

PROYECTO FIN DE GRADO

TÍTULO: Sistema Inteligente de Predicción de Contramedidas para Drones

AUTOR: Juan Camilo Zuñiga Flor

GRADO: Ingeniería Electrónica de Comunicaciones

TUTOR/A: Jesús Rodríguez Molina

DEPARTAMENTO: Departamento de Telemática y Electrónica (DTE)

VºBº TUTOR/A

Miembros del tribunal de calificación:

PRESIDENTE: Francisco Javier Ortega González

TUTOR: Jesús Rodríguez Molina

SECRETARIO: Mario San Emeterio de la Parte

Fecha de lectura: 30-09-2025

Calificación:

El secretario,



La finalización de este TFG durante estos últimos meses supone el reflejo de todo el sacrificio que he realizado desde el momento que comencé el grado. Durante este camino he tenido la fortuna de conocer excelentes personas, tanto los compañeros con los que he podido compartir largas horas de trabajo, como los profesores que me han preparado y ayudado de cara a mi etapa profesional. La mayoría de nosotros los estudiantes, tenemos un pensamiento continuo sobre la finalización de la titulación y no entramos a valorar todo el recorrido que se realiza para llegar hasta este momento.

Para comenzar, me gustaría agradecer a mi padre Jeison y a mi madre Rosa, por la constante preocupación y el constante apoyo que me han ofrecido para lograr la titulación. Tengo claro que la persona que soy hoy en día es debido a ellos. Me han enseñado a tener un sacrificio por las cosas, desarrollando una gran capacidad de trabajo para cumplir cualquier tarea, y a ser capaz de sobreponerme ante las adversidades que he tenido a lo largo de estos años. También hago conmemoración a la frase de mi padre "Hay tiempo para todo", y a que no existe la frase "No tengo tiempo", nosotros somos quienes ponemos nuestros propios límites, y me siento orgulloso de haber disfrutado esta etapa realizando mi deporte favorito y actividades laborables.

También quería agradecer a mi grupo de amigos, desde que comencé la carrera aprendieron a dejarme mi espacio cuando tenía que estar concentrado con los exámenes, y estuvieron motivándome y preocupándose por cómo me iba con las calificaciones. Nunca me dejaron de lado cuando estaba desaparecido debido a los estudios y disfrutamos cuando se podía disfrutar, me mostraban su alegría cada vez que iba superando semestres y veían el final cada vez más cerca, eso me reconfortaba.

Por último, quiero agradecer a todas las personas que me han acompañado en este camino, a todo mi entorno que me ha proporcionado enfoques distintos de los problemas que me han surgido, a todos los compañeros que me han ayudado a crecer intelectualmente y a desarrollar mi comunicación y a colaborar por un mismo objetivo, y a mis tutores del TFG por su paciencia y dedicación.

Gracias a todos.

Resumen

Este Proyecto Final de Grado está encaminado a ofrecer un sistema inteligente basado en Machine Learning que se combina con una interfaz gráfica para indicar que método de mitigación es el más apropiado para un dron determinado, partiendo de los datos que son capaces de proporcionar los sensores que colaboran en la detección de un dron, con el objetivo de reducir costes, y rapidez en la toma de decisiones.

Para cumplir este propósito, nosotros hemos propuesto un modelo de Machine Learning que utiliza un algoritmo de decisión basado en árbol proporcionándole una base de datos donde se reúnen una ristra de información, esta, es debida a drones que han sido detectados. La historia de este algoritmo, como toma decisiones, los principales conceptos y cómo es su rendimiento nos ayudará a comprender su forma de operar, y el buen comportamiento que tiene manejando datos.

Seguidamente, debemos estudiar y decidir cuál es la contramedida más adecuada para cada dron según sus características y parámetros de vuelo, esta decisión la tomaremos para poder entrenar el modelo de Machine Learning y que nos proporcione la solución final a través de una interfaz gráfica. Esta parte en su conjunto representa la parte central de nuestro sistema, de manera que la solución proporcionada sea la más efectiva y la vez la más económica.

Para llevar a cabo esta tarea, debemos justificar la elección de un modelo por encima de la existencia de los otros, realizando una comprobación teórica y analizando los resultados de precisión que es capaz de proporcionar el modelo. Analizaremos la estructura del modelo basado en el árbol de decisiones y entenderemos los conceptos clave que utiliza para conseguir la respuesta automatizada una vez lo tenemos entrenando con una cantidad de datos. Por otro lado, describiremos las herramientas de programación que nos ayudarán a elaborar el sistema, deberemos adquirir conocimientos sobre bases de datos, sobre diseño de interfaces gráficas y sobre modelos de Machine Learning. Además, hemos diseñado la interfaz gráfica, de manera que el usuario tenga uso sencillo y robusto de la aplicación.

Una vez que tenemos una idea global a nivel teórico del sistema y como se van a relacionar la base de datos con Machine Learning y la interfaz gráfica, el siguiente paso que se trata en el proyecto es el sector al que se quiere dar servicio, en nuestro caso, apuntamos al sector de defensa aérea. Numerosos modelos de drones existentes en el mercado son explicados en los primeros apartados, también incluimos el análisis de los procedimientos de mitigación que tenemos en cuenta en el sistema, y explicamos como trabajan, para poder dar la solución óptima. Por consiguiente, realizaremos un análisis de los resultados obtenidos mostrando la eficacia y precisión del sistema.

El objetivo fundamental del desarrollo de este proyecto de investigación es la familiarización con el entorno de Machine Learning, una de las ramas que derivan de la Inteligencia Artificial.

Abstract

This Final Degree Project is aimed at offering an intelligent system based on Machine Learning, combined with a graphical interface, to indicate the most appropriate mitigation method for a specific drone. This is based on data provided by sensors involved in drone detection, with the goal of reducing costs and speeding up decision-making.

To achieve this goal, we have proposed a Machine Learning model that uses a tree-based decision algorithm, supported by a database that gathers a wide range of information obtained from previously detected drones. The development history of this algorithm, how it makes decisions, the main concepts involved, and its performance will help us understand how it operates and how effectively it handles data.

Next, we must study and determine the most suitable countermeasure for each drone, based on its characteristics and flight parameters. This decision will allow us to train the Machine Learning model so it can provide the final solution through a graphical interface. This stage represents the core of our system, ensuring that the proposed solution is both the most effective and the most cost-efficient.

To carry out this task, we must justify the choice of one model over others by conducting theoretical validation and analyzing the accuracy results that the model can deliver. We will examine the structure of the tree-based model and understand the key concepts it uses to generate automated responses once trained with sufficient data. In addition, we will describe the programming tools that will help us develop the system, and we will need to gain knowledge in databases, graphical interface design, and Machine Learning models. Moreover, we have designed the graphical interface in such a way that it provides users with a simple yet robust application experience.

Once we have a global theoretical understanding of the system and how the database, Machine Learning, and graphical interface will interact, the next step addressed in the project is the target sector. In our case, we are focusing on the air defense sector. Several drone models currently on the market are explained in the initial sections. We also include an analysis of the mitigation procedures considered by the system, explaining how they work in order to provide the optimal solution. Consequently, we will carry out an analysis of the results obtained, showcasing the system's effectiveness and precision.

The primary goal of developing this research project is to become familiar with the Machine Learning environment, one of the branches derived from Artificial Intelligence.

INDICE

Resumen	5
Abstract.....	7
Índice de figuras.....	12
Índice de tablas	14
Lista de acrónimos	15
1 Introducción.....	17
1.1 Motivación.....	17
1.2 Objetivos del proyecto.....	18
1.3 Estructura del proyecto	18
2 Estado del Arte.....	20
2.1 Definición de dron/Historia.....	20
2.2 Identificación de drones [4]	21
2.3 Clasificación de drones [6][7]	22
2.4 ¿Qué es un sistema de detección de drones? [24]	31
2.4.1 Tecnologías Tradicionales de Detección de Drones	32
2.4.2 Tecnologías Avanzadas de Detección de Drones	33
2.4.3 Tecnologías futuras [32]	34
2.5 Amenazas provocadas por drones [33][34]	34
2.6 Ataques de mitigación/Contra medidas [35][36].....	36
2.6.1 Contra medidas Electrónicas	36
2.6.2 Contra medidas Cinemáticas	36
2.6.3 Contra medidas/Mitigaciones	36
2.7 Introducción al Machine Learning [38]	39
2.7.1 Definición Machine Learning.....	39
2.7.2 Algoritmos de Decisiones	39
2.8 Base de Datos: Agrupación de la información [39].....	41
2.8.1 ¿Qué es SQLITE?	41
2.9 Entorno de desarrollo	42
2.9.1 Visual Studio Code [40].....	42
2.9.2 Python [41]	42
3 Planteamiento del problema	44
3.1 Problema y objetivos de diseño	44
3.2 Especificaciones de diseño.....	44
3.3 Limitaciones.....	45
4 Solución técnica	46

4.1 Implementación Base de Datos.....	46
4.1.1 Creación BBDD.....	46
4.1.2 Insertar datos.....	48
4.2 Desarrollo del Machine Learning	49
4.2.1 Selección del Algoritmo empleado.....	49
4.2.2 Importación de los datos.....	50
4.2.3 Entrenamiento Machine Learning.....	51
4.2.4 Obtención del Modelo.....	53
4.2.5 Predicciones y Evaluación del Modelo	53
4.2.6 Representación.....	55
4.3 Diseño e implementación de la Interfaz gráfica.....	57
4.3.1 Uso del paquete Tkinter	57
4.3.2 Título de la ventana/Nombre de la aplicación	57
4.3.3 Hora del sistema	58
4.3.4 Diseño de la tabla	58
4.3.5 Funciones de actualización.....	62
4.3.6 Aparición de la contramedida como resultado	63
4.4 Resultado.....	64
5. Presupuesto	66
5.1 Presupuesto hardware	66
5.2 Presupuesto software	66
5.3 Presupuesto mano de obra	66
5.4 Presupuesto total	67
6. Conclusiones y líneas futuras de estudio.....	67
Citas y Referencias.....	69
Anexo 1.....	75
Anexo 2.....	77

Índice de figuras

Figura 1. Dron Parrot Disco [8].	23
Figura 2. DJI MAVIC 4 [9].	23
Figura 3. UGV AH-30 [10].	23
Figura 4. Chasing M2 ROV Submarino [11].	24
Figura 5. Motor Brushed vs Motor Brushless [15].	27
Figura 6. Dron Xiaomi Yi Erida [17].	28
Figura 7. DJI MAVIC PRO [18].	28
Figura 8. Dron Yuneec Typhoon [19].	28
Figura 9. Dron Profesional OT-1450 [20].	29
Figura 10. Dron Coaxial Multi-rotor MX860 [21].	29
Figura 11. Descripción funcionamiento Radar Pasivo vs Activo [27].	32
Figura 12. Descripción funcionamiento sensor RF [30].	33
Figura 13. Shaded-136 [37].	38
Figura 14. Logo de PYTHON [42].	43
Figura 15. Conexión a la BBDD.	46
Figura 16. Método que crea una BBDD.	46
Figura 17. Diagrama de flujo para decisión de contramedida.	48
Figura 18. Método executemany() para insertar datos en la BBDD.	49
Figura 19. Creación del modelo del Machine Learning.	49
Figura 20. Función read_sql_query().	51
Figura 21. Codificación de variables categóricas.	51
Figura 22. Selección de los datos.	52
Figura 23. División de datos para entrenamiento y test.	52
Figura 24. Método fit().	53
Figura 25. Guardado del modelo.	53
Figura 26. Método predict().	53
Figura 27. Función score().	53
Figura 28. Precisión del Modelo.	54
Figura 29. Precisión vs 'max_depth'.	54
Figura 30. Precisión vs 'random_state'.	55
Figura 31. Módulos para representación del Algoritmo.	55
Figura 32. Representación Algoritmo.	55
Figura 33. Algoritmo basado en árbol de decisiones.	56
Figura 34. Librería Tkinter.	57
Figura 35. Código para creación del HMI.	58
Figura 36. Objeto para la Hora del sistema.	58
Figura 37. Datos de detección de drones.	59
Figura 38. Obtención valor del % de amenaza.	59
Figura 39. Terminal: Datos de detección de drones actualizado.	60
Figura 40. Creación objeto Style() y uso del tema 'clam'.	60
Figura 41. Estilo para el encabezado de las columnas.	60
Figura 42. Estilo para las celdas.	61
Figura 43. Creación objeto TreeView().	61
Figura 44. Inserción de encabezados en TreeView.	61
Figura 45. Creación de tags para las filas del TreeView.	61
Figura 46. Inserción de datos en TreeView.	62

Figura 47. Función parpadear_fila().....	62
Figura 48. Función actualizar_reloj().	62
Figura 49. Función mainloop().	62
Figura 50. Llamada a mostrar_contramedida().....	63
Figura 51. Variable para las columnas que espera el modelo.	63
Figura 52. Carga del modelo ML.	63
Figura 53. Función mostrar_contramedida().....	63
Figura 54. HMI por defecto.....	64
Figura 55. HMI parpadeo primera fila.....	64
Figura 56. HMI ejemplo contramedida recomendada.....	65
Figura 57. Datos extraídos BBDD 1.	75
Figura 58. Datos extraídos BBDD 2.	75
Figura 59. Datos extraídos BBDD 3.	76
Figura 60. Datos extraídos BBDD 4.	76
Figura 61. Datos extraídos BBDD 5.	76
Figura 62. Datos seleccionados para ML 1.....	77
Figura 63. Datos seleccionados para ML 2.....	78
Figura 64. Datos seleccionados para ML 3.....	78
Figura 65. Datos seleccionados para ML 4.....	79
Figura 66. Datos seleccionados para ML 5.....	79

Índice de tablas

Tabla 1. Tabla de clasificación por normativa [23].	31
Tabla 2. Obtención del parámetro Amenaza en %.	47
Tabla 3. Codificación contramedida.	52
Tabla 4. Comparación de resultados.	65
Tabla 5. Presupuesto hardware	66
Tabla 6. Presupuesto software	66
Tabla 7. Presupuesto Mano de obra.....	67
Tabla 8. Presupuesto total	67

Lista de acrónimos

Acrónimo	Definición
TFG	Trabajo Fin de Grado
PFG	Proyecto Fin de Grado
AESA	Agencia Estatal de Seguridad Aerea
UAV	Unmaned Aerial Vehicle
UAS	Unmaned Aerial System
RPA	Remotely Piloted Aircraft
RPAS	Remotely Piloted Aircraft System
UGV	Unmaned Ground Vehicle
AUV	Autonomous Underwater Vehicle
FPV	First Person View
VANT	Vehículo Aéreo No Tripulado
UE	Unión Europea
EASA	Agencia Europea de Seguridad Aérea
FPV	Vista en Primera Persona
ISR	Información, Seguridad y Reconocimiento
DoS	Denial of Service
GPS	Sistema de Posicionamiento Global
RF	Radio Frecuencia
MTOM	Masa Máxima al Despegue
RTH	Return To Home
IF	Infrarrojos
RF	Radio Frecuencia
PEM	Pulso Electro-Magnético
MHP	Microwave High Power
HMI	Human Machine Interface
SQL	Structured Query Language
BBDD	Base de Datos
CC	Corriente Continua

1 Introducción

1.1 Motivación

Hoy en día, nos encontramos en una época de tecnologías emergentes, con numerosos cambios y avances a todos los niveles en este sector. En lo que se refiere a la tecnología de un dron, cada vez se desarrollan modelos con más sistemas integrados, con el objetivo de mejorar funcionalidades. Sectores como las fuerzas militares tienen los modelos más avanzados del mercado, para brindar seguridad, pero es precisamente esto lo que provoca que a su vez se desarrollen sistemas que sean capaces de impactar en estos equipos aéreos, para el caso de que actúen de una manera no permitida o éticamente incorrecta. Se puede poner en peligro una sociedad, e incluso una persona, atentando contra su integridad y su privacidad. Por tanto, a la vez que avancen las tecnologías integradas en los drones, deben avanzar los sistemas de detección y mitigación de estos.

