



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y
Tecnología

Máster Universitario en Inteligencia Artificial

Análisis de filtros inteligentes para ecos de radares de automoción

Trabajo fin de estudio presentado por:	Jaime González Gómez
Tipo de trabajo:	Comparativa de soluciones
Director/a:	Almudena Ruiz Iniesta
Fecha:	20 de julio de 2023

Resumen

Este proyecto busca una solución para la aplicación de un sistema inteligente de filtrado de *clutter* que pueda ser implementado en los radares de tráfico ofrecidos en el mercado. Este filtro inteligente tiene la responsabilidad de limpiar la señal medida para así disminuir los costes de procesamiento posteriores, debido a la disminución de datos a analizar. Esto implica alejarse de los modelos de redes neuronales, los cuales no pueden ser utilizados en la mayoría de los sistemas portables, y el entrenamiento de modelos clásicos de clasificación, ajustándose así a unos criterios de memoria ocupada y tiempo de procesamiento. Las tareas del proyecto han sido organizadas con la metodología Kanban, enumerando así todas las tareas a realizar. Los resultados muestran que, aunque los resultados finales no son los deseados para el propósito inicial, el objetivo es alcanzable utilizando datos bien etiquetados.

Palabras clave: automoción, *clutter*, detección de objetos, radar, tráfico

Abstract

This project seeks a solution for the application of an intelligent *clutter* filtering system that can be implemented in the traffic radars offered in the market. This intelligent filter has the responsibility to clean the measured signal to decrease the subsequent processing costs, due to the decrease of data to be analyzed. This implies moving away from neural network models, which cannot be used in most portable systems, and the training of classical classification models, thus adjusting to the criteria of occupied memory and processing time. The project tasks have been organized using the Kanban methodology, thus listing all the tasks to be performed. The results show that, although the results are not the desired ones for the initial purpose, the objective is achievable using well labeled data.

Keywords: automotive, *clutter*, object detection, radar, traffic

Índice de contenidos

1. Introducción	1
1.1. Motivación	1
1.2. Planteamiento del problema	2
1.3. Estructura de la memoria	3
2. Contexto y estado del arte	6
2.1. Inicios de la conducción autónoma	6
2.2. Sensores referentes a la automoción	9
2.3. Radares. Ecos y <i>clutter</i>	10
2.3.1. Teoría básica de un sistema radar	11
2.3.2. Clasificación del clutter	12
2.4. Estudios previos y evolución de la conducción autónoma	13
2.5. Limitaciones de proyectos de <i>machine learning</i> para la automoción	18
2.6. Aportación del trabajo	19
3. Objetivos y metodología de trabajo	21
3.1. Objetivo general	21
3.2. Objetivos específicos	21
3.2.1. Tareas presentes en el proyecto	22
4. Planteamiento de la comparativa	25
4.1. Sistemas utilizados en el estudio	26
4.2. Escenarios de estudio	27
4.2.1. Escenario: Autovía	31
4.2.2. Escenario: Intersección	38
4.3. Caracterización del problema	44

4.3.1.	Análisis de los datos.....	44
4.3.2.	Decisión de la tarea a abordar.....	48
4.4.	Definición de los modelos.....	50
4.4.1.	Clasificador de Bayes	50
4.4.2.	Random Forest	51
4.4.3.	Máquinas Vector de Soporte (SVM).....	51
4.4.4.	Clasificador de K-Vecinos.....	52
4.5.	Estrategia para el entrenamiento.....	53
4.5.1.	Vector de entrada de los modelos	53
4.5.2.	Ajuste de los hiperparámetros	53
4.5.3.	Selección y aceptación del filtro final.....	56
4.6.	Métricas de la comparativa	56
5.	Desarrollo de la comparativa	58
5.1.	Tratamiento de los datos.....	58
5.2.	Entrenamiento y obtención de los modelos.....	59
5.2.1.	Planificación de los entrenamientos	60
5.2.2.	Escenario de autovía.....	62
5.2.3.	Escenario de intersección.....	67
5.3.	Testeo de los modelos en datos sin etiquetar.....	70
5.3.1.	Filtros para autovía.....	70
5.3.2.	Filtros para intersección	73
6.	Discusión y análisis de resultados	77
7.	Conclusiones y trabajo futuro	80
7.1.	Conclusiones	80

7.2. Líneas de trabajo futuro	82
Referencias bibliográficas.....	83
Anexo A. Artículo de investigación.....	87

Índice de figuras

Figura 1: Sistema integrado en el vehículo de Dickmanns. Fuente: (García Oliva, 2018).....	7
Figura 2: Representación de los niveles de conducción autónoma. Fuente: (km77)	8
Figura 3: Sistemas de sensores en un vehículo inteligente. Fuente: (Gomez, 2017).....	10
Figura 4: Representación de las mediciones principales de un radar de automoción	12
Figura 5: Tres diferentes casos de reflejos de espejo. Fuente: (Holder, Linnhoff, Rosenberger, Popp, & Winner, 2019)	13
Figura 6: Detección de vehículos basado en los datos escaneados. Fuente: (Keat, Pradalier, & Laugier, 2005)	14
Figura 7: Configuración del vehículo utilizado en la creación del KITTI (Geiger, Lenz, & Urtasun, 2012).....	14
Figura 8: Modelo propuesto para <i>DeepReflects</i> . Fuente: (Ulrich, Bosch, Gläser, & Timm, 2021)	16
Figura 9: Gráfica Rango - Tiempo de un trayecto.....	27
Figura 10: Gráfica Velocidad radial - Tiempo de un trayecto.....	28
Figura 11: Gráfica Azimut - Tiempo de un trayecto	28
Figura 12: Gráfica RCS - Tiempo de un trayecto.....	29
Figura 13: Gráfica Rango – Ciclos en <i>clutter</i> de tipo espejo.....	30
Figura 14: Gráfica Velocidad radial – Ciclos en <i>clutter</i> de tipo espejo.....	30
Figura 15: : Gráfica Azimut – Ciclos en <i>clutter</i> de tipo espejo.....	31
Figura 16: Escenario de autovía. Gráfica de Rango.	32
Figura 17: Escenario de autovía. Gráfica de Velocidad radial.	32
Figura 18: Escenario de autovía. Gráfica de Azimut.....	33
Figura 19: Escenario de autovía. Gráfica de RCS.	34
Figura 20: Escenario de autovía. Representación de trayectos en la gráfica Rango.....	35

Figura 21: Escenario de autovía. Representación de trayectos en la gráfica Velocidad radial.	35
Figura 22: Escenario de autovía. Representación de trayectos en la gráfica Azimut.	36
Figura 23: Escenario de autovía. Representación del <i>clutter</i>	36
Figura 24: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Rango. .	37
Figura 25: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Velocidad radial.	37
Figura 26: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Azimut.	38
Figura 27: Escenario de intersección. Gráfica de Rango.	39
Figura 28: Escenario de intersección. Gráfica de Velocidad radial.	40
Figura 29: Escenario de intersección. Gráfica de Azimut.	40
Figura 30: Escenario de intersección. Gráfica de RCS.	41
Figura 31: Escenario de intersección. Representación del <i>clutter</i>	41
Figura 32: Escenario de intersección. Caso particular de peatones en la gráfica de Rango. ...	42
Figura 33: Escenario de intersección. Caso particular de peatones en la gráfica de Velocidad radial.	42
Figura 34: Escenario de intersección. Caso particular de peatones en la gráfica de Azimut...	43
Figura 35: Escenario de intersección. Caso particular de interferencias en la gráfica de Rango.	43
Figura 36: Escenario de intersección. Caso particular de interferencias en la gráfica de Velocidad radial.	44
Figura 37: Escenario de intersección. Caso particular de interferencias en la gráfica de Azimut.	44
Figura 38: Escenario de autovía. Matriz triangular de correlación.	48
Figura 39: Escenario de intersección. Matriz triangular de correlación.	48

Figura 40: Estimaciones de autovía con el filtro de Random Forest.....	72
Figura 41: Estimaciones de autovía con el filtro de SVM.....	72
Figura 42: Estimaciones de autovía con el filtro de K-Vecinos.....	73
Figura 43: Estimaciones de intersección con el filtro de SVM.....	75
Figura 44: Estimaciones de intersección con el filtro de K-Vecinos.....	75
Figura 45: Gráfica de memoria utilizada por los filtros.....	78
Figura 46: Gráfica de tiempo de procesamiento de los filtros por detección.....	78

Índice de tablas

Tabla 1: Datos generales del escenario de autovía	32
Tabla 2: Datos generales del escenario de intersección	39
Tabla 3: Escenario de autovía. Tabla de variables.....	45
Tabla 4: Escenario de intersección. Tabla de variables.	46
Tabla 5: Escenario de autovía. Conteo, media y percentiles de los valores.	46
Tabla 6: Escenario de intersección. Conteo, media y percentiles de los valores.....	47
Tabla 7: Datos de entrenamiento y test después de la separación.	60
Tabla 8: Resultados del clasificador Bayesiano para autovía.....	62
Tabla 9: Resultados del clasificador Random Forest para autovía.....	63
Tabla 10: Matriz de confusión del modelo de Random Forest seleccionado para autovía.	64
Tabla 11: Resultados de la primera iteración del modelo de SVM en autovía.	64
Tabla 12: Resultados de la segunda iteración del modelo de SVM en autovía.	65
Tabla 13: Matriz de confusión del clasificador SVM seleccionado para autovía.	65
Tabla 14: Resultados de la primera iteración de K-Vecinos para autovía.....	66
Tabla 15: Resultados de la segunda iteración de K-Vecinos para autovía.....	66
Tabla 16: Matriz de confusión del clasificador K-Vecinos seleccionado para autovía.....	67
Tabla 17: Resultados del clasificador Bayesiano para intersección.	67
Tabla 18: Resultados del clasificador Random Forest para intersección.....	68
Tabla 19: Resultados del clasificador SVM para intersección.	69
Tabla 20: Matriz de confusión del clasificador SMV seleccionado para intersección.	69
Tabla 21: Resultados del clasificador de K-Vecinos para intersección.....	70
Tabla 22: Matriz de confusión del clasificador K-Vecinos seleccionado para intersección.	70
Tabla 23: Tiempos de procesamiento con los filtros para autovía.	71

Tabla 24: Estimaciones de clutter con los filtros para autovía.....	71
Tabla 25: Tiempos de procesamiento con los filtros para intersección.....	74
Tabla 26: Estimaciones de clutter con los filtros para intersección.....	74
Tabla 27: Comparación de los resultados en los filtros de autovía.....	77
Tabla 28: Comparación de los resultados en los filtros de intersección.....	77

1. Introducción

La evolución del campo de la inteligencia artificial ha supuesto también grandes avances en la tecnología, desde algoritmos que permiten detectar y analizar anomalías a través de radiografías para el ámbito médico hasta asistentes de voz que pueden instalarse en cualquier hogar. En las últimas décadas, esta evolución ha avanzado en conjunto con la implementación de avances para conseguir un vehículo autónomo.

El primer y mayor inconveniente para la obtención de un vehículo autónomo es conseguir que el autómatas desarrolle una percepción de su entorno que sea en tiempo real, que permita generalizar en distintos entornos (como dentro de una ciudad o en autopista) y que permita identificar los obstáculos que lo rodean. Este sistema de percepción se ha intentado desarrollar con diferentes estrategias, utilizando generalmente cámaras, sistemas radar y/o sistemas LIDAR como sensores. Esta característica es un problema que puede ser resuelto con varios modelos o varias perspectivas distintas, pero sigue manteniéndose como uno de los mayores inconvenientes debido al coste económico y computacional que supone trasladar la estrategia a un modelo de mercado asequible para el usuario medio.

El presente trabajo pretende aportar una estrategia diferente para abordar este problema, específico para las soluciones radar, cuyo objetivo es preprocesar la información previa a la clasificación de objetos. Para ello, se propone el estudio de varios modelos inteligentes, como la creación de filtros inteligentes para falsas detecciones en los datos obtenidos por el radar.

La solución se encuentra enfocada para escenarios de control de tráfico, debido a que el sistema radar se encontrará siempre estático simplificando el problema, con el objetivo de conseguir una aplicación de filtrado que pueda ser implementado en sistemas de una capacidad de memoria y procesamiento limitada. Todos los modelos propuestos en el trabajo han sido entrenados con datos reales y no simulados.

1.1. Motivación

Varias de las situaciones mencionadas se han implementado en diversos casos. Muchos de estos plantean el problema, ya mencionado, de que el producto final sería un vehículo muy costoso para el ciudadano medio, debido a la tecnología que lo compone, o necesitaría de una

alta potencia de procesamiento en un sistema portable y con una eficiencia energética no muy elevada. Algunos ejemplos de estos casos son vehículos dotados de un sistema LIDAR, como el utilizado para la creación de la base de datos KITTI (Geiger, Lenz, & Urtasun, 2012) o el vehículo autónomo de Tesla.

Mientras que actualmente la inteligencia artificial se encuentra muy enfocada en la investigación y resolución de problemas a través de redes neuronales profundas, el enfoque actual de la industria para vehículos inteligentes se centra en el uso de modelos más simples y que ocupen menos uso de memoria, a la vez que una apuesta por los sistemas radar en vez de los sistemas LIDAR.

Actualmente, la clasificación de objetos a partir de datos de entrada de un sistema radar se percibe como uno de los grandes retos. Las soluciones propuestas de la comunidad científica no se centran únicamente en qué modelos utilizar, sino también en cómo tratar los datos de entrada, cómo normalizarlos y cuáles seleccionar para su uso en el modelo.

Por todo ello, en este trabajo se aborda el objetivo de filtrar los datos y simplificar este primer estado para seleccionarlos. Se analizan, posteriormente, cómo el uso diversos modelos simples se adaptan como filtros inteligentes para los ecos en las detecciones de los radares. El objetivo también se basa en la idea de evitar el uso de redes neuronales pues, aunque puedan retornar unos resultados excelentes, es actualmente muy complejo una puesta en marcha de redes neuronales en sistemas con limitación de potencia, memoria y capacidad de procesamiento de datos.

1.2. Planteamiento del problema

Los ecos de radar son toda aquella captación de este, debido a la captación de las señales emitidas del mismo. En estos ecos se encuentran las detecciones deseadas, las cuales permiten la detección de objetos en el campo de la automoción, al igual que el ruido o detecciones falsas (*clutter*). Mientras que algunas veces puede ser sencillo diferenciar las detecciones falsas de las verdaderas, si por ejemplo muestran una velocidad muy alta en un entorno donde es imposible, en otros casos puede ser muy complicado diferenciarlos de los ecos verdaderos, únicamente comparando sus características individualmente.

En el campo de la inteligencia artificial es muy conocida la frase *“garbage in garbage out”*, indicando que si se entrena un modelo con malos datos de entrada la salida esperada también tendrá una baja confianza. Esta frase puede utilizarse en una gran variedad de campos matemáticos y no solo en la inteligencia artificial, pues el uso de buenos datos es crucial para un buen procesamiento independientemente de la aplicación.

En este trabajo se plantea buscar una solución para los malos datos de entrada de un modelo clasificador, buscando el entrenamiento de modelos inteligentes portables para que funcionen como filtro de los ecos falsos del radar. Para ello, se plantea que las detecciones falsas pueden ser encontradas y clasificadas si se comparan varias de sus características a la vez, presentando las detecciones falsas no linealidades extremas continuamente en al menos una de estas características.

Los modelos por definir en el trabajo serán, por ello, entrenados con datos reales obtenidos con radares de tráfico de automoción. Estos datos definirán unos modelos aplicados a casos reales en los que el radar se encuentra en una posición estática y, por lo tanto, los ecos del radar pueden ser analizados con una mayor facilidad y precisión.

1.3. Estructura de la memoria

En este trabajo se desglosarán, a continuación, los siguientes capítulos:

- **Capítulo 2: Contexto y estado del arte**, donde se expone el contexto del estudio en cuestión. Este se desglosa en varias secciones, comenzando por una introducción a la historia de los vehículos inteligentes. A continuación, se explican las nociones básicas del funcionamiento de un radar, así como sus ventajas generales, para poder explicar los distintos tipos de falsas detecciones que se pueden encontrar en estos sistemas. Por último, se expone el estado del arte, en el que se desglosan otras investigaciones relacionadas con el ámbito de la conducción inteligente, introducidas tal como un resumen y vistazo rápido por toda la cronología de la rama de investigación dedicada al ámbito de vehículos inteligentes y del enfoque dado por los avances en la inteligencia artificial. Finalmente se expondrán las limitaciones de los proyectos de machine learning en aplicaciones de automoción, consolidando las razones de por qué

el estudio a presentar es importante, tanto para la industria como para el ámbito científico de investigación.

- **Capítulo 3: Objetivos y metodología de trabajo** reúne las metas finales que se esperan alcanzar, con este proyecto, previas a los análisis y resultados que se obtendrán. En este capítulo se especifican los objetivos generales y cuantificados, ofreciendo el contexto previo para mostrar el conocimiento a aportar por el trabajo. El trabajo se desarrolla mediante una metodología Kanban, la cual se define en este capítulo dando las razones de su elección. Posteriormente se enumeran y analizan todas las tareas a realizar, ofreciendo una explicación de la planificación para su desarrollo.
- **Capítulo 4: Planteamiento de la comparativa** recopila todos los modelos a utilizar en el trabajo, al igual que las razones de selección de los mismos. Estos modelos son clasificados, dentro del trabajo, como modelos portables, los cuales buscan poder cumplir el objetivo de que el modelo inteligente desarrollado pueda ser incluido en un sistema radar con características similares a los ya existentes en el mercado de radares de tráfico y automoción. Se desarrollan los escenarios de automoción sobre los que se crearán los modelos (control de tráfico para autovía y para intersecciones), al igual que un análisis para cada escenario sobre las detecciones, al igual que las características necesarias para definir una detección como verdadera. Posteriormente, se analizan y muestran las grabaciones que se usan para el entrenamiento de los modelos, explicando cómo se obtiene la clasificación de los datos en las mismas. Al ser el método de clasificación puramente humano y no matemático, se espera que puedan encontrarse ciertas inconsistencias. Finalmente, se explican y desarrollan los modelos a utilizar, indicando las razones por las que los modelos son seleccionados para la tarea y las expectativas de cada modelo.
- **Capítulo 5: Desarrollo de la comparativa** comienza explicando la extracción de características para su uso en los entrenamientos del modelo. Posteriormente se muestra en su contenido el entrenamiento de los modelos, así como la matriz de confusión de cada uno junto con las métricas y la analítica de sus resultados.
- **Capítulo 6: Discusión y análisis de los resultados**, conteniendo el resultado organizado en una tabla de todos los resultados conjuntos del capítulo anterior, finalizado por una

gráfica en la que se representa la eficiencia de cada modelo. Posteriormente a la presentación de los resultados conjuntos se añade el análisis sobre cuál modelo es mejor para seleccionar, teniendo en cuenta el rendimiento, la calidad de las predicciones y el tamaño del mismo modelo.

2. Contexto y estado del arte

En este capítulo sirve de introducción para todos los temas que se abordarán en el trabajo. Por ello, el capítulo se encuentra dividido en varias secciones:

- **Historia de la conducción autónoma:** Se explica y desarrolla el marco histórico de los comienzos de la conducción autónoma, así como los niveles que definen la autonomía de un vehículo.
- **Sensores referentes a la automoción:** En el que se enumeran y definen los distintos y más utilizados sensores con la finalidad de la creación de vehículos inteligentes.
- **Radares. Ecos y clutter:** Se explican los conceptos básicos de los sistemas radar, así como ruido existente en sus detecciones, definido como *clutter*. También se enuncian las técnicas matemáticas actuales para el filtrado de *clutter*.
- **Estado del arte:** Donde se agrupan todos los estudios académicos del campo de la inteligencia artificial relacionados con la conducción autónoma. Estos estudios se encuentran organizados cronológicamente, teniendo como líneas concluyentes aquellas últimas investigaciones basadas en aplicaciones inteligentes para el filtrado de *clutter*.
- **Limitaciones de proyectos de *machine learning* para la automoción:** Enunciando el gran problema que presenta el intentar instalar aplicaciones de *machine learning* en sistemas que puedan ser usados hoy en día por la sociedad. Al presentarse estos inconvenientes, se desarrolla como conclusión la aportación de este trabajo.

2.1. Inicios de la conducción autónoma

El primer hecho catalogado de vehículos autónomos proviene de la compañía Milwaukee durante los años 20. “Milwaukee realizó una demostración, que, aunque algo primitiva, podemos decir que fue la semilla del coche autónomo, tuvo lugar en diciembre de 1926 gracias a un sistema capaz de controlar un vehículo por radio” (García Oliva, 2018). A este hecho le precedió la exposición del coche eléctrico de Norman Bel Geddes en el pabellón Futurama, en la Exposición Universal de Nueva York de 1939. Norman presentó un vehículo eléctrico, el cual se desplazaba gracias a un circuito integrado en el pavimento (López, 2020). Mientras que estas fueron las primeras aportaciones de los vehículos autónomos, la definición

de “autónomo” utilizada hasta la fecha hacía referencia a un vehículo no manejado por un humano en su interior.

Ernst Dickmanns desarrolló el primer vehículo autónomo moderno en 1987, el cual incluía un sistema basado en inteligencia artificial. El vehículo funcionaba a través de un sistema integrado, como se muestra en la Figura 1, el cual analizaba el entorno del vehículo con una cámara de video. Este alcanzó velocidades de 100 km/h en autopistas sin tráfico.



Figura 1: Sistema integrado en el vehículo de Dickmanns. Fuente: (García Oliva, 2018)

Durante la década de los 90, tanto Defense Advance Research Projects Agency (DARPA) como Dickmanns continuaron sus investigaciones realizando varias aportaciones al campo, precediendo al invierno de las investigaciones en este campo. No fue hasta el año 2012 en el que se realizaron ensayos de conducción autónoma, organizados por grupos españoles. Estos ensayos reavivaron el interés académico y comercial en el campo de la conducción autónoma.

El entendimiento que se contiene hoy en día de un vehículo autónomo se basa en los seis niveles de conducción autónoma (km77):

- **Nivel 0:** En vehículo no se presenta ninguna automatización, teniendo el conductor que realizar todas las tareas que engloban la conducción.
- **Nivel 1:** El vehículo presenta aplicaciones básicas de asistencia para el conductor, recayendo todavía la gran mayoría de las tareas sobre el conductor. Un ejemplo de una aplicación de nivel 1 es el manejo automático del volante para aparcar el vehículo, recayendo sobre el conductor el manejo de los pedales.

- **Nivel 2:** Este es una extensión del nivel anterior, ofreciendo una automatización parcial para la conducción en casos específicos. Pudiendo el vehículo controlar sus movimientos laterales y longitudinales, el conductor sigue siendo el responsable en cualquier situación, ya que la parada o cambio de ruta frente a obstáculos no se encuentra incluida en el nivel.
- **Nivel 3:** Se ofrece una automatización completa pero condicional. Mientras el vehículo ofrece una automatización total de la conducción en ciertas tareas, el conductor sigue siendo responsable en toda situación. El mismo ha de intervenir y retomar el control del vehículo frente a cualquier situación de riesgo o que el vehículo no interprete correctamente.
- **Nivel 4:** El vehículo está capacitado para una alta automatización de la conducción. Dentro de unos ámbitos de funcionamiento especificados, no se requiere de la asistencia del conductor. Aun así, frente a casos de peligro sigue siendo mandatorio que el conductor retome el control del vehículo. Hoy en día, ya existen vehículos que se clasifican de nivel 4.
- **Nivel 5:** El vehículo se encuentra completamente capacitado para la conducción autónoma, sin requerir de asistencia o presencia de un conductor.

La Figura 2 muestra una representación de estos seis niveles.

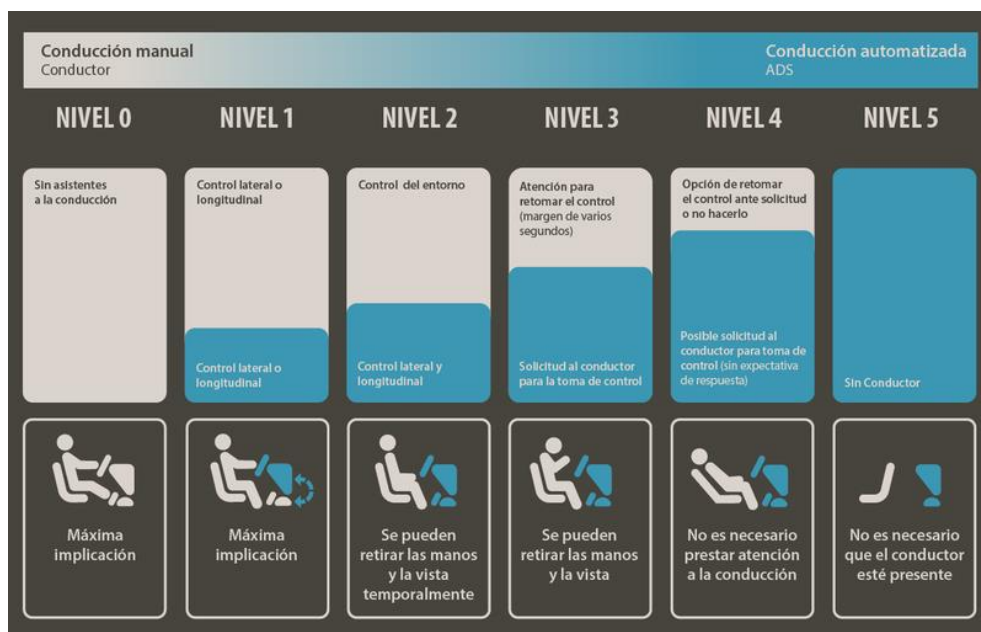


Figura 2: Representación de los niveles de conducción autónoma. Fuente: (km77)

2.2. Sensores referentes a la automoción

La creación de un vehículo autónomo supone la creación de un robot que pueda desplazarse, identificar e interactuar con su entorno. Los sensores dotan al robot de información que, siendo esta procesada, permite la comunicación de este con su entorno. En el campo de los vehículos autónomos los sensores se clasifican en activos y pasivos (Dawkins, 2021) en función de si necesitan emitir una señal para poder percibir (sensores activos) o pueden detectar el entorno sin la emisión de ninguna onda adicional (sensores pasivos).

Los sensores más utilizados como tecnología de conducción autónoma son:

- **Sensores ultrasónicos:** Sensores que emiten ondas sónicas no audibles y permiten la detección de obstáculos a pocos metros y baja velocidad. Son bastante frecuentes en diversos tipos de alarmas debido a su costo económico.
- **Sistemas globales de navegación por satélite (GNSS):** Permiten la triangulación del posicionamiento de un sistema en el planeta a través de una red de satélites. El más conocido es el sistema GPS, desarrollado por el Departamento de Defensa de EE.UU. y basado en una red de 24 satélites.
- **Cámaras:** Permiten una obtención rápida del entorno, pudiendo utilizar sus datos para aplicaciones de reconocimiento de objetos. El mayor inconveniente de los sistemas de cámaras es el procesamiento de los datos obtenidos para aplicaciones en tiempo real y con un costo computacional asequible.
- **Radar (Radio Detecting And Ranging):** Se basan en un sistema de antenas para emitir y captar ondas electromagnéticas, comúnmente en la banda de frecuencias de 24 GHz (corto alcance) y la de 77 GHz (largo alcance). Los datos son procesados basándose en el efecto Doppler, descomponiendo las detecciones en rango, velocidad radial y azimut o ángulo de llegada.
- **LIDAR (Laser Imaging Detection And Ranging):** Sistema basado en la emisión y recepción de haces de láser pulsados. Permiten la creación de nubes de puntos 2D y 3D, en función del LIDAR, con un alto nivel de detalle en el contorno de los objetos detectados y el entorno. Su mayor desventaja es su alto coste económico y computacional para el tratado de los datos.

