



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

**Análisis de Coincidencia y Adherencia de
Trayectorias a Plan de Vuelo utilizando
Inteligencia Artificial**

Autor(a): Rubén González Velasco

Tutores: Antonio Jiménez Martín y Sergi Mas Pujol

Madrid, julio de 2025

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Análisis de Coincidencia y Adherencia de Trayectorias a Plan de Vuelo
utilizando Inteligencia Artificial

julio de 2025

Autor: Rubén González Velasco

Tutores: Antonio Jiménez Martín
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Sergi Mas Pujol
Centro de Referencia de Investigación, Desarrollo e Innovación ATM

Resumen

El control del tráfico aéreo es un sistema crítico que requiere evaluar la adherencia entre trayectorias planificadas y ejecutadas para garantizar la seguridad y eficiencia del espacio aéreo. Se estudiarán trayectorias de planes de vuelo, que representan las intenciones operacionales previas al despegue, y trayectorias radar, que reflejan el recorrido real, afectado por factores como clima, congestión y decisiones tácticas. La comparación entre ambas fuentes es fundamental para detectar desviaciones y mejorar la gestión táctica, enmarcándose en el proyecto DELFOS de CRIDA A.I.E.

El análisis de adherencia presenta importantes desafíos técnicos. Las trayectorias radar sufren ruido, pérdidas de señal y muestreos irregulares, mientras que los planes de vuelo son representaciones abstractas. Esta disparidad en formato y resolución exige métodos robustos y escalables para comparar trayectorias automáticamente, permitiendo validar modelos predictivos y optimizar la gestión operativa. La metodología desarrollada aborda tanto discrepancias espaciales como temporales. Las primeras se calculan mediante distancias geodésicas combinando componentes horizontales y verticales, mientras que las diferencias temporales consideran desfases en inicio, duración y dilataciones temporales.

Además, este Trabajo Fin de Máster propone *AdherNet*, un modelo de Inteligencia Artificial basado en *redes neuronales recurrentes con atención*, entrenado mediante aprendizaje supervisado para estimar la similitud entre trayectorias sin necesidad de alineamientos costosos. El modelo procesa directamente coordenadas geográficas y marcas temporales, con dos variantes: una para comparar planes inicial y final, y otra para medir adherencia entre plan final y trayectoria radar real. *AdherNet* utiliza una arquitectura siamesa con codificadores *BiGRU* y *atención multi-cabeza* para generar embeddings densos. Se aplican normalización por capas, conexiones residuales y enmascaramiento para manejar trayectorias de longitud variable. El entrenamiento supervisado optimiza la función *MSE*, evaluándose con métricas *R²* y *MAE*.

Los casos de uso incluyen comparación directa de trayectorias, evaluación de adherencia radar-plan y identificación de trayectorias anónimas mediante búsqueda eficiente. El modelo puede aplicarse como métrica de validación predictiva, *clustering* de flujos de vuelo y mejora de calidad de datos.

En resumen, este Trabajo Fin de Máster ofrece una solución innovadora basada en Inteligencia Artificial para medir adherencia entre planes de vuelo y trayectorias reales, superando limitaciones técnicas clásicas y proporcionando herramientas potentes para la mejora continua del control del tráfico aéreo en el contexto del proyecto DELFOS.

Abstract

Air traffic control is a critical system that requires assessment of the adherence between planned and executed trajectories to ensure the safety and efficiency of airspace. Flight plan trajectories, which represent the operational intentions prior to take-off, and radar trajectories, which reflect the actual path, affected by factors such as weather, congestion and tactical decisions, will be studied. The comparison between the two sources is essential to detect deviations and improve tactical management, as part of CRIDA A.I.E.'s DELFOS project.

Adherence analysis presents significant technical challenges. Radar trajectories suffer from noise, signal loss and irregular sampling, while flight plans are abstract representations. This disparity in format and resolution requires robust and scalable methods to compare trajectories automatically, enabling predictive models to be validated and operational management to be optimised. The methodology developed addresses both spatial and temporal discrepancies. The former are calculated using geodetic distances combining horizontal and vertical components, while temporal differences consider lags in onset, duration and time dilations.

In addition, this work proposes *AdherNet*, an artificial intelligence model based on recurrent neural networks with attention, trained by supervised learning to estimate the similarity between trajectories without the need for costly alignments. The model directly processes geographic coordinates and time stamps, with two variants: one to compare initial and final plans, and another to measure adherence between final plan and actual radar trajectory. *AdherNet* uses a *Siamese* architecture with *BiGRU encoders* and *multi-head attention* to generate dense embeddings. Layered normalisation, residual connections and masking are applied to handle variable length trajectories. Supervised training optimises the *MSE* function, evaluated with R^2 and *MAE* metrics.

Use cases include direct trajectory comparison, radar-plan stickiness assessment and identification of anonymous trajectories by efficient search. The model can be applied as a predictive validation metric, flight flow clustering and data quality improvement.

In summary, this work offers an innovative artificial intelligence-based solution to measure adherence between flight plans and real trajectories, overcoming classical technical limitations and providing powerful tools for the continuous improvement of air traffic control in the context of the DELFOS project.

Agradecimientos

Quiero comenzar agradeciendo a mis padres y a toda mi familia, cuyo apoyo incondicional y ánimo constante han sido la base que me ha permitido afrontar este trabajo de fin de máster. Durante estos meses, su confianza y acompañamiento han sido una fuente constante de motivación y fuerza para seguir adelante.

Durante este tiempo, he tenido la oportunidad de realizar mis prácticas en la empresa CRIDA A.I.E, lo que supuso un auténtico reto personal y profesional. Cada día, partiendo desde Segovia, me enfrentaba a un viaje de más de una hora de ida y otra de vuelta, conduciendo con constancia y dedicación. Aun así, esta experiencia me ha brindado una perfecta entrada al mundo laboral, acercándome a la realidad de la profesión y al esfuerzo que implica integrarse en un entorno empresarial.

El trabajo de fin de máster que he desarrollado en CRIDA A.I.E ha estado estrechamente vinculado con el ámbito del máster en Inteligencia Artificial, lo que ha enriquecido enormemente mi aprendizaje. Aplicar técnicas y modelos en problemas reales, y entender cómo se transforman en soluciones concretas, ha sido una gran experiencia que difícilmente podría haberse conseguido solo con la formación académica.

Quiero agradecer de forma especial a Sergi Mas Pujol y a todos mis compañeros en la empresa, por integrarme a mí y a los demás becarios en la empresa, por tratarnos como a uno más del equipo, por su apoyo y por facilitar un ambiente de trabajo que ha sido muy enriquecedor.

También extiendo mi agradecimiento a Antonio Jiménez Martín, mi tutor académico, por sus comentarios, orientación y ayuda en la organización tanto de las prácticas como de este Trabajo de Fin de Máster, lo que ha facilitado mucho el desarrollo de todo el proceso, y a la Universidad Politécnica de Madrid por organizar estas prácticas que han sido clave para conectar la teoría con la práctica profesional.

Por último, no puedo dejar de agradecer a mis amigos del máster, que me han acompañado durante esta etapa de mi vida, brindándome apoyo, ánimo y compañía en los momentos más intensos.

Este Trabajo Fin Máster es parte del proyecto de I+D+i PID2021-122209OB-C31 y de la Ayuda RED2022-134540-T financiados por MICIU/AEI/10.13039/501100011033.

Tabla de contenidos

1. Introducción	1
1.1. Motivación y contexto del problema	1
1.1.1. El sistema de control de tráfico aéreo	1
1.1.2. Importancia del análisis de eficiencia y desafíos de los datos . . .	2
1.2. Planteamiento del problema	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Estructura del documento	5
2. Conceptos previos	7
2.1. Trayectoria	7
2.2. Distancia entre puntos de trayectorias	8
2.2.1. Distancia espacial	8
2.2.2. Distancia temporal	8
2.3. Métrica de similitud de trayectorias	9
2.3.1. Propiedades deseables de las métricas	10
3. Estado del arte	11
3.1. Métricas tradicionales de similitud de trayectorias	12
3.1.1. Escaneo lineal (Linear scan)	12
3.1.2. Programación dinámica (Dynamic programming)	12
3.1.3. Enumeración (Enumeration)	14
3.2. Limitaciones de métricas tradicionales ante datos reales	14
3.2.1. Necesidad de interpolación	14
3.3. Codificación profunda de trayectorias (Trajectory Encoding)	16
3.3.1. Modelos de codificación autosupervisados basados en RNN	16
3.3.2. Modelos de codificación autosupervisada basados en Atención . .	17
3.3.3. Modelos de codificación autosupervisada basados en CNN	18
3.3.4. Modelos de codificación supervisados basados en RNN y Atención	18
3.3.5. Modelos de codificación supervisados basados en GNN y Atención	19
3.4. Aprendizaje de métricas para trayectorias (Metric learning)	19
3.4.1. Métodos de aprendizaje de métricas basados en RNN	19
3.5. Análisis comparativo de métodos de similitud de trayectorias	20
4. Metodología	23
4.1. Método de etiquetado	23
4.1.1. Similitud espacial	24

4.1.2. Similitud temporal	25
4.1.3. Similitud de trayectorias	25
4.2. Diseño del modelo <i>AdherNet</i>	26
4.2.1. Codificador de trayectorias (<i>BiGRU + Attention</i>)	27
4.2.2. Modelo de comparación y predicción de similitud	29
4.2.3. Entrenamiento y evaluación del modelo	29
5. Resultados	33
5.1. Datos	33
5.1.1. Estructura de los datos crudos	33
5.1.2. Obtención de trayectorias a partir de los datos crudos	34
5.1.3. Características de los datos	35
5.2. Experimentación	36
5.2.1. Evaluación de resultados y estrategias de etiquetado	36
5.2.2. Comparación entre arquitecturas simples y complejas	39
5.2.3. Funcionalidades de búsqueda y ordenación	39
6. Conclusión	43
6.1. Casos de uso	44
6.2. Trabajos futuros	45
Bibliografía	47

Capítulo 1

Introducción

1.1. Motivación y contexto del problema

1.1.1. El sistema de control de tráfico aéreo

El Control del Tráfico Aéreo (ATC, por sus siglas en inglés) es un servicio esencial que garantiza la seguridad, el orden y la eficiencia de las operaciones aéreas, tanto en vuelo como en tierra. Su función principal es prevenir colisiones entre aeronaves, organizar los flujos de tráfico y proporcionar asistencia a las tripulaciones durante todas las fases del vuelo. Esta labor implica una supervisión constante por parte de controladores aéreos que, asistidos por sistemas automatizados, toman decisiones tácticas en tiempo real para responder a condiciones operativas cambiantes como el volumen de tráfico, la meteorología o la disponibilidad del espacio aéreo.

En España ENAIRE es el proveedor de servicios de navegación aérea del país (ANSP, por sus siglas en inglés) responsable de más de dos millones de vuelos anuales. Sus controladores operan desde torres de control, centros de control de área y aproximación, y utilizan diversas tecnologías de vigilancia para mantener las separaciones mínimas entre aeronaves, autorizar maniobras y gestionar los flujos de tráfico. Esta red de vigilancia incluye radares primarios y secundarios, así como sistemas basados en posicionamiento satelital como el ADS-B (*Automatic Dependent Surveillance-Broadcast*), que permiten conocer con precisión la ubicación y trayectoria de cada aeronave en tiempo real.

En este entorno tan regulado y crítico, los planes de vuelo y las trayectorias reales se convierten en piezas fundamentales para la operación. Un plan de vuelo es un documento que contiene la intención operativa de un vuelo: incluye información sobre la ruta prevista, niveles de altitud, puntos de paso, tiempos estimados y otros parámetros relevantes. Existen dos versiones clave de este plan: el plan de vuelo inicial, que se presenta antes del despegue y refleja la intención original del operador aéreo, y el plan de vuelo final, que corresponde con la última actualización enviada e incorpora las modificaciones aprobadas por el control de tráfico antes del vuelo. Es importante destacar que ambas versiones siguen siendo intenciones de vuelo, no registros de ejecución real.

Por otro lado, las trayectorias reales se obtienen a partir de las trazas radar, que recogen las posiciones espaciales que ha ocupado la aeronave a lo largo de su vuelo.

Estas trazas son observaciones discretas generadas por los sistemas de vigilancia mencionados anteriormente, y constituyen una representación fiel (aunque a veces incompleta o ruidosa) del comportamiento real de la aeronave. A diferencia del plan de vuelo, que representa una intención, la trayectoria registrada representa la ejecución efectiva del vuelo, condicionada por factores como restricciones del espacio aéreo, decisiones del controlador o condiciones meteorológicas adversas.

El análisis conjunto de planes de vuelo y trayectorias radar no solo es esencial para la gestión operativa en tiempo real, sino que también desempeña un papel clave en el análisis post-operacional de modelos predictivos, en la evaluación de la eficiencia del sistema y en la mejora continua de la planificación táctica y estratégica del tráfico aéreo. Es precisamente en este marco donde se sitúa el presente Trabajo Fin de Máster, como parte del proyecto interno DELFOS (*CriDa aEriaL FOrecaSt*), una iniciativa de CRIDA A.I.E. orientada al desarrollo de capacidades avanzadas de predicción del tráfico aéreo instrumental en el espacio aéreo español. Dentro de este proyecto, el análisis de adherencia entre trayectorias planificadas y reales surge como una herramienta crítica para evaluar la calidad de las predicciones y detectar comportamientos anómalos o ineficiencias del sistema.

1.1.2. Importancia del análisis de eficiencia y desafíos de los datos

El estudio de la eficiencia operativa en el sistema de control de tráfico aéreo es crucial tanto desde la perspectiva de la seguridad como desde la económica y medioambiental. Cuando una aeronave se desvía significativamente del plan previsto, pueden generarse situaciones que comprometan pérdida de separación mínima entre vuelos, lo que exige una rápida intervención del controlador para evitar incidentes. Estas desviaciones, además, pueden conllevar rutas más largas o altitudes subóptimas, lo que se traduce en un mayor consumo de combustible, aumento de emisiones y costes adicionales para las aerolíneas. Desde el punto de vista del gestor del espacio aéreo, el cumplimiento razonable de los planes de vuelo permite una mejor utilización de los recursos disponibles y una mayor capacidad del sistema, aspectos fundamentales en entornos de tráfico denso.

A pesar de su importancia, el análisis sistemático del grado de adherencia entre lo planificado y lo ejecutado presenta una serie de obstáculos técnicos considerables. Uno de los principales reside en la heterogeneidad e imperfección inherente a los datos disponibles. Las trazas radar, aunque reflejan la realidad operativa, están sujetas a errores de posicionamiento, pérdidas de señal en determinadas zonas geográficas y variaciones en las frecuencias de muestreo según el sistema de vigilancia utilizado. Por su parte, los planes de vuelo, en tanto que representan una intención futura, pueden diferir sustancialmente del comportamiento final debido a múltiples factores impredecibles: condiciones meteorológicas adversas, restricciones temporales del espacio aéreo, conflictos de tráfico o instrucciones específicas del ATC que obligan a modificar el rumbo o la altitud durante el vuelo.

Además, debe tenerse en cuenta la diferencia conceptual fundamental entre los distintos tipos de datos: mientras los planes de vuelo (ya sean iniciales o finales) encapsulan una secuencia de puntos que describen la intención operacional, las trayectorias radar reflejan una ejecución dinámica sujeta a múltiples influencias externas. Esta disparidad hace que las comparaciones directas requieran técnicas robustas capaces de manejar no solo variabilidad estructural, sino también incertidumbre es-

pacial y temporal.

En este contexto, la evaluación automática de la adherencia entre ambas representaciones cobra un valor estratégico. No solo permite identificar patrones recurrentes de desviación y validar modelos de predicción de trayectorias, sino que también abre la puerta a sistemas de alerta temprana, a la monitorización continua del rendimiento del sistema y a la generación de métricas objetivas que respalden decisiones de planificación y gestión. Para lograrlo, es necesario abordar el problema desde una perspectiva que combine el conocimiento operativo con herramientas analíticas avanzadas capaces de lidiar con datos imperfectos y de alto volumen.

1.2. Planteamiento del problema

Como se ha descrito en la sección anterior, existe una necesidad operativa creciente de cuantificar hasta qué punto las trayectorias reales de las aeronaves se ajustan a lo previsto en los planes de vuelo, tanto en términos de eficiencia como de seguridad. Esta comparación entre intención y ejecución permite detectar desviaciones relevantes, validar modelos predictivos y apoyar la toma de decisiones en la gestión táctica del tráfico aéreo. Sin embargo, la elevada complejidad y heterogeneidad de los datos disponibles, derivada de la diversidad de fuentes, formatos y resoluciones temporales dificulta la aplicación directa de métodos tradicionales a gran escala.