El concepto más importante de este trabajo es el uso del Machine Learning, de manera que podamos indicar en un área determinada de vigilancia la mejor manera de aplicar contramedidas hacia un determinado dron que sea considerado una amenaza. Durante todo el proyecto final de grado investigaremos la teoría básica del funcionamiento del modelo de Machine Learning, así como las diversas categorías de clasificación de drones, los sensores que contribuyen a su detección y las contramedidas que se pueden aplicar.

Con el uso de este modelo veremos como de efectivo y de preciso es, partiendo de una base de datos que recogerá drones que han sido estudiados para que se realice la predicción de la contramedida a aplicar. Esta contramedida se decidirá en función de las características de vuelo del dron que estarán almacenadas en la base de datos, con esta información no solo se entrenará el modelo del Machine Learning, sino que, además, servirá para testear la precisión del modelo.

Tal y como hemos adelantado en esta lectura, nosotros veremos cómo trabaja el sistema basado en un modelo de Machine Learning como complemento a un sistema de detección de drones, para finalmente, llegar a la solución de la predicción de una contramedida determinada. Ante cualquier dron detectado se especifica la contramedida necesaria suponiendo, si se ha considerado que representa una amenaza, ya sea de mayor o menor nivel. Conjuntamente, el sistema se basa en el desarrollo del modelo trabajando con una base de datos y con la interfaz gráfica de la aplicación, después de haber estudiado los principios teóricos que tiene la toma de decisiones del modelo para determinar cómo encamina sus decisiones para proporcionar una predicción. El objetivo final es familiarizarnos con este tipo de sistemas basados en IA, ya que se trata de una tecnología puntera que nos va a acompañar en toda nuestra etapa profesional.

Además, tal como he comentado antes respecto a la base de datos, se reúne información de una serie de características de vuelo, en algunos casos también el modelo del dron; esto lo proporcionan los sensores electrónicos que participan en un sistema general de detección de

drones. Estos valores son simulados, por la no disponibilidad de estos tipos de sistemas de detección. Pero lo realmente importante es el estudio de como elegir la contramedida más adecuada según el porcentaje de riesgo o amenaza que supone el dron, esa elección que realizamos supone el entrenamiento del modelo de Machine Learning.

Finalmente, se constituye un servicio basado en entregar a los clientes - sector de vigilancia y seguridad aérea - un producto que cumpla con efectividad y el menor coste posible las operaciones de mitigación sobre un dron, para facilitar el proceso de actuación. Se considera que este sistema proporcionará a un gran número de clientes una gran ayuda, ahorrando tiempo de respuesta ante situaciones de emergencia, reduciendo costes y aumentando la protección del territorio.

1.2 Objetivos del proyecto

El objetivo principal es crear un sistema que, para un dron detectado en un área controlada por un sistema de detección, indique al usuario cuál es la mejor contramedida para neutralizar a un dron según sus características de vuelo. Éstas se eligen entre: la velocidad, la altura, la posición, el porcentaje de amenaza, así como si se trata de un dron comercial o si usa un sistema de navegación satelital. Este sistema va a estar instalado en el aérea determinada que se requiere garantizar la vigilancia y la seguridad de la zona y trabajará junto con un sistema de detección formado por un conjunto de sensores electrónicos que serán los que proporcionen los datos para la base de datos del sistema. Se presentan los siguientes objetivos:

- Analizar qué datos pueden ser obtenidos de un sistema de detección de drones compuesto por sensores electrónicos.
- Analizar las características de los drones y las contramedidas que se pueden aplicar.
- Establecer la predicción de la mitigación sobre el dron y estudiar la precisión de la respuesta automatizada.

1.3 Estructura del proyecto

La estructura del proyecto está planteada según los objetivos del apartado anterior. A continuación, detallamos los capítulos en los que se divide este trabajo:

- **'2 Estado del Arte'**, en este apartado se explican los conceptos básicos relacionados con la temática del trabajo. También se analizan los diferentes sensores electrónicos que contribuyen a la detección de un dron, los diferentes tipos de drones existentes y las diferentes medidas de mitigación que se puedan aplicar sobre un dron. Respecto al software se presentarán; la base de datos utilizada, el modelo de Machine Learning elegido y la interfaz gráfica usada para el sistema.
- En cuanto al capítulo **'3 Planteamiento del problema'**, definiremos los requisitos objetivo del problema y reflejaremos las limitaciones que acortan el proyecto.
- En el capítulo llamado **'4 Solución técnica'**, expondremos los detalles de nuestro trabajo para la predicción de la contramedida a aplicar a un dron, a partir de los datos incluidos en la base de datos, y como es el proceso de entrenamiento de esta predicción.
- En último lugar, culminaremos este trabajo analizando los resultados obtenidos, las conclusiones y presentando el presupuesto del proyecto.

2 Estado del Arte

2.1 Definición de dron/Historia

Para aclarar y definir el concepto de dron con precisión, se deben entender las siglas básicas que los hacen referencia a la hora de buscar información sobre ellos para de esta manera saber distinguir las terminologías y no provocar confusiones. Estas son: “UAV” (Unmanned Aerial Vehicle), “UAS” (Unmanned Aerial System), “RPA”/“RPAs” (Remotely Piloted Aircraft) y “RPAS” (Remotely Piloted Aircraft System) [1].

UAV, por sus siglas en inglés “*Unmanned Aerial Vehicle*”, se denomina a cualquier vehículo aéreo no tripulado por un ser humano. Debido a este término tan general, se usa muy frecuentemente para hacer referencia a un dron. Reúne todo tipo de drones, desde los que son dirigidos por un controlador humano hasta los controlados por un programa informático.

UAS, por sus siglas en inglés “*Unmanned Aerial System*”, distinguido del término anterior. Consiste en un sistema de aeronave no tripulada reúne una aeronave no tripulada y todo el equipo necesario para un funcionamiento seguro y eficiente. De manera que un UAV es una parte de un UAS.

RPA, por sus siglas en inglés “*Remotely Piloted Aircraft*”. Es el término que se utiliza para designar una aeronave no tripulada, dirigida a distancia desde una estación de pilotaje remoto. Correspondería a un tipo de UAV, en ciertas ocasiones, cuando se hablan de estas naves generalmente, nos referimos como RPAs y no se debe confundir con el último término RPAS.

RPAS, por sus siglas en inglés “*Remotely Piloted Aircraft System*”. Es similar a la diferencia entre UAV y UAS, en este caso, el término RPAS se refiere a todo el sistema que hace necesario para poner en funcionamiento las aeronaves (enlace de comunicaciones, estación en tierra etc.).

En cuanto a la historia de los drones, su origen se remonta al siglo XIX, donde se produjeron avances significativos en el sector de la aviación. En esta época destacó la figura del británico Sir George Cayley que investigó los principios de vuelo y estableció fundamentos en la aerodinámica. Posteriormente ya en el siglo XX, otro nombre que cabe mencionar, es el del ingeniero Abraham E. Karem quién desarrolló uno de los primeros drones llamado ‘Albatross’. Las primeras apariciones significativas se produjeron durante la Primera Guerra Mundial, aunque no eran controlados remotamente fue un hecho importante. Siguió los desarrollos de investigación sobre el tema de control remoto para su completo desarrollo y utilización en operaciones militares. El dron desarrollado por el ingeniero Abraham significó el cimiento sobre la fabricación de drones militares. A partir de los 2000 se empezaron a desarrollar los drones que comúnmente conocemos actualmente, que son los civiles y comerciales [2].

Siguiendo el camino para entender completamente el concepto, La palabra dron (drone en inglés) hace referencia a un vehículo aéreo no tripulado (VANT). Con la entrada en vigor del decreto el 31 de diciembre del 2020 [3], los drones pasaron de nombrarse RPAS, a llamarse UAS. En definitiva, muchas variantes para referirse a una misma terminología, una aeronave sin tripulación que es capaz de mantener un nivel de vuelo controlado y propulsada por un motor de explosión, eléctrico o de reacción.

2.2 Identificación de drones [4]

Otro de los temas a tratar de vital importancia para este sector, es el de conocer en qué consiste la identificación de los drones, para ello se van a explicar principales normativas y recomendaciones que se deben llevar a cabo sobre este asunto. Se comenzará explicando qué es la placa identificativa de un dron, también conocida como la matrícula del dron. Este concepto se refiere al elemento que permite identificar la aeronave y al operador de forma unívoca, la regulación no indica una manera concreta de la colocación que debe tener, por lo que el operador tiene cierta libertad a la hora de ubicarla, pudiendo ser vinilos, pegatinas QR o placas.

Para tener consciencia sobre la obligatoriedad de una placa identificativa en un dron, primero se debe entender que varía dependiendo del tipo o concepto de la operación y la legislación, estas actividades se definen como operaciones EASA y operaciones NO EASA, esto es, las actividades que están bajo la jurisdicción del reglamento de la EASA y que cumplen con sus normas y regulaciones. Seguidamente, se debe tener en cuenta que actualmente se sigue el reglamento de la Unión Europea 2019/947 [5], el cual indica que los operadores deben incluir su número de registro en las aeronaves no tripuladas que cumplan las condiciones establecidas en la normativa UE. Por lo que no es necesario que lleven placas ignífugas como se ejercía anteriormente, pero sí obligatorio que se incluya el número de registro de operador, el cuál reúne datos como DNI del usuario y modelo del dron. Por otro lado, si es obligatorio el uso de la placa ignífuga para operaciones NO EASA, entre las que se encuentran: actividades por servicios militares, control fronterizo, vigilancia.

Generalmente, los drones que operan bajo operaciones EASA que por tanto deben llevar el número de registro del operador son:

1. Aeronaves no tripuladas utilizadas en la categoría abierta, esta categoría corresponde a operaciones de bajo riesgo, como actividades recreativas o comerciales, pertenecientes a las categorías A1, A2 y A3 donde se tienen en cuenta restricciones de distancia a personas o zonas urbanas (en la [Tabla 1](#). Tabla de clasificación por normativa se encuentran detalladas estas categorías):
 - Con una MTOM de 250 gr o superior, o que puedan transferir una energía cinética superior a 80 Julios a un ser humano en caso de colisión.

- Las equipadas con un sensor capaz de capturar datos personales, salvo que sea conforme con la Directiva 2009/48/CE, esta se refiere a una directiva dirigida a garantizar la seguridad de los niños.
2. Las aeronaves no tripuladas utilizadas en la categoría específica independientemente de su masa. Esta categoría corresponde a una etapa intermedia de regulación de drones, la cual requiere una mayor supervisión y gestión de riesgo que la categoría abierta, pero no la certificación exigida por las operaciones de mayor riesgo. Como ejemplos tenemos vuelos comerciales, vuelos de investigación, vuelos conjuntamente a otros drones.

Como conclusión de este apartado, se recalca que ante operaciones NO EASA la placa identificativa es un elemento esencial que ante los reglamentos de la UE se debe incluir el número de registro del operador. La ubicación de esta, aunque no se precisa la zona, debemos asegurarnos de que sea visible y accesible.

2.3 Clasificación de drones [6][7]

En este apartado se tratarán los tipos de drones y su clasificación. No existe una clasificación fija, hay diversas clasificaciones que se establecen según factores o características que poseen estos sistemas. Se comienza estableciendo una clasificación según el medio en el que los drones desarrollan su actividad:

- **Drones Aéreos o Voladores**

Son aquellos que establecen su movimiento a través del aire, a su vez, se pueden dividir en dos categorías dependiendo de cómo sea su tipo de ala: ala fija o ala rotatoria.

- **Drones de Ala fija**

Son aquellos que tienen la forma aerodinámica como la de un avión, suelen empezar su vuelo al lanzarlos con la mano. Esta estructura provoca una limitación de diseño, que les impide realizar un vuelo estacionario, por lo que no pueden permanecer quietos en un punto.



Figura 1. Dron Parrot Disco [8].

- Drones de Ala Rotatoria

Estos son de los drones más conocidos, también llamados multi-rotores. Este diseño si permite un vuelo estacionario, lo que permite obtener una imagen fija y estable gracias a sus motores instalados en las alas.



Figura 2. DJI MAVIC 4 [9].

