificadas como púlsares, Active Galactic Nucleus (AGNs), etc.) para después identificar desviaciones significativas que podrían corresponder a fenómenos desconocidos, incluyendo posibles señales de materia oscura.

1.2.1. Tecnologías y herramientas

El desarrollo del proyecto se ha realizado íntegramente en **Python 3.9.2**, utilizando principalmente la biblioteca **Scikit-learn** para la implementación de algoritmos de ML. Esta elección tecnológica se fundamenta en:

- **Ecosistema científico maduro:** Python ofrece bibliotecas especializadas como NumPy, Pandas y Matplotlib para procesamiento, análisis y visualización de datos
- **Implementación robusta de OCSVM:** Scikit-learn proporciona una implementación optimizada y bien documentada del algoritmo OneClassSVM

- **Reproducibilidad:** Entorno estándar ampliamente utilizado en la comunidad científica
- **Escalabilidad:** Capacidad para manejar datasets de gran tamaño como el catálogo 4FGL

El desarrollo se ha estructurado mediante **Jupyter Notebooks**, facilitando la documentación interactiva del código, la visualización de resultados intermedios y la reproducibilidad completa de los experimentos realizados.

1.3. Objetivos del proyecto

1.3.1. Objetivo principal

Desarrollar y validar un framework computacional basado en OCSVM para detectar fuentes anómalas en el catálogo Fermi-LAT, con el propósito de identificar candidatos potenciales a señales de materia oscura.

1.3.2. Objetivos específicos

- **Diseño del pipeline Extract, Transform, Load (ETL):** Implementar un sistema robusto de preprocesamiento para el catálogo 4FGL, incluyendo limpieza, normalización y selección de características espectrales
- **Desarrollo del modelo OCSVM:** Construir y optimizar modelos con diferentes combinaciones de características espectrales (configuraciones 2F y 4F para un set de datos 4FGL-Data Release 3 (DR3) y 3F para un set más nuevo 4FGL-Data Release 4 (DR4))
- **Optimización de hiperparámetros:** Implementar búsqueda sistemática en rejilla para los parámetros críticos ν y γ , con validación cruzada
- **Benchmarking comparativo:** Evaluar el rendimiento contra modelos de redes neuronales (ANN) de referencia establecidos en la literatura ([Gammaldi et al., 2023](#))
- **Análisis de candidatos:** Identificar y caracterizar fuentes clasificadas como anómalas, evaluando su potencial astrofísico
- **Validación metodológica:** Demostrar la robustez y complementariedad del enfoque semi-supervisado mediante métricas cuantitativas

1.4. Estructura del documento

El documento está organizado de manera que cada capítulo construye sobre los anteriores, manteniendo tanto rigor técnico como claridad expositiva:

- **Capítulo 1 (este capítulo):** Contextualización, objetivos y estructura general
- **Capítulo 2:** Fundamentos teóricos de ML, detección de anomalías y OCSVM. Revisión del estado del arte en búsqueda de materia oscura
- **Capítulo 3:** metodología ágil aplicada, planificación temporal y gestión de recursos
- **Capítulo 4:** Implementación y validación de modelos ANN de referencia basados en literatura previa

- **Capítulo 5:** Desarrollo experimental completo del enfoque OCSVM, desde preprocesamiento hasta análisis de resultados
- **Capítulo 6:** Modelo experimental complementario utilizando datos actualizados del catálogo 4FGL-DR4 con configuración de 3 características
- **Capítulo 7:** Síntesis de resultados, limitaciones identificadas y direcciones futuras
- **Apéndices:** Recursos complementarios, incluyendo repositorio de código y datasets procesados

1.5. Alcance y limitaciones

Es importante establecer que este proyecto no pretende resolver definitivamente el enigma de la materia oscura. Su propósito es desarrollar y validar una metodología computacional robusta que:

- Demuestre la viabilidad de enfoques semi-supervisados en astrofísica experimental
- Proporcione herramientas complementarias a los métodos supervisados existentes
- Establezca un framework replicable para futuros estudios interdisciplinarios
- Ilustre el impacto de la ISI en investigación científica avanzada

Los resultados se interpretan como candidatos potenciales que requieren validación adicional mediante análisis teóricos complementarios, manteniéndose dentro del rigor científico apropiado para este tipo de investigación exploratoria.

Capítulo 2

Marco teórico

2.1. Contexto del problema y fuente de datos

El telescopio espacial Fermi-LAT, operativo desde 2008, ha permitido avances significativos en el estudio del universo en la banda de rayos gammas de alta energía. A lo largo de más de una década de observaciones, el telescopio Fermi-LAT ha recopilado datos que han dado lugar a catálogos públicos progresivamente más completos, siendo uno de los más recientes el 4FGL, que contiene información detallada de más de **5000 fuentes gamma detectadas** (Abdollahi et al., 2020).

Como se observa en la figura 2.1, estas fuentes no se distribuyen aleatoriamente en el cielo: muchas se concentran en el plano galáctico correspondiente a la Vía Láctea, mientras que otras aparecen más dispersas. La mayoría han podido ser asociadas a objetos astrofísicos conocidos (púlsars, AGNs, remanentes de supernova) gracias a observaciones complementarias. Sin embargo, **un porcentaje significativo** (cerca de un tercio) **permanece sin asociación clara**, como se aprecia en la figura 2.2.

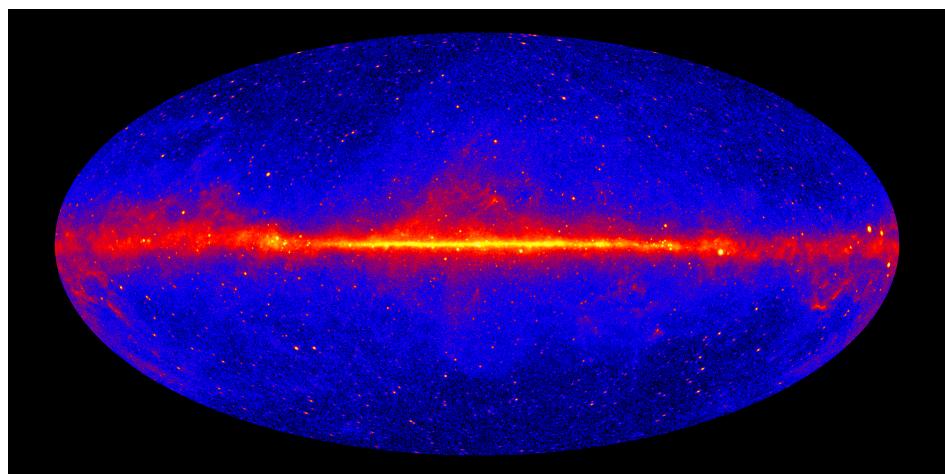


Figura 2.1. Mapa del cielo completo de rayos gammas registrado por Fermi-LAT. Se observa una clara concentración de fuentes en el plano galáctico donde se ubica la Vía Láctea.

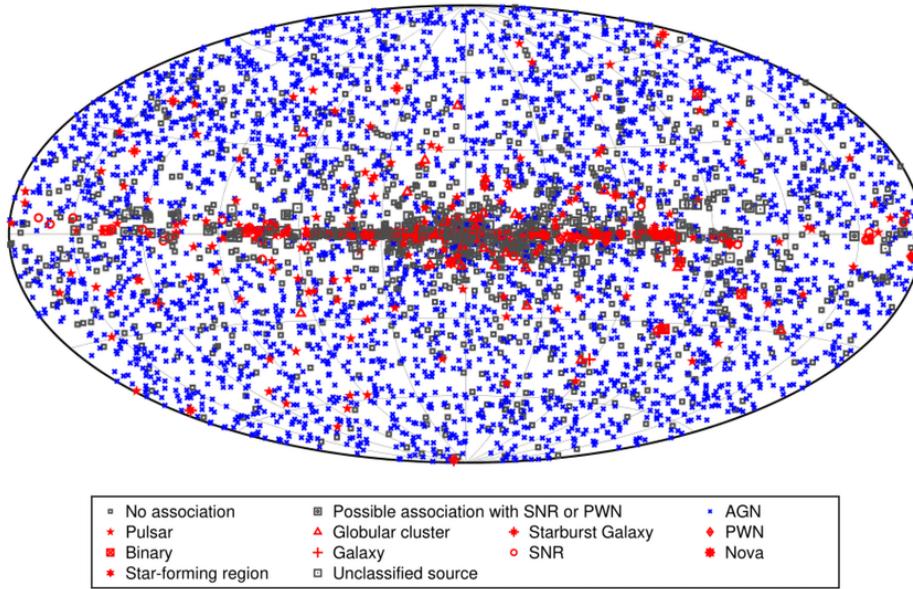


Figura 2.2. Distribución de las aproximadamente 5000 fuentes del catálogo 4FGL en coordenadas galácticas. Las fuentes asociadas con clases conocidas aparecen en color, mientras que las no identificadas (UNIDs) se muestran en gris.

2.1.1. El catálogo 4FGL como dataset

Desde una perspectiva de ingeniería informática, el 4FGL constituye un dataset estructurado que proporciona para cada fuente detectada:

- **Atributos posicionales:** Coordenadas celestes con incertidumbres asociadas
- **Parámetros espectrales:** Energía de pico (E_{peak}), curvatura espectral (β), índices espectrales y sus incertidumbres relativas
- **Métricas de calidad:** Test Statistic (TS) que cuantifica la significancia estadística de la detección
- **Indicadores temporales:** Variables de variabilidad en diferentes escalas
- **Metadatos de asociación:** Enlaces a contrapartes identificadas cuando están disponibles

Para este proyecto, siguiendo precedentes establecidos por Gammaldi et al. (2023), se han seleccionado como características principales: la energía pico (E_{peak}), la curvatura espectral (β), su incertidumbre relativa (β_{rel}), y la significancia estadística (σ_d). Esta selección se basa en criterios tanto computacionales (dimensionalidad manejable) como físicos (relevancia para la caracterización espectral).

2.1.2. Motivación científica: búsqueda de materia oscura

La materia oscura, cuya existencia se deduce de observaciones como las curvas de rotación galácticas y efectos de lentes gravitacionales (Bertone and Hooper, 2018), representa uno de los mayores enigmas de la física contemporánea. Entre las hipótesis más aceptadas destaca el modelo de *WIMPs*, cuya aniquilación podría generar rayos gamma con distribuciones energéticas características detectables por Fermi-LAT (Cirelli et al., 2011).

Este contexto astrofísico plantea un **problema computacional específico**: identificar patrones anómalos en un dataset donde las potenciales señales de interés (materia

oscura) son estadísticamente raras y no están etiquetadas, mientras que las fuentes conocidas (AGNs, púlsars) constituyen la clase mayoritaria bien caracterizada.

2.2. Fundamentos de ML

El ML es una disciplina que desarrolla algoritmos capaces de **extraer patrones automáticamente de los datos** sin programación explícita para cada tarea específica (Samuel, 1959). Según Özer Çelik and Altunaydin (2018), constituye "el proceso de cambio y mejora del comportamiento mediante la exploración de nueva información a lo largo del tiempo".

En las últimas décadas, estas técnicas han experimentado un crecimiento exponencial, impulsando avances en campos diversos como la conducción autónoma, medicina personalizada y física de partículas (Carleo et al., 2019). Su objetivo esencial es **identificar patrones en datasets complejos** para resolver problemas que serían intratables mediante programación tradicional.

2.2.1. Taxonomía de enfoques de aprendizaje

Basándose en la disponibilidad de datos etiquetados, los enfoques de ML se clasifican en (Bobadilla, 2021):

- **Supervisado:** Requiere datasets con pares entrada-salida conocidos. Incluye regresión (predicción de valores continuos) y clasificación (asignación categórica)
- **No supervisado:** Opera con datos sin etiquetas, buscando estructuras latentes mediante clustering o reducción de dimensionalidad
- **Semi-supervisado:** Combina pequeños conjuntos etiquetados con grandes volúmenes de datos sin etiquetar
- **Por refuerzo:** Aprende mediante interacción con un entorno, optimizando señales de recompensa acumulada

2.3. Detección de anomalías: marco teórico

La **detección de anomalías** identifica patrones que se desvían significativamente del comportamiento esperado en un dataset. Las anomalías pueden representar errores, fraudes, fallos técnicos o, en contextos científicos, fenómenos raros de interés (Chandola et al., 2009).

Esta disciplina tiene aplicaciones en ciberseguridad (detección de intrusiones), finanzas (fraude), medicina (diagnóstico asistido) y, relevante para este proyecto, en física para identificar "nueva física" (Bou Nassif et al., 2022).

2.3.1. Clasificación metodológica

Según la información disponible para entrenamiento, se distinguen tres enfoques (Bou Nassif et al., 2022):

- **Supervisado:** Requiere ejemplos etiquetados de ambas clases (normal y anómala). Limitado por la escasez típica de anomalías etiquetadas

- **Semi-supervisado:** Entrena exclusivamente con instancias normales, detectando como anómalas las desviaciones significativas del patrón aprendido
- **No supervisado:** Infiere normalidad basándose en frecuencia o densidad de datos, asumiendo rareza estadística de anomalías

Para este proyecto se adopta el enfoque **semi-supervisado**, dado que se dispone de abundantes fuentes astrofísicas clasificadas (datos normales) pero no de ejemplos confirmados de materia oscura (anomalías).

2.4. OCSVM

2.4.1. Fundamentos conceptuales

El algoritmo OCSVM es una técnica semi-supervisada que modela el contorno de una única clase en lugar de separar múltiples clases (Schölkopf et al., 2001). Está basado en , una técnica clásica diseñada originalmente para separar dos clases mediante una frontera de decisión óptima.

Como ilustra la figura 2.3, el tradicional busca la frontera que maximiza el margen entre clases. Los **vectores de soporte** (puntos más cercanos a la frontera) determinan esta separación óptima, mientras que las líneas punteadas delimitan las zonas de margen.

SVM Clásico: Separación de Dos Clases

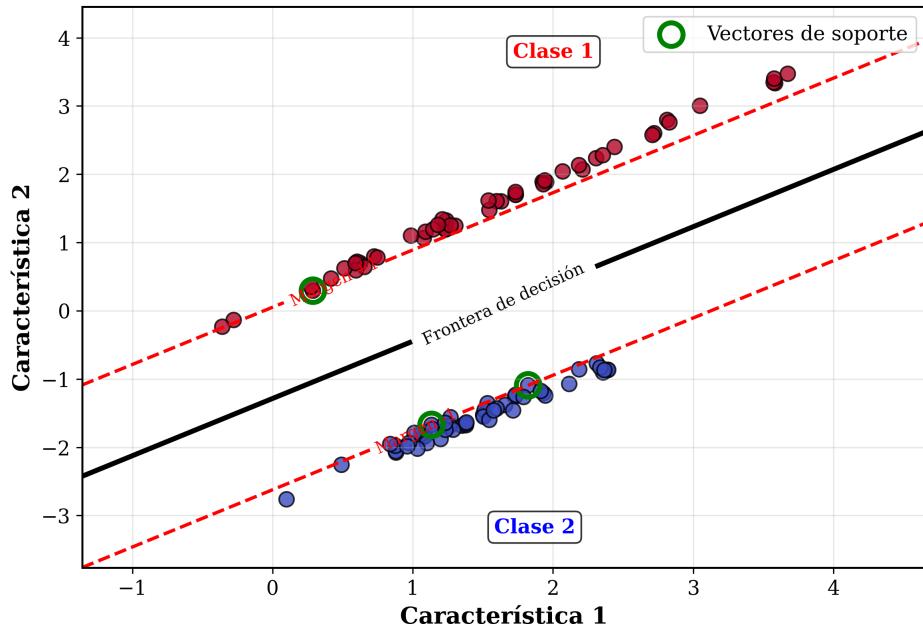


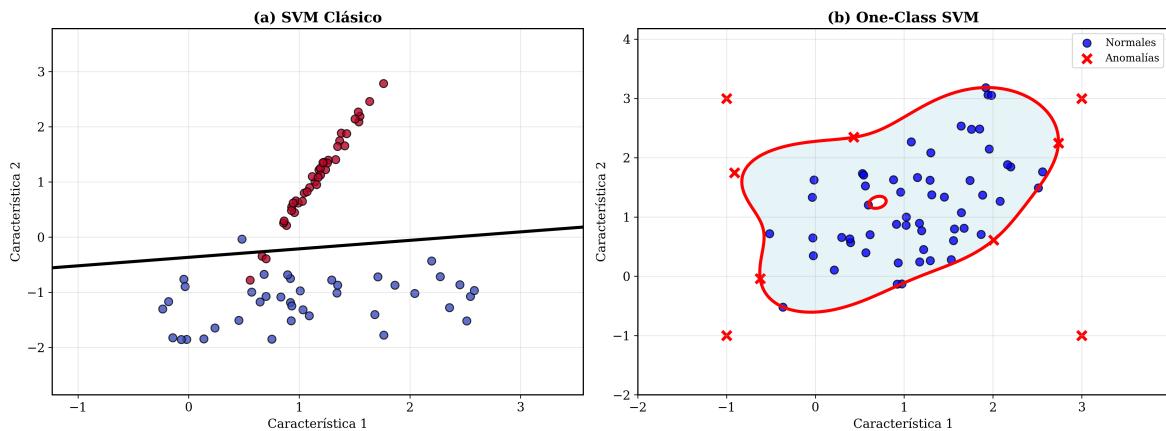
Figura 2.3. clásico: separación de dos clases mediante una frontera de decisión lineal. Los círculos verdes indican los vectores de soporte que determinan el margen óptimo.

OCSVM adapta este concepto para trabajar con una sola clase: construye una frontera que delimita la región donde se concentran los datos normales, utilizando una kernel (una transformación matemática que permite encontrar patrones complejos) para proyectar los datos a un espacio de mayor dimensión donde los patrones son más identificables.

2.4.2. Funcionamiento algorítmico

El RBF, utilizado en este proyecto, es una función matemática que mide la similitud entre puntos basándose en su distancia, permitiendo capturar relaciones no lineales complejas entre características. Una vez transformados los datos mediante esta función, el algoritmo construye una **hiperfrontera** —conceptualizable como una "burbuja" multidimensional— que rodea la mayoría de los datos de entrenamiento, maximizando el margen respecto al origen.

La figura 2.4 ilustra esta diferencia fundamental: mientras clásico separa dos clases conocidas, OCSVM delimita una región normal y detecta como anomalías los puntos externos.



2.4.3. Parámetros de configuración

Los hiperparámetros clave que determinan el comportamiento del OCSVM son:

- ν (**nu**): Controla la fracción de outliers permitidos y el número de vectores de soporte. Valores altos permiten mayor flexibilidad pero incrementan falsos positivos
- Kernel: Función de transformación del espacio de características. El RBF es óptimo para fronteras no lineales complejas
- γ (**gamma**): Parámetro del RBF que determina la influencia individual de cada punto. Valores pequeños generan fronteras suaves; valores grandes producen ajustes más estrictos

La figura 2.5 muestra el resultado final: la hiperfrontera roja delimita la región normal (sombreado azul), mientras que las anomalías detectadas (a las que se denomina también outliers) aparecen como cruces rojas en la región anómala (sombreado coral).

One-Class SVM: Detección de Anomalías

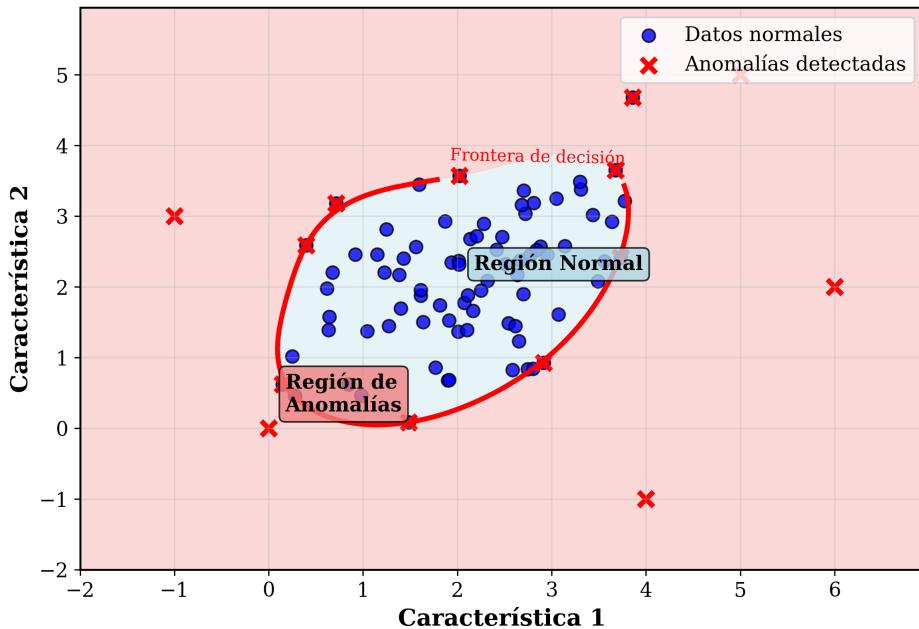


Figura 2.5. OCSVM aplicado: delimitación de la región normal (azul) mediante hiperfrontera (línea roja) y detección de anomalías (cruces rojas) en regiones externas.

2.5. Metodología experimental

2.5.1. Justificación del enfoque OCSVM

La elección de OCSVM para este proyecto se fundamenta en sus ventajas específicas para el contexto astrofísico:

- **Independencia de etiquetas anómalas:** Solo requiere ejemplos de la clase normal (fuentes identificadas)
- **Robustez ante desequilibrios:** Efectivo incluso con anomalías extremadamente escasas
- **Capacidad no lineal:** Los kernels capturan patrones espectrales complejos
- **Sensibilidad ajustable:** Los hiperparámetros ν y γ permiten calibrar la detección según criterios científicos

2.5.2. Pipeline experimental

El flujo experimental implementado (figura 2.6) comprende las siguientes etapas:

1. **Datos iniciales (ASTRO):** Se parte de un conjunto de datos astronómicos con características espectrales en dos configuraciones: 2 características (2F) y 4 características (4F).
2. **Preprocesamiento:** Aplicación de transformaciones logarítmicas y selección de variables relevantes para optimizar la representación de los datos espectrales.
3. **Modelado dual:**
 - **Modelo OCSVM:** Implementación del algoritmo OCSVM utilizando las características 2F y 4F preprocesadas.

- **Modelo ANN (Referencia):** ANN empleada como método de referencia, incorporando características 2F, 4F, repeticiones y valores medios.
4. **Predicción sobre fuentes no identificadas (UNIDs):** Aplicación de ambos modelos para la detección de anomalías versus la estimación de probabilidad de materia oscura (DM) en el conjunto de fuentes no identificadas.
 5. **Resultados:** Generación de dos rankings comparativos:
 - Ranking de anomalías detectadas por OCSVM
 - Ranking de probabilidad de DM estimada por ANN
- Estos rankings permiten realizar un análisis comparativo entre ambos enfoques metodológicos.

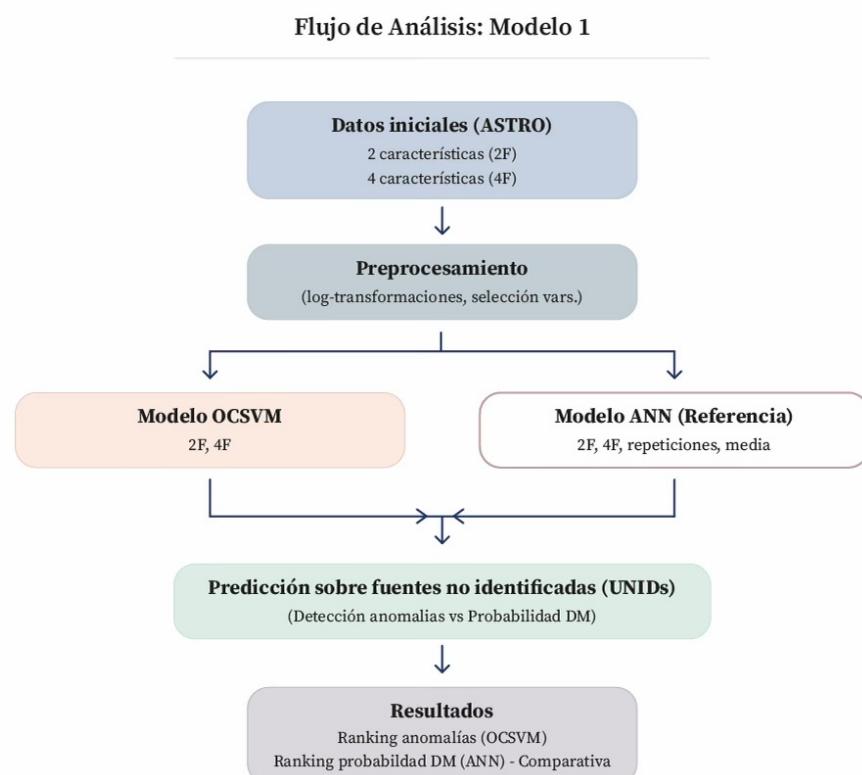


Figura 2.6. Pipeline experimental inicial: desarrollo del modelo OCSVM con características 2F y 4F, utilizando datos del estudio previo de ANNs como referencia de validación.

Adicionalmente, se desarrolla un modelo experimental complementario (capítulo 6) que utiliza una fuente de datos actualizada del catálogo 4FGL-DR4, en contraste con la versión DR3 empleada en los modelos principales.

Este segundo pipeline (figura 2.7) implementa un enfoque simplificado con 3 características (3F) y se aplica directamente sobre fuentes no asociadas (Unassociated source (UNAS)) para la detección de anomalías.

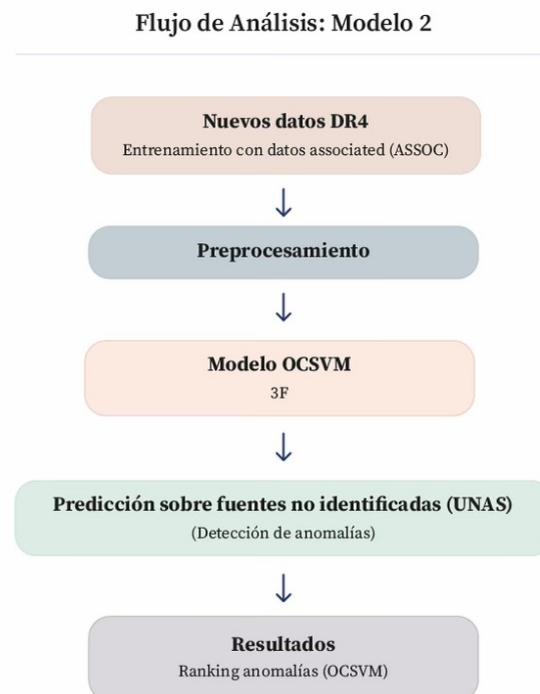


Figura 2.7. Pipeline experimental complementario: modelo OCSVM con 3 características aplicado sobre datos actualizados del catálogo 4FGL-DR4 para la detección de anomalías en fuentes no asociadas (UNAS).

2.6. Estado del arte y contribuciones

2.6.1. Antecedentes en búsqueda de materia oscura

Los primeros enfoques utilizaron análisis estadísticos tradicionales: Mirabal et al. (2012) establecieron criterios físicos para candidatos de materia oscura entre UNIDs, como ausencia de contrapartes y espectros compatibles con teorías de aniquilación. Posteriormente, Gammaldi et al. (2023) desarrollaron metodologías sistemáticas con ANNs, estableciendo un marco robusto que sirve como referencia para este trabajo.

2.6.2. Detección de anomalías en física

Aunque menos explorada en astrofísica gamma, la detección de anomalías ha demostrado efectividad en física de partículas para identificar fenómenos raros. Carleo et al. (2019) destacan su potencial para descubrir "nueva física", mientras Zimek et al. (2012) documentan la eficacia de OCSVM en contextos con abundantes ejemplos normales,

anomalías estadísticamente raras y necesidad de fronteras no lineales—características que coinciden con los requisitos de este proyecto.

2.7. Relevancia para la Ingeniería en Sistemas de Información

Este proyecto demuestra la aplicación de técnicas avanzadas de sistemas de información a problemas científicos complejos, abordando desafíos típicos de la ingeniería informática:

Gestión de Big Data: Procesamiento eficiente de catálogos astronómicos con miles de registros y múltiples atributos, implementando estrategias de almacenamiento y acceso optimizadas.

Desarrollo de pipelines analíticos: Implementación de flujos de trabajo reproducibles y escalables para análisis de datos científicos, incluyendo preprocesamiento, validación y evaluación automatizada.

Optimización algorítmica: Ajuste sistemático de hiperparámetros en problemas de alta dimensionalidad, aplicando técnicas de grid search y validación cruzada.

Visualización de resultados: Desarrollo de herramientas gráficas para interpretación de resultados complejos, facilitando la comunicación de hallazgos científicos.

Este marco metodológico establece un precedente valioso para la intersección entre sistemas de información y ciencias experimentales, demostrando cómo la ingeniería informática puede contribuir eficazmente a problemas fundamentales de la física contemporánea.

Capítulo 3

Gestión del Proyecto

3.1. Metodología ágil adoptada

Para la planificación y ejecución de este TFG se ha adoptado un enfoque basado en **metodologías ágiles**, específicamente Scrum complementado con prácticas de eXtreme Programming (XP). Esta elección se justifica por la naturaleza iterativa inherente a los proyectos de *Machine Learning*, donde el proceso ETL (*Extract, Transform, Load*) estructura fases de definición del problema, preparación de datos, desarrollo de modelos y evaluación de manera cíclica.