El problema que se aborda en este trabajo es el de medir de forma automática y eficiente la similitud entre dos representaciones secuenciales (trayectorias) del comportamiento de una aeronave. Para resolverlo, se propone el desarrollo de un modelo basado en técnicas de aprendizaje profundo que estime una medida de similitud entre dos secuencias de puntos espaciotemporales, sin necesidad de calcular explícitamente distancias geodésicas ni de realizar alineamientos temporales costosos. Esta estimación se basará únicamente en los datos originales (sin transformaciones previas complejas), lo que denominamos secuencias crudas: series ordenadas de coordenadas geográficas (latitud, longitud, altitud) asociadas a marcas temporales, tal como se recogen en los planes de vuelo y en los sistemas de vigilancia radar.

El enfoque adoptado se basa en aprendizaje supervisado porque se dispone de un conjunto de pares de trayectorias cuyo grado de similitud puede ser estimado mediante métricas clásicas como *Dynamic Time Warping*. Estas métricas se utilizan como referencia para entrenar el modelo, que aprenderá a aproximarlas mediante una arquitectura de red neuronal recurrente con atención. Este tipo de arquitectura permite capturar dependencias complejas en secuencias temporales, es robusta frente a pequeñas perturbaciones y escalable a grandes volúmenes de datos, lo que la hace adecuada para este problema.

Se desarrollarán dos variantes del modelo, en función del tipo de comparación que se desea realizar. La primera se centra en el análisis de adherencia entre el plan de vuelo inicial (presentado por la aerolínea antes del despegue) y el plan de vuelo final (que recoge las modificaciones autorizadas por el ATC antes del vuelo). Cabe mencionar que las modificaciones durante o después del vuelo son excepcionales y se limitan a casos específicos como vuelos de emergencia o militares. Este análisis permite estudiar la magnitud y frecuencia de los cambios estructurales introducidos por el sistema de control. La segunda variante evalúa la similitud entre el plan final y la trayectoria radar, lo que permite medir el grado de cumplimiento efectivo de

las autorizaciones y detectar desviaciones que pueden tener implicaciones operativas importantes.

El ámbito del estudio se sitúa en el espacio aéreo español y se apoya en datos reales proporcionados por CRIDA A.I.E. en el marco del proyecto interno DELFOS, cuyo objetivo es mejorar la capacidad de predicción y análisis del tráfico aéreo instrumental. En este contexto, contar con un modelo capaz de estimar rápidamente la adherencia entre lo planificado y lo ejecutado permitirá optimizar procesos de monitorización, validación de predicciones, búsqueda de trayectorias similares y análisis retrospectivo de incidentes.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar y entrenar un modelo basado en redes neuronales recurrentes con mecanismos de atención, capaz de estimar automáticamente la similitud espaciotemporal entre representaciones secuenciales de comportamiento aéreo, utilizando como entrada secuencias originales de coordenadas y tiempos (planes de vuelo y trayectorias radar), con el fin de evaluar la adherencia operativa de manera eficiente y escalable.

1.3.2. Objetivos específicos

- O1: Definir una métrica de referencia que cuantifique la similitud entre pares de trayectorias espacio-temporales, para su uso como objetivo supervisado durante el entrenamiento.
- O2: Construir un conjunto de datos representativo, que incluya planes de vuelo iniciales, planes de vuelo finales y trayectorias radar, correctamente alineados y etiquetados.
- O3: Desarrollar dos variantes del modelo: una para medir la adherencia entre el plan inicial y el plan final (modificaciones pre-vuelo autorizadas por el ATC), y otra para evaluar la desviación entre el plan final y la trayectoria real registrada (cumplimiento operativo).
- O4: Implementar un codificador secuencial con atención que capture dinámicas temporales y espaciales relevantes en las trayectorias, minimizando la pérdida de información contextual.
- O5: Entrenar y validar el modelo con técnicas modernas de aprendizaje profundo y regularización, incluyendo validación cruzada y análisis de sensibilidad.
- O6: Evaluar el rendimiento del sistema en términos de precisión, capacidad de generalización a nuevas rutas y eficiencia computacional frente a métricas tradicionales como *DTW* y *FastDTW*.
- O7: Explorar casos de uso del modelo entrenado, como la búsqueda eficiente de trayectorias similares en grandes bases de datos y la monitorización automática de desviaciones relevantes.

1.4. Estructura del documento

El documento se organiza en cinco capítulos. El Capítulo 2 establece las definiciones teóricas necesarias para comprender la propuesta. En el Capítulo 3 se realiza una revisión de los trabajos relacionados. En el Capítulo 4 se describe la metodología empleada, el diseño del modelo *AdherNet* y la experimentación realizada. En el Capítulo 5 se discuten los resultados obtenidos y diversos casos de uso. Finalmente, el Capítulo 6 recoge las conclusiones y perspectivas futuras.

Capítulo 2

Conceptos previos

En este capítulo se establecen las definiciones fundamentales necesarias para comprender los métodos de comparación existentes entre trayectorias aéreas y el enfoque propuesto en este Trabajo Fin de Máster, que se centra en el desarrollo de métricas de similitud robustas para el análisis de patrones de vuelo en el contexto del tráfico aéreo.

En primer lugar, se define trayectoria aérea continua y discreta (Figura 2.1) y cómo se representa. A continuación, se introducen distintas distancias entre puntos de trayectorias, tanto espaciales como temporales. Por último, se describe el concepto de métrica de similitud de trayectorias y las propiedades deseables que deben cumplir para su uso en aplicaciones del tráfico aéreo.

2.1. Trayectoria

Una trayectoria aérea es la ruta tridimensional (latitud, longitud y altitud) que sigue una aeronave a lo largo del tiempo durante un vuelo. Esta trayectoria puede representarse matemáticamente como una función del tiempo:

$$T(t) = (\text{lat}(t), \text{lon}(t), \text{alt}(t)) \quad \text{para } t \in [t^{\text{inicial}}, t^{\text{final}}],$$

donde:

- $\text{lat}(t)$: latitud de la aeronave en el instante t (grados decimales)
- $\text{lon}(t)$: longitud de la aeronave en el instante t (grados decimales)
- $\text{alt}(t)$: altitud de la aeronave en el instante t (niveles de vuelo)
- $[t_{\text{inicio}}, t_{\text{fin}}]$: intervalo temporal del vuelo

En la práctica, las trayectorias se registran como secuencias discretas de observaciones en instantes específicos. Una *trayectoria discreta* se define como:

$$T = \{p_1, \dots, p_n\},$$

donde cada $p_i = (\text{lat}_i, \text{lon}_i, \text{alt}_i, t_i)$ representa un punto con $(\text{lat}_i, \text{lon}_i, \text{alt}_i)$ como posición espacial y t_i como la marca temporal correspondiente. La trayectoria contiene un total de n puntos, ordenados cronológicamente, es decir, $t_1 < t_2 < \dots < t_n$.

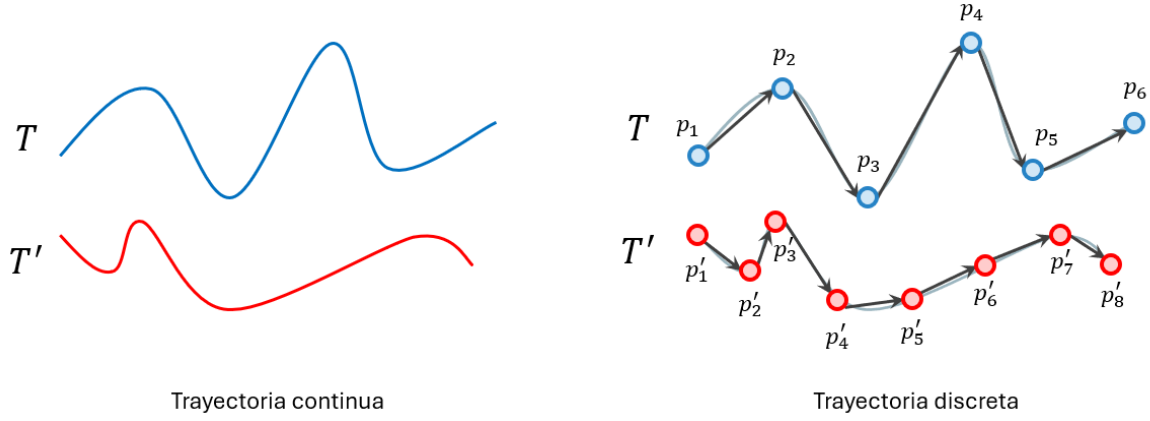


Figura 2.1: Trayectorias continua y discreta

2.2. Distancia entre puntos de trayectorias

La comparación cuantitativa entre trayectorias aéreas requiere definir métricas apropiadas que capturen tanto diferencias espaciales como temporales.

2.2.1. Distancia espacial

Para dos trayectorias $T_1(t)$ y $T_2(t)$ definidas en un intervalo temporal común $[t_a, t_b]$, la distancia espacial en un instante t se calcula como:

$$d_{\text{geodésica}}(T_1(t), T_2(t)) = \sqrt{d_{\text{Haversine}}^2(T_1(t), T_2(t)) + d_{\text{Vertical}}^2(T_1(t), T_2(t))}.$$

La distancia de Haversine corresponde a la componente horizontal y calcula la distancia geodésica (sobre la superficie terrestre) entre dos puntos geográficos, considerando la curvatura de la Tierra R . Matemáticamente se define como:

$$d_{\text{Haversine}}(T_1(t), T_2(t)) = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta \text{lat}}{2} \right) + \cos(\text{lat}_1) \cos(\text{lat}_2) \sin^2 \left(\frac{\Delta \text{lon}}{2} \right)} \right),$$

donde:

- R = radio medio de la Tierra (≈ 6.371 km)
- $\text{lat}_1, \text{lat}_2$ son las latitudes de los puntos en radianes y $\Delta \text{lat} = \text{lat}_2 - \text{lat}_1$
- $\Delta \text{lon} = \text{lon}_2 - \text{lon}_1$ es la diferencia de longitudes en radianes

La distancia vertical mide la separación vertical absoluta entre aeronaves en un instante dado. Matemáticamente se define como:

$$d_{\text{Vertical}}(T_1(t), T_2(t)) = |\text{alt}_1(t) - \text{alt}_2(t)|.$$

2.2.2. Distancia temporal

Las diferencias temporales entre trayectorias se manifiestan de múltiples formas y requieren un análisis multidimensional

Conceptos previos

- **Retraso inicial:** Representa la diferencia entre los tiempos de inicio de los vuelos.

$$d_{\text{inicio}} = |t_1^{\text{inicial}} - t_2^{\text{inicial}}|$$

- **Retraso en el trayecto:** Mide la diferencia en duración total entre los vuelos.

$$d_{\text{duración}} = |t_1^{\text{inicial}} - t_2^{\text{inicial}}| + |t_1^{\text{final}} - t_2^{\text{final}}|$$

- **Dilatación temporal :** se refiere a cómo varía la velocidad a la que se recorren dos trayectorias. Aunque dos trayectorias sigan rutas geográficas similares, pueden diferir en el tiempo que tardan en recorrerlas, lo que implica una compresión o expansión temporal. Para medir esta diferencia, se comparan los puntos equivalentes de ambas trayectorias a lo largo del tiempo. Si una trayectoria va más lenta o más rápida, se dice que el tiempo se ha “estirado” o “comprimido”. En el caso continuo, se pueden utilizar funciones de reparametrización del tiempo; mientras que en el caso discreto, se pueden emplear métodos como *Dynamic Time Warping* (DTW), que ajusta las secuencias temporales para alinearlas y medir cuánto difieren en su evolución.

En la Figura 2.2 se puede ver una representación gráfica de estos conceptos.



Figura 2.2: Distancia temporal

2.3. Métrica de similitud de trayectorias

Una métrica de similitud de trayectorias es una función que cuantifica el grado de semejanza entre dos trayectorias aéreas. Formalmente, se define como:

$$s : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R},$$

donde \mathcal{T} representa el espacio de todas las trayectorias posibles y $s(T_1, T_2)$ indica el valor de similitud entre las trayectorias T_1 y T_2 . Valores altos indican trayectorias más similares, y valores bajos, más diferentes. En la Figura 2.3 se puede ver una representación gráfica de dos trayectorias artificiales en 2D y su grado de similitud. Esta idea se extiende a trayectorias 4d (latitud, longitud, altitud, tiempo).

Comúnmente se normaliza al rango $[0, 1]$ donde 1 = similitud máxima, 0 = similitud mínima.

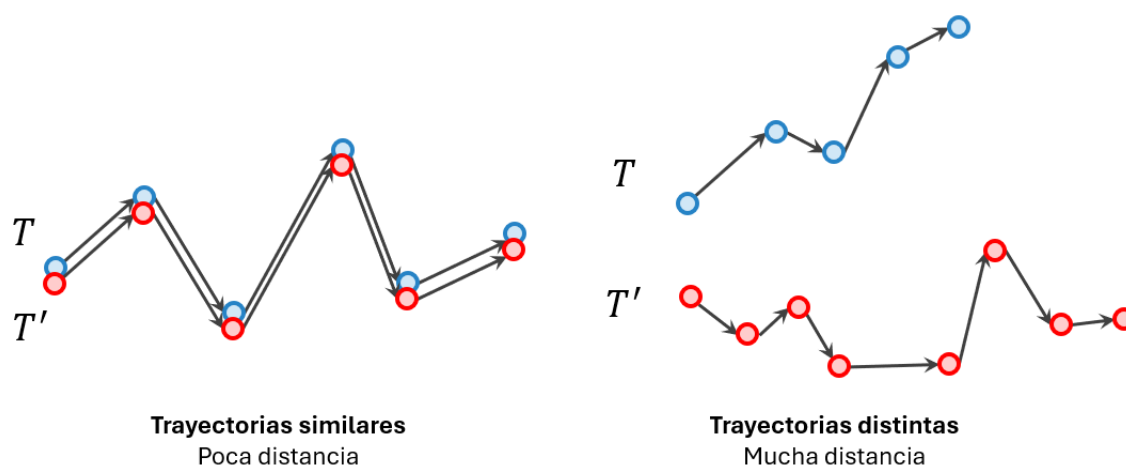


Figura 2.3: Interpretación de la similitud entre trayectorias 2d

Las métricas de similitud están inversamente relacionadas con las métricas de distancia. Si $d(T_1, T_2)$ es una métrica de distancia, una métrica de similitud equivalente puede obtenerse mediante una transformación. En nuestro caso: $s(T_1, T_2) = e^{-\alpha \cdot d(T_1, T_2)}$, donde $\alpha > 0$ es un parámetro de escalado.

2.3.1. Propiedades deseables de las métricas

Para que una métrica de similitud de trayectorias sea útil en el contexto aeronáutico, debe cumplir ciertas propiedades:

1. Robustez ante ruido: Tolerancia a errores de medición en las posiciones radar
2. Invarianza ante muestreo: Independencia de la frecuencia de muestreo de los datos
3. Sensibilidad direccional: Capacidad de distinguir trayectorias con orientaciones diferentes
4. Eficiencia computacional: Escalabilidad para grandes volúmenes de datos
5. Interpretabilidad operativa: Resultados comprensibles para expertos en aviación

Capítulo 3

Estado del arte

Los métodos de similitud de trayectorias se clasifican en tres paradigmas principales: las *métricas tradicionales* basadas en principios geométricos (interpretables pero rígidas ante variaciones de muestreo) y los *métodos de aprendizaje automático*. Los enfoques de aprendizaje automático para similitud de trayectorias se pueden clasificar en dos paradigmas principales: El *trajectory encoding* que genera representaciones vectoriales reutilizables mediante aprendizaje profundo, y el *metric learning* que aprende funciones de similitud directas entre pares de trayectorias. En la Figura 3.1 se presenta una clasificación esquematizada de los métodos estudiados.