- **Drones Terrestres: Unmanned Ground Vehicles, UGV**

Aquellos que realizan su movimiento sobre una superficie. Actualmente tienen un uso comercial, a pesar de que en sus inicios fueron utilizados para transportar suministros y materiales a zonas de difícil acceso. La aplicación más clara y extendida de este tipo es el ocio, donde encontramos los tan populares coches radiocontrol. Por otro lado, se están desarrollando sistemas de este tipo en el sector de defensa, como suministro militar, a continuación, se muestra un ejemplo de uno de ellos con origen en China:



Figura 3. UGV AH-30 [10].

- **Drones Submarinos: Autonomous Underwater Vehicles, AUV**

Como su propio nombre indica, este tipo de dron es el que establece su movimiento en el entorno acuático. Está dotado de un diseño hidrodinámico que le permite realizar trabajos hasta cierta profundidad. Estos drones son excepcionales para la exploración del fondo marino y resultan muy útiles en el sector de salvamento.

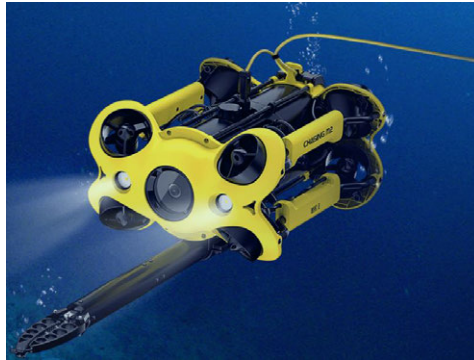


Figura 4. Chasing M2 ROV Submarino [11].

Como se ha podido observar, las aplicaciones que tienen los drones son altamente extensas. A continuación, se explicarán los tipos de drones en función de sus principales usos, así como las características básicas de dichas aeronaves, esta es la clasificación más utilizada por el sector:

- **Drones para Principiantes**

El primer tipo que va a ser tratado, como su propio nombre indica, está pensado para aquellas personas que están comenzando a pilotar drones. Pertenecen a los equipos más económicos que cuentan con una buena resistencia a los golpes. Además, en su gran mayoría poseen un tipo de motores que simplifican su manejo y potencia a niveles de aprendizaje. Con ellos, se puede obtener experiencia de vuelo, con el objetivo de tener más seguridad y control sobre el dron cuando este se encuentre a una determinada distancia y/o altura.

- **Drones con Cámara**

Sin lugar a duda, la fotografía y el vídeo representan el uso más extendido en el mundo de los drones. Desde realizar una simple foto para redes sociales, como para capturar escenografías para películas. Existen numerosas variantes de este tipo de drones con cámara, desde los más económicos con cámaras Wifi que se pueden visualizar en el teléfono móvil, hasta drones profesionales de grabación cinematográfica, pasando por las cámaras FPV (“Vista en Primera Persona”) utilizadas en los drones de carreras, que permiten. La máxima resolución que se puede encontrar hoy en día en el mercado es la de un vídeo con la incorporación de la nueva tecnología 8K y una resolución de cámara de hasta 48MP [12].

- **Drones de Carreras**

Este tipo de drones representa el entretenimiento y la competencia a través de la velocidad. Estos suelen tener un diseño pequeño, en su mayoría de fibra de carbono, y con motores de alta potencia que llegan a alcanzar hasta los 200 km/h. En esta actividad, se requiere un alto nivel de pilotaje y experiencia para controlarlos satisfactoriamente. Una de las características de estos drones, es la de que es necesario instalar una cámara de retransmisión en tiempo real llamada cámara FPV, que permita transmitir la imagen a través de unas gafas que poseen una pantalla digital, similares a las de realidad virtual. Como curiosidad, se presenta el evento más conocido en este sector: Drone Racing League en Estados Unidos, la liga de los drones de carreras [13]. La velocidad máxima registrada en estos tipos de drones es de 263.1 km/h logrado por dron RacerX en esta competición [14].

- **Drones Profesionales**

Este tipo de drones están destinados a realizar trabajos de todo tipo, como los más habituales de fotografía y vídeo. Otros trabajos profesionales se corresponden con: la revisión de instalaciones mediante cámaras térmicas instaladas, los destinados a otro tipo de sectores como el transporte de mercancía, como el sector de la agricultura o el de vigilancia. La gran mayoría de estos drones están diseñados específicamente en base a la actividad laboral para la que están destinados. Se les implementa un chasis específico, como es el caso, por ejemplo, del uso de un tanque, permite administrar líquido si es para uso agrícola, llevar un flotador en caso de realizar actividades de salvamento, o transportar polvo extintor para incendios entre muchos otros usos.

Presentan unos tipos de motores que generan la potencia suficiente para levantar el tipo de carga asociado al trabajo que se quiere realizar. En cuanto a las dimensiones, pueden llegar a superar los 2 metros de diámetro, en cuanto sus velocidades, suelen rondar una velocidad máxima de 20 m/s.

Estos tipos de aplicaciones descritas hasta el momento, se podrían reunir y denominar a estos equipos como drones comerciales, que son de acceso al público. Resulta sencilla su detección, ya que incluso podemos obtener información tanto de su modelo como de su tierra de control. La última actividad que va a ser presentada es la del uso en el sector militar, estos drones emplean tecnologías mucho más avanzadas que impiden en muchos sus casos detecciones simples.

- **Drones Militares**

La descripción de drones militares es bastante extensa. Se va a poner el foco en sus principales usos y tipos de operaciones que realizan. Hay drones fabricados desde la industria militar que suelen ser los drones más caros, implementados para operaciones militares de alto riesgo que proporcionan herramientas del tipo estratégico o letal. En consideración a las misiones que realizan estos drones, se encuentran las de

reconocimiento local, de operaciones rápidas encubiertas, vigilancia prolongada y tareas más avanzadas como producir daños en objetivos.

Por otro lado, se encuentran adaptaciones de drones comerciales a los que se les incorporan o modifican sus parámetros de diseño y su configuración con fines militares. Como en el caso anterior, son productos que una persona no perteneciente a esta área no podría acceder. Algunas de estas modificaciones consisten en aumentar la autonomía de vuelo del dron, cambios del firmware para incorporar funciones como regreso automático, altitud máxima extendida, ocultación de identificación para sistemas de detección o desactivación del módulo GNSS. Además, se suelen incorporar sistemas jamming para evitar spoofing, y sistemas que mejoran el fuego de artillería, conceptos que explicaremos en el apartado [2.6 Ataques de mitigación/Contra medidas](#)

En tercer lugar, se hará una clasificación de drones según su método de propulsión, es decir, según la manera en la que se le suministra energía para que los drones puedan producir el movimiento. Para ello se empezará definiendo previamente dos tipos de motores: el Brushed (con escobillas) y el Brushless (sin escobillas), para posteriormente realizar esta clasificación [16].

- **Motores de dron Brushed**

También conocidos como motores con escobillas. Este tipo de motores son los que están presentes en la mayoría de los drones de ocio, están alimentados con corriente continua y regulan la velocidad de giro del motor a través de una resistencia variable. Representan los motores más económicos, pero requieren más mantenimiento que un motor Brushless debido a la fricción producida por las escobillas. La potencia proporcionada por estos es inferior a la que proporciona el otro tipo de motor, además la mayoría de estos motores no están en contacto con la hélice directamente.

Su diseño consta de un motor que está unido a una pequeña rueda dentada, la cual hará girar otra rueda dentada más grande y esta, a su vez, moverá el motor. A este mecanismo se le denomina reductor, ya que estamos reduciendo la potencia máxima proporcionada por el motor. Como aspecto positivo encontramos que este mecanismo alargará la vida útil del motor porque no lo hará trabajar en su máximo esfuerzo y además ayudará a un manejo más suave del dron.

- **Motores de dron Brushless**

La característica principal de estos motores es la no utilización de escobillas que hagan contacto con el rotor, por tanto, no existe esta fricción. Esto hace que su mantenimiento sea mínimo y que su funcionamiento sea muy eficiente. En comparación, la potencia de estos motores es muy superior a la de un motor Brushed, que es capaz de proporcionar unas velocidades entre 50-70km/h, frente a los 150-200km/h de un motor Brushless, guardando la característica de estar alimentado con CC.

Su diseño consta de un sensor de efecto Hall que mide la variación del campo magnético

producido dentro del rotor, este se comunica con el controlador electrónico, que servirá para regular la velocidad del motor regulando el paso de la corriente hacia las bobinas, a este controlador electrónico se le denomina ESC (Electronic Speed Controller). Además, se considera un motor silencioso y válido para ambientes inflamables ya que emite menos energía calorífica.

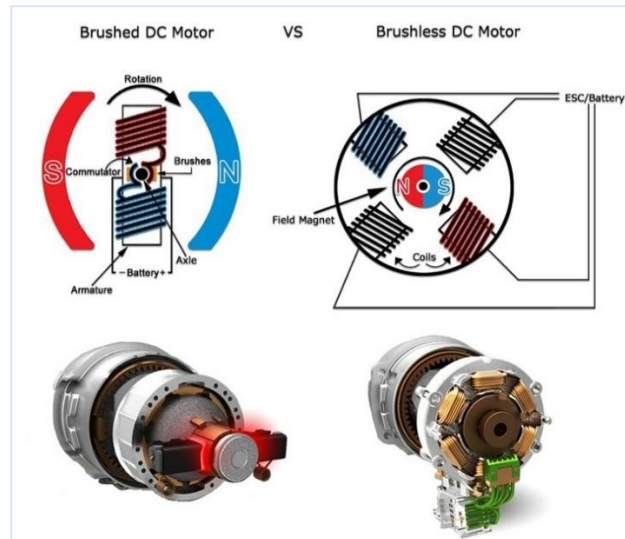


Figura 5. Motor Brushed vs Motor Brushless [15].

A continuación, se lista la clasificación correspondiente:

- Alimentados por baterías eléctricas.
- Alimentados por gasolina.
- Drones híbridos, que tal como se entiende usan tanto baterías como gasolina. Estos drones híbridos aumentan las prestaciones en comparación de los drones eléctricos. Los drones híbridos pueden mantenerse en el aire con más autonomía, alcanzar sobre unas velocidades de 80-90 km/h y transportar una carga útil de mayor masa.

En el siguiente paso, se pretende analizar los tipos de dron según su número de hélices. Esta es otra forma de clasificación según el número de brazos que presenta la aeronave. Cada uno de estos brazos lleva instalado su propio motor:

- **Tricóptero o dron de tres hélices**

Está formado por 3 brazos, el diseño característico de este dron tiene forma de T o Y. La composición consta de dos brazos delanteros que contienen un motor que gira en sentido opuesto en cada uno de ellos para generar más potencia, y un brazo trasero que cumple la función de servomotor para aportar estabilidad.



Figura 6. Dron Xiaomi Yi Erida [17].

- **Cuadricóptero o dron de cuatro hélices**

Estos drones tienen 4 brazos y son el tipo de dron más extendido en el mercado. Debido a su diseño en cruz, presentan 4 motores para cada una de las 4 hélices, de manera que se configuran de forma equidistante dos a dos, y así se logra estabilizar el dron por completo.



Figura 7. DJI MAVIC PRO [18].

- **Hexacóptero o dron de seis hélices**

Estos tipos de dron cuentan con 6 brazos y 6 motores, son los más usados en el ámbito profesional ya que cuentan con una gran estabilidad, por ello suelen llevar cámaras integradas para tomar imágenes de alta resolución. Permite que ante una avería en un motor sea capaz de mantenerse en vuelo con los otros 5 motores y aterrizar sin ningún inconveniente. Se pueden encontrar diseños tanto con motores Brushed, como con motores Brushless.



Figura 8. Dron Yuneec Typhoon [19].

- **Octocóptero o dron de ocho hélices**

Si en las tipologías anteriores se podían encontrar dos variantes de cada dron usando motores Brushed o Brushless, en estos, generalmente, solo se van a encontrar motores sin escobillas. Son drones que cuentan con 8 brazos y 8 motores, por un lado, cuentan con más potencia que los hexacópteros, pero son más grandes y pesados.



Figura 9. Dron Profesional OT-1450 [20].

- **Drones coaxiales**

Este tipo de drones se caracterizan por tener el doble de motores que de brazos. El término coaxial, hace referencia a que cada uno de los brazos contiene dos motores, con sus correspondientes hélices. La mayoría de estos drones suelen ser tricópteros, con 6 motores y 6 hélices, o cuadricópteros, con 8 motores y 8 hélices, aunque también existen versiones con más brazos.

Estos drones fueron creados con la idea de aumentar la capacidad de carga, ya que se incrementa la potencia aumentando el número de motores. Suelen utilizarse para levantar grandes cámaras profesionales, paquetes para el transporte o cualquier otro tipo de trabajos que requiera levantar una carga pesada.



Figura 10. Dron Coaxial Multi-rotor MX860 [21].

Por último, se va a clasificar los tipos de drones según normativa. Primeramente, cabe indicar que el uso de los drones está regulado por normas y leyes, y estas cambian en función del país en el que nos encontremos. Hoy en día existen títulos de piloto oficial de drones, los cuales habilitan a volar drones según su masa máxima al despegue (MTOM). Para obtener dichos títulos, es necesario pasar un examen teórico (60 horas de teoría) y otro práctico. No obstante, desde el 29 de diciembre del 2017 [22], existe una normativa específica destinada al uso

recreativo de los drones. Dicha normativa, explica que los drones de menos de 250 gramos de MTOM podrán volar dentro de espacios urbanos y sobre aglomeraciones de personas y edificios, siempre y cuando no se superen los 20 metros de altura durante el vuelo. La clasificación es la siguiente:

- **Clase C0**

Los drones que cuentan con este marcado de clase son a los que menos requisitos técnicos se les exigen. La característica para tener en cuenta es la de que deben estar diseñados para evitar dañar a personas y deben estar alimentados con baterías.

- **Clase C1**

En este mercado de clase se exigen mayores requisitos técnicos. En primer lugar, deben estar identificados con un número de serie único. En segundo lugar, deben tener un sistema de identificación a distancia directa o de red, que les permita estar registrados durante su trayectoria. A parte de eso, se debe incluir un sistema de geo consciencia de manera que se le indique al piloto del dron alertas sobre las limitaciones del espacio aéreo. Otros requisitos son: debe estar equipado con un sistema de aviso de batería baja para el UAS y la estación de control, un sistema para finalizar el vuelo de forma segura RTH (Return To Home), además de que deben estar equipados con luces para uso nocturno y de controlabilidad.