El enfoque ágil aporta beneficios clave mediante una **estrategia incremental y de aprendizaje continuo**, favoreciendo la incorporación progresiva de conocimiento, revisión constante de avances y capacidad de ajustar objetivos conforme se obtienen resultados parciales.

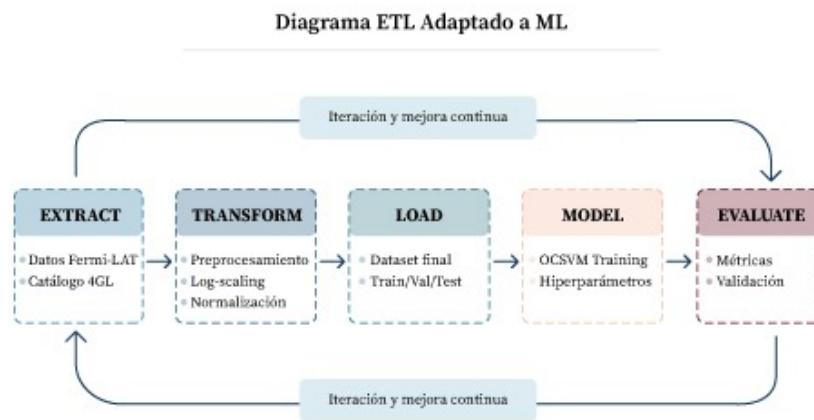


Figura 3.1. Ciclo de vida ETL adaptado a Machine Learning, mostrando la naturaleza iterativa del desarrollo de modelos que justifica la adopción de metodologías ágiles.

3.2. Organización de roles

La adaptación de Scrum a un proyecto académico individual ha requerido una distribución específica de roles:

- **Product Owner (Tutor):** Responsable de definir objetivos, priorizar tareas principales, proporcionar retroalimentación durante reuniones de seguimiento y supervisar la calidad de entregables
- **(Alumno):** Desempeño integral de funciones operativas, incluyendo planificación de Sprints (Sprints), ejecución de tareas, implementación de modelos, documentación de resultados y redacción de la memoria
- **Scrum Master (Autogestionado):** Funciones de facilitación de procesos y eliminación de impedimentos asumidas por el alumno mediante autogestión, con supervisión puntual del tutor para resolver bloqueos específicos

3.3. Estructura del proyecto

3.3.1. Organización en épicas y sprints

El proyecto se estructuró en cuatro épicas principales, desarrolladas mediante *Sprints* de aproximadamente dos semanas:

1. **EPIC-01 - Definición y organización inicial:** Establecimiento de objetivos, alcance del proyecto, marco teórico inicial y organización del *backlog*
2. **EPIC-02 - Análisis del modelo base:** Ejecución, análisis y documentación del modelo ANN preexistente para establecer línea base comparativa
3. **EPIC-03 - Desarrollo del modelo OCSVM:** Diseño, implementación, entrenamiento y optimización del nuevo modelo de detección de anomalías mediante iteraciones incrementales
4. **EPIC-04 - Integración y documentación:** Análisis comparativo de resultados, preparación de productos finales y redacción de la memoria definitiva

Diagrama del Ciclo de Vida Iterativo

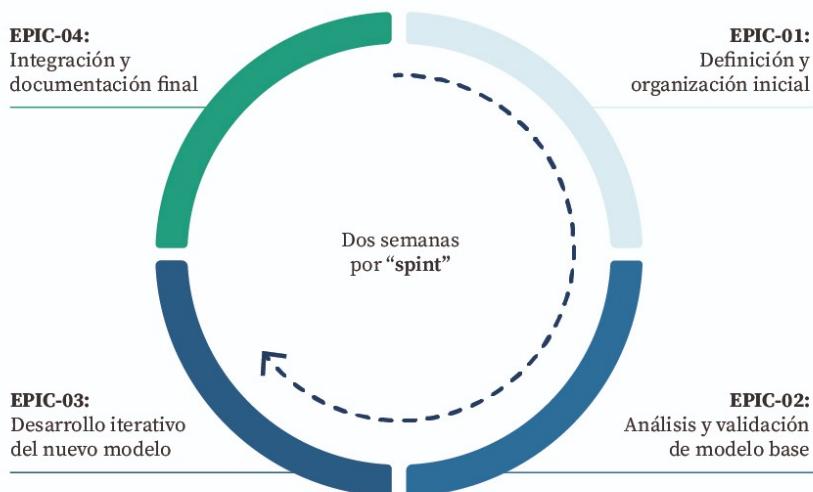


Figura 3.2. Ciclo de vida iterativo organizado en sprints para la gestión ágil del TFG.

3.3.2. Estructura de cada sprint

Cada *sprint* siguió una estructura estándar:

- **Planificación:** Definición de User Stories con criterios de aceptación claros, priorizadas según impacto y viabilidad
- **Ejecución:** Desarrollo de tareas con seguimiento mediante reuniones periódicas con el tutor y autogestión de progreso
- **Revisión:** Evaluación del cumplimiento de objetivos y documentación de logros
- **Retrospectiva:** Reflexión sobre áreas de mejora para aplicar en ciclos siguientes

3.4. Definición de requisitos del sistema

Previo al desarrollo de los modelos de detección de anomalías, se realizó un análisis sistemático de requisitos para garantizar que la solución propuesta cumpliera con los objetivos científicos del proyecto y fuera técnicamente viable dentro del contexto de investigación astrofísica.

3.4.1. Requisitos funcionales

Los requisitos funcionales definen las capacidades específicas que debe proporcionar el sistema de detección de anomalías:

1. **RF-01 - Detección de anomalías:** El sistema debe identificar fuentes astrofísicas con características atípicas que puedan corresponder a candidatos de materia oscura, utilizando técnicas de aprendizaje no supervisado
2. **RF-02 - Clasificación binaria:** El sistema debe clasificar cada fuente como “normal” (ASTRO) o “anómala” (candidato a materia oscura) basándose en sus características observacionales
3. **RF-03 - Generación de puntuaciones de decisión:** El sistema debe proporcionar *decision scores* cuantitativos que permitan rankear las anomalías según su grado de desviación respecto al comportamiento normal
4. **RF-04 - Procesamiento de múltiples configuraciones:** El sistema debe soportar el análisis con diferentes combinaciones de características (2F, 3F, 4F) para evaluar el impacto de cada variable en la detección
5. **RF-05 - Comparabilidad con métodos existentes:** El sistema debe generar resultados comparables con el modelo ANN de referencia para validar la complementariedad metodológica
6. **RF-06 - Visualización de resultados:** El sistema debe proporcionar representaciones gráficas de las fronteras de decisión y distribuciones de *scores* para facilitar la interpretación astrofísica

3.4.2. Requisitos no funcionales

Los requisitos no funcionales establecen las características de calidad y restricciones operativas del sistema:

1. **RNF-01 - Precisión en fuentes normales:** El sistema debe clasificar correctamente al menos el 99 % de las fuentes ASTRO conocidas como “normales”, manteniendo alta especificidad
2. **RNF-02 - Consistencia de detección:** Los outliers detectados deben mantenerse consistentes en múltiples ejecuciones y proyecciones del espacio de características, con variabilidad inferior al 5 %
3. **RNF-03 - Escalabilidad:** El sistema debe procesar eficientemente conjuntos de datos con miles de fuentes sin degradación significativa del rendimiento
4. **RNF-04 - Interpretabilidad:** Los resultados deben ser interpretables desde el punto de vista astrofísico, proporcionando insights sobre las características que definen las anomalías
5. **RNF-05 - Reproducibilidad:** El sistema debe producir resultados reproducibles mediante el uso de semillas aleatorias fijas y documentación completa de parámetros
6. **RNF-06 - Flexibilidad paramétrica:** El sistema debe permitir el ajuste de hiperparámetros (especialmente ν y γ) para adaptarse a diferentes niveles de sensibilidad en la detección

3.5. Planificación temporal

3.5.1. Planificación inicial

La figura 3.3 muestra la planificación inicial mediante diagrama de Gantt. Aunque visualmente pueda parecer un enfoque en cascada, cada bloque del Gantt representa una épica que internamente se desarrolló de manera iterativa mediante *sprints*. Esta representación híbrida permitió visualizar dependencias de alto nivel entre épicas mientras mantenía flexibilidad iterativa dentro de cada una.

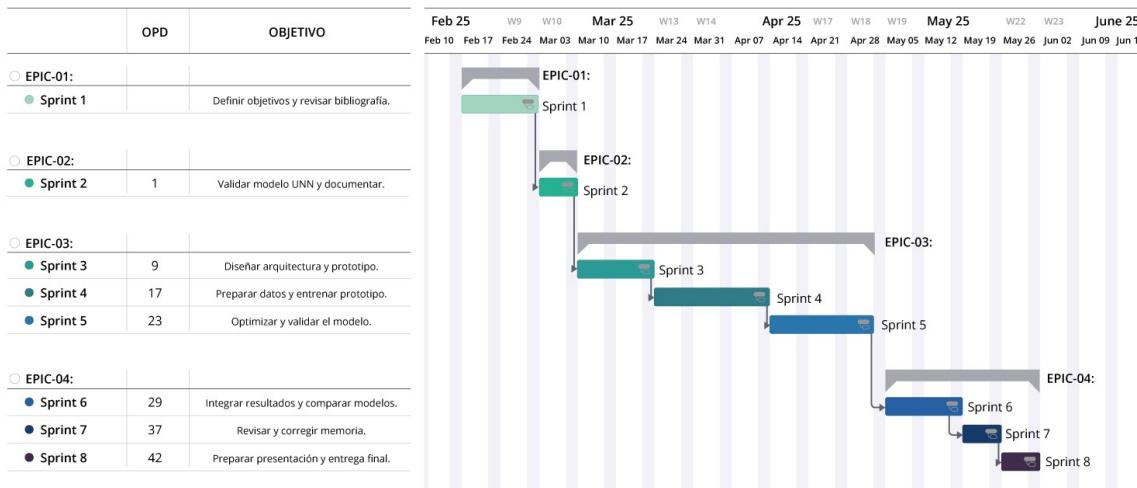


Figura 3.3. Planificación temporal inicial del proyecto. El diagrama de Gantt muestra la distribución de épicas y sprints con sus dependencias y progresión secuencial.

La tabla 3.1 muestra la organización del proyecto en épicas y en *sprints*.

Sprint	Épica	Objetivo	Entregables
Sprint 1	EPIC-01	Definir alcance y organizar <i>backlog</i>	Alcance definido, <i>backlog</i> priorizado, marco teórico inicial
Sprint 2	EPIC-02	Validar modelo base ANN	Código ejecutado, resultados iniciales, metodología preliminar
Sprint 3	EPIC-03 (I)	Diseñar arquitectura OCSVM	Arquitectura definida, prototipo base
Sprint 4	EPIC-03 (II)	Entrenar y obtener resultados preliminares	Prototipo entrenado, resultados preliminares
Sprint 5	EPIC-03 (III)	Optimizar y validar modelo	Modelo optimizado, documentación de mejoras
Sprint 6	EPIC-04 (I)	Integrar y comparar resultados	Resultados integrados, análisis comparativo
Sprint 7	EPIC-04 (II)	Revisar borrador final	Borrador revisado y corregido
Sprint 8	EPIC-04 (III)	Preparar defensa y entrega	Presentación final, memoria definitiva

Tabla 3.1. Organización del proyecto en épicas y *sprints*

En el Apéndice II se incluye la organización más detallada del proyecto, con la subdivisión de los *sprints* en tareas o *tasks*, así como los criterios de aceptación.

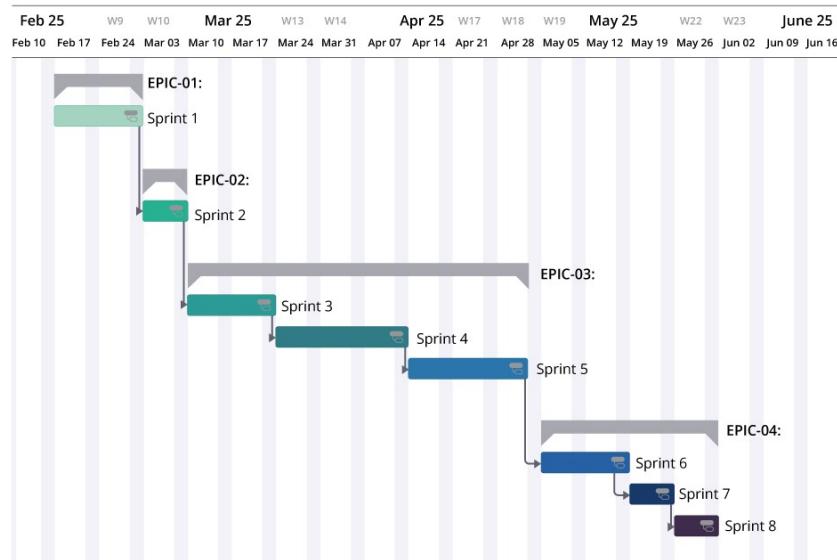


Figura 3.4. Vista de gestión detallada mostrando la organización jerárquica de épicas, *sprints*.

3.5.2. Ejecución real y adaptaciones

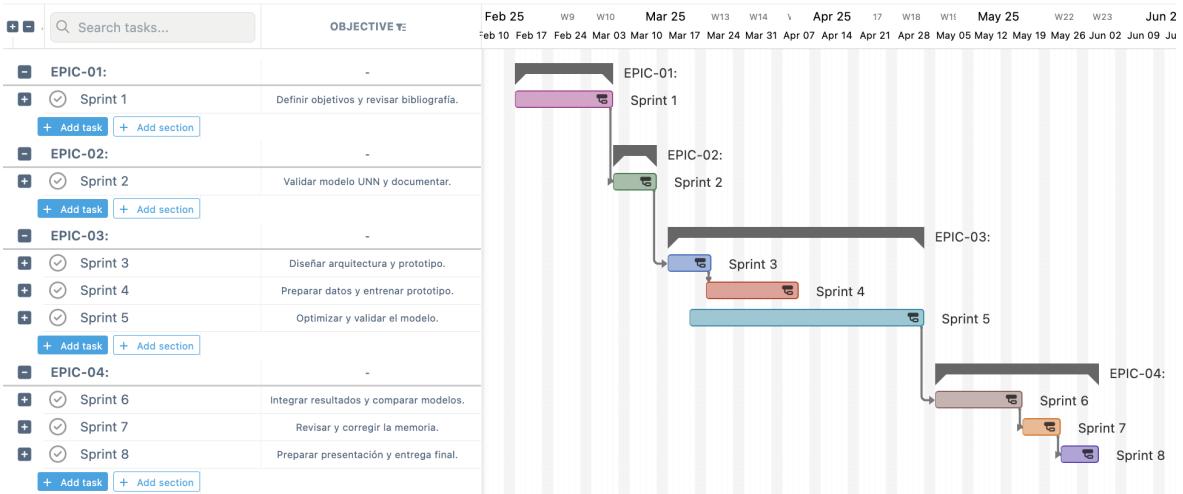


Figura 3.5. Planificación final ejecutada mostrando las adaptaciones realizadas durante el desarrollo del proyecto respecto a la planificación inicial.

Durante la ejecución del proyecto, la planificación inicial experimentó adaptaciones necesarias debido a la naturaleza exploratoria de la investigación en *machine learning*. La figura 3.5 ilustra cómo se ajustaron los tiempos y prioridades:

- **Extensión de EPIC-03:** El desarrollo del modelo OCSVM requirió iteraciones adicionales para la optimización de hiperparámetros y la incorporación de nuevas características (extensión a 3F con datos DR4)
- **Refinamiento iterativo:** Los resultados preliminares motivaron ajustes en la metodología y análisis adicionales no contemplados inicialmente

Esta flexibilidad demostró la efectividad de combinar **planificación estructural de alto nivel** (épicas secuenciales) con **desarrollo ágil de bajo nivel** (iteraciones dentro de cada épica), permitiendo adaptarse a los desafíos inherentes de la investigación aplicada sin perder el control temporal del proyecto.

3.6. Implementación técnica

3.6.1. Entorno de desarrollo

El desarrollo técnico se realizó utilizando:

- **Python 3.9.2** con **Scikit-Learn**: Suite completa para el ciclo de desarrollo de modelos ML (preprocesamiento, entrenamiento, ajuste de hiperparámetros)
- **Jupyter Notebooks** en **Visual Studio Code**: Análisis interactivo, visualización de resultados intermedios y documentación integrada del código

3.6.2. Desarrollo de modelos

Como se introduce más arriba y para concretar nuevamente, los modelos desarrollados fueron:

Modelo OCSVM

Se implementaron múltiples variantes del modelo One-Class Support Vector Machine:

- **OCSVM-2F**: Modelo inicial con dos características ($\varepsilon_{\text{peak}}$ y β)
- **OCSVM-4F**: Extensión incorporando σ_{det} y β_{rel}
- **OCSVM-3F**: Modelo adicional con datos DR4 utilizando tres características específicas (α , β y flux)

Modelo de referencia ANN

Se analizó y validó el modelo de red neuronal previamente desarrollado, sirviendo como **referencia comparativa** para evaluar la complementariedad metodológica y contrastar la selección de candidatos a materia oscura.

3.6.3. Métricas de evaluación

Dado el carácter semi-supervisado del OCSVM, se emplearon métricas específicas para detección de anomalías:

- **Fracción de outliers**: Porcentaje de fuentes clasificadas como anómalas, validando consistencia con el parámetro ν
- **Decision scores**: Análisis de distribución para asegurar separación clara entre regiones normales y anómalas
- **Consistencia de frontera**: Validación de que la frontera englobe apropiadamente fuentes ASTRO (>99 %) manteniendo capacidad discriminatoria
- **Robustez**: Verificación de consistencia de outliers detectados en múltiples proyecciones del espacio de características

La validación se complementó con **visualización gráfica** de fronteras de decisión y distribuciones, facilitando la interpretación astrofísica de patrones detectados y la identificación de candidatos coherentes con expectativas teóricas de señales de materia oscura.

Capítulo 4

Modelo de referencia (ANN)

Como se estableció en los capítulos anteriores, para validar la efectividad del enfoque OCSVM propuesto es fundamental disponer de una línea base de comparación robusta. Este capítulo analiza y reproduce el modelo de **redes neuronales artificiales** (ANN, por sus siglas en inglés) desarrollado en estudios previos (Gammaldi et al., 2023), proporcionando un *benchmark* supervisado contra el cual evaluar el rendimiento de las técnicas de detección de anomalías implementadas.

Las **redes neuronales artificiales** son modelos computacionales inspirados en el funcionamiento del cerebro humano, compuestos por unidades de procesamiento interconectadas (neuronas artificiales) organizadas en capas. En el contexto de este proyecto, la ANN aborda el problema como una tarea de **clasificación supervisada**, aprendiendo a distinguir entre fuentes astrofísicas convencionales y simulaciones de señales de materia oscura mediante entrenamiento con ejemplos etiquetados de ambas clases.

Este enfoque supervisado contrasta fundamentalmente con la metodología semi-supervisada del OCSVM, proporcionando una perspectiva complementaria para la identificación de candidatos a materia oscura en el catálogo Fermi-LAT.

4.1. Arquitectura y configuración general

Los modelos de referencia implementan una arquitectura de **red neuronal artificial** basada en el perceptrón multicapa (Multi-Layer Perceptron (MLP)), diseñada específicamente para clasificación binaria de fuentes gamma.

La arquitectura común de ambos modelos (2F para dos parámetros y 4F para cuatro parámetros) consta de tres componentes principales:

1. La **capa de entrada** recibe las características espectrales de cada fuente gamma, con un tamaño variable según el modelo: 2 neuronas para el modelo 2F y 4 neuronas para el modelo 4F, correspondientes al número de parámetros espectrales utilizados como variables de entrada.
2. La **capa oculta** constituye el núcleo de procesamiento de la red, con 21 neuronas que procesan la información de entrada. Cada neurona aplica una función de activación ReLU (Rectified Linear Unit), que permite a la red aprender relacio-

nes no lineales entre las variables de entrada y mantiene solo los valores positivos, estableciendo los negativos en cero.

3. La **capa de salida** consiste en una única neurona con función de activación sigmoidal que produce la clasificación final. Esta función convierte la salida en un valor entre 0 y 1, interpretable como la probabilidad de que una fuente pertenezca a la clase de interés (posible señal de materia oscura).

El entrenamiento se realiza utilizando el conjunto de datos etiquetados que incluye fuentes astrofísicas convencionales del catálogo Fermi-LAT como Clase 0 y **fuentes simuladas** con características espectrales compatibles con emisión **de materia oscura** como Clase 1, generadas mediante modelos teóricos, aplicándose transformación logarítmica en base 10 para mejorar la distribución de los datos y facilitar el procesamiento por los algoritmos de *machine learning*.

Para la optimización se emplea el algoritmo **Adaptive Moment Estimation (Adam)** con una tasa de aprendizaje de 0.015, que adapta automáticamente la velocidad de aprendizaje durante el entrenamiento.

El preprocessamiento de datos se realiza mediante **StandardScaler**, que normaliza las variables de entrada restando la media y dividiendo por la desviación estándar. Esta normalización es crucial para que todas las características contribuyan equitativamente al proceso de aprendizaje, independientemente de sus escalas originales.

Dada la *naturaleza estocástica* de las redes neuronales artificiales, donde la **inicialización aleatoria de pesos** puede influir en los resultados finales, se implementó una metodología robusta de validación que incluye **dos niveles de aleatorización**.

- Primero, se aplica **validación cruzada estratificada**, dividiendo los datos en múltiples subconjuntos manteniendo la proporción original de clases. El modelo se entrena y evalúa repetidamente utilizando diferentes combinaciones de estos subconjuntos, proporcionando una estimación más confiable del rendimiento.
- Segundo, se realizan **ejecuciones múltiples independientes**, ejecutando todo el proceso tres veces con diferentes semillas aleatorias para abordar la variabilidad inherente tanto en la inicialización de pesos como en las divisiones de datos. Los resultados finales se obtienen promediando estas ejecuciones independientes.

Esta metodología garantiza que los resultados obtenidos sean estadísticamente robustos y reproducibles, aspectos fundamentales para establecer la confiabilidad del modelo de referencia sobre el cual se desarrollarán las mejoras propuestas en este trabajo.

4.2. Entrenamiento y aplicación del modelo ANN 2F

El modelo 2F utiliza las dos características espectrales fundamentales: $\log_{10}(E_{\text{peak}})$ y $\log_{10}(\beta)$, con validación cruzada de 5 pliegues y 2 repeticiones (10 evaluaciones por ejecución). El entrenamiento completo genera 30 modelos independientes (3 ejecuciones \times 10 evaluaciones).

Una vez completado el entrenamiento, el modelo se aplica al conjunto de fuentes no identificadas (UNIDs) del catálogo 4FGL-DR3. Para cada UNID, cada ejecución genera 10 probabilidades independientes, calculándose la probabilidad media, desviación estándar y coeficiente de variación por ejecución. El "consenso" se refiere al análisis de concordancia entre las 3 ejecuciones independientes, calculando probabilidad consenso y variabilidad entre ejecuciones.

Métrica	Ejecución 1	Ejecución 2	Ejecución 3
Candidatos moderados ($p \geq 50\%$)	1	0	1
Máxima probabilidad media	0.523	0.478	0.501
UNIDs con baja incertidumbre	9	8	12

Tabla 4.1. Resultados de aplicación del modelo ANN 2F a UNIDs

El análisis revela un comportamiento conservador: 12 UNIDs únicas aparecen como candidatos en al menos una ejecución, pero ninguno muestra consenso completo en las tres ejecuciones.

Los **candidatos principales** son:

1. **Source 664** (probabilidad consenso 0.523 ± 0.027)
2. **Source 1114** (0.501 ± 0.027).

La distribución de los resultados de predicción del modelo ANN 2F y de los candidatos en el espacio de características se puede observar en la figura 1 y figura 2 del Apéndice III.

4.3. Entrenamiento y aplicación del modelo ANN 4F

El modelo 4F extiende las características incorporando $\log_{10}(\sigma_{\text{det}})$ (*significancia de detección*) y $\log_{10}(\beta_{\text{rel}})$ (*variante relativística del parámetro espectral*), con validación cruzada simplificada de 5 pliegues y 1 repetición (5 evaluaciones por ejecución). El entrenamiento genera 15 modelos independientes.

La aplicación a UNIDs sigue la misma metodología, pero cada ejecución genera 5 probabilidades independientes en lugar de 10. El modelo 4F muestra comportamiento significativamente diferente al 2F, evidenciando el impacto de las características adicionales.

Métrica	Ejecución 1	Ejecución 2	Ejecución 3
Candidatos moderados ($p \geq 50\%$)	22	25	26
Máxima probabilidad media	0.785	0.743	0.672
UNIDs con baja incertidumbre	118	122	125

Tabla 4.2. Resultados de aplicación del modelo ANN 4F a UNIDs

El modelo 4F identifica 53 UNIDs únicas como **candidatos**, con 3 mostrando consenso completo:

1. **Source 371** (probabilidad consenso 0.785 ± 0.019)
2. **Source 821** (0.743 ± 0.051)

3. **Source 596** (0.672 ± 0.035)

La distribución espacial es más uniforme comparada con la localización específica del modelo 2F ([figura 3](#) del [Ápendice III](#)).

4.4. Análisis comparativo y línea base

La comparación directa revela el impacto significativo de incorporar características adicionales: el modelo 4F identifica $4.4 \times$ más candidatos únicos, alcanza mayor confianza (rango 0.534-0.785 vs 0.455-0.523), y muestra mejor reproducibilidad en candidatos top (3 con consenso completo vs 0). Ambos modelos mantienen estabilidad entre ejecuciones (variabilidad 0.0071 vs 0.0081).

Los resultados ANN establecen una línea base específica para validación cruzada con OCSVM. Los **candidatos prioritarios** incluyen los de consenso completo del modelo 4F (**Sources 371, 821, 596**) y los principales del modelo 2F (**Sources 664, 1114**). Esta línea base permite evaluar detección convergente, complementaria, validación cruzada y exploración espacial del enfoque OCSVM.

La metodología implementada proporciona benchmarks cuantitativos, framework robusto para cuantificar incertidumbre, objetivos específicos de validación y criterios para evaluar complementariedad entre paradigmas de aprendizaje automático aplicados a la búsqueda de materia oscura. Las figuras ilustrativas de distribuciones de probabilidades y espacios de características se incluyen en el [Ápendice correspondiente](#).

Capítulo 5

Desarrollo experimental (OCSVM)

Establecido el modelo de referencia basado en ANNs para la clasificación binaria de fuentes gamma, el presente capítulo desarrolla el núcleo experimental de este trabajo: la implementación de un enfoque alternativo basado en OCSVM para la detección de anomalías en el catálogo de fuentes no identificadas de Fermi-LAT.

A diferencia del modelo de referencia que requiere ejemplos etiquetados de ambas clases (fuentes convencionales y candidatas a materia oscura simuladas), el enfoque OCSVM propuesto representa un cambio paradigmático en la estrategia de búsqueda. Mientras que el modelo ANN aprende a distinguir entre patrones "normales" y patrones "compatibles con materia oscura" para asignar probabilidades específicas, el OCSVM adopta un enfoque de **detección de anomalías**: aprende exclusivamente las características de fuentes astrofísicas conocidas y etiqueta como outliers aquellas fuentes no identificadas que se desvían significativamente de este patrón aprendido.

Esta diferencia metodológica fundamental implica que ambos enfoques no necesariamente identificarán los mismos candidatos. El modelo ANN busca fuentes que se asemejen a simulaciones teóricas de materia oscura, mientras que el OCSVM identifica fuentes que simplemente son "anómalias" respecto al comportamiento astrofísico convencional. En principio, si existieran verdaderas señales de materia oscura entre las fuentes no identificadas, cabría esperar cierta coincidencia entre los candidatos principales de ambos métodos. Sin embargo, como se analizará en las conclusiones, los resultados sugieren que ambos enfoques son más bien complementarios que convergentes, lo que plantea interesantes implicaciones para la estrategia de búsqueda.

El desarrollo experimental se estructura en torno a la aplicación sistemática de OCSVM sobre diferentes conjuntos de datos y configuraciones, evaluando su capacidad para identificar fuentes con características espectrales inusuales que podrían corresponder a emisiones de materia oscura. A continuación se detallan los datos utilizados, el preprocesamiento aplicado y la metodología experimental implementada.

5.1. Datos utilizados y preprocesamiento

5.1.1. Dataset del estudio previo

Los datos utilizados para el entrenamiento del modelo provienen del estudio previo con ANNs ([Gammaldi et al., 2023](#)), que combina fuentes identificadas del catálogo Fermi-LAT con fuentes simuladas de materia oscura. El dataset contiene **5.662 observaciones** balanceadas entre dos clases:

- **Fuentes astrofísicas (ASTRO):** 2,831 observaciones etiquetadas como 0.0
- **Fuentes de materia oscura simuladas (DM):** 2,831 observaciones etiquetadas como 1.0

Cada observación está caracterizada por cuatro parámetros espectrales fundamentales previamente transformados a escala logarítmica (\log_{10}):

- **Log(E_peak):** Energía del pico espectral ()
- **Log(beta):** Curvatura espectral
- **Log(sigma):** Significancia estadística de detección
- **Log(beta_rel):** Error relativo de la curvatura espectral

5.1.2. Análisis exploratorio

El **análisis de distribuciones** individuales (detallado en el Apéndice IV) revela características distintivas: E_peak presenta distribución aproximadamente normal (media=1.018), beta muestra comportamiento bimodal (media=-0.819) indicando dos poblaciones distintas, sigma exhibe sesgo izquierdo típico de datos de significancia (media=1.061), y beta_rel presenta distribución simétrica centrada en -0.062 ([figura 4 - Ap. IV](#)).