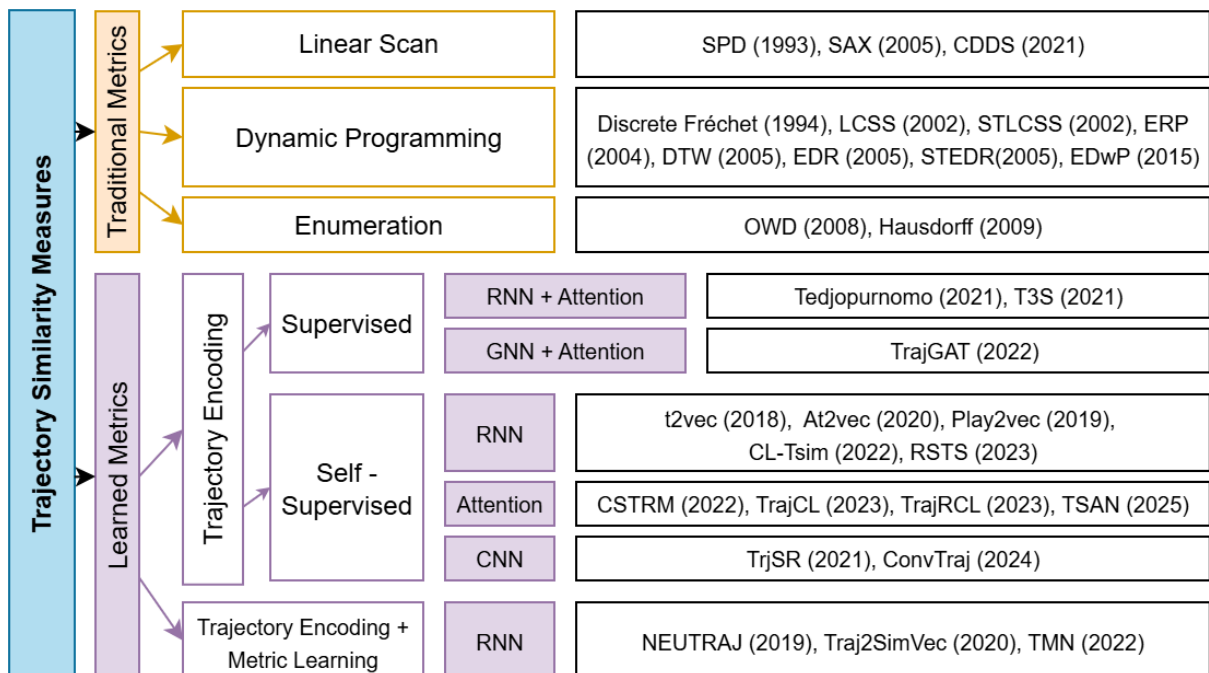


Figura 3.1: Clasificación de Métodos de Similitud de Trayectorias

3.1. Métricas tradicionales de similitud de trayectorias

Las métricas tradicionales emplean enfoques matemáticos y algorítmicos para medir la similitud entre trayectorias basándose en principios geométricos bien establecidos. Operan directamente sobre las coordenadas espaciales sin requerir aprendizaje previo, lo que las hace interpretables y deterministas, pero limitadas en su capacidad para capturar patrones complejos.

En general, comparan pares de puntos de dos trayectorias para encontrar correspondencias óptimas. Las distancias agregadas a partir de los puntos coincidentes formulan la medida de similitud final. La mayoría presentan complejidad $O(n^2)$ y son sensibles a variaciones en las frecuencias de muestreo. Según el mecanismo de correspondencia entre puntos, estas métricas tradicionales pueden agruparse en tres categorías: Escaneo Lineal, Programación Dinámica, y Enumeración. A continuación, se describen brevemente los métodos más representativos de cada categoría.

3.1.1. Escaneo lineal (Linear scan)

Establecen correspondencias directas basándose en la posición secuencial. Complejidad: $O(n)$. Son eficientes pero rígidos ante deformaciones temporales. Algunos métodos representativos de esta categoría son:

- *SPD* [1]: Para cada par de segmentos alineados, evalúa la distancia perpendicular entre el punto medio de un segmento y la línea del otro segmento.
- *SAX* [2] : Convierte las trayectorias en representaciones simbólicas mediante discretización y luego aplica técnicas de comparación de cadenas.
- *CDD* [5]: Comprime las trayectorias como wavelets, permitiendo comparaciones eficientes en términos de almacenamiento y cómputo.

La principal limitación de esta clase de métricas es que no manejan variaciones en velocidad de vuelo ni diferencias de muestreo entre planes y trayectorias radar.

3.1.2. Programación dinámica (Dynamic programming)

Los métodos basados en programación dinámica construyen una matriz donde cada celda (i, j) representa el coste de alinear el punto i de la primera trayectoria con el punto j de la segunda. Luego buscan el camino de menor coste total a través de esta matriz, determinando así la correspondencia óptima entre puntos de ambas trayectorias. Suelen tener complejidad: $O(mn)$. Algunos métodos representativos de esta categoría son:

- *DTW (Dynamic Time Warping)* [20]: Encuentra un alineamiento flexible permitiendo correspondencias múltiples. Maneja diferencias de velocidad, pero tiene un coste computacional elevado.
- *Discrete Fréchet* [15]: Preserva orden temporal estricto, mide distancia máxima entre entidades que recorren trayectorias. Ventaja: Sensible al orden. Limitación: Rígido ante variaciones locales.
- *LCSS* [35]: Subsecuencia común más larga con umbral espacial ϵ . Es tolerante a ruido local, pero requiere ajuste de parámetros.

- *EDR/ERP* [10] [9]: Distancia de edición para secuencias reales. *EDR* cuenta operaciones, *ERP* añade penalizaciones proporcionales. Es flexible ante inserciones/eliminaciones, y sensible a elección de umbral.
- *STEDR/STLCSS* [25]: Versiones espacio-temporales de *EDR* y *LCSS* que incorporan restricciones temporales explícitas. Evalúan tanto la proximidad espacial como la coherencia temporal en el alineamiento, a costa de una mayor complejidad de parámetros.

DTW [20] y su versión acelerada *FastDTW* [28] se han popularizado ampliamente porque ofrecen un buen equilibrio entre flexibilidad, interpretabilidad y rendimiento. A diferencia de otros métodos de programación dinámica, *DTW* permite deformaciones temporales suaves y sin restricciones estrictas, lo que lo hace adecuado para trayectorias con velocidades variables. Además, *FastDTW* reduce drásticamente el coste computacional, haciéndolo viable para aplicaciones en tiempo real o con grandes volúmenes de datos, algo que no ocurre con métodos como *Fréchet* o *LCSS*, que suelen ser más rígidos o sensibles a parámetros. Esta combinación de adaptabilidad y eficiencia explica su uso extendido frente a otras alternativas.

La distancia *DTW* [20] es una métrica diseñada para series temporales y puede aplicarse a datos de trayectorias. *DTW* permite alinear un punto de una trayectoria con uno o más puntos consecutivos de otra, lo que resulta útil en trayectorias con velocidades variables.

La complejidad temporal de *DTW* es $\mathcal{O}(mn)$, donde m y n son las longitudes de las trayectorias comparadas respectivamente. Aunque *DTW* no es una métrica en sentido estricto, es ampliamente utilizada para medir similitud entre trayectorias.

La distancia *DTW* entre dos trayectorias T_1 y T_2 se define recursivamente como [30]:

$$DTW(T_1, T_2) = \begin{cases} 0 & \text{si } m = n = 0 \\ \infty & \text{si } m = 0 \text{ o } n = 0 \\ \text{dis}(T_1^m, T_2^n) + \min \begin{cases} DTW(\text{Head}(T_1), T_2) \\ DTW(T_1, \text{Head}(T_2)) \\ DTW(\text{Head}(T_1), \text{Head}(T_2)) \end{cases} & \text{en otro caso} \end{cases}$$

donde $\text{Head}(\cdot)$ obtiene todos los puntos de la trayectoria excepto el último, es decir:

$$\text{Head}(T) = \langle p_1, p_2, \dots, p_{n-1} \rangle.$$

Debido a la alta complejidad computacional de *DTW* en su forma tradicional ($\mathcal{O}(mn)$), se han desarrollado algoritmos aproximados como *FastDTW* [28]. Este método reduce significativamente el costo computacional a $\mathcal{O}(n)$ en tiempo y espacio, mediante un enfoque de reducción de resolución y refinamiento progresivo.

FastDTW permite encontrar una alineación cercana a la óptima sin necesidad de calcular la matriz de distancias completa, lo que lo hace adecuado para comparar trayectorias largas o para aplicaciones en tiempo real donde la eficiencia es crítica. Aunque se sacrifica una pequeña cantidad de precisión, el compromiso entre exactitud y velocidad suele ser aceptable en la práctica.

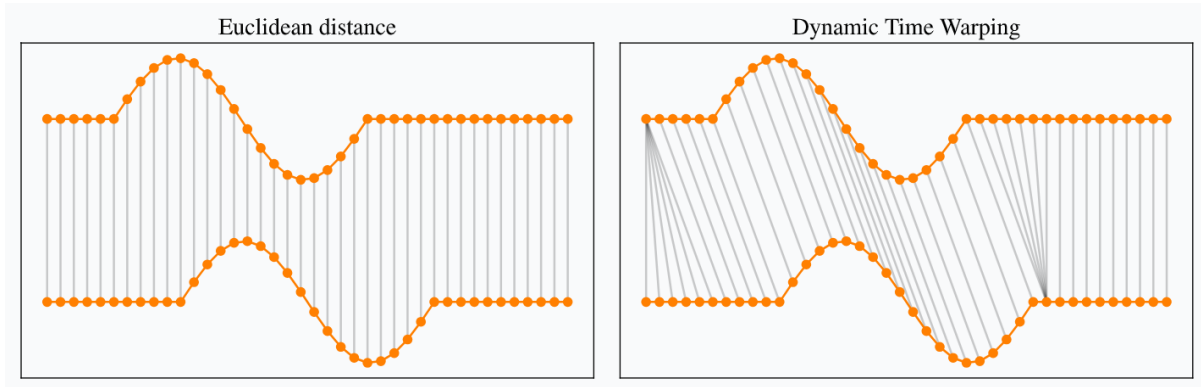


Figura 3.2: Alineamiento de puntos con distancia euclídea y dtw. Imagen de Romain Tavenard [31]

3.1.3. Enumeración (Enumeration)

Se consideran múltiples combinaciones o permutaciones de segmentos de trayectoria para encontrar la mejor correspondencia. Dan la mejor coincidencia posible, pero su complejidad computacional puede ser prohibitiva en conjuntos de datos grandes.

- *Hausdorff* [26]: Calcula la distancia máxima punto-trayectoria. Detecta desviaciones críticas, pero es muy sensible a outliers.
- *OWD* [22]: Versión dirigida de Hausdorff.

3.2. Limitaciones de métricas tradicionales ante datos reales

Los métodos tradicionales asumen tasas de muestreo uniformes, pero la realidad es diferente. En aviación, los datos provienen de fuentes diversas: planes de vuelo (waypoints estratégicos), radar (alta frecuencia), *ADS-B* (frecuencia variable), y *ACARS* (eventos específicos). Cada sistema tiene configuraciones diferentes, las aeronaves optimizan transmisiones para ahorrar energía, y la cobertura varía según la región.

Un vuelo típico muestra esta variabilidad: Un plan de vuelo Madrid-Barcelona puede contener entre 5 y 20 puntos con una separación no homogénea que puede ser superior a los $150km$ en algunos casos. La traza radar real del mismo vuelo tiene 180 puntos separados por $3km$. Esta heterogeneidad requiere normalización para comparaciones válidas. La Figura 3.3 representa esta casuística.

Cuando *DTW* intenta alinear 5 waypoints con 180 puntos radar, produce resultados distorsionados. Los waypoints únicos se corresponden con decenas de puntos radar, creando falsas “paradas” en ubicaciones que son solo referencias de navegación. Los segmentos largos se comprimen artificialmente, y las variaciones reales de velocidad se confunden con artefactos de muestreo diferente.

3.2.1. Necesidad de interpolación

La interpolación espacial constituye la alternativa principal para abordar las densidades dispares en datos aeronáuticos. Aunque existen otros enfoques como métri-

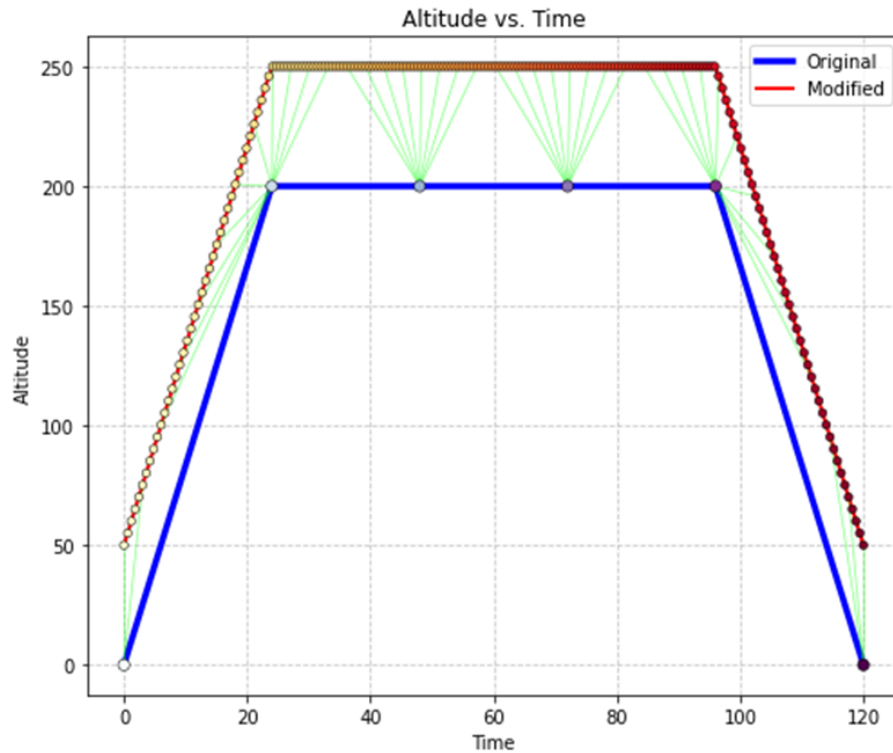


Figura 3.3: Alineación DTW con diferentes frecuencias de muestreo

cas geométricas (*Hausdorff*, *Fréchet*), representaciones vectoriales, submuestreo inteligente y métricas estadísticas, estos presentan limitaciones críticas. Las métricas geométricas pierden información temporal esencial, las representaciones vectoriales reducen la interpretabilidad, y el submuestreo elimina datos valiosos. Por tanto, la interpolación emerge como la solución más equilibrada al normalizar densidades heterogéneas preservando la estructura temporal y la interpretabilidad operacional.

EDwP [27] aborda estas limitaciones mediante proyecciones dinámicas que eliminan la necesidad de interpolación fija. En lugar de crear puntos artificiales, permite que cualquier punto del segmento planificado se alinee óptimamente con puntos radar mediante proyección continua. Este enfoque logra hasta 50 % mejor precisión que métodos tradicionales y se adapta automáticamente a diferentes densidades sin ajustes manuales.

Sin embargo, según la documentación técnica, *EDwP* tiene una complejidad computacional cuadrático. Para comparar un plan típico (5 puntos) con radar (180 puntos), *EDwP* requiere $(5+180)^2 = 185^2 = 34,225$ operaciones por trayectoria, lo que se traduce en unas 8-9 horas para procesar 1000 vuelos a una velocidad de aproximadamente 1,000-1,200 operaciones por segundo.

Este estudio acepta métodos con menor adaptabilidad que *EDwP* y precisión reducida ($\sim 85\%$ vs $\sim 95\%$) como compromiso necesario al procesar volúmenes de 1,000 a 50,000 trayectorias. La investigación identifica alternativas como *FastDTW* como óptimas para estos volúmenes masivos, priorizando eficiencia computacional e interpretabilidad sobre máxima precisión teórica, manteniendo fidelidad suficiente para detectar patrones temporales significativos en análisis a gran escala.

3.3. Codificación profunda de trayectorias (Trajectory Encoding)

Los métodos de *trajectory encoding* se centran en generar representaciones vectoriales (embeddings) individuales para cada trayectoria, donde la función de codificación $f_e : \mathcal{D} \rightarrow \mathbb{R}^d$ mapea cada trayectoria a un espacio de embeddings de dimensión d . La similitud entre dos trayectorias se calcula posteriormente mediante métricas simples en el espacio de embeddings, incluyendo *distancia Manhattan* (L1), *distancia Euclidea* (L2), *similitud coseno*, *distancia Chebyshev*, o *distancia Minkowski* [7, 34].

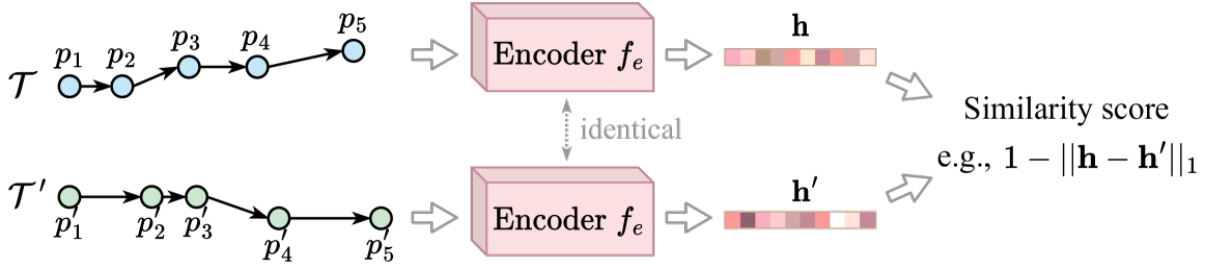


Figura 3.4: Codificación Profunda De Trayectorias. Imagen: Chang et al. [7]

Los métodos de codificación profunda de trayectorias se dividen en autosupervisados y supervisados, según arquitecturas como *RNN*, *Transformers*, *CNN* y *GNN*. En esta sección, primero se presentan los métodos de codificación profunda de trayectorias, divididos en modelos *autosupervisados*, que integran técnicas como el *aumento de datos* y el *aprendizaje contrastivo* para mejorar la precisión de la comparación, y modelos *supervisados*, que destacan en clasificación, estimación de tiempo de viaje y *linking*, donde las etiquetas permiten optimizar objetivos específicos y obtener resultados interpretables. [11]

3.3.1. Modelos de codificación autosupervisados basados en RNN

La naturaleza secuencial de las *Redes Neuronales Recurrentes* (RNN) las hace particularmente adecuadas para procesar trayectorias aéreas, que esencialmente son secuencias ordenadas de puntos geoespaciales con dependencias temporales. Las RNN generan un vector de representación (*embedding*) para cada trayectoria procesando punto a punto, y acumulando contexto mediante su memoria interna. Nótese que las RNN tradicionales presentan limitaciones importantes como el problema del *desvanecimiento del gradiente*, lo que motiva el uso de variantes como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*) [12, 19].