- **Clase C2**

Las aeronaves no tripuladas con este marcado de clase conllevan los requisitos de sus clases predecesoras. Adicionalmente deben tener un enlace de datos de comunicación que proteja las funciones de mando y control, además de integrar una función de baja velocidad de 3 m/s que puede operar el piloto desde la estación de control.

- **Clase C3**

Estos drones comparten las características con los de C2, pero se diferencian principalmente por el aumento del límite de la MTOM. Estos drones, exclusivamente podrán operar en entornos de categoría abierta A3, detallada más adelante.

- **Clase C4**

Las exigencias técnicas para este mercado de clases son menores que las de sus anteriores clases, esto es debido a que se han pensado para la práctica del aeromodelismo. Este tipo de dron debe ser controlable de forma segura y no pueden tener modos de control automático, excepto para la asistencia a la estabilización del vuelo.

- **Clase C5**

Debido a las operaciones a las que van destinados, estos drones son de los que más requisitos técnicos deben cumplir, partiendo de las exigencias ya tratadas de la clase C2 y C3. Adicionalmente deben incorporar un modo de baja velocidad seleccionable que limite la velocidad a 5 m/s como máximo, tener un método de recuperación del enlace de datos, mando y control, y en caso de fallo, un sistema de aterrizaje seguro. Son válidos los drones de clase C3 que lleven instalados estos sistemas para convertirse en una clase C5.

- **Clase C6**

Los drones con clase C6 reúnen todos los requisitos de su predecesora, además, permiten que sean estructuras de ala fija, no están limitados al uso de solo alimentación eléctrica, pudiendo usar motores de combustión.

UAS		OPERACIÓN		OPERADOR
Clase	MTOM	Subcategoría	Restricciones operacionales	Registro de operador
Construcción propia o sin clase	< 250 g	A1	No se recomienda volar por encima de personas	Sólo si tiene cámara (sensor)
C0			No permitido el vuelo sobre reuniones de personas	
C1	< 900 g o una energía de impacto inferior a 80 Julios	A1	No permitido el vuelo sobre reuniones de personas	
C2	< 4 kg	A2	Prever razonablemente que no volará por encima de ninguna persona no participante. En caso de vuelo inesperado, el piloto reducirá al mínimo posible el tiempo durante el cual el dron vuele por encima de estas personas	
C3			No permitido el vuelo sobre personas no participantes	
C4			30 m de cualquier persona no participante	
C5	< 25 kg	A3	No volar cerca de personas	SI
C6			Distancia de 150 m respecto de: zonas residenciales, comerciales, industriales o recreativas	
C5	< 10 kg	STS-ES-01	Operaciones VLOS (Visual Line of Sight)	
C6			No ser de ala fija	
C5	< 10 kg	STS-ES-01	Sobre una zona terrestre controlada en entorno urbano	
C6			Incluir medios para reducir el efecto de la dinámica de impacto	
C5	< 10 kg	STS-ES-02	Operaciones BVLOS (Beyond Visual Line of Sight)	
C6			Sobre una zona terrestre controlada en entorno escasamente poblado	
C5	< 10 kg	STS-ES-02	Tener una velocidad máxima respecto al suelo en vuelo horizontal de 50 m/s	
C6			Tener un sistema independiente de apagado de emergencia	

Tabla 1. Tabla de clasificación por normativa [23].

2.4 ¿Qué es un sistema de detección de drones? [24]

Para hablar sobre los sistemas de detección de drones, primero se debe comprender la necesidad del desarrollo de este tipo de sistemas. En la época actual, los drones se han convertido en una herramienta considerable para diferentes industrias, derivando en un gran

desafío en términos de seguridad. Detectar drones se ha convertido en una necesidad crítica tanto para la seguridad pública como para la privacidad de las personas. En este apartado se describirán los métodos más comunes y avanzados para detectar drones en el espacio aéreo. La detección de drones no resulta sencilla, a medida que estos UAV se han vuelto más sofisticados tecnológicamente, las tecnologías para detectarlos han ido desarrollándose progresivamente, combinando una serie de sensores electrónicos con el objetivo de crear un sistema que sea efectivo:

2.4.1 Tecnologías Tradicionales de Detección de Drones

- **Sensor Radar [25][26]**

Los radares representan uno de los métodos más generalizados. Utilizan señales electromagnéticas para detectar y monitorear a tiempo real la ubicación, la velocidad y la trayectoria de los objetos. El funcionamiento consiste en emitir ondas electromagnéticas que rebotan en los objetos y regresan a la estación de radar. Los drones, siendo objetos que contienen materiales metálicos o semimetálicos, reflejan estas ondas, lo que permite localizarlos, siendo válida esta localización para cualquier situación ambiental. No obstante, se pueden tener dificultades para detectar drones de pequeño tamaño, especialmente si vuelan a baja altitud.

El funcionamiento del radar descrito anteriormente corresponde a un radar activo, existe otro tipo de radar que se denomina pasivo. Este último detecta la posición de los drones a partir de un funcionamiento diferente; este radar no emite señales, si no, que analiza la detección de señales reflejadas por objetos de señales ambientales como las producidas por antenas de radio, comunicaciones móviles, satélites. Esto permite que el sistema radar sea difícilmente localizable y neutralizable por parte del adversario, lo que lo convierte en un gran avance en el sector de defensa.



Figura 11. Descripción funcionamiento Radar Pasivo vs Activo [27].

- **Sensor Óptico [28]**

Los sensores ópticos utilizan cámaras de alta resolución para observar la trayectoria de los drones. Con el uso de tecnología avanzada, se pueden identificar los drones a larga distancia identificando de qué tipo de dron se trata. Sin embargo, la eficacia de este

método queda limitada por la visibilidad, que se ve afectada por las condiciones climáticas y requieren un cierto nivel de luz para trabajar de forma efectiva, hasta 2km en condiciones normales día y noche y hasta 4km en las mejores condiciones posibles.

- **Sensor Acústico [29]**

Los sensores acústicos son dispositivos electrónicos que detectan los sonidos emitidos por los drones mientras vuelan. Estos sensores son muy útiles en áreas donde el radar y las cámaras tienen limitaciones. Su limitación queda establecida por los ambientes ruidosos, ya que depende de la firma acústica del dron y sus motores, pero analizan las frecuencias sonoras características de los motores de los drones para detectar su presencia comparándolas con bibliotecas de sonidos integradas en su memoria de almacenamiento.

2.4.2 Tecnologías Avanzadas de Detección de Drones

Con el continuo avance de este sector, la industria ha desarrollado métodos más avanzados para detectar y neutralizar drones de manera eficaz. Algunas de estas tecnologías son:

- **Sensor de Radio Frecuencia**

Los drones generalmente dependen de señales de radiofrecuencia para la comunicación con sus operadores. Los radares de RF interceptan estas señales y analizan el tráfico de comunicación para identificar al dron. Además, en estas señales se pueden obtener datos de comando y control a través de los protocolos de comunicación usados como Wifi, Bluetooth, o información de telemetría como el estado de la batería, posición y velocidad. Estos sistemas son altamente efectivos y permiten localizar drones incluso sin necesidad de una línea de visión directa.

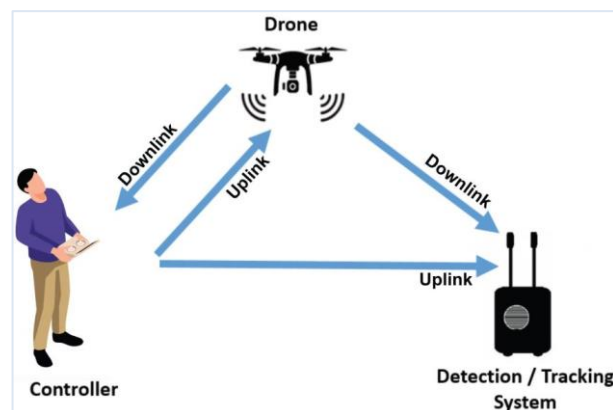


Figura 12. Descripción funcionamiento sensor RF [30].

- **Sensor de Infrarrojos (IF) [31]**

Consiste en un sistema formado por un emisor y un receptor. El emisor transmite un haz de luz del espectro no visible para el ser humano. Por otro lado, el receptor recibe el haz de luz reflejado en el objeto. Este haz de luz de espectro no visible es la luz IR que corresponde a una longitud de onda superior a 700 nm. El sensor es efectivo para condiciones climáticas adversas y para drones de pequeño tamaño, en cualquier momento del día.

- **Inteligencia Artificial (IA) y Análisis de Datos**

La inteligencia artificial ha supuesto un gran avance en la detección de drones. El uso de los algoritmos avanzados de IA procesa grandes cantidades de datos provenientes de sensores ópticos, acústicos, de RF, de IR y radares con el objetivo de identificar patrones y distinguir a los drones de otros objetos. Esto mejora la precisión de la detección, al ser un sistema compacto formado por todos los elementos de los que se procesa la información.

Dependiendo de las especificaciones sobre el área que se desea vigilar, se combinan unas tecnologías u otras, formando sistemas de sensores integrados. Por otro lado, de la misma manera que se desarrollan sistemas para detección de drones, se diseñan drones para que no sean identificados durante sus operativas, que provocan el enfrentamiento de varios desafíos. Varios ejemplos de estos desafíos son: los drones diseñados con pequeño tamaño y velocidad alta, aquellos que integran tecnologías de evasión que provocan interferencias de señales en los sistemas de detección, o los fabricados con materiales específicos para no reflejar determinadas señales.

2.4.3 Tecnologías futuras [32]

El campo de la detección de drones está en constante evolución. Con los avances tecnológicos, se están implementando soluciones más sofisticadas para mejorar la seguridad en el espacio aéreo. Una de ellas es la del uso de drones para detectar otros drones, equipados con cámaras y sensores que identifican a los drones objetivo. En segundo lugar, los organismos internacionales están trabajando juntos para establecer protocolos y normas globales para la detección y el control de drones en el espacio aéreo, lo que incluye el uso compartido de datos de radar y tecnología de inteligencia artificial. Hoy en día existe un compromiso con el desarrollo de tecnologías de detección con el objetivo de garantizar la seguridad en el espacio aéreo.

2.5 Amenazas provocadas por drones [33][34]

Generalmente, los usos más habituales con drones derivan en su uso recreativo, sin embargo,

profesionales especializados en el sector lo que verdaderamente contemplan es la amenaza que estos drones pueden representar en las organizaciones, infraestructuras críticas o en las personas, a partir de usos maliciosos. Tal y como se ha comentado sobre el desarrollo de las tecnologías instaladas en los drones, provoca que sus aplicaciones también se vayan desarrollando y transformando. Si se piensa desde la perspectiva del atacante, donde principalmente se busca eludir las organizaciones de seguridad o penetrar en ellas, la defensa y seguridad de un territorio se convierte en un gran inconveniente. En el mundo actual, es una obviedad que estos usos maliciosos se están extendiendo considerablemente y representan una gran amenaza mundial.

A continuación, comentaremos diversos tipos de actividades ilícitas en las que se pueden utilizar los drones:

- **ISR, Información, Seguridad y Reconocimiento**

Consiste en la recopilación de información, vigilancia y reconocimiento. Habitualmente representa la amenaza más producida ya que permite recopilar información crítica. Como ejemplos de información crítica, encontramos: los horarios de entrega de materias primas, los cambios de turno de los empleados, comportamientos del departamento de seguridad, los vehículos, etc. Se tiene como objetivo, planificar mejor los ataques y ejecutarlos de una manera más eficaz. De igual modo, pueden recopilar datos de la arquitectura de la red de una organización si son equipados de dispositivos como Raspberry Pi o similares.

- **DoS, Denial of Service**

Esta amenaza corresponde con el ataque por denegación del servicio, esto significa, los ataques que provocan una interrupción o parada de servicios de un sistema operativo. A través de perturbaciones e interferencias con señales físicas o a través de la red, se tiene como objetivo bloquear comunicaciones radio o bloquear datos a través de conexiones inalámbricas (Wifi, Bluetooth) u otros dispositivos.

- **Tráfico ilícito**

Esta amenaza consiste en un contrabando en el sector del transporte de mercancías como armas, drogas, dinero en lugares como la cárcel, zonas fronterizas u otro lugar.

- **Ataque de Dron**

Nos referimos a la amenaza que representa un dron cuando su misión es la de disparar, o explotar al colisionar con un objetivo. Actualmente vemos este tipo de amenazas reflejado en las guerras, perteneciente al ámbito militar. Supone un gran desafío a la hora de defenderse ante este tipo de ataques, por ello el desarrollo de mecanismos de mitigación y de contramedidas es fundamental para reducir al dron que represente una amenaza.

2.6 Ataques de mitigación/Contramedidas [35][36]

Con el objetivo de prevenir, reducir y corregir las amenazas producidas por los drones se desarrollan los sistemas de contramedidas. Estas amenazas corresponden a las descritas en el apartado anterior [2.5 Amenazas provocadas por drones \[33\]\[34\]](#), donde están incluidas todas las actividades no autorizadas, que representan aquellas operaciones que ejecuta un dron en una zona sin autorización y/o permiso. Estos ataques causan numerosos problemas a la aviación pública, comunitaria y civil, y por ello este tipo de sistemas juegan un papel fundamental, cuyas funcionalidades deben incluir el control obligatorio sobre drones, el corte de la comunicación entre el dron y su control remoto, la obligación al dron a regresar o a aterrizar, o incluso la destrucción. Se pueden establecer dos tipos de clasificaciones: contramedidas fijas o móviles, y contramedidas electrónicas o cinemáticas.

2.6.1 Contramedidas Electrónicas

En este tipo de contramedidas el objetivo consiste en interrumpir el canal de control del dron. Para esta funcionalidad, por un lado, si se trata de un dron comercial, se toma conciencia de que la mayoría de estos drones operan con frecuencias entre 2,4 GHz y 5,8 GHz. Por otro lado, si el control se realiza de forma satelital, se procede a bloquear el sistema de posicionamiento, como el GPS o Beidou (Sistema de navegación por satélite de China).