La **matriz de correlación** ([figura 5 - Ap. IV](#)) muestra relaciones moderadas entre variables, destacando Log(E_peak)-Log(beta_rel) ($r=0.36$) como la correlación más fuerte, y correlaciones negativas significativas Log(beta)-Log(sigma) ($r=-0.29$) y Log(sigma)-Log(beta_rel) ($r=-0.35$). Estas correlaciones moderadas confirman que las variables aportan información complementaria, justificando su inclusión conjunta.

Los **gráficos de dispersión** ([figura 6 - Ap. IV](#)) confirman separabilidad entre clases: ASTRO forma nubes compactas en regiones de baja energía y alta significancia, mientras DM se extiende hacia valores más altos, evidenciando patrones de separación consistentes en el espacio multidimensional.

5.1.3. Calidad y preparación de datos

El análisis de calidad reveló un dataset limpio: 0 valores faltantes, 0 filas duplicadas, y outliers presentes pero mantenidos por representar variabilidad natural de los datos astrofísicos.

Se generaron tres datasets para entrenamiento OCSVM:

1. **Dataset ASTRO:** 2,831 observaciones para entrenamiento como clase normal
2. **Dataset DM:** 2,831 simulaciones para posibles análisis comparativos futuros

5.1.4. Fuentes no identificadas (UNIDs)

El segundo conjunto de datos utilizado en este trabajo corresponde a UNIDs del catálogo 3FGL del telescopio espacial Fermi-LAT.

Este dataset representa el objetivo final de aplicación del modelo OCSVM: identificar entre las **1.125 fuentes no identificadas** aquellas que podrían ser candidatas a emisión de materia oscura. Cada fuente UNID está caracterizada por los mismos cuatro parámetros espectrales que el dataset de entrenamiento:

- **E_peak:** Energía del pico espectral
- **beta:** Curvatura espectral
- **sigma_det:** Significancia estadística de detección
- **beta_Rel:** Error relativo de la curvatura espectral
- **number:** Identificador único de la fuente (0-1124)

A diferencia del dataset de entrenamiento, estas fuentes carecen de clasificación conocida, constituyendo el conjunto de datos sobre el cual se aplicará el modelo entrenado para detectar posibles anomalías.

5.1.5. Dataset UNIDs y compatibilización de escalas

El análisis del dataset UNIDs reveló un problema crítico de incompatibilidad: los datos se encontraban en **escala lineal**, mientras que el modelo OCSVM fue entrenado con datos en **escala logarítmica**. Esta incompatibilidad se manifestó en rangos completamente diferentes y distribuciones incongruentes que impedían la aplicación directa del modelo.

Transformación logarítmica requerida

Para resolver la incompatibilidad, se implementó transformación logarítmica (\log_{10}) sobre todas las variables UNIDs:

$$\text{Log}(X) = \log_{10}(X) \quad \text{para } X \in \{\text{E_peak}, \beta, \sigma, \beta_{\text{rel}}\}$$

La verificación visual mediante histogramas y scatter plots comparativos (detallada en el Apéndice V) demostró el éxito de la transformación: las distribuciones UNIDs transformadas mostraron superposición clara con las distribuciones ASTRO, validando la correcta preparación de los datos.

Dataset final estructurado

El dataset UNIDs transformado fue estructurado consistentemente con el dataset de entrenamiento, generando el archivo **unids_transformed_complete.txt**, dataset completo con identificadores para trazabilidad.

Esta preparación garantiza que el modelo OCSVM pueda aplicarse confiablemente para identificar candidatos potenciales entre las 1.125 fuentes no identificadas.

Las figuras 8 y 9 del Apéndice V muestran los gráficos de dispersión de las fuentes UNIDs en los diferentes espacios de características antes y después de su transformación logarítmica, mientras que la figura 10 muestra una comparativa analítica de las distribuciones de las fuentes ASTRO y UNID una vez aplicada la transformación.

5.2. Desarrollo del modelo OCSVM

5.2.1. Motivación y justificación del enfoque semi-supervisado

Como se introduce en apartados anteriores, en este trabajo se utiliza el algoritmo OCSVM como herramienta para modelar el comportamiento de fuentes astrofísicas conocidas (ASTRO) y detectar posibles fuentes anómalas entre los objetos no identificados (UNIDs) del catálogo 4FGL.

El modelo OCSVM está diseñado específicamente para tareas de detección de anomalías cuando se dispone únicamente de datos pertenecientes a una sola clase (en este caso, fuentes astrofísicas conocidas). El objetivo es aprender una **frontera de decisión** o **hiperfrontera** que delimite la región del espacio de características donde se concentran los datos normales, de forma que cualquier punto que caiga fuera de esta región pueda considerarse una posible anomalía.

5.2.2. Arquitectura del modelo OCSVM

El comportamiento del modelo OCSVM depende principalmente de dos hiperparámetros clave:

Parámetro	Descripción	Impacto
ν (nu)	Proporción máxima de datos de entrenamiento que el modelo puede considerar anómalos	Si ν es pequeño, la frontera se ajusta ampliamente. Si ν es grande, se toleran más outliers
γ (gamma)	Define la influencia de cada muestra sobre la forma de la frontera (RBF)	Valores altos producen fronteras ajustadas; valores bajos dan fronteras suaves

Tabla 5.1. Hiperparámetros clave del modelo OCSVM

El objetivo del modelo OCSVM es detectar candidatos a materia oscura entre los UNIDs, partiendo del supuesto de que las fuentes ASTRO representan el patrón de normalidad. Por tanto, **es deseable que la frontera de decisión englobe la mayor parte de las fuentes ASTRO**, evitando overfitting (muchas fuentes ASTRO como anómalas) y underfitting (frontera demasiado amplia que incluya también UNIDs anómalos).

En este trabajo se ha optado por ajustar la frontera de decisión de forma que se incluyan la mayoría (idealmente $>99\%$) de fuentes ASTRO como normales, identificando como anómalos únicamente los UNIDs cuya representación se aleje significativamente del comportamiento aprendido.

5.2.3. Modelo OCSVM con 2 características (2F)

Para el desarrollo del modelo se seleccionaron dos características fundamentales: Log(E_peak) y Log(beta). Estas variables capturan las propiedades espectrales más distintivas de las fuentes astrofísicas y son críticas para identificar posibles desviaciones que podrían indicar señales de materia oscura.

Los datos se dividieron en conjuntos de entrenamiento (60 %), validación (20 %) y test (20 %) para garantizar evaluación robusta. Se aplicó normalización StandardScaler para asegurar que ambas características contribuyan equitativamente al modelo.

Consulta del código asociado

El código completo para el entrenamiento del modelo OCSVM 2F puede consultarse en el apéndice VII. El pipeline general se describe en el algoritmo 1, con implementaciones específicas del grid search en el listado 3 y el entrenamiento final en el listado 4.

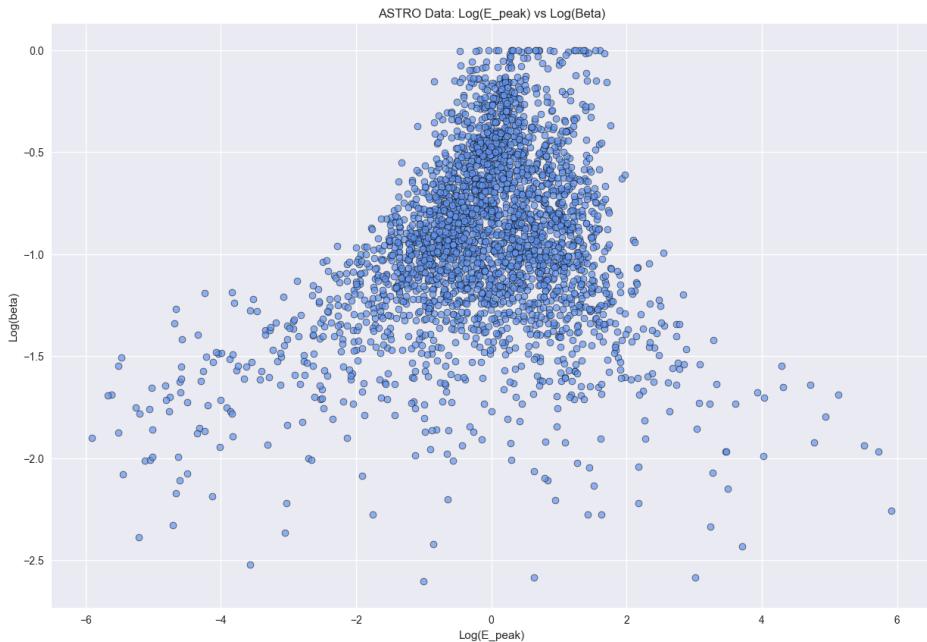


Figura 5.1. Distribución de las fuentes astrofísicas conocidas (ASTRO) en el espacio de características Log(E_peak) vs Log(beta)

Optimización de hiperparámetros

Parámetro Gamma ($\gamma = 0.1$)

La selección del parámetro gamma requirió un análisis cuidadoso del balance entre flexibilidad y generalización. Tras exploración sistemática de valores en el rango [0.001 - 10], se estableció $\gamma = 0.1$ como valor óptimo:

- **Valores menores ($\gamma < 0.1$):** Generaban fronteras excesivamente simples que no capturaban la estructura irregular de las distribuciones espectrales astrofísicas
- **Valores mayores ($\gamma > 0.1$):** Producían overfitting con micro-fronteras que clasificarían ruido estadístico como anomalías
- $\gamma = 0.1$: Proporcionaba el balance óptimo, permitiendo que la frontera capturara la forma natural de la distribución sin crear regiones fragmentadas

Parámetro Nu ($\nu = 0.001$)

Se realizó búsqueda exhaustiva del parámetro ν en el rango [0.001 - 0.2], optimizando para minimizar fuentes astrofísicas clasificadas incorrectamente como anomalías. El resultado óptimo fue $\nu = 0.001$, representando un límite superior del 0.1 % de errores permitidos, reflejando la expectativa de que las señales de materia oscura son extremadamente raras.

Resultados del modelo final 2F

El modelo OCSVM optimizado demostró comportamiento consistente y astrofísicamente coherente:

Rendimiento:

- **Entrenamiento:** 3 outliers detectados (0.1 %, consistente con ν)
- **Test:** 5 outliers detectados (0.2 %, dentro del rango esperado)
- **Precisión global:** 99.1 % (562 de 567 muestras correctamente clasificadas)

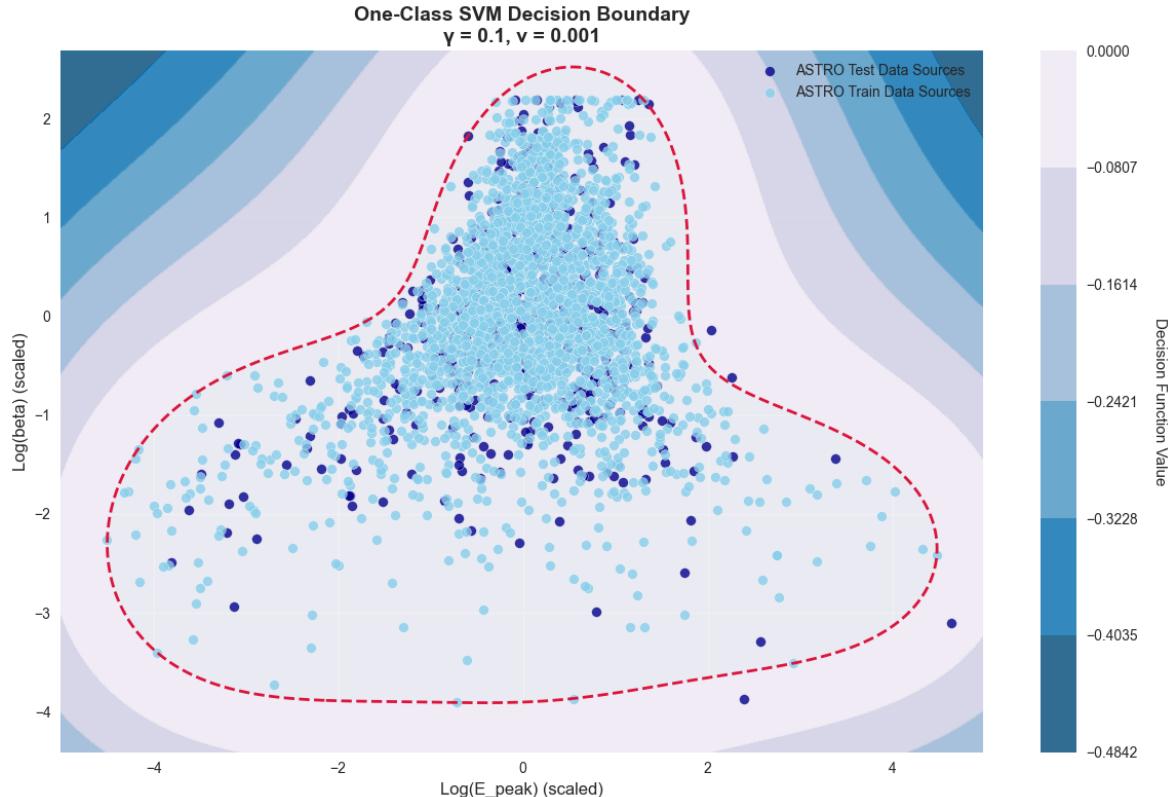


Figura 5.2. Frontera de decisión del modelo OCSVM en el espacio $\text{Log}(E_{\text{peak}})$ vs $\text{Log}(\beta)$

La visualización de la frontera de decisión (figura 5.2) revela que el modelo ha aprendido exitosamente la distribución espectral típica de fuentes astrofísicas. La frontera no circular captura la correlación natural entre energía de pico y el parámetro β , reflejando procesos físicos subyacentes.

Análisis de métricas: El análisis de *decision scores* reveló media de 0.0281 y desviación estándar de 0.0139, indicando distribución concentrada que evidencia consistencia en clasificaciones. Los pocos valores negativos corresponden a anomalías detectadas,

situándose apenas por debajo del umbral (análisis estadístico completo en el Apéndice VI).

En el Apéndice IX se incluye ejemplos de cómo varía la hiperfrontera del modelo OCSVM 2F y el porcentaje de outliers para casos de underfitting (figura 14) y overfitting (figura 15).

5.2.4. Modelo OCSVM con 4 características (4F)

Como evolución del modelo bidimensional, se desarrolló una versión ampliada incorporando cuatro características espectrales: Log(E_peak), Log(beta), Log(sigma) y Log(beta_Rel). Esta extensión tiene como objetivo incrementar el poder discriminatorio mediante información espectral adicional que pueda revelar firmas más sutiles de potenciales señales de materia oscura.

Consulta del código asociado

La implementación del modelo OCSVM 4F se encuentra en el apéndice VIII. Los aspectos específicos incluyen la selección extendida de características (listado 5) y el ajuste de hiperparámetros para mayor dimensionalidad (listado 6).

Optimización de hiperparámetros para alta dimensionalidad

La transición al espacio tetradimensional requirió reajuste cuidadoso de hiperparámetros:

Parámetro Gamma ($\gamma = 0.02$)

Se redujo significativamente el valor de gamma de 0.1 (modelo 2D) a 0.02 (modelo 4D), fundamentándose en principios de ML para espacios de alta dimensionalidad:

- **Curse of dimensionality:** En espacios de mayor dimensión, los datos tienden a dispersarse, requiriendo kernels menos restrictivos
- **Prevención de overfitting:** Un γ menor permite mayor generalización evitando fronteras excesivamente complejas
- **Coherencia astrofísica:** Mantiene capacidad para capturar correlaciones naturales entre las cuatro variables espirales

Parámetro Nu ($\nu = 0.002$)

Se ajustó ligeramente a $\nu = 0.002$ (frente a 0.001 en modelo 2D), permitiendo margen ligeramente mayor de tolerancia a outliers, reconociendo que la mayor dimensionalidad puede introducir variabilidad natural adicional.

Resultados del modelo 4F

El modelo tetradimensional demostró rendimiento excepcional y superior al modelo 2D:

- **Precisión global:** 99.29 % (563 de 567 muestras correctamente clasificadas)
- **Outliers detectados:** 4 fuentes (0.71 % vs 0.88 % del modelo 2D)
- **Reducción relativa:** 20 % menos falsos positivos que el modelo 2D

Los *decision scores* exhibieron una distribución notablemente diferente: media de 0.3185 (vs 0.0281 en 2D) y mayor dispersión ($\text{std}=0.0876$), indicando mayor confianza discriminatoria pero también mayor complejidad del espacio 4D.

Visualización multidimensional

Para abordar la limitación de visualizar fronteras 4D, se implementó una estrategia de proyecciones multidimensionales:

- **Proyecciones bidimensionales:** Se generaron las 6 combinaciones posibles de pares de variables, fijando dimensiones restantes en valores medios. Esta aproximación preserva la integridad del modelo 4D mientras proporciona interpretabilidad física de correlaciones específicas entre pares de variables espectrales (figura 16).
- **Proyecciones tridimensionales:** Se implementaron visualizaciones 3D para las 4 combinaciones de tripletas de variables, proporcionando mayor contexto dimensional y validación cruzada visual de outliers (Ap. X - figura 17).

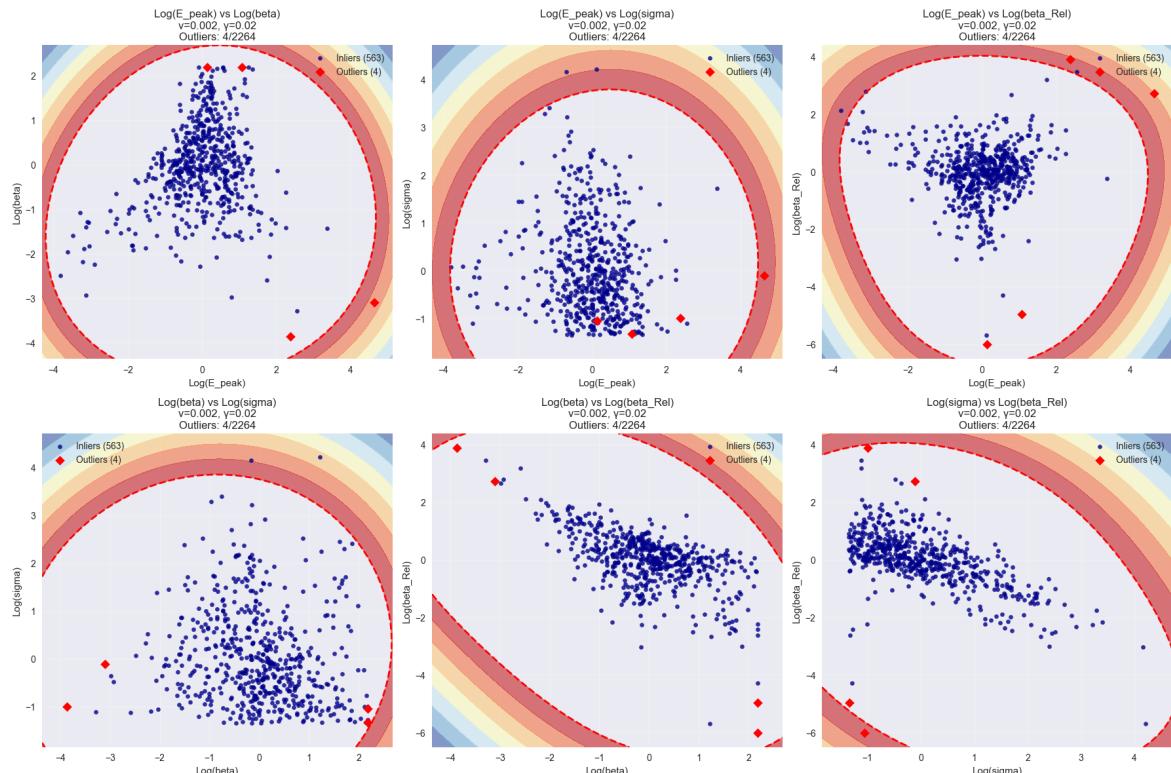


Figura 5.3. Conjunto completo de proyecciones bidimensionales del modelo OneClassSVM 4F. Se muestran las 6 combinaciones posibles de pares de características. Las regiones coloreadas indican los niveles de la función de decisión, mientras que la línea discontinua roja marca la frontera de clasificación.

Los outliers identificados (puntos rojos en lafigura 16) aparecen consistentemente como anómalos en múltiples proyecciones, validando su clasificación como anomalías genuinas independientes de la dimensión específica visualizada.

Ganancia por dimensionalidad adicional

Comparación modelo 2D vs 4D:

- **Reducción de falsos positivos:** De 5 a 4 outliers (reducción del 20 %)
- **Mayor confianza estadística:** Decision scores con mayor separación
- **Información complementaria:** Log(sigma) y Log(beta_Rel) proporcionan contexto físico adicional sobre forma espectral completa

El modelo 4D representa mejora metodológica significativa: sacrifica simplicidad visual de la frontera 2D a cambio de representación más completa y robusta del espacio espectral astrofísico, resultando en detección más precisa aunque metodológicamente más compleja.

Al igual que para el modelo anterior, se incluye en el Apéndice XI ejemplos de la variación de la hiperfrontera y los outliers durante el entrenamiento del modelo OCSVM 4F para casos de underfitting ([figura 19](#)) y overfitting ([figura 20](#)).

5.3. Aplicación a fuentes no identificadas

La aplicación de los modelos OCSVM entrenados a las 1,125 fuentes no identificadas (UNIDs) del catálogo Fermi-LAT representa la fase de implementación práctica donde los algoritmos de ML desarrollados se aplican al problema real de detección de candidatos a materia oscura.

5.3.1. Aplicación a fuentes UNID con OCSVM 2F

La aplicación del modelo OCSVM bidimensional a las fuentes UNIDs constituye el momento donde la metodología de ML se convierte en herramienta de descubrimiento científico. Estas fuentes UNIDs constituyen el conjunto de datos objetivo, ya que no han podido ser asociadas con fuentes astrofísicas conocidas mediante métodos tradicionales.

Implementación técnica

Los datos de las 1,125 fuentes UNIDs se procesaron manteniendo coherencia metodológica con el pipeline de entrenamiento.

Aspecto crítico de implementación: Se aplicó únicamente `transform()` usando el escalado preentrenado, manteniendo la integridad del sistema de normalización y evitando cualquier filtración de información del conjunto objetivo.

Consulta del código asociado

El código completo para la aplicación del modelo OCSVM 2F a fuentes UNID puede consultarse en el Apéndice apéndice XII. La implementación específica del sistema de ranking se detalla en el listado 11, mientras que el algoritmo general del proceso se describe en el algoritmo 2.

El sistema implementado genera múltiples métricas de evaluación:

- **Decision scores:** Distancias computadas por la función de decisión del
- **Predicciones binarias:** Clasificación basada en umbral (inlier: +1, outlier: -1)

- **Anomaly scores:** Transformación 'anom_scores = -decision_scores' para interpretación intuitiva
- **Anomaly rankings:** Normalización a percentiles [0-100 %] mediante MinMaxScaler

Resultados de clasificación

El modelo implementado detectó un número reducido de anomalías:

- **Total de fuentes procesadas:** 1,125 UNIDs
- **Outliers detectados:** 4 fuentes (0.4 %)
- **Inliers detectados:** 1,121 fuentes (99.6 %)

La tasa de detección del 0.4 % es inferior a la del conjunto de test (0.88 %), indicando comportamiento **consistente y conservador** del algoritmo en datos independientes.

Análisis estadístico del modelo

Se puede consultar un análisis estadístico completo del modelo aplicado en el Apéndice VI.

Caracterización de candidatos detectados

Los 4 candidatos más anómalos identificados presentan características distintivas:

ID UNID	Log(E_peak)	Log(beta)	SVM Score	Anomaly Rank (%)
1054	3.47	-1.04	-0.0366	100.0
275	-2.28	-0.86	-0.0095	99.9
1116	0.77	-2.62	-0.0057	99.8
1017	1.68	~0.00	-0.0038	99.7

Tabla 5.2. Candidatos principales detectados por el modelo OCSVM 2F

Interpretación técnica: Los candidatos muestran valores extremos en diferentes dimensiones del espacio de características:

- **UNID 1054:** Valor máximo de Log(E_peak) en el dataset
- **UNID 275:** Valor mínimo de Log(E_peak)
- **UNID 1116:** Valor mínimo de Log(beta)
- **UNID 1017:** Combinación atípica de parámetros espectrales

Validación visual del modelo

La visualización (figura 5.4) confirma que los 4 *outliers* se posicionan consistentemente fuera de la región de normalidad definida durante el entrenamiento, mientras que la distribución de *inliers* sigue patrones esperados dentro de la frontera establecida.

Evaluación del rendimiento del sistema

El sistema desarrollado logra:

- **Reducción dramática del espacio de búsqueda:** De 1,125 a 4 candidatos (99.6 % de reducción)

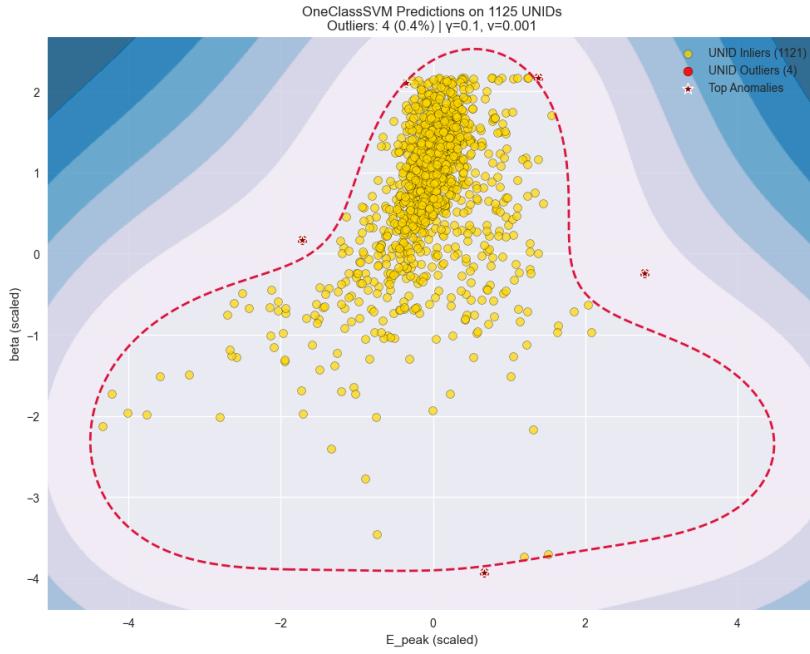


Figura 5.4. Aplicación de la frontera de decisión OCSVM 2F sobre fuentes UNIDs. Los 4 outliers se posicionan consistentemente fuera de la región de normalidad.

- **Alta precisión en la clasificación:** Solo 0.4 % de potenciales falsos positivos
- **Ranking discriminativo:** Separación clara entre percentiles de normalidad y anomalías
- **Robustez algorítmica:** Generalización efectiva y performance consistente en datos independientes

Los 4 candidatos identificados constituyen un conjunto altamente refinado para análisis de seguimiento, representando una reducción del 99.6 % en el espacio de búsqueda mientras mantiene alta confianza en la calidad de la selección.

5.3.2. Aplicación a fuentes UNID con OCSVM 4F

La extensión del modelo OneClassSVM a cuatro características [Log(E_peak), Log(beta), Log(sigma), Log(beta_Rel)] se aplicó al mismo conjunto de 1,125 fuentes UNIDs manteniendo coherencia metodológica con el *pipeline* 2F.

Implementación técnica

El preprocesado siguió el protocolo establecido:

Consulta del código asociado

La aplicación del modelo OCSVM 4F a fuentes UNID se documenta en el apéndice XIII, destacando las diferencias respecto al modelo 2F (apéndice XII). La implementación específica incluye extracción de características 4D (listado 12) y análisis estadístico (listado 14).

El modelo 4F optimizado ($\gamma = 0.02$, $\nu = 0.002$) se aplicó para generar *decision scores*, predicciones binarias y rankings de anomalía siguiendo el mismo sistema de métricas implementado en el modelo 2F.

Resultados de clasificación

El modelo tetradimensional detectó un número ligeramente superior de anomalías:

- **Total de fuentes procesadas:** 1,125 UNIDs
- **Outliers detectados:** 5 fuentes (0.4 %)
- **Inliers detectados:** 1,120 fuentes (99.6 %)

La tasa de detección del 0.4 % es idéntica al modelo 2F, pero con una anomalía adicional detectada (5 vs 4), sugiriendo que las características adicionales proporcionan información discriminatoria complementaria.

Análisis estadístico comparativo

Se puede consultar un análisis estadístico completo del modelo 4F así como comparativo de los modelos 2F y 4F en el Apéndice VI.