A la hora de elegir los sensores a utilizar en el vehículo para la aplicación de conducción autónoma, el coste de cada uno de ellos puede ser crucial. La Figura 3 muestra, de forma genérica, el rango de costes usuales para cada uno de los tipos de sensores.

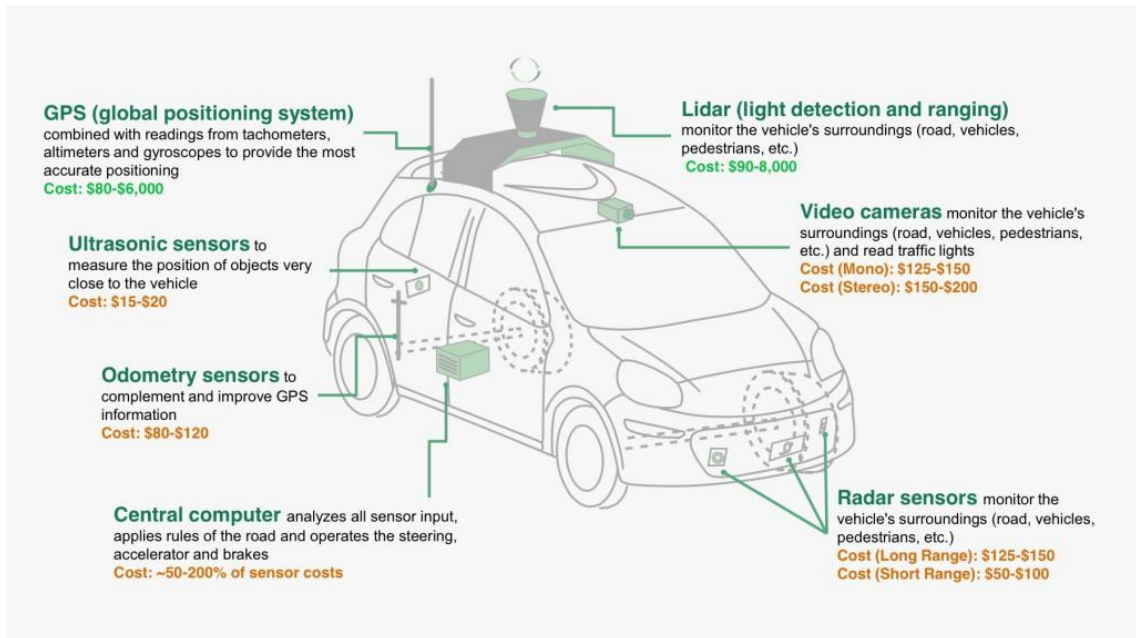


Figura 3: Sistemas de sensores en un vehículo inteligente. Fuente: (Gomez, 2017).

2.3. Radares. Ecos y clutter

Dentro de todas las posibilidades de sensores, este trabajo se centra únicamente en los radares como sistemas para la toma de datos. El uso de un sensor en concreto también se encuentra estrictamente relacionado con el entendimiento de la información proporcionada por el mismo. Por ello, esta sección se centra en explicar:

- Teoría básica de un radar: Donde se explica el funcionamiento básico de un radar, al igual que las cuatro variables que detecta y lo que indica cada una de ellas.
- Entendimiento de clutter: En muchos campos se habla de “ruido” para las variables con valores no deseados. En el campo de uso de radares el término de “clutter” hace referencia al ruido y a otras detecciones no deseadas, las cuales que no se ajustan a la definición de “ruido”.

2.3.1. Teoría básica de un sistema radar

Para entender cómo se origina el *clutter* o ruido en las detecciones del radar, es necesario conocer los principios básicos que permiten el funcionamiento del mismo. Las detecciones en rango del radar son posibles gracias a las propiedades de la energía electromagnética radiada (Wolff, s.f.).

El principio del radar se basa en radiar energía electromagnética, la cual es reflejada por superficies conductoras (reflectores). La energía radiada viaja a una velocidad constante de 300.000 km/s a través del aire, permitiendo determinar la distancia del reflector midiendo el tiempo que la onda tarda en volver desde que se envió el pulso. La energía radiada viaja en una línea recta sin grandes variaciones, permitiendo conocer también el ángulo de llegada o azimut y la elevación con un sistema de antenas emisoras y receptoras.

Debido a sus propiedades, los sistemas radar contienen múltiples ventajas técnicas:

- No es sensible a las variaciones de la luz en el entorno, pudiendo operar en buenas condiciones tanto en el día como en la noche.
- Puede operar en diversas condiciones atmosféricas como lluvia, niebla o nevadas. No obstante, lluvias muy intensas impiden su correcto funcionamiento.
- Los radares pueden utilizarse para la detección de objetos en movimiento, pudiendo ser usados en aplicaciones de reconocimiento de objetos para radares de alta resolución.

De la señal procesada del eco del radar se obtienen todas las detecciones. Cada detección se caracteriza en un vector de cuatro dimensiones:

- **Rango:** Distancia que separa a la detección del sistema radar en una línea recta. El rango es comúnmente medido en metros.
- **Velocidad radial:** Velocidad de la detección a lo largo de la línea de visión definida por la recta entre el radar y la detección. Un valor negativo indica que la detección se aproxima al radar, mientras que uno positivo indica que se aleja.
- **Ángulo de llegada (azimut):** Ángulo definido entre el eje frontal del radar y la recta imaginaria que une la detección y el sistema.

- **Intensidad (*Radar Cross Section - RCS*):** Potencia de la señal, reflejada sobre un área del objeto detectado (reflector), con la que el eco vuelve al radar. El RCS es influenciado por varios factores físicos (forma, tamaño y material que componen el objeto) y posicional (azimut), sin verse afectado por la distancia a la que se encuentra el objeto detectado.

La Figura 4, obtenida con la herramienta web “darw.io”, muestra una representación de las dimensiones físicas obtenidas por una medición de radar. La única dimensión no interpretada en el gráfico es el RCS, al ser esta una dimensión más abstracta cuyo significado puede simplificarse en “cómo de visible es un objeto para el radar”.

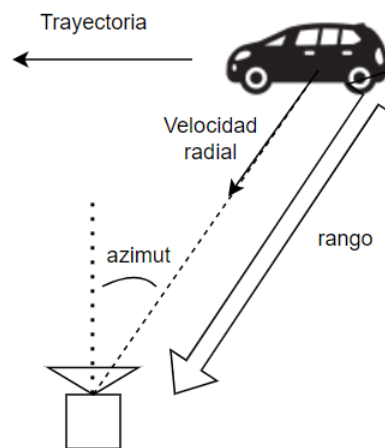


Figura 4: Representación de las mediciones principales de un radar de automoción

2.3.2. Clasificación del clutter

Debido también a las propiedades de los radares, estos pueden tener “detecciones falsas”, clasificadas como falsos positivos y falsos negativos y definidas como el resultado de distorsiones en la onda enviada por la propia tecnología. Holder et al, definen las detecciones falsas como *artefactos*: “Un artefacto en una medición de radar es una distorsión perceptible en las lecturas del sensor que es inevitable debido al principio de medición del sensor y a su diseño” (Holder, Linnhoff, Rosenberger, Popp, & Winner, 2019). Dentro de su mismo estudio, se clasifican los artefactos según su origen:

- **Reflejos de espejo (*Mirror reflections*):** Ocurren cuando la señal enviada por el radar rebota en varias superficies conductoras, definiendo este efecto como propagación

multitrayecto. El radar interpreta este fenómeno como la detección de un objeto inexistente, denominado como objeto fantasma. Los reflejos de espejo suelen caracterizarse por tener velocidades anómalas. Este fenómeno se ve representado en la Figura 5.

- **Aliasing:** Defectos introducidos al muestrear la señal con un periodo de muestreo más relajado que el del propio eco enviado por el sistema.
- **Ruido en la medición (*Clutter*):** Dentro del *clutter* se definen las detecciones que superan aleatoriamente el umbral de detección y las detecciones asociadas a objetos persistentes.
- **Procesamiento de una señal no ideal:** La señal detectada contiene ruido en sí misma, producido por no linealidades en sus propios componentes electrónicos.

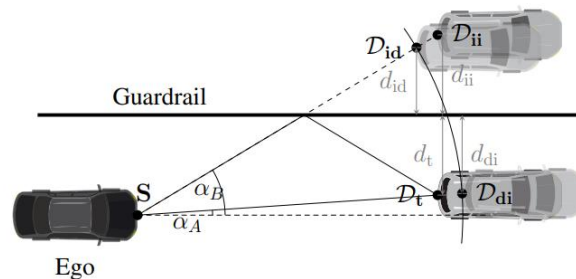


Figura 5: Tres diferentes casos de reflejos de espejo.

Fuente: (Holder, Linnhoff, Rosenberger, Popp, & Winner, 2019)

2.4. Estudios previos y evolución de la conducción autónoma

En las dos últimas décadas, la investigación de las aplicaciones de conducción autónoma ha entrado en auge, gracias también al rápido crecimiento de la inteligencia artificial en la sociedad. A medida que la tecnología y la industria avanza en el campo, buscando llevar sus resultados a mercado, se observa una tendencia de cambio en los estudios y las publicaciones.

Las primeras publicaciones se basan, en su mayoría, en sistemas LIDAR 2D. Una de ellas hace referencia a la búsqueda de espacio de aparcamiento y mapeado de la zona (Keat, Pradalier, & Laugier, 2005). Esta presenta una alta tasa de falsos negativos, necesitando el algoritmo de un primer filtrado de bordes para todos aquellos con forma de "L" (Figura 6) y una posterior etapa de clústering y segmentación para la construcción de los hipotéticos vehículos. Este proyecto de investigación también es abordado por otros con ciertas modificaciones, tales

como el uso de cámaras como segundo tipo de sensor del sistema (Sun, Bebis, & Miller, 2006), presentando aun así unos resultados similares a los presentados por Keat et al. No obstante, ambas investigaciones fallan en el mismo objetivo de poder utilizar el algoritmo descrito en tiempo real.

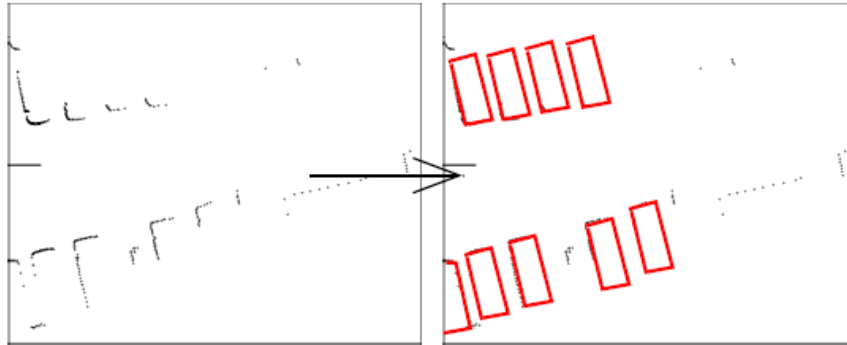


Figura 6: Detección de vehículos basado en los datos escaneados. Fuente: (Keat, Pradalier, & Laugier, 2005)

El entorno científico comienza a interesarse más en los posteriores años gracias a los avances tecnológicos, gracias a la creación de bases de datos como KITTI (Geiger, Lenz, & Urtasun, 2012), presentando material para investigaciones de odometría, semántica y detección de objetos gracias a los datos captados por LIDAR y cámaras (Figura 7). Estas mejoras tecnológicas también animan posteriores investigaciones, como la toma de testigo de Keat et al. para aplicaciones de parking usando LIDAR 2D de rango (Zhou, Navarro-Serment, & Hebert, 2012), utilizando extracción de características y clasificación con métodos de inteligencia artificial.

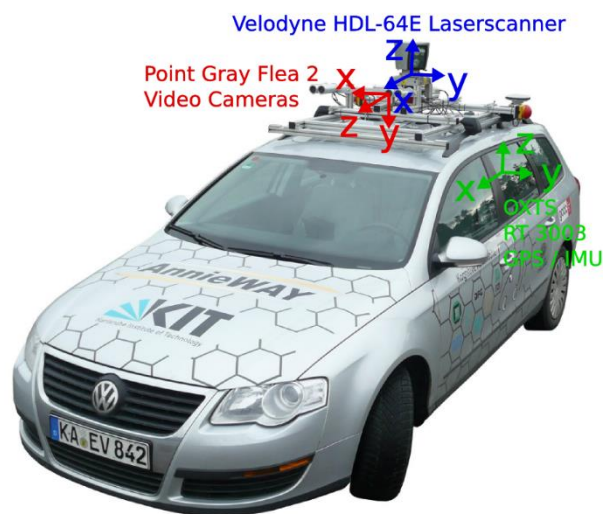


Figura 7: Configuración del vehículo utilizado en la creación del KITTI (Geiger, Lenz, & Urtasun, 2012)

Estos producen un punto de inflexión, en el que se comienzan a publicar una mayor cantidad de artículos con sistemas de radar. Siendo los sistemas radar más asequibles económicamente para el mercado y disponiendo datos que pueden ser procesados en tiempo real, aparecen investigaciones retomando la línea de trabajo anterior, Dube et al. presenta un algoritmo que funciona en tiempo real para aplicaciones de detección de vehículos estacionados (Dube, Hahn, Schutz, Dickmann, & Gingras, 2014). Esta investigación se basa en el *occupancy grid*, un antiguo método diseñado para la percepción de robots (Elfes, 1989).

También comienza a ser el auge de las redes neuronales, publicándose investigaciones de métodos bayesianos para reducir el *overfitting* (Maeda, 2014), conferencias acumulando el estado del arte sobre esta rama de la ciencia (Lecun, Bengio, & Hinton, 2015), metodologías para encontrar los pesos entre neuronas usando retropropagación (Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015) y nuevas propuestas de redes bayesianas para el trabajo con un gran volumen de datos de entrada (Li, Hernandez-Lobato, & Turner, 2015) la cual retoma el trabajo realizado 10 años atrás (Minka, 2001) modificando el algoritmo propuesto en su comienzo.

Con ello, la tecnología radar se empieza a investigar juntamente con la inteligencia artificial, basándose estas primeras publicaciones en modelos simples, como una modificación del modelo *Random Forest* incluyendo un filtro alfa-beta para la clasificación de objetos en entornos costeros (Jochumsen, Ostergaard, Jensen, Clemente, & O. Pedersen, 2016).

Esta nueva tendencia de investigación se basa en el complicado reto de la clasificación de objetos, cada modelo basándose en distintas estrategias y mostrando sus resultados. El modelo propuesto por Scheiner et al. busca realizar un agrupamiento previo, usando DBSCAN sobre las detecciones del radar, para usar esta información como datos de entrada para los modelos de *Random Forest* y LSTM (Scheiner, Appenrodt, Dickmann, & Sick, 2018). El mismo reto es también abordado a través de redes neuronales profundas utilizando la matriz azimuth-rango (Patel, y otros, 2019). Estos son puntos de partidas retomados para un cubo 3D de radar azimuth-rango-doppler (Palffy, Dong, Kooij, & Gavrila, 2020), extrayendo de este otros sub-cubos inferiores usando DBSCAN para tomarlos como entrada de un modelo CNN.

Posteriormente se publican nuevas estrategias para el mismo reto, siendo uno *DeepReflects* (Ulrich, Bosch, Gläser, & Timm, 2021): un modelo de redes neuronales profundas en la información obtenida de los reflectores detectados por el radar, simplificando también las clases a etiquetar a 4 (Figura 8). Simultáneamente, también se presenta *DeepHybrid* (Cozma, Morgan, Stolz, Stoeckel, & Rambach, 2021), otro modelo de redes neuronales profundas que toma una entrada híbrida a partir de las matrices rango-azimut y rango-doppler.

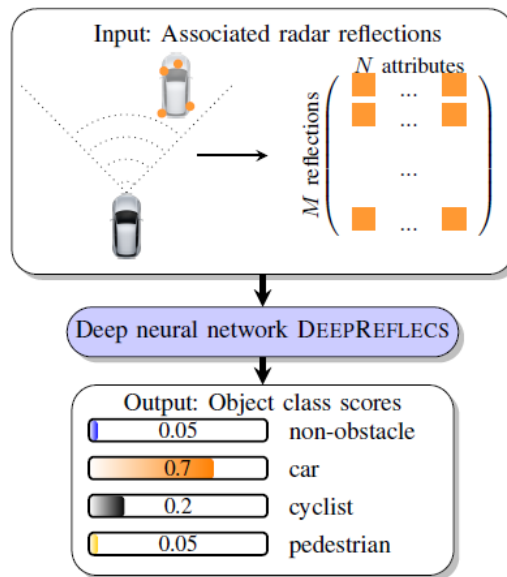


Figura 8: Modelo propuesto para *DeepReflects*. Fuente: (Ulrich, Bosch, Gläser, & Timm, 2021)

Mientras los modelos anteriores presentan buenos resultados, divergen de la idea que busca una compatibilidad de las aplicaciones e investigaciones dentro del mercado actual, pues un modelo de redes neuronales profundo necesita de unos requerimientos costosos para soluciones portables. Retomando esta idea, se publica también un modelo basado en múltiples SVM en cascada (Arab, Ghaffari, Chioukh, Tatu, & Dufour, 2021), creando también así un modelo de caja blanca con unos resultados excelentes, pero únicamente probado sobre datos simulados o medidos en un laboratorio.

Al encontrarse el campo de la investigación de aplicaciones ADAS con radar con estudios previos muy selectos, la comunidad comienza a escribir publicaciones con objetivos alternativos, como la detección de objetos fantasmas (Giebenhain, Wang, Anklam, & Goldluecke, 2021). También se encuentran publicaciones de extensas metodologías que puedan ser reutilizadas por los radares 4D de la próxima generación (Zhou, y otros, 2022).

Concluyendo los algoritmos vistos, se observa una tendencia general del procesamiento de las nubes de datos, obtenidas con sensores tipos radar o LIDAR con la posible ayuda de cámaras, para el reconocimiento de objetos. Esta fusión de sensores en conjunto con redes neuronales profundas y o convolucionales buscan la finalidad de sustituir el análisis probabilístico de los datos, orientado en los trayectos de los objetos, para escenarios de único y múltiples sensores. Naidu et al. presentan un análisis de los datos, enfocados a casos *multi sensor multi target* (Naidu, Gopalratnam, & Raol, 2003), a través del uso del Filtro de Kalman basado en Nearest Neighbours (NNKF) y filtros probabilísticos de asociación de datos (PDAF). Naidu afirma que ambos algoritmos son eficaces en presencia de *clutter* moderado, mas siguen sin obtener resultados perfectos.

Al ser este tipo de procesamiento muy costoso matemáticamente, gran parte de la comunidad científica busca la alternativa con el entrenamiento profundo para llegar el objetivo. En contraparte, este trabajo a presentar busca basarse en una de las etapas previas a la detección de objetos, pudiendo también apoyar a posteriores procesados, como el presentado por Naidu et al. Una de las grandes desventajas que poseen estos estudios es la correcta clasificación de las detecciones en trayectos, proponiéndose en este trabajo un filtro para *clutter*.

Mientras que ya existen aplicaciones que puedan detectar los trayectos de objetos en función, una solución de filtrado de *cluttering* es, en esencia, una aplicación similar a la detección de trayectos con una diferencia: no requiere de un etiquetado multiclase para los trayectos detectados. Sin este tipo de soluciones, todo el peso del filtrado de *clutter* recae en métodos estadísticos y matemáticos, como los presentados en el trabajo de Naidu et al.

El campo de investigación referente al filtrado de *clutter* se encuentra muy unido al campo de procesamiento de señal. En este, referente al proyecto a presentar, se encuentran pocas publicaciones, como el estudio de estrategias matemáticas para implementar en aplicaciones inteligentes (Addabbo, Han, Orlando, & Ricci, 2021). Addabbo et al. presentan los métodos matemáticos, basándose en la matriz de varianza y covarianza, para las diferentes estrategias de clasificación del *clutter*.

Darzikolaei et al. presentaron una estrategia perceptrón multicapa (MLP) para la clasificación del *clutter* (Darzikolaei, Ebrahimzade, & Gholami, 2015). La investigación presenta una comparación de varios modelos MLP, cada uno basado en una distinta función de activación para sus capas, basándose estas en las distribuciones de Rayleigh, Weibull, K-distribuciones y Gaussiana. Aún siendo un artículo escaso, los resultados muestran unos ratios de éxito casi perfectos para algunas distribuciones, situándose la distribución de Rayleigh como la que mejor se adapta, seguida de la distribución Gaussiana.

Ambas publicaciones se basan en los estudios proporcionados por los modelados autorregresivos de K-distribuciones (Bouvier, Martinet, Favier, Sedano, & Artaud, 1995). El trabajo se basa en la clasificación de *clutter* para sistemas radar utilizados en vehículos aéreos, presentando un único modelo MLP para clasificar el *clutter* en función del parámetro de la distribución.

Siendo las publicaciones más recientes directamente relacionadas con el filtrado de *clutter* son escasas, la aplicación es de extrema importancia para soluciones inteligentes relacionadas con el campo del radar. Un filtro con unos buenos resultados resulta en una obtención de datos de entrada más limpios, consiguiendo una clasificación más robusta tanto por métodos matemáticos tradicionales como por modelos de inteligencia artificial. Si, a su vez, el filtro es lo suficientemente ligero como para poder implementarse en sistemas portables, los recursos necesarios para posteriores procesamientos se reducirán, aumentando la eficiencia de los sistemas con sensores radar integrados.

2.5. Limitaciones de proyectos de *machine learning* para la automoción

El estudio de Darzikolaei et al. muestra una generalización óptima, la cual todavía no puede ser implementada en sistemas de automoción. Mientras que las redes neuronales presentan unos resultados excepcionales en multitud de campos, como puede ser el tratamiento de imagen o reconocimiento e interpretación de un lenguaje, presentan dos características que hacen imposible, hoy en día, su implementación en sistemas portátiles: la energía y el tamaño.

La energía constante que requiere una red neuronal, no solo para su entrenamiento sino también para su uso, es un requerimiento impensable para muchos sistemas portables. En

este trabajo, se entiende por sistema portable al conjunto de elementos que no requieren de una computadora para su funcionamiento.

Suponiendo que la energía no sea el principal problema, nos encontramos con que las redes neuronales requieren de una alta capacidad de memoria en el sistema. Proporcionar esta memoria puede ser un requerimiento imposible si no se desea que el producto final sea demasiado costoso. Un aumento en los componentes del producto final implica también un aumento en el calor que el mismo genera, requiriendo de una mejor refrigeración y, consecuentemente, un espacio mayor que cumpla dos objetivos: mejorar el rendimiento de la refrigeración y el almacenaje de todos los componentes adicionales.

Esta cadena de consecuencias es hoy en día uno de los grandes problemas por los que no se pueden instalar muchos proyectos de *machine learning* en sistemas portables. Por ello, es necesario también no solo la investigación en dos puntos principales:

- Alternativas para traer las soluciones dentro de la sociedad, en entornos con recursos muy limitados.
- Búsqueda de una mejora tecnológica para conseguir componentes más reducidos, con un menor consumo energético y un rendimiento mayor.

La investigación desarrollada en este trabajo muestra así posicionarse en la primera línea enunciada. Por ello, el aporte de este trabajo busca un gran impacto en el salto del uso de soluciones inteligentes, relacionadas con un gran procesamiento de datos, en la tecnología disponible para el grueso de la sociedad.

2.6. Aportación del trabajo

Conociendo la historia y evolución de las investigaciones de la conducción autónoma, se puede observar una clara tendencia migratoria de sistemas más potentes y costosos, como es el LIDAR, a sistemas más simples y económicos en el mercado, como son los sistemas radar en conjunto a cámaras de video. Este cambio también indica, a un costo computacional, el procesado de un número mucho menor de variables.

Uno de los grandes problemas que poseen los sistemas radar es que su información es compleja de analizar, manteniéndose hoy en día métodos matemáticos y estadísticos con umbrales para su filtrado. Una incorrecta identificación del *clutter* puede crear detecciones

fantasmas que impidan encontrar los resultados deseados, en función de la aplicación a desarrollar.

Por ello, este trabajo se basa en el análisis de la utilización de varios métodos básicos del análisis de datos con inteligencia artificial, desplegables para sistemas embebidos actuales. El objetivo es la creación de una referencia que pueda ser usada en un futuro para la obtención de mejores resultados.

3. Objetivos y metodología de trabajo

En este capítulo se presentan las bases tomadas para el diseño de los filtros y la organización tomada para llevar a cabo el trabajo. El capítulo se divide en dos secciones:

- **Objetivo general:** Se introduce el objetivo general como la referencia base para el diseño de los filtros, continuado de la importancia de la investigación que presenta el trabajo para la actualidad.
- **Objetivos específicos:** Se enumeran y describen los objetivos a realizar para la finalización del trabajo. Estos objetivos son precedidos por la introducción de los escenarios sobre los que se trabaja, al igual que las especificaciones de diseño deseadas sobre los resultados para dar como bueno a un modelo.

3.1. Objetivo general

El objetivo general de este trabajo es la creación de varios filtros de *clutter*, basados en modelos predictivos estadísticos, que puedan ser portables e instalables en sistemas radar de automoción contemporáneos.

Es imprescindible que todos los filtros que cumplan este objetivo cumplan también las siguientes características:

- Han de ser modelos que puedan ser instalados en sistemas portables, es decir, con recursos de memoria y procesamiento limitados.
- Deben ser capaces de separar los datos en detecciones verdaderas y detecciones falsas.
- Ha de ser capaz de trabajar con datos de un radar que pueda poseer interferencias de otros radares.

3.2. Objetivos específicos

Para definir un conjunto de filtros predictivos que puedan utilizarse en automoción, el trabajo analizará primero el caso más simple. Estos filtros se definirán únicamente para casos de control de tráfico, en los que el radar mantiene un posicionamiento estacionario, para aplicaciones de autopista y de intersecciones.

En estos casos estacionarios, el nivel medio de *clutter* que cada grabación puede mostrar es de casi el 50%, un número alto generado principalmente en las detecciones que se definen como ruido y por los objetos estáticos en el escenario.