2.6.2 Contramedidas Cinemáticas

Estas contramedidas suponen los sistemas de última defensa, es decir, aquellos que se usan como última opción para neutralizar al dron inhabilitándolo o derribándolo en caso de emergencia ya que es el que más recursos económicos utiliza. Concretamente, esta técnica provoca que el dron quede fuera de funcionamiento al accionar un método de intervención física, como pueden ser los proyectiles, que pueden variar su tamaño, coste y capacidad según cada tipo de dron al que van destinados. En su mayoría estas soluciones necesitan punto de vista al objetivo, lo que muchas veces no es posible por edificios altos, obstáculos... Por consiguiente, el derribe del dron o el impacto de los proyectiles que colisionan en un lugar incorrecto pueden ocasionar daños a personas o a infraestructuras.

2.6.3 Contramedidas/Mitigaciones

Una vez se han presentado los dos tipos de contramedidas, se pasa a desarrollar cuales son los procesos que se siguen para la implementación de este tipo de sistemas. La primera etapa consiste en la detección de estas aeronaves en el espacio aéreo determinado a proteger, considerando sus parámetros de vuelo, para diferenciar los que pueden parecer una amenaza de los que no. Como segunda etapa, encontramos el análisis de estos parámetros de vuelo junto con su trayectoria para identificar anomalías y clasificar si se trata de una amenaza. Como última

etapa, se procedería a la ejecución de la mitigación disponible y/o más apropiada sobre el dron calificado negativamente. Para el diseño del sistema total se toman en cuenta consideraciones legales y presupuestarias para llegar a la solución deseada. A continuación, vamos a presentar los ataques de mitigación existentes:

- **Jamming**

Este mecanismo está formado por sistemas de radiofrecuencia, consiste en emitir ráfagas de señales RF, que sobre modulen la señal del controlador y eviten de esta manera que el dron reciba instrucciones. A este tipo de sistemas se les denomina bloqueadores, posteriormente explicaremos qué tipos existen.

Corresponde con la contramedida más económica y simple de usar, ya que simplemente inhabilita durante un período de tiempo los drones que reciben esta señal. Sin embargo, el proceso es sensible a los entornos en los que se desarrolla porque puede interferir con otros sistemas de comunicación cercanos, y sobre drones con actividades autorizadas. El efecto de esta contramedida depende de la intensidad de señal de RF que emita el bloqueador en comparación con el nivel de señal que le llega al dron por parte del controlador, es decir, depende tanto de la potencia de transmisión como de la distancia al dron. Con esta mitigación no se toma el control del dron, se provoca la desconexión de su operador. Las acciones que realiza un dron cuando no tiene el control de su operador, son la de regresar a su punto de partida o la de aterrizar, esto a su vez puede suponer una amenaza si vuela sobre un espacio aéreo sensible, asimismo, cuando el efecto del jamming deje de ser efectivo, el operador del dron suele volver a adquirir el control. Vamos a definir los tipos de bloqueadores que existen:

- **Bloqueadores direccionales:** son aquellos que se sitúan apuntando a una dirección específica; esto permite un mayor alcance, ya que causan menos interferencias de señal en el área circundante. Estos sistemas transmiten continuamente, si se estuviera destinado a un conjunto de drones, por ejemplo, no sería efectivo.
- **Bloqueadores omnidireccionales:** estos sistemas producen interferencia en todas las direcciones, pero tiene menor alcance. Por otro lado, no tienen por qué estar emitiendo continuamente, ya que pueden ocasionar más problemas en sistemas de comunicaciones cercanos y drones autorizados.
- **Bloqueadores portátiles:** un operador es el encargado de utilizar estos bloqueadores, que son móviles como su nombre indica. El operador debe apuntar al objetivo, de manera que la mitigación del dron depende proporcionalmente a la potencia de emisión de la señal y al nivel de alerta del operador, teniendo en cuenta que esta señal es menor para no poner en riesgo la salud del operador. El método es efectivo para un cierto punto sensible donde se aviste el dron y el alcance del bloqueador lo permita.

El alcance medio de este tipo de sistemas ronda aproximadamente radios de 500m a 2km.

- **Spoofing**

Esta técnica consiste en la suplantación de identidad sobre los drones, implica la generación y transmisión de señales falsas GNSS (como GPS) en un área específica para engañar los sistemas de navegación satelital de los drones haciéndoles creer que está en un lugar diferente al real. Esto deriva en que se pueda controlar la ubicación que el dron percibe y, por ende, que sea posible controlar su trayectoria. Esta tecnología implica más dificultades que bloquear con interferencia las señales de radio, ya que todos los dispositivos del área pueden recibir ubicaciones falsas y tener comportamientos inadecuados, sobre todo alterar los sistemas de navegación en los vehículos, como coches, aviones, barcos o helicópteros.

- **Drones kamikaze**

Estos drones están diseñados para contener munición, son sistemas de armamento aéreo que se autodestruyen al impactar con el objetivo (en este caso implica atentar contra drones no autorizados o estaciones base militares causando daños explosivos). La efectividad con la que se pueda desenvolver este método es complicada debido al vuelo no predecible del dron objetivo, y debido al control desde tierra con el que se controla.

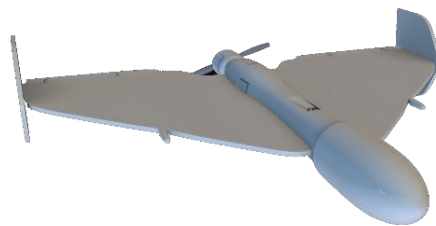


Figura 13. Shaded-136 [37].

- **Armamento/Tirador Inteligente**

Este tipo de tiradores consisten en un sistema montado en un rifle que realiza disparos precisos gracias a una mira especial que realiza el cálculo y el manejo del rifle antes del disparo, haciéndolo más preciso que usar un rifle convencional. Tienen una efectividad de como máximo 250 m de distancia, se complica ante distancias más grandes y ante drones más pequeños.

- **Láser**

Este es un dispositivo óptico que proyecta un haz de luz intenso sobre el dron, pueden destruir la estructura del dron y/o sus componentes electrónicos, quemando estructuras o componentes, como el caso de sensores y cámaras. Por otro lado, tiene como limitación la línea de vista al dron, lo que es complicado en entornos sensibles y sobre drones pequeños. La destrucción de los drones supone también un desafío por el peligro que suponen al derribarlos.

- **Pulso Electromagnético (EMP)/Microondas de alta potencia (HPM)**

Esta tecnología está compuesta de un generador (antena) que emplea pulsos de energía electromagnética de alta potencia en ráfagas cortas, que dañan los componentes electrónicos en un área determinada. Se considera una opción de último recurso, ya que puede dañar todo el equipamiento electrónico dentro de la zona aplicada.

2.7 Introducción al Machine Learning [38]

Todos somos conscientes de que la inteligencia artificial está adquiriendo un papel cada vez más importante en todas las industrias, facilitando procesos. Una de las ramas que la componen es el Machine Learning o Aprendizaje Automático, en este apartado se van a exponer en qué consiste y de qué manera se procesan los datos con los que trabaja.

2.7.1 Definición Machine Learning

Se define como un campo científico que usa una serie de algoritmos sobre conjuntos de datos en búsqueda de patrones recurrentes con el objetivo final de que estos algoritmos aprendan de forma autónoma a realizar una tarea, a hacer predicciones sobre los datos y mejorar su rendimiento con el tiempo. De ahí el nombre de Aprendizaje Automático, los algoritmos aprenden a partir de los datos y mejoran su rendimiento para realizar una tarea.

Una vez se ha comprendido este tipo de IA, se van a establecer los pasos para llevar a cabo este tipo de sistema. En primer lugar, se deben seleccionar y preparar el conjunto de datos con el que se va a alimentar al Machine Learning. Posteriormente, se debe seleccionar un algoritmo con el que ejecutar el conjunto de datos, esta selección depende del tipo y volumen de datos, y del problema que se quiera resolver. Seguidamente, se obtiene el modelo, que es el resultado del Machine Learning, por consiguiente, como último paso, sería el uso del modelo con otros datos para evaluar su rendimiento y ver posibles mejoras.

2.7.2 Algoritmos de Decisiones

Como se ha comentado anteriormente, uno de los pasos consiste en seleccionar el algoritmo con el que se va a entrenar al modelo. Como existen numeras variantes de ellos, se va a realizar una clasificación de los tipos de algoritmos existentes, y se indicarán algunos ejemplos con los algoritmos más populares y utilizados:

- **Aprendizaje Supervisado**

Este algoritmo consiste en un entrenamiento utilizando un conjunto de datos etiquetados, estos corresponden a los datos de entrada y de salida correspondiente. Contienen la información que se quiere determinar. A partir de esta asociación de los datos de entrada con los de salida, el algoritmo es capaz de realizar predicciones sobre datos no etiquetados, de manera que no requiere tantos datos de entrenamiento como los demás métodos porque se pueden comparar los resultados del modelo con los datos etiquetados.

- **Algoritmos de Regresión Lineal**

Permiten comprender las relaciones entre los datos, estos algoritmos son utilizados para predecir el valor de una variable dependiente a partir de otra independiente, como ejemplo, se predeciría número de aprobados en un curso de un alumno según sus horas de estudio.

- **Algoritmos de Regresión Logística**

Estos algoritmos se utilizan cuando las variables objetivo son binarias, es decir, dos posibles resultados. Un ejemplo de uso sería la predicción de la probabilidad de compra de un cliente a partir de su historial de compras o datos demográficos, que se corresponden con las variables de entrada al algoritmo.

- **Algoritmos de Árbol de Decisiones**

Son utilizados tanto para pronosticar valores numéricos como para clasificar datos en categorías. Consisten en una secuencia ramificada de decisiones enlazadas donde cada una de las decisiones se realiza después de que el sistema analice la variable que mejor discrimina los datos. De esta manera, el algoritmo se puede representar según un diagrama de árbol.

- **Redes Neuronales**

Consisten en imitar la interconectividad del cerebro humano para procesar los datos de entrenamiento de entrada a través de capas de nodos. Cada nodo está compuesto por entradas, ponderaciones, un umbral y un resultado, si el valor de salida supera un umbral, entonces se activa el nodo pasando los datos a la siguiente capa de red. El punto negativo de este sistema es que no es posible interpretar cuáles son los pasos que ha seguido el algoritmo para proporcionar la predicción.

- **Aprendizaje No Supervisado**

En este tipo de sistemas los datos no tienen etiquetas, a partir de estos datos la máquina explora en busca de posibles patrones para resolver los problemas de asociación de los

datos. Requiere grandes cantidades de datos y resulta más complejo, pero ayuda a identificar relaciones entre los datos que los humanos puedan pasar por alto.

- Agrupación de K-medias

Para explicar este método se debe definir previamente el término de clúster, se refiere a una variable que reúne un conjunto de datos de diferente tipo. En tal caso, esta agrupación de K-medias consiste en identificar grupos de los datos sin etiquetas en distintos clústeres cuando localiza ciertos grupos de datos que son similares entre sí.

- Aprendizaje Semi Supervisado

Consiste en un aprendizaje que combina los dos anteriores, se utiliza en el caso de no disponer de suficientes datos etiquetados. Consiste en entrenar al modelo a partir de un conjunto de datos etiquetados más pequeños y otro de datos no etiquetados más grande, de manera que estos datos etiquetados sirven como guía para clasificar los datos no etiquetados.

- Aprendizaje Reforzado

Por último, se encuentra este tipo de algoritmo que realiza un aprendizaje semejante al que los humanos realizan, esto es, a través de pruebas, consiguiendo éxito o error. Estas probabilidades de éxito y error son medidas por el algoritmo que aprende a tomar decisiones más favorables con el tiempo. Se suele utilizar en la gestión de recursos, robótica y videojuegos.

2.8 Base de Datos: Agrupación de la información [39]

Con el objetivo de manejar grandes cantidades de datos, en los que se va a apoyar el Machine Learning, es conveniente que el sistema esté compuesto de una base de datos donde se recoja la información que va a ser procesada de una manera estructurada, por ello, se presentará uno de muchos de los softwares disponibles que están destinados al manejo de bases de datos:

2.8.1 ¿Qué es SQLITE?

SQLite es una herramienta de software libre de sistema de gestión de base de datos que permite almacenar información de una forma ordenada y eficaz. Implementa el estándar SQL92 además de agregar extensiones que facilitan su uso en cualquier ambiente de desarrollo, de manera que se puedan realizar todo tipo de consultas de lenguaje SQL. Este software fue creado en el año 2000 por D. Richard Hip, el código es de dominio público, permitiendo que, gracias a numerosos

colaboradores, se haya mejorado. Como se ha comentado anteriormente, SQLite permite interactuar con un gran número de entornos de desarrollo como Java, Python, Ruby, #C, entre otros, a través de librerías.

- **SQLite3**

A través de esta herramienta, es posible establecer comunicación con SQLite, es una interfaz que permite conectarse a la base de datos y realizar consultas, añadir campos o modificaciones. Por tanto, se representa como un módulo que contiene una serie de comandos/funciones para realizar estas tareas. La extensión para la base de datos es `'.db'`.

2.9 Entorno de desarrollo

Durante este apartado vamos a presentar un entorno de programación muy popular en estos días, que corresponde con el lugar de desarrollo en el que está basada nuestra aplicación. A través de este entorno de trabajo y junto con el lenguaje de programación de Python, es realizable el manejo de la base de datos del sistema, el modelo de Machine Learning y la interfaz de gráfica de la aplicación.

2.9.1 Visual Studio Code [40]

También llamado VSCode, es una utilidad software para la edición de código fuente que es empleada ampliamente por una gran comunidad de desarrollares, fue creada por Microsoft. Esta herramienta es gratuita y multiplataforma, para que pueda ser usada en macOS, Windows o Linux. En cuanto a sus características más destacadas, se encuentran las siguientes:

- Gran cantidad de extensiones y herramientas.
- Autocompletado inteligente.
- Código abierto.
- Depuración integrada.
- Compatibilidad con Git para control de versiones.