Caracterización de candidatos 4F

Los Top 5 candidatos más anómalos identificados por el modelo 4F:

ID	$\text{Log}(E_{\text{peak}})$	$\text{Log}(\beta)$	$\text{Log}(\sigma)$	$\text{Log}(\beta_{\text{Rel}})$	SVM	Rank (%)
307	1.51	0.00	0.62	-3.05	-0.1769	100.0
285	-0.75	-1.25	0.66	1.87	-0.0559	79.8
166	1.33	0.00	0.64	-2.42	-0.0337	76.2
923	-0.22	-0.25	0.63	0.74	-0.0328	76.0
1116	0.77	-2.62	0.75	1.60	-0.0018	70.8

Tabla 5.3. Top 5 candidatos detectados por el modelo OCSVM 4F

Análisis técnico de los candidatos

- **UNID 307:** Score de anomalía más extremo (-0.1769)
- **UNID 285:** Combinación de energía de pico baja con $\text{Log}(\beta_{\text{Rel}})$ positivo alto (1.87)
- **UNID 1116:** Único candidato común con modelo 2F, confirmando su robustez como anomalía genuina independiente de la dimensionalidad del modelo
- **UNIDs 166 y 923:** Parámetros espectrales con combinaciones específicas que generan alta anomalía en el espacio 4D

Análisis visual multidimensional

Las proyecciones en espacios de mayor dimensionalidad ofrecen una perspectiva más rica sobre la distribución de los datos y la separación de anomalías:

- **Proyecciones con $\text{Log}(\sigma)$ y $\text{Log}(\beta_{\text{Rel}})$:** revelan correlaciones adicionales que no eran evidentes en el modelo bidimensional (2F).
- **Consistencia interdimensional:** los candidatos anómalos se mantienen como outliers a lo largo de múltiples subespacios proyectados, lo que refuerza su validez.

Las visualizaciones tetradimensionales correspondientes al modelo OCSVM 4F, aplicado a las fuentes UNIDs, pueden consultarse en el Apéndice XIV. En dichas representaciones se observan claramente los cinco outliers identificados.

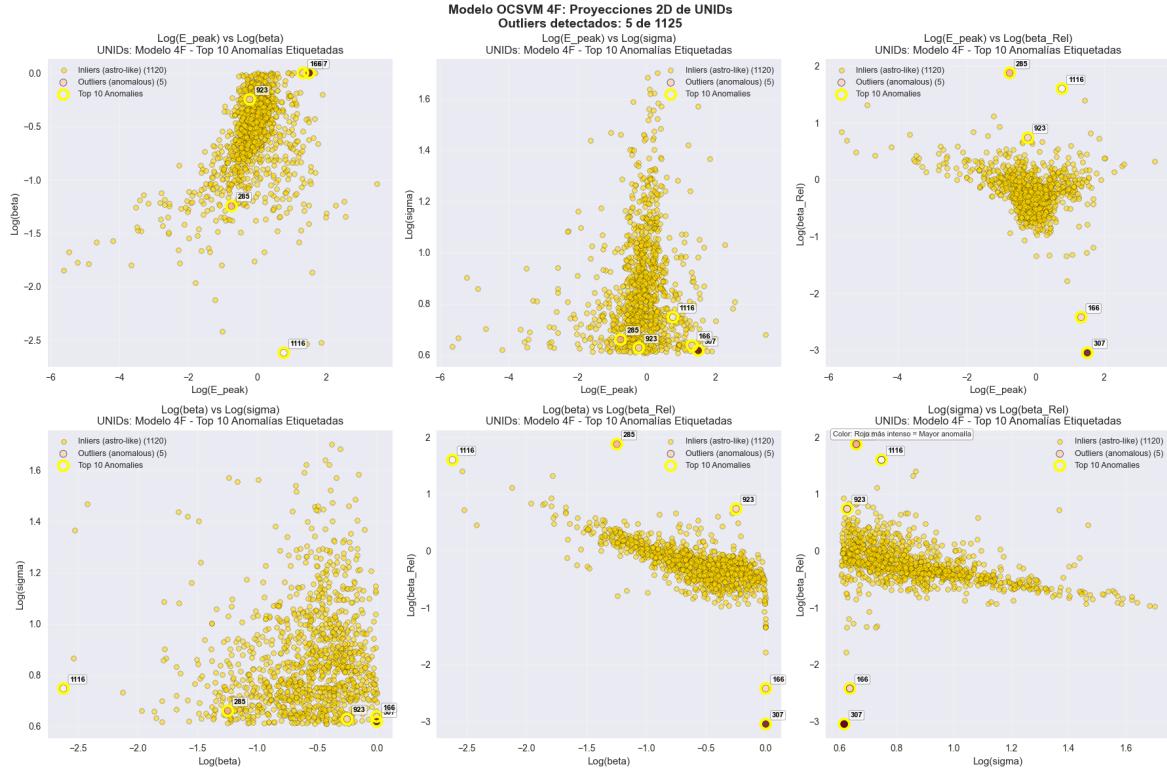


Figura 5.5. Las 6 proyecciones bidimensionales de las predicciones hechas por el modelo OCSVM 4F sobre las UNIDs

5.3.3. Análisis de consenso entre modelos

Para evaluar la robustez de los enfoques 2F y 4F OCSVM, se implementó un análisis sistemático de consenso entre modelos. Esta metodología permite identificar candidatos con diferentes niveles de confianza basados en la convergencia de las predicciones.

Clasificación por consenso

Los candidatos se clasificaron según su detección simultánea por ambos modelos versus aquellos identificados de manera exclusiva:

- **Consenso:** Detectados por ambos modelos (2F y 4F) - 1 candidato
- **Modelo-específicos:** Detectados únicamente por un modelo - 7 candidatos

Esta distribución resultó en un total de 8 candidatos únicos, con el UNID 1116 como único caso de consenso robusto, como queda ilustrado en el diagrama de Venn de la figura 5.6 .

Comparación técnica 2F vs 4F

Métrica	Modelo 2F	Modelo 4F
Outliers detectados	4 (0.4 %)	5 (0.4 %)
Decision scores media	0.034	0.249
Candidatos únicos	3 (75 %)	4 (80 %)

Análisis Comparativo	Resultado
Candidatos comunes	1 (UNID 1116)
Total candidatos únicos	8
Overlap	12.5 %
Detecciones únicas	87.5 %

Tabla 5.4. Rendimiento cuantitativo: OCSVM 2F vs 4F

Validación del enfoque multidimensional

El análisis revela la efectividad del sistema 4F mediante tres indicadores clave:

- **Valor diferencial:** Los 4 candidatos exclusivos del modelo 4F (UNIDs 307, 285, 166, 923) demuestran que la información espectral adicional captura anomalías no detectables en 2D
- **Robustez validada:** El candidato UNID 1116, detectado consistentemente por ambos modelos, representa una anomalía independiente de la dimensionalidad del espacio de características
- **Capacidad discriminatoria:** Las características Log(sigma) y Log(beta_Rel) aportan información genuinamente complementaria, evidenciada por la tasa de detección sostenida (0.4 %) con alta especificidad

5.3.4. Caracterización y ranking de candidatos

Se desarrolló un sistema de clasificación multicriterio donde el Anomaly Rank representa el percentil de anomalía dentro del conjunto de datos completo.

Sistema de priorización por tiers

Tier	UNID	Modelo	Anomaly Rank (%)	Características distintivas
1	1116	2F	99.8	Consenso robusto; $\text{Log}(\beta) = -2.62$
		4F	70.8	
2	307	4F	100.0	Anomalía máxima en espacio 4D
2	1054	2F	100.0	$\text{Log}(E_{\text{peak}})$ extremo = 3.47
3	275	2F	100.0	$\text{Log}(E_{\text{peak}})$ mínimo del conjunto
3	1017	2F	99.7	Configuración paramétrica atípica
3	285	4F	79.8	$\text{Log}(E_{\text{peak}})$ bajo con $\text{Log}(\beta_{\text{Rel}})$ alto
3	166	4F	76.2	Combinación espectral inusual
3	923	4F	76.0	Combinación espectral inusual

Tabla 5.5. Clasificación por *tiers* de candidatos según prioridad científica

Los criterios de estratificación priorizan: (1) candidatos de consenso por máxima confianza, (2) anomalías extremas modelo-específicas por alta especificidad, y (3) desviaciones moderadas pero consistentes.



Figura 5.6. Diagrama de Venn de los candidatos detectados por OCSVM 2F y OCSVM 4F. En la región izquierda se muestran los UNIDs únicos de 2F (1054, 275, 1017), en la región derecha los únicos de 4F (307, 285, 166, 923) y en la intersección el UNID común (1116).

5.4. Análisis comparativo ANN vs OCSVM

Este análisis cuantifica la complementariedad entre los enfoques supervisados (ANN) y semi-supervisados (OCSVM), evaluando la efectividad relativa y las sinergias potenciales del framework híbrido desarrollado.

5.4.1. Metodología de selección expandida

Para asegurar una comparación equilibrada, se amplió la selección más allá de los umbrales de detección específicos:

- **Modelos ANN:** Top 10 candidatos con mayor probabilidad de materia oscura por configuración dimensional
- **Modelos OCSVM:** Top 10 candidatos con mayor *anomaly score* (valores más negativos) por configuración dimensional

Esta metodología maximiza el potencial de identificar patrones de convergencia algorítmica al considerar el espectro completo de candidatos rankeados.

5.4.2. Diferencias metodológicas fundamentales

Aspecto	ANN (Supervisado)	OCSVM (Semi-supervisado)
Datos de entrenamiento	Fuentes ASTRO + DM simuladas (balanceadas)	Solo fuentes ASTRO (clase normal)
Paradigma de detección	Discriminación entre patrones ASTRO/DM	Modelado de normalidad y detección de outliers
Criterio de anomalía	Reconocimiento de patrones específicos	Desviación estadística distributiva
Métrica de ranking	Probabilidad de clase DM [0, 1]	Decision score (distancia al hipoplano)

Tabla 5.6. Paradigmas metodológicos contrastantes entre ANN y OCSVM

5.4.3. Resultados de complementariedad algorítmica

Métricas de convergencia

El análisis reveló una divergencia total entre los rankings de ambos algoritmos:

Comparación	ANN Top 10	OCSVM Top 10	Overlap	Candidatos únicos
2F	10	10	0 (0.0 %)	20
4F	10	10	0 (0.0 %)	20
Completo	20	20	0 (0.0 %)	40

Tabla 5.7. Complementariedad total entre rankings ANN y OCSVM

Interpretación de la divergencia algorítmica

La ausencia completa de solapamiento (0 %) (figura 5.7) confirma que ambos enfoques operan en regímenes de detección fundamentalmente distintos:

- **OCSVM:** Identifica desviaciones estadísticas respecto a la distribución astrofísica normal
- **ANN:** Reconoce patrones estructurales compatibles con simulaciones teóricas de materia oscura

Esta divergencia total valida la hipótesis de independencia metodológica y demuestra la ausencia de redundancia algorítmica en el framework híbrido.

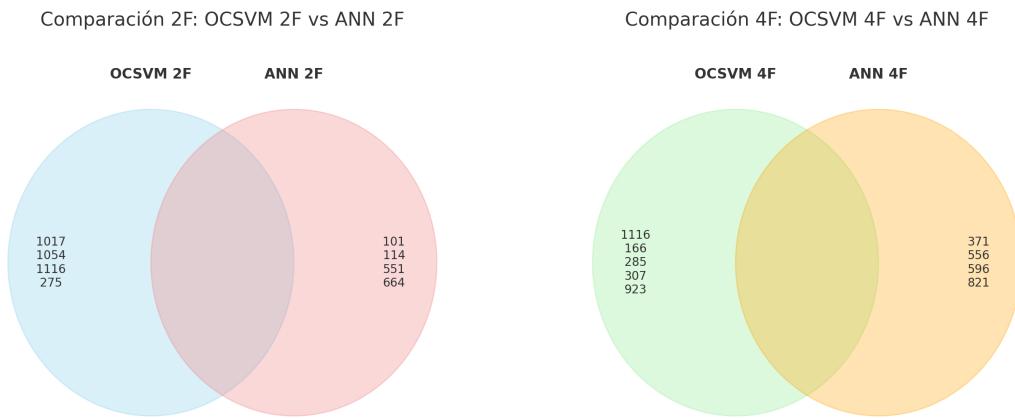


Figura 5.7. Izquierda: comparación de candidatos OCSVM 2F vs ANN 2F. Derecha: comparación de candidatos OCSVM 4F vs ANN 4F.

Capítulo 6

Otros experimentos adicionales

6.1. Implementación de OCSVM con datos DR4

Durante la fase final del desarrollo del proyecto, se obtuvo acceso al **cuarto data release (DR4)** del telescopio espacial Fermi-LAT, una versión significativamente más refinada respecto a publicaciones anteriores. Este conjunto de datos actualizados representó una excelente oportunidad para validar la metodología de detección de anomalías desarrollada y evaluar su rendimiento sobre datos reales con mayor calidad y resolución.

Características del conjunto DR4

El dataset proporcionado por la colaboración Fermi-LAT incluía un total de **5 068 fuentes gamma**, organizadas en dos grandes grupos:

- **Fuentes asociadas (assoc):** 3 784 fuentes con identificación astrofísica conocida, utilizadas para el entrenamiento del modelo.
- **Fuentes no asociadas (unas):** 1 284 fuentes sin correspondencia identificada, constituyendo el objetivo principal del análisis.

Las variables utilizadas para el análisis corresponden a **características espectrales físicas directamente interpretables**, sin necesidad de transformación logarítmica:

- **Alpha (α)** – pendiente espectral.
- **Beta (β)** – curvatura espectral.
- **Flux** – flujo gamma observado (integrado).

Estas características ofrecen una base sólida para un análisis interpretable y riguroso en términos físicos, permitiendo identificar candidatos de forma coherente con las teorías astrofísicas actuales.

Preprocesamiento y estrategia experimental

Dado que los datos contienen valores extremos propios del dominio astrofísico, se optó por el uso de `RobustScaler` en lugar de `StandardScaler`, con el fin de preservar la estructura estadística relevante. El conjunto de datos fue dividido en las siguientes proporciones:

- **Entrenamiento:** 60 % (2 270 muestras)

- **Validación:** 20 % (757 muestras)
- **Test:** 20 % (757 muestras)

Para optimizar el modelo, se realizó una **búsqueda de hiperparámetros** explorando distintas parejas de valores de ν y γ . El criterio de selección equilibró la **proporción de outliers detectados en validación con la distribución de los decision scores**.

Finalmente, la combinación óptima encontrada fue:

- $\nu = 0.002$
- $\gamma = 0.02$

Resultados en el conjunto de test

Con los valores de ν y γ seleccionados, el modelo OCSVM se entrenó sobre el 80 % de las fuentes asociadas (entrenamiento + validación) y se evaluó en el 20 % restante (757 muestras). Los resultados en este conjunto de prueba fueron:

- **Inliers detectados:** 755 (99.74 %)
- **Outliers detectados:** 2 (0.26 %)
- **Decision score promedio:** 0.4107 ± 0.1136
- **Decision score mínimo:** -0.2879
- **Decision score máximo:** 0.6346

Tabla 6.1. Resultados del OCSVM en el conjunto de test (ASOC).

Métrica	Valor
Tamaño del conjunto (ASOC test)	757
Inliers	755 (99.74 %)
Outliers	2 (0.26 %)
Score promedio	0.4107
Score mínimo	-0.2879
Score máximo	0.6346

Las visualizaciones que ilustran la frontera de decisión del OCSVM y la clasificación final de las muestras de test se muestran en el Apéndice XVI.1.

6.2. Aplicación a fuentes no asociadas

El modelo ajustado sobre todas las fuentes asociadas se aplicó posteriormente al conjunto de **1 284 fuentes no asociadas** (UNAS). Los resultados obtenidos fueron:

- **Fuentes clasificadas como normales (inliers):** 1 275 (99.30 %)
- **Candidatos anómalos (outliers):** 9 (0.70 %)

Las visualizaciones correspondientes a la aplicación sobre UNAS se encuentran en el Apéndice XVI.3.

Tabla 6.2. Métricas de anomalía en el conjunto de fuentes no asociadas (UNAS).

Métrica	Valor
Tamaño del conjunto (UNAS)	1 284
Inliers	1 275 (99.30 %)
Outliers (candidatos DM)	9 (0.70 %)
Score promedio	0.4246
Score (desvío estándar)	0.1469
Score más anómalo (mínimo)	-0.5816
Score menos anómalo (máximo)	0.6415

Comparación de experimentos

En la Tabla 6.3, se actualiza el resumen comparativo de los experimentos OCSVM realizados hasta la fecha, indicando el número de características empleadas, la cantidad de candidatos detectados y la tasa resultante.

Tabla 6.3. Resumen comparativo de experimentos OCSVM desarrollados

Experimento	Nº de features	Candidatos	Tasa	Interpretabilidad
OCSVM 2F	2	4	0.4 %	
OCSVM 4F	4	5	0.4 %	
DR4 3F	3	9	0.7 %	

6.3. Conclusiones del experimento DR4

- Validación práctica de un framework robusto de detección de anomalías en datos reales.
- Optimización de hiperparámetros: $\nu = 0.002$, $\gamma = 0.02$, con excelente capacidad de generalización.
- Confirmación de escalabilidad a conjuntos de más de 4 000 muestras.

Capítulo 7

Conclusiones y líneas futuras

Este Trabajo de Fin de Grado ha desarrollado un sistema informático basado en un *framework* dual que combina técnicas de aprendizaje automático supervisado (ANN) y detección de anomalías semi-supervisada (OCSVM), aplicado al análisis de datos astronómicos para identificar posibles candidatos a materia oscura en fuentes no identificadas (UNIDs) del catálogo Fermi-LAT. La implementación del sistema ha demostrado la efectividad y complementariedad de ambas técnicas computacionales.

7.1. Principales contribuciones y resultados del sistema

El desarrollo del sistema informático ha permitido validar una arquitectura híbrida de análisis de datos astronómicos, estableciendo las siguientes contribuciones técnicas principales:

Se ha implementado un sistema que demuestra la **complementariedad absoluta** entre los módulos ANN y OCSVM, con un solapamiento nulo (0 %) en la identificación de candidatos anómalos. Este resultado del sistema, inicialmente inesperado dado que se esperaba cierta convergencia en las detecciones, revela aspectos fundamentales sobre el comportamiento de los algoritmos implementados. Mientras que el módulo ANN identifica patrones que se asemejan a simulaciones teóricas de materia oscura, el módulo OCSVM detecta registros que se desvían del comportamiento estadístico convencional. La **ausencia total de convergencia** en los resultados algorítmicos sugiere que los candidatos identificados por cada módulo corresponden a diferentes tipos de anomalías en los datos, validando la arquitectura dual del sistema.

La arquitectura escalable de los módulos OCSVM ha conseguido una **reducción del 99.3 % del espacio de búsqueda**, procesando eficientemente las 1115 fuentes no identificadas originales e identificando 8 candidatos únicos de alta prioridad. Esta optimización computacional representa un avance significativo en términos de eficiencia de procesamiento para futuros análisis de datos astronómicos.

El **registro UNID 1116** emerge como el candidato prioritario del análisis, siendo el único detectado consistentemente por ambos modelos OCSVM implementados (2F

y 4F). Su perfil de datos extremadamente atípico lo identifica como el objetivo más prometedor para futuros análisis.

Desde el punto de vista de ingeniería de sistemas, se ha establecido un **protocolo robusto y replicable** para la evaluación comparativa entre algoritmos supervisados y semi-supervisados aplicados a datasets astronómicos, demostrando que la transición de "reconocimiento de patrones específicos" a "detección de anomalías distributivas" produce resultados algorítmicos completamente independientes.

La arquitectura del sistema desarrollado posee **aplicabilidad directa** a otros catálogos de datos científicos de alta dimensionalidad, incluyendo futuros datasets del Cherenkov Telescope Array (CTA) y misiones espaciales de próxima generación. Desde una **perspectiva de ingeniería de software**, las metodologías implementadas son adaptables a otros dominios que requieren detección de patrones anómalos en datasets con clases minoritarias, incluyendo aplicaciones en ciberseguridad, análisis financiero y sistemas de diagnóstico automático.

7.2. Limitaciones técnicas y consideraciones del sistema

La **ausencia de validación externa directa** constituye la limitación fundamental del sistema desarrollado. Los candidatos identificados por los algoritmos requieren validación mediante sistemas externos o estudios complementarios para establecer definitivamente su relevancia científica.

La **dependencia del módulo ANN en datos de entrenamiento simulados** introduce sesgos algorítmicos hacia patrones específicos de los datos de entrenamiento, limitando potencialmente la capacidad de detección de patrones que no se ajusten a las simulaciones utilizadas durante la fase de entrenamiento del sistema.

El **desbalance de clases** en los datasets astronómicos reales presenta desafíos intrínsecos para cualquier sistema de machine learning aplicado a este dominio, especialmente considerando la escasez esperada de registros positivos genuinos en los datos de entrada.

7.3. Líneas futuras de desarrollo del sistema

Las líneas de desarrollo futuras del sistema se estructuran en torno a tres ejes principales:

Mejoras algorítmicas y arquitectónicas: Integración de arquitecturas de deep learning más avanzadas y desarrollo de técnicas de ensemble learning específicamente adaptadas al procesamiento de datasets científicos, manteniendo la complementariedad entre módulos supervisados y no supervisados del sistema.

Ampliación de fuentes de datos: Incorporación de información multidimensional y series temporales de variabilidad para mejorar la capacidad de caracterización de candidatos y la discriminación algorítmica entre diferentes tipos de anomalías en los datos.

Desarrollo de módulos de validación: Implementación de protocolos automáticos de validación y verificación específicos para los candidatos identificados por el sistema, incluyendo métricas de confianza basadas en la consistencia de detección algorítmica.

7.3.1. Reflexión final

Este trabajo demuestra que la implementación sistemática de arquitecturas complementarias de machine learning puede revelar comportamientos inesperados en el análisis de datasets científicos complejos. La complementariedad absoluta observada en los resultados algorítmicos, lejos de ser una limitación del sistema, valida la necesidad de diversidad metodológica en el diseño de sistemas de detección de anomalías.

El desarrollo de sistemas de análisis de datos científicos se beneficia significativamente de arquitecturas que reconozcan y aprovechen la complementariedad inherente entre diferentes aproximaciones algorítmicas. Los resultados obtenidos subrayan que el progreso en el análisis automático de datasets complejos puede requerir no solo algoritmos más sofisticados, sino también estrategias de sistema que maximicen la cobertura del espacio de análisis mediante criterios de detección independientes y complementarios.

Apéndices

I. Entregables

Este apéndice documenta los entregables técnicos generados durante el desarrollo del Trabajo de Fin de Grado, proporcionando la información necesaria para la reproducibilidad y evaluación del trabajo realizado.

I.1. Repositorio del proyecto

Todo el código fuente, documentación técnica y materiales complementarios se encuentran disponibles públicamente en:

Repositorio:

https://github.com/martacanirome4/DarkMatter_ML_TFG

El proyecto está organizado con la siguiente estructura de directorios:

- TFG_codigo_final/: Código principal y scripts del proyecto
 - data/: Datos intermedios y transformados
 - results/: Resultados de modelos OCSVM y ANN organizados por tipo
 - notebooks/: Jupyter Notebooks con implementaciones de modelos
 - OCSVM_*.ipynb: Modelos de detección de anomalías One-Class SVM
 - ANN_*.ipynb: Implementación y comparación con redes neuronales
- notebooks/: Jupyter notebooks exploratorios y experimentales iniciales
- docs/: Documentación del proyecto
- requirements.txt: Especificación del entorno reproducible

I.2. Herramientas y dependencias

El proyecto está desarrollado en Python y utiliza las siguientes librerías principales especificadas en requirements.txt:

- scikit-learn: Implementación de algoritmos de machine learning (One-Class SVM, ANN)
- pandas: Manipulación y análisis de datos del catálogo Fermi-LAT
- numpy: Operaciones numéricas y álgebra lineal
- matplotlib: Generación de gráficos y visualizaciones
- seaborn: Visualizaciones estadísticas avanzadas

I.3. Instrucciones de reproducibilidad

Para reproducir los experimentos y resultados del trabajo:

1. Configuración inicial:

```
git clone https://github.com/martacanirome4/DarkMatter_ML_TFG  
cd DarkMatter_ML_TFG  
pip install -r requirements.txt
```

2. Ejecución del código principal:

Navegar a la carpeta TFG_codigo_final/ que contiene:

- Los datos procesados en data/
- Los notebooks organizados por metodología (OCSVM, ANN)
- Los resultados almacenados en data/results/

3. Generación de resultados:

La ejecución completa regenera automáticamente todas las figuras, tablas y archivos de resultados utilizados en la memoria del TFG, almacenándolos en las carpetas correspondientes dentro de results/.

II. Planificación detallada del proyecto

A continuación, se presenta la planificación detallada del proyecto estructurada en bloques de trabajo (*EPICs*), organizados por sprints. Cada tabla recoge las tareas planificadas, sus identificadores, descripciones y dependencias.

EPIC 01 – Definición del Alcance y Organización del Proyecto

Objetivo: Definir el marco de trabajo, los objetivos generales y organizar el backlog inicial.

Sprint 1 (Semanas 1–2)

ID Tarea	Descripción	Dependencias
TSK-EP1-S1-01	Definir alcance y objetivos del proyecto	—
TSK-EP1-S1-02	Crear backlog inicial y priorizar tareas	TSK-EP1-S1-01
TSK-EP1-S1-03	Revisión exhaustiva de literatura sobre UNN	—
TSK-EP1-S1-04	Revisión de literatura sobre técnicas de detección de anomalías	—
TSK-EP1-S1-05	Redacción preliminar de introducción y marco teórico	TSK-EP1-S1-03, TSK-EP1-S1-04

EPIC 02 – Validación del Modelo Existente

Objetivo: Ejecutar, comprender y documentar exhaustivamente el modelo base UNN.

Sprint 2 (Semanas 3–4)

ID Tarea	Descripción	Dependencias
TSK-EP2-S2-01	Ejecutar y validar funcionamiento del código UNN	—
TSK-EP2-S2-02	Ánálisis detallado del código fuente	TSK-EP2-S2-01
TSK-EP2-S2-03	Documentación y comentarios de funciones clave	TSK-EP2-S2-02
TSK-EP2-S2-04	Documentar resultados y métricas del modelo UNN	TSK-EP2-S2-01
TSK-EP2-S2-05	Redactar metodología preliminar basada en hallazgos	TSK-EP2-S2-04

EPIC 03 – Desarrollo del Modelo de Detección de Anomalías

Objetivo: Diseñar, implementar, entrenar y optimizar el nuevo modelo de detección.

Sprint 3 (Semanas 5–6)

ID Tarea	Descripción	Dependencias
TSK-EP3-S3-01	Revisar y definir requisitos técnicos específicos	TSK-EP1-S1-01
TSK-EP3-S3-02	Diseñar arquitectura preliminar del modelo	TSK-EP3-S3-01
TSK-EP3-S3-03	Implementar estructura base del prototipo	TSK-EP3-S3-02
TSK-EP3-S3-04	Realizar experimentos exploratorios (spikes)	TSK-EP3-S3-02
TSK-EP3-S3-05	Redactar metodología (sección diseño del modelo)	TSK-EP3-S3-02

Sprint 4 (Semanas 7–8)

ID Tarea	Descripción	Dependencias
TSK-EP3-S4-01	Preparación y partición de conjuntos de datos	—
TSK-EP3-S4-02	Configurar entorno de experimentación y métricas	—
TSK-EP3-S4-03	Entrenar prototipo inicial del modelo	TSK-EP3-S4-01, TSK-EP3-S4-02
TSK-EP3-S4-04	Evaluuar y analizar resultados preliminares	TSK-EP3-S4-03
TSK-EP3-S4-05	Documentar resultados iniciales obtenidos	TSK-EP3-S4-04

Sprint 5 (Semanas 9–10)

ID Tarea	Descripción	Dependencias
TSK-EP3-S5-01	Optimización sistemática de hiperparámetros	TSK-EP3-S4-04
TSK-EP3-S5-02	Implementar validación cruzada estratificada	TSK-EP3-S5-01

ID Tarea	Descripción	Dependencias
TSK-EP3-S5-03	Evaluación final en conjunto de prueba independiente	TSK-EP3-S5-02
TSK-EP3-S5-04	Documentar mejoras implementadas y resultados finales	TSK-EP3-S5-03
TSK-EP3-S5-05	Revisión parcial de progreso con tutor académico	TSK-EP3-S5-03

EPIC 04 – Integración, Análisis Comparativo y Documentación Final

Objetivo: Integrar resultados, realizar análisis comparativo exhaustivo y finalizar documentación y presentación.