Cada escenario se caracteriza principalmente por las velocidades a las que se desplazan los objetos y las trayectorias que puedan tomar. Estas dos variables definen una gran diferencia:

- *Autovía*: Caracterizado por un rango de velocidades alto, el cual se centra en la estimación general de que todo vehículo viaja entre 80 km/h y 120 km/h. En función del tramo de autovía, los vehículos no disponen de muchas posibilidades para tener trayectorias diferentes. Estas dos características permiten que el filtrado de *clutter* tipo ruido sea muy sencillo, por lo que se establece que el modelo ha de filtrar un mínimo del 85% del *clutter* del escenario.
- *Intersección*: La intersección es un escenario de tráfico dentro de poblado (velocidades limitadas superiormente a 60 km/h) definido como el que muestra la disposición de datos más caótica. Esto se debe a la detección de diferentes objetos a velocidades medias muy dispares, como puede ser la comparación entre la velocidad de un vehículo y un peatón. Adicionalmente, cada objeto posee múltiples rutas para continuar su trayectoria. Siendo la trayectoria de los vehículos la más limitada, estos pueden continuar su trayectoria en cuatro direcciones: recto, hacia uno de los lados o realizar un cambio de sentido. Debido a todo esto, es complicado diferenciar los objetos del ruido a velocidades bajas, al igual que no existe una similitud general para la comparación de las velocidades y trayectorias. Por ello, los modelos entrenados para la intersección buscan únicamente un 70% de éxito en el limpiado de clutter.

3.2.1. Tareas presentes en el proyecto

A continuación, se enumeran todos los objetivos a realizar. Los mismos se pueden entender con tareas, algunos de ellos con sus subtareas.

- **Etiquetado de datos**: Se ha de realizar un etiquetado de los datos para entrenar, marcando manualmente las detecciones que puedan conformar la trayectoria de un objeto. Toda trayectoria ha de mostrar una linealidad en sus datos de *rango*, *velocidad* y *azimut*, por lo que el etiquetado requiere de la puesta en gráficas de estos datos. Todas las grabaciones de escenarios se guardan en archivos CSV.

- Escenario de autovía: Dado una grabación como el escenario de autovía, es necesario identificar todas las trayectorias de objetos de ambas vías. El escenario proporcionado también contiene detecciones de objetos transversales, los cuales han de ser etiquetados como ruido al no ser vehículos en las vías principales.
- Escenario de intersección: Dado el escenario de intersección, identificar todas aquellas detecciones de objetos no persistentes. Al ser este un escenario extremadamente ruidoso, se supondrá que la tarea ha alcanzado su objetivo cuando la gran mayoría de objetos, siendo estos claramente identificables, hayan sido clasificados. En estos objetos también se encuentran peatones y ciclistas.
- **Preparación de los datos**: Posteriormente al etiquetado en formato CSV, los datos son exportados a formato JSON. Al realizarse un etiquetado manual, muchos de los trayectos no son identificados en orden temporal, por lo que los datos necesitan un post-procesamiento en el que se reasignan todos los valores de los trayectos asignados desde 0 hasta el máximo de trayectos en cada grabación. La asignación ha de hacerse en orden temporal. Posteriormente, los datos han de ser tratados para asimilar la forma deseada para el vector de entrada para los filtros diseñados.
- **Definición de los modelos para los filtros**: Análisis de todos los modelos a utilizar para la creación de los filtros. Al utilizarse pocos datos, será imposible realizar modelos profundos. Cada modelo se partirá en dos filtros distintos, uno para cada escenario (intersección y autovía). Es necesario también expresar el vector de características de entrada normalizado para cada filtro.
- **Entrenamiento de los filtros**: Entrenar todos los filtros para buscar la combinación de hiperparámetros óptima. El entrenamiento de los filtros requiere la monitorización de que el mismo consigue alcanzar los objetivos. En caso de que no sea posible el requerimiento de los objetivos, analizar la causa y definir una posible solución o alternativa.

- **Selección y almacenamiento de los filtros:** Selección de cada filtro, en caso de que cumpla con las especificaciones de diseño, que muestre los mejores resultados. Posteriormente almacenamiento de este en un archivo “*pk*” con la librería de *pickle*¹.
- **Comparación de los filtros:** Comparación de los filtros dentro de cada escenario basándose en los siguientes criterios:
 - Porcentaje de clutter correctamente filtrado.
 - Espacio de memoria ocupado por el modelo en cuestión.
 - Velocidad de procesamiento del modelo por detección.

¹Módulo de Python para el almacenamiento y carga de datos en formato binario.

4. Planteamiento de la comparativa

En este capítulo se analizan y desarrollan los conceptos que definen el problema, para el cual se desea implementar la solución. El capítulo se divide en las siguientes secciones:

- **Sistemas utilizados en el estudio:** Se enumeran los sistemas radar, así como sus condiciones ambientales, utilizados para la obtención de los datos. Al ser sistemas de distintas frecuencias, se anuncia la solución tomada para la normalización de los datos con respecto al tiempo. Finalizando, se mencionan cómo los sistemas utilizados componen los escenarios de estudio.
- **Escenarios de estudio:** Se presentan los escenarios desde el análisis de las detecciones, al igual que las características únicas de cada uno. En la caracterización de los escenarios se muestra también la cuantificación del ruido de cada uno, al igual que la definición para identificar un trayecto en cada uno de ellos y la presentación de casos especiales a tener en cuenta.
- **Caracterización del problema:** Se realiza un análisis de los datos referentes a ambos escenarios, identificando las variables existentes, las posibles correlaciones entre ellas y los datos anómalos existentes. Posteriormente se explica la decisión de definir el problema como de clasificación o de regresión, enunciando así la variable objetivo.
- **Definición de los modelos:** Se enumeran todos los distintos modelos a utilizar, estableciendo cuáles son las razones de cada modelo por las que se ha seleccionado. De la misma forma, cada modelo se encuentra acompañado por las desventajas de este que afectan directamente a la aplicación.
- **Estrategias de entrenamiento:** Se definen las estrategias planeadas para un ajuste óptimo de todos los modelos, definiendo el vector de entrada genérico para todos ellos. En estas se incluyen los hiperparámetros de cada modelo y que ajustes se toman, definiendo cómo se decide el valor óptimo para cada uno de ellos.
- **Métricas de la comparativa:** Recopila las métricas para la comparación de los modelos entre sí. En estas métricas se incluye la alternativa de entrenamiento en los modelos que lo necesiten, aplicando el criterio de selección entre los resultados del mismo

modelo y escenario. Por último, se explica el uso de las redes entrenadas en un archivo sin clasificar, buscando aplicar el filtro y observar los resultados de este.

4.1. Sistemas utilizados en el estudio

Para la obtención de datos, se utilizaron varios sensores de la familia UMRR-12, de largo alcance, proporcionados por la compañía Smartmicro². Los sistemas de largo alcance funcionan con frecuencias de 24 GHz para un umbral máximo de 300m.

Mientras que los sistemas utilizados contienen las mismas especificaciones, es necesario la normalización de los datos en el dominio del tiempo buscando una representación atemporal. Como solución, se almacenan y analizan todos los datos sobre los ciclos, siendo cada ciclo la representación atemporal de la información detectada y procesada por cada periodo de muestreo, es decir, el intervalo de tiempo en el que tarda el sistema radar en enviar un eco, recibirlo y procesarlo, obteniendo múltiples detecciones por ciclo.

Mientras que la representación es completamente atemporal para todas las detecciones de cada ciclo, los ciclos muestran una continuidad entre sí. Esto permite también analizar aquellos ciclos vacíos, entendiéndose por “ciclo vacío” aquel que siendo correctamente procesado no presenta ninguna detección.

Los radares mencionados conforman los siguientes escenarios:

- Un radar de largo alcance posicionado en la autovía, en el espacio intermedio de carriles a ambos sentidos. Este escenario permite detectar los vehículos que circulan en ambos sentidos: acercándose y alejándose al radar. Este escenario contiene detecciones no deseadas de tráfico transversal, proveniente de un puente que se encuentra en escena.
- Cuatro radares de corto alcance posicionados cada uno en las esquinas de una intersección de cuatro vías. Cada radar detecta al completo el tráfico que proviene de la vía que se encuentra en frente y parcialmente el proveniente de las laterales. El tráfico de la última vía, la que alinea su dirección con el ángulo de visión del radar, está

² Compañía con la sede principal en Brunswick, Alemania, centrada en la creación de radares para soluciones de control de tráfico y automoción.

completamente oculto para el radar hasta que el vehículo se encuentra dentro de la intersección. Este caso se comparte para todos los radares.

4.2. Escenarios de estudio

El estudio se centra en dos principales escenarios de control de tráfico: autovías e intersecciones. Para el tratamiento de los escenarios, se convirtieron las grabaciones a archivos JSON, almacenando la información del rango, la velocidad radial, el azimut y el índice del ciclo en el cual se obtuvo la detección. Estas son las principales mediciones para la identificación de trayectos, ya que todo trayecto muestra una linealidad en las tres gráficas de rango, velocidad y azimut sobre los ciclos.

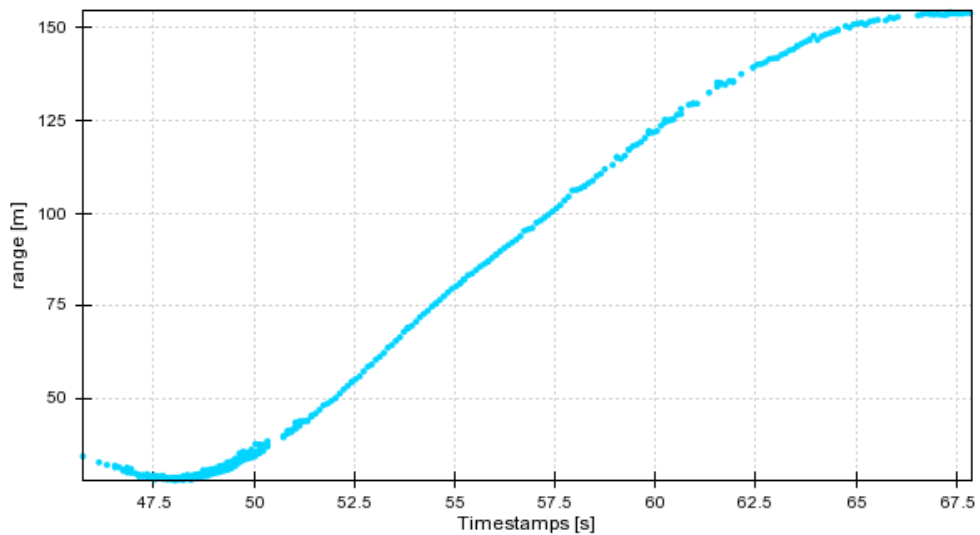


Figura 9: Gráfica Rango - Tiempo de un trayecto

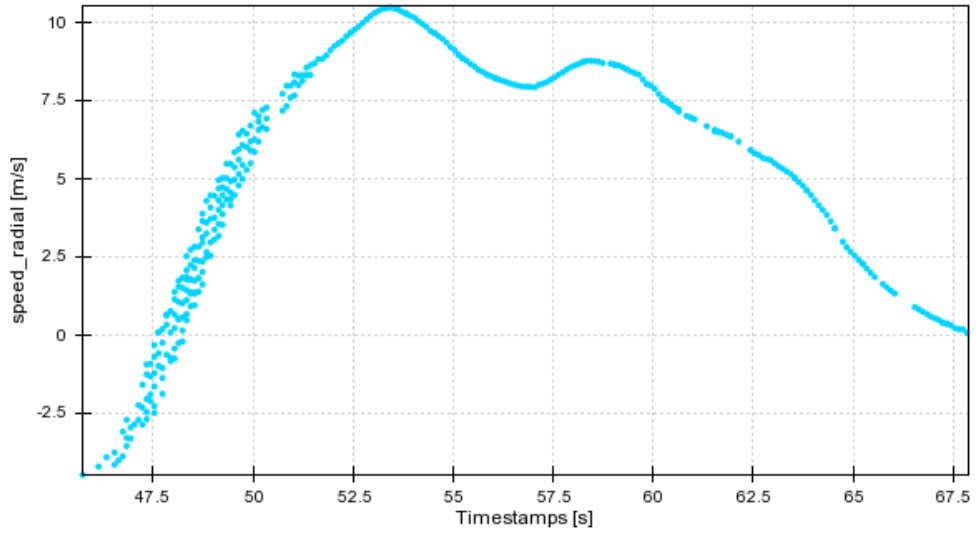


Figura 10: Gráfica Velocidad radial - Tiempo de un trayecto

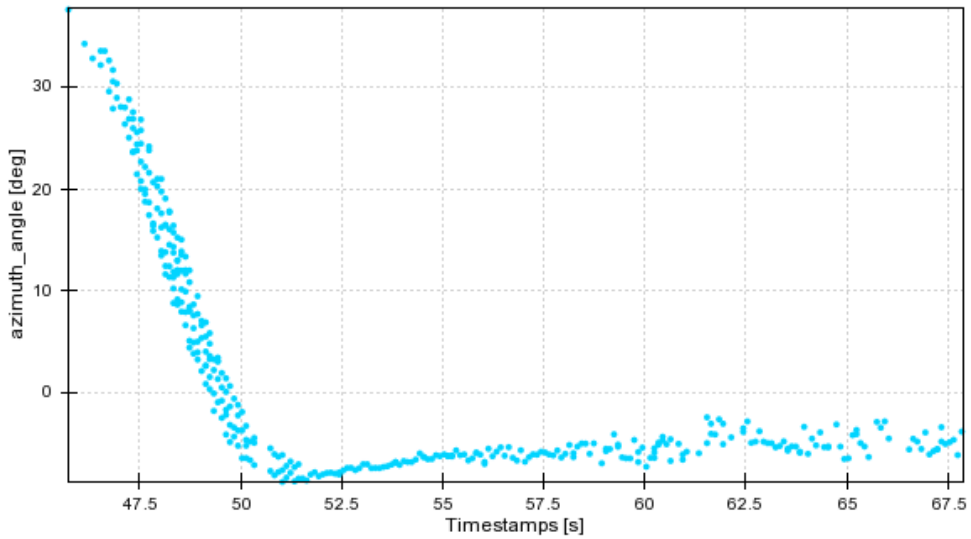


Figura 11: Gráfica Azimut - Tiempo de un trayecto

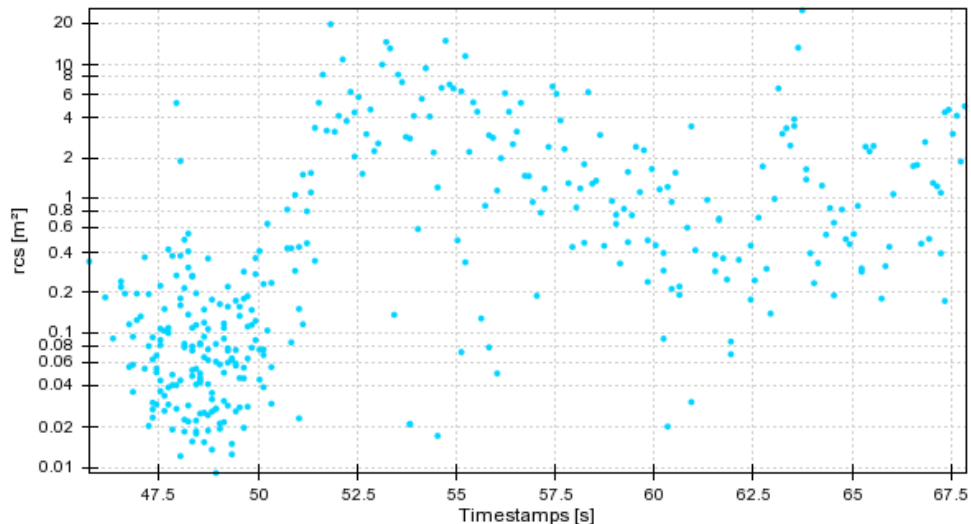


Figura 12: Gráfica RCS - Tiempo de un trayecto

Las Figura 9, Figura 10 y Figura 11 muestran la representación de las tres dimensiones sobre la medición de un trayecto identificado, pudiendo observarse la linealidad mencionada a lo largo del tiempo. De entre las tres, también se observa una mayor dispersión en varias detecciones del azimut, la cual es producida por las limitaciones del sistema radar.

Dentro de cada detección, existen no-linealidades aceptadas dentro de un rango en el que no se incluyen valores atípicos. Todos los trayectos han sido etiquetados manualmente, sin una rúbrica que explique las características deseadas para este rango de aceptación de los valores. Se observa como este rango de aceptación es generalmente mayor en las detecciones graficadas del azimut.

La Figura 12 muestra la gráfica del RCS en escala logarítmica, de la que se extrae la deducción de que no se puede extraer una relación entre las mediciones de forma intuitiva como en las otras dimensiones.

Dentro de cada grabación, existen distintos tipos de *clutter*, siendo uno de ellos los ecos de espejo. Las Figura 13, Figura 14 y Figura 15 muestran la representación de las gráficas de las tres dimensiones más importantes de un ruido de eco de espejo detectado. Estas gráficas son una clara representación de que todo trayecto que presente una fuerte no linealidad en todas sus detecciones, en al menos una de estas dimensiones, ha de ser clasificado como *clutter* y, por ende, filtrado por los modelos a diseñar.

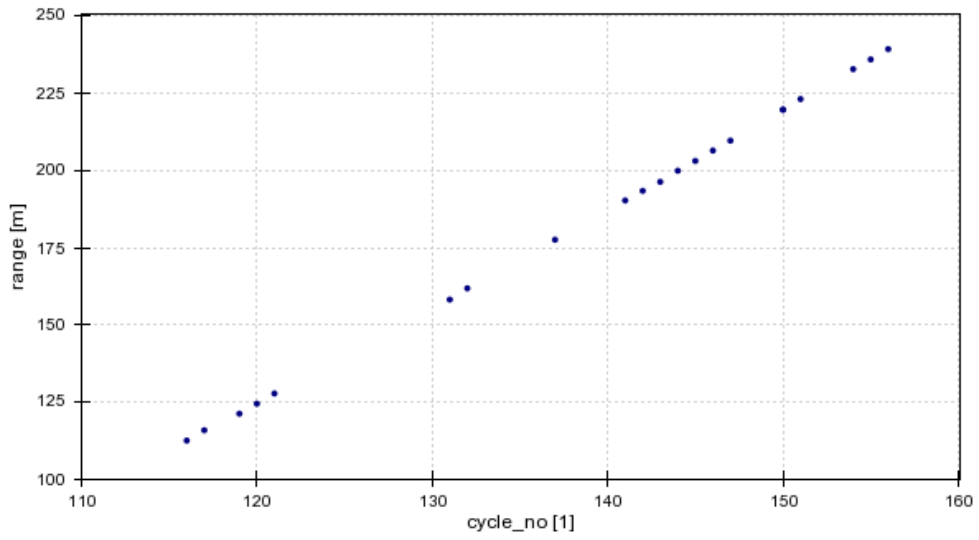


Figura 13: Gráfica Rango – Ciclos en *clutter* de tipo espejo.

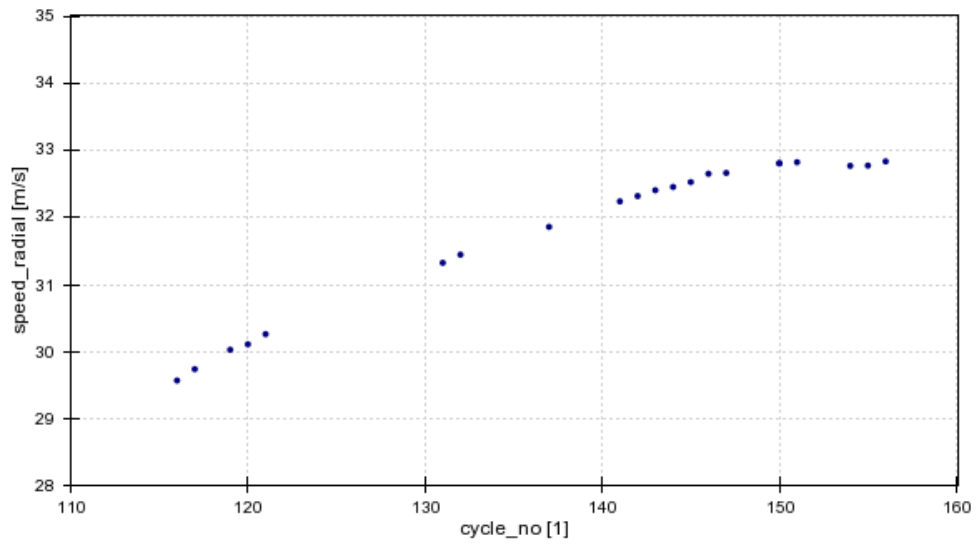


Figura 14: Gráfica Velocidad radial – Ciclos en *clutter* de tipo espejo.

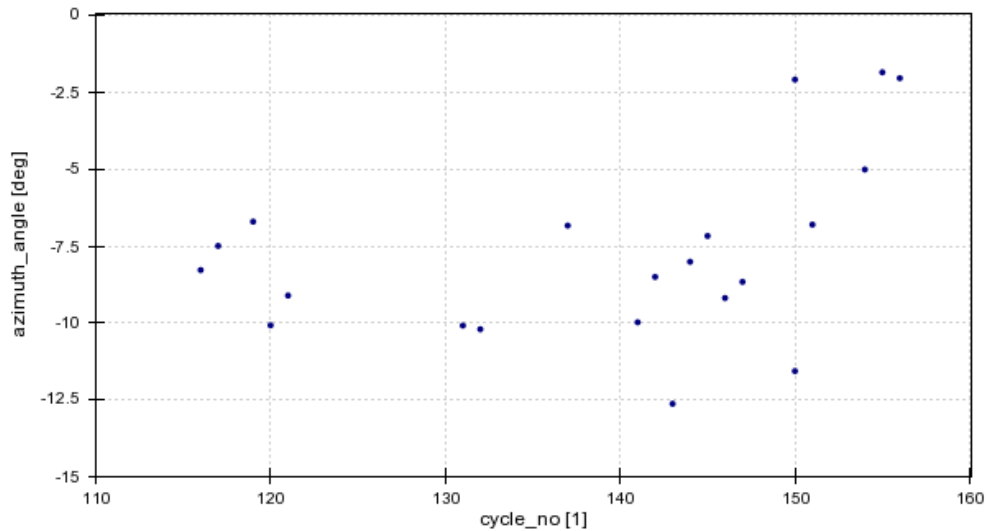


Figura 15: : Gráfica Azimut – Ciclos en *clutter* de tipo espejo.

En los escenarios explicados a continuación se representan por colores los trayectos etiquetados. Todo conjunto de detecciones del mismo color representa el mismo trayecto, facilitando la ayuda visual de la identificación de los trayectos y el *clutter*. Al existir un mayor número de trayectos que de divisiones existentes para la paleta de colores, los colores se repiten en múltiples trayectos.

4.2.1. Escenario: Autovía

El escenario de autovía a presentar se caracteriza por:

- Posicionamiento del radar cercano al centro entre los carriles, desplazado ligeramente.
- Visualización de objetos trasladándose a ambos sentidos en los carriles.
- Objetos viajando a velocidades entre el rango de 90 km/h y 120 km/h.
- No existe presencia de peatones o ciclistas.
- Claras trayectorias definidos por los trayectos entrantes (se acercan al radar) y los salientes (se alejan del radar).

La Tabla 1 representa los datos generales del escenario sobre el conteo de las detecciones, *clutter* y las trayectorias etiquetadas.

Escenario: Autovía	
Detecciones de radar	75592
Detecciones correspondientes a <i>clutter</i>	33446
Trayectorias etiquetadas	712
Ratio <i>clutter</i> /detecciones	44,245%

Tabla 1: Datos generales del escenario de autovía

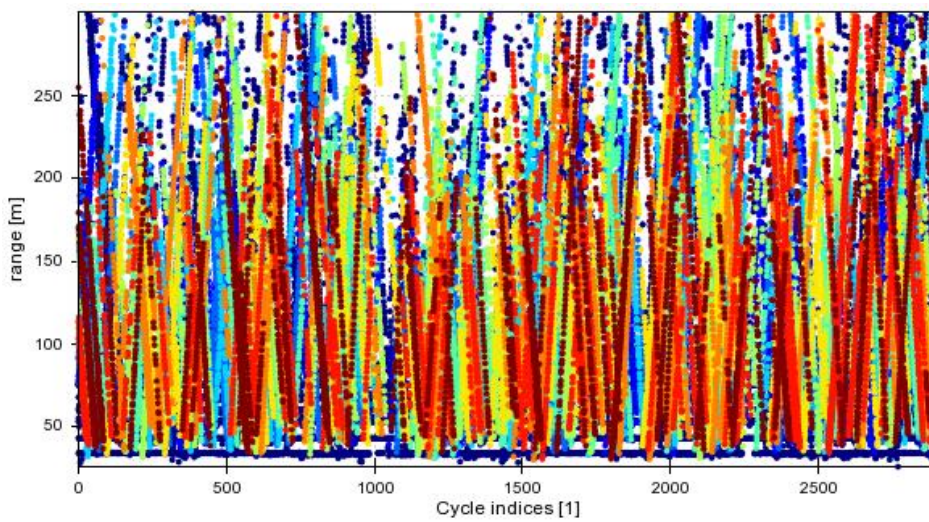


Figura 16: Escenario de autovía. Gráfica de Rango.

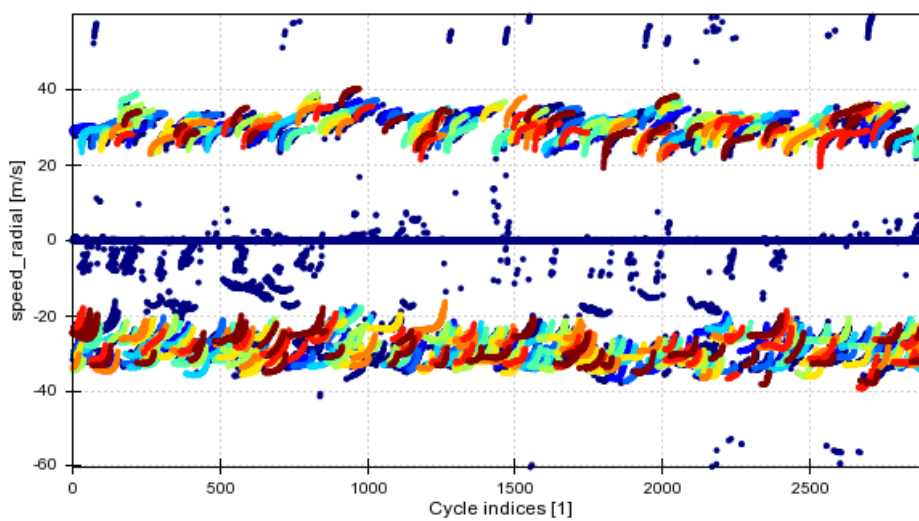


Figura 17: Escenario de autovía. Gráfica de Velocidad radial.