A lo largo de los últimos años se han propuesto diversos modelos basados en RNN para medir similitud de trayectorias:

- *T2Vec (2018)* [21] introduce una representación vectorial fija para cada trayectoria, utilizando una función de pérdida sensible a la proximidad espacial. Su arquitectura basada en GRU permite capturar la forma y dirección de la trayectoria, mientras que su función de pérdida penaliza errores entre trayectorias cercanas. El modelo utiliza una función de pérdida que considera tanto la distancia espacial como la reconstrucción de la trayectoria original.

- *At2Vec (2020)* [17] extiende *T2Vec* para incorporar información de actividades en trayectorias con puntos de interés (POI). El modelo añade información de actividades a un codificador-decodificador para obtener representaciones más ricas que capturan tanto patrones espaciales como semánticos de las trayectorias.
- *Play2Vec (2019)* [36] adapta la arquitectura *T2Vec* para el análisis específico de jugadas deportivas, demostrando la versatilidad del enfoque de encoding para dominios especializados.
- *CL-Tsim (2022)* [13] genera muestras diversas mediante aumentación (*down-sampling/perturbación*). Utiliza *trajectory encoding* maximizando similitud entre pares positivos, minimizando entre negativos. Contribución: primer método contrastivo puro para similitud de trayectorias. Entrada: trayectorias 2D; salida: *embeddings* robustos con diversidad de muestras.
- *RSTS (2023)* [39] extiende *t2vec* mediante un enfoque autosupervisado con arquitectura RNN basada en *encoder-decoder* GRU. Introduce una cuadrícula 3D donde la tercera dimensión modela tiempo, utilizando *trajectory encoding* para generar *embeddings* de dimensión fija. Su contribución principal es la primera extensión sistemática de métodos 2D al dominio espaciotemporal. Procesa trayectorias 3D (latitud, longitud, tiempo) y produce vectores de representación que preservan similitud espaciotemporal.

3.3.2. Modelos de codificación autosupervisada basados en Atención

Los mecanismos de autoatención permiten capturar relaciones a largo plazo de manera más eficiente que las RNN, siendo especialmente beneficiosos para trayectorias largas al asignar diferentes pesos a partes de la trayectoria.

- *CSTRM (2022)* [23] emplea una arquitectura *Transformer* ligera con enfoque de *two-view learning*, donde dos versiones diferentes de una trayectoria pasan por codificadores simétricos. Incorpora un *projection head* para mapear a un espacio específico donde se aplica la pérdida contrastiva, considerando tanto componentes espaciales como temporales. El modelo utiliza *BERT* [14] con solo 3 capas en lugar de las 12 tradicionales para mejorar la eficiencia.
- *TrajCL (2023)* [4] introduce aprendizaje contrastivo autosupervisado mediante aumentaciones específicas para trayectorias, incluyendo eliminación de puntos (*point masking*), desplazamiento de puntos (*point shifting*), truncamiento de trayectorias (*trajectory truncating*) y simplificación de trayectorias (*trajectory simplification*). Utiliza un codificador base *Transformer* con un módulo de autoatención dual (*DualSTB*) que captura características estructurales y espaciales, y optimiza con pérdida contrastiva *InfoNCE* para maximizar la similitud entre vistas aumentadas de la misma trayectoria.
- *TrajRCL (2023)* [6] combina aprendizaje contrastivo con reconstrucción de datos mediante una función de pérdida dual que integra reconstrucción y contraste. Incluye un módulo reconstructor que intenta recuperar la trayectoria original desde el *embedding* codificado, forzando representaciones más ricas que preservan detalles estructurales.
- *TSAN (2025)* [16] propone la primera red de aprendizaje de representaciones específicamente diseñada para similitud de trayectorias de vuelo con arquitectura

3.3. Codificación profunda de trayectorias (Trajectory Encoding)

RNN+Attention self-supervised. Combina tres módulos principales: augmentación de trayectorias (TAM), extracción de características (FEM) y codificación de trayectorias (TEM). Utiliza TCNs con convoluciones dilatadas para capturar características espaciotemporales de granularidad fina, seguidas de mecanismo de atención *Siamese* que facilita interacción entre vectores de características. Aborda específicamente el ruido en datos de vuelo mediante augmentación con segmentación y distorsión, optimizado para entornos 3D complejos de redes de tráfico aéreo. Emplea función de pérdida triplet maximizando distancia entre trayectorias disímiles. Entrada: coordenadas 3D + sensores aeronáuticos QAR; salida: *embeddings* espaciotemporales específicos de vuelo.

3.3.3. Modelos de codificación autosupervisada basados en CNN

Las *Redes Neuronales Convolucionales* (o CNN por sus siglas en inglés) ofrecen ventajas en términos de velocidad y eficiencia durante el entrenamiento, aunque pueden presentar limitaciones para capturar adecuadamente la dinámica temporal inherente en las trayectorias.

- *TrajSR (2021)* [3] emplea arquitecturas convolucionales para el procesamiento de trayectorias, aprovechando la eficiencia computacional de las CNN mediante la conversión de trayectorias en representaciones tipo imagen. El modelo utiliza 13 capas convolucionales apiladas, lo que resulta computacionalmente costoso pero permite capturar patrones espaciales en las trayectorias.
- *ConvTraj (2024)* [8] usa una CNN híbrida autosupervisada con convoluciones 1D y 2D para representar trayectorias como imágenes binarias. Supera a métodos basados en RNN y *Transformers* enfocándose en similitud local, con solo el 5% de sus parámetros y una velocidad 240x mayor en entrenamiento. Genera *embeddings* que preservan similitud local y global en trayectorias 2D.

3.3.4. Modelos de codificación supervisados basados en RNN y Atención

Las *Redes Neuronales Recurrentes* (RNNs) están diseñadas para procesar datos secuenciales, como trayectorias, donde el orden temporal es relevante. Sin embargo, presentan limitaciones, como la incapacidad de capturar relaciones espaciales globales. Por ello, se integran mecanismos de atención que permiten al modelo enfocarse en partes relevantes de la secuencia sin depender únicamente del orden.

- *T3S (2021)* [38] propone un enfoque de aprendizaje supervisado que utiliza un codificador dual basado en GRU para generar representaciones de trayectorias. El modelo emplea *triplet loss* para optimizar el espacio latente, minimizando la distancia entre trayectorias similares y maximizándola entre trayectorias diferentes, incluyendo un mecanismo de muestreo estratégico para generar tripletas de entrenamiento relevantes.
- *Tedjopurnomo (2021)* [33] usa una arquitectura *RNN+Attention* usando *GRU autoencoder* y *self-attention* con tres funciones de pérdida: representación, punto-a-punto y patrón para aprendizaje multi-nivel. Su contribución es el *framework* multiobjetivo que aprende similitud a diferentes granularidades simultáneamente. Combina *encoding* y *metric learning* procesando trayectorias *2D+tiempo*, generando *embeddings* fijos.

3.3.5. Modelos de codificación supervisados basados en GNN y Atención

Los enfoques basados en *Redes Neuronales de Grafos* modelan las trayectorias como estructuras de grafo, permitiendo capturar relaciones espaciales complejas entre puntos de la trayectoria.

- *TrajGAT (2022)* [41] utiliza mecanismos de atención en grafos (*Graph Attention Networks*) para modelar las relaciones entre puntos de trayectorias, capturando tanto la estructura espacial como las dependencias entre diferentes segmentos de la trayectoria. Aunque permite modelar relaciones espaciales complejas, puede ser computacionalmente costoso y requiere preprocesamiento adecuado para estructurar los datos en forma de grafo.

3.4. Aprendizaje de métricas para trayectorias (Metric learning)

Los métodos de *trajectory encoding + metric learning* extienden el enfoque de codificación individual combinando la función $f_e : \mathcal{D} \rightarrow \mathbb{R}^d$ con el aprendizaje simultáneo de una función de similitud paramétrica $f_s(\cdot, \cdot)$. Esta aproximación dual optimiza tanto la representación vectorial como la medida de distancia de forma integrada, superando las limitaciones de métricas fijas mediante funciones adaptativas específicas del dominio de trayectorias.

3.4.1. Métodos de aprendizaje de métricas basados en RNN

Estos métodos, además de aprovechar arquitecturas como GRU y LSTM, las combinan con funciones de pérdida diseñadas para mejorar la calidad de los *embeddings*. Algunas propuestas integran módulos de matching que aprenden a replicar técnicas clásicas como DTW, fusionando aprendizaje profundo con métodos tradicionales.

- *NEUTRAJ (2019)* [40] evita el uso de DTW mediante un enfoque guiado por “semillas” (puntos de referencia). Utiliza GRU con una unidad de memoria atencional y una función de pérdida por *ranking*, mejorando la eficiencia y la calidad de las representaciones.
- *Traj2SimVec (2020)* [42] mejora NEUTRAJ con una arquitectura LSTM supervisada. Innovaciones: selección eficiente via *k-d trees* y pérdida subtrayectorias para alineamiento punto a punto. Emplea *encoding y metric learning* con supervisión auxiliar. Contribución: *matching* óptimo de subtrayectorias y muestreo inteligente mejorando rendimiento semántico/geométrico. Entrada: trayectorias 2D; salida: *embeddings* completos y *alignments*.
- *TMN (2022)* [37] (*Trajectory Matching Network*) utiliza arquitectura RNN *supervised dual-branch* con LSTM para aprender *embeddings* y *matching* de puntos simultáneamente. Su módulo de *matching* simula métodos no aprendidos tradicionales como DTW/EDR con parámetros aprendibles. Principal contribución: puente entre métodos clásicos y *deep learning* mediante *matching* explícito. Requiere pares de trayectorias 2D como entrada, produce *scores* de similitud directos y alineamientos punto-a-punto.

3.5. Análisis comparativo de métodos de similitud de trayectorias

Los métodos de similitud de trayectorias se pueden clasificar en tres enfoques principales: las *métricas tradicionales* basadas en principios geométricos, que ofrecen interpretabilidad pero son poco robustas ante diferencias de muestreo; los métodos de *trajectory encoding*, que emplean redes neuronales para generar representaciones vectoriales reutilizables; y el *metric learning*, que entrena directamente funciones de similitud sobre pares de trayectorias. En este apartado se presenta un análisis comparativo de estos enfoques basado en resultados experimentales previamente publicados en la literatura [7, 11, 30, 34], sin que constituyan una contribución original de este estudio.

Las métricas tradicionales pueden subdividirse según el mecanismo de comparación empleado. La Tabla 3.1 recoge una comparación entre técnicas clásicas como los métodos de escaneo lineal, los enfoques basados en programación dinámica, métodos de enumeración y versiones optimizadas como *FastDTW*. En términos de complejidad computacional, las técnicas lineales son las más eficientes ($\mathcal{O}(n)$), pero presentan una alta rigidez ante deformaciones temporales. Por su parte, los métodos basados en programación dinámica, aunque flexibles, son computacionalmente costosos ($\mathcal{O}(mn)$). *FastDTW* busca mitigar esta desventaja con un compromiso entre eficiencia y precisión. Dentro de este grupo, *EDwP* destaca por su capacidad para mantener correlaciones de ranking elevadas (*Spearman* >0.75) incluso en presencia de muestreos no uniformes, y por su mejor rendimiento en tareas de clasificación multiclase sobre datasets como *ASL*, superando a métricas como *EDR* y *LCSS* [27].

Leyenda: n = longitud de la trayectoria, m = longitud de la segunda trayectoria, d = dimensión del *embedding*

Método	Complejidad	Ventajas	Limitaciones
Escaneo Lineal (SPD, SAX, CDDS)	$\mathcal{O}(n)$	Eficientes	Rígidos ante deformaciones temporales
Programación Dinámica (DTW, Fréchet, LCSS)	$\mathcal{O}(mn)$	Flexibles, manejan diferencias de velocidad	Alto coste computacional
Enumeración (Hausdorff, OWD)	$\mathcal{O}(mn)$	Mejor coincidencia posible	Complejidad prohibitiva
<i>FastDTW</i>	$\mathcal{O}(n)$	Eficiencia optimizada	Compromiso eficiencia-precisión

Tabla 3.1: Comparativa de métodos de alineamiento temporal

En contraste, los enfoques basados en aprendizaje profundo permiten modelar relaciones más complejas y no lineales entre trayectorias. Las arquitecturas basadas en RNN (recogidas en la Tabla 3.2) como *T2Vec*, *NEUTRAJ* o *Traj2SimVec* utilizan variantes de GRU y mecanismos de atención espacial. Las RNN ofrecen una ventaja clave: su capacidad para capturar dependencias temporales de largo plazo. Procesan cada punto de la trayectoria secuencialmente, acumulando contexto y generando un *em-*

bedding final que representa la trayectoria completa. Esta propiedad las hace idóneas para escenarios donde el orden temporal es crítico. Además, su estructura compacta permite un entrenamiento más estable en escenarios con menos datos. No obstante, su mayor debilidad es la eficiencia: la naturaleza secuencial de las RNN impide una paralelización efectiva, lo que ralentiza tanto el entrenamiento como la inferencia, especialmente en trayectorias largas. Su complejidad temporal típica es $\mathcal{O}(nd^2)$, lo que limita su escalabilidad.

Método	Año	Características Principales	Limitaciones	Complejidad
<i>T2Vec</i>	2018	GRU + función de pérdida espacial, primer enfoque DL para trayectorias	Procesamiento secuencial lento, representaciones limitadas	$\mathcal{O}(nd^2)$
<i>NEUTRAJ</i>	2019	Enfoque por "semillas", <i>speedup</i> 50x-1000x, módulo de atención espacial	Escalabilidad limitada, módulo espacial costoso	$\mathcal{O}(nd^2)$
<i>Traj2SimVec</i>	2020	<i>k-d trees</i> + pérdida sub-trayectorias, muestreo eficiente	Mejora 25% en HR@10 vs <i>NEUTRAJ</i> , dependencia de estructuras auxiliares	$\mathcal{O}(nd^2)$
<i>TMN</i>	2022	Arquitectura <i>dual-branch</i> , modelo de coincidencia de puntos	No genera embeddings individuales, requiere pares de entrada	$\mathcal{O}(nd^2 + n^2d)$

Tabla 3.2: Comparativa de métodos basados en RNN

Por otro lado, los modelos basados en *Transformers* y mecanismos de atención (Tabla 3.3) como *T3S*, *TrajCL*, *CSTRM* o *TrajRCL* presentan un cambio de paradigma. Estos métodos procesan todas las posiciones de la trayectoria simultáneamente, lo que permite una paralelización masiva en *hardware* como GPUs. Aunque su complejidad teórica es mayor ($\mathcal{O}(n^2d)$), en la práctica resultan más eficientes que las RNN para trayectorias largas. Los mecanismos de atención tienen la capacidad de capturar relaciones globales entre puntos lejanos en la trayectoria, no limitándose a vecindades temporales locales. Esto se traduce en representaciones más expresivas, que capturan tanto la estructura global como las peculiaridades locales de la trayectoria. En particular, modelos como *TrajCL* emplean estrategias contrastivas auto-supervisadas sin recurrir a RNNs, y muestran una notable capacidad de generalización entre dominios. *TrajRCL*, por su parte, combina atención con codificaciones topológicas multi-escala mediante curvas de *Hilbert* y pérdidas duales (contrastiva y reconstrucción), mejorando la calidad de los embeddings para tareas *downstream* como clasificación o *clustering*.

3.5. Análisis comparativo de métodos de similitud de trayectorias

Método	Año	Innovación Principal	Ventajas	Complejidad
T3S	2021	Triplet loss + codificador dual GRU, combinación <i>self-attention</i> y LSTM	Optimización del espacio latente, características topológicas	$\mathcal{O}(n^2d)$
CSTRM	2022	Two-view learning + Transformer ligero, Masked Language Modeling	Eficiencia computacional, aprendizaje auto-supervisado	$\mathcal{O}(n^2d)$
TrajCL	2023	Augmentaciones específicas + InfoNCE, encoder puramente self-attention	Mejor generalización <i>cross-dataset</i> , sin dependencia RNN	$\mathcal{O}(n^2d)$
TrajRCL	2023	Pérdida dual: contrastiva + reconstrucción, codificación Hilbert	Representaciones más ricas, estructura espacial multi-escala	$\mathcal{O}(n^2d)$

Tabla 3.3: Comparativa de métodos basados mecanismos de atención

Finalmente, la Tabla 3.4 recoge métodos basados en *CNNs* y *GNNs*, que aunque menos frecuentes, exploran representaciones espaciales alternativas. *TrajSR* emplea redes convolucionales profundas para tareas de reconstrucción con super-resolución, pero pierde información direccional y secuencial. Por su parte, *TrajGAT*, basado en grafos y atención, requiere una compleja fase de preprocesamiento estructural (*quad-tree* multinivel) y presenta un alto coste computacional ($\mathcal{O}(nn_{ed})$).