Sprint 6 (Semanas 11–12)

ID Tarea	Descripción	Dependencias
TSK-EP4-S6-01	Integrar resultados UNN y modelo propio desarrollado	TSK-EP2-S2-04, TSK-EP3-S5-03
TSK-EP4-S6-02	Realizar análisis comparativo preliminar detallado	TSK-EP4-S6-01
TSK-EP4-S6-03	Documentar proceso de integración y análisis	TSK-EP4-S6-02
TSK-EP4-S6-04	Revisión parcial con tutor (análisis comparativo)	TSK-EP4-S6-02

Sprint 7 (Semanas 13–14)

ID Tarea	Descripción	Dependencias
TSK-EP4-S7-01	Revisión integral y ajustes del borrador del informe	—
TSK-EP4-S7-02	Incorporar feedback del tutor y correcciones finales	TSK-EP4-S7-01

Sprint 8 (Semana 15)

ID Tarea	Descripción	Dependencias
TSK-EP4-S8-01	Revisión final del informe y preparación de presentación	—
TSK-EP4-S8-02	Pruebas de contingencia y ajustes de último momento	TSK-EP4-S8-01

Criterios de Aceptación

ID Tarea	Criterio de Aceptación
TSK-EP1-S1-01	Documento de alcance aprobado formalmente por el tutor, con objetivos generales claramente definidos y medibles.
TSK-EP2-S2-02	Código fuente completamente revisado con comentarios explicativos exhaustivos y verificación de ausencia de errores críticos.
TSK-EP3-S3-03	Prototipo ejecuta correctamente al menos tres casos de prueba representativos sin fallos.
TSK-EP3-S4-03	Modelo prototípico entrenado exitosamente con métricas básicas documentadas (precisión, recall, F1-score mínimo).
TSK-EP3-S5-02	Validación cruzada completada con reporte detallado de métricas y análisis de variabilidad.
TSK-EP4-S6-01	Resultados integrados en documento único con tablas comparativas, gráficas y análisis estadístico.
TSK-EP4-S7-02	Feedback del tutor aplicado satisfactoriamente a un mínimo del 90 % de los comentarios recibidos.

III. Replicación del Estudio con Redes Neuronales

Este apéndice presenta los resultados de la aplicación de modelos de redes neuronales artificiales (ANN) a los datos de fuentes no identificadas, replicando el enfoque de clasificación supervisada desarrollado en el estudio principal.

III.1. Distribución de Probabilidades del Modelo ANN 2F

El modelo de red neuronal con dos características genera probabilidades de clasificación que reflejan la confianza en la identificación de candidatos a materia oscura entre las fuentes no identificadas.

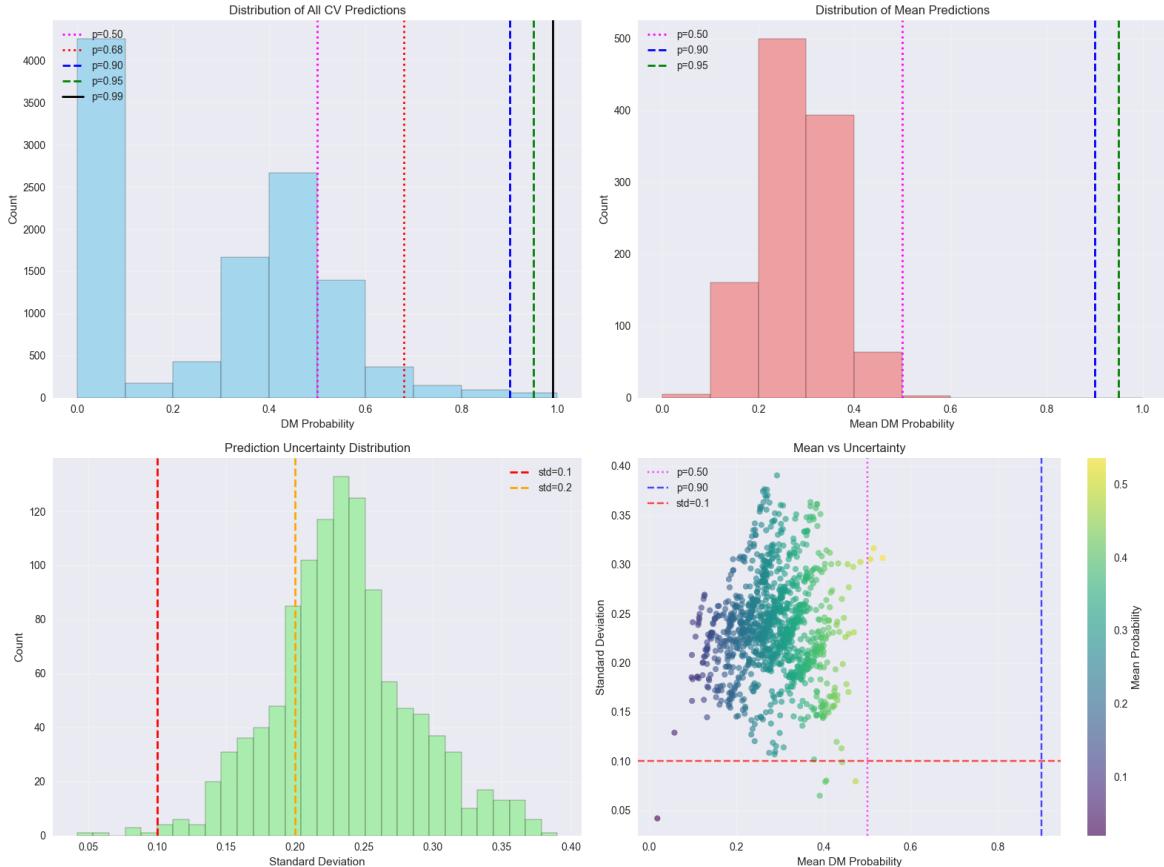


Figura 1. Distribución de probabilidades de clasificación del modelo ANN 2F sobre fuentes no identificadas. La concentración en valores bajos es coherente con la expectativa de escasez de señales genuinas de materia oscura.

III.2. Localización Espacial de Candidatos ANN 2F

La visualización en el espacio de características bidimensional revela los patrones de distribución espacial de los candidatos identificados por el modelo de red neuronal con mayor confianza.

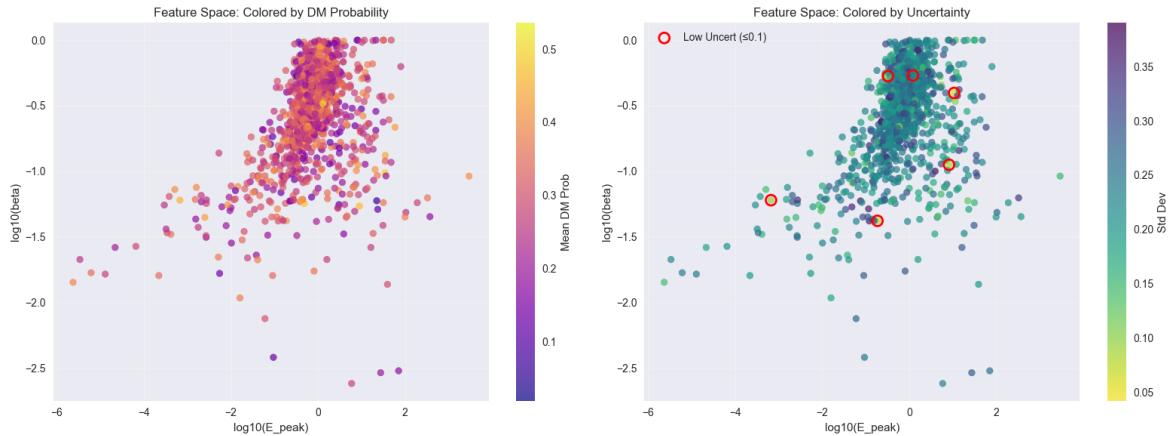


Figura 2. Localización de candidatos identificados por el modelo ANN 2F en el espacio de características $\log_{10}(E_{\text{peak}})$ vs $\log_{10}(\beta)$. Los candidatos con mayor probabilidad se concentran en regiones específicas del espacio de parámetros.

III.3. Comparación con el Modelo ANN 4F

El modelo extendido a cuatro características muestra un comportamiento diferente en la distribución espacial de candidatos, proporcionando una perspectiva complementaria del análisis de clasificación.

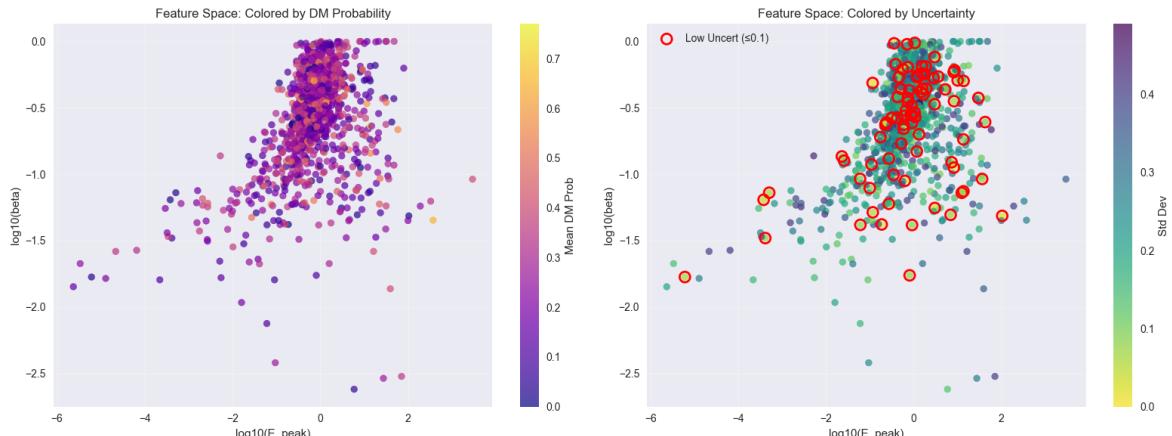


Figura 3. Distribución de candidatos identificados por el modelo ANN 4F en el espacio de características $\log_{10}(E_{\text{peak}})$ vs $\log_{10}(\beta)$. Se observa una distribución más uniforme comparada con la localización específica del modelo 2F.

IV. Análisis Exploratorio de Datos

Este apéndice presenta las visualizaciones detalladas del análisis exploratorio realizado sobre los datasets de fuentes astrofísicas (Astro) y materia oscura (DM) utilizados en el estudio.

IV.1. Distribuciones de Variables Espectrales

Las distribuciones individuales de las cuatro características espectrales ($\text{Log}(E_{\text{peak}})$, $\text{Log}(\beta)$, $\text{Log}(\sigma)$ y $\text{Log}(\beta_{\text{Rel}})$) revelan los patrones estadísticos fundamentales de cada tipo de fuente y permiten identificar diferencias en los rangos de valores y formas distributivas.

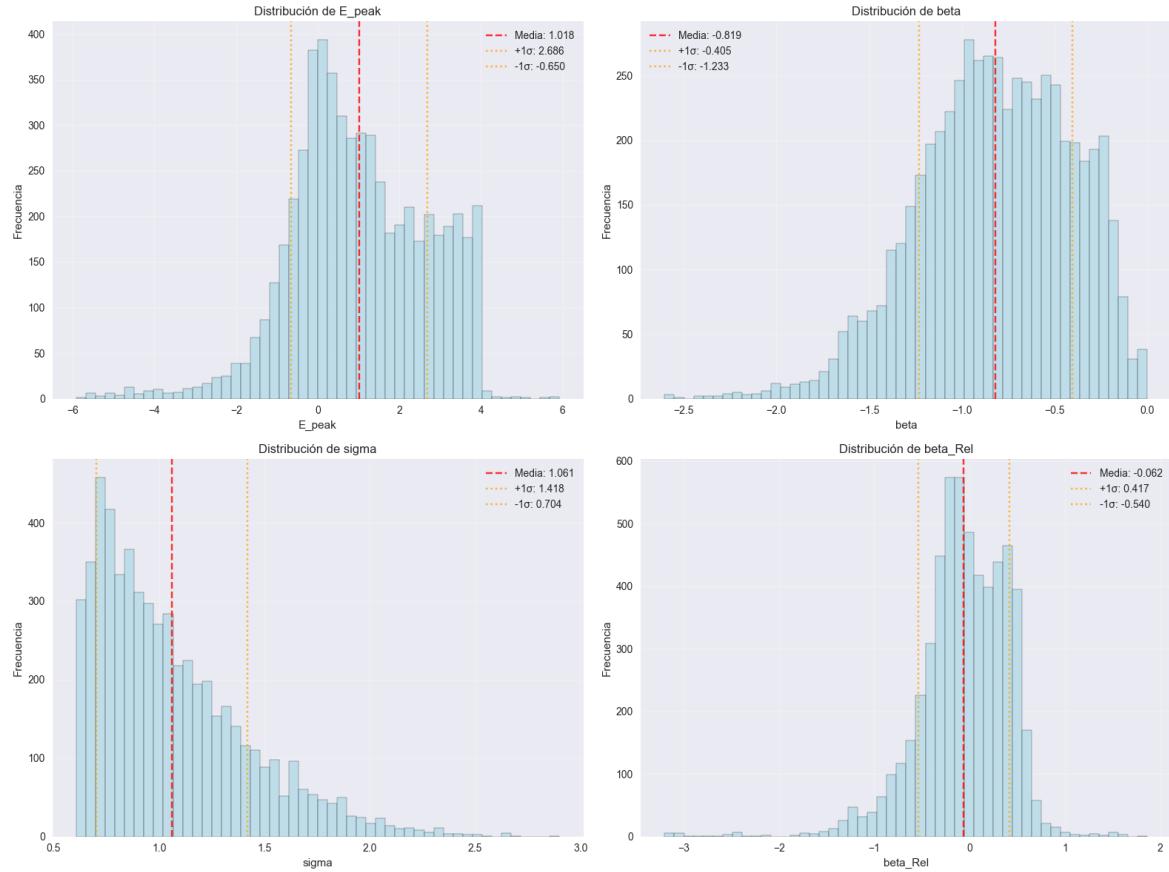


Figura 4. Distribuciones individuales de las cuatro variables espectrales para fuentes astrofísicas (Astro) y materia oscura (DM). Se muestran las diferencias en centralización, dispersión y forma distributiva entre ambos tipos de fuentes.

IV.2. Análisis de Correlaciones

La matriz de correlación cuantifica las relaciones lineales entre las variables espectrales, proporcionando insights sobre las dependencias intrínsecas en los datos y la redundancia potencial entre características.

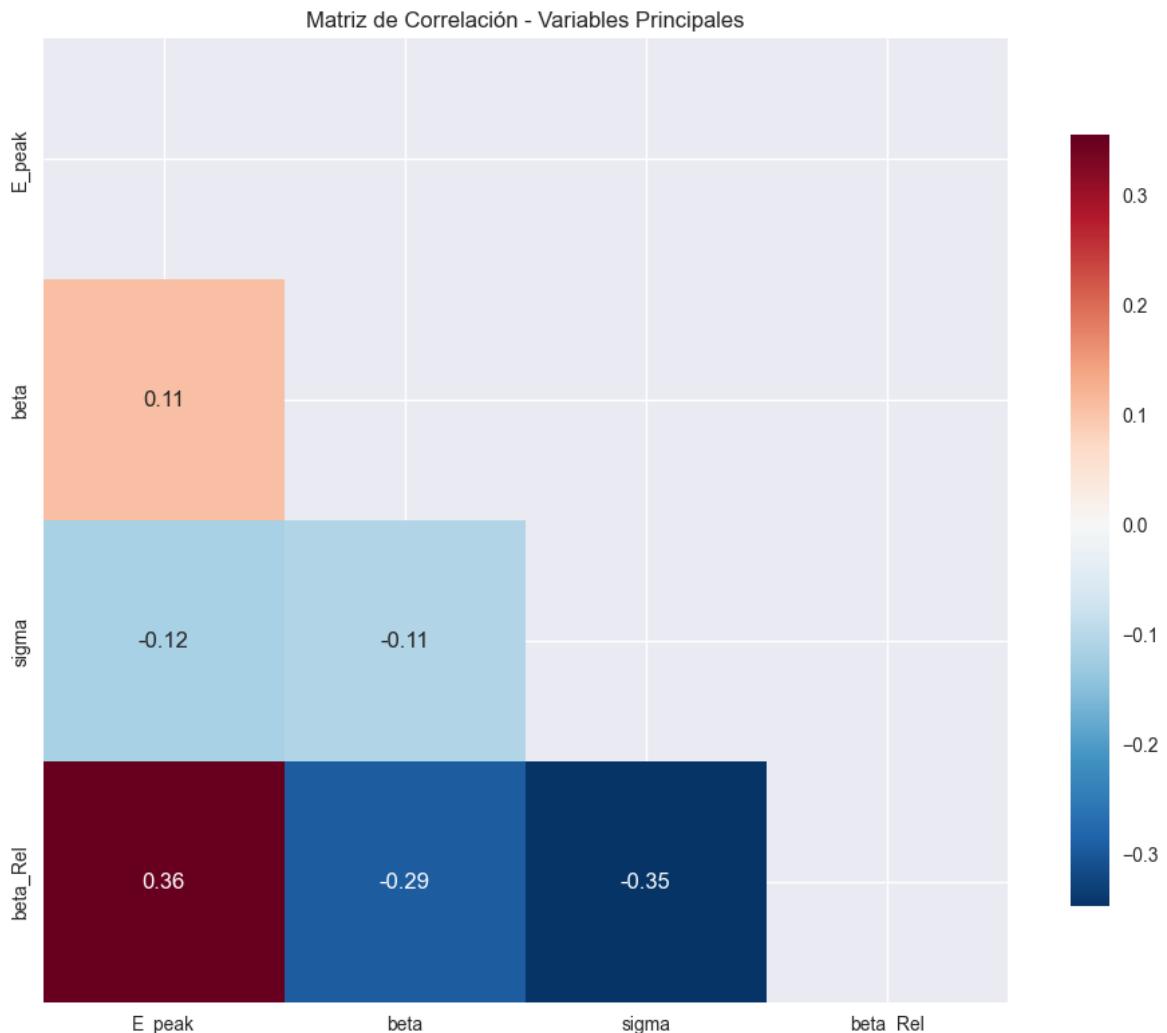


Figura 5. Matriz de correlación entre las variables espectrales de fuentes astrofísicas y materia oscura. Los valores numéricos y la escala de colores indican la fuerza y dirección de las correlaciones lineales entre pares de variables.

IV.3. Relaciones Multidimensionales

Los scatter plots multidimensionales permiten visualizar las relaciones no lineales entre variables y la separabilidad natural entre clases de fuentes en el espacio de características.

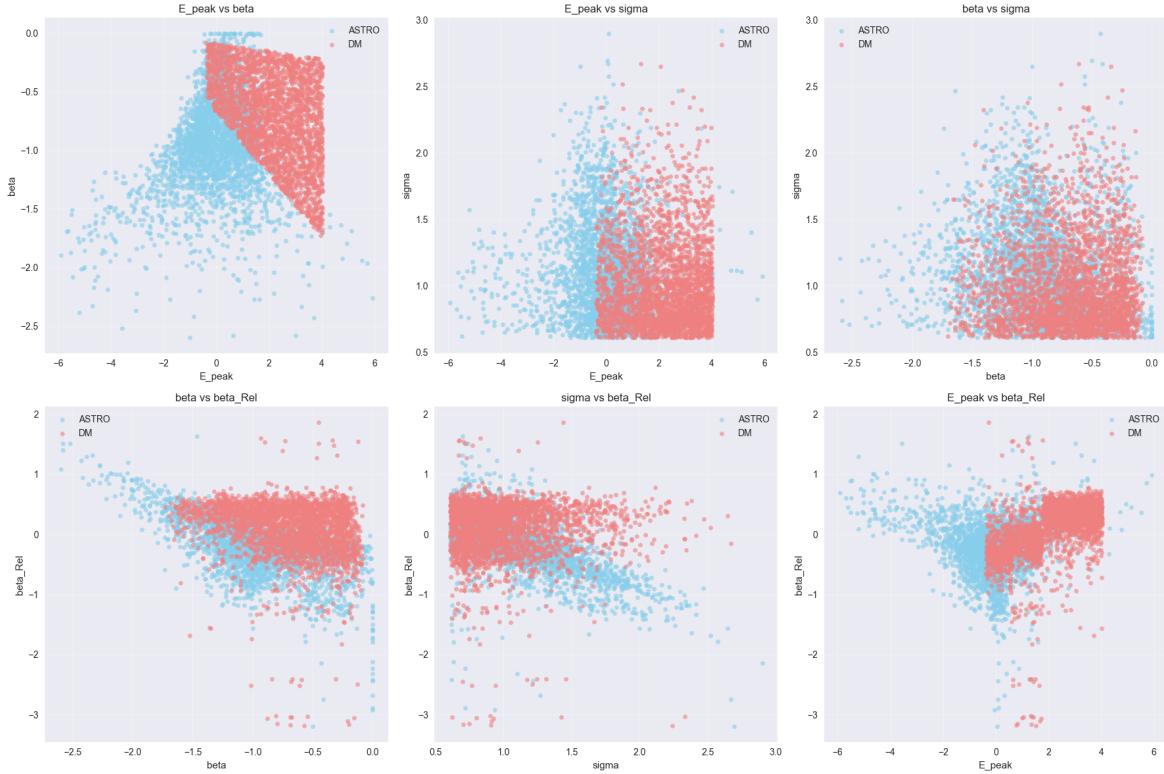


Figura 6. Matriz de dispersión multidimensional mostrando las relaciones por pares entre todas las variables espectrales. Los puntos de diferentes colores representan fuentes astrofísicas y de materia oscura, evidenciando patrones de agrupamiento y separabilidad en el espacio multidimensional.

V. Transformación del Dataset UNIDs

Este apéndice documenta el proceso de transformación logarítmica aplicado al dataset UNIDs para normalizar las distribuciones de las características espectrales y temporales antes de su procesamiento por los modelos de machine learning.

V.1. Distribuciones Iniciales de las Variables

Las variables del dataset UNIDs presentan distribuciones asimétricas con diferentes escalas. E_peak muestra una concentración en valores bajos, beta exhibe distribución aproximadamente uniforme, sigma_det sigue un patrón exponencial decreciente, y beta_Rel se concentra cerca de cero con algunos valores extremos.

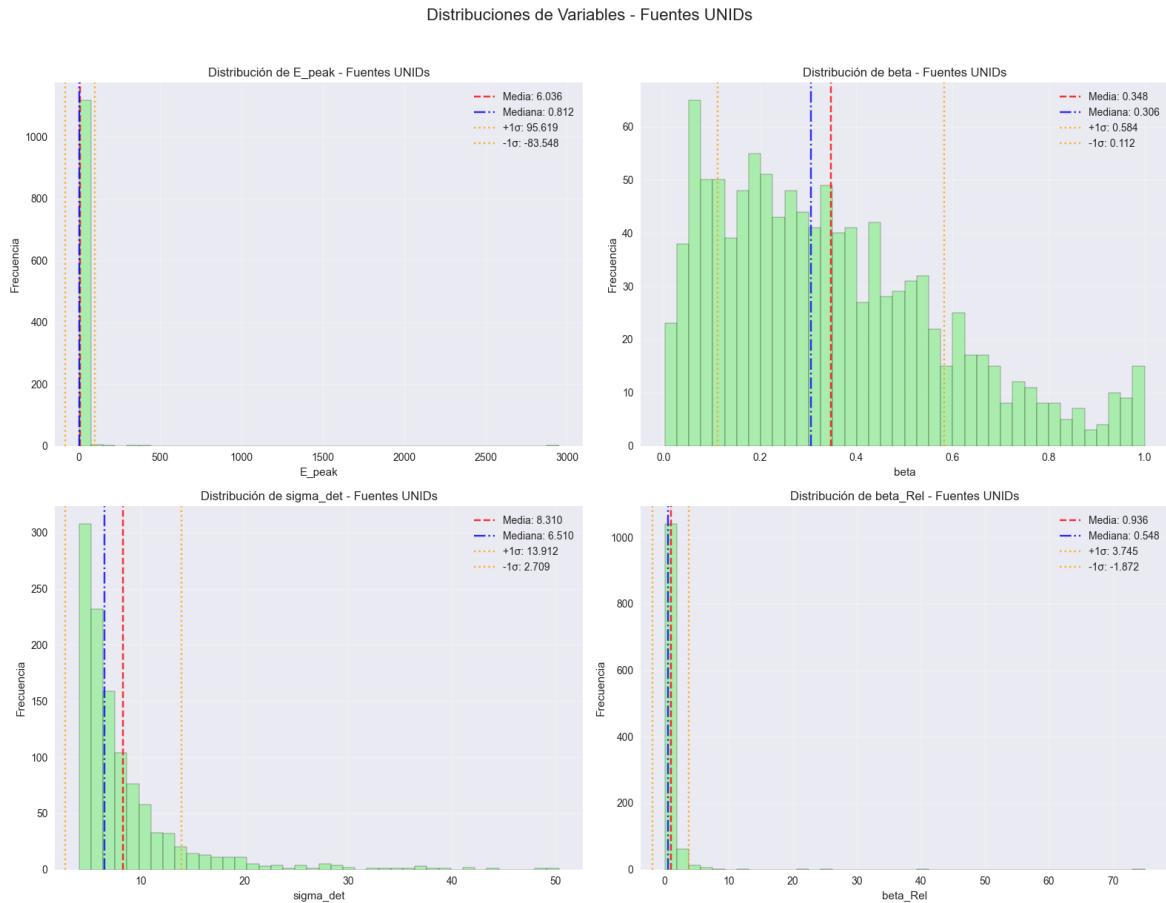


Figura 7. Distribuciones iniciales de las cuatro variables del dataset UNIDs: E_peak (energía pico), beta (índice espectral), sigma_det (significancia de detección) y beta_Rel (índice espectral relativista). Se observan distribuciones asimétricas y con diferentes escalas que requieren normalización.

V.2. Comparación Pre y Post Transformación

Los scatter plots multidimensionales permiten visualizar el impacto de la transformación logarítmica en las correlaciones entre variables y la mejora en la simetría de las distribuciones.

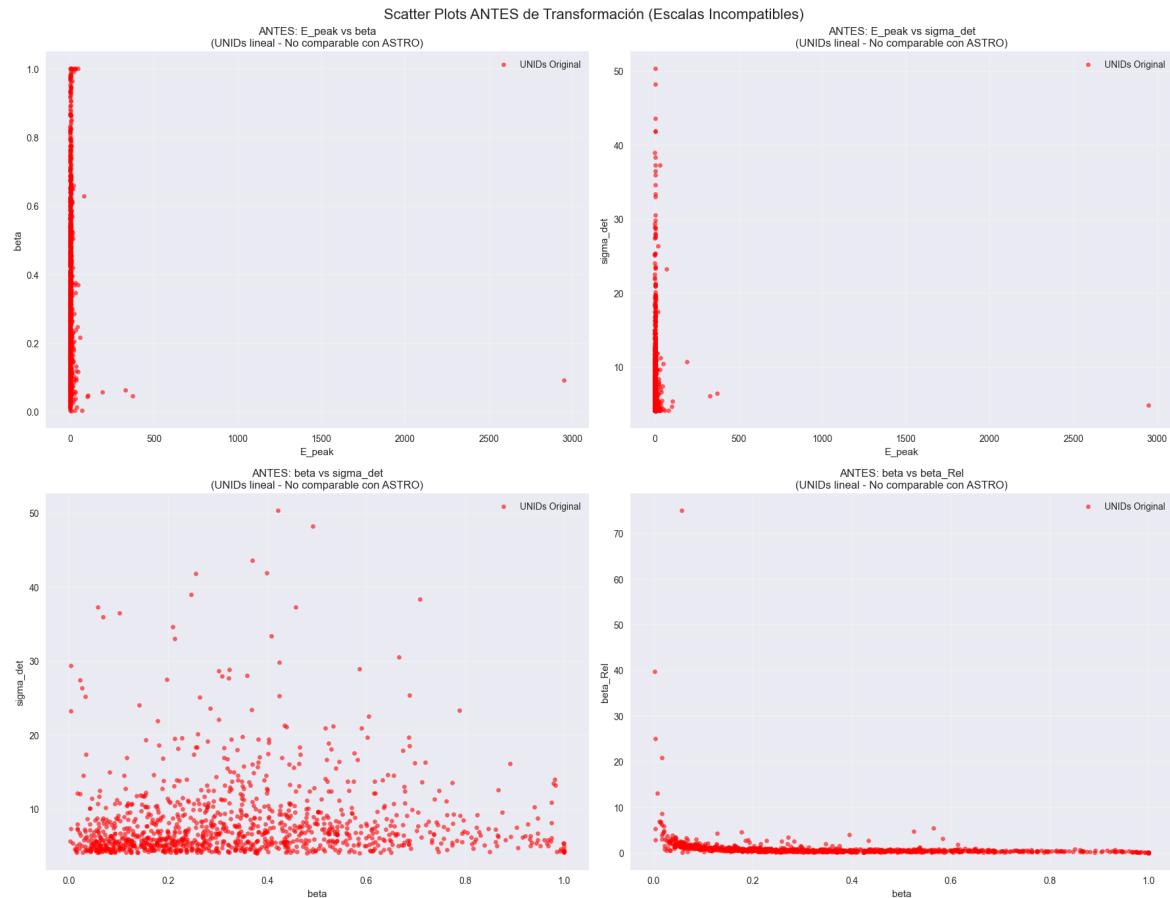


Figura 8. Matrices de scatter plots multidimensionales antes de la transformación logarítmica, mostrando las correlaciones originales entre variables con escalas heterogéneas.

Tras la aplicación de la transformación logarítmica, las distribuciones muestran mayor simetría y las correlaciones entre variables se vuelven más evidentes y lineales.