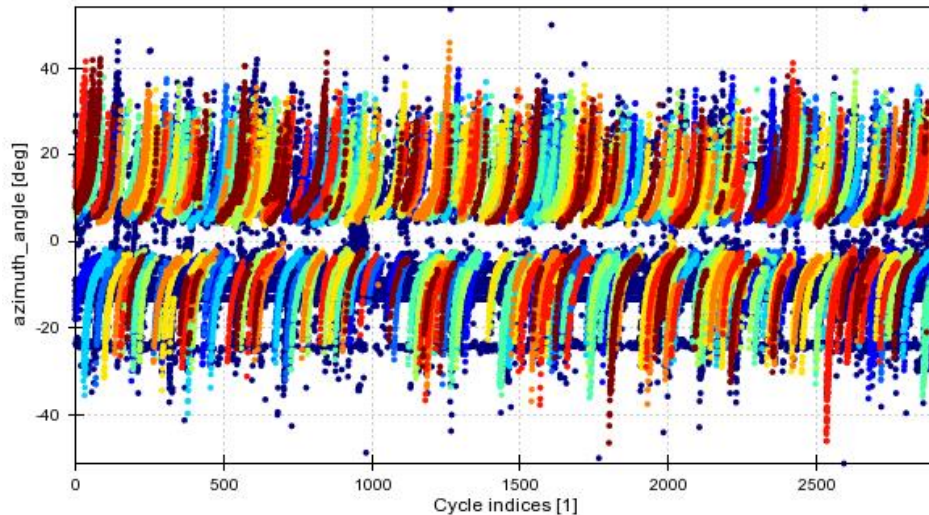


Figura 18: Escenario de autovía. Gráfica de Azimut.

Las Figura 16, Figura 17 y Figura 18 muestra las gráficas de rango, velocidad radial y azimut mostrando todos los trayectos y detecciones tipo *clutter*. Con un rápido análisis de estas se pueden identificar las siguientes relaciones:

- El área de detección de los trayectos es simétrico y no centrado en cero para las gráficas de velocidad radial y azimut.
- Existe una gran población de *clutter* con valores cercanos a 0 en la gráfica de velocidad radial.
- El *clutter* general muestra pendientes nulas en todas las gráficas.

En la Figura 19 se muestra la representación del RCS medido en el escenario. Al igual que comentado anteriormente, es difícil encontrar una relación entre las diferentes mediciones que conforman los trayectos, al contrario que en las gráficas anteriores.

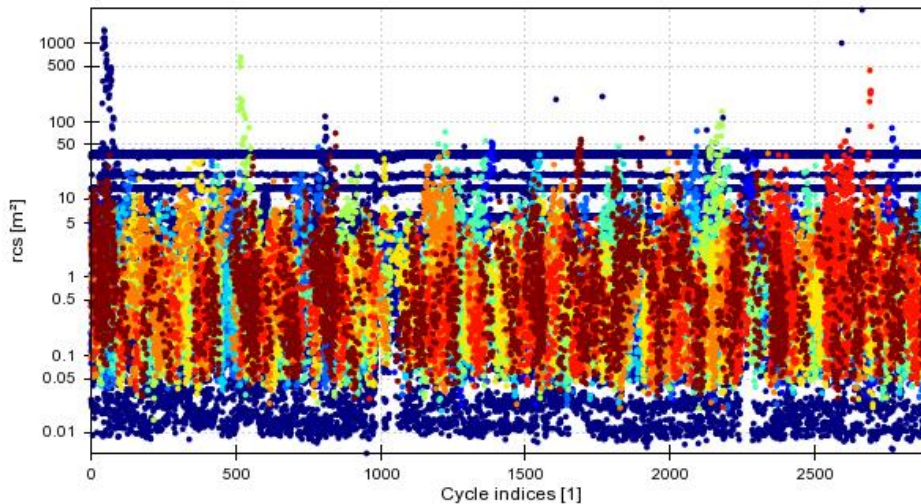


Figura 19: Escenario de autopista. Gráfica de RCS.

Debido a la caracterización del escenario, la identificación de los trayectos es relativamente sencilla. Las Figura 20, Figura 21 y Figura 22 muestran las gráficas de dos trayectos identificados, representando el trayecto entrante en marrón y el saliente en naranja. En esta ampliación, se observa claramente las características principales de todos los trayectos del escenario:

- Poseen una linealidad constante en las tres gráficas.
- Los trayectos muestran una relación simétrica con el eje de abscisas, en la gráfica de rango, y las diagonales mayores, en las gráficas de velocidad radial y azimut.
- La gráfica de rango muestra trayectorias rectas.
- Las gráficas de velocidad radial y azimut muestran trayectorias exponenciales.
- Las velocidades radiales absolutas de los trayectos se encuentran en el rango de 20 m/s a 30 m/s, mostrando dos franjas diferenciadas para tomar como correcto un trayecto.

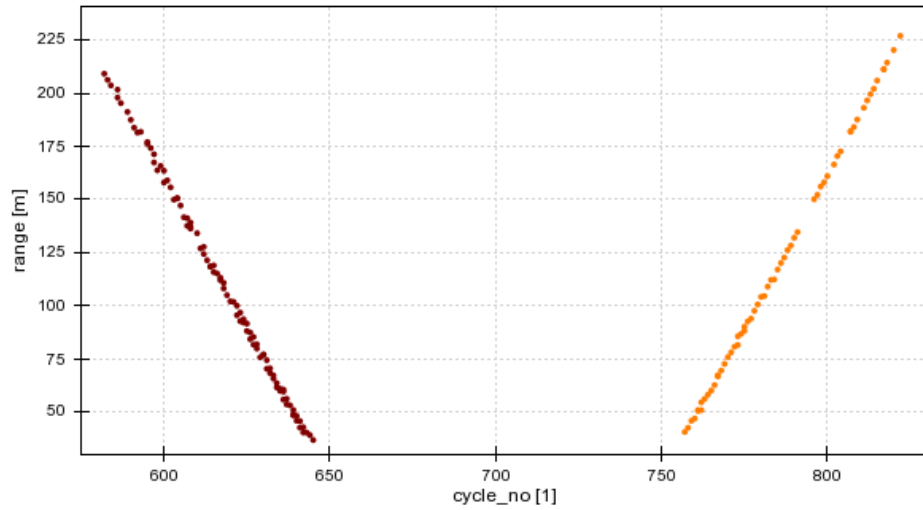


Figura 20: Escenario de autovía. Representación de trayectos en la gráfica Rango.

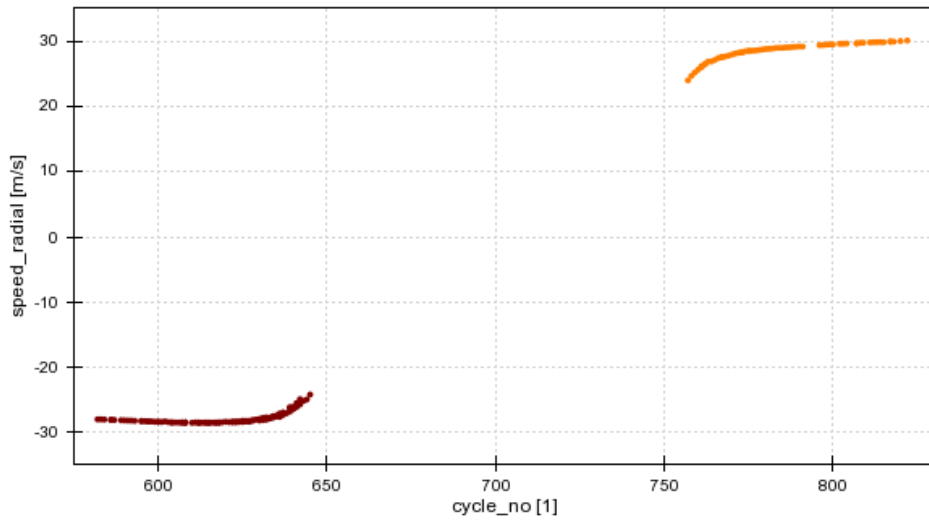


Figura 21: Escenario de autovía. Representación de trayectos en la gráfica Velocidad radial.

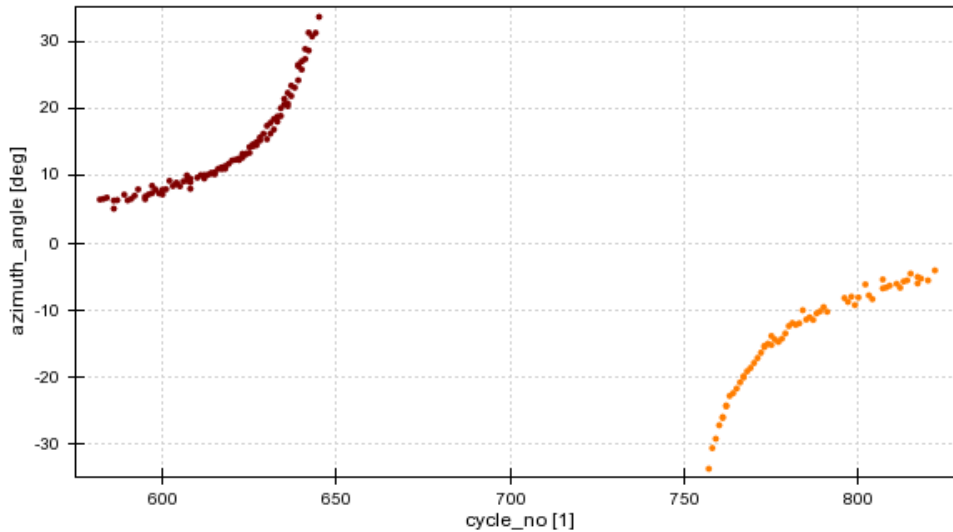


Figura 22: Escenario de autovía. Representación de trayectos en la gráfica Azimut.

Conociendo las características del escenario y la identificación de los trayectos, a continuación, se muestra la representación del *clutter* en el escenario, el cual equivale a un 44,245% de las detecciones. A continuación, en la Figura 23, se muestra la representación de la dispersión del *clutter* en el escenario sobre la gráfica de rango. En esta, se aprecia como existen dos claros tipos de objetos que definen el *clutter* del escenario:

- Objetos persistentes, los cuales son fáciles de filtrar pues sus velocidades son significativamente bajas a los objetos a detectar deseados.
- *Clutter* general, proviniendo la mayoría de los efectos de espejo.

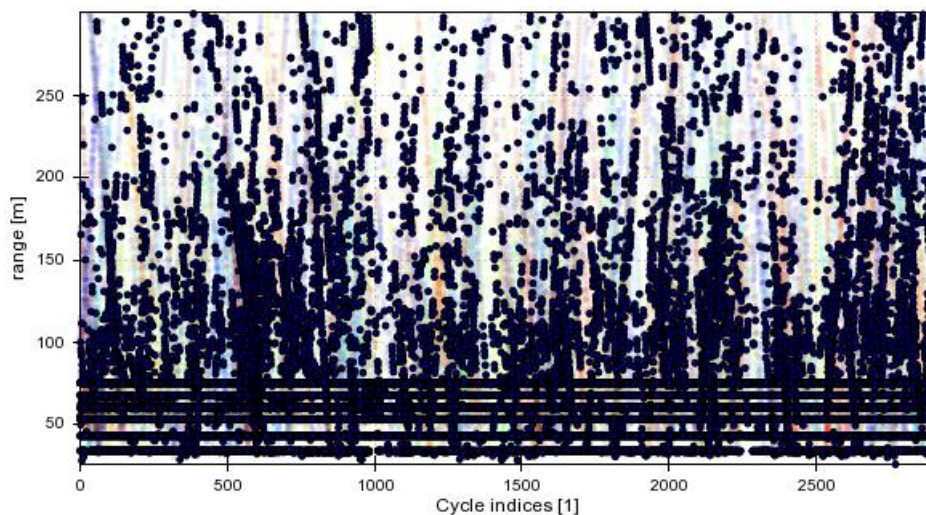


Figura 23: Escenario de autovía. Representación del *clutter*.

Así mismo, el escenario contiene un caso particular a tener en cuenta: el tráfico cruzado. Debido a la localización del radar, este detecta objetos correctos, pero no deseados correspondientes al tráfico de un puente cercano. Las Figura 24, Figura 25 y Figura 26 muestran las gráficas de rango, velocidad radial y azimut de uno de estos objetos detectados.

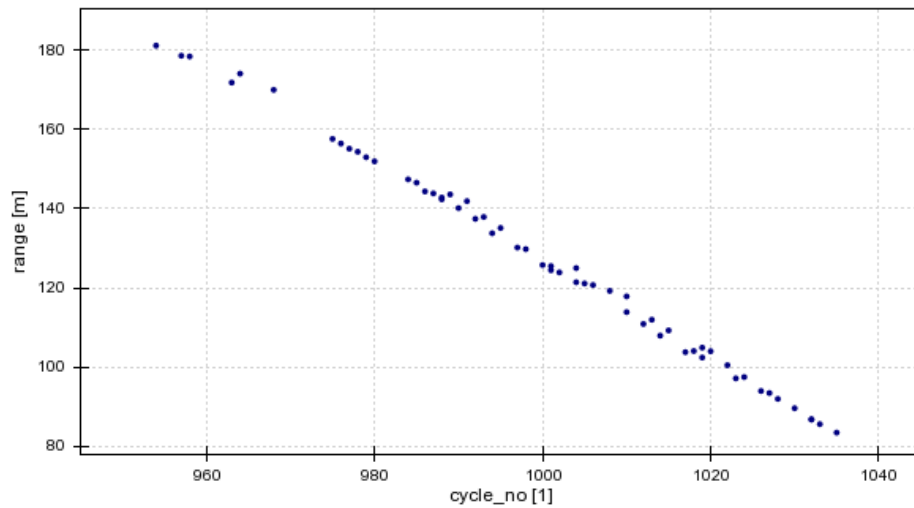


Figura 24: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Rango.

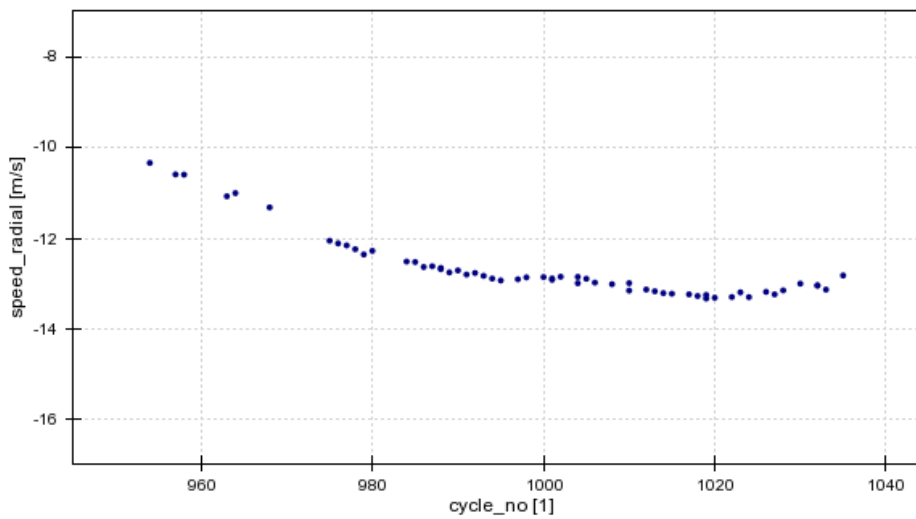


Figura 25: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Velocidad radial.

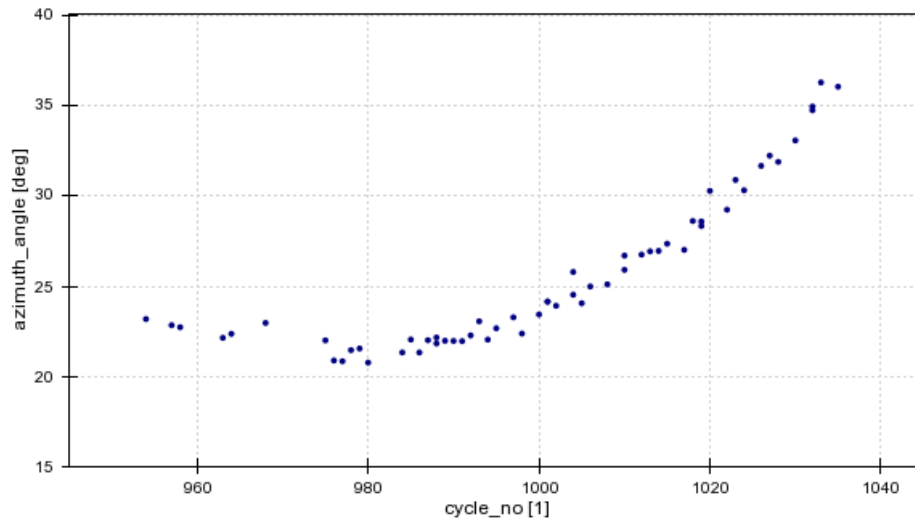


Figura 26: Escenario de autovía. Caso particular de tráfico cruzado en la gráfica de Azimut.

Los trayectos de estos objetos han sido clasificados como *clutter*, al ser elementos no deseados que detectar por el sistema en cuestión. Estos trayectos se caracterizan, principalmente, por una velocidad radial absoluta por lo general inferior a 20 m/s. Tanto las gráficas de rango como de azimut presentan características parecidas a los objetos deseados a detectar.

4.2.2. Escenario: Intersección

El escenario de intersección a presentar se caracteriza por:

- La existencia de cuatro radares enfocados en la misma escena, lo cual genera interferencias en las mediciones obtenidas por cada radar.
- Varias posibles trayectorias para los vehículos que se detecten desde el mismo punto inicial: continuar recto, girar hacia un lado o hacer un cambio de sentido.
- La detección de objetos transitando de un estado de reposo (velocidad nula) a un modelo cinemático con aceleración, al igual que el caso inverso.
- Las velocidades radiales se encuentran en el rango de ± 17 m/s.
- Existe la presencia de objetos no motorizados, los cuales presentan trayectos con una señal de velocidad radial muy ruidosa con un valor medio casi constante.
- Parte del ruido procede de objetos persistentes, los cuales son objetos estáticos en la escena que pueden ser detectados, ya sea porque introducen pequeñas variaciones en el movimiento (como la oscilación de las hojas de las plantas por el viento) o por

estar compuestos de un material muy reflectivo (como las señales de tráfico o los semáforos).

Escenario: Intersección	
Detecciones de radar	79383
Detecciones correspondientes a <i>clutter</i>	42885
Trayectorias etiquetadas	299
Ratio <i>clutter</i> /detecciones	54,023%

Tabla 2: Datos generales del escenario de intersección

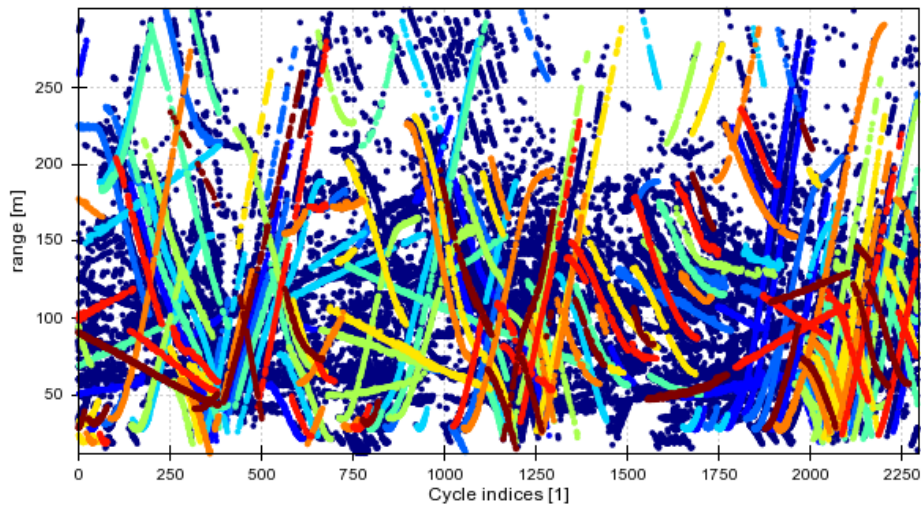


Figura 27: Escenario de intersección. Gráfica de Rango.³

³ Toda detección de color azul marino hace referencia a detecciones de *clutter*. Los 299 trayectos se reparten entre 9 colores. Por ello, múltiples trayectos no relacionados entre ellos comparten el mismo color.

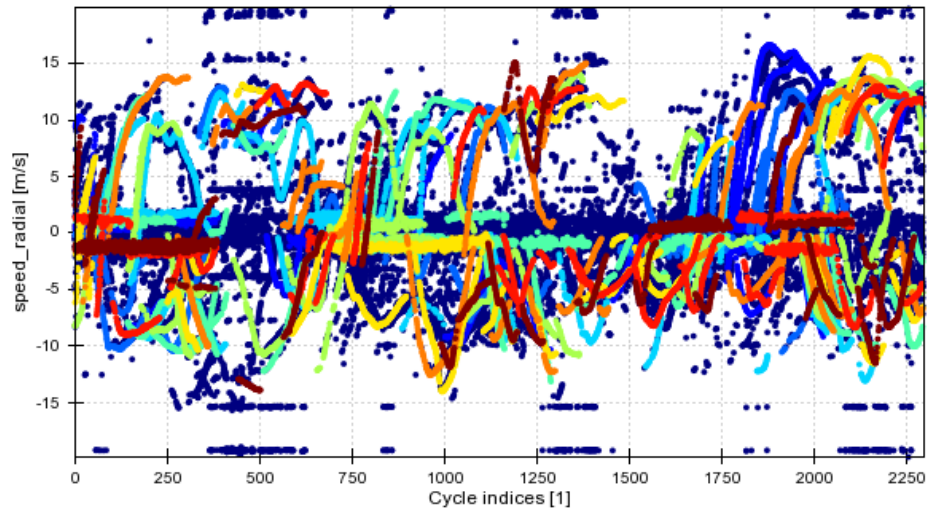


Figura 28: Escenario de intersección. Gráfica de Velocidad radial.

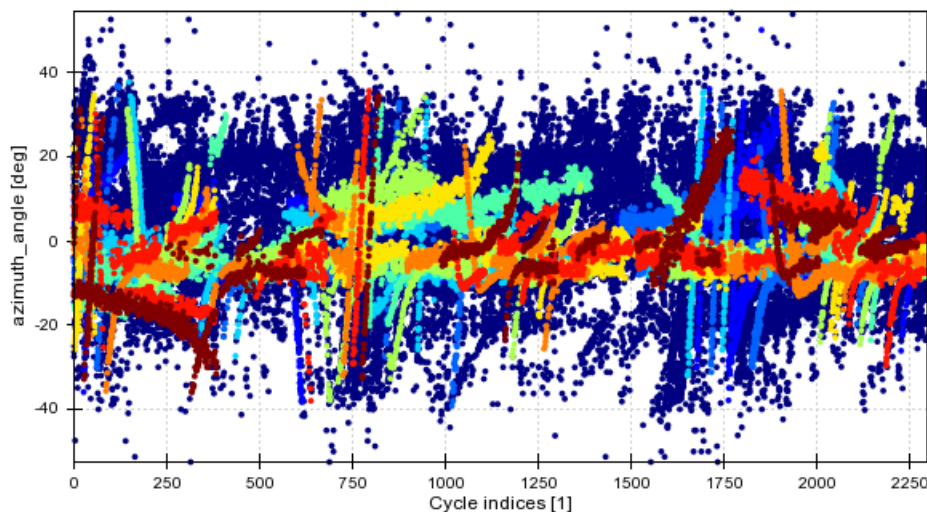


Figura 29: Escenario de intersección. Gráfica de Azimut.

En las Figura 27, Figura 28 y Figura 29 se muestran las gráficas de las tres dimensiones principales (rango, velocidad radial y azimut) del escenario, mostrando todos los trayectos etiquetados, así como las detecciones de *clutter*. En comparación al escenario de autovía, se aprecia como el total de trayectos no muestran una tendencia a seguir. Las detecciones del escenario se caracterizan por:

- Trayectos rectos y curvos que varían a distintas velocidades (Figura 27).
- Velocidades dependientes en función del objeto y la trayectoria (Figura 28).
- Ángulos de azimut que muestran múltiples direcciones, complicando la separación de los trayectos (Figura 29).

Debido a los datos de los trayectos, todos fueron etiquetados buscando las linealidades comentadas en Figura 9, Figura 10 y Figura 11. La Figura 30 muestra la representación del RCS del escenario, en la cual es difícil apreciar cómo los trayectos pueden diferenciarse entre sí.

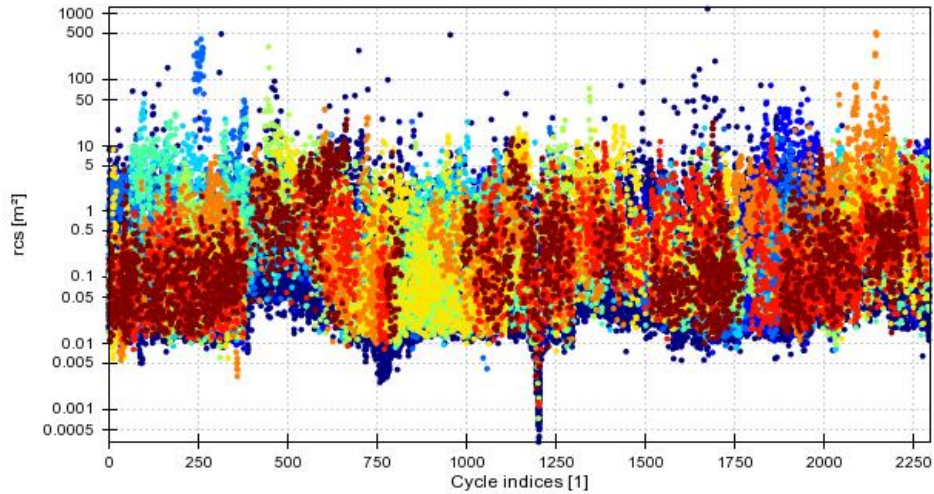


Figura 30: Escenario de intersección. Gráfica de RCS.

Debido a las características del entorno en la grabación del escenario, el ruido captado por el radar alcanza el 54%, como se aprecia en la Tabla 2.

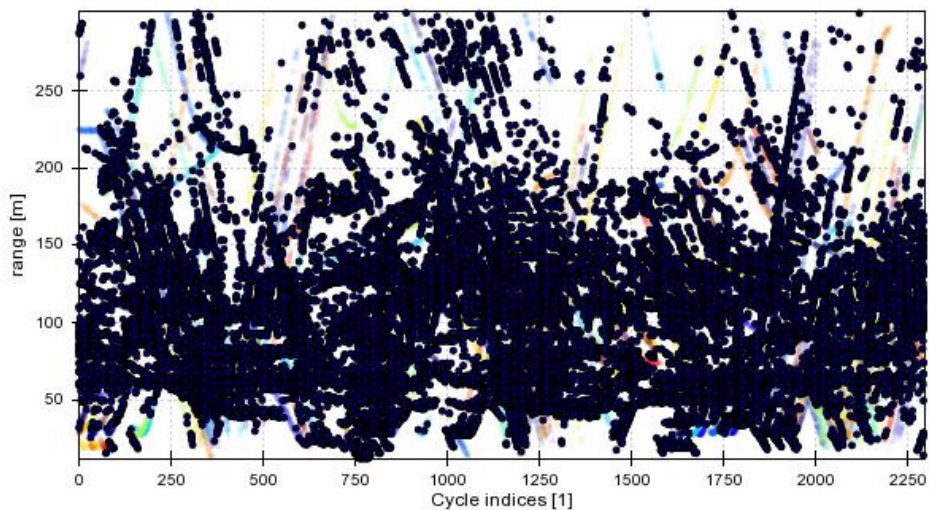


Figura 31: Escenario de intersección. Representación del clutter.