Método	Año	Características Principales	Limitaciones	Complejidad
TrajSR (CNN)	2021	13 capas convolucionales, reconstrucción super-resolución	Pierde información secuencial, no distingue direcciones	$\mathcal{O}(mk^2n_kc)$
TrajGAT (GNN)	2022	Mecanismos de atención en grafos, <i>quad-tree</i> multinivel	Alto costo computacional, preprocesamiento complejo	$\mathcal{O}(nn_ed)$

Tabla 3.4: Comparativa de métodos basados en CNN y GNN

En conclusión, las métricas clásicas ofrecen eficiencia y simplicidad, pero presentan limitaciones estructurales. Las RNNs son útiles en contextos donde la secuencia estricta es esencial, pero sufren de baja eficiencia en escalas grandes. En cambio, los modelos basados en atención, especialmente aquellos que combinan mecanismos de *self-attention* y atención cruzada, representan una alternativa más eficiente y precisa para medir similitud entre trayectorias, gracias a su paralelización, expresividad y capacidad de generalización.

Capítulo 4

Metodología

La metodología seguida se compone de varias fases claramente diferenciadas:

1. Recopilación de datos: se obtienen trayectorias planificadas y reales de vuelos a través de fuentes internas (CFMU y FLOWS).
2. Preprocesamiento: se normalizan los datos y se filtran vuelos con información incompleta.
3. Etiquetado: se interpolan las trayectorias a intervalos regulares, se calculan métricas de similitud entre trayectorias mediante *DTW geodésico* y se generan etiquetas supervisadas.
4. Construcción del modelo: se diseña una arquitectura de red neuronal que incluye codificación secuencial, atención y comparación de embeddings.
5. Entrenamiento: se entrena el modelo con técnicas de regularización, optimización adaptativa y evaluación continua.
6. Evaluación y análisis: se analizan los resultados del modelo sobre datos de test, comparando predicciones y métricas reales.

Aunque los dos primeros pasos (recopilación y preprocesamiento de datos) son fundamentales para el correcto funcionamiento del sistema, se han pospuesto en la estructura del documento y se explicarán con más detalle en la sección de resultados (capítulo 5), donde se contextualizan mejor al presentar los datos utilizados. Por ello, este capítulo comienza directamente en el paso 3, centrado en el método de etiquetado.

4.1. Método de etiquetado

En esta sección se describe el proceso de etiquetado de trayectorias de vuelo, que consiste en calcular medidas de similitud espacio-temporales para generar etiquetas supervisadas que alimentan el modelo. Se detallan las métricas de similitud espacial y temporal, y su combinación en una métrica global.

La métrica utilizada se calcula utilizando una distancia basada en *Dynamic Time Warping (DTW)*, que permite alinear trayectorias con velocidades variables y muestreos irregulares. Más concretamente, se utiliza una versión del algoritmo *FastDTW*,

que utiliza distancias geodésicas, y que recibe trayectorias interpoladas. Estas trayectorias se interpolan con el objetivo aumentar la precisión del algoritmo *FastDTW* (ver Sección 3.2.1).

Para el etiquetado, se han interpolado puntos adicionales en cada trayectoria a intervalos regulares de 5 km mediante interpolación lineal paramétrica. El valor de 5 km se ha elegido empíricamente tras observar que distancias mayores generaban trayectorias demasiado dispersas para *DTW*, mientras que valores menores incrementaban el tiempo de cómputo sin beneficios claros. No es un parámetro sagrado, pero sí razonable para el compromiso precisión-coste computacional.

Dado un par de puntos consecutivos:

$$P_i = (t_i, \text{lat}_i, \text{lon}_i, \text{alt}_i), \quad P_{i+1} = (t_{i+1}, \text{lat}_{i+1}, \text{lon}_{i+1}, \text{alt}_{i+1}).$$

Si $d_{\text{geodésica}}(P_i, P_{i+1}) > 5 \text{ km}$, se calcula el número de puntos a interpolar:

$$n = \left\lfloor \frac{d_{\text{geodésica}}(P_i, P_{i+1})}{5 \text{ km}} \right\rfloor.$$

Los puntos intermedios se generan mediante interpolación lineal paramétrica con fracciones equiespaciadas $f_j = \frac{j}{n+1}$ para $j = 1, 2, \dots, n$:

$$\vec{X}_j = \vec{X}_i + f_j \times (\vec{X}_{i+1} - \vec{X}_i) \quad \text{donde} \quad \vec{X} = \begin{bmatrix} t \\ \text{lat} \\ \text{lon} \\ \text{alt} \end{bmatrix}.$$

Esta formulación vectorizada garantiza que los puntos interpolados mantengan proporcionalidad temporal y espacial, distribuyéndose uniformemente a lo largo del segmento original. El proceso se aplica a ambas trayectorias originales T_1 y T_2 , resultando en las trayectorias interpoladas espacialmente T'_1 y T'_2 con mayor densidad de puntos para una comparación *DTW* más precisa.

4.1.1. Similitud espacial

Para medir la similitud espacial, se ha utilizado el algoritmo *FastDTW* de la librería *fastdtw* [28], el cual recibe la función de distancia geodésica $d_{\text{geodésica}}$ (necesaria para trayectorias aéreas) como parámetro `dist`:

```
1 def fastdtw_geodesic(t1, t2):
2     dist, path = fastdtw(t1, t2, dist=ta_spatial_dist)
```

Así se obtiene la distancia espacial, $d_{\text{DTW}} = \text{FastDTW}_{\text{Geo}}(T'_1, T'_2)$, donde T'_1 y T'_2 son trayectorias interpoladas espacialmente.

A continuación, se transforma esta distancia en una medida de similitud normalizada, según la fórmula $\text{sim}_{\text{Esp}} = e^{-\beta |d_{\text{DTW}}|}$, donde d_{DTW} es la distancia calculada mediante *FastDTW* geodésico, y β es un parámetro de escalado que controla la sensibilidad de la función exponencial.

4.1.2. Similitud temporal

Para considerar los tres factores de diferencia temporal entre trayectorias, es decir d_1 : retraso inicial, d_2 : retraso del trayecto, y d_3 : dilatación temporal, la distancia temporal total se calcula siguiendo un enfoque por pesos:

$$d_T = w_1 \times d_1 + w_2 \times d_2 + w_3 \times d_3,$$

donde w_1 , w_2 y w_3 son los pesos asignados a cada componente.

Posteriormente, la distancia temporal se transforma en similitud temporal según la fórmula: $\text{sim}_{\text{Tem}} = e^{-\alpha|d_T|}$, donde α es un parámetro de ajuste.

Mientras que d_1 : retraso inicial y d_2 : diferencia en duración del trayecto, se calculan sencillamente con las fórmulas expuestas en la sección 2.2.2, la dilatación temporal se estima a partir del alineamiento entre dos trayectorias utilizando *FastDTW*.

Para medir cuán lejos se desvía la alineación del camino diagonal (que representaría una evolución temporal perfectamente sincronizada), se calcula el índice de dilatación temporal idil como el cociente entre la longitud del camino óptimo (path) y la longitud máxima entre las trayectorias t_1 y t_2 :

$$\text{idil} = \frac{\text{len}(\text{path})}{\max(\text{len}(t_1), \text{len}(t_2))}, \quad \text{idil} \in [1, 1.5].$$

Este valor refleja cuántos pasos adicionales (alineamientos no diagonales) se han introducido para lograr la mejor correspondencia entre trayectorias, y pretende cuantificar la dilatación (ver Figura 4.1). Para convertir este índice en una medida interpretable en unidades temporales, se utiliza la siguiente fórmula:

$$\text{Dilatación temporal} = 2 \times (\text{idil} - 1) \times \text{duración media de las 2 trayectorias}$$

De esta manera, se obtiene una medida proporcional al grado de compresión o expansión temporal entre las trayectorias comparadas, normalizada sobre su duración conjunta.

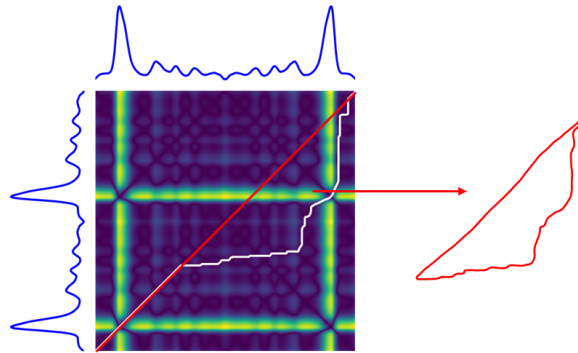


Figura 4.1: Dilatación temporal en la matriz de costes. Más área equivale a más dilatación. Imagen original: Romain Tavenard [32]

4.1.3. Similitud de trayectorias

Una vez calculadas por separado las similitudes espacial y temporal, es necesario combinarlas para obtener una medida global que refleje tanto las diferencias geométricas como las variaciones temporales entre trayectorias de vuelo.

La similitud total entre dos trayectorias se define como una combinación ponderada de ambas componentes:

$$\text{sim}_{\text{Total}} = \gamma \times \text{sim}_{\text{Esp}} + (1 - \gamma) \times \text{sim}_{\text{Tem}},$$

donde $\gamma \in [0, 1]$ controla el peso relativo entre las dimensiones espacial y temporal. Sustituyendo las expresiones previamente definidas:

$$\text{sim}_{\text{Total}} = \gamma \times e^{-\beta|d_{\text{DTW}}|} + (1 - \gamma) \times e^{-\alpha|d_T|},$$

donde d_{DTW} es la distancia calculada mediante *FastDTW* geodésico y d_T la distancia temporal combinada.

Esta expresión define una métrica de similitud flexible y continua, capaz de capturar tanto las desviaciones espaciales como las variaciones temporales entre trayectorias. No obstante, su efectividad depende en gran medida de una correcta parametrización.

El conjunto completo de parámetros del modelo se denota por:

$$\theta = (\alpha, \beta, w_1, w_2, w_3, \gamma),$$

donde α y β controlan el escalado de las componentes temporal y espacial respectivamente, w_1 , w_2 y w_3 ponderan las distintas componentes temporales, y γ regula la combinación final. La calibración de estos parámetros se realiza mediante diferentes estrategias:

1. Parámetros de escalado (α , β): Se ajustan mediante una búsqueda binaria para encontrar constantes de decaimiento óptimas. El algoritmo analiza distribuciones de penalizaciones temporales y distancias *fastDTW* geodésicas en muestras representativas, ajustando para que un 50 % de similitudes supere el umbral objetivo mediante: $\mathbb{E}[e^{-\lambda \times \text{valores}}] = 0.5$.
2. Pesos temporales (w_1 , w_2 , w_3): Se mantienen flexibles según requerimientos técnicos. Por defecto: $w_1 = w_2 = w_3 = 1$, otorgando igual importancia a retraso inicial, diferencia en duración y dilatación temporal.
3. Factor de combinación (γ): Configurable según aplicación. Por defecto: $\gamma = 0.5$, proporcionando balance equitativo entre componentes espacial y temporal.

La flexibilidad paramétrica de este método permite adaptar la métrica a diferentes casos de uso. Según el contexto específico puede ser necesario priorizar aspectos temporales (retrasos) sobre espaciales (desviaciones geográficas) o viceversa. Esta parametrización ajustable optimiza la utilidad práctica del modelo sin requerir reentrenamiento para cada aplicación específica.

La similitud resultante $\text{sim}_{\text{Total}} \in [0, 1]$ proporciona una medida interpretable donde valores cercanos a 1 indican alta similitud espacio-temporal y valores cercanos a 0 representan trayectorias significativamente diferentes.

4.2. Diseño del modelo *AdherNet*

AdherNet es un modelo de aprendizaje profundo basado en redes neuronales siamesas, específicamente diseñado para evaluar la similitud entre trayectorias de vuelo. La

arquitectura combina redes recurrentes bidireccionales con mecanismos de atención para capturar tanto las dependencias temporales locales como las relaciones globales presentes en las secuencias de trayectorias aeronáuticas. El modelo se estructura en dos módulos principales: un codificador compartido que genera representaciones densas de las trayectorias, y un comparador que evalúa la similitud entre pares de trayectorias. Esta arquitectura siamesa permite que el modelo aprenda patrones comunes en las trayectorias mientras mantiene la capacidad de identificar diferencias significativas entre ellas.

4.2.1. Codificador de trayectorias (BiGRU + Attention)

El módulo codificador de *AdherNet* está diseñado para generar representaciones densas y significativas de trayectorias de vuelo de longitud variable. La arquitectura combina una red GRU bidireccional con un mecanismo de *atención multi cabeza* para capturar tanto las dependencias temporales locales como las relaciones globales en las secuencias de trayectorias.

La clase `EnhancedTrajectoryEncoder` implementa el codificador mediante una arquitectura compuesta por tres componentes principales integrados secuencialmente.

El primer componente utiliza una *red GRU bidireccional* con L capas (por defecto $L = 2$) que procesa la secuencia de entrada en ambas direcciones temporales. Para una secuencia de entrada $X = x_1, x_2, \dots, x_T$ donde $x_t \in \mathbb{R}^d$, la *GRU bidireccional* procesa la información en ambas direcciones temporales generando estados ocultos que capturan el contexto pasado y futuro:

$$\begin{aligned}\vec{h}_t &= \text{GRU}_{\text{forward}}(x_t, \vec{h}_{t-1}), & \overleftarrow{h}_t &= \text{GRU}_{\text{backward}}(x_t, \overleftarrow{h}_{t+1}) \\ h_t &= [\vec{h}_t \oplus \overleftarrow{h}_t].\end{aligned}$$

Aquí, \vec{h}_t y \overleftarrow{h}_t son los estados ocultos generados por las GRU que procesan la secuencia hacia adelante (pasado \rightarrow futuro) y hacia atrás (futuro \rightarrow pasado), respectivamente. La operación \oplus representa la concatenación, por lo que el vector resultante h_t (de longitud $2h$) contiene la información contextual completa en el instante t , combinando ambos sentidos de lectura. Así, la salida completa se representa como una secuencia:

$$H = \{h_1, h_2, \dots, h_T\}.$$

El segundo módulo aplica un mecanismo de *atención multi-cabeza* sobre la secuencia de estados ocultos H , con el objetivo de capturar dependencias globales en la trayectoria. La atención se implementa tomando la misma secuencia H como consulta (*query*), clave (*key*) y valor (*value*):

$$A = \text{MultiHead}(Q = H, K = H, V = H).$$

Cada posición temporal h_t puede así atender a cualquier otra dentro de la secuencia, ponderando su relevancia mediante productos escalares normalizados entre consultas y claves. Este mecanismo complementa la codificación secuencial de la GRU con relaciones no locales.

Si se proporciona una máscara de atención, donde un valor de 1 indica posiciones inválidas (como tokens de padding), esta se incorpora al cálculo de las puntuaciones de atención añadiendo un valor negativo muy grande (por ejemplo, -10^9) en las posiciones enmascaradas antes de aplicar la función softmax. Esto fuerza a que las probabilidades asociadas a esas posiciones sean prácticamente cero, impidiendo que contribuyan a la salida del mecanismo de atención.

Finalmente, la salida de atención A se combina con la salida original de la GRU mediante una conexión residual, generando una secuencia enriquecida:

$$\tilde{H} = H + A,$$

donde \tilde{H} contiene, para cada paso temporal, una representación que integra tanto el contexto local (capturado por la GRU) como dependencias de largo alcance (proporcionadas por la atención).

Para poder trabajar con toda la secuencia a la vez, en la tercera y última fase necesitamos convertir esta secuencia \tilde{H} , que tiene una representación para cada instante de tiempo, en un solo vector que resuma toda la trayectoria. Esto se hace calculando el promedio de todos los vectores de la secuencia:

$$p = \frac{1}{T} \sum_{t=1}^T \tilde{h}_t,$$

donde \tilde{h}_t es el vector que representa la información en el instante t , y T es la longitud total de la secuencia. Este vector promedio $p \in \mathbb{R}^{2h}$ pasa por una serie de transformaciones para refinar su información. Primero se aplica una transformación lineal que mantiene la dimensionalidad:

$$p' = GELU(W_1 p + b_1),$$

donde $W_1 \in \mathbb{R}^{2h \times 2h}$ y $b_1 \in \mathbb{R}^{2h}$ son los pesos y sesgos entrenables de la primera capa lineal. Esta transformación actúa como un filtro que resalta los patrones más relevantes del vector p , que resume la trayectoria. La función de activación *GELU* (*Gaussian Error Linear Unit*) introduce no linealidad de forma suave, permitiendo una activación progresiva en lugar de cortes bruscos como en ReLU. Se ha demostrado que *GELU* ofrece un mejor rendimiento en tareas de regresión complejas y modelos profundos [18].