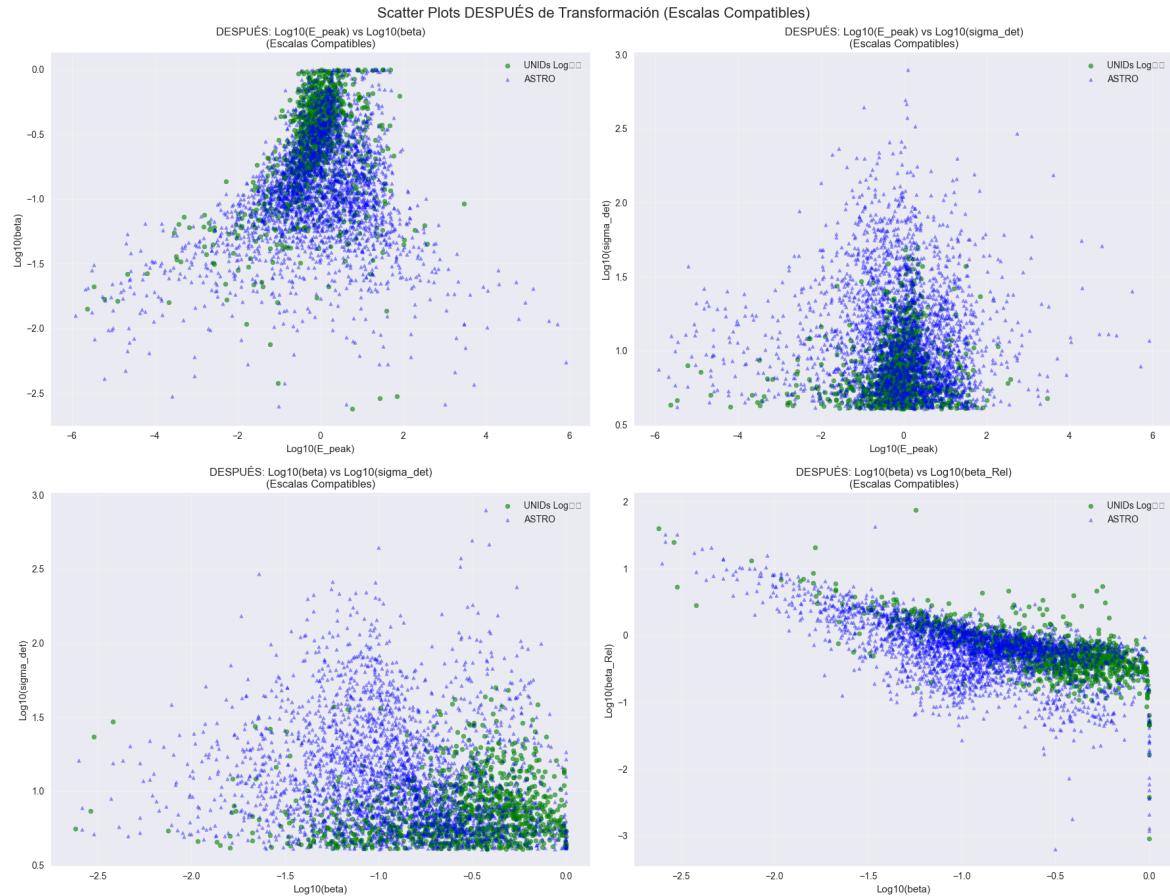


Figura 9. Matrices de scatter plots multidimensionales después de la transformación logarítmica, evidenciando la mejora en la simetría de las distribuciones y la clarificación de las correlaciones entre variables.

V.3. Comparación de Distribuciones de UNID vs ASTRO antes y después

La efectividad del proceso de normalización se evidencia en la comparación directa de las distribuciones, donde se aprecia la transformación de distribuciones asimétricas hacia formas más simétricas y apropiadas para el análisis estadístico.

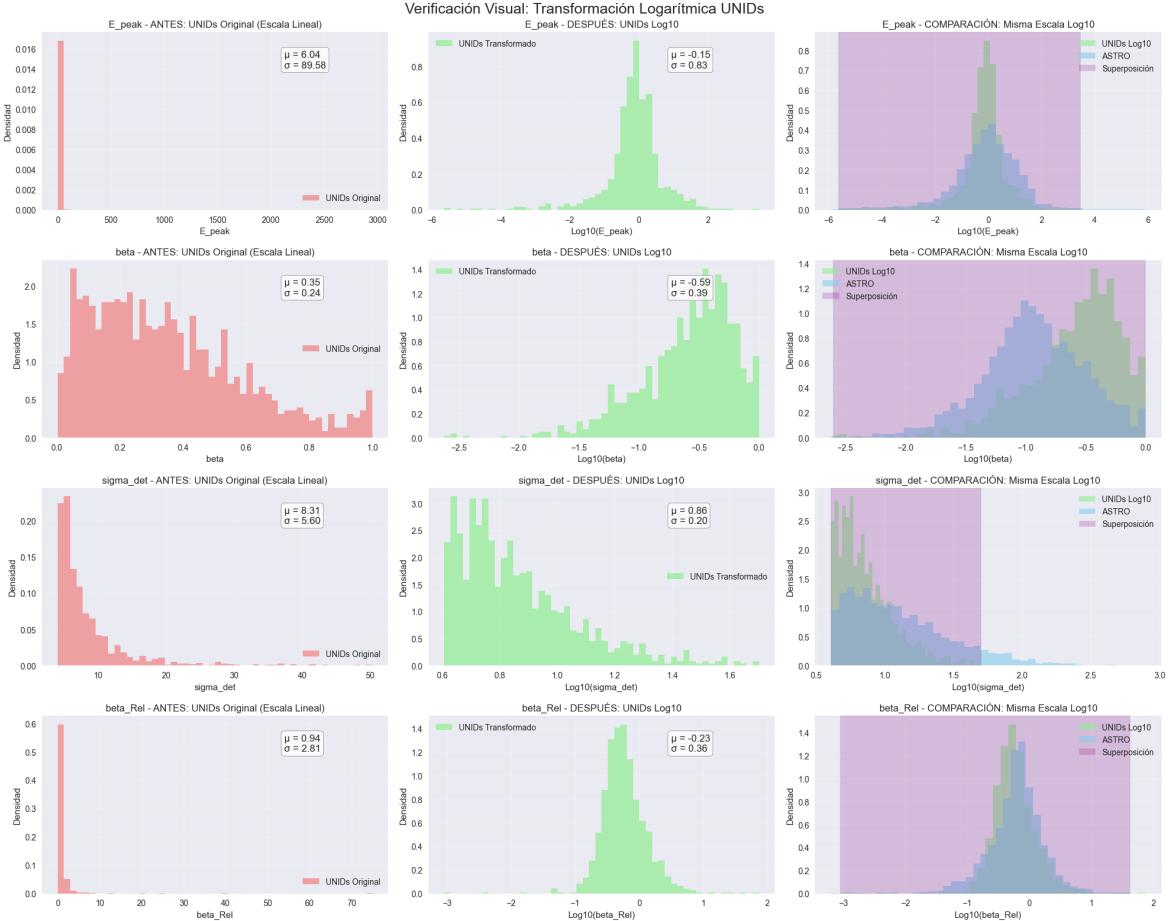


Figura 10. Comparación directa de las distribuciones antes (panel superior) y después (panel inferior) de la transformación logarítmica, demostrando la efectividad del proceso de normalización.

VI. Análisis estadístico detallado de modelos OCSVM

VI.1. Modelo OCSVM 2F - Métricas completas

Distribución de decision scores

- Media: 0.0281
- Desviación estándar: 0.0139
- Skewness: -0.542
- Kurtosis: 0.859
- Rango: [-0.0397, 0.0895]

Tests de significancia

- Test de Mann-Whitney: p-value = 1.51e-11
- Separación estadísticamente significativa entre inliers y outliers

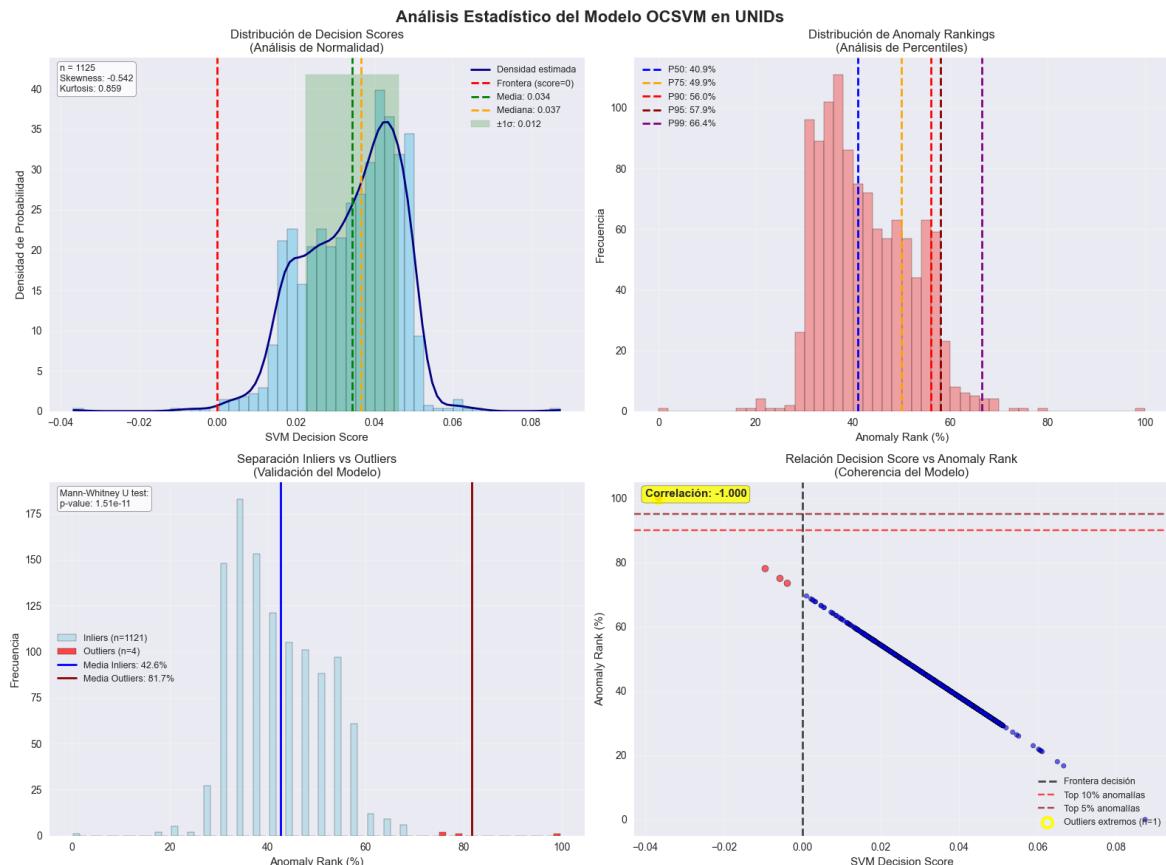


Figura 11. Análisis estadístico completo del modelo OCSVM 2F aplicado a fuentes UNIDs

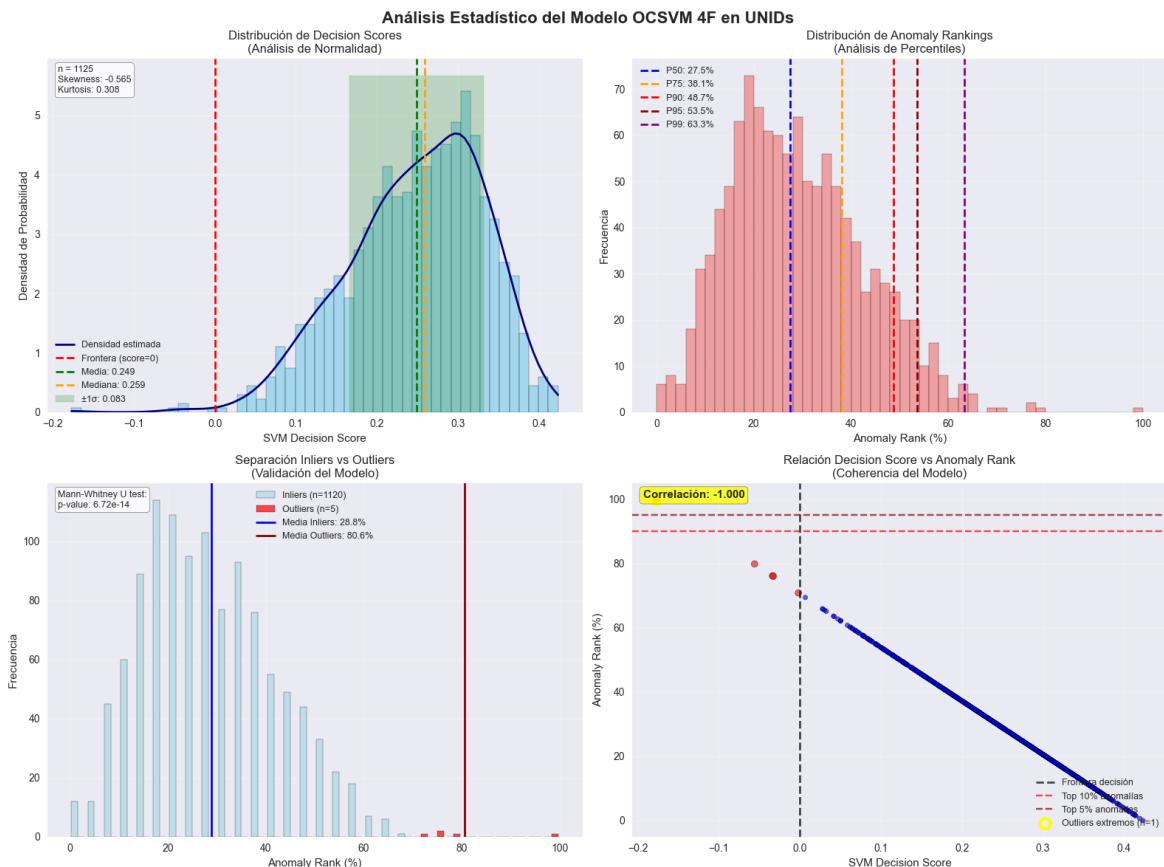
VI.2. Modelo OCSVM 4F - Métricas completas

Distribución de decision scores

- Media: 0.3185 ($6.3 \times$ mayor que modelo 2F)
- Desviación estandar: 0.0876
- Skewness: -0.565
- Kurtosis: 0.859
- Rango: [-0.1286, 0.4277]

Análisis de percentiles

- Modelo 2F: P90 = 56.0
- Modelo 4F: P90 = 48.7



VI.3. Modelos OCSVM 2F vs 4F - Análisis comparativo

El análisis estadístico completo del modelo 4F revela diferencias importantes respecto al modelo 2F:

Métrica	Modelo 2F	Modelo 4F	Interpretación
Decision scores media	0.034	0.249	$6.3 \times$ mayor confianza
Skewness	-0.542	-0.565	Sesgo negativo más pronunciado
P90 percentil	56.0 %	48.7 %	Distribución más concentrada
Mann-Whitney p-value	1.51e-11	6.72e-04	Separación estadísticamente significativa

Tabla 10. Comparación estadística entre modelos OCSVM 2F y 4F

VII. Modelo OCSVM 2F (código)

Implementación del modelo One-Class SVM con 2 características.

VII.1. Pipeline del modelo

Algorithm 1 Entrenamiento OCSVM 2F

Entrada: Dataset D con características $\{\text{Log}(E_{\text{peak}}), \text{Log}(\beta)\}$

Salida: Modelo OCSVM entrenado y evaluado

- 1: Cargar datos y seleccionar características $X \leftarrow D[\text{Log}(E_{\text{peak}}), \text{Log}(\beta)]$
 - 2: Dividir datos: X_{train} (60 %), X_{val} (20 %), X_{test} (20 %)
 - 3: Normalizar características con StandardScaler
 - 4: **Grid Search:**
 - 5: **para** $\nu \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2\}$ **hacer**
 - 6: **para** $\gamma \in \{0.1\}$ **hacer**
 - 7: Entrenar $\text{modelo}_{\text{OCSVM}}(\nu, \gamma)$ con X_{train}
 - 8: Evaluar en X_{val} y contar outliers
 - 9: Actualizar mejores parámetros si $\text{outliers} < \text{mejor}_{\text{outliers}}$
 - 10: **end para**
 - 11: **end para**
 - 12: Reentrenar modelo final con $X_{\text{train}} \cup X_{\text{val}}$
 - 13: Evaluar en X_{test} y generar matriz de confusión
 - 14: **return** Modelo final y métricas de evaluación
-

VII.2. Configuración de datos y características

```

1 import pandas as pd
2 import numpy as np
3 from sklearn.svm import OneClassSVM
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import confusion_matrix
7
8 # Cargar dataset
9 data_path = "../data/astro_data_with_labels.txt"
10 df_astro = pd.read_csv(data_path, sep='\s+')
11
12 # Selección de características principales
13 features = ['Log(E_peak)', 'Log(beta)']
14 X = df_astro[features].values
15
16 # División estratificada: 60% train, 20% val, 20% test
17 X_train, X_temp, y_train, y_temp = train_test_split(
18     X, y, test_size=0.4, random_state=42
19 )
20 X_val, X_test, y_val, y_test = train_test_split(
21     X_temp, y_temp, test_size=0.5, random_state=42
22 )

```

Listing 1. Carga y selección de características

VII.3. Preprocesamiento y normalización

```

1 # Escalado de características
2 scaler = StandardScaler()
3 X_train_scaled = scaler.fit_transform(X_train)
4 X_val_scaled = scaler.transform(X_val)
5 X_test_scaled = scaler.transform(X_test)

```

Listing 2. Normalización de características

VII.4. Búsqueda de hiperparámetros

La selección de hiperparámetros se basa en minimizar el número de outliers en el conjunto de validación, utilizando un enfoque conservador para evitar falsos positivos.

```

1 # Hiperparámetros a explorar
2 gamma_values = [0.1] # Justificación: balance óptimo basado en análisis previo
3 nu_values = [0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2]
4
5 # Tracking de resultados
6 results = []
7 best_outliers = np.inf
8 best_model = None
9 best_params = {}
10 best_mean_score = 0.0
11
12 # Grid Search
13 print("Buscando hiperparámetros que minimicen outliers en ASTRO (validación)...")
14 for nu in nu_values:
15     for gamma in gamma_values:
16         model = OneClassSVM(kernel='rbf', nu=nu, gamma=gamma)
17         model.fit(X_train_scaled)
18
19         # Calcular score de decisión para análisis más detallado
20         decision_scores = model.decision_function(X_val_scaled)
21         preds_val = model.predict(X_val_scaled)
22
23         # Predicciones: 1 = normal, -1 = outlier
24         preds_val = model.predict(X_val_scaled)
25         pred_labels = np.where(preds_val == 1, 0, 1) # Mapear a 0 = normal, 1 = anomalía
26         true_labels = y_val.astype(int)
27         n_outliers = np.sum(preds_val == -1)
28         mean_score = np.mean(decision_scores) # Distancia media al hiperplano
29
30         results.append({

```

```

31         'nu': nu,
32         'gamma': gamma,
33         'val_outliers': n_outliers,
34         'mean_decision_score': mean_score
35     })
36
37     # Actualizar mejor modelo
38     if n_outliers < best_outliers:
39         best_outliers = n_outliers
40         best_model = model
41         best_params = {'nu': nu, 'gamma': gamma}
42         best_mean_score = mean_score
43
44 # Mostrar resultado óptimo
45 print(f"\nMejor combinación de hiperparámetros:")
46 print(f" - nu = {best_params['nu']}")
47 print(f" - gamma = {best_params['gamma']}")
48 print(f" - Outliers (val set): {best_outliers} de {len(X_val_scaled)}
        } muestras")
49 print(f" - Mejor distancia media al hiperplano: {best_mean_score}")
50
51 # Crear DataFrame con resultados
52 df_results = pd.DataFrame(results)
53 display(df_results.sort_values(by='val_outliers'))
```

Listing 3. Grid Search para optimización de hiperparámetros

VII.5. Entrenamiento del modelo final

```

1 # Reentrenar con X_train + X_val
2 X_final_train = np.vstack([X_train, X_val])
3 y_final_train = np.concatenate([y_train, y_val])
4
5 scaler_final = StandardScaler()
6 X_final_train_scaled = scaler_final.fit_transform(X_final_train)
7 X_test_scaled = scaler_final.transform(X_test)
8
9 # Entrenar modelo final
10 final_model = OneClassSVM(kernel='rbf', gamma=best_params['gamma'], nu
    =best_params['nu'])
11 final_model.fit(X_final_train_scaled)
12
13 # Análisis de la distribución de scores
14 decision_scores_test = final_model.decision_function(X_test_scaled)
15
16 # Evaluar en test
17 test_preds = final_model.predict(X_test_scaled)
18 test_labels = np.where(test_preds == 1, 0, 1)
19 cm = confusion_matrix(y_test, test_labels)
20
21 print(f"Matriz de confusión:\n{cm}")
```

Listing 4. Entrenamiento del modelo final

VIII. Modelo OCSVM 4F (código)

Este modelo extiende el OCSVM 2F (apéndice VII) incorporando características espetrales adicionales del catálogo 4FGL-DR3. El pipeline general sigue el algoritmo 1 con las modificaciones específicas detalladas a continuación.

VIII.1. Configuración de características extendida

```

1 # Mismo dataset base que OCSVM 2F
2 data_path = "../data/astro_data_with_labels.txt"
3 df_astro = pd.read_csv(data_path, sep='\s+')
4
5 # Características extendidas para análisis espectral completo
6 features = ['Log(E_peak)', 'Log(beta)', 'Log(sigma)', 'Log(beta_Rel)']
7
8 # Selección dinámica de características
9 X = df_astro[features].values
10
11 print(f"Forma del dataset: {X.shape}")
12 print(f"Distribución de clases: {np.unique(y, return_counts=True)}")
13
14 # División idéntica al modelo 2F para comparabilidad
15 X_train, X_temp, y_train, y_temp = train_test_split(
16     X, y, test_size=0.4, random_state=42
17 )
18 X_val, X_test, y_val, y_test = train_test_split(
19     X_temp, y_temp, test_size=0.5, random_state=42
20 )

```

Listing 5. Selección de características 4D

VIII.2. Hiperparámetros optimizados para 4D

La selección de hiperparámetros se ajusta para el espacio de características expandido:

```

1 # Parámetros optimizados para espacio 4D
2 selected_gamma = 0.02 # Reducido respecto a 2F (0.1) para mayor
# dimensionalidad
3 selected_nu = 0.001 # Mantenido para consistencia en expectativa de
# outliers

```

Listing 6. Configuración de hiperparámetros para 4F

Nota sobre grid search: El código del repositorio del proyecto incluye una implementación completa de búsqueda en malla (comentada) que explora sistemáticamente el espacio de hiperparámetros. Los valores seleccionados resultan de este análisis previo.

VIII.3. Entrenamiento del modelo 4F

```

1 # Combinar train + val para entrenamiento final (igual que modelo 2F)
2 X_final_train = np.vstack([X_train, X_val])
3 y_final_train = np.concatenate([y_train, y_val])
4
5 # Normalización crítica para 4 dimensiones
6 scaler_final = StandardScaler()
7 X_final_train_scaled = scaler_final.fit_transform(X_final_train)
8
9 # Modelo final con hiperparámetros ajustados
10 final_model = OneClassSVM(kernel='rbf', gamma=selected_gamma, nu=
    selected_nu)
11 final_model.fit(X_final_train_scaled)
12
13 # Evaluación en conjunto de test
14 X_test_scaled = scaler_final.transform(X_test)
15 decision_scores_test = final_model.decision_function(X_test_scaled)
16
17 print(f"\nDecision scores estadísticas:")
18 print(f"  Media: {np.mean(decision_scores_test):.4f}")
19 print(f"  Std: {np.std(decision_scores_test):.4f}")
20 print(f"  Min: {np.min(decision_scores_test):.4f}")
21 print(f"  Max: {np.max(decision_scores_test):.4f}")

```

Listing 7. Entrenamiento con características extendidas

VIII.4. Evaluación y comparación

```

1 # Predicciones y métricas
2 test_preds = final_model.predict(X_test_scaled)
3 test_labels = np.where(test_preds == 1, 0, 1)
4
5 # Estadísticas de detección
6 n_inliers = np.sum(test_preds == 1)
7 n_outliers = np.sum(test_preds == -1)
8 total_samples = len(test_preds)
9
10 print(f"\nEvaluación en el conjunto de test:")
11 print(f"Total de muestras: {total_samples}")
12 print(f"Inliers detectados: {n_inliers} ({n_inliers/total_samples
    *100:.2f} %)")
13 print(f"Outliers detectados: {n_outliers} ({n_outliers/total_samples
    *100:.2f} %)")
14
15 # Matriz de confusión
16 cm = confusion_matrix(y_test, test_labels)
17 print(f"Matriz de confusión:\n{cm}")

```

Listing 8. Análisis de rendimiento 4F vs 2F

IX. Análisis de Overfitting y Underfitting en One-ClassSVM 2F

En este apéndice se presenta un análisis detallado de los fenómenos de overfitting y underfitting en el modelo OneClassSVM aplicado a la clasificación de fuentes astrofísicas utilizando las características Log(E_peak) y Log(beta).

IX.1. Configuración Óptima

Mediante validación cruzada y análisis de la estructura de los datos astrofísicos, se determinaron los parámetros óptimos:

$$\gamma = 0.1, \quad \nu = 0.001 \quad (1)$$

Esta configuración permite que el modelo capture la forma característica de los datos de entrenamiento sin sobreajustarse al ruido específico del conjunto de datos.

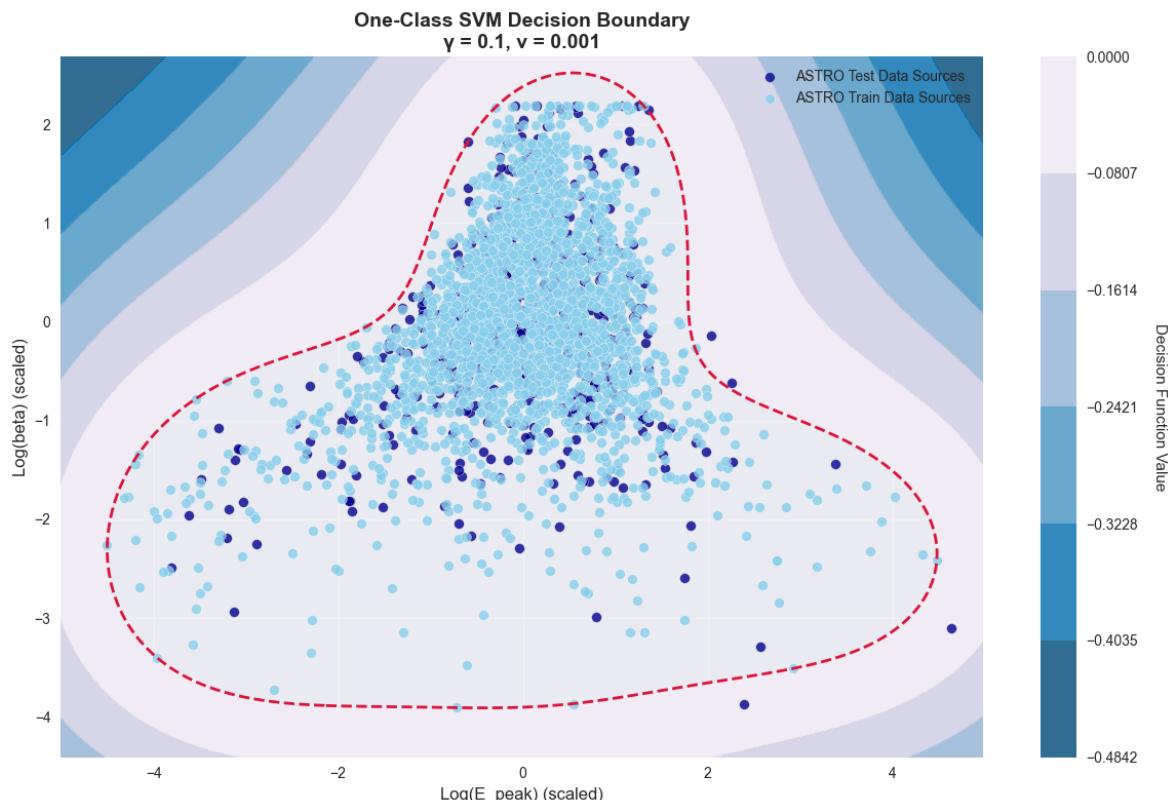


Figura 13. Frontera de decisión óptima del modelo OneClassSVM con $\gamma = 0.1$ y $\nu = 0.001$. La línea discontinua roja marca la frontera de decisión, mientras que las regiones azules representan zonas de normalidad con diferentes niveles de confianza.