En la Figura 31 se muestra la representación del clutter detectado en la gráfica de rango, siendo esta en la que mejor se aprecia el impacto del clutter. De la misma, se aprecia como la gran mayoría de ruido proviene de fuentes cercanas y a medio rango (menor a 200 metros).

En este escenario, también se presenta un caso particular de trayecto: los peatones. La gráfica de los peatones (Figura 32, Figura 33, Figura 34) se caracteriza por:

- Una pendiente poco inclinada en la evolución del rango.
- Señales bastante ruidosas en la velocidad y el azimut.
- Una velocidad radial de modulo menor a 2 m/s.

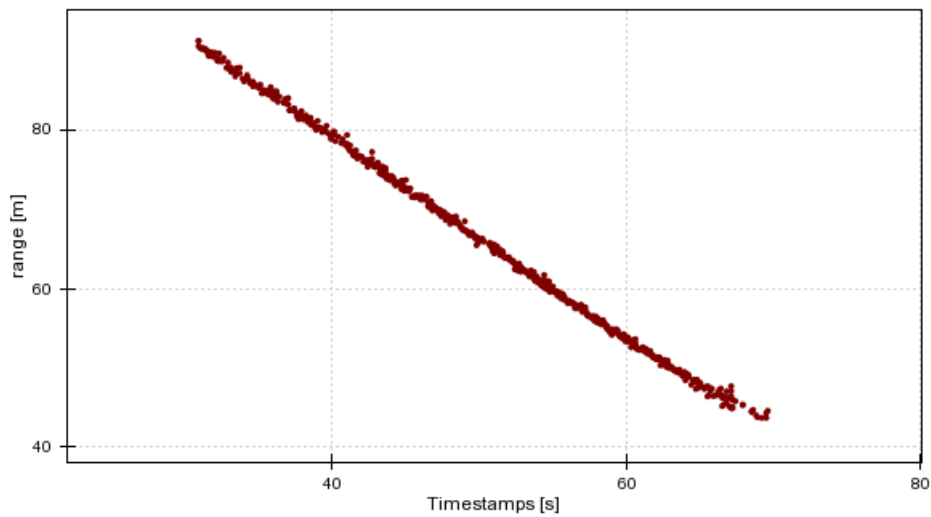


Figura 32: Escenario de intersección. Caso particular de peatones en la gráfica de Rango.

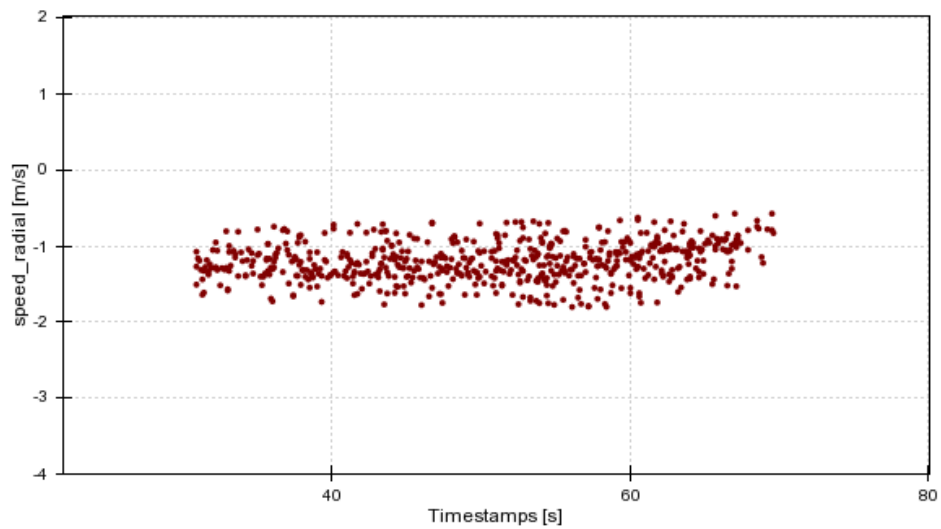


Figura 33: Escenario de intersección. Caso particular de peatones en la gráfica de Velocidad radial.

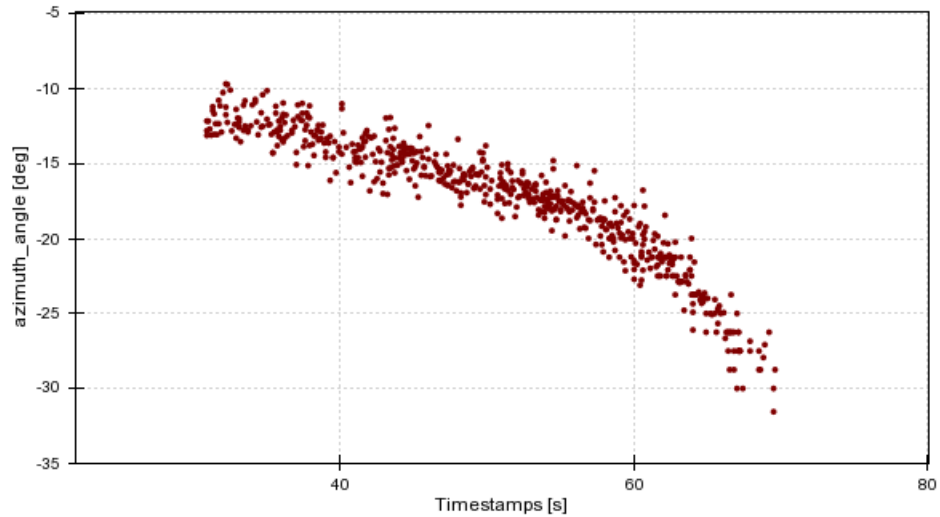


Figura 34: Escenario de intersección. Caso particular de peatones en la gráfica de Azimut.

Por último, el escenario también se caracteriza por un caso particular de *clutter*: interferencias de otros radares posicionados en el escenario. Las gráficas de este tipo de *clutter* (Figura 35, Figura 36, Figura 37) se caracterizan, principalmente, por una pendiente casi nula en las representaciones de las tres dimensiones principales. Este tipo de *clutter* es fácilmente apreciable en la Figura 28, repitiendo el patrón en varias franjas horizontales de tiempo y, dentro de la misma franja, a distintas velocidades constantes.

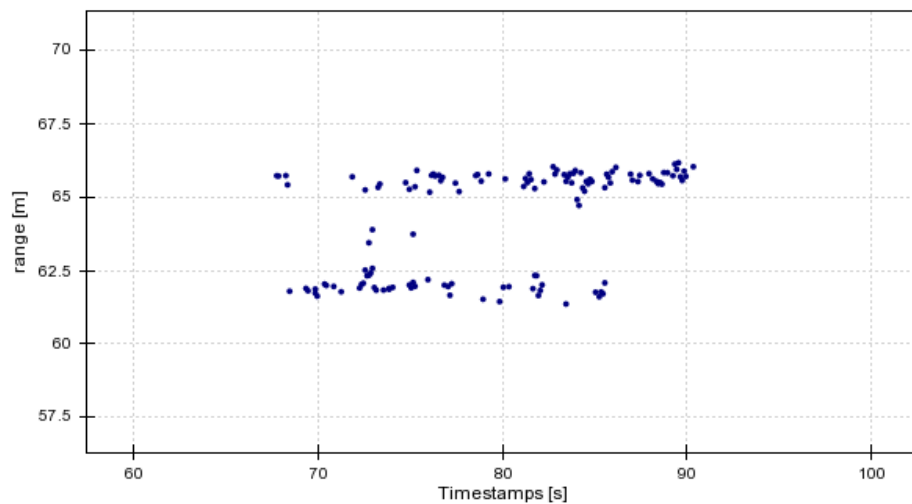


Figura 35: Escenario de intersección. Caso particular de interferencias en la gráfica de Rango.

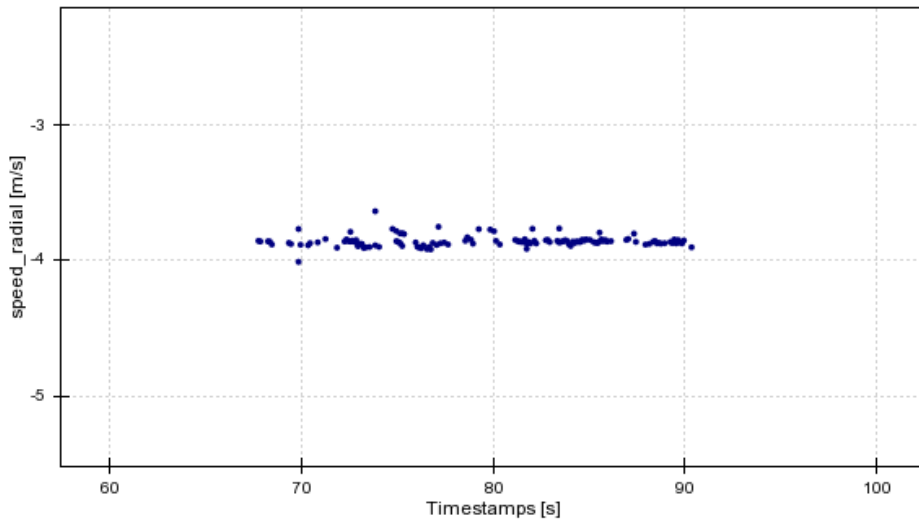


Figura 36: Escenario de intersección. Caso particular de interferencias en la gráfica de Velocidad radial.

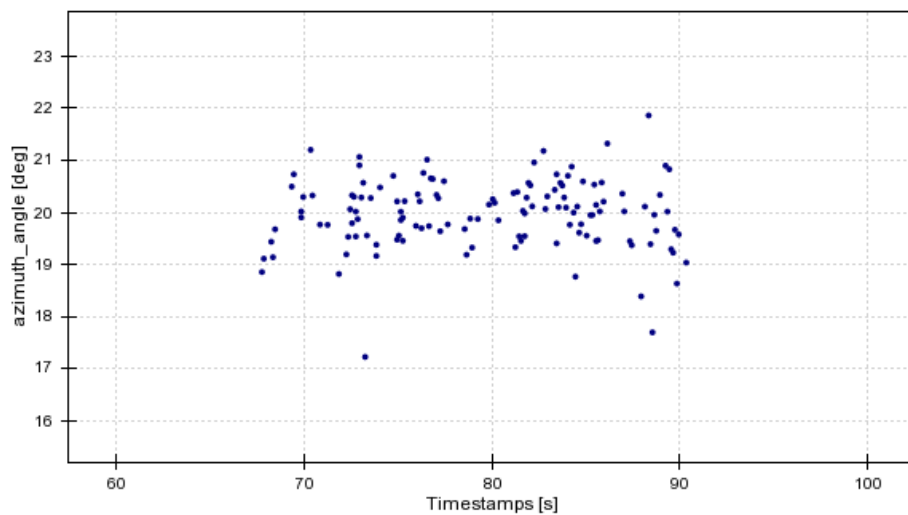


Figura 37: Escenario de intersección. Caso particular de interferencias en la gráfica de Azimut.

4.3. Caracterización del problema

4.3.1. Análisis de los datos

En las grabaciones utilizadas, cada detección se compone por el vector de dimensiones compuesto por:

- Índice del ciclo: Indica el ciclo en el que la detección fue computada. Cada grabación comienza siempre en el ciclo 0.

- **Rango:** Distancia que determina la recta entre la detección y el radar, definida en metros. Esta distancia siempre es positiva, ya que una distancia negativa indicaría que la detección se encuentra por detrás del radar.
- **Azimut:** Ángulo de llegada de la detección.
- **Velocidad radial:** Velocidad definida en metros por segundo a la que se detecta el objeto. Puede tomar valores positivos y negativos.
- **RCS:** Indica cuán detectable es un objeto, directamente relacionado con el tamaño y el material del objeto.
- **ID de trayecto:** Identificador numérico único que define todas las detecciones del mismo trayecto. Se encuentra referenciado en 0, utilizándose el valor “-1” para todas las detecciones de *clutter* u objetos no deseados.

Estas variables, junto con los valores numéricos de algunas detecciones a comienzo y final de cada grabación, se muestran en la Tabla 3 (autovía) y la Tabla 4 (intersección). En las tablas se muestra:

- La presencia del ciclo inicial vacío (0) en la Tabla 3, mostrando el ejemplo de la existencia de ciclos sin detecciones.
- Como la relación ciclos/detecciones se encuentra definida por el escenario, necesitándose un número mucho menor de ciclos en el escenario de intersección para la obtención de un número mayor de detecciones.

	Ciclo	Rango	Azimut	Velocidad radial	RCS	ID de trayecto
0	1	42,776466	0,307333	0,000559	0,417580	-1
1	1	52,931915	-0,230743	0,000113	13,873329	-1
2	1	56,562119	-0,289089	33,821888	0,150873	0
...
75589	2899	139,343185	-0,162135	27,804670	0,398758	613
75590	2899	144,797928	0,182263	-30,202490	1,182412	-1
75591	2899	182,933655	0,136401	-32,899307	0,315523	-1

Tabla 3: Escenario de autovía. Tabla de variables.

	Ciclo	Rango	Azimut	Velocidad radial	RCS	ID de trayecto
0	0	24,093849	0,385204	-6,174253	0,014541	0
1	0	28,991386	-0,011614	2,446499	0,131749	1
2	0	29,216211	0,050021	3,542815	0,022332	1
...
79380	247	229,386948	-0,129530	-1,809415	0,254688	-1
79381	247	237,225769	-0,098844	7,494000	0,502562	35
79382	247	263,295166	-0,094467	-5,170663	2,333734	34

Tabla 4: Escenario de intersección. Tabla de variables.

El análisis de las variables en ambos escenarios se muestra en las tablas a continuación. Generalizando el análisis de la Tabla 5 y la Tabla 6 se observan similitudes:

- Existen valores anómalos en el RCS, siendo valor del máximo mucho mayor que el del percentil 75.
- La velocidad radial en ambos escenarios es casi simétrica en 0.
- Los rangos de distancia y azimut son similares.

	Rango	Azimut	Velocidad radial	RCS	ID de trayecto
Conteo	75592	75592	75592	75592	75592
Media	104,627457	-0,002520	-1,398293	5,795334	155,446092
std	60,519550	0,250114	23,786809	21,148641	199,412983
min	25,457861	-0,894481	-60,252869	0,005457	-1,0
25%	60,190609	-0,181948	-25,813576	0,217497	-1,0
50%	79,217026	-0,098823	-0,000287	0,709101	31,0
75%	137,921871	0,210654	27,238566	3,731858	317,0
max	299,996674	0,938115	59,739830	2689,839600	616,0

Tabla 5: Escenario de autovía. Conteo, media y percentiles de los valores.

	Rango	Azimut	Velocidad radial	RCS	ID de trayecto
Conteo	79383	79383	79383	79383	79383
Media	104,752518	0,019098	-0,007982	0,750818	51,293098
std	55,134115	0,228785	6,007657	7,568353	77,316539
min	12,198697	-0,916298	-19,863453	0,000318	-1,0
25%	61,655106	-0,117792	-3,520847	0,048115	-1,0
50%	96,653061	-0,022281	-0,548920	0,124513	-1,0
75%	136,004784	0,170153	1,548885	0,386142	91,0
max	299,973083	0,943091	19,859989	1181,337158	261,0

Tabla 6: Escenario de intersección. Conteo, media y percentiles de los valores.

Las conclusiones del análisis de las variables muestran que, ciertamente, existen valores anómalos cuya existencia se deba a una o varias de las siguientes causas:

- Problemas referentes a los componentes del sistema del radar.
- Una incorrecta extracción de características de la matriz rango-doppler.
- Un caso anómalo puntual referente al escenario y no al sistema de medición.

La estrategia para el tratamiento de estos datos se explica, posteriormente, en la sección 5.1.

A continuación, se presentan las matrices de correlación para el escenario de autovía (Figura 38) y para el de intersección (Figura 39). En ambas figuras se representa únicamente la matriz triangular inferior, ignorando la diagonal principal, para así evitar mostrar información redundante.

Ninguno de los valores de las características supera el umbral, en valor absoluto, del 0.8. Esto indica que no existe fuerte correlación entre las variables y todas aportan información útil para el modelo final. Aun así, las mayores correlaciones existentes son:

- Identificador de trayecto y número de ciclo: A medida que avanza el tiempo (pasan los ciclos) de la grabación, tiene sentido que cada vez se identifique más objetos y sus trayectos
- Azimut y velocidad radial (autovía): Tomando como referencia la representación de estas variables en el tiempo (Figura 17, Figura 18, Figura 21, Figura 22) se puede

observar cómo ambas se ajustan a una ecuación exponencial, indicando así la relación que tienen.

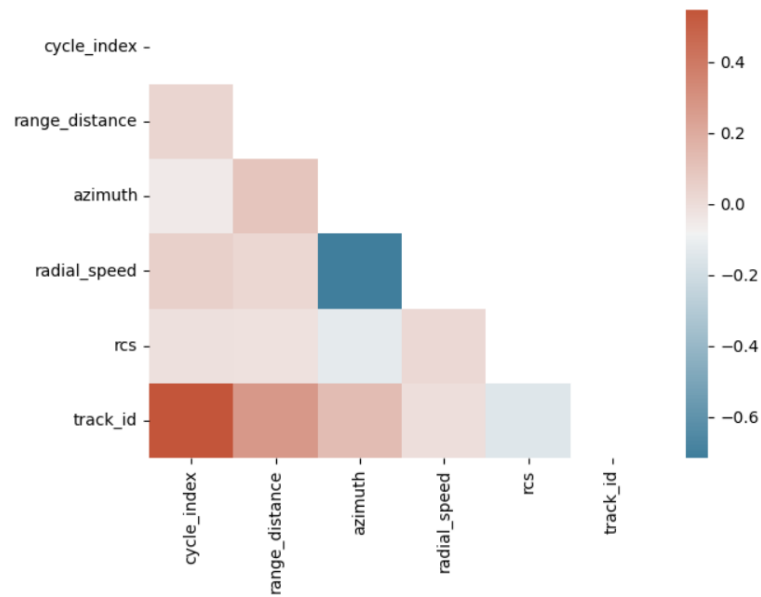


Figura 38: Escenario de autovía. Matriz triangular de correlación.

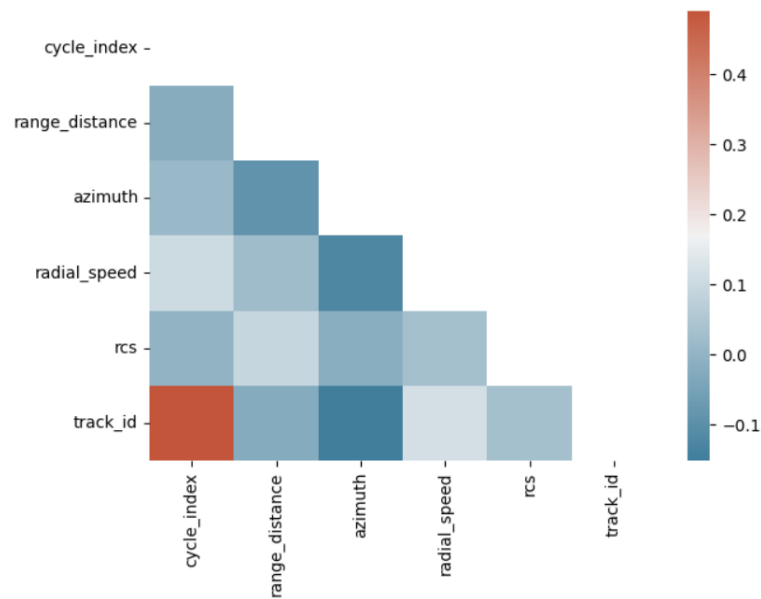


Figura 39: Escenario de intersección. Matriz triangular de correlación.

4.3.2. Decisión de la tarea a abordar

Tal como se ha presentado hasta ahora, el objetivo de esta investigación es definir un modelo que pueda ser utilizado para el filtrado de *clutter*, orientado para las mediciones de radares

de automoción. Para ello, es importante también definir a qué tipo de problema se enfrenta el estudio.

En función del enfoque de la variable objetivo, el mismo problema puede ser definido como un problema de:

- Regresión, en caso de buscar el porcentaje de similitud de la detección a evaluar para ser *clutter* o *target*.
- Clasificación, devolviendo la etiqueta para cada detección de si es *clutter* o *target*.

La etiqueta *target* hace referencia a una detección deseada, es decir, que es parte de un objeto deseado a detectar y mostrará un trayecto en su evolución en el tiempo. Los términos de *clutter* y *target* han sido extraídos del vocabulario comúnmente usado en el campo de procesamiento de la señal de radares.

Mientras que la caracterización del problema como uno de regresión puede ser válida, no solo para la identificación de la detección sino también para el uso matemático del valor estimado, en este estudio se decide plantearlo como un problema de clasificación.

De esta forma, se decide la utilización de la variable objetivo como una variable nueva "*is_clutter*", la cual toma el valor 1 para toda detección que sea *clutter* y 0 para toda aquella que forme parte de un trayecto. Esta decisión define claramente el problema como uno de clasificación binaria.

La principal razón por la que se toma esta caracterización es la necesidad de identificar los valores indeseados para eliminarlos y limpiar el número de detecciones por ciclo, en lugar de identificar cada trayecto de objetos en grabaciones que se desconoce el número máximo de objetos deseados. No obstante, al igual que se mencionó previamente, ambas soluciones se encuentran correlacionadas. Por ello, tras la limpieza de todas las detecciones de tipo *clutter* se obtendrán todos los trayectos en un estado previo a la segmentación de estos.

Por consecuencia, las métricas a usar definirán cómo de bien el modelo final será capaz de identificar el *clutter*, alineándose así con las cuantificaciones de los objetivos definidos en el apartado 3.2.

4.4. Definición de los modelos

Independientemente de las características únicas de cada escenario, desarrolladas en la sección 4.2, el desarrollo de un modelo inteligente para una aplicación de tráfico se define sobre unas líneas generales. En estas se especifican cuáles son los cuerpos que se desean detectar como objetos y las condiciones generales de funcionamiento. Por ejemplo, se define que un árbol no es un objeto deseado a detectar y que el escenario siempre se encontrará en un terreno no fluvial.

Estas líneas generales son las bases para las definiciones de los algoritmos, ya que definen también las especificaciones técnicas del material para poder utilizarse, siendo estas:

- Baja utilización de memoria.
- Actuación del sistema en tiempo real.
- Conocimiento de las matemáticas que utiliza el algoritmo.

Debido al uso de las variables, desarrolladas en el apartado 4.3.1, y de las especificaciones para la aplicación, a continuación, se enuncian los modelos escogidos para la definición del filtro. Todos ellos se encuentran disponibles a través del paquete de *scikit-learn*⁴.

4.4.1. Clasificador de Bayes

El clasificador de Bayes (Langley, Iba, & Thompson, 1992) es un algoritmo altamente utilizado para escenarios de probabilística condicional, cuyo objetivo es el de clasificar datos nuevos a clases ya predefinidas. Debido al bajo número de variables, se espera que el clasificador de Bayes sea el modelo más rápido de todos, siendo así un buen candidato para la aplicación.

El clasificador de Bayes puede, a su vez, ser un buen modelo para la aplicación de filtrado de *clutter* ya que puede manejar la probabilidad desconocida de los datos, esparcidos en rango, velocidad radial, azimut y RCS. El mismo, también asume que los datos se ajustan a una distribución de Gauss, habiéndose demostrado este comportamiento ya en los resultados proporcionados por Darzikolaei et al (Darzikolaei, Ebrahimzade, & Gholami, 2015). En otras palabras, se utilizará el Clasificador Gaussiano de Bayes (*GaussianNB*).

⁴ Librería altamente utilizada en Machine Learning: <https://scikit-learn.org/>

La mayor desventaja del clasificador es que asume que cada variable de entrada no está relacionada con la presencia o ausencia de cualquier otra variable de entrada, hipótesis que se ha desmentido en la sección 4.2. Esta característica del modelo puede producir resultados sub-óptimos en la clasificación de los datos.

El clasificador también tiene la desventaja de necesitar conocer la probabilidad previa de cada clase. Esta característica implica que el modelo final será aceptable únicamente para el sistema radar y escenario y localización específicos para el que fue entrenado. Puesto que la probabilidad viene dada por varias características tecnológicas y ambientales, las cuales resultan en una mayor o menor creación de *clutter* en los datos detectados, este inconveniente impide que el modelo pueda ser utilizado en otros escenarios de forma genérica.

4.4.2. Random Forest

El algoritmo de Random Forest (Ho, 1995) es conocido por combinar versatilidad y potencia de enfoque, ensamblando múltiples árboles de decisiones utilizando los métodos de *bagging* o *boosting* y reduciendo así la varianza.

Random Forest es un algoritmo que puede resultar de alta utilidad para un filtro de *clutter*, ya que una de sus características principales es que no es susceptible al ruido. Como se puede observar por la presentación de los escenarios en la sección 4.2, los entornos de automoción presentan mediciones bastante ruidosas.

La mayor desventaja a priori de la utilización del modelo de Random Forest es que es el único modelo propuesto de caja negra, impidiendo conocer en exactitud la matemática utilizada para la predicción para nuevas detecciones. En este modelo, la clasificación se determina por el conteo de los votos obtenidos en los estimadores.

4.4.3. Máquinas Vector de Soporte (SVM)

Las Máquinas Vector de Soporte o SVM (Cortes & Vapnik, 1995) es un algoritmo efectivo para cálculos con datos de múltiples dimensiones, pudiendo ajustarse bien a los datos que muestran relaciones lineales. Esto lo hace un algoritmo versátil para el objetivo del trabajo, que se los escenarios genéricos de grabaciones de radar se componen de cuatro variables dimensionales principales sin contar el tiempo.

Se espera que este modelo presente buenos resultados en el escenario de autovía, ya que como se ve en sus gráficas generales (Figura 17, Figura 18, Figura 19) la mayor concentración de *clutter* puede ser fácilmente identificable en una franja intermedia, acotada en las dos dimensiones de velocidad radial y azimut. No obstante, se espera también una eficiencia menor en el escenario de intersección, puesto que, como se muestra en sus gráficas (Figura 27, Figura 28, Figura 29), no se aprecia ningún tipo de franja en la que el *clutter* se conglomere. El modelo a utilizar es el *C-Support Vector Classifier (SVC)*.

La mayor desventaja de un modelo de SVM es que tiene un alto coste computacional, pudiendo resultar en un modelo válido como filtro, pero incompatible con la capacidad computacional disponible que un sistema radar básico pueda ofrecer para la solución de un filtro integrado.