2.9.2 Python [41]

Cuando se trata sobre Python se refiere a un lenguaje de programación, que es utilizado para desarrollo software, aplicaciones web, ciencia de datos y Machine Learning. Es un software gratuito utilizado por gran cantidad de desarrolladores porque permite que sea ejecutado en múltiples plataformas diferentes. Las características principales son:

- Lenguaje de alto nivel.
- Lenguaje orientado a los objetos.
- Sintaxis básica.
- Gran biblioteca que permite realizar multitud de funcionalidades.



Figura 14. Logo de PYTHON [42].

3 Planteamiento del problema

Durante este capítulo, se procede a plantear el problema que se pretende resolver, a partir de las descripciones técnicas del 2 Estado del Arte, para finalmente presentar la solución adoptada en el posterior 4 Solución técnica.

3.1 Problema y objetivos de diseño

Una vez que se ha dado a conocer tanto los diferentes drones que existen 2.3 Clasificación de drones [6][7], como los diversos métodos de detección que hay para identificar características sobre su trayectoria 2.4 ¿Qué es un sistema de detección de drones? [24] y las amenazas que pueden suponer en el espacio aéreo 2.5 Amenazas provocadas por drones [33][34], se plantea resolver la situación de qué medida aplicar ante la amenaza de un dron. Los objetivos son:

- Diseño y manipulación de la base de datos donde se recogen los drones detectados, en ellos se incluyen las características de vuelo e información sobre el modelo.
- Generación del modelo de Machine Learning que sea capaz de proporcionar una predicción de contramedida efectiva para mitigar al dron a partir de la selección de un algoritmo basado en un árbol de decisión.
- Diseño de una interfaz gráfica amigable con el usuario que permita visualizar como está funcionando el sistema, esto es, permite ver la detección de 10 drones y seleccionar sobre que dron se quiere obtener una predicción de la contramedida.
- Evaluación del rendimiento del sistema a través de pruebas con los datos simulados sobre drones de la base de datos que no se han utilizado para la obtención del modelo.

El sistema final se muestra a través de un HMI, que resulta una aplicación compacta donde es fundamental la comunicación con la base de datos.

3.2 Especificaciones de diseño

Las especificaciones con las que cuenta el diseño son las siguientes:

- La base de datos estará formada por un total de más de 100 drones diferentes, incluyendo la información sobre el vuelo de esa detección simulada. Parte de estos datos servirán para el entrenamiento de modelo de Machine Learning y la parte restante se usará para la comprobación de su funcionamiento y precisión.

- Los valores simulados sobre la detección de un dron que incluimos en la base de datos, será la información que es capaz de proporcionar un sistema de detección formado por los siguientes sensores electrónicos: sensor radar, sensor RF, sensor IR, sensor acústico y sensor óptico.
- El árbol de decisiones del modelo debe ser capaz de predecir las siguientes contramedidas para drones: jamming BB, jamming GNSS, spoofing, laser, PEM/MHP, dron kamikaze.
- La aplicación se desarrolla en el entorno de Visual Studio Code con lenguaje de programación Python.
- El sistema opera sin necesidad de ningún hardware especializado.

3.3 Limitaciones

El sistema cuenta con una serie de restricciones, que suponen una limitación debido a que es un entorno de simulación. Por un lado, no se dispone de ningún sistema software que nos sea capaz de proporcionar datos reales de detección de drones, y por ello, se crean datos simulados a partir de los posibles valores que proporcionan estos tipos de sistemas. Por otro lado, a la hora de predecir la contramedida a aplicar a un dron, no es posible observar de forma real si la medida que se ha tomado ha sido eficaz contra el dron. De cualquier manera, podemos evaluar la precisión del modelo de Machine Learning que es en donde está centrado el proyecto.

4 Solución técnica

Durante este capítulo, se procede a estructurar la solución adoptada en distintos apartados, de manera que podamos observar cómo se ha ido resolviendo el problema que se ha planteado en el capítulo anterior.

4.1 Implementación Base de Datos

Para la gestión de la base de datos, elegimos la interfaz proporcionada por SQLite para Python llamada sqlite3. Con ella se procede a crear la base de datos llamada “detección_drones.db”.

4.1.1 Creación BBDD

Primeramente, se debe tener una conexión a la base de datos, para ello, se crea el objeto Conexión mediante la función connect(), que realiza la conexión a la base de datos que le se le indica como parámetro, o la crea en el caso de que no exista.

```
mi_conexion = sqlite3.connect("database/deteccion_drones.db")
cursor = mi_conexion.cursor()
```

Figura 15. Conexión a la BBDD.

A continuación, en orden a ejecutar comandos SQL se debe crear un objeto Cursor mediante el método cursor(). El primer comando será para crear una tabla en la base de datos, que contiene la siguiente estructura:

```
cursor.execute('''
CREATE TABLE IF NOT EXISTS detecciones (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    velocidad REAL,
    altura REAL,
    distancia REAL,
    amenaza REAL,
    comercial TEXT,
    gps TEXT,
    modelo TEXT,
    peso REAL,
    contramedida TEXT
)
''')
```

Figura 16. Método que crea una BBDD.

Como se puede observar, se ha creado la tabla “detecciones” con 10 campos/columnas en la base de datos. Para cada columna se establece el tipo de dato e indicamos que el primer campo ‘id’ sea la clave primaria de nuestra base de datos, esto implica que identificaremos mediante un número entero cada fila y se autoincrementará su valor empezando en ‘id = 1’.

- Campo '**velocidad**': es un tipo de número real que identifica la velocidad con la que se mueve el dron en m/s. En el sistema se establece un rango de velocidades para los drones, el valor máximo será 133 m/s que es el nivel máximo registrado en el dron Peregreen v2 [43], el cuál sostiene un récord mundial.
- Campo '**altura**': este es un número real que indica la altura respecto de la superficie a la que se encuentra la aeronave en m. En el sistema se contará con una altura máxima de 200 m.
- Campo '**distancia**': este campo indica la distancia existente entre el dron y el sistema de detección en m. Supone el alcance que se tiene al dron, luego es un parámetro fundamental, ya que según la contramedida utilizada se podría hacer un cálculo del tiempo de llegada al objetivo. Como valor máximo establecido en el sistema es de 2000 m.
- Campo '**amenaza**': este campo indica el % de riesgo/amenaza provocada por el dron detectado, supone el campo que mejor discrimina las variables para la predicción. Este valor se proporcionará considerando las características del rumbo que está tomando un dron. El nivel de amenaza máximo se considerará cuando la velocidad es muy elevada y la distancia al sistema es relativamente corta, mientras que un nivel de amenaza muy pequeño será cuando se conoce el modelo de un dron de uso recreativo a una distancia lejana. La forma de generar el valor de este campo será a través de una combinación lineal, es decir, como función lineal de la velocidad y la distancia, de manera que se formarán intervalos del nivel de amenaza entre el 0 % y el 100 % en distancias de 20 %. Cada uno de estos rangos, tendrá en cuenta unos parámetros u otros, por lo que esta combinación lineal será diferente para los intervalos seleccionados, en la siguiente tabla se muestra detalladamente la obtención de este valor:

Rango de Amenaza (%)	Velocidad (v)	Distancia (d)	Función Lineal
0-20	—	✘	$\% = \frac{20}{2000} \cdot d$
21-40	—	✘	$\% = 21 + \frac{19}{2000} \cdot d$
41-60	✘	✘	$\% = 41 + \frac{19}{2000} \cdot d, 20 \frac{m}{s} \leq v < 35 \frac{m}{s}$
61-80	✘	✘	$\% = 61 + \frac{19}{2000} \cdot d, 35 \frac{m}{s} \leq v < 50 \frac{m}{s}$
81-100	✘	✘	$\% = 81 + \frac{19}{2000} \cdot d, v < 50 \frac{m}{s}$

Tabla 2. Obtención del parámetro Amenaza en %.

- Campo '**modelo**': esta columna es la identificación del modelo del dron, el sistema de detección no siempre identifica este modelo por lo que no hay problema que esté vacío en algunas detecciones. Este campo no se usa para la predicción del algoritmo, es principalmente informativo.

- Campo **'peso'**: el valor de este campo representa la masa de un dron en gramos (g) siempre y cuando se conozca el modelo del dron, es un valor obtenido a partir de su modelo, si no es desconocido. Este campo no se usa para la predicción del algoritmo, es principalmente informativo.
- Campo **'comercial'**: el valor de este campo es de tipo texto e indica si el dron pertenece a un equipo comercial, como puede ser para ocio o uso profesional. Valores: 'Sí', 'No'.
- Campo **'gps'**: el valor de este campo es de tipo texto e indica si el dron posee sistema de navegación por satélite siempre y cuando sea comercial. Valores: 'Sí', es comercial y usa navegación por satélite; 'No', no es comercial, o es comercial y no usa navegación por satélite.
- Campo **'contramedida'**: este corresponde con el campo solución de nuestro sistema, a partir de los campos de entrada se determinará cuál es la mejor contramedida que se debe aplicar. Las contramedidas que se tendrán en cuenta en el sistema son: Jamming Banda Base (Banda de Comunicación), Jamming GNSS (Sistema de Navegación por Staélite), Spoofing, Laser, PEM/MHP o Dron Kamikaze. Para seleccionar que contramedida es más eficaz en cada caso se elabora un diagrama que refleja la decisión de la mitigación:

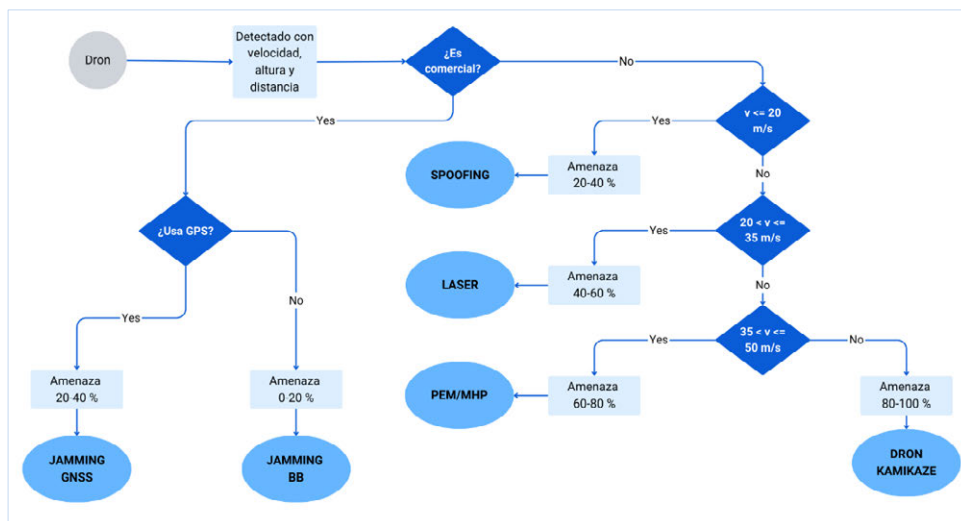


Figura 17. Diagrama de flujo para decisión de contramedida.

4.1.2 Insertar datos

Para insertar los datos en la tabla de la BBDD se debe utilizar el objeto Cursor. Se crea un objeto Lista denominado 'detecciones_data' donde cada elemento de la lista es una tupla, es decir, una agrupación de variables, que representa la información de cada dron. Esta lista contiene 149 elementos, es decir 149 tuplas que representan los drones detectados. En cada elemento

siempre se tienen los valores de la velocidad, la altura, la distancia, la amenaza, si es comercial, si usa gps y la contramedida, y en algunos casos, el modelo y el peso, cuando se trata de un dron comercial. Al final del documento, en el [Anexo 1](#), se muestran los valores de esta variable debido a su extensión.

```
cursor.executemany('''
INSERT INTO detecciones (velocidad, altura, distancia, amenaza, comercial, gps, modelo, peso, contramedida)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
''', detecciones_data)
```

Figura 18. Método `executemany()` para insertar datos en la BBDD.

4.2 Desarrollo del Machine Learning

Durante este apartado se procede a explicar el desarrollo que se ha seguido para obtener el modelo del Machine Learning.

4.2.1 Selección del Algoritmo empleado

La elección del algoritmo para el sistema es el del árbol de decisiones. Este algoritmo es eficaz para hacer predicciones precisas en estructura de datos etiquetados, como es el caso de este sistema, donde a partir de las entradas velocidad, altura, distancia, nivel de amenaza, comercial y gps; conocemos la salida, la contramedida. Este modelo tiene un buen manejo de variables no lineales, donde no existen relaciones lineales entre las variables de entrada salvo la relación del nivel de amenaza con la velocidad y la distancia. Además, permite tener un alto nivel de interpretabilidad, ya que puede ser obtenida una visualización del modelo y observar cómo se han ido ramificando las decisiones que toma algoritmo hasta llegar a la variable objetivo.