IX.2. Underfitting

El underfitting ocurre cuando el modelo es demasiado simple para capturar la estructura subyacente de los datos. En OneClassSVM, esto se manifiesta típicamente con:

- Valores de γ muy bajos (kernel muy suave)
- Valores de ν muy altos (permitiendo muchos outliers)

Ejemplo de Underfitting Severo

$$\gamma = 0.001, \nu = 0.7 \quad (2)$$

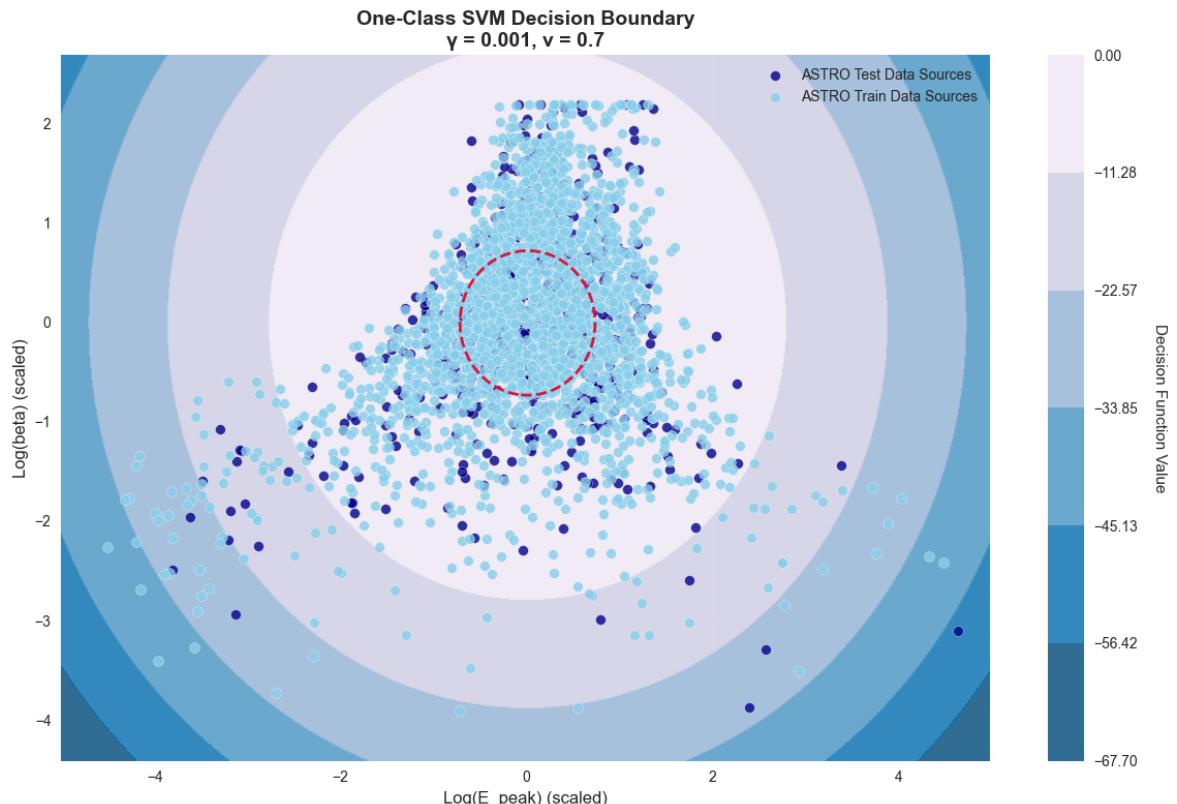


Figura 14. Ejemplo de underfitting severo. La frontera de decisión es extremadamente simplista y no logra distinguir entre la región de datos normales y el espacio circundante.

IX.3. Overfitting

El overfitting se produce cuando el modelo se ajusta excesivamente a los datos de entrenamiento, perdiendo capacidad de generalización. En OneClassSVM se caracteriza por:

- Valores de γ muy altos (kernel muy específico)
- Valores de ν muy bajos (muy pocos outliers permitidos)

Ejemplo de Overfitting Severo

$$\gamma = 10.0, \quad \nu = 0.0001 \quad (3)$$

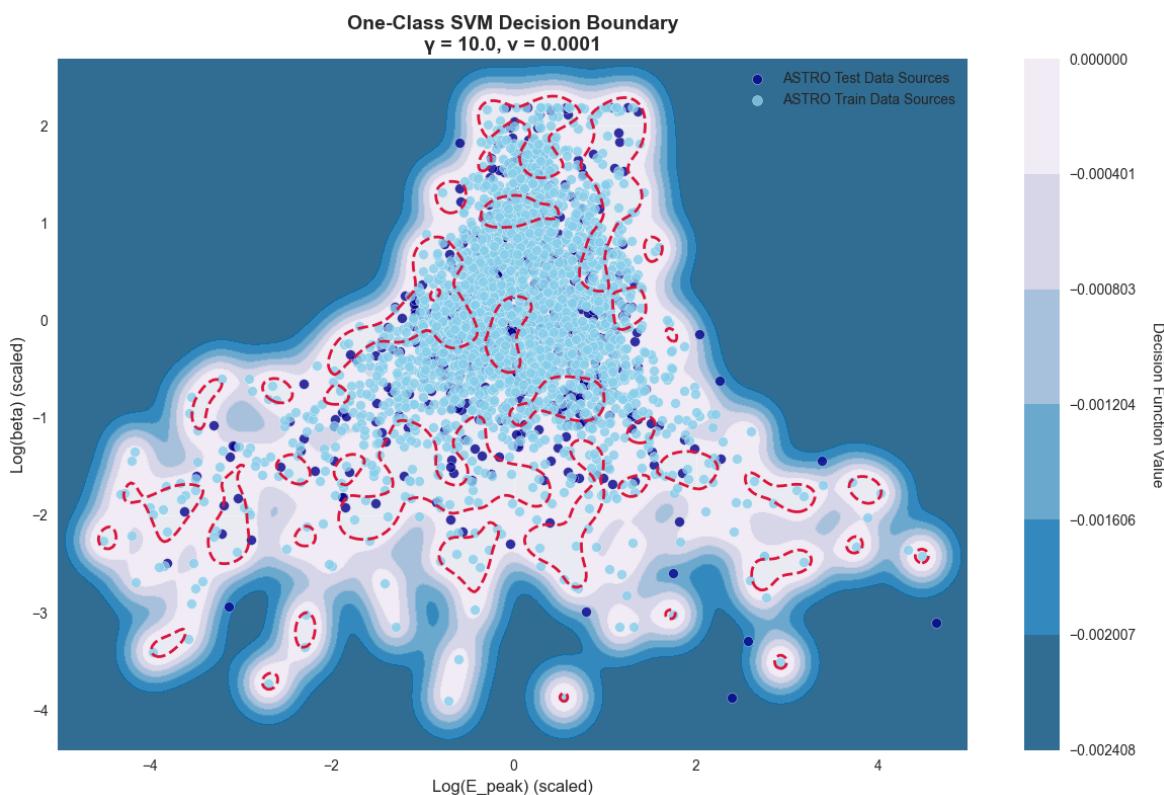


Figura 15. Ejemplo de overfitting severo. El modelo crea múltiples regiones desconectadas y fronteras extremadamente complejas, memorizando el ruido específico del conjunto de entrenamiento.

IX.4. Criterios de Evaluación

Para identificar el nivel de ajuste apropiado en OneClassSVM, se pueden utilizar los siguientes criterios:

Indicadores de Buen Ajuste

- La frontera de decisión captura la forma general de los datos sin ser excesivamente específica
- El porcentaje de outliers detectados en el conjunto de entrenamiento es consistente con el valor de ν
- El modelo mantiene un rendimiento similar en datos de validación independientes

Indicadores de Underfitting

- Frontera de decisión demasiado amplia o geométricamente simple
- Muchos puntos legítimos clasificados como anomalías
- Rendimiento pobre tanto en entrenamiento como en validación

Indicadores de Overfitting

- Frontera de decisión con formas irregulares o múltiples regiones desconectadas
- Excelente rendimiento en entrenamiento pero pobre en validación
- Alta sensibilidad a pequeñas variaciones en los datos de entrada

X. Visualizaciones Multidimensionales del Modelo OneClassSVM 4F

Este apéndice presenta las visualizaciones del modelo OneClassSVM 4F entrenado con los parámetros óptimos ($\gamma = 0.02$, $\nu = 0.002$) aplicado al conjunto de entrenamiento de fuentes astrofísicas.

X.1. Proyecciones Bidimensionales

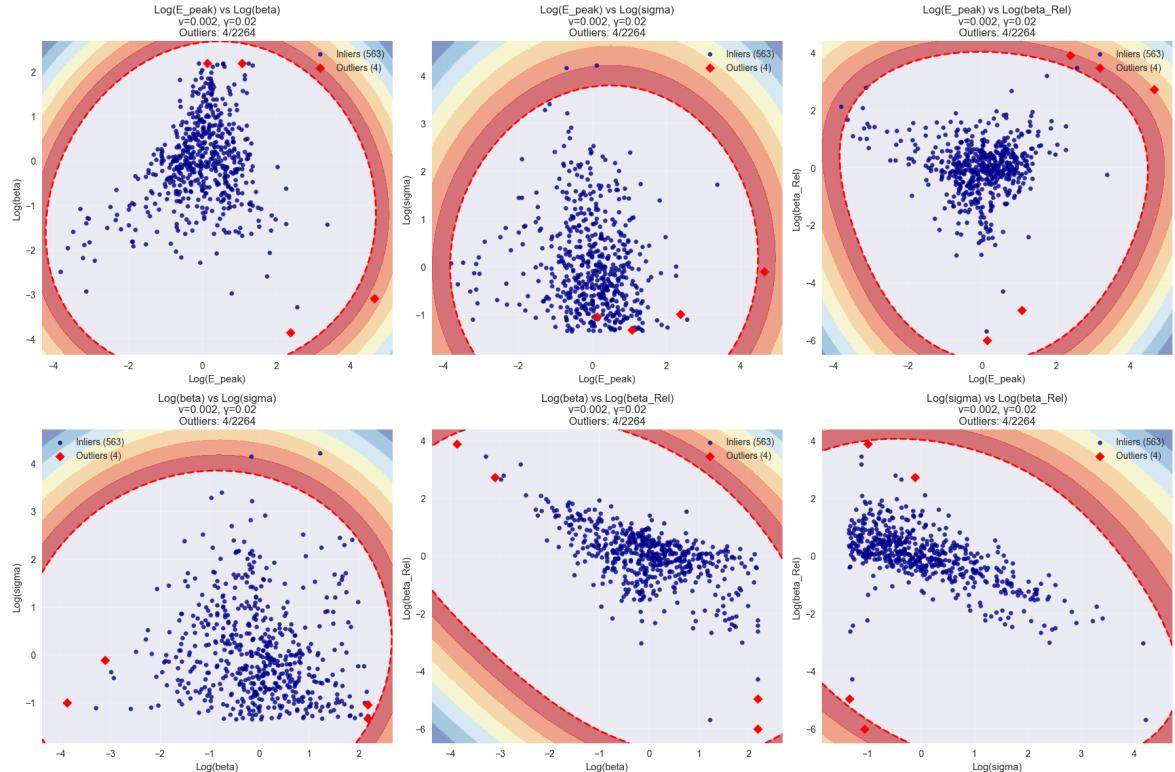


Figura 16. Conjunto completo de proyecciones bidimensionales del modelo OneClassSVM 4F. Se muestran las 6 combinaciones posibles de pares de características. Las regiones coloreadas indican los niveles de la función de decisión, mientras que la línea discontinua roja marca la frontera de clasificación.

X.2. Proyecciones Tridimensionales

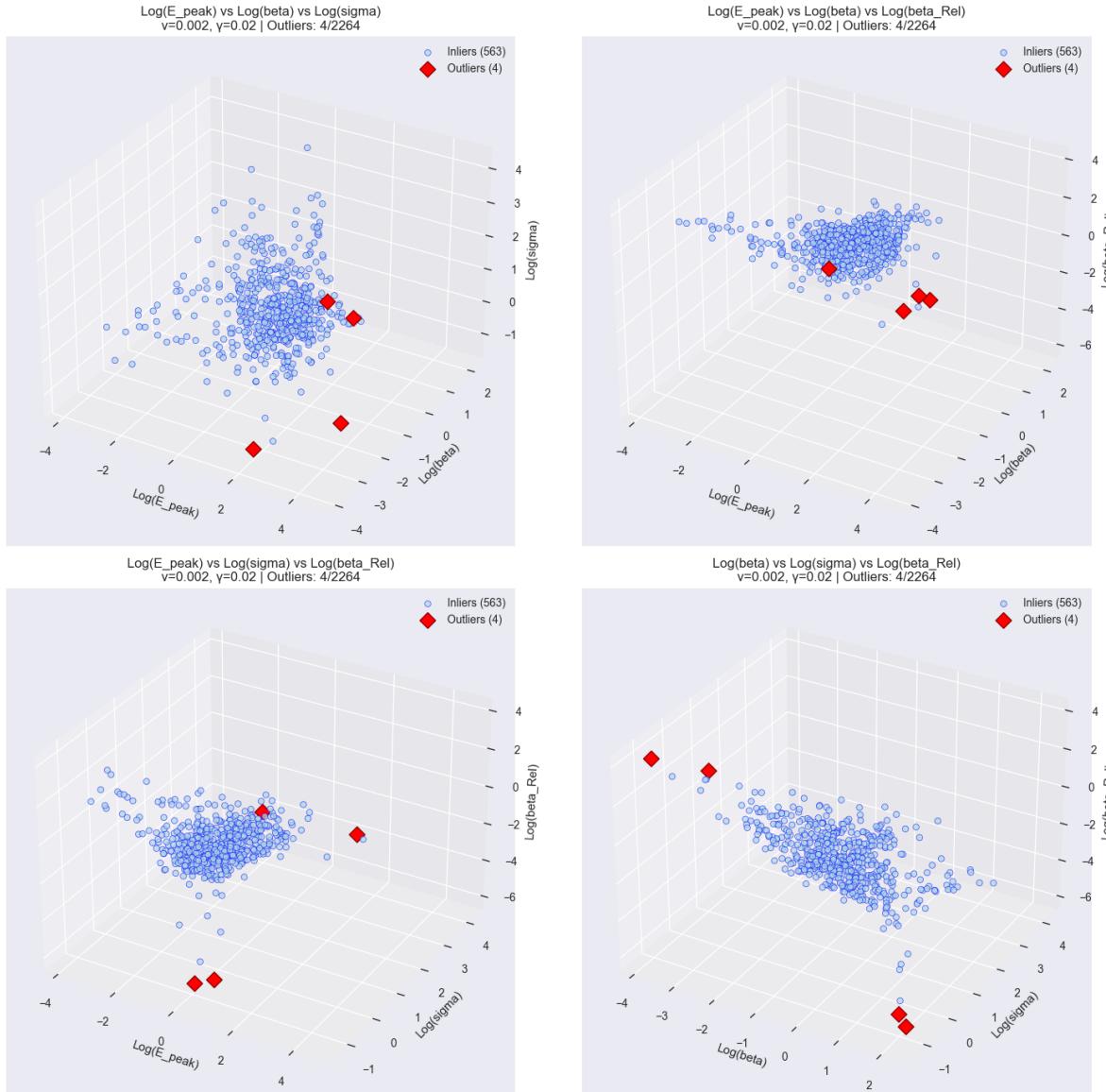


Figura 17. Proyecciones tridimensionales del modelo OneClassSVM 4F mostrando las 4 combinaciones posibles de tripletas de características. Las superficies representan isosuperficies de la función de decisión, con puntos azules indicando inliers y puntos rojos indicando outliers detectados por el modelo.

XI. Análisis de Overfitting y Underfitting en One-ClassSVM 4F

En este apéndice se presenta un análisis detallado de los fenómenos de overfitting y underfitting en el modelo OneClassSVM extendido a cuatro características (4F), aplicado a la clasificación de fuentes astrofísicas utilizando las variables Log(E_peak), Log(beta), Log(sigma) y Log(beta_Rel).

La visualización se realiza mediante proyecciones bidimensionales de todas las combinaciones posibles de pares de características, resultando en 2 gráficas complementarias.

XI.1. Configuración Óptima

Mediante validación cruzada exhaustiva y análisis de la estructura multidimensional de los datos astrofísicos, se determinaron los parámetros óptimos para el modelo 4F:

$$\gamma = 0.02, \quad \nu = 0.002 \quad (4)$$

Esta configuración representa un equilibrio entre la capacidad de capturar la complejidad del espacio 4F y la generalización a nuevas observaciones.

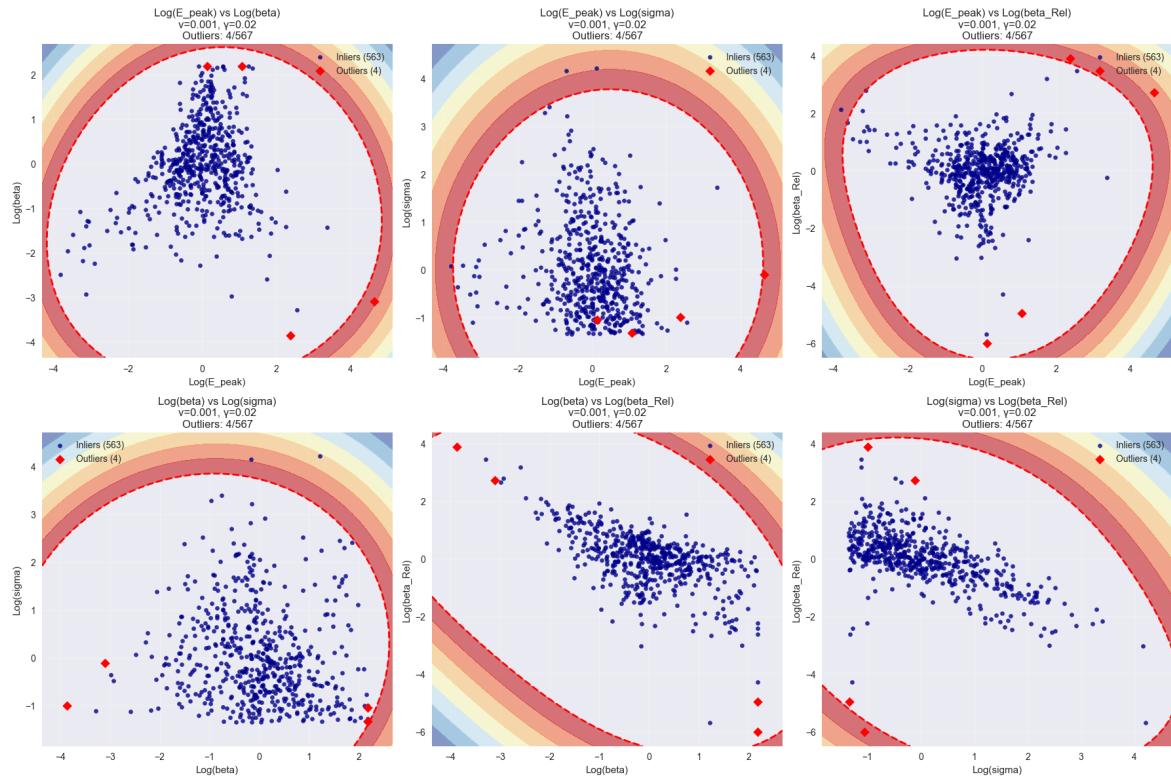


Figura 18. Fronteras de decisión óptimas del modelo OneClassSVM 4F con $\gamma = 0.02$ y $\nu = 0.002$. Se muestran las seis proyecciones bidimensionales del espacio tetradimensional, donde las líneas discontinuas rojas marcan las fronteras de decisión y las regiones coloreadas representan zonas de normalidad con diferentes niveles de confianza.

XI.2. Underfitting en el Modelo 4F

El underfitting en el contexto 4F se manifiesta cuando el modelo no logra capturar la estructura compleja del espacio multidimensional, resultando en fronteras de decisión excesivamente simplificadas.

Underfitting Severo

$$\gamma = 0.001, \quad \nu = 0.5 \quad (5)$$

Esta configuración extrema resulta en un modelo que clasifica como anómalas grandes porciones de datos que deberían considerarse normales.

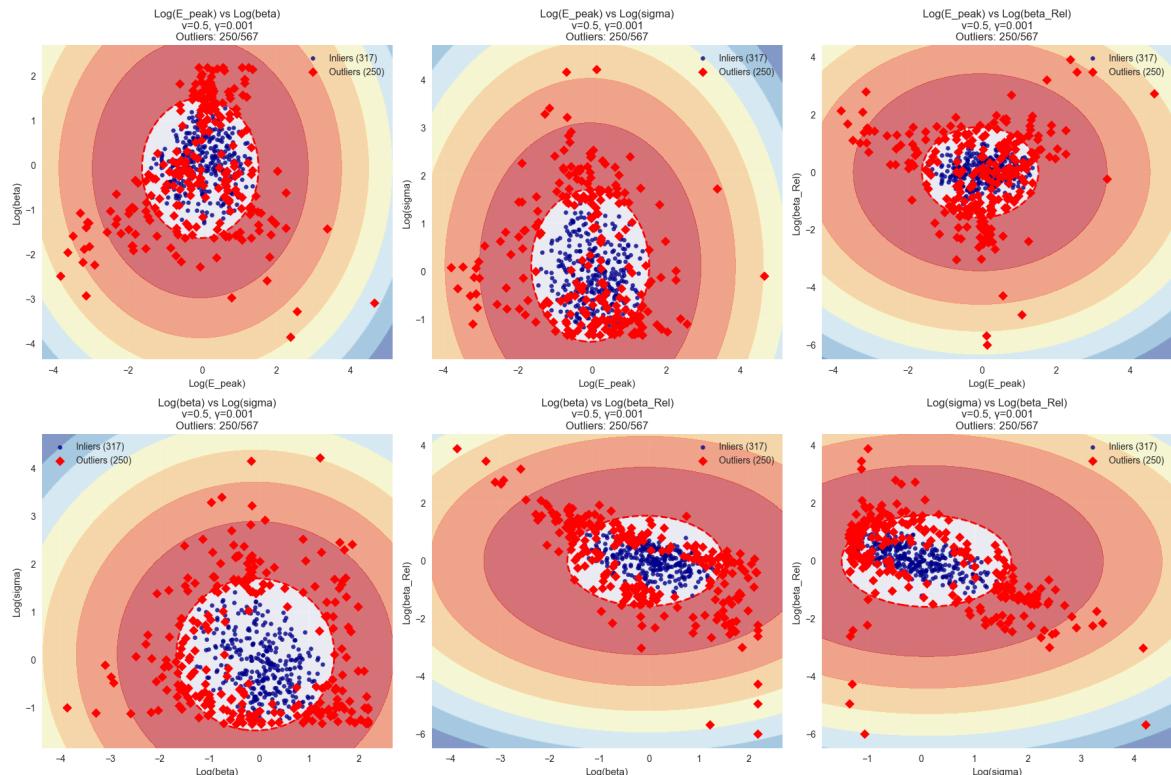


Figura 19. Ejemplo de underfitting severo en el modelo 4F. El modelo permite que hasta el 50 % de los datos sean considerados outliers, resultando en fronteras de decisión extremadamente simplistas que no reflejan la estructura real de los datos astrofísicos.

XI.3. Overfitting en el Modelo 4F

El overfitting en modelos 4F es particularmente problemático debido a la mayor dimensionalidad, que permite al modelo memorizar patrones específicos del conjunto de entrenamiento sin generalizar apropiadamente.

Overfitting Severo

$$\gamma = 1.0, \quad \nu = 0.0001 \quad (6)$$

Esta configuración extrema resulta en un modelo que se ajusta excesivamente a los datos de entrenamiento, creando fronteras de decisión altamente complejas y específicas.

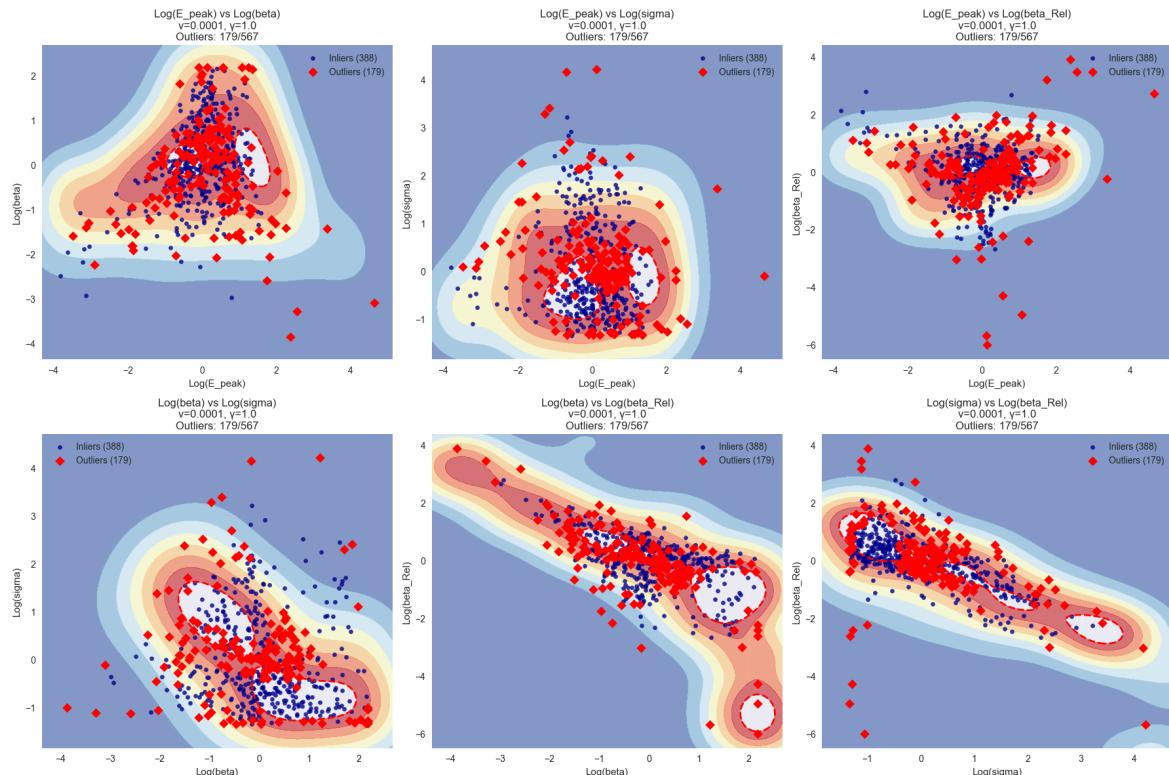


Figura 20. Ejemplo de overfitting severo en el modelo 4F. El modelo crea fronteras de decisión extremadamente complejas con múltiples regiones desconectadas, "islas" de normalidad aisladas, y sensibilidad excesiva al ruido específico del conjunto de entrenamiento.

XI.4. Desafíos Específicos del Modelo 4F

Maldición de la Dimensionalidad

El modelo 4F enfrenta desafíos adicionales relacionados con el aumento de dimensionalidad:

- **Esparsidad de datos:** En espacios de mayor dimensión, los puntos tienden a estar más dispersos
- **Visualización limitada:** Solo podemos observar proyecciones 2D del espacio 4D real
- **Complejidad computacional:** Mayor costo computacional para el cálculo de distancias y kernels

XII. Predicción UNID con Modelo OCSVM 2F (código)

Este apéndice detalla la aplicación del modelo OCSVM 2F entrenado para detectar anomalías en el conjunto de fuentes no identificadas (UNID) del catálogo 4FGL.

XII.1. Pipeline de predicción

Algorithm 2 Detección de anomalías en fuentes UNID

Entrada: Modelo OCSVM entrenado, scaler ajustado, datos UNID

Salida: Clasificación de fuentes UNID y ranking de anomalías

- 1: Cargar dataset de fuentes UNID
 - 2: Extraer características $\{\text{Log}(E_{\text{peak}}), \text{Log}(\beta)\}$
 - 3: Normalizar con el scaler del entrenamiento
 - 4: Calcular scores de decisión $s_i = \text{modelo.decision_function}(X_{\text{unid}})$
 - 5: Clasificar: $\text{pred}_i = \text{modelo.predict}(X_{\text{unid}})$ donde $\text{pred}_i \in \{-1, 1\}$
 - 6: Convertir scores a percentiles de anomalía
 - 7: Rankear candidatos por score de anomalía
 - 8: Identificar top-k fuentes más anómalas
 - 9: **return** Rankings y estadísticas de detección
-

XII.2. Carga y preprocessamiento de datos UNID

```

1 # Cargar datos de fuentes UNID
2 unids_path = "../data/unids_transformed_complete.txt"
3 df_unids = pd.read_csv(unids_path, sep='\t')
4
5 # Extraer las mismas características usadas en entrenamiento
6 X_unids_log = df_unids[["Log(E_peak)", "Log(beta)"]].values
7
8 # CRÍTICO: Usar el mismo scaler del entrenamiento
9 X_unids_scaled = scaler.transform(X_unids_log)
10
11 print(f"Datos UNID cargados: {len(df_unids)} fuentes")
12 print(f"Características: {df_unids[['Log(E_peak)', 'Log(beta)']].describe()}")

```

Listing 9. Carga de fuentes no identificadas

XII.3. Aplicación del modelo y cálculo de scores

```

1 # Aplicar el modelo entrenado
2 decision_scores = final_model.decision_function(X_unids_scaled)
3 preds_unids = final_model.predict(X_unids_scaled)
4
5 # Estadísticas de predicción
6 n_outliers = np.sum(preds_unids == -1)
7 n_inliers = np.sum(preds_unids == 1)
8 total_unids = len(preds_unids)

```

```

9
10 outlier_percentage = n_outliers / total_unids * 100
11 inlier_percentage = n_inliers / total_unids * 100
12
13 # Separar inliers y outliers para visualización
14 inliers = X_unids_scaled[preds_unids == 1]
15 outliers = X_unids_scaled[preds_unids == -1]
16
17 # Identificar las 5 fuentes más anómalas
18 top_anomalies = X_unids_scaled[np.argsort(decision_scores) [:5]]
19
20 print(f"RESULTADOS DE PREDICCIÓN:")
21 print(f" - Total UNIDs: {total_unids}")
22 print(f" - Outliers: {n_outliers} ({outlier_percentage:.1f} %)")
23 print(f" - Inliers: {n_inliers} ({inlier_percentage:.1f} %)")

```

Listing 10. Predicción y cálculo de scores de anomalía

XII.4. Sistema de ranking y análisis de candidatos

```

1 from sklearn.preprocessing import MinMaxScaler
2
3 # Convertir decision scores a anomaly scores (invertir: más negativo =
4 # más anómalo)
5 anom_scores = -decision_scores
6
7 # Escalar a percentiles [0, 100] para interpretabilidad
8 anom_percent = MinMaxScaler(feature_range=(0, 100)).fit_transform(
9     anom_scores.reshape(-1, 1)
10 ).flatten()
11
12 # Crear DataFrame de resultados completo
13 df_unids_results = df_unids.copy()
14 df_unids_results["svm_score"] = decision_scores
15 df_unids_results["prediction"] = preds_unids
16 df_unids_results["Anomaly_Score"] = anom_scores
17 df_unids_results["Anomaly_Rank(%)"] = anom_percent
18
19 # Identificar IDs de outliers detectados
20 outlier_ids = df_unids[preds_unids == -1].index.tolist()
21 print(f"Outliers detectados (IDs): {outlier_ids}")

```

Listing 11. Generación de rankings de anomalía

XII.5. Interpretación de resultados

Métricas de evaluación:

- **Decision Score:** Distancia al hiperplano de separación (valores más negativos = más anómalos)
- **Anomaly Score:** Transformación de decision scores para facilitar interpretación

- **Anomaly Rank(%):** Percentil de anomalía (0-100 %, donde 100 % = más anómalo)

Criterios de selección:

- **Predicción = -1:** Fuente clasificada como outlier por el modelo
- **Alto Anomaly Rank:** Percentil superior indica mayor probabilidad de anomalía

XIII. Predicción UNID con Modelo OCSVM 4F (código)

Esta sección detalla la aplicación del modelo OCSVM 4F a fuentes UNID, siguiendo el mismo pipeline del apéndice XII pero con características espectrales extendidas.