4.4.4. Clasificador de K-Vecinos

El clasificador de K-vecinos (Fix & Hodges, 1951) es un algoritmo flexible e intuitivo, permitiendo que un modelo a partir de este algoritmo sea fácil de implementar y de gran valor a la hora de estimar los valores para los nuevos datos. Se basa en el modelo estadístico de los k-vecinos cercanos, obteniendo como un algoritmo supervisado para clasificación y regresión, pero no para clústering.

La clasificación de una nueva observación, utilizando el algoritmo de K-vecinos, se decide por el conteo de las etiquetas de los vecinos de la observación, siendo el factor 'K' el número de vecinos que afectan a la votación. Esto destapa la desventaja de que el algoritmo necesita retener el conjunto de datos utilizado para el entrenamiento, pudiendo resultar en un modelo que ocupe demasiada memoria si se utiliza un conjunto demasiado grande.

No obstante, se estima que el clasificador es una buena referencia debido a las relaciones lineales, representadas en todas las gráficas en la sección 4.2, que muestran los trayectos de los objetos deseados en sus tres dimensiones de rango, velocidad radial y azimut. Mientras que la regla se aplica para la detección de trayectos, será necesario ajustar K correctamente para que una observación pueda ser clasificada también como *clutter*.

4.5. Estrategia para el entrenamiento

En la sección 4.4 se definen todos los modelos a utilizar para la generación de los filtros de *clutter*. Aunque ninguno de los modelos a utilizar sea excesivamente complejo de entender, como puede suceder en el caso de las redes neuronales profundas, es necesario definir una buena estrategia para el entrenamiento de estos. De esta manera, se pueden alcanzar los mejores resultados en un tiempo menor y descartar el ajuste innecesario de algunos parámetros.

4.5.1. Vector de entrada de los modelos

Puesto que el proyecto principal del estudio es la comparación de varios modelos para la misma solución, uno de los puntos de diseño de los modelos es que han de compartir el mismo vector de entrada.

Todas las variables, las cuales pueden ser utilizadas para la entrada de los modelos, que son comunes para ambos escenarios son el número de ciclo, rango, azimut, velocidad radial y RCS. No obstante, el número de ciclo no aporta una mayor información al modelo, ya que:

- Ninguno de los algoritmos enunciados en la sección 4.4 utiliza datos recurrentes para la estimación.
- Esta variable puede aportar la información extra del número de detecciones por ciclo, la cual no es una constante en el tiempo ni afecta directamente sobre cada detección para su estimación como *clutter* o trayecto.

Por todo ello, se define el vector de entrada para cada modelo como:

$$input = [rango \quad azimut \quad v_{radial} \quad RCS]$$

4.5.2. Ajuste de los hiperparámetros

Todos los modelos enunciados para la creación del filtro contienen distintos hiperparámetros, los cuales es de crucial importancia entender para poder ajustarlos a los escenarios de automoción.

El modelo de **GaussianNB** contiene únicamente un hiperparámetro:

- *'priors'*: Identifica las probabilidades a priori de cada clase, definido como un array en el que la i -ésima posición indica la probabilidad de la clase ' i '.

Para identificar las probabilidades se utilizan los datos de la Tabla 1 y la Tabla 2, en las que se muestra la ratio *clutter*/detección. Al ser este un problema de clasificación binaria, la probabilidad de la clase *target* queda definida como la unidad menos la probabilidad de la clase *clutter*. A partir de estos datos, también se puede obtener la probabilidad de ambas clases realizando una media de las ratios *clutter*/detección. Esto determina que la probabilidad de *clutter* de manera genérica es del 49,134%

Como estrategia genérica, se realizarán dos entrenamientos en cada escenario: uno con las probabilidades específicas en cada escenario y otro con las probabilidades genéricas, obtenidas analizando ambos escenarios en conjunto. Se estima que este segundo entrenamiento funcione mejor ya que, aunque exista una diferencia en cada escenario, es probable que en el set de datos utilizado existan detecciones etiquetadas de manera incorrecta. El uso de las probabilidades genéricas disminuirá el peso que estos fallos en el etiquetado puedan tener.

El modelo de **Random Forest** contiene dos parámetros principales que se ajustarán:

- *n_estimators*: que define el número de árboles estimadores que se crean.
- *criterion*: que define el criterio matemático entre '*gini*', '*log_loss*' y '*entropy*' para la medición de la calidad del split en cada estimador.

Aunque el modelo contenga otros hiperparámetros, debido a que las características del vector de entrada son solo cuatro se estima que no necesitan mayor ajuste. Con ello, la estrategia para el ajuste de este modelo es realizar su entrenamiento para diferentes números de estimadores, tomando valores entre 4 (longitud del vector de entrada) y 50 (la mitad del valor por defecto), cada uno entrenado y evaluado con cada criterio posible.

Para **C-Support Vector Classification** los hiperparámetros a ajustar son:

- C : Parámetro de regularización, estrictamente positivo, que define el *soft margin* en la función de coste. Un valor mayor se traduce en un margen menor para la aceptación de errores, por lo que un valor muy alto puede producir un overfitting en el modelo.

- *kernel*: Especifica el tipo de kernel usado. De los disponibles, solo se usarán '*poly*', y '*rbf*'. Se descartan los kernels lineales y el sigmoid debido a que no se adaptan a los datos analizados.
- *gamma*: Coeficiente para el kernel utilizado y fijándose con '*scale*' y con '*auto*'.
- *degree*: Coeficiente del polinomio para el kernel '*poly*'. Los coeficientes a utilizar serán 3, 5 y 7.

El resto de hiperparámetros se mantendrán en los valores por defecto que la propia librería de *scikit-learn* define. De manera similar a la estrategia de *Random Forest*, para cada kernel se ajustarán distintos los distintos valores de '*gamma*', ajustando también distintos valores de '*degree*' para el kernel radial.

Por último, para el modelo de ***KNeighborsClassifier*** se pueden encontrar los hiperparámetros:

- *n_neighbors*: Número de vecinos utilizados para el voto. Se definirán varios vecinos con valores entre 3 y 30, buscando evaluar el rendimiento utilizando valores por debajo y por encima del definido por defecto y con sentido a lo anterior explicado en el trabajo.
- *weights*: La función utilizada para los pesos, pudiendo escogerse entre '*uniform*', para evaluar todos los puntos con los mismos pesos, y '*distance*', ofreciendo mayores pesos a los puntos más cercanos.
- *algorithm*: Siendo este el algoritmo para computar cuáles son los vecinos de cada observación, siendo este uno entre '*ball_tree*', '*kd_tree*' y '*brute*'. Según la propia documentación del modelo, el valor por defecto ajusta este parámetro al mejor algoritmo.

Este modelo tendrá dos fases de entrenamiento para el ajuste, comprobando en la primera fase cuál es el mejor modelo para los parámetros de pesos y algoritmos, utilizando a 5 vecinos. En esta primera fase se desea verificar el ajuste óptimo para los pesos, ya que existen situaciones en las que los vecinos más cercanos a una detección no tienen por qué ser los más relevantes, al igual que encontrar el algoritmo óptimo para el escenario.

Se toma la decisión de esta primera fase de entrenamiento, modificando el parámetro del algoritmo, para poder observar también el impacto de cada uno y si existe una combinación entre algoritmo y pesos que pueda ser mejor y hubiera permanecido oculta de otra manera.

En la segunda fase de entrenamiento se evaluarán cuántos vecinos son necesarios para la obtención de los mejores resultados.

4.5.3. Selección y aceptación del filtro final

Dentro de los resultados de todos los filtros entrenados con un mismo modelo, en un mismo escenario, se requieren reglas para la selección del modelo. Por ello, se descartará todo modelos que no cumpla los siguientes requerimientos:

- Superación de los objetivos específicos definidos: precisión de al menos 85% para autovía y 70% para intersección, para las predicciones sobre los datos de validación.
- Un intervalo de diferencia entre los resultados de entrenamiento y validación no superior al 7%. En el caso de que un modelo tenga overfitting, se estima que este intervalo será mayor.

Tras el descarte, el mejor filtro entre los resultados de un mismo modelo será seleccionado por los mejores resultados de precisión y F1-score sobre los datos de validación.

4.6. Métricas de la comparativa

Una vez entrenados los modelos, las métricas de comparación servirán para decidir cuál de los modelos es mejor en función de los requerimientos finales de la aplicación. Estas métricas se utilizan únicamente para la comparación de los filtros creados con distintos modelos dentro de un mismo escenario, nunca para filtros de distintos escenarios, ya se hayan entrenado con el mismo o distintos modelos base.

Las métricas aplican principalmente a todo modelo que no haya sido descartado, según los requerimientos enunciados en el apartado 4.5.3. Por ello, se definen las siguientes métricas para la comparativa de los modelos:

- **Precisión (*accuracy*) y F1-score:** Estos resultados indican lo eficiente que es un modelo para etiquetar correctamente una observación nueva. Utilizando ambos, esta métrica

se representa como una gráfica con los resultados constituyendo los ejes de abscisas y ordenadas.

- **Memoria necesaria para el modelo:** Tal como se ha descrito previamente, los filtros desarrollados en el trabajo se han diseñado con el objetivo de ser implementados en tecnología al alcance de la sociedad hoy en día. Esta métrica puntúa como mejor a aquel filtro que requiera de un tamaño disponible de memoria menor para el almacenamiento de los pesos ya entrenados. La comparación de memoria se llevará a cabo comprobando el tamaño de los archivos creados, con ayuda de *pickle*, en los que se almacena el modelo.
- **Tiempo de procesamiento en las predicciones:** Otro de los requerimientos es que el modelo sea capaz de realizar predicciones en tiempo real. Por ello, esta métrica puntúa como mejor modelo a aquel que realice predicciones, sobre el mismo número de datos, en un menor tiempo.

5. Desarrollo de la comparativa

En este capítulo se desarrollan los pasos llevados a cabo, que no hayan sido explicados anteriormente, para la obtención de los modelos finales. Cada modelo presentado se muestra con los resultados, compuestos por las métricas de precisión, F1 y la matriz de confusión.

El capítulo se divide en las siguientes secciones:

- **Tratamiento de los datos:** Se explica el tratamiento de los datos llevado a cabo para la obtención de los conjuntos de entrenamiento y validación. También se enuncia la lógica tras la cual obtener la variable objetivo y la ratio utilizada para separar los datos en los respectivos conjuntos, siendo estas decisiones genéricas para todos los escenarios.
- **Entrenamiento y obtención de los modelos:** En estos, se enuncian los casos de entrenamiento seleccionados, organizados en tablas mostrando los parámetros utilizados para cada modelo. En las mismas, se muestra si el modelo es válido o se descarta, anunciando posteriormente cuál es el resultado elegido como modelo final y la memoria que ocupa el mismo. En este apartado no se muestra la comparativa de las métricas, enumeradas en la sección 4.6.
- **Testeo de los modelos en datos sin etiquetar:** En este apartado se lleva a cabo la evaluación de los modelos, no descartados, sobre grabaciones que no han sido previamente etiquetadas. Las grabaciones utilizadas en ambos escenarios hacen referencia a una grabación de una fecha diferente y con el mismo sistema radar, para el escenario de autovía, y una con la misma fecha pero con un sistema radar distinto, para el escenario de intersección.

5.1. Tratamiento de los datos

Como se muestra en los análisis, ilustrados en la y la Tabla 6, existen valores anómalos en el RCS. Para el tratamiento de estos, se elimina todo valor con un RCS superior a 4,5. Se toma este valor como cota superior teniendo en cuenta el valor más alto del percentil 75 entre ambos escenarios, ofreciendo también un margen extra de error.

Con esto, se establece como aceptable un falso máximo valor de 4,5 sabiendo que, cualquier valor cercano o superior a este, será definido como *clutter*.

Una vez se hayan preparado las características que compongan el vector de entrada, los datos serán normalizados con la estrategia min-max. Con esta estrategia, todos los datos se encontrarán normalizados con un valor positivo entre 0 y 1.

Posteriormente, es necesario modificar la variable que especifica el identificador de cada trayecto por otra que determine si la detección es o no del tipo *clutter*. Para la obtención del resultado simplemente se necesita la creación de una nueva columna, cuyo valor es '1' si el valor del identificador es '-1' y '0' para cualquier otro valor. De esta forma se obtiene la variable objetivo, con las etiquetas binarias, usándose la clase positiva para el *clutter*.

Por último, los subconjuntos de entrenamiento y validación se crean a partir de los datos generales (sin mezclar los escenarios entre sí). Estos datos son reordenados de manera aleatoria y separados en entrenamiento y validación con los porcentajes 80/20 respectivamente. Es seguro y recomendado realizar esta rutina para la preparación de los datos por dos motivos:

- La selección aleatoria de los datos para los conjuntos de entrenamiento y validación aumenta la precisión en las predicciones, para el modelo ya entrenado, debido a que no se aprende relaciones y generaliza mejor.
- Por la estructura de los modelos a utilizar, no se requiere ninguna recursividad para los modelos. Por ello, cada detección puede analizarse independientemente.

5.2. Entrenamiento y obtención de los modelos

En esta sección se muestran los resultados de todos los entrenamientos. Separando los escenarios en dos apartados, para cada modelo se muestra una tabla de resultados del entrenamiento, indicando la última columna si el modelo es válido según los requerimientos definidos. Posteriormente, se menciona cuál de todos los entrenamientos conforma el modelo seleccionado, indicando así los valores de su matriz de confusión y el espacio de memoria que ocupa el mismo.

5.2.1. Planificación de los entrenamientos

Previamente a la realización de los entrenamientos de los modelos, es necesario definir cómo se realizará para el ajuste de los hiperparámetros, la selección de los modelos y su posterior análisis.

Comenzando por los datos, estos son separados en una relación 80/20 para los datos de entrenamiento y test, respectivamente. Esta separación se muestra en la Tabla 7.

	Train (80%)		Test (20%)	
	Detecciones	<i>Clutter</i>	Detecciones	<i>Clutter</i>
Autovía	60473	26705	15119	6741
Intersección	63506	34340	15877	8545

Tabla 7: Datos de entrenamiento y test después de la separación.

La estrategia para la elección de los hiperparámetros depende de cada modelo, ya que cada uno tiene un número distinto de hiperparámetros.

- *GaussianNB*: No necesita una estrategia especial al tener solo un hiperparámetro y dos casos de estudio para cada escenario.
- *Random Forest*: Se define la estrategia de expandir cada estimador hasta que agote las posibilidades. Con ello, se necesitará una iteración para ajustar sus dos hiperparámetros y seleccionar el mejor modelo.
- *SVM*: Se ignora el kernel lineal, ya que los datos de los escenarios no se ajustan al mismo. Posteriormente, este modelo necesitará dos iteraciones para el ajuste de los hiperparámetros: la primera para analizar cuál kernel se ajusta mejor a los datos y con qué valor de gamma, y la segunda, cuáles hiperparámetros referentes del kernel seleccionado devuelven el mejor modelo. Se selecciona el mismo kernel para ambos escenarios.
- *K-Neighbors*: Al igual que con SVM, el modelo necesita de dos iteraciones. La primera busca encontrar la mejor combinación de pesos y algoritmo a usar. La segunda establece el número de vecinos necesarios para la obtención del mejor modelo.

Los escenarios que han sido planteados son también muy dispares entre ellos, presentando cada uno unas características propias que afectan en el entrenamiento y en la decisión de la selección de un modelo:

- Autovía: Este escenario presenta la gran mayoría de clutter en una banda central, al igual que todos los objetos muestran la misma evolución temporal en trayectoria y velocidad. Por ello, se aceptan modelos que lleguen al overfitting en este escenario, ya que el caso general siempre se asemejará al utilizado en el entrenamiento. Con ello, se toma como buenos a los modelos con un porcentaje de éxito de al menos del 85%
- Intersección: Es el escenario más caótico, en el que no se aprecia ninguna banda de separación entre *clutter* y *targets*. Se estima que los porcentajes en las métricas de este escenario serán inferiores a las obtenidas en el escenario de autovía. Entonces, todo modelo con un porcentaje igual o superior al 70% de éxito es aceptado.

Para los modelos válidos únicamente, se calcula la matriz de confusión. En esta, el parámetro a destacar es el de los falsos positivos (*targets* definidos como *clutter*), buscando que este sea el más bajo posible. En un análisis de un único instante de tiempo de las detecciones de radar, los *targets* muestran características que permiten agruparlos en una estimación de objetos, estado así las detecciones de *clutter* alejadas de la ROI, sin vecinos cercanos o con valores muy dispares.

Mientras que el caso de los filtros no detectando todas las detecciones de *clutter* no es tan problemático, el filtrado incorrecto de *targets* puede ocasionar varios problemas como:

- No llegar al umbral mínimo de *targets* vecinos para detectar un objeto. Esto puede suceder en objetos pequeños.
- La creación de no linealidades en el trayecto de un objeto.

Esta razón es la que marca como tan importante el número de falsos positivos en las matrices de confusión.

5.2.2. Escenario de autovía

Comenzando por el clasificador Bayesiano ***GaussianNB***, existen únicamente dos casos a estudiar en función de la probabilidad dada.

La probabilidad de *clutter* para ambos casos, de manera independiente, se definen como la dada por el escenario (44,245% para autovía) y como la probabilidad genérica de *clutter* (49,134%).

Prob. <i>Clutter</i>	Prob. Target	Accuracy		F1 Score		Modelo válido
		Train	Test	Train	Test	
44,245%	55,755%	83,87%	84,26%	80,77%	81,38%	No
49,134%	50,866%	82,84%	83,11%	79,98%	80,46%	No

Tabla 8: Resultados del clasificador Bayesiano para autovía.

Debido a los resultados representados en la Tabla 8, este modelo no supera la cota del 85% de acierto aunque se acerque. Por ello, este modelo se descarta como válido, con los datos actuales, para la creación de un filtro de *clutter*.

En el entrenamiento de ***Random Forest*** se utilizaron, sobre los criterios '*gini*', '*entropy*' y '*log_loss*', el siguiente número de estimadores: 4, 8, 10, 12, 20 y 50. Como se muestran en los resultados de la Tabla 9, el modelo es apto para la creación de un filtro con cualquier número de estimadores.

n_estimators	criterion	Accuracy		F1 Score		Modelo válido
		Train	Test	Train	Test	
4	gini	98,04%	93,21%	97,73%	91,98%	Sí
	entropy	98,03%	93,09%	97,72%	91,84%	Sí
	log_loss	98,10%	93,08%	97,81%	91,84%	Sí
8	gini	99,03%	93,49%	98,90%	92,35%	Sí
	entropy	99,00%	93,47%	98,86%	92,33%	Sí
	log_loss	98,97%	93,38%	98,82%	92,20%	Sí
10	gini	99,16%	93,52%	99,04%	92,39%	Sí
	entropy	99,13%	93,49%	99,01%	92,33%	Sí
	log_loss	99,24%	93,54%	99,13%	92,41%	Sí
12	gini	99,31%	93,47%	99,22%	92,31%	Sí
	entropy	99,38%	93,47%	99,29%	92,32%	Sí
	log_loss	99,35%	93,56%	99,26%	92,41%	Sí
20	gini	99,68%	93,56%	99,64%	92,43%	Sí
	entropy	99,70%	93,64%	99,66%	92,51%	Sí
	log_loss	99,69%	93,64%	99,64%	92,53%	Sí
50	gini	99,96%	93,74%	99,95%	92,65%	Sí
	entropy	99,97%	93,81%	99,96%	92,73%	Sí
	log_loss	99,97%	93,77%	99,96%	92,61%	Sí

Tabla 9: Resultados del clasificador Random Forest para autovía.

El modelo escogido es aquel que tiene mejores resultados en el conjunto de evaluación, siendo este el de 50 estimadores y el criterio de 'entropy'. Este modelo, aunque tenga un sobreajuste muy alto, se considera como válido por el hecho de que todo tipo de objeto que intervenga en el escenario tendrá siempre unas características similares:

- La trayectoria realizada y el azimut no presentan un amplio margen para variaciones.
- La evolución temporal de la velocidad sigue un claro patrón en todos los objetos.

El archivo que almacena el modelo ocupa un total de 21,2 MB de memoria. La Tabla 10 muestra los valores de la matriz de confusión del modelo.

	<i>Target</i> estimado	<i>Clutter</i> estimado
<i>Target</i> real	8216	162
<i>Clutter</i> real	791	5950

Tabla 10: Matriz de confusión del modelo de Random Forest seleccionado para autovía.

La Tabla 10 muestra los valores de la matriz de confusión del modelo, en el que se muestra un bajo número de falsos positivos. Este valor es el 1,9 % sobre el número total de *targets*, siendo así este un buen modelo.

Respecto al entrenamiento con **SVM**, puesto que es el modelo de mayor computación y un entrenamiento más longevo, se realizó una primera iteración para la comprobación de los kernels '*rbf*' y '*poly*' con los dos posibles valores de gamma ('*auto*' y '*scale*'). Tal como se reporta en la Tabla 11, se obtienen mejores resultados con un valor '*scale*' para el parámetro gamma. Por ello, se continuaron todos los entrenamientos con ese valor para gamma. Siguiendo la misma lógica, se observa que los resultados para el kernel radial son mejores, por lo que los posteriores modelos utilizan únicamente este tipo de kernel.

kernel	C	degree	gamma	Accuracy		F1 Score		Modelo válido
				Train	Test	Train	Test	
rbf	20	NA	auto	89,42%	89,46%	86,40%	86,59%	Sí
			scale	92,11%	92,23%	90,22%	90,48%	Sí
poly	20	3	auto	89,23%	89,26%	86,12%	86,30%	Sí
			scale	89,48%	89,53%	86,49%	86,71%	Sí

Tabla 11: Resultados de la primera iteración del modelo de SVM en autovía.

Para la segunda iteración, se comprobaron los resultados para distintos valores de C, variando en 6 distintos valores desde 1 hasta 100.

kernel	gamma	C	Accuracy		F1 Score		Modelo válido
			Train	Test	Train	Test	
rbf	scale	1	90,82%	90,90%	88,41%	88,64%	Sí
		5	91,42%	91,53%	89,26%	89,53%	Sí
		20	92,11%	92,23%	90,22%	90,48%	Sí
		50	92,45%	92,52%	90,69%	90,88%	Sí
		80	92,59%	92,70%	90,89%	91,12%	Sí
		100	92,65%	92,77%	90,96%	91,22%	Sí

Tabla 12: Resultados de la segunda iteración del modelo de SVM en autovía.

Los resultados de la Tabla 12 muestran cómo, a medida que se incrementa el valor de C, la estimación se vuelve más precisa. Siendo estos todos los modelos a entrenar, evitando aumentar el valor de C a uno superior para no caer en overfitting, el modelo seleccionado es el que se ajusta con un valor de 100. El archivo que almacena el modelo ocupa un total de 521,9 KB de memoria.

	<i>Target</i> estimado	<i>Clutter</i> estimado
<i>Target</i> real	8350	28
<i>Clutter</i> real	1065	5676

Tabla 13: Matriz de confusión del clasificador SVM seleccionado para autovía.

La matriz de confusión de este se muestra en la Tabla 13. Como se explicó anteriormente en el apartado 5.2.1, los falsos positivos son el valor más importante, ya que un número elevado de falsos positivos producirá que los objetos no sean detectados. Como consecuencia, los sistemas que utilizaran estos filtros podrían volverse inseguros y no cumplir con los estándares mínimos para su funcionamiento comercial.

En el último modelo de **K-Ve**cinos, también se optó por la estrategia de realizar dos iteraciones de entrenamiento. En la primera iteración, utilizando el valor predefinido de vecinos, se comprobó cuál era la mejor combinación para pesos y algoritmo de criterio. Como se muestra en la Tabla 14, se observa que el ajuste de los pesos por la distancia de los vecinos produce un overfitting, con una diferencia de resultados superior al 7% que se comentó anteriormente. Por esa razón, se descartaron los modelos con los pesos configurados así.

neighbors	algorithm	weights	Accuracy		F1 Score		Modelo válido
			Train	Test	Train	Test	
5	ball_tree	uniform	94,54%	93,37%	93,52%	92,18%	Sí
		distance	100%	93,23%	100%	92,05%	No
	kd_tree	uniform	94,54%	93,37%	93,52%	92,18%	Sí
		distance	100%	92,23%	100%	92,05%	No
	brute	uniform	94,54%	93,37%	93,52%	92,18%	Sí
		distance	100%	92,23%	100%	92,05%	No

Tabla 14: Resultados de la primera iteración de K-Vecinos para autovía.

También se observan unos resultados idénticos para cada modelo, independientemente del algoritmo usado. Debido a ello, se realiza la hipótesis de que el algoritmo no es un punto de inflexión para el modelo. Así pues, para la segunda iteración se utilizó como parámetro 'auto' para la selección del algoritmo. Según la documentación, este parámetro escoge entre el mejor algoritmo de todos. En esta segunda iteración, se comprobaron los resultados con un distinto número de vecinos.

algorithm	weights	neighbors	Accuracy		F1 Score		Modelo válido
			Train	Test	Train	Test	
auto	uniform	3	95,23%	92,72%	94,41%	91,51%	Sí
		5	94,54%	93,37%	93,52%	92,18%	Sí
		10	93,86%	93,46%	92,61%	92,17%	Sí
		20	93,52%	93,38%	92,16%	92,05%	Sí
		30	93,30%	93,29%	91,86%	91,92%	Sí
		50	93,04%	93,06%	91,51%	91,62%	Sí
		100	92,71%	92,78%	91,05%	91,23%	Sí

Tabla 15: Resultados de la segunda iteración de K-Vecinos para autovía.

Observando la tabla, se observa como existen fluctuaciones entre los resultados. Pasando todos los modelos entrenados los requisitos deseados, el modelo seleccionado es aquel entrenado con 10 vecinos, ya que es el que obtiene los valores más altos en las métricas sobre

los datos de evaluación. La Tabla 16 muestra los valores de la matriz de confusión de este. La memoria necesaria que toma el archivo con el modelo ya entrenado es de 2.7MB.

	<i>Target</i> estimado	<i>Clutter</i> estimado
<i>Target</i> real	8310	68
<i>Clutter</i> real	921	5820

Tabla 16: Matriz de confusión del clasificador K-Vecinos seleccionado para autovía.

5.2.3. Escenario de intersección

Comenzando por el clasificador **GaussianNB**, al igual que en el escenario de autovía se tomaron en cuenta dos entrenamientos: el primero con la probabilidad específica del *clutter* para el escenario (54.023%) y el segundo con la probabilidad general (49.134%).

Prob. Clutter	Prob. Target	Accuracy		F1 Score		Modelo válido
		Train	Test	Train	Test	
54,023%	45,977%	64,43%	64,58%	67,54%	67,33%	No
49,134%	50,866%	63,84%	63,88%	62,90%	62,50%	No

Tabla 17: Resultados del clasificador Bayesiano para intersección.

Analizando los resultados de la Tabla 17 se observa que, al igual que en el caso de autovía, el clasificador funciona mejor con la probabilidad específica obtenida para el escenario. No obstante, el porcentaje de acierto se encuentra por debajo del 70%. Por ello, también se descarta el clasificador Bayesiano para su uso en un escenario de intersección.