A continuación, se aplica una conexión residual sumando el vector original p a la salida transformada p' , junto con una regularización mediante dropout:

$$r = p + Dropout(p').$$

El *Dropout* es una técnica que desactiva aleatoriamente una fracción de las neuronas durante el entrenamiento, lo que obliga al modelo a no depender de activaciones concretas. Esto mejora la capacidad de generalización y reduce el sobreajuste [29].

Para estabilizar el entrenamiento y mantener los valores en una escala controlada, se aplica una normalización por capas, que ajusta la media y la varianza de las activaciones en cada capa, ayudando a una convergencia más rápida y estable, especialmente útil en arquitecturas profundas o con atención.

$$z = \text{LayerNorm}(r).$$

Finalmente, se proyecta el resultado al espacio deseado del embedding mediante una segunda transformación lineal:

$$e = W_2 z + b_2,$$

donde $W_2 \in \mathbb{R}^{d_e \times 2h}$ y $b_2 \in \mathbb{R}^{d_e}$. El vector final $e \in \mathbb{R}^{d_e}$ es una representación densa y compacta de la trayectoria completa, que será utilizada posteriormente en la comparación entre trayectorias.

4.2.2. Modelo de comparación y predicción de similitud

La clase `EnhancedSimilarityModel` implementa la arquitectura siamesa que constituye el segundo módulo principal de *AdherNet*. Esta red toma como entrada dos trayectorias codificadas T_1 y T_2 , y produce una predicción escalar de similitud $\hat{s} \in \mathbb{R}$. El diseño siamés garantiza que ambas trayectorias sean procesadas por el mismo codificador compartido ϕ (correspondiente al módulo codificador descrito en la sección anterior), generando los embeddings correspondientes:

$$e_1 = \phi(T_1, M_1), \quad e_2 = \phi(T_2, M_2).$$

A partir de estos embeddings, se construye una representación comparativa mediante la concatenación de tres componentes: los dos vectores originales y su diferencia absoluta elemento a elemento. Esto permite capturar tanto las características individuales como su discrepancia directa:

$$r = [e_1 \oplus e_2 \oplus |e_1 - e_2|].$$

La representación combinada r se introduce en una red de comparación ψ , que consiste en una red con tres capas lineales, activaciones GELU y regularización mediante *dropout*. La salida final es un escalar de similitud predicha:

$$\hat{s} = \psi(r),$$

donde ψ representa la función compuesta por las capas lineales y las activaciones:

$$\psi(r) = \text{Linear}_3 \circ \text{GELU} \circ \text{Linear}_2 \circ \text{GELU} \circ \text{Linear}_1(r).$$

Aquí, cada `Linear` incluye sus respectivos pesos W_i y sesgos b_i . Esta arquitectura permite modelar relaciones no lineales entre las trayectorias de entrada y su similitud.

4.2.3. Entrenamiento y evaluación del modelo

Como función de pérdida principal, el modelo *AdherNet* utiliza el *Error Cuadrático Medio* (MSE), cuya fórmula es:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N (\hat{s}_i - s_i)^2,$$

donde θ representa todos los parámetros del modelo, \hat{s}_i es la predicción del modelo y s_i es el valor objetivo de similitud para el par de trayectorias i . La elección del MSE se justifica por su capacidad para penalizar con mayor severidad los errores grandes,

lo que resulta especialmente útil en tareas de regresión donde se requiere una aproximación precisa del valor escalar de similitud. Además, al tratarse de un objetivo continuo y diferenciable, su optimización es eficiente y estable mediante gradiente descendente.

El entrenamiento del modelo se basa en varias estrategias diseñadas para optimizar el rendimiento y garantizar una generalización robusta. En primer lugar, se utiliza el optimizador *AdamW* [24], una variante del algoritmo *Adam* que incluye decaimiento de pesos para una *regularización L2* más explícita y estable. Esta elección se justifica por su capacidad para adaptarse a diferentes escalas de parámetros, acelerando la convergencia en problemas de regresión, especialmente en arquitecturas profundas. La función de pérdida total que se minimiza durante el entrenamiento es:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}(\theta) + \lambda \sum_i \|\theta_i\|_2^2,$$

donde $\mathcal{L}(\theta)$ es el *error cuadrático medio* (MSE), λ es el *coeficiente de regularización* y θ_i denota los parámetros del modelo.

Para ajustar dinámicamente la tasa de aprendizaje durante el entrenamiento, se utiliza una *planificación adaptativa* con un *scheduler* que reduce la tasa η si la pérdida de validación \mathcal{L}_{val} no mejora después de un número determinado de épocas (*patience*). Esto permite refinar el aprendizaje en etapas tardías del entrenamiento y evitar estancamientos:

$$\eta_{t+1} = \begin{cases} \eta_t \times \gamma & \text{si } \mathcal{L}_{\text{val}}^{(t)} \geq \mathcal{L}_{\text{val}}^{(t-p)} \\ \eta_t & \text{en caso contrario} \end{cases},$$

donde $\gamma = 0.5$ es el factor de reducción y $p = 5$ es la paciencia.

Además, para evitar el sobreajuste, se emplea *early stopping*: el entrenamiento se detiene automáticamente si la pérdida de validación no mejora durante $P = 10$ épocas consecutivas. Esto evita entrenar de más una red que ya ha alcanzado su capacidad de generalización.

Por último, se implementa *gradient clipping* para estabilizar la retropropagación en redes profundas. Esta técnica mitiga la explosión de gradientes que puede hacer divergir la optimización. El recorte se realiza según:

$$\tilde{g} = \begin{cases} g & \text{si } \|g\|_2 \leq \tau \\ \frac{\tau \times g}{\|g\|_2} & \text{si } \|g\|_2 > \tau \end{cases},$$

donde g es el gradiente original, \tilde{g} es el gradiente recortado y $\tau = 1.0$ es el umbral establecido.

Para ajustar los hiperparámetros del modelo *AdherNet*, se emplea una estrategia de búsqueda enfocada, que prioriza los parámetros con mayor impacto en el rendimiento. Esta técnica es más rápida y práctica que una búsqueda exhaustiva, ya que reduce el espacio de búsqueda al evitar combinaciones irrelevantes.

El proceso evalúa combinaciones de parámetros clave como tamaño de las capas ocultas, tasa de aprendizaje, tamaño de lote y dropout, mientras que otros parámetros se mantienen con valores predeterminados o calculados automáticamente.

mediante reglas empíricas, facilitando así la configuración. Esta metodología permite obtener configuraciones competitivas en tiempos mucho menores, ideal para prototipado rápido o cuando los recursos computacionales son limitados. Aunque no explora exhaustivamente todo el espacio, ofrece un equilibrio efectivo entre eficiencia y calidad de ajuste.

En conjunto, estas estrategias garantizan un entrenamiento eficiente, estable y robusto, mitigando problemas habituales como sobreajuste, estancamiento en la optimización o inestabilidad numérica.

Finalmente, para evaluar el rendimiento del modelo, se emplean varias métricas complementarias que permiten obtener una visión más completa del comportamiento del sistema:

- *Coeficiente de determinación (R^2)*: mide la proporción de la varianza total de los valores verdaderos que es explicada por el modelo. Valores cercanos a 1 indican una buena capacidad predictiva.

$$R^2 = 1 - \frac{\sum_{i=1}^N (s_i - \hat{s}_i)^2}{\sum_{i=1}^N (s_i - \bar{s})^2}.$$

- *Error absoluto medio (MAE)*: proporciona una medida directa de la magnitud media del error, menos sensible a valores atípicos que el MSE.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |s_i - \hat{s}_i|.$$

Capítulo 5

Resultados

En este capítulo se presentan los resultados obtenidos tras la aplicación de los métodos desarrollados para el análisis y comparación de trayectorias de vuelo. En primer lugar, se detalla el proceso de tratamiento y caracterización de los datos utilizados, así como los desafíos inherentes a su naturaleza multidimensional y con ruido. Posteriormente, se describen los experimentos realizados para evaluar la precisión, generalización y eficiencia de los modelos propuestos, incluyendo comparativas entre diferentes arquitecturas y estrategias de etiquetado. Finalmente, se analizan los resultados en términos de rendimiento y aplicabilidad en escenarios reales de gestión del tráfico aéreo.

5.1. Datos

En esta sección se explica el proceso seguido para trabajar con los datos de trayectorias de vuelo. Se describe qué datos se han usado, cómo se han preparado y qué dificultades presentan. Además, se detallan los métodos aplicados para obtener trayectorias útiles a partir de los datos originales.

Los datos utilizados provienen de fuentes internas de CRIDA y se dividen en dos conjuntos principales: planes de vuelo (CFMU - Central Flow Management Unit) extraídos de EUROCONTROL, la organización europea que gestiona el flujo de tráfico aéreo centralizadamente, y trazas de vuelo (FLOWS) obtenidos de PALESTRA, sistema de ENAIRE (proveedor de servicios de navegación aérea de España). Ambos conjuntos contienen registros de diferentes fechas que se especifican según cada experimento.

5.1.1. Estructura de los datos crudos

A continuación se describen las principales variables contenidas en cada conjunto de datos, con el objetivo de comprender su significado y utilidad para la construcción de trayectorias.

Los planes de vuelo (o FPL por sus siglas en inglés), a partir de ahora, FPL1 para planes iniciales (tipo 1) y FPL3 para planes modificados (tipo 3), incluyen registros con múltiples campos que describen tanto características del vuelo como puntos de navegación. Las variables más relevantes son:

- `flightkey`: Identificador único del vuelo.

- `processDateReference`: Fecha en la que se procesaron los datos del vuelo.
- `FlightModelType`: Tipo de plan de vuelo, (1: plan inicial, 3: plan modificado).
- `FL`: Nivel de vuelo (Flight Level), en cientos de pies (por ejemplo, FL350 = 35,000 ft).
- `eto`: Estimated Time Over, tiempo estimado de paso sobre el punto dado.
- `latitude`: Latitud geográfica del punto, expresada en grados decimales.
- `longitude`: Longitud geográfica del punto, también en grados decimales.

Las trazas (o TRK, del inglés *track*), reflejan los datos reales de posición y altitud del avión durante el vuelo, registrados en distintos instantes. Los campos principales que componen este conjunto son:

- `flightKey`: Identificador único del vuelo.
- `dateReference`: Fecha de vuelo.
- `time`: Hora exacta del registro, en formato 9HHMMSS.
- `lat`: Latitud geográfica del punto, expresada en grados decimales.
- `lng`: Longitud geográfica del punto, también en grados decimales.
- `modo_c`: Modo C del transpondedor, que indica la altitud reportada del avión (en cientos de pies).

5.1.2. Obtención de trayectorias a partir de los datos crudos

El proceso de construcción de trayectorias se realiza a través de los siguientes pasos:

1. Carga y estandarización de datos (`load_and_standardize_csv`): Se carga el archivo CSV, se estandarizan los nombres de cada columna y se asegura la presencia de las columnas necesarias.
2. Limpieza y conversión (`clean_and_prepare_dataframe`): Se ordenan los datos por `flightKey` y tiempo, se convierten las fechas estimadas (`eto`) en marcas temporales absolutas (en segundos), y se eliminan registros inválidos o duplicados.
3. Preprocesamiento y construcción (`preprocess_csv`): Este método principal combina los pasos anteriores y realiza:
 - La detección de tipos de modelo de vuelo (`FlightModelType`).
 - La agrupación de los datos por `flightkey`.
 - El filtrado de vuelos con menos de un número mínimo de puntos válidos. 20 puntos es una práctica habitual en estudios de trayectoria [7] y proporciona una base firme para análisis comparativos. En este caso se usan mínimo 5 puntos, lo cual es válido para trayectorias muy cortas o con muestreo pobre, como son los planes de vuelo (que en nuestro caso tienen entre 2 y 30 puntos no duplicados).
 - La construcción de trayectorias como listas ordenadas de puntos, cada uno compuesto por: `timestamp_sec`, `latitude`, `longitude` y `fl`.

4. Resultado final: Un conjunto de trayectorias organizadas en diccionarios, agrupados por tipo de trayectoria (por ejemplo, `FPL1` para planes iniciales, `FPL3` para modificados y `TRK` para trazas). Si el tipo no está disponible o no es relevante para el experimento, las trayectorias se agrupan bajo una única categoría general.

5.1.3. Características de los datos

El conjunto de datos presenta diversas características que representan desafíos importantes en el análisis de trayectorias. Estas características se dividen en tres grupos principales:

1. *Muestreo con ruido, no uniforme, y baja densidad*

Este es uno de los principales desafíos mencionados en la literatura, compuesto por los siguientes aspectos:

- a) Muestreo no uniforme y baja densidad de puntos: Las trayectorias no siempre están registradas con la misma frecuencia ni tienen la misma cantidad de puntos, debido a problemas técnicos como fallos del GPS, diferencias en los dispositivos de captura o interrupciones en la transmisión de datos. Esto afecta a la representatividad del recorrido y puede hacer parecer diferentes trayectorias que son, en esencia, similares.
- b) Datos incompletos (segmentos faltantes): Algunos tramos de las trayectorias pueden estar ausentes por causas como pérdida de señal, fallos de sensores, errores de almacenamiento o problemas en la carga de datos al Data Warehouse (DWH). Estos huecos dificultan el uso de técnicas que requieren correspondencia directa punto a punto.
- c) Ruido en el muestreo (errores GPS): Especialmente en zonas urbanas, la señal GPS puede estar distorsionada, lo que genera errores en la localización. Esto puede hacer que trayectorias que en realidad son cercanas parezcan estar separadas espacialmente.

2. *Duración y velocidad variables*

Este grupo de problemas se refiere a las diferencias dinámicas y temporales entre trayectorias:

- a) Desfase temporal entre trayectorias: Dos trayectorias pueden seguir exactamente la misma ruta, pero en diferentes momentos del día (por ejemplo, 8:00 AM vs 6:00 PM). Si solo se considera la dimensión espacial, parecen similares, pero el tiempo puede ser un factor clave en análisis como el del tráfico.
- b) Duraciones temporales diferentes: Algunas trayectorias tienen una duración significativamente distinta a otras, lo que puede deberse a cambios en la ruta, esperas en el aire o diferencias operacionales. Estas variaciones afectan su alineación temporal y dificultan su comparación directa.
- c) Velocidad variable: La velocidad puede variar dentro de una misma trayectoria, lo que influye en la frecuencia de muestreo y en la forma percibida del recorrido. Los métodos que comparan punto por punto tienden a fallar cuando las velocidades no son consistentes.

3. Datos 4D de vuelo

El análisis de trayectorias de vuelo presenta una complejidad adicional debido a su naturaleza 4D: incorporan no solo latitud y longitud (en coordenadas geodésicas), sino también altitud (expresada en niveles de vuelo) y tiempo.

Esta estructura multidimensional introduce desafíos significativos, ya que cualquier técnica de comparación o análisis debe considerar simultáneamente variaciones espaciales, verticales y temporales.

5.2. Experimentación

La experimentación se estructura en múltiples dimensiones para evaluar exhaustivamente el rendimiento de los métodos de adherencia entre trayectorias. Se analizan dos comparaciones clave:

- FPL1-FPL3: Comparación entre el plan de vuelo inicial (FPL1) y su correspondiente plan modificado (FPL3). Se han recogido 50,000 pares de trayectorias en un periodo de dos meses (del 1 de febrero al 31 de marzo de 2024).
- FPL3-TRK: Comparación entre el plan modificado (FPL3) y la trayectoria radar (TRK) correspondiente. Se emplean 10,000 pares registrados durante los días 1 y 2 de agosto de 2022.

5.2.1. Evaluación de resultados y estrategias de etiquetado

Para analizar el impacto de la estrategia de etiquetado en el rendimiento de *AdherNet*, se compararon dos enfoques: el etiquetado automático basado en la distancia DTW y la métrica combinada propuesta, que pondera de forma equitativa la similitud espacial y temporal. La distancia DTW proporciona una medida sencilla y directa, ideal como aproximación inicial o cuando se busca simplicidad en el proceso de entrenamiento. Por su parte, la métrica combinada introduce mayor flexibilidad, permitiendo ajustar la sensibilidad del modelo hacia diferentes aspectos del problema según las necesidades de la aplicación.