La forma en la que se implementa este modelo en Python es a través del módulo `scikit-learn` de la librería `Sklearn`, que es una distribución software para Machine Learning. El árbol es por tanto un objeto de esta librería, para instanciar esta clase `DecisionTreeClassifier()` se hace referencia a la librería con `sklearn.tree`.

```
model = DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

Figura 19. Creación del modelo del Machine Learning.

Como se puede observar, el modelo se llama 'model', lo siguiente consiste en explicar los atributos del constructor:

- `criterion = 'entropy'`

Este atributo indica la métrica utilizada para dividir los nodos del árbol, consiste en medir la calidad de una división. Encontramos dos criterios, por un lado 'Gini', se establece por defecto, mide la impureza de un nodo y es más rápido computacionalmente. Por otro lado, en cuanto al criterio que se ha utilizado 'Entropy', consiste en el uso de la entropía de la información, este concepto mide la cantidad de desorden de la información que hay en un nodo (cantidad de impureza), ya que a la hora de establecer los nodos, los datos etiquetados (salida) se van dividiendo por las ramas del árbol de manera que en cada nodo van quedando clasificadas las variables objetivo que son distintas. Por tanto, interesa que el nivel de impureza de los nodos, es decir, el nivel de entropía sea lo más bajo posible. Junto a este parámetro se encuentra otra definición que es inversamente proporcional a la entropía, llamada ganancia de la información, el árbol utiliza la ganancia de la información para tomar las mejores decisiones, a mayor ganancia de la información menor es la entropía y mejor se clasifican los datos. Esta medida de la entropía se calcula con la siguiente fórmula [44]:

$$Entropía(S) = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

- S: conjunto de datos.
- p_i : probabilidad de los elementos de la clase i en el conjunto S.
- n: número de clases posibles, en nuestro caso las diferentes contramedidas.

- **max_depth = 5**

Este atributo limita la profundidad máxima del árbol, es decir, el número de niveles de decisiones. Este valor lo puede interpretar el árbol de forma independiente, lo habitual es que sea muy profundo, pero con la necesidad de que haya muchos datos. Como nuestra cantidad de datos no es tan elevada, se realizan pruebas con niveles de profundidad bajo, entre 1 que corresponde con el valor mínimo, y 25, punto a partir del cual no se observan cambios. Es el primer parámetro que se establece para proporcionar un buen ajuste y una buena interpretabilidad del modelo. El valor ha sido seleccionado después de una serie de pruebas con distintos valores con el objetivo de obtener la mayor precisión del modelo posible; en el [4.2.5 Predicciones y Evaluación del Modelo](#), se muestra la gráfica de la variación de precisión del modelo a partir de la profundidad.

4.2.2 Importación de los datos

Para seleccionar los datos procedentes de la base de datos se realiza una consulta para extraer toda la información existente. Para el manejo de estos datos Python ofrece una librería muy útil en estos casos, llamada Pandas, que ofrece funcionalidades para leer y escribir en ficheros con formato de base de datos SQL, se realiza la importación de la librería como 'pd' para operar con ella. La librería contiene la clase DataFrame, este objeto es un conjunto de series, y un objeto Serie es similar a un array de una dimensión donde los elementos tienen que ser del mismo tipo. Por tanto, se consigue un conjunto de datos estructurado en forma de tabla donde cada columna

es una Serie, es decir, se trae la base de datos a un objeto de Python para operar con él. Este DataFrame posee dos índices, para acceder a sus elementos se puede hacer a través del índice para las columnas, y a través del índice para las filas. A partir de la función que se presenta a continuación, se está realizando una consulta SQL de manera que se selecciona todos los datos de la tabla existente y se le asigna a un objeto DataFrame:

```
df = pd.read_sql_query("SELECT * FROM detecciones", mi_conexion)
```

Figura 20. Función `read_sql_query()`.

4.2.3 Entrenamiento Machine Learning

El primer paso para el entrenamiento del modelo consiste en seleccionar los datos que van a ser necesarios, teniendo en cuenta que para que las variables sean entendidas por el modelo, deben ser de tipo numéricas, el primer paso es hacer un procesamiento de las variables comercial, gps y contramedida, para su transformación de tipo cualitativas en cuantitativas.

- **Procesamiento de variables categóricas**

Este paso consiste en la conversión de los datos categóricos: comercial, gps y contramedida, en datos cuantitativos para que pueda ser procesado por el modelo de ML. Para ello se debe utilizar la clase `LabelEncoder()` perteneciente al módulo `sklearn.preprocessing` de la librería Sklearn. Este objeto permite codificar etiquetas con valor entre 0 y `n_clases-1`, es decir, para cada etiqueta de un grupo de etiquetas le asigna un valor a cada etiqueta que sea diferente.

```
df["comercial"] = label_encoder.fit_transform(df["comercial"])
df["gps"] = label_encoder.fit_transform(df["gps"])
df["contramedida"] = label_encoder.fit_transform(df["contramedida"]) # Variable objetivo
```

Figura 21. Codificación de variables categóricas.

Una vez que la clase `LabelEncoder` ha sido instanciada con nombre `label_encoder`, se le aplica el método `fit_tranform()` a la parte del DataFrame correspondiente donde se tienen los datos, para seleccionar una Serie de nuestro DataFrame se usa `df["campo"]`. Por un lado, las variables correspondientes a comercial y gps obtienen un valor de '0' para 'No' y de '1' para 'Sí'. Por otro lado, las contramedidas Jamming BB, Jamming GNSS, Spoofing, Laser y Dron Kamikaze pasan a asignarles los siguientes valores:

Contramedida	Asignación ML
Jamming BB	1
Gamming GNSS	2
Spoofing	5

Laser	3
PEM/MHP	4
Dron Kamikaze	0

Tabla 3. Codificación contramedida.

- **Selección de los datos**

Las variables de entrada usadas para entrenar el modelo son la velocidad, la altura, la distancia, el nivel de amenaza, si es comercial y si usa gps. Por otro lado, la variable de salida es la contramedida. De esta forma una vez que se han traído los datos de la base de datos debemos crear las variables de entrada y de salida para el entrenamiento del árbol. Las demás variables son informativas por lo que no se tienen en cuenta para la generación de los datos de entrenamiento.

```
x = df[["velocidad", "altura", "distancia", "amenaza", "comercial", "gps"]]
y = df["contramedida"]
```

Figura 22. Selección de los datos.

De esta manera 'x' será un DataFrame con las variables de entrada al modelo e 'y' se corresponderá con otro DataFrame que contiene la variable objetivo. Debido a la extensión de estas variables, se muestra en el [Anexo 2](#).

- **Método de entrenamiento**

Para entrenar al modelo primero se debe usar la clase `train_test_split` del módulo `sklearn.model_selection` de la librería Sklearn. Este objeto permite dividir los datos en datos para testeo y datos para entrenamiento, indicándole la forma de cómo dividirlos mediante dos atributos que serán explicados más adelante:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
```

Figura 23. División de datos para entrenamiento y test.

El atributo 'test_size' se refiere a la parte de los datos que se van a usar el testeo, su valor va de 0 a 1. De esta manera, su valor supone el tanto por ciento de los datos que haya. En este caso, tiene un valor de 0.2, lo cual significa que el 20% de los datos se van a usar para el testeo, y por tanto, la parte restante que son el 80% de los datos, se va a usar para entrenamiento.

En cuanto al atributo 'random_state', su valor sirve para indicar como se hace la separación de los datos en entrenamiento y prueba. Al darle un valor de 2, esto implica que los datos se van a ir dividiendo hasta en 2 ocasiones para ir seleccionando datos en entrenamiento y prueba. Este valor ha sido seleccionado después de una serie de pruebas con distintos valores con el objetivo de obtener la mayor precisión del modelo posible, como se muestra en el [4.2.5 Predicciones y Evaluación del Modelo](#).

Seguidamente se procede a realizar el entrenamiento práctico utilizando el método fit() de la siguiente manera:

```
model.fit(x_train, y_train)
```

Figura 24. Método fit().

4.2.4 Obtención del Modelo

Una vez que se ha entrenado el ML, el siguiente paso es guardar el modelo entrenado en un archivo llamado 'modelo_contramedidas.pkl'. Este tipo de extensión se denominada Pickle, es la forma de serializar objetos en Python con el objetivo de poder cargar el modelo sin necesidad de volver a entrenarlo. Este proceso permite:

- Usar el modelo sin necesidad de que sea reentrenado cada vez que se ejecute el programa.
- Reutilizar el modelo ya entrenado para hacer predicciones nuevas.
- Compartir el modelo con otros usuarios.

La biblioteca de Python que permite realizar este funcionamiento es la denominada Joblib; se muestra a continuación el uso del método dump() al que se le pasa como parámetros el modelo y el archivo destino.

```
joblib.dump(model, "modelo_contramedidas.pkl")
```

Figura 25. Guardado del modelo.

4.2.5 Predicciones y Evaluación del Modelo

A partir de este momento, el modelo es completamente utilizable, de forma que los siguientes pasos consistirán en realizar pruebas y observar cómo se comporta en las predicciones:

```
y_pred = model.predict(x_test)
```

Figura 26. Método predict().

En la anterior figura se realiza una predicción sobre los datos 'x_test', para observar los datos de salida. Si se quisiera calcular la precisión del modelo en este momento se deben comparar los resultados obtenidos en 'y_pred' con 'y_test' que corresponde a los datos que deberían salir como resultado. Sin embargo, una forma más efectiva de realizar este cálculo de precisión, es la que ofrece la librería DecisionTreeClassifier con el método 'score()':

```
accuracy = model.score(x_test, y_test)
```

Figura 27. Función score().

El funcionamiento de este método consiste en obtener la precisión del modelo entre 0.00 como punto menos preciso y 1.00 como punto más preciso. Internamente se hace una llamada a la función descrita anteriormente `predict()` con el parámetro 'x_test' y compara esas predicciones con 'y_test' para obtener el resultado realizando la siguiente fórmula:

$$accuracy = \frac{n^{\circ} \text{aciertos}}{n^{\circ} \text{total de muestras}} [45]$$

Precisión del modelo: 0.90

Figura 28. Precisión del Modelo.

En este caso se obtiene una precisión máxima en el sistema de **0.90**. En primer lugar, se muestra la figura correspondiente a la variación de esta precisión a partir de la variable 'max_depth'. Como se puede observar en la [Figura 29](#), a partir del valor de 3 se obtiene una precisión por encima del 85 %. La elección del valor `max_depth = 5` da la mayor precisión obtenida posible y además permite tener una buena interpretabilidad del modelo.

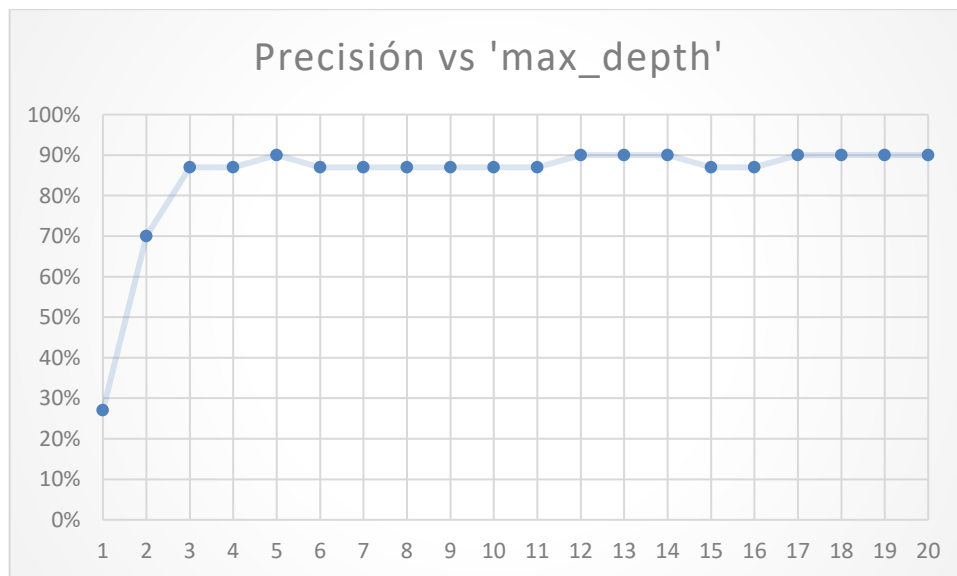


Figura 29. Precisión vs 'max_depth'.

En segundo lugar, se va a reflejar la variación de la precisión a partir del parámetro 'max_depth'. Tal y como se puede observar en la [Figura 30](#), el valor que se le da este parámetro para que realice la división de los datos en entrenamiento y test provoca una mayor fluctuación en la precisión del modelo. La elección de `random_state = 2` supone el mayor nivel de precisión que se puede obtener.

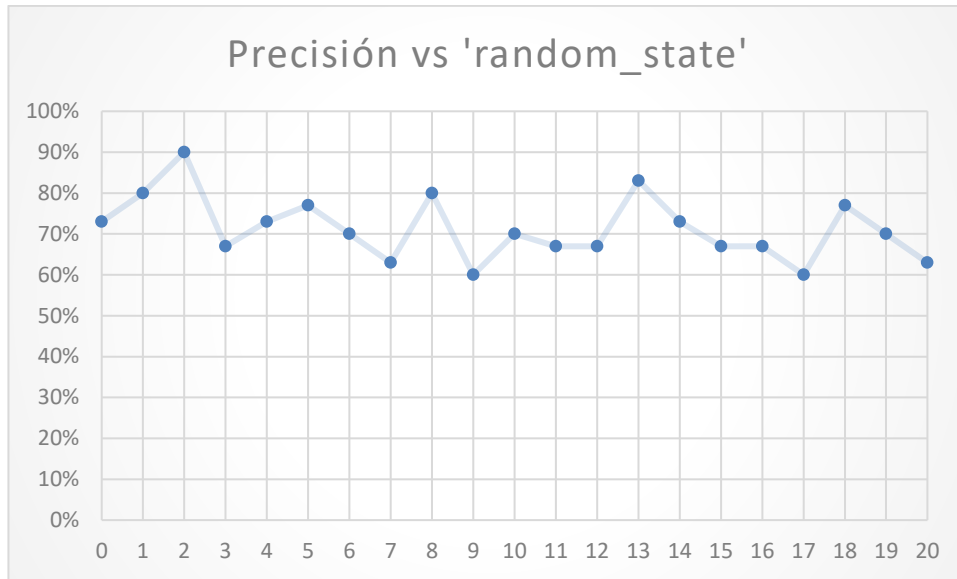


Figura 30. Precisión vs 'random_state'.

4.2.6 Representación

En este último apartado del entrenamiento se procede a observar de una manera gráfica el algoritmo de decisión del Machine Learning que se ha utilizado. Para este caso, usar el árbol de decisiones como algoritmo permite realizar un seguimiento de cómo se han ido clasificando los datos hasta obtener el resultado final. Para un apoyo en la representación del mismo se usa la librería proporcionada por Python llamada Matplotlib, donde dentro de ella se usa el módulo pyplot, junto a esta se debe combinar otro módulo proporcionado por sklearn.tree de la librería Sklearn; llamado plot_tree:

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

Figura 31. Módulos para representación del Algoritmo.

```
plt.figure()
plot_tree(decision_tree=model, feature_names=x.columns, filled=True, fontsize=6)
plt.show()
```

Figura 32. Representación Algoritmo.