Continuando con el clasificador de **Random Forest**, la estrategia de entrenamiento es idéntica a la utilizada en el escenario de autovía. Estos entrenamientos se encuentran documentados en la Tabla 18.

n_estimators	criterion	Accuracy		F1 Score		Modelo válido
		Train	Test	Train	Test	
4	gini	96,76%	63,76%	96,94%	65,62%	No
	entropy	96,66%	63,99%	96,85%	65,62%	No
	log_loss	96,69%	64,84%	96,88%	65,33%	No
8	gini	98,71%	64,30%	98,81%	66,98%	No
	entropy	98,76%	62,92%	98,85%	66,15%	No
	log_loss	98,73%	63,65%	98,82%	66,43%	No
10	gini	99,04%	63,14%	99,11%	66,96%	No
	entropy	99,12%	63,57%	99,19%	66,76%	No
	log_loss	99,14%	63,76%	99,21%	67,08%	No
12	gini	99,28%	63,37%	99,34%	66,81%	No
	entropy	99,29%	63,21%	99,35%	66,63%	No
	log_loss	99,35%	64,10%	99,40%	67,51%	No
20	gini	99,72%	63,12%	99,74%	67,17%	No
	entropy	99,77%	63,91%	99,79%	67,42%	No
	log_loss	99,74%	63,91%	99,76%	63,91%	No
50	gini	99,98%	63,17%	99,99%	67,35%	No
	entropy	99,97%	63,86%	99,97%	68,02%	No
	log_loss	99,98%	63,90%	99,98%	67,96%	No

Tabla 18: Resultados del clasificador Random Forest para intersección.

Desafortunadamente, ninguno de los modelos entrenados con Random Forest para intersección es aceptable para ser usado como filtro. En estos se observa como no cumplen ninguno de los dos requisitos: existe una diferencia mayor al 7% y los resultados de test no superan el 70%. Incluso los números indican un claro overfitting.

Debido a los resultados obtenidos para el escenario de autovía, la estrategia para el modelo de **SVM** en intersección se forma por una única iteración, estableciendo como constante los valores del kernel y gamma.

kernel	gamma	C	Accuracy		F1 Score		Modelo válido
			Train	Test	Train	Test	
rbf	scale	1	69,67%	69,50%	71,73%	71,33%	No
		5	70,82%	70,44%	73,23%	72,69%	Sí
		20	71,87%	71,27%	74,12%	73,36%	Sí
		50	72,33%	71,71%	74,57%	73,83%	Sí
		80	72,49%	72,04%	74,78%	74,19%	Sí
		100	72,56%	72,12%	74,88%	74,29%	Sí

Tabla 19: Resultados del clasificador SVM para intersección.

A partir de los resultados de la Tabla 19, se selecciona como modelo el entrenado para un valor de 100 en el parámetro “C”. Este modelo ocupa 1.9MB de memoria. La matriz de confusión de este se muestra en la Tabla 20.

	<i>Target</i> estimado	<i>Clutter</i> estimado
<i>Target</i> real	5055	2277
<i>Clutter</i> real	2149	6396

Tabla 20: Matriz de confusión del clasificador SMV seleccionado para intersección.

Terminando los entrenamientos con el modelo de **K-Ve**cinos, y siendo un caso similar a SVM por los resultados, se realizó una única iteración de entrenamiento, en el que se utilizaron los pesos “uniform” y el algoritmo “auto”, variando el número de vecinos participantes en el voto.

algorithm	weights	n_neighbors	Accuracy		F1 Score		Modelo válido
			Train	Test	Train	Test	
auto	uniform	3	91,46%	83,91%	91,96%	84,70%	No
		5	89,22%	84,01%	89,72%	84,60%	Sí
		10	86,39%	83,34%	86,50%	83,28%	Sí
		20	84,72%	82,62%	84,80%	82,52%	Sí
		30	83,77%	82,33%	82,83%	82,20%	Sí
		50	82,73%	81,70%	82,76%	81,56%	Sí
		100	80,89%	79,89%	80,82%	79,64%	Sí

Tabla 21: Resultados del clasificador de K-Vecinos para intersección

De los resultados mostrados en la Tabla 21 se determina que el mejor modelo es el creado con 5 vecinos. El archivo que almacena este modelo ocupa 3MB. Los valores de la matriz de confusión de este se muestran en la Tabla 22.

	<i>Target</i> estimado	<i>Clutter</i> estimado
<i>Target</i> real	6364	968
<i>Clutter</i> real	1571	6974

Tabla 22: Matriz de confusión del clasificador K-Vecinos seleccionado para intersección.

5.3. Testeo de los modelos en datos sin etiquetar

Habiéndose entrenado y seleccionado los modelos que pueden ser usados para filtros, se requiere de la medición del tiempo medio que cada modelo requiere para la predicción de nuevos resultados. Para la realización de esta tarea, se utilizaron nuevas grabaciones para cada escenario, cada una perteneciente a la misma autovía e intersección utilizadas para el entrenamiento.

5.3.1. Filtros para autovía

Los datos de autovía utilizados para la validación se obtuvieron utilizando el mismo radar, en la misma geolocalización, durante otra fecha distinta y realizando una grabación más longeva. En consecuencia, en esta grabación se identifican un total de 92643 detecciones no etiquetadas.

La evaluación de estas detecciones se llevó a cabo con los modelos guardados de *Random Forest*, SVM y K-Vecinos, siendo el clasificador Bayesiano el único descartado. La Tabla 23 recopila los tiempos utilizados por cada filtro.

Modelo	Tiempo de procesamiento total	Tiempo por detección
Random Forest	907,0224 ms	9,79 μ s
SVM	226,9463 s	2,45 ms
K-Vecinos	3,1288 s	30,78 μ s

Tabla 23: Tiempos de procesamiento con los filtros para autovía.

Obteniendo así los resultados, estos se analizaron de dos formas:

- Cuantificación del número de *clutter* estimado y su ratio frente al total de las detecciones de la grabación.
- Representación visual del archivo eliminando el *clutter*.

Modelo	Clutter estimado	Ratio clutter/detecciones
Random Forest	64503	69,63 %
SVM	70328	75,92 %
K-Vecinos	66759	72,06 %

Tabla 24: Estimaciones de clutter con los filtros para autovía.

Tras la estimación del *clutter* en los escenarios, estos se analizaron con la herramienta gráfica utilizada anteriormente para obtener una representación visual. Por no haber una necesidad extra, las gráficas a continuación muestran únicamente la representación de cada archivo para la velocidad radial, la cual es información suficiente para detectar si el filtro realiza un buen trabajo. En estas gráficas se muestra en color rojo todas las detecciones de *clutter* y en color azul, aquellas pertenecientes a trayectos.

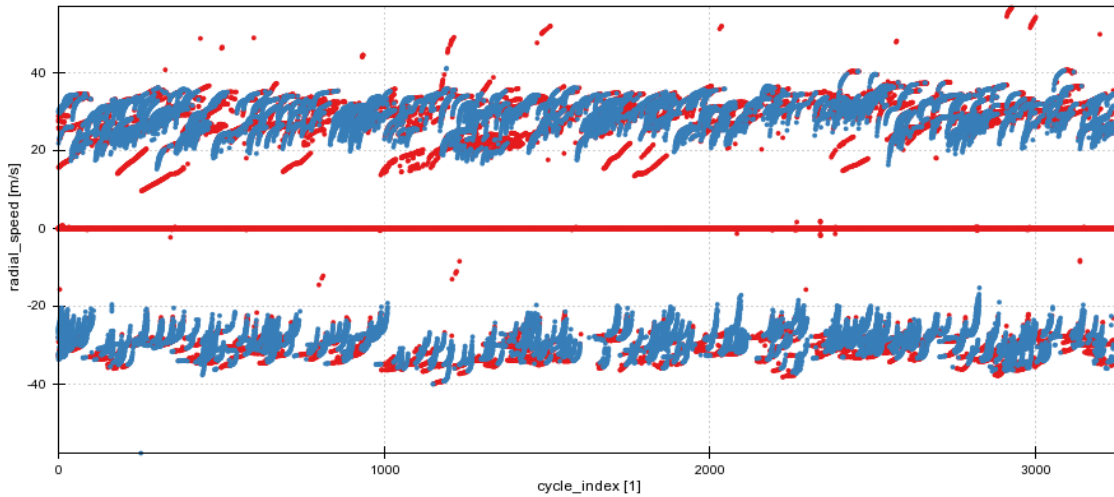


Figura 40: Estimaciones de autovía con el filtro de Random Forest.

En la Figura 40 se representan las estimaciones obtenidas con el filtro de Random Forest. En estas se aprecian:

- Un perfecto filtrado de las detecciones de velocidades casi nulas.
- Un perfecto filtrado del tráfico transversal (velocidad entre 10 y 20).
- Una aproximación decente para todos los trayectos en general, apreciándose problemas para el comienzo de los trayectos en el tráfico que se aproxima (velocidades negativas).

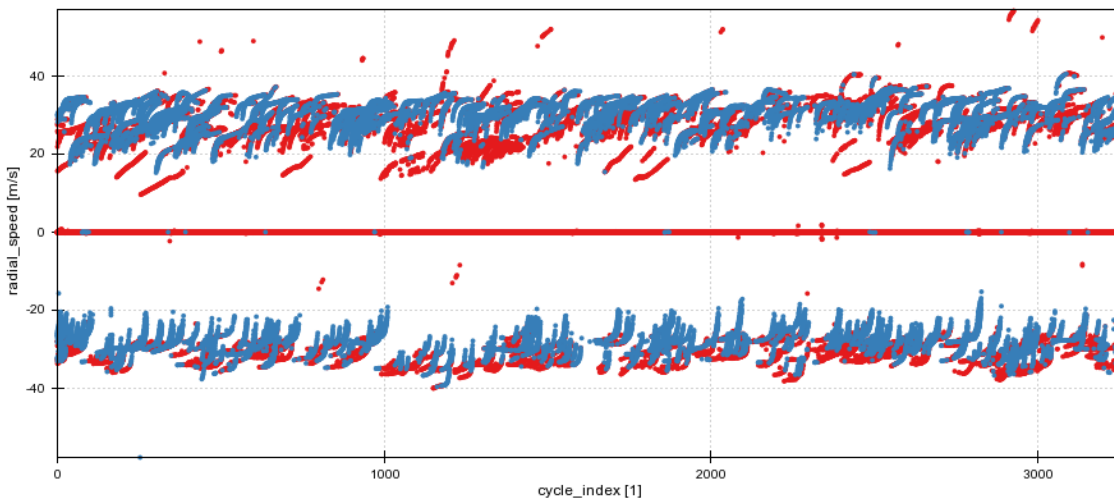


Figura 41: Estimaciones de autovía con el filtro de SVM.

La Figura 41 muestran los resultados obtenidos con el filtro de SVM, en los que se aprecia:

- Un filtrado decente de las detecciones de velocidad nula, siendo el modelo que más falsos negativos presenta.
- Un buen filtrado del tráfico transversal.
- Detecciones robustas sobre los trayectos identificados, pero obteniendo muchos falsos positivos para otros trayectos de forma genérica.

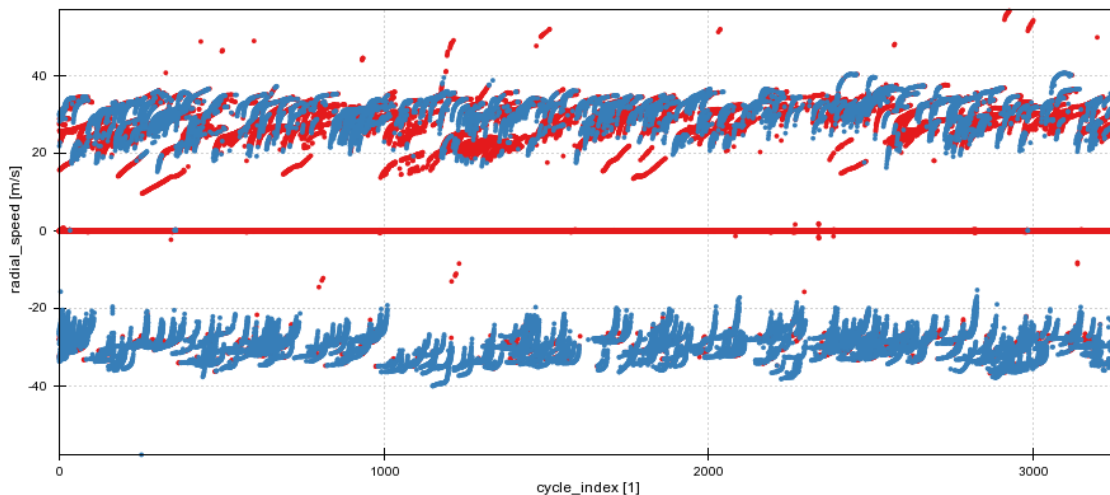


Figura 42: Estimaciones de autovía con el filtro de K-Vecinos.

La Figura 42 muestra así la gráfica obtenida con el filtro de K-Vecinos. En estos resultados se aprecian que:

- El filtro actúa de manera más relajada en el tráfico que se acerca (velocidades negativas en la banda de -20 a -40).
- El tráfico transversal, aún siendo en el sentido contrario (velocidad entre 10 y 20) es filtrado en su mayoría.
- La gran mayoría de ruido con velocidad casi nula es filtrado, a excepción de dos detecciones que no causarán gran impacto en algoritmos de tracking.
- El tráfico saliente es el tramo más sensible, teniendo conjuntos de detecciones incorrectamente etiquetados. En comparación con Random Forest, se encuentran más detecciones definidas como *clutter*.

5.3.2. Filtros para intersección

Los datos de validación de intersección se componen por las detecciones de otro sistema radar, sobre la misma intersección, pero con distinta orientación sobre ella. Estos fueron

grabados simultáneamente que los datos utilizados para el entrenamiento. La grabación contiene 65840 detecciones no etiquetadas.

Los filtros que utilizar en este escenario son los creados por los modelos de SVM y K-Vecinos, habiendo descartado los modelos de clasificador Bayesiano y *Random Forest*.

Los resultados de los análisis se encuentran en la Tabla 25 y la Tabla 26 a continuación, basándose estas en los mismos requisitos que se utilizaron para autovía.

Modelo	Tiempo de procesamiento total	Tiempo por detección
SVM	192,9622 s	2,93 ms
K-Vecinos	1,9918 s	30,25 μ s

Tabla 25: Tiempos de procesamiento con los filtros para intersección.

Modelo	Clutter estimado	Ratio clutter/detecciones
SVM	25948	39,41 %
K-Vecinos	35933	54,58 %

Tabla 26: Estimaciones de clutter con los filtros para intersección.

A continuación, al igual que en el caso de autovía, se presentan las gráficas obtenidas con las estimaciones de estos filtros. Manteniendo el sistema de colores, se presenta de rojo todas las detecciones clasificadas como *clutter* y de azul, como detecciones que forman parte de trayectorias.

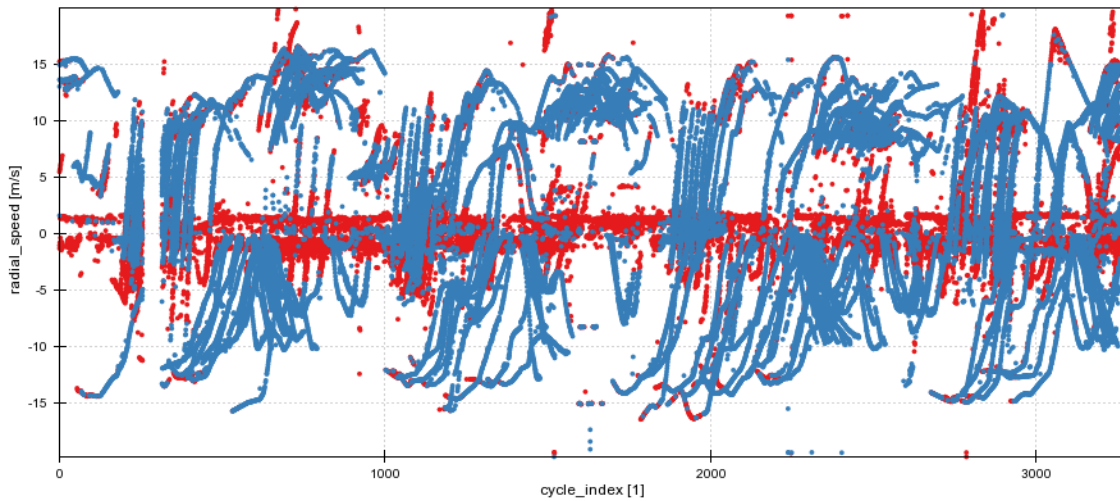


Figura 43: Estimaciones de intersección con el filtro de SVM.

La Figura 43 muestra los resultados obtenidos con el filtro de SVM, en los que se observa:

- Una robusta detección de trayectorias, donde se ven ininterrumpidas por la mayor parte de sus detecciones.
- Una deficiencia en el filtrado del *clutter* debido a interferencias de otros radares.
- Un fuerte filtrado a las detecciones de velocidades bajas, eliminando trayectos de peatones.

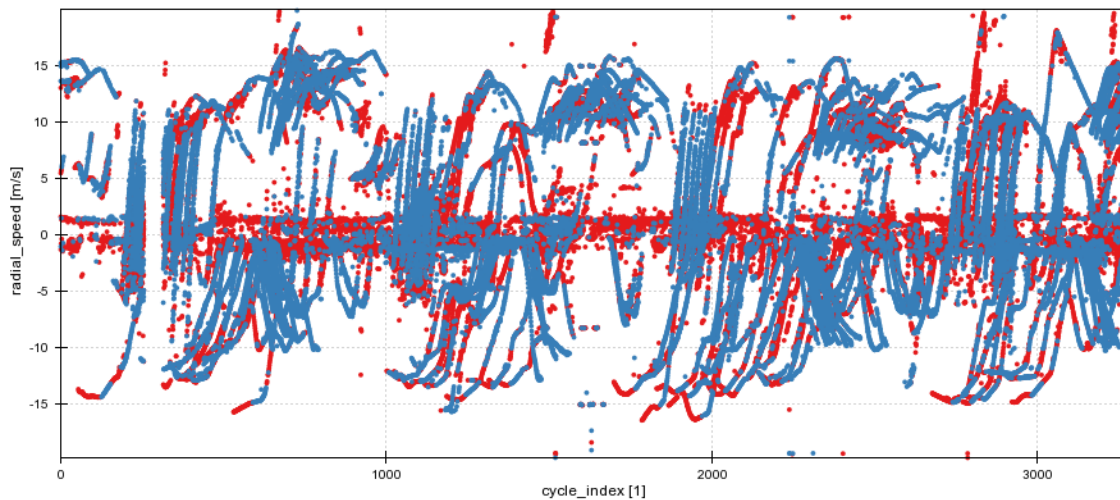


Figura 44: Estimaciones de intersección con el filtro de K-Vecinos.

En la Figura 44 se muestra la gráfica con los resultados obtenidos con el algoritmo de K-Vecinos. Sobre estos se aprecia, de forma genérica, un filtrado ineficiente sobre todos los datos. Esto se respalda en:

- Falsos positivos como *clutter* en varios tramos intermitentes a lo largo de distintos trayectos.
- Un mal filtrado de detecciones de *clutter* debido a las interferencias de otros radares.
- Varios falsos positivos y falsos negativos en la banda de detecciones de bajas velocidades.

Al igual que con SVM, este filtro no detecta correctamente los peatones, los cuales se caracterizan por encontrarse en la banda de bajas velocidades durante tramos de tiempo longevos.

6. Discusión y análisis de resultados

En este capítulo se analizan los resultados de los filtros previamente entrenados y seleccionados. Para la comparación, se recopilan todos los datos a evaluar por las métricas, definidas en la sección 4.6, en tablas de escenarios (Tabla 27 y Tabla 28).

De la misma forma, en la Figura 45 y la Figura 46 se muestran representaciones visuales de la memoria usada y el tiempo de procesamiento por detección respectivamente. En ambas gráficas, un número menor indica un mejor resultado.

Filtros de clutter para autovía						
Modelo	Memoria	Tiempo por detección	Accuracy		F1 Score	
			Train	Test	Train	Test
Random Forest	21,2 MB	9,79 μ s	99,97%	93,81%	99,96%	92,73%
SVM	521,9 KB	2,45 ms	92,65%	92,77%	90,96%	91,22%
K-Vecinos	2,7 MB	30,78 μ s	93,86%	93,46%	92,61%	92,17%

Tabla 27: Comparación de los resultados en los filtros de autovía.

Filtros de clutter para intersección						
Modelo	Memoria	Tiempo por detección	Accuracy		F1 Score	
			Train	Test	Train	Test
SVM	1,9 MB	2,93 ms	72,56%	72,12%	74,88%	74,29%
K-Vecinos	3,0 MB	30,25 μ s	89,22%	84,01%	89,72%	84,60%

Tabla 28: Comparación de los resultados en los filtros de intersección.

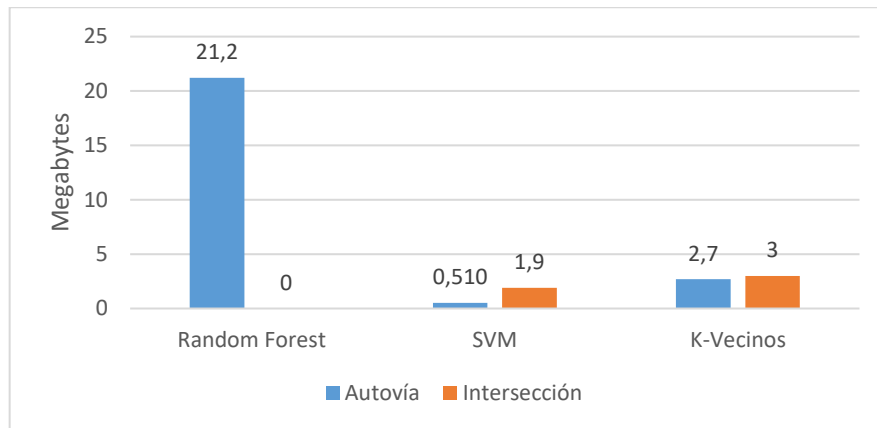


Figura 45: Gráfica de memoria utilizada por los filtros.

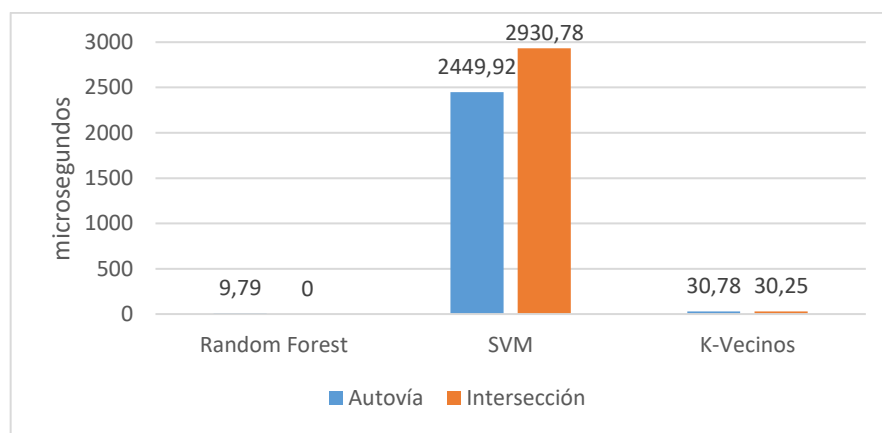


Figura 46: Gráfica de tiempo de procesamiento de los filtros por detección.

Para la definición de si un módulo genérico es válido para un sistema portable en tiempo real, este ha de cumplir dos requisitos principales:

- La memoria utilizada ha de ser menor a 32 MB.
- El tiempo de respuesta ha de ser menor a 5 ms.

Estas especificaciones genéricas para un sistema básico muestran que no todos los filtros creados son apropiados para cualquier tipo de solución:

Random Forest presenta siempre el modelo más rápido de todos, a cambio de ser el modelo que más memoria ocupa. Aunque se ajuste al requerimiento de memoria, en el proceso de diseño final este modelo es sólo un módulo, siendo crítico la consideración del espacio ocupado también por otros módulos necesarios.

De la misma forma, los resultados de evaluación, graficados en la Figura 40, muestran que el modelo todavía muestra un porcentaje importante de falsos positivos y falsos negativos, los cuales pueden aparecer debido al *overfitting* del modelo.. Por ello, este modelo parece el menos estable, en cuanto a resultados, de los seleccionados.

Support Vector Machine, en contraparte, demuestra ser el modelo que menos memoria requiere. La mayor desventaja que presenta es ser el modelo que más lento y costoso computacionalmente.

En la Figura 46, se observa cómo el tiempo de procesamiento es cercano a los 3ms por detección, el cual es un valor crítico para sistemas en tiempo real. Este dato descarta a SVM como filtro que pueda ser usado en tiempo real en sistemas portables, teniendo también en cuenta que estos suelen estar formados por componentes de menor potencia para el cálculo computacional. No obstante, las gráficas de la Figura 43 muestran que, debido a la continuidad en los negativos detectados, este modelo puede llegar a ser una buena opción para post-procesamientos offlines.

Por último, **K-Vecinos** presenta el modelo con mejores cualidades para el uso de un filtro en tiempo real para sistemas portables. La memoria que requiere el modelo es un valor que puede ser considerado en la fase de diseño de un sistema portable. A su vez, el tiempo de procesamiento requerido es aceptable, pudiendo ser mejor, pero si ser un mal tiempo para su implementación en tiempo real.

La mayor desventaja de este modelo se encuentra en sus predicciones. Podría considerarse la realización de múltiples entrenamientos con distintas bases de datos, pero ello también indicaría un aumento de la memoria a utilizar. Este caso necesitaría de un estudio más profundo.

7. Conclusiones y trabajo futuro

Tras el desarrollo del trabajo, en este capítulo se muestra en conjunto las conclusiones, remarcando cuál era el objetivo inicial, de los resultados obtenidos. Posteriormente, se muestran varias líneas de trabajo futuro, pues se considera que, efectivamente como inicialmente estimado, la estrategia de un filtro inteligente para la eliminación del clutter puede ser una buena adición para investigaciones más profundas.

7.1. Conclusiones

Se planteó como objetivo del trabajo, la construcción de un filtro inteligente en tiempo real, que pudiera ser utilizado en soluciones al alcance de la sociedad, para aplicaciones de ciudades inteligentes y conducción autónoma con sistemas radar integrados. Para ello, se ha llevado a cabo una serie de comparativas y pruebas experimentales que valoren la viabilidad de la creación de dicho filtro. Estas pruebas se han basado en escenarios de control de tráfico, ya que definen un escenario simple, al no modificarse el posicionamiento del sensor, pero lo suficientemente complejo en función de su entorno.