En primer lugar, se estudió el comportamiento del modelo cuando se entrenó utilizando las etiquetas basadas en DTW. La Figura 5.1 muestra la dispersión de las predicciones frente a los valores reales para el conjunto de datos FPL1-FPL3. Los puntos azules forman una línea diagonal bastante recta desde el origen (0,0) hasta aproximadamente (2500, 2500), con una línea de referencia roja que representa la predicción perfecta (donde predicción = valor real). La mayoría de los puntos se encuentran muy próximos a esta línea diagonal, lo que sugiere que el modelo tiene un buen rendimiento predictivo. Este comportamiento se mantiene constante a lo largo de todo el rango de valores analizado, mostrando una dispersión baja y sin patrones evidentes de sesgo, lo que refuerza la impresión de estabilidad y precisión del modelo. En este escenario, el modelo alcanzó un coeficiente de determinación $R^2 = 0.9602$ y una pérdida de entrenamiento de 6376.67, lo que confirma la precisión y la consistencia del ajuste.

Resultados

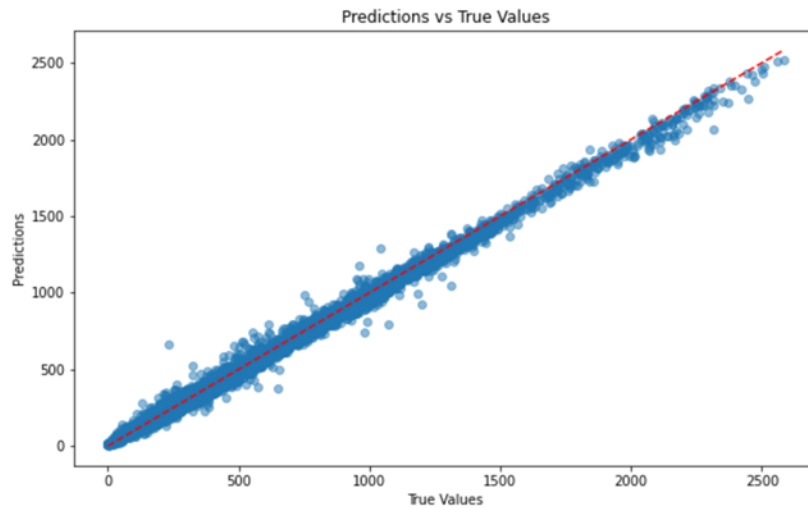


Figura 5.1: Dispersión entre las predicciones de AdherNet y las etiquetas DTW, en 50,000 pares $FPL1-FPL3$ recogidos del 1 de febrero al 31 de marzo de 2024

Cuando se emplea la métrica combinada (Figura 5.2), el comportamiento global sigue siendo bueno ($R^2 = 0.9602$), pero se aprecia una mayor dispersión en las zonas de baja similitud. Esto sugiere que la mayor complejidad de la métrica puede introducir cierta incertidumbre en los casos extremos, donde el modelo parece tener más dificultades para ajustar su predicción. En esta configuración, el modelo alcanzó un *error cuadrático medio* (MSE) en test de 11.44 y un R^2 de 0.9796, lo que evidencia un buen ajuste incluso frente a la complejidad añadida por la métrica.

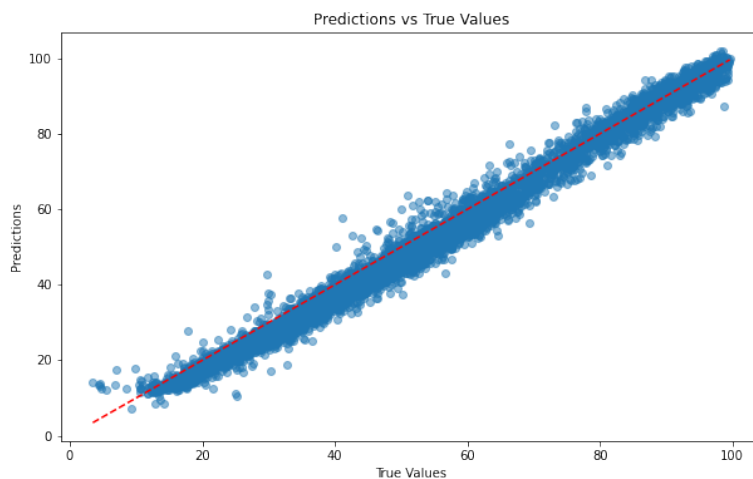


Figura 5.2: Dispersión entre las predicciones de AdherNet y las etiquetas de la métrica combinada, en 50,000 pares $FPL1-FPL3$ recogidos del 1 de febrero al 31 de marzo de 2024

Aunque ambos etiquetados permiten al modelo alcanzar un ajuste notable, el uso de la métrica combinada introduce mayor dispersión en pares de trayectorias con

similitudes bajas (o distancias altas). Esto podría deberse a la naturaleza más rica y compleja de la métrica, que pondera aspectos temporales y espaciales, dificultando la tarea del modelo en rangos extremos. En resumen, los resultados muestran que AdherNet puede ajustarse a diferentes métricas de similitud, capturando patrones tanto en distancias simples como en medidas combinadas más complejas.

El modelo también fue evaluado en datos de otro año, en concreto 10,000 pares FPL1-FPL3 recogidos el 1 de agosto de 2022. Como se observa en la Figura 5.3, existe una fuerte correlación lineal entre las predicciones del modelo y los valores reales, reespaldado con un $R^2 = 0.8987$. La concentración de puntos a lo largo de la diagonal indica que el modelo mantiene su capacidad predictiva incluso cuando se aplica a datos de un período temporal diferente.

Este experimento actúa como validación del modelo, ya que pretende simular las condiciones reales de operación, donde los patrones de vuelo pueden variar significativamente según factores como las condiciones meteorológicas, el tráfico aéreo o cambios en los procedimientos operacionales. Un modelo que funciona bien con datos de un día específico podría fallar completamente cuando se aplica a datos de una semana diferente.

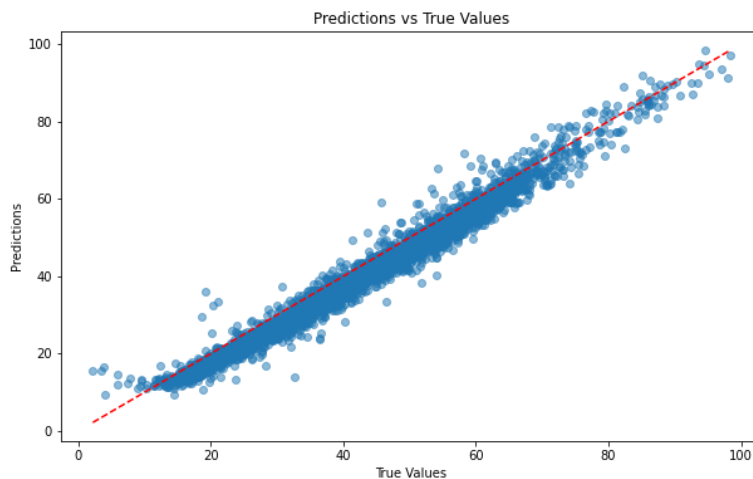


Figura 5.3: Dispersión entre las predicciones del AdherNet y etiquetas de la métrica combinada en 10,000 pares FPL1-FPL3 del 1 de agosto de 2022.

Se ha seguido un procedimiento análogo con el conjunto de datos FLP3-TRK, aplicando las mismas estrategias de etiquetado y entrenamiento del modelo. Los resultados obtenidos son comparables a los presentados para FPL1-FPL3, tanto en las gráficas de dispersión como en los valores del coeficiente de determinación. Concretamente, el modelo alcanzó un error cuadrático medio en test de 62.53 y un R^2 de 0.9315, mostrando un buen ajuste que confirma la capacidad de generalización de *AdherNet* a distintos tipos de datos y relaciones espaciales y temporales. Esta consistencia refuerza la idea de que las estrategias de etiquetado propuestas, ya sea con la métrica combinada o con DTW, funcionan adecuadamente para diferentes conjuntos y contextos operacionales, manteniendo la robustez y precisión del modelo.

5.2.2. Comparación entre arquitecturas simples y complejas

El diseño de modelos para comparar trayectorias aéreas debe equilibrar precisión y eficiencia. Las redes feed-forward (ffNN) son simples, rápidas de entrenar y fáciles de desplegar. Sin embargo, al ignorar la estructura secuencial de los datos, sacrifican capacidad de representación. En contraste, AdherNet es una arquitectura compleja que combina redes GRU bidireccionales, mecanismos de atención multi-cabeza y capas de comparación no lineales. Este diseño no es gratuito: está pensado para capturar dependencias temporales y patrones contextuales a lo largo de las trayectorias.

La comparación empírica entre ambas arquitecturas pone en evidencia esta diferencia. Como se mostró en la Figura 5.2, AdherNet alcanza un coeficiente de determinación de $R^2 = 0.9796$ en test. En cambio, una ffNN entrenada con el mismo dataset apenas alcanza $R^2 = 0.7168$ (Figura 5.4).

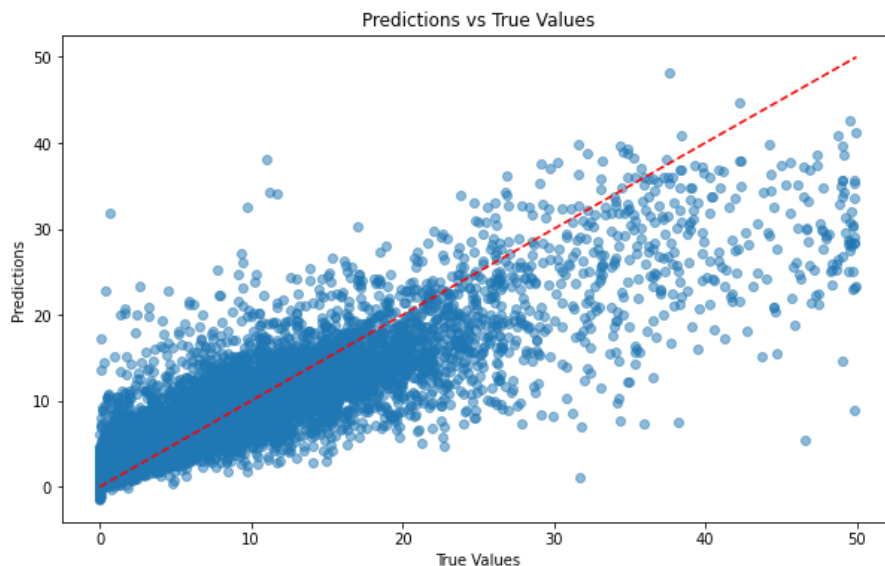


Figura 5.4: Resultados de test en el entrenamiento de una ffNN usando etiquetas DTW.

Estos resultados reflejan una diferencia sustancial en rendimiento, especialmente en los valores altos de similitud, donde la ffNN muestra errores sistemáticos. En cambio, AdherNet mantiene una alta precisión en todo el rango. Esta ganancia no solo justifica la mayor complejidad de la arquitectura, sino que la hace necesaria en contextos donde la calidad de predicción es crítica, como en aplicaciones operativas de gestión del tráfico aéreo.

La superioridad de AdherNet no es accidental, sino consecuencia directa de su capacidad para modelar la estructura secuencial y contextual de las trayectorias. La complejidad arquitectónica y el mayor uso de recursos están plenamente justificados por el rendimiento obtenido.

5.2.3. Funcionalidades de búsqueda y ordenación

Se evaluará la capacidad del sistema para identificar trayectorias similares y ordenar listas por similitud, más allá de la simple clasificación binaria de adherencia. Estas

funcionalidades son esenciales para la aplicabilidad práctica del sistema. La búsqueda eficiente es crítica cuando se debe encontrar trayectorias similares en bases de datos masivas. Se utilizaron 1000 parejas de trayectorias de planes tipo 1 y tipo 3. Para simular un escenario realista y exigente, se eliminaron los identificadores de las trayectorias y se pidió a cada modelo que identificara la pareja correcta entre 1000 opciones. Esto implica que cada trayectoria se compara con todas las del conjunto de referencia, dando lugar a 1 millón de comparaciones. Esta búsqueda por fuerza bruta representa el peor caso computacional.

Resultados de ejecución

Los resultados que se presentan a continuación corresponden a la evaluación realizada sobre 1000 trayectorias emparejadas, salvo en el caso del método DTW clásico, que se limitó a 100 trayectorias debido a su alto coste computacional. El método AdherNet combinado con búsqueda por similitud coseno obtuvo un tiempo total de búsqueda de 4.36 segundos, aunque el proceso completo, incluyendo el encoding, alcanzó aproximadamente los 4.31 segundos (siendo el tiempo de búsqueda en sí prácticamente despreciable, menor de 0.05 s).

En contraste, el enfoque basado en una red neuronal feedforward (ffNN) requirió 1 hora y 11 minutos, mientras que *FastDTW*, aplicado directamente sobre las trayectorias sin interpolación, necesitó 7 horas y 1 minuto. Por su parte, el DTW clásico resultó ser el método más costoso: su ejecución sobre solo 100 trayectorias tomó un promedio de 15.77 segundos por comparación, lo que extrapolado a 1000 trayectorias daría un tiempo estimado superior a las 43 horas (a 154.8 segundos por iteración), haciéndolo inviable para su uso práctico. La siguiente tabla resume el rendimiento de cada uno de estos métodos.

Método	Precisión@1	Precisión@5	Ranking promedio	Similitud media	Tiempo medio (s)
FFNN	0.162	0.346	7.615	67.0426	4.3454
AdherNet	0.957	1.000	1.055	100.00	0.0044
<i>FastDTW</i>	0.984	0.996	1.043	86.6638	25.1431
DTW clásico	1.000	1.000	1.000	87.6539	15.7650

Tabla 5.1: Comparación de métodos al emparejar 1000 trayectorias

Análisis de resultados

En el análisis de los resultados se han empleado varias métricas que permiten evaluar no solo la precisión de los métodos de emparejamiento de trayectorias, sino también su eficiencia computacional. Cada una de estas métricas aporta información complementaria sobre el rendimiento de los algoritmos, tanto en términos de calidad como de viabilidad práctica.

La métrica Precisión@1 indica la proporción de veces en que el método evaluado situó la trayectoria correcta en la primera posición del ranking de resultados. Es un indicador de exactitud directa y estricta. En este sentido, el DTW clásico logró un valor perfecto de 1.000, seguido de cerca por *FastDTW* (0.984) y AdherNet (0.957), mientras que la red neuronal feedforward (FFNN) obtuvo un resultado significativamente inferior, con solo 0.162. Por otro lado, la Precisión@5 mide la frecuencia con la que la trayectoria correcta se encuentra entre las cinco primeras posiciones del ranking.

Resultados

Esta métrica es más permisiva que la anterior, pero sigue siendo útil para tareas en las que se permite cierta tolerancia en la respuesta. Aquí, tanto AdherNet como el DTW clásico alcanzan la máxima puntuación de 1.000, *FastDTW* se aproxima mucho con 0.996, y FFNN mejora respecto a Precisión1 pero aún queda por detrás con un 0.346.

El ranking promedio representa la posición media en la que se encuentra la trayectoria emparejada correctamente. Cuanto más cercano a 1, mejor. Una vez más, el DTW clásico lidera con un valor exacto de 1.000, lo que significa que siempre acierta en la primera posición. *FastDTW* y AdherNet también presentan un rendimiento excelente, con valores apenas por encima de 1, mientras que FFNN muestra un desempeño considerablemente peor con un ranking promedio de 7.615.

La similitud media refleja el valor promedio de la métrica de similitud utilizada para comparar trayectorias. En este caso, AdherNet presenta una similitud media de 100.00, lo que sugiere una normalización o escalado del valor. *FastDTW* y el DTW clásico obtienen también valores elevados (86.66 y 87.65, respectivamente), mientras que FFNN queda muy por debajo, con una similitud media de 67.04. Esta métrica puede ayudar a entender cómo de cercanas se perciben las trayectorias emparejadas, aunque no siempre se correlaciona directamente con la precisión si no se acompaña de un buen ranking.

Finalmente, el tiempo medio por comparación refleja la eficiencia computacional de cada método. AdherNet destaca con una velocidad sobresaliente (0.0044 s), lo que lo convierte en el método más rápido con diferencia. FFNN presenta un tiempo medio bajo también (4.35 s), aunque no compite en precisión. *FastDTW* y DTW clásico, por el contrario, son considerablemente más lentos, con tiempos medios de 25.14 s y 15.77 s respectivamente, lo que limita su escalabilidad.

En conjunto, el análisis revela que AdherNet ofrece una combinación óptima de alta precisión y bajo coste computacional, lo que lo convierte en una alternativa extremadamente atractiva para tareas de emparejamiento a gran escala. *FastDTW* y DTW clásico siguen destacando en precisión absoluta, pero su coste temporal los hace menos prácticos, especialmente en contextos donde la eficiencia es crítica. Por su parte, el enfoque con FFNN queda rezagado tanto en precisión como en similitud, a pesar de su relativa eficiencia.