XIII.1. Diferencias respecto al modelo 2F

Características utilizadas:

- **Modelo 2F:** {Log(E_peak), Log(beta)}
- **Modelo 4F:** {Log(E_peak), Log(beta), Log(sigma), Log(beta_Rel)}

El algoritmo general sigue el algoritmo 2 con las adaptaciones para el espacio de características expandido.

XIII.2. Extracción de características 4D

```

1 # Cargar mismo dataset que modelo 2F
2 unids_path = "../data/unids_transformed_complete.txt"
3 df_unids = pd.read_csv(unids_path, sep='\t')
4
5 # Extraer características extendidas (4F vs 2F)
6 feature_cols = ["Log(E_peak)", "Log(beta)", "Log(sigma)", "Log(
    beta_Rel)"]
7 X_unids = df_unids[feature_cols].values
8
9 # CRÍTICO: Usar el scaler del modelo 4F entrenado
10 X_unids_scaled = scaler_final.transform(X_unids)
11
12 print(f"Características extraídas: {X_unids.shape}")
13 print(f"Features: {feature_cols}")

```

Listing 12. Aplicación del modelo 4F a fuentes UNID

XIII.3. Predicción y análisis de resultados

```

1 # Aplicar modelo OCSVM 4F
2 decision_scores = final_model.decision_function(X_unids_scaled)
3 unids_preds = final_model.predict(X_unids_scaled)
4
5 # Estadísticas de detección
6 total_unids = len(unids_preds)
7 n_outliers = np.sum(unids_preds == -1)
8 n_inliers = np.sum(unids_preds == 1)
9
10 outlier_percentage = n_outliers / total_unids * 100
11 inlier_percentage = n_inliers / total_unids * 100
12
13 # Máscaras para análisis posterior
14 inliers_mask = unids_preds == 1
15 outliers_mask = unids_preds == -1

```

```
16
17 print(f"RESULTADOS DE PREDICCIÓN (Modelo 4F):")
18 print(f" - Total UnIDs analizadas: {len(unids_preds)}")
19 print(f" - Outliers detectados: {n_outliers} ({outlier_percentage:.1f} %)")
20 print(f" - Inliers detectados: {n_inliers} ({inlier_percentage:.1f} %)")
21
22 # Identificar outliers específicos
23 outlier_ids = df_unids.loc[outliers_mask, 'number']
24 print(f"Outliers detectados (IDs): {outlier_ids.tolist()}")
```

Listing 13. Detección de anomalías con modelo 4F

XIII.4. Sistema de ranking extendido

```
1 # Sistema de ranking idéntico al modelo 2F
2 anom_scores = -decision_scores
3 anom_percent = MinMaxScaler(feature_range=(0, 100)).fit_transform(
4     anom_scores.reshape(-1, 1)
5 ).flatten()
6
7 # DataFrame de resultados con información 4D
8 df_unids_results = df_unids.copy()
9 df_unids_results["svm_score"] = decision_scores
10 df_unids_results["prediction"] = unids_preds
11 df_unids_results["Anomaly_Score"] = anom_scores
12 df_unids_results["Anomaly_Rank(%)"] = anom_percent
```

Listing 14. Ranking con información espectral completa

XIV. Visualizaciones Multidim. de Predicciones con OCSVM 4F

XIV.1. Proyecciones bidimensionales completas

El modelo OCSVM 4D aplicado a las fuentes UNID genera 6 proyecciones bidimensionales posibles, donde podemos confirmar la ubicación de aquellas fuentes clasificadas como *outliers* (círculos amarillos con ID) en los diferentes espacios de características:

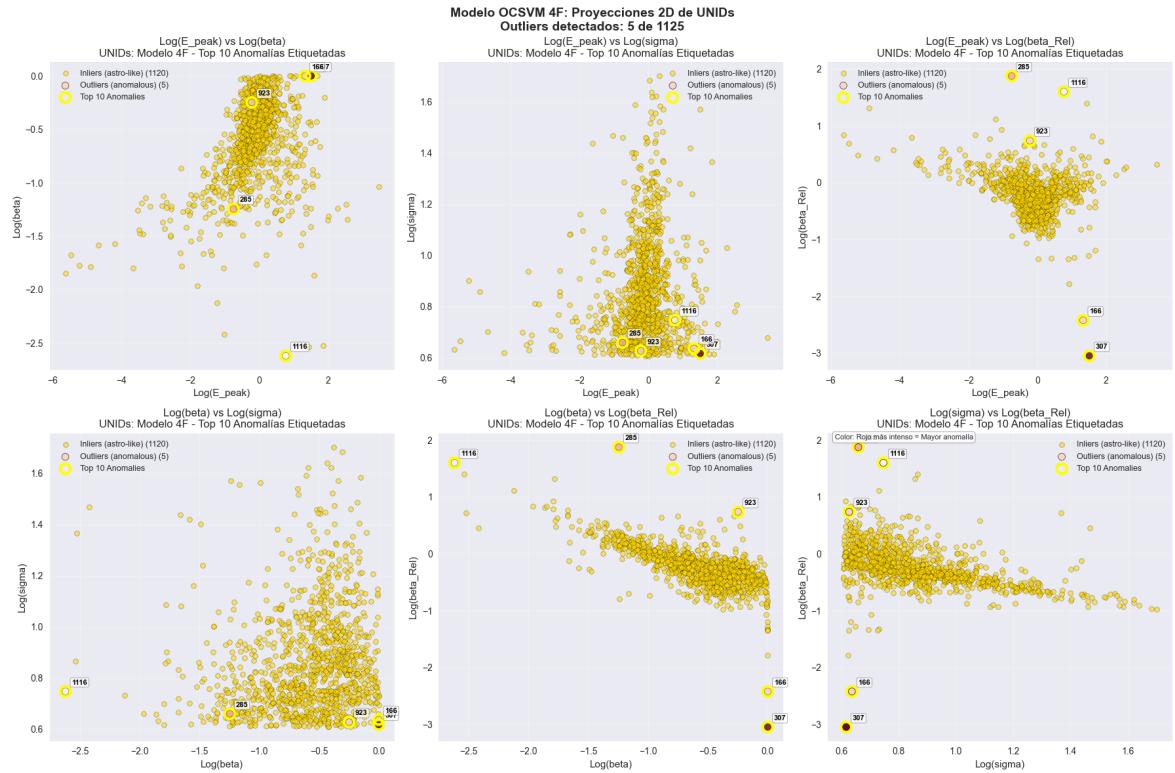


Figura 21. Las 6 proyecciones bidimensionales de las predicciones hechas por el modelo OCSVM 4F sobre las UNIDs

XIV.2. Proyecciones tridimensionales

El modelo OCSVM 4D aplicado a las fuentes UNID genera además 4 proyecciones tridimensionales posibles, esta vez en azul, podemos observar la ubicación de aquellas fuentes clasificadas como *outliers* (puntos rojos) en los diferentes espacios de características 3D:

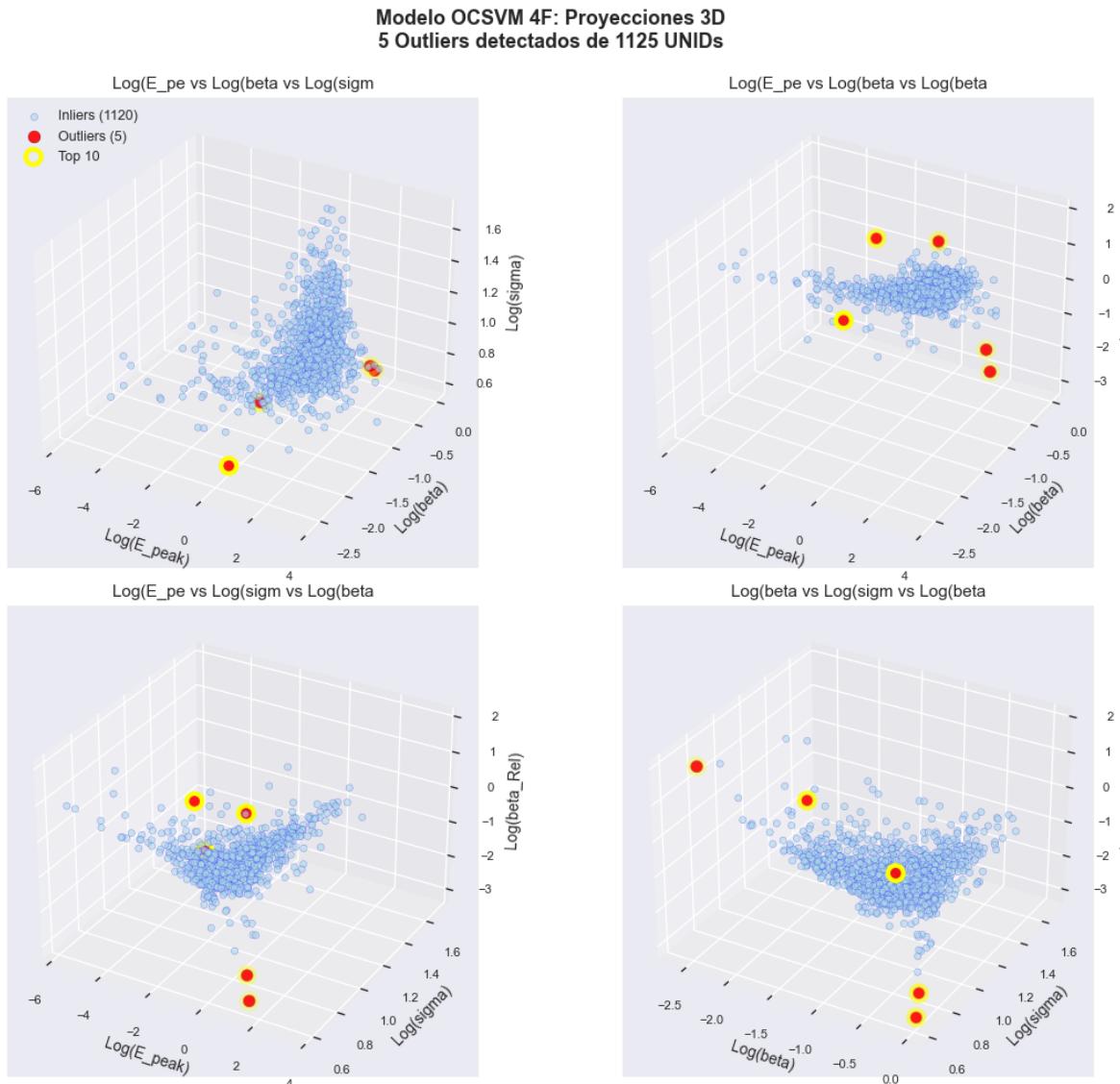


Figura 22. Proyecciones tridimensionales de las predicciones hechas por el modelo OCSVM 4F sobre las UNIDs

XV. Análisis detallado ANN vs OCSVM

Este apéndice contiene el análisis detallado del posicionamiento cruzado, rankings individuales y caracterización específica de candidatos para el framework comparativo ANN vs OCSVM.

XV.1. Análisis detallado 2F (2 características)

Candidatos identificados por método

OneClassSVM 2F - Outliers detectados:

- **Candidatos:** [275, 1017, 1054, 1116]
- **Top 10 anomalías:** [126, 273, 275, 564, 592, 1017, 1043, 1054, 1111, 1116]

ANN 2F - Top candidatos:

- **Top 4:** [101, 551, 664, 1114]
- **Top 10:** [96, 101, 127, 138, 551, 560, 663, 664, 1113, 1114]
- **Alto consenso (≥ 0.5):** [664, 1114]

Análisis de posicionamiento cruzado 2F

Evaluación de outliers OCSVM en sistema ANN 2F:

UNID	Posición ANN	Consenso ANN	Interpretación
1054	#58	0.409	Mejor compatibilidad ANN-OCSVM en 2F
1017	#342	0.324	Consenso ANN moderado-bajo
275	#561	0.282	Consenso ANN bajo
1116	#630	0.270	Menor compatibilidad con ANN

Tabla 11. Ranking de outliers OCSVM 2F en sistema ANN 2F

Evaluación de top ANN en sistema OCSVM 2F:

UNID	Anomaly Rank OCSVM	Clasificación	Interpretación
101	64.1 %	Inlier	Consenso ANN alto, normal para OCSVM
551	33.7 %	Inlier	Dentro de distribución normal
664	31.1 %	Inlier	Alto consenso ANN, típico OCSVM
1114	32.3 %	Inlier	Alto consenso ANN, típico OCSVM

Tabla 12. Ranking de top ANN 2F en sistema OCSVM 2F

XV.2. Análisis detallado 4F (4 características)

Candidatos identificados por método

OneClassSVM 4F - Outliers detectados:

- **Candidatos:** [166, 285, 307, 923, 1116]
- **Top 10 anomalías:** [64, 117, 166, 285, 307, 562, 843, 923, 1109, 1116]

ANN 4F - Top candidatos:

- **Top 4:** [371, 556, 596, 821]
- **Top 10:** [29, 106, 172, 371, 556, 560, 596, 622, 781, 821]
- **Alto consenso (≥ 0.7):** [371, 821]

Análisis de posicionamiento cruzado 4F

Evaluación de outliers OCSVM 4F en sistema ANN 4F:

UNID	Pos. ANN 4F	Consenso ANN	Cambio vs 2F	Observaciones
285	#256	0.319	Mejora significativa	Mayor compatibilidad en 4D
923	#254	0.321	Nuevo candidato 4F	Alta compatibilidad cruzada
166	#352	0.285	Nuevo candidato 4F	Compatibilidad moderada
307	#723	0.166	Nuevo candidato 4F	Baja compatibilidad
1116	#1105	0.020	Deterioro extremo	Pérdida significativa

Tabla 13. Evolución del posicionamiento cruzado en modelos 4F

Evaluación de top ANN 4F en sistema OCSVM 4F:

UNID	Anomaly Rank OCSVM	Clasificación	Interpretación
371	29.8 %	Inlier	Alto consenso ANN, distribución normal OCSVM
556	16.0 %	Inlier	Patrón reconocido ANN, típico estadísticamente
596	20.6 %	Inlier	Candidato ANN sin anomalía estadística
821	39.3 %	Inlier	Alto consenso ANN, normal para OCSVM

Tabla 14. Evaluación de candidatos ANN 4F en sistema OCSVM 4F

XV.3. Caracterización individual de candidatos destacados

Candidatos con mayor compatibilidad cruzada

UNID 1054 (Modelo 2F):

- **OCSVM 2F:** Outlier extremo (100 % anomaly rank)
- **ANN 2F:** Posición #58 (consenso 0.409)
- **Característica:** Mejor ejemplo de compatibilidad relativa entre métodos
- **Interpretación técnica:** Anomalía estadística con ciertos patrones estructurales

UNID 285 (Modelo 4F):

- **OCSVM 4F:** Outlier detectado (79.8 % anomaly rank)
- **ANN 4F:** Posición #256 (consenso 0.319)
- **Evolución:** Mejora significativa en compatibilidad vs 2F
- **Interpretación técnica:** Beneficio de información espectral adicional

UNID 923 (Modelo 4F):

- **OCSVM 4F:** Outlier detectado
- **ANN 4F:** Posición #254 (consenso 0.321)
- **Característica:** Nuevo candidato 4F con alta compatibilidad cruzada
- **Interpretación técnica:** Beneficio específico del espacio 4D

Candidatos de alta especificidad por método

Alto consenso ANN sin detección OCSVM:

UNID	Modelo	Consenso ANN	Rank OCSVM	Característica
664	2F	0.583	31.1 % (Inlier)	Patrón específico ANN
1114	2F	0.567	32.3 % (Inlier)	Patrón específico ANN
371	4F	0.756	29.8 % (Inlier)	Patrón complejo 4D
821	4F	0.701	39.3 % (Inlier)	Patrón complejo 4D

Tabla 15. Candidatos de alta especificidad ANN

Outliers OCSVM sin reconocimiento ANN:

UNID	Modelo	Rank OCSVM	Pos. ANN	Característica
275	2F	99.9 % (Outlier)	#561	Anomalía estadística pura
1017	2F	99.9 % (Outlier)	#342	Anomalía estadística pura
307	4F	100 % (Outlier)	#723	Anomalía extrema 4D
166	4F	Outlier	#352	Anomalía específica 4D

Tabla 16. Candidatos de alta especificidad OCSVM

XV.4. Evolución dimensional del posicionamiento

Análisis de consistencia cruzada

Candidatos que aparecen en ambas dimensionalidades:

UNID	Método	Status 2F	Status 4F	Evolución
1116	OCSVM	Outlier	Outlier	Consistente
560	ANN	Top 10	Top 10	Consistente

Tabla 17. Candidatos consistentes across dimensionalidades

Métricas de estabilidad dimensional

Método	Candidatos 2F	Candidatos 4F	Overlap interno
OCSVM	4 outliers	5 outliers	25 % (1/4)
ANN Top-4	4 candidatos	4 candidatos	0 % (0/4)
ANN Top-10	10 candidatos	10 candidatos	10 % (1/10)

Tabla 18. Estabilidad dimensional por método

XV.5. Síntesis técnica del análisis detallado

Patrones de comportamiento identificados

1. **Complementariedad robusta:** 0 % overlap en todas las comparaciones
2. **Especialización metodológica:** Cada algoritmo mantiene criterios específicos
3. **Evolución dimensional variable:** Algunos candidatos mejoran/empeoran en 4D
4. **Consistencia algorítmica:** Patrones de detección sostenidos across dimensiones

Implicaciones para el framework

El análisis detallado confirma que:

- La complementariedad no es accidental sino estructural
- Cada método aporta valor específico no replicable por el otro
- La dimensionalidad adicional modifica compatibilidades pero mantiene divergencia
- El framework dual maximiza cobertura sin redundancia significativa

Esta caracterización detallada valida las conclusiones del análisis principal y proporciona evidencia granular de la efectividad del framework de ensemble implementado.

XVI. Figuras del experimento OCSVM con 3F del DR4

XVI.1. Planos 2D: ASOC

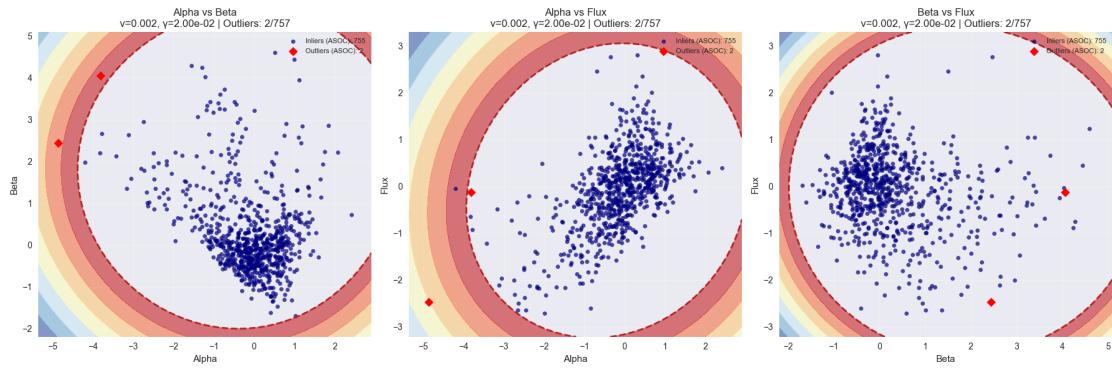


Figura 23. Gráficas 2D para el conjunto de test de fuentes asociadas (ASOC). Cada panel muestra uno de los pares de características, con la frontera de decisión del OCSVM (línea discontinua) y la clasificación resultante: inliers en azul y outliers en rojo. Se indica el total de outliers detectados: 2/757.

XVI.2. Planos 2D: UNAS

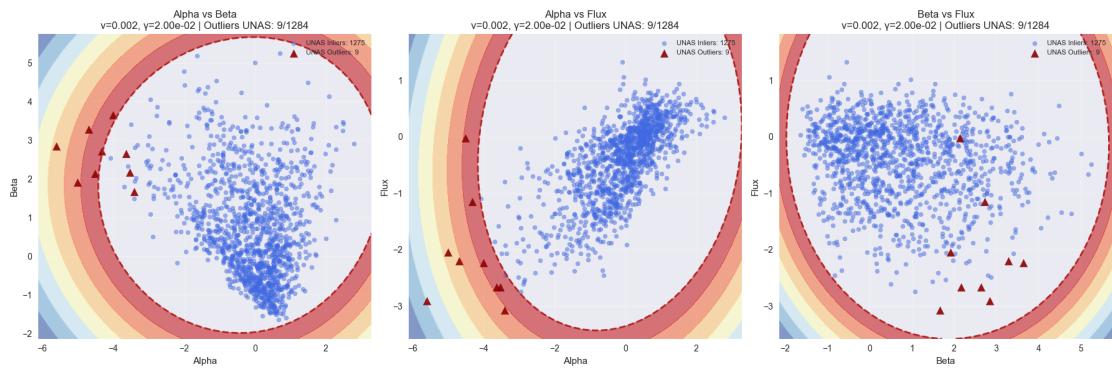


Figura 24. Gráficas 2D para las fuentes no asociadas (UNAS). Cada panel corresponde a pares de características, aplicando el mismo modelo OCSVM ($\nu = 0.002$, $\gamma = 0.02$). Se muestran los inliers (1 275/1 284) en azul semitransparente y los outliers (9/1 284) en rojo.

XVI.3. Visualización 3D de UNAS

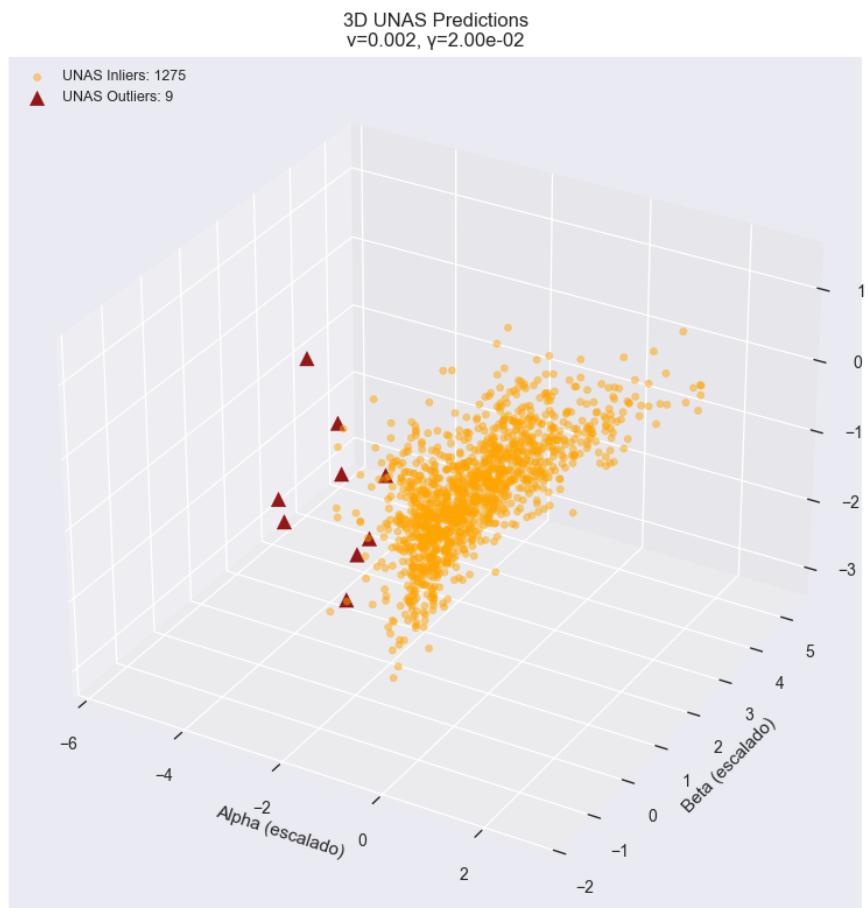


Figura 25. Visualización tridimensional de las predicciones sobre fuentes UNAS. Ejes α , β y flux (todos escalados). Los 1 275 inliers aparecen en naranja, y los 9 outliers (candidatos a materia oscura) en triángulos rojos.

Bibliografía

- S. Abdollahi, F. Acero, M. Ackermann, M. Ajello, W. B. Atwood, L. Baldini, J. Ballet, G. Barbiellini, et al. Fermi large area telescope fourth source catalog. *The Astrophysical Journal Supplement Series*, 247(1):33, 2020. doi: 10.3847/1538-4365/ab6bcb.
- Anthropic. Claude: An ai assistant by anthropic. <https://claude.ai>, 2024. Accedido en mayo de 2025.
- G. Bertone and D. Hooper. History of dark matter. *Reviews of Modern Physics*, 90(4):045002, 2018. doi: 10.1103/RevModPhys.90.045002.
- J. Bobadilla. *Machine learning y deep learning usando Python, Scikit y Keras*. Ediciones Paraninfo, 2021.
- A. Bou Nassif, M. Azzeh, and L. F. Capretz. Anomaly detection techniques in data science: A review. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2022.
- G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.
- V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.
- M. Cirelli, G. Corcella, A. Hektor, G. Hütsi, M. Kadastik, P. Panci, et al. Pppc 4 dm id: A poor particle physicist cookbook for dark matter indirect detection. *Journal of Cosmology and Astroparticle Physics*, 2011(03):051, 2011. doi: 10.1088/1475-7516/2011/03/051.
- V. Gammaldi, B. Zaldívar, M. A. Sánchez-Conde, and J. Coronado-Blázquez. A search for dark matter among fermi-lat unidentified sources with systematic features in machine learning. *Monthly Notices of the Royal Astronomical Society*, 521(2):2751–2767, 2023.
- GitHub. Github copilot: Your ai pair programmer. <https://copilot.github.com>, 2023. Accedido en mayo de 2025.
- N. Mirabal, D. Nieto, and S. Pardo. The exotic fraction among unassociated fermi sources. *Monthly Notices of the Royal Astronomical Society*, 424(1):64–68, 2012.
- OpenAI. Chatgpt: Optimizing language models for dialogue. <https://openai.com/chatgpt>, 2023. Accedido en mayo de 2025.
- Planck Collaboration. Planck 2013 results. i. overview of products and scientific results. *Astronomy & Astrophysics*, 571:A1, 2013. doi: 10.1051/0004-6361/201321529.
- A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.
- P. M. Saz Parkinson, H. Xu, P. L. H. Yu, D. Salvetti, M. Marelli, and A. D. Falcone. Classification and ranking of fermi lat gamma-ray sources from the 3fgl catalog using machine learning techniques. *The Astrophysical Journal*, 820(1):8, 2017.

Bibliografía

- B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.
- Özer Çelik and S. S. Altunaydin. A research on machine learning methods and its applications. Unpublished manuscript, 2018.

Nota sobre herramientas de inteligencia artificial: Durante el desarrollo de este trabajo se utilizaron herramientas de inteligencia artificial generativa [Anthropic \(2024\)](#); [GitHub \(2023\)](#); [OpenAI \(2023\)](#) como asistentes técnicos en tareas de redacción, revisión de código y documentación, siguiendo la normativa académica vigente. Todas las decisiones metodológicas, análisis de resultados y conclusiones científicas son responsabilidad exclusiva del autor.