Para llevar a cabo la implementación del filtro deseado, el primer punto era identificar las limitaciones de memoria y capacidad de procesamiento que puede disponer un sistema portable. Debido a estas limitaciones, el uso de las redes neuronales no es una solución válida, ya que estos sistemas necesitan mayor capacidad de procesamiento y memoria que la disponible en la mayoría de sistemas portables de gamma media, los cuales utilizan una memoria del orden de megabytes y no disponen de unidades de alto procesamiento, como son las GPU. Por ello, se optó por explorar implementaciones basadas modelos clásicos de machine learning para clasificación.

La base de datos utilizada fue obtenida gracias a las grabaciones proporcionadas por la compañía Smartmicro, las cuales presentaban todas las detecciones, a lo largo de 3 minutos, para los escenarios de intersección y de autovía. Estos datos mostraban, para cada detección, las cuatro dimensiones detectables por el sistema: rango, velocidad radial, azimuth y RCS. Estas grabaciones fueron sometidas a un etiquetado manual, identificando así todos los trayectos posibles en cada una. Cada conjunción de puntos se tomó como un trayecto válido si presentaba una evolución lineal en las gráficas temporales de velocidad, azimuth y rango.

Una vez estuvieron todos los datos etiquetados, estos fueron sometidos a un análisis que permitió identificar detecciones con valores anómalos en el RCS, como se ha desarrollado en la sección 4.3. Con ello, en el post procesamiento de los datos se eliminaron estos valores anómalos y se migró la información de los trayectos etiquetados a una etiqueta nueva, la cual define si cada detección es *clutter* o no. Con las bases de datos preparadas, se continuó con la selección de los modelos a utilizar.

Dentro de los modelos clásicos se seleccionaron: Random Forest, SVM y k-vecinos debido a que las dimensiones de los radares presentan características separables al *clutter*, pueden ser separados por kernels (en función del escenario), y muestran proximidad a detecciones de su misma clase, pues las detecciones clasificadas como objetos suelen componerse de grupos de detecciones con características similares.

De los modelos clásicos y después de las experimentaciones realizadas con cada uno de ellos, se concluyó que tres de ellos, Random Forest, SVM y k-vecinos, presentaban resultados que podrían indicar un modelo aceptable para su utilización. No obstante, las limitaciones de memoria y procesamiento impuestas hacían que el modelo de SVM no fuera válidos para los escenarios planteados en la experimentación (muestra un tiempo de procesamiento por detección superior al requerido para un sistema en tiempo real).

Después de la implementación, entrenamiento y validación de los modelos, la principal conclusión del trabajo es que dado que cada escenario se rige por las normas de su entorno y los objetos que se desplazan en él, se demuestra con los resultados mostrados en las secciones 5.2 y 5.3 que se requiere la creación de filtros específicos en función del escenario y/o sistema radar, utilizando los modelos que mejor se ajusten en cada caso definido por el escenario.

Las gráficas de las estimaciones demuestran que, incluso basándose en un etiquetado manual sin una clara visualización de los objetos, es posible la creación de filtros desplegados en sistemas embebidos. Una implementación industrial requerirá de un trabajo más profundo para el cumplimiento de los estándares de eficiencia y seguridad.

Se considera que, siendo parte del problema el etiquetado manual, sin referencia más que comprobar una continuidad lineal en las dimensiones de rango, velocidad radial y azimut, un mejor etiquetado devolverá los resultados deseados.

Finalmente, aunque el estudio no defina modelos implementables a escala industrial, los modelos obtenidos se definen como una gran base de referencia para la creación de filtros implementables.

7.2. Líneas de trabajo futuro

- Expansión de la base de datos: Definir una base de datos con un etiquetado superior, que pueda definir unos mejores resultados en los modelos definidos.
- Búsqueda de otros modelos: Utilización de distintos modelos que cumplan con el objetivo inicial de ser utilizados como filtros en sistemas portables de tiempo real.
- Modificación de las variables de entrada: Definición de un distinto vector de entrada, que incluya variables distintas, ya sean añadidas o cambiadas por las actuales, al rango, velocidad radial, azimut y RCS.
- Sistemas de clasificación de trayectos: Un filtro de *clutter* es también un detector de trayectos de objetos deseados. La solución de implementar un modelo detector de trayectos es también una línea que seguir.
- Entrenamientos con distintos datos y/o escenarios: Las bases de datos utilizadas se obtuvieron de los escenarios de intersección y autovía de dos países distintos. Siendo ambos escenarios bastante ruidosos, es posible que la utilización de distintas bases de datos devuelva mejores resultados.
- Migración de los modelos: El proyecto presenta varios modelos de filtrado de *clutter* para soluciones de tráfico. La implementación de estos para los vehículos autónomos indica la migración de estos a soluciones de automoción.

Referencias bibliográficas

- Addabbo, P., Han, S., Orlando, D., & Ricci, G. (2021). Learning Strategies for Radar Clutter Classification. *IEEE Transactions on Signal Processing*, 69, 1070 - 1082. doi:10.1109/TSP.2021.3050985
- Arab, H., Ghaffari, I., Chioukh, L., Tatu, S., & Dufour, S. (2021). Machine learning based object classification and identification scheme using an embedded millimeter-wave radar sensor. *Sensors*. doi:10.3390/s21134291
- Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight Uncertainty in Neural Networks. *32nd International Conference on Machine Learning (ICML 2015)*. doi:10.48550/arXiv.1505.05424
- Bouvier, C., Martinet, L., Favier, G., Sedano, H., & Artaud, M. (1995). Radar clutter classification using autoregressive modelling, K-distribution and neural network. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. doi:10.1109/ICASSP.1995.480091
- Cortes, C., & Vapnik, V. (Septiembre de 1995). Support-vector networks. *Machine Learning*, 20, 273-297. doi:https://doi.org/10.1007/BF00994018
- Cozma, A.-E., Morgan, L., Stolz, M., Stoeckel, D., & Rambach, K. (2021). DeepHybrid: Deep Learning on Automotive Radar Spectra and Reflections for Object Classification. *IEEE International Intelligent Transportation Systems Conference (ITSC)*. doi:10.1109/ITSC48978.2021.9564526
- Darzikolaei, M. A., Ebrahimzade, A., & Gholami, E. (2015). Classification of radar clutters with Artificial Neural Network. *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)* (págs. 577-581). IEEE. doi:10.1109/KBEI.2015.7436109
- Dawkins, T. (3 de Agosto de 2021). *¿Qué sensores se utilizan en los vehículos autónomos?* Obtenido de Level Five Supplies: <https://levelfivesupplies.com/automoviles-autonomos-101-que-sensores-se-utilizan-en-los-vehiculos-autonomos/>

- Dube, R., Hahn, M., Schutz, M., Dickmann, J., & Gingras, D. (2014). Detection of parked vehicles from a radar based occupancy grid. *IEEE Intelligent Vehicles Symposium, Proceedings*, (págs. 1415-1420). doi:10.1109/IVS.2014.6856568
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 46-57. doi:10.1109/2.30720
- Fix, E., & Hodges, J. L. (1951). *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*. Randolph Field, Texas: USAF School of Aviation Medicine.
- García Oliva, C. (13 de Agosto de 2018). *Esta es la historia del coche autónomo, y ojo porque no es tan nuevo como lo pintan...*. Obtenido de Autonomación: <https://www.autonocion.com/historia-coche-autonomo/>
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/CVPR.2012.6248074
- Giebenhain, S., Wang, L., Anklam, C., & Goldluecke, B. (2021). Radar Ghost Target Detection via Multimodal Transformers. *IEEE Robotics and Automation Letters*, 6(4), 7758 - 7765. doi:10.1109/LRA.2021.3100176
- Gomez, J. (28 de Julio de 2017). *Por qué la conducción autónoma requiere de tantos sensores*. Obtenido de High Motor: <https://www.highmotor.com/imagen-muestra-que-conduccion-autonoma-requiere-tantos-sensores.html>
- Ho, T. K. (1995). Random decision forests. En IEEE (Ed.), *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Montreal, QC, Canada. doi:10.1109/ICDAR.1995.598994
- Holder, M., Linnhoff, C., Rosenberger, P., Popp, C., & Winner, H. (2019). Modeling and Simulation of Radar Sensor Artifacts for Virtual Testing of Autonomous Driving. *Tagung Automatisiertes Fahren*.
- Jochumsen, L., Ostergaard, J., Jensen, S., Clemente, C., & O. Pedersen, M. (2016). A recursive kinematic random forest and alpha beta filter classifier for 2D radar tracks. *Eurasip Journal on Advances in Signal Processing*. doi:10.1186/s13634-016-0378-3

- Keat, C. T., Pradalier, C., & Laugier, C. (2005). Vehicle detection and car park mapping using laser scanner. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (págs. 2054-2060). doi:10.1109/IROS.2005.1545233
- km77. (s.f.). *Conducción autónoma | Niveles y tecnología*. Obtenido de <https://www.km77.com/reportajes/varios/conduccion-autonoma-niveles>
- Langley, P., Iba, W., & Thompson, K. (Julio de 1992). An analysis of Bayesian classifiers. *Aaii, 90*, 223-228.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 436-444. doi:10.1038/nature14539
- Li, Y., Hernandez-Lobato, J. M., & Turner, R. E. (2015). Stochastic Expectation Propagation. *Neural Information Processing Systems (NIPS)*. doi:10.48550/arXiv.1506.04132
- López, J. M. (15 de Agosto de 2020). *El vehículo autónomo es el futuro, pero tiene más años que la mayoría de nosotros*. Obtenido de Hipertextual: <https://hipertextual.com/2020/08/origen-historia-vehiculo-autonomo>
- Maeda, S.-i. (2014). A Bayesian encourages dropout. doi:10.48550/arXiv.1412.7003
- Minka, T. P. (2001). Expectation Propagation for Approximate Bayesian Inference. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 362-369.
- Naidu, V., Gopalratnam, G., & Raol, J. (August de 2003). Evaluation of Data Association and Fusion Algorithms for Tracking in the Presence of Measurement Loss., (pág. 5733). doi:10.2514/6.2003-5733
- Palffy, A., Dong, J., Kooij, J. F., & Gavrila, D. M. (2020). CNN Based Road User Detection Using the 3D Radar Cube. *IEEE Robotics and Automation Letters*, 1263-1270. doi:10.48550/arXiv.2004.12165
- Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., & Yang, B. (2019). Deep learning-based object classification on automotive radar spectra. *IEEE Radar Conference (RadarConf)*. doi:10.1109/RADAR.2019.8835775

- Scheiner, N., Appenrodt, N., Dickmann, J., & Sick, B. (2018). Radar-based Feature Design and Multiclass Classification for Road User Recognition. *Proceedings of IEEE Intelligent Vehicles Symposium*, 779-786. doi:10.1109/IVS.2018.8500607
- Sun, Z., Bebis, G., & Miller, R. (2006). On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Learning*, 28(5), 694-711. doi:10.1109/TPAMI.2006.104
- Ulrich, M., Bosch, F. T., Gläser, C., & Timm, F. (2021). DeepReflecs: Deep Learning for Automotive Object Classification with Radar Reflections. *IEEE Radar Conference (RadarConf21)*. doi:10.1109/RadarConf2147009.2021.9455334
- Wolff, C. (s.f.). *Physical fundamentals of the radar principle*. Obtenido de Radar Tutorial: <https://www.radartutorial.eu/01.basics/Physical%20fundamentals%20of%20the%20radar%20principle.en.html>
- Zhou, J., Navarro-Serment, L. E., & Hebert, M. (2012). Detection of parking spots using 2D range data. *15th International IEEE Conference on Intelligent Transportation Systems*. doi:10.1109/ITSC.2012.6338706
- Zhou, Y., Liu, L., Zhao, H., López-Benítez, M., Yu, L., & Yue, Y. (2022). Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors*, 22(11), 4208. doi:10.3390/s22114208

Anexo A. Artículo de investigación

Análisis de filtros inteligentes para ecos de radares de automoción



Jaime González Gómez

Universidad Internacional de la Rioja, Logroño (España)

20 de julio de 2023

RESUMEN

Se propone la comparación de varios algoritmos de clasificación que puedan ser usados para la aplicación del filtrado de *clutter*, en entornos de automoción donde la toma de datos se realiza mediante sistemas radar. Se evita la creación de estos basándose en redes neuronales buscando que puedan ser desplegados y utilizados en sistemas embebidos. Los modelos seleccionados son el clasificador Bayesiano, *Random Forest*, SVM y el clasificador de K-Vecinos. Entrenando los modelos para escenarios de control de tráfico de autovía e intersección, se observa que de forma general el modelo de SVM muestra los mejores resultados para un modelo offline, mientras que para su uso online los ideales son *Random Forest* en autovía y K-Vecinos en intersección. Estos modelos pueden ser utilizados para fases previas al procesamiento de los datos para la detección de objetos con sistemas de radar.

PALABRAS CLAVE

automoción, *clutter*, detección de objetos, radar, tráfico

I. INTRODUCCIÓN

LA conducción autónoma es un campo de investigación en auge. No obstante, los resultados que muestran las múltiples investigaciones se encuentran lejos del alcance de la sociedad, debido a un uso tecnológico que, económicamente, no se encuentra disponible para la media en la sociedad. Otro de los mayores inconvenientes para la obtención de un vehículo autónomo es conseguir que el autómata desarrolle una percepción de su entorno que sea en tiempo real.

El trabajo aportado busca abordar un procesado previo a los datos obtenidos con sistemas radar, pudiendo así utilizarse este en aplicaciones de control de tráfico y automoción. La detección de objetos a través del análisis de los ecos del radar es una práctica compleja, la cual se basa actualmente en métodos estadísticos y matemáticos sobre nubes de puntos con porcentajes cercanos al 50% de *clutter*.

La realización del trabajo lleva consigo las tareas de tomar y analizar grabaciones correspondientes a los escenarios de autovía e intersección, necesitando ambos un etiquetado manual. Posteriormente requiere el entrenamiento y selección de los modelos buscando que se adapten al objetivo final, obteniendo así filtros de *clutter* inteligentes que puedan ser utilizados en todo tipo de soluciones.

II. ESTADO DEL ARTE

En las dos últimas décadas, la investigación de las aplicaciones de conducción autónoma ha entrado en auge, gracias también al rápido crecimiento de la inteligencia artificial en la sociedad. A medida que la tecnología y la industria avanza en el campo, buscando llevar sus resultados a mercado, se observa una tendencia de cambio en los estudios y las publicaciones.

Las primeras publicaciones se basan, en su mayoría, en sistemas LIDAR 2D. Una de ellas hace referencia a la búsqueda de espacio de aparcamiento y mapeado de la zona [6]. Fue gracias a la creación de bases de datos como KITTI [4] las que causaron que el campo se volviese más atractivo. Estas bases de datos presentan material para investigaciones de odometría, semántica y detección de objetos gracias a los datos captados por LIDAR y cámaras, ofreciendo así datos al alcance de muchos.

Estas comienzan una tendencia migratoria de investigar con sistemas Lidar a investigar con sistemas radar, siendo estos también más asequibles económicamente para el mercado y disponiendo datos que pueden ser procesados en tiempo real. Aparecen investigaciones retomando la línea de trabajo anterior, Dube et al. presenta un algoritmo que funciona en tiempo real para aplicaciones de detección de vehículos estacionados [3].

En este tiempo comienza también a suceder el auge de las redes neuronales, las cuales abren una nueva alternativa para el reto de la clasificación de objetos. El modelo propuesto por Scheiner et al. busca realizar un

agrupamiento previo, usando DBSCAN sobre las detecciones del radar, para usar esta información como datos de entrada para los modelos de *Random Forest* y LSTM [10]. El mismo reto es también abordado a través de redes neuronales profundas utilizando la matriz azimuth-rango [9]. Estos son puntos de partidas retomados para un cubo 3D de radar azimuth-rango-doppler [8], extrayendo de este otros sub-cubos inferiores usando DBSCAN para tomarlos como entrada de un modelo CNN.

Mientras los modelos anteriores presentan buenos resultados, divergen de la idea que busca una compatibilidad de las aplicaciones e investigaciones dentro del mercado actual, pues un modelo de redes neuronales profundo necesita de unos requerimientos costosos para soluciones portables. Retomando esta idea, se publica también un modelo basado en múltiples SVM en cascada [1], creando también así un modelo de caja blanca con unos resultados excelentes, pero únicamente probado sobre datos simulados o medidos en un laboratorio.

Al encontrarse el campo de la investigación de aplicaciones ADAS con radar con estudios previos muy selectos, la comunidad comienza a escribir publicaciones con objetivos alternativos, como la detección de objetos fantasmas [5]. También se encuentran publicaciones de extensas metodologías que puedan ser reutilizadas por los radares 4D de la próxima generación [11].

Naidu et al. presentan un análisis de los datos, enfocados a casos *multi sensor multi target* [7], a través del uso del Filtro de Kalman basado en Nearest Neighbours (NNKF) y filtros probabilísticos de asociación de datos (PDAF). Naidu afirma que ambos algoritmos son eficaces en presencia de *clutter* moderado, mas siguen sin obtener resultados perfectos.

Al ser este tipo de procesamiento muy costoso matemáticamente, gran parte de la comunidad científica busca la alternativa con el entrenamiento profundo para llegar el objetivo. En contraparte, este trabajo a presentar busca basarse en una de las etapas previas a la detección de objetos, pudiendo también apoyar a posteriores procesados, como el presentado por Naidu et al.

Las investigaciones del filtrado de datos de radar son escasas, siendo de las más actuales modelados autorregresivos de K-distribuciones [2]. El trabajo se basa en la clasificación de *clutter* para sistemas radar utilizados en vehículos aéreos, presentando un único modelo MLP para clasificar el *clutter* en función del parámetro de la distribución. Posteriores documentos sobre el mismo ámbito se centran en el procesado de datos con métodos estadísticos y matemáticos, sin entrar en el campo de la inteligencia artificial.

Consecuentemente, el despliegue de modelos de *machine learning* no es posible actualmente en sistemas de automoción. Mientras que las redes neuronales presentan unos resultados excepcionales en multitud de campos, como puede ser el tratamiento de imagen o reconocimiento e interpretación de un lenguaje, presentan dos características que hacen imposible, hoy en día, su implementación en sistemas portátiles: la energía requerida para su uso y el tamaño de memoria a utilizar en un sistema embebido.

III. OBJETIVOS

El objetivo de este trabajo es la creación de varios filtros de *clutter*, basados en modelos predictivos estadísticos, que puedan ser portables e instalables en sistemas radar de automoción contemporáneos.

Es imprescindible que todos los filtros que cumplan este objetivo, cumplan también las siguientes características:

- Han de ser modelos que puedan ser instalados en sistemas portables, es decir, con recursos de memoria y procesamiento limitados.
- Deben ser capaces de separar los datos en detecciones verdaderas y detecciones falsas.
- Ha de ser capaz de trabajar con datos de un radar que pueda poseer interferencias de otros radares.

IV. CONTRIBUCIÓN

La aportación de este trabajo es la evaluación de distintos modelos de inteligencia artificial, siendo estos entrenados para filtros inteligentes de *clutter*, manteniendo el objetivo de la creación de uno que pueda ser utilizado en sistemas embebidos.

Para ello, se comparan cuatro modelos (Clasificador Bayesiano, SVM, *Random Forest* y K-Vecinos) en dos escenarios distintos: autovía e intersección. Estos escenarios han sido etiquetados de manera manual, definiéndose como *target* a toda aquella detección, perteneciente a un grupo que presente una evolución lineal temporal en las gráficas de rango, velocidad radial y azimuth.

El escenario de autovía se caracteriza por varios factores, siendo el más referente que todos los objetos detectados se encuentran en velocidades de entre 80 km/h y 120 km/h. Con ello, se diferencian franjas en las que se sitúan la mayoría de los *targets* o la mayoría del *clutter*, pues la gran mayoría de *clutter* es procedente de ruido en bajas frecuencias de la señal o de objetos persistentes en la escena. Ambos casos de *clutter* se detectan con velocidades cercanas a 0. La otra característica del escenario es que todos los objetos siguen rutas similares.

En contraparte, el escenario de intersección muestra un entorno caótico con objetos con múltiples trayectorias, velocidades no constantes. En este, los objetos suelen viajar a un máximo de 60 km/h siempre que sean vehículos, pues muchas detecciones son referentes a peatones o ciclistas. Debido a la baja velocidad de los mismos, no se pueden distinguir franjas, similares a las del escenario anterior, pudiendo filtrar fácilmente el *clutter*.

Para el entrenamiento y evaluación, cada escenario se dividió en las partes 80/20. Siendo este un problema de clasificación binaria, la variable objetivo se identifica como *clutter*.

Todos los modelos se definen con el mismo vector de entrada, constituido por las cuatro dimensiones que definen una detección: rango, azimuth, velocidad radial y RCS. Estos valores se encuentran normalizados con la estrategia minmax para cada escenario.

V. EVALUACIÓN Y RESULTADOS

Dado las distintas características de cada escenario, el porcentaje mínimo de acierto, sobre los datos de test, para tomar un modelo como válido se define en el 85% para el

escenario de autovía y 70% para intersección.

Los hiperparámetros de cada modelo han sido buscados a través de varias iteraciones de entrenamiento, seleccionando siempre el mejor de los resultados.

Escenario de autovía

A través de varias iteraciones de entrenamientos para los modelos, se ajustaron todos los hiperparámetros. La selección de los mejores modelos fue realizada para las siguientes configuraciones:

- **GaussianNB**: Probabilidad a priori de 44,245% para *clutter*. La probabilidad se obtuvo a partir del análisis de los datos del escenario.
- **Random Forest**: 50 estimadores con el criterio de “entropy”.
- **SVM**: Kernel radial y con C = 100.
- **K-Vecinos**: 10 vecinos para el conteo de voto y pesos uniformes.

Siendo estos los mejores filtros para cada modelo, el único que no cumple con las condiciones de un ajuste al 85% es el clasificador Bayesiano. Los resultados de los modelos se muestran en la Tabla I.

Escenario de intersección

La estrategia de entrenamiento, selección y validación es la misma que la llevada a cabo en el escenario de autovía. Con ello, los hiperparámetros configurados en los modelos seleccionados son los siguientes:

- **GaussianNB**: Probabilidad a priori de 54,023% para *clutter*. La probabilidad se obtuvo a partir del análisis de los datos del escenario.
- **Random Forest**: 4 estimadores con el criterio de “log_loss”.
- **SVM**: Kernel radial y con C = 100.
- **K-Vecinos**: 5 vecinos para el conteo de voto y pesos uniformes.

En este escenario, son dos los modelos que no se ajustan al requerimiento del 70% de acierto. Estos son el clasificador Bayesiano y el modelo de *Random Forest*. Los resultados de los modelos se muestran en la Tabla II.

VI. DISCUSIÓN

Tras la obtención de los modelos se interpreta que, en función de la aplicación a diseñar, el mejor modelo puede variar.

En aplicaciones de autovía, *Random Forest* presenta un overfitting deseado, ya que los datos nuevos del escenario siempre se ajustarán a los entrenados. Esto se cumple gracias a las características predecibles y acotadas de los objetos del escenario.

En aplicaciones de intersección, *K-Vecinos* es el mejor modelo. *Random Forest* no consigue los mismos resultados ya que entra de nuevo en overfitting, pero en este caso las condiciones no se ajustan con la misma facilidad que en el escenario de autovía. Al ser un escenario tan ruidoso el de intersección, el voto por el número de vecinos es la mejor estrategia.

TABLA I

RESULTADOS DE LOS MODELOS SELECCIONADOS DE AUTOVÍA

Modelo	Train	Test	Válido
GaussianNB	80,77%	81,38%	No
Random Forest	99,96%	92,73%	Sí
SVM	90,96%	91,22%	Sí
K-Vecinos	92,61%	92,17%	Sí

TABLA II

RESULTADOS DE LOS MODELOS SELECCIONADOS DE INTERSECCIÓN

Modelo	Train	Test	Válido
GaussianNB	67,54%	67,33%	No
Random Forest	99,97%	68,02%	No
SVM	74,88%	74,29%	Sí
K-Vecinos	89,72%	84,60%	Sí

VII. CONCLUSIONES

Siendo el objetivo del trabajo la creación de un filtro inteligente en tiempo real, que pueda ser utilizado en soluciones al alcance de la sociedad, los resultados del mismo muestran que todavía es complicada su implementación.

Estableciéndose como primer problema la memoria y capacidad de procesamiento que puede disponer un sistema portable, el uso de las redes neuronales no es una solución viable. Por ello, la búsqueda se basa en los modelos clásicos de clasificación.

De los modelos propuestos, únicamente tres (*Random Forest*, *SVM* y *K-Vecinos*) presentan resultados que podrían indicar un modelo aceptable para su utilización. No obstante, los mismos problemas de memoria y procesamiento eliminan al modelo de *SVM* para ambos escenarios.

No es posible la creación de un filtro de *clutter* general para todos los escenarios, ya que cada uno presenta unas características únicas regidas por el entorno del sensor y los objetos que se encuentran en el mismo.

REFERENCIAS

- [1] Arab, H., Ghaffari, I., Chioukh, L., Tatu, S., & Dufour, S. (2021). Machine learning based object classification and identification scheme using an embedded millimeter-wave radar sensor. *Sensors*. doi:10.3390/s21134291
- [2] Bouvier, C., Martinet, L., Favier, G., Sedano, H., & Artaud, M. (1995). Radar clutter classification using autoregressive modelling, K-distribution and neural network. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. doi:10.1109/ICASSP.1995.480091
- [3] Dube, R., Hahn, M., Schutz, M., Dickmann, J., & Gingras, D. (2014). Detection of parked vehicles from a radar based occupancy grid. *IEEE Intelligent Vehicles Symposium, Proceedings*, (págs. 1415-1420). doi:10.1109/IVS.2014.6856568

- [4] Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/CVPR.2012.6248074
- [5] Giebenhain, S., Wang, L., Anklam, C., & Goldluecke, B. (2021). Radar Ghost Target Detection via Multimodal Transformers. *IEE*
- [6] Keat, C. T., Pradalier, C., & Laugier, C. (2005). Vehicle detection and car park mapping using laser scanner. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (págs. 2054-2060). doi:10.1109/IROS.2005.1545233
- [7] Naidu, V., Gopalratnam, G., & Raol, J. (August de 2003). Evaluation of Data Association and Fusion Algorithms for Tracking in the Presence of Measurement Loss., (pág. 5733). doi:10.2514/6.2003-573
- [8] Palffy, A., Dong, J., Kooij, J. F., & Gavrila, D. M. (2020). CNN Based Road User Detection Using the 3D Radar Cube. *IEEE Robotics and Automation Letters*, 1263-1270. doi:10.48550/arXiv.2004.12165
- [9] Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., & Yang, B. (2019). Deep learning-based object classification on automotive radar spectra. *IEEE Radar Conference (RadarConf)*. doi:10.1109/RADAR.2019.8835775
- [10] Scheiner, N., Appenrodt, N., Dickmann, J., & Sick, B. (2018). Radar-based Feature Design and Multiclass Classification for Road User Recognition. *Proceedings of IEEE Intelligent Vehicles Symposium*, 779-786. doi:10.1109/IVS.2018.8500607
- [11] Zhou, Y., Liu, L., Zhao, H., López-Benítez, M., Yu, L., & Yue, Y. (2022). Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors*, 22(11), 4208. doi:10.3390/s22114