Capítulo 6

Conclusión

El presente Trabajo Fin de Máster ha abordado el problema de cuantificar la adherencia entre trayectorias de vuelo planificadas y ejecutadas mediante técnicas de Inteligencia Artificial (IA), en el contexto del control de tráfico aéreo. En particular, se ha desarrollado *AdherNet*, un modelo basado en redes neuronales recurrentes con mecanismos de atención, capaz de estimar automáticamente la similitud espaciotemporal entre pares de trayectorias sin necesidad de realizar alineamientos explícitos ni transformaciones complejas.

A lo largo del estudio, se ha demostrado que este enfoque es viable, eficiente y escalable, permitiendo procesar trayectorias crudas (latitud, longitud, altitud y tiempo) de forma directa. Se han desarrollado y evaluado dos variantes del modelo, orientadas respectivamente a medir la coherencia entre plan de vuelo inicial y final, y entre plan final y trayectoria radar. Los resultados obtenidos muestran que *AdherNet* ofrece un rendimiento competitivo respecto a métricas clásicas como *FastDTW*.

La principal contribución es el desarrollo de *AdherNet*, un modelo supervisado basado en redes neuronales recurrentes con atención, diseñado específicamente para estimar la similitud espaciotemporal entre trayectorias sin necesidad de alineamientos explícitos ni preprocesamientos complejos en la fase de inferencia. Además del modelo, se ha propuesto una métrica de adherencia interpretable y calibrable, que combina distancia geodésica y dilatación temporal con parámetros ajustables, permitiendo adaptar la sensibilidad del sistema sin reentrenar. Esta métrica no solo ha servido como base supervisada para el entrenamiento, sino que resulta útil también como referencia operativa para validar comportamientos en vuelo.

Estas contribuciones se han desarrollado de forma alineada con los objetivos definidos al inicio del proyecto, en la Sección 1.3.2. Se cumplió con la definición de una métrica de referencia robusta (O1) y la construcción de un conjunto de datos representativo a partir de datos reales proporcionados por CRIDA (O2), el cual puede ser aprovechado para investigaciones futuras tanto en entornos supervisados como auto-supervisados o contrastivos. Se implementaron dos variantes del modelo (O3), centradas en dos escenarios distintos pero complementarios: la comparación entre planes de vuelo inicial y final, y la comparación entre el plan final y la trayectoria radar.

El diseño de la arquitectura (O4) ha demostrado ser adecuado para procesar trayectorias crudas, capturando relaciones temporales y espaciales sin necesidad de estruc-

turas auxiliares complejas. El entrenamiento (O5) se realizó con técnicas modernas de regularización y validación, obteniendo un modelo estable, preciso y generalizable. Finalmente, el modelo se ha validado en tareas relevantes como búsqueda de trayectorias similares y detección automatizada de desviaciones (O7), demostrando su aplicabilidad práctica en entornos de análisis post-operacional o como componente de sistemas de alerta en tiempo casi real (O6).

Este proyecto ha sido un ejercicio práctico de cómo trasladar técnicas avanzadas de IA a un dominio crítico como el tráfico aéreo, respetando sus restricciones y particularidades. El desarrollo de herramientas que procesen datos imperfectos, heterogéneos y en gran volumen requiere un equilibrio constante entre complejidad algorítmica, eficiencia computacional y utilidad operativa. La experiencia también ha evidenciado que trabajar en la intersección entre investigación y aplicación concreta es clave para que la IA tenga un impacto real y sostenido en sectores operativos. La metodología desarrollada no es un fin en sí mismo, sino una base sobre la que construir sistemas más robustos, flexibles y útiles para quienes gestionan la complejidad del espacio aéreo cada día.

En conjunto, se ofrece una base sólida y extensible para el desarrollo de herramientas inteligentes de monitorización en la gestión del tráfico aéreo, alineadas con los objetivos del proyecto DELFOS.

6.1. Casos de uso

El sistema desarrollado tiene aplicación directa en diversos escenarios operativos dentro del entorno ATM. Uno de los más evidentes es la comparación entre dos trayectorias concretas, ya sean planificadas o ejecutadas. Esto resulta útil tanto para validar modificaciones introducidas por el ATC como para evaluar propuestas de mejora en rutas aéreas. Disponer de una métrica cuantitativa de adherencia permite fundamentar decisiones sobre eficiencia, seguridad o cumplimiento normativo con datos objetivos y comparables.

Otro caso de uso clave es la evaluación de la adherencia de una trayectoria radar a su plan de vuelo correspondiente. Dado que las trayectorias reales suelen presentar ruido, pérdidas de señal o muestreos irregulares, la capacidad de comparar estos datos con los planes previstos de forma robusta y automática tiene un enorme valor práctico. Esta funcionalidad permite monitorizar el grado de cumplimiento operativo, detectar desviaciones significativas en tiempo real y facilitar auditorías posteriores. También sirve como punto de partida para mejorar la predicción de trayectorias o validar el impacto de nuevas regulaciones.

Un tercer caso de uso, más ambicioso pero igualmente viable, es la identificación de trayectorias radar anónimas o incompletas. En contextos donde los datos no están totalmente etiquetados o identificados (por errores de etiquetado, pérdida de datos o conflictos en sistemas de fusión), poder buscar en una base de datos de planes conocidos aquella trayectoria más similar puede resolver ambigüedades y mejorar la calidad del dato. Gracias a la eficiencia del espacio de embeddings generado por *AdherNet*, estas búsquedas pueden realizarse de forma rápida incluso en bases de datos de gran tamaño, lo que permite construir sistemas operativos capaces de corregir o enriquecer la información en tiempo real.

En conjunto, estos casos de uso evidencian que el sistema no solo tiene una base teórica sólida, sino que responde a necesidades concretas y recurrentes en el dominio del control del tráfico aéreo.

6.2. Trabajos futuros

La metodología presentada en este Trabajo Fin de Máster ofrece múltiples aplicaciones prácticas dentro del ámbito del control del tráfico aéreo. Su principal valor radica en proporcionar un marco sistemático y eficiente para comparar trayectorias planificadas y ejecutadas a partir de datos crudos, sin necesidad de alineamientos explícitos ni preprocesamientos intensivos. Además, su flexibilidad permite ajustar la sensibilidad de la comparación según las necesidades del entorno operativo, priorizando el componente temporal, el espacial o una combinación de ambos.

Como evolución natural de esta metodología, el modelo *AdherNet* puede integrarse como un embudo de procesamiento que, ante incertidumbres o dudas en el emparejamiento entre dos trayectorias, recurra a la métrica tradicional definida para asegurar una evaluación robusta y coherente. Esta aproximación híbrida potencia el enfoque propuesto, combinando la eficiencia del aprendizaje automático con la solidez de las métricas clásicas, especialmente en casos de trayectorias complejas o atípicas. Asimismo, la aplicación de *AdherNet* al análisis de grandes volúmenes de datos permitiría identificar operadores o tipos de vuelo con mayor propensión a modificar sus planes, lo cual resulta valioso para tareas de auditoría, regulación o benchmarking operativo.

En el ámbito de la predicción de trayectorias, la metodología aquí expuesta puede utilizarse como métrica de validación entre trayectorias predichas y observadas, ofreciendo un criterio cuantitativo para evaluar el rendimiento de algoritmos predictivos. Esta aplicación resulta especialmente relevante para el desarrollo de sistemas avanzados de gestión del tráfico aéreo.

Finalmente, el uso de *AdherNet* como medida de similitud para tareas de clustering abre la puerta a la estimación de flujos de vuelo entre distintas regiones o aeropuertos. Además, constituye una herramienta útil para mejorar la calidad de conjuntos de datos incompletos mediante la inferencia automática de datos faltantes, apoyada en la similitud entre trayectorias parciales y trayectorias de referencia.

Bibliografía

- [1] R. Agrawal, C. Faloutsos y A. Swami. “Efficient similarity search in sequence databases”. En: *Foundations of Data Organization and Algorithms*. Ed. por D. B. Lomet. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, págs. 69-84.
- [2] P. Bakalov, M. Hadjieleftheriou, E. Keogh y V. J. Tsotras. “Efficient trajectory joins using symbolic representations”. En: *Proceedings of the 6th International Conference on Mobile Data Management*. Ayia Napa, Cyprus: Association for Computing Machinery, 2005, págs. 86-93.
- [3] H. Cao, H. Tang, Y. Wu, F. Wang e Y. Xu. “On Accurate Computation of Trajectory Similarity via Single Image Super-resolution”. En: *Proceedings of the International Joint Conference on Neural Networks*. 2021, págs. 1-9.
- [4] Y. Chang, J. Qi, Y. Liang y E. Tanin. “Contrastive Trajectory Similarity Learning with Dual-Feature Attention”. En: *Proceedings of the 39th IEEE International Conference on Data Engineering*. 2023, págs. 2933-2945.
- [5] Y. Chang, J. Qi, E. Tanin, X. Ma y H. Samet. “Sub-trajectory Similarity Join with Obfuscation”. En: *Proceedings of the 33rd International Conference on Scientific and Statistical Database Management*. Tampa, FL, USA: Association for Computing Machinery, 2021, págs. 181-192.
- [6] Y. Chang, E. Tanin, X. Cao y J. Qi. “Spatial Structure-Aware Road Network Embedding via Graph Contrastive Learning”. En: *Proceedings of the 26th International Conference on Extending Database Technology*. 2023, págs. 144-156.
- [7] Y. Chang, E. Tanin, G. Cong, C. S. Jensen y J. Qi. “Trajectory Similarity Measurement: An Efficiency Perspective”. En: *Proceedings of the VLDB Endowment* 17.9 (mayo de 2024), págs. 2293-2306.
- [8] Z. Chang, L. Yu, H. Li, S. Wu, G. Chen y D. Zhang. “Revisiting CNNs for Trajectory Similarity Learning”. En: *Proceedings of the VLDB Endowment* 18.4 (mayo de 2025), págs. 1013-1021.
- [9] L. Chen y R. Ng. “On the Marriage of LP-norms and Edit Distance”. En: *Proceedings of the Thirtieth International Conference on Very Large Data Bases*. 2004, págs. 792-803.
- [10] L. Chen, M. T. Özsu y V. Oria. “Robust and fast similarity search for moving object trajectories”. En: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. Baltimore, Maryland: Association for Computing Machinery, 2005, págs. 491-502.

- [11] W. Chen, Y. Liang, Y. Zhu, Y. Chang, K. Luo, H. Wen, L. Li, Y. Yu, Q. Wen, C. Chen, K. Zheng, Y. Gao, X. Zhou e Y. Zheng. “Deep Learning for Trajectory Data Management and Mining: A Survey and Beyond”. En: *CoRR abs/2403.14151* (2024). arXiv: 2403.14151.
- [12] J. Chung, C. Gulcehre, K. Cho e Y. Bengio. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. En: *NIPS 2014 Workshop on Deep Learning*. 2014.
- [13] L. Deng, Y. Zhao, Z. Fu, H. Sun, S. Liu y K. Zheng. “Efficient Trajectory Similarity Computation with Contrastive Learning”. En: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. Atlanta, GA, USA: Association for Computing Machinery, 2022, págs. 365-374.
- [14] J. Devlin, M. Chang, K. Lee y K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. En: *CoRR abs/1810.04805* (2018). arXiv: 1810.04805.
- [15] T. Eiter y H. Mannila. *Computing Discrete Fréchet Distance*. Inf. téc. Technical University of Vienna, 1994.
- [16] K. Fang, X. Ma, Y. Hu, Y. Liu, L. Zheng y J. Shang. “TSAN: Temporal Siamese Attention Network for Flight Trajectory Similarity Learning”. En: *IEEE Sensors Journal* 25.2 (2025), págs. 3848-3858.
- [17] T.-Y. Fu y W.-C. Lee. “Trembr: Exploring Road Networks for Trajectory Representation Learning”. En: *ACM Transactions on Intelligent Systems and Technology* 11.1 (2020), 10:1-25.
- [18] D. Hendrycks y K. Gimpel. “Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units”. En: *CoRR abs/1606.08415* (2016). arXiv: 1606.08415.
- [19] S. Hochreiter y J. Schmidhuber. “Long Short-Term Memory”. En: *Neural Computation* 9.8 (nov. de 1997), págs. 1735-1780.
- [20] E. Keogh y C. A. Ratanamahatana. “Exact Indexing of Dynamic Time Warping”. En: *Knowledge and Information Systems* 7.3 (2005), págs. 358-386.
- [21] X. Li, K. Zhao, G. Cong, C. S. Jensen y W. Wei. “Deep Representation Learning for Trajectory Similarity Computation”. En: *Proceedings of the 34th IEEE International Conference on Data Engineering*. 2018, págs. 617-628.
- [22] B. Lin y J. Su. “One Way Distance: For Shape based Similarity Search of Moving Object Trajectories”. En: *GeoInformatica* 12 (2008), págs. 117-142.
- [23] X. Liu, X. Tan, Y. Guo, Y. Chen y Z. Zhang. “CSTRM: Contrastive Self-Supervised Trajectory Representation Model for Trajectory Similarity Computation”. En: *Computer Communications* 185 (2022), págs. 159-167.
- [24] I. Loshchilov y F. Hutter. “Fixing Weight Decay Regularization in Adam”. En: *CoRR abs/1711.05101* (2017). arXiv: 1711.05101.
- [25] P.-F. Marteau. “Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2 (2008), págs. 306-318.
- [26] H.-S. Park y C.-H. Jun. “A Simple and Fast Algorithm for K-medoids Clustering”. En: *Expert Systems with Applications* 36.2 (2009), págs. 3336-3341.

- [27] S. Ranu, P. Deepak, A. D. Telang, P. Deshpande y S. Raghavan. "Indexing and Matching Trajectories under Inconsistent Sampling Rates". En: *Proceedings of the 31st IEEE International Conference on Data Engineering*. 2015, págs. 999-1010.
- [28] S. Salvador y P. Chan. "Toward accurate dynamic time warping in linear time and space". En: *Intelligent Data Analysis* 11.5 (oct. de 2007), págs. 561-580.
- [29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". En: *Journal of Machine Learning Research* 15.56 (2014), págs. 1929-1958.
- [30] Y. Tao, A. Both, R. I. Silveira, K. Buchin, S. Sijben, R. S. Purves, P. Laube, D. Peng, K. Toohey y M. Duckham. "A comparative analysis of trajectory similarity measures". En: *GIScience & Remote Sensing* 58.5 (2021), págs. 643-669.
- [31] R. Tavenard. *An introduction to Dynamic Time Warping*. <https://rtavenar.github.io/blog/dtw.html>. 2021.
- [32] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar y E. Woods. "Tslearn, A Machine Learning Toolkit for Time Series Data". En: *Journal of Machine Learning Research* 21.118 (2020), págs. 1-6.
- [33] D. A. Tedjopurnomo, X. Li, Z. Bao, G. Cong, F. Choudhury y A. K. Qin. "Similar Trajectory Search with Spatio-temporal Deep Representation Learning". En: *ACM Transactions on Intelligent Systems and Technology* 12.6 (2021), 77:1-26.
- [34] M. Todoric y T. Mastelic. "Comparison of Similarity Measures for Trajectory Clustering - Aviation Use Case". En: *Journal of Communications Software and Systems* 19.2 (jun. de 2023), págs. 178-187.
- [35] M. Vlachos, G. Kollios y D. Gunopulos. "Discovering Similar Multidimensional Trajectories". En: *Proceedings of the 18th International Conference on Data Engineering*. 2002, págs. 673-684.
- [36] Z. Wang, C. Long, G. Cong y C. Ju. "Effective and Efficient Sports Play Retrieval with Deep Representation Learning". En: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. 2019, págs. 499-509.
- [37] P. Yang, H. Wang, D. Lian, Y. Zhang, L. Qin y W. Zhang. "TMN: Trajectory Matching Networks for Predicting Similarity". En: *Proceedings of the 38th IEEE International Conference on Data Engineering*. 2022, págs. 1700-1713.
- [38] P. Yang, H. Wang, Y. Zhang, L. Qin, W. Zhang y X. Lin. "T3S: Effective Representation Learning for Trajectory Similarity Computation". En: *Proceedings of the 37th IEEE International Conference on Data Engineering*. 2021, págs. 2183-2188.
- [39] S. B. Yang, J. Hu, C. Guo, B. Yang y C. S. Jensen. "LightPath: Lightweight and Scalable Path Representation Learning". En: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, págs. 2999-3010.
- [40] D. Yao, G. Cong, C. Zhang y J. Bi. "Computing Trajectory Similarity in Linear Time: A Generic Seed-guided Neural Metric Learning Approach". En: *Proceedings of the 35th IEEE International Conference on Data Engineering*. 2019, págs. 1358-1369.

- [41] D. Yao, H. Hu, L. Du, G. Cong, S. Han y J. Bi. “TrajGAT: A Graph-based Long-term Dependency Modeling Approach for Trajectory Similarity Computation”. En: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, págs. 2275-2285.
- [42] H. Zhang, X. Zhang, Q. Jiang, B. Zheng, Z. Sun, W. Sun y C. Wang. “Trajectory Similarity Learning with Auxiliary Supervision and Optimal Matching”. En: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 2020, págs. 11-17.