Combinando ambos módulos, se crea la figura correspondiente en una ventana nueva con los parámetros introducidos en plot_tree() para ajustar las características del mismo, así como personalizar fuentes o colores. El parámetro 'model' corresponde al modelo, 'x.columns' corresponde a los nombres de las columnas de los datos, 'filled=True' proporciona los colores a los nodos de decisión y con 'fontsize=6' se establece el tamaño de la fuente. Como resultado final, en la [Figura 33](#) se representa el modelo de árbol de decisión entrenado para la predicción de contramedidas frente a drones. Este modelo clasifica una detección en una de las contramedidas disponibles a partir de las variables observadas en el dron (velocidad, altura,

distancia, porcentaje de amenaza, carácter comercial y señal GPS).

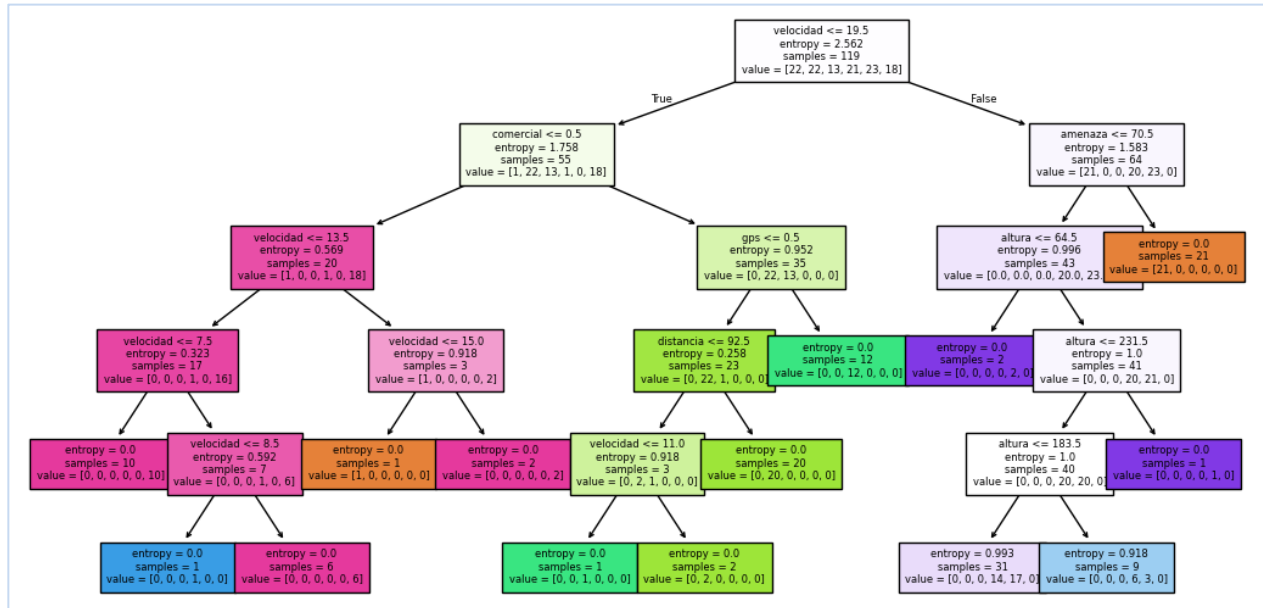


Figura 33. Algoritmo basado en árbol de decisiones.

A continuación, se procede a interpretar las decisiones que ha tomado el modelo. Cada nodo del árbol corresponde a una condición de decisión sobre una variable. Por ejemplo, el nodo raíz establece la primera división con la condición $\text{velocidad} \leq 19.5$. A partir de esta regla, los datos se bifurcan en dos ramas:

- Si la condición se cumple (rama izquierda), el modelo explora primero si el dron es de tipo comercial y posteriormente utiliza información de GPS y distancia;
- Si no se cumple (rama derecha), las variables más determinantes pasan a ser la altura y el porcentaje de amenaza.

En cada nodo se muestra la siguiente información:

- **Condición de división:** regla que separa los datos en dos grupos.
- **Entropy:** medida de impureza de las clases en ese nodo (valores bajos indican mayor certeza en la clasificación).
- **Samples:** número de muestras del conjunto de entrenamiento que alcanzaron dicho nodo.
- **Value:** distribución de las muestras por cada clase de contramedida.
- **Color del nodo:** refleja la clase mayoritaria en ese punto; cuanto más intenso es el color,

mayor es la pureza de la clasificación.

Los nodos terminales representan la predicción final del modelo. En ellos, la entropía es nula o muy baja, lo que significa que la mayoría de las muestras pertenecen a una sola clase, de manera que el modelo tiene un alto grado de certeza sobre la contramedida a aplicar.

De esta forma, el árbol de decisión permite comprender de manera intuitiva cómo el sistema inteligente establece sus predicciones. A diferencia de otros algoritmos más opacos, el árbol ofrece transparencia e interpretabilidad, al mostrar explícitamente qué variables y umbrales son considerados en cada paso. Además, permite validar si las reglas generadas por el modelo coinciden con el criterio experto, aportando un respaldo adicional para su aplicación en escenarios reales de detección y neutralización de drones.

4.3 Diseño e implementación de la Interfaz gráfica

Durante este apartado se continúa describiendo la implementación de la interfaz gráfica diseñada. La funcionalidad como se ha comentado anteriormente consiste en crear la ventana de la aplicación, en la que se implementan una serie de widgets/objetos: un nombre de ventana, el nombre de la aplicación, una tabla, la hora actual y la solución final que solo aparecerá al realizar una selección en la tabla.

4.3.1 Uso del paquete Tkinter

Para implementar el HMI se utiliza el paquete Tkinter que corresponde con una interfaz estándar que proporciona Python como un kit de herramientas para desarrollar el GUI (Guide User Interface). Se debe comprobar que el paquete está instalado con el comando `python -m tkinter`. Para disponer de los objetos que se van a necesitar se importan los módulos `tkinter.tk` y `tkinter.ttk`:

```
import tkinter as tk
from tkinter import ttk
```

Figura 34. Librería Tkinter.

4.3.2 Título de la ventana/Nombre de la aplicación

Una vez la ventana es creada con la instancia de la clase correspondiente como se podrá observar en la [Figura 35](#), elegimos el título de la ventana que aparecerá en la barra superior: “*Sistema Inteligente de Predicción de Contramedidas para Drones*”. Posteriormente se establece el tamaño de nuestra ventana en píxeles y se indica que la ventana no se pueda reajustar ni a lo

ancho ni a lo alto, dejándola fija. Además, se incorpora como nombre de la aplicación "S.I.P.C.P" en la parte alta de la ventana.

```
# Crear ventana principal
ventana = tk.Tk()
ventana.title("Sistema Inteligente de Predicción de Contramedidas para Drones")
ventana.geometry("1200x500")
ventana.resizable(False, False)

# Nombre APP
tk.Label(ventana, text=" S.I.P.C.D", font=("Arial", 14, "bold")).pack(pady=10)
```

Figura 35. Código para creación del HMI.

4.3.3 Hora del sistema

Seguidamente se establece la hora en la ventana de la aplicación; para ello se usa la librería Datetime, la cual permite tomar la hora actual del sistema donde está corriendo la aplicación, de esta forma (recoge en este caso la hora del sistema Windows donde se está ejecutando). Se crea un objeto StringVar() de la librería Tkinter para indicar que es widget tipo texto que va cambiando su valor. Posteriormente se indica en qué lugar de la ventana se desea que aparezca la hora. La función de actualización que provoca que esta hora se vaya actualizando está definida en el 4.3.5 Funciones de actualización.

```
reloj_var = tk.StringVar()
reloj_label = tk.Label(ventana, textvariable=reloj_var, font=("Arial", 10), anchor="e", justify="right")
reloj_label.pack(side="bottom", anchor="e", padx=10, pady=5)
```

Figura 36. Objeto para la Hora del sistema.

4.3.4 Diseño de la tabla

El widget con el cual el usuario va a interactuar se trata de una tabla. En la ventana se muestra una tabla con 10 registros correspondientes a drones detectados que simulan un entorno de detección en tiempo real, el usuario deberá seleccionar el dron sobre el cual quiera obtener la recomendación de la contramedida a aplicar. Para simular el entorno en tiempo real, cada vez que se realiza una acción de ver la contramedida a aplicar, se mostrará el dron que ha sido seleccionado. Los registros que se usan para esta implementación han sido creados en una variable como se mostrará a continuación. Primeramente, para realizar pruebas se usaron los registros correspondientes a los extraídos de la base de datos que a su vez se habían dividido para el entrenamiento y para el test, por lo que estos no se reflejan en esta explicación.

```
datos = [(12, 96, 368, 0, 1, 1),
        (36, 20, 132, 0, 0, 0),
        (51, 10, 925, 0, 0, 0),
        (4, 35, 717, 0, 1, 1),
        (6, 30, 1394, 0, 1, 0),
        (17, 20, 137, 0, 1, 1),
        (15, 28, 1309, 0, 1, 0),
        (22, 200, 1456, 0, 0, 0),
        (4, 10, 1134, 0, 0, 0),
        (10, 80, 141, 0, 0, 0),
        (11, 20, 48, 0, 1, 0),
        (4, 10, 1122, 0, 1, 0),
        (13, 170, 3476, 0, 0, 0),
        (23, 80, 748, 0, 0, 0),
        (7, 10, 1097, 0, 1, 1),
        (10, 80, 2135, 0, 1, 1),
        (16, 27, 513, 0, 1, 0),
        (9, 20, 77, 0, 1, 0),
        (8, 100, 329, 0, 0, 0)]
```

Figura 37. Datos de detección de drones.

El cuarto campo de cada detección corresponde con el nivel de amenaza, la estructura de estos datos es la misma que se presentó en la base de datos para el entrenamiento del Machine Learning, por lo que este valor de la amenaza se obtiene de la misma manera como está definido en la [Tabla 2](#). Obtención del parámetro Amenaza en %. del [4.1.1 Creación BBDD](#). A continuación, se muestra el código correspondiente a la obtención de este valor, así como la terminal con la actualización de los datos que van a ser utilizados en el sistema:

```
detecciones_aux = [] # Objeto lista para asignar valor
for d in datos:
    if (d[4] == 1):
        if (d[5] == 1):
            amenaza_aux = 21 + 19*d[2]/2000
        else:
            amenaza_aux = 20*d[2]/2000
    else:
        if (d[0] <= 20):
            amenaza_aux = 21 + 19*d[2]/2000
        elif (20 < d[0] <= 35):
            amenaza_aux = 41 + 19*d[2]/2000
        elif (35 < d[0] <= 50):
            amenaza_aux = 61 + 19*d[2]/2000
        else: # deteccion[0] > 50
            amenaza_aux = 81 + 19*d[2]/2000
    detecciones_aux.append((d[0], d[1], d[2], int(amenaza_aux), *d[4:]))
datos = detecciones_aux
```

Figura 38. Obtención valor del % de amenaza.

```
Datos:
(12, 96, 368, 24, 1, 1)
(36, 20, 132, 62, 0, 0)
(51, 10, 925, 89, 0, 0)
(4, 35, 717, 27, 1, 1)
(6, 30, 1394, 13, 1, 0)
(17, 20, 137, 22, 1, 1)
(15, 28, 1309, 13, 1, 0)
(22, 200, 1456, 54, 0, 0)
(4, 10, 1134, 31, 0, 0)
(10, 80, 141, 22, 0, 0)
(11, 20, 48, 0, 1, 0)
(4, 10, 1122, 11, 1, 0)
(13, 170, 3476, 54, 0, 0)
(23, 80, 748, 48, 0, 0)
(7, 10, 1097, 31, 1, 1)
(10, 80, 2135, 41, 1, 1)
(16, 27, 513, 5, 1, 0)
(9, 20, 77, 0, 1, 0)
(8, 100, 329, 24, 0, 0)
```

Figura 39. Terminal: Datos de detección de drones actualizado.

Para representar la tabla se usa la clase `TreeView` de la librería `Tkinter`. Este objeto es una colección de elementos en árbol donde cada elemento tiene una lista de valores y se representa en columnas sucesivas, además quedan identificados por un id. Esta identificación permite conocer cuál es el dron de la tabla que ha seleccionado. Además, para mayor personalización de esta tabla se utiliza el objeto `Style()` de la librería `Tkinter` que permite darle alternancia de colores a cada fila, definir que colores se van a utilizar para la tabla, el tipo de letra para las celdas y los nombres de las columnas, así como su tamaño y los márgenes. Como se muestra más abajo se usa un tema denominado 'clam' ya que permite más personalización que otros temas por defecto.

```
style = ttk.Style(ventana)
style.theme_use("clam") # Usa el tema 'clam', que permite más personalización
```

Figura 40. Creación objeto `Style()` y uso del tema 'clam'.

El encabezado de las columnas se personaliza con letra en negrita de color blanco sobre un fondo azul verdoso:

```
style.configure("Treeview.Heading", font=("Arial", 11, "bold"), background="#005f73", foreground="white")
```

Figura 41. Estilo para el encabezado de las columnas.

Para las celdas restantes se aplica su tamaño y una anchura de celda. Con el método `map()` es posible cambiar el color cuando una fila está seleccionada, además, se establece por defecto un color de fondo blanco para las celdas con bordes de gris claro con anchura de un píxel. El método `layout()` permite reajustar el objeto tabla para que su contenido ocupe todo el tamaño de la tabla 'nsw' (North, South, West, East):

```
style.configure("Treeview", font=("Arial", 10), rowheight=28)
style.map("Treeview", background=[("selected", "#0a9396")])
style.layout("Treeview", [('Treeview.treearea', {'sticky': 'nswe'})]) # para evitar márgenes raros
style.configure("Treeview", background="white", fieldbackground="white")
style.configure("Treeview", bordercolor="#ccc", borderwidth=1)
```

Figura 42. Estilo para las celdas.

Una vez se tiene configurado el estilo, se procede a crear el objeto para poder introducir datos en él. Primeramente, se define una lista que se pasa como parámetro a la tabla para indicar los encabezados de las columnas. Se establece que la longitud mostrada en tabla sea de 10 filas y como el objeto TreeView tiene una columna por defecto no es mostrada mediante el parámetro 'show=headings':

```
encabezados = ["Velocidad [m/s]", "Altura [m]", "Distancia [m]", "Amenaza [%]", "Comercial", "GPS"]
table = ttk.Treeview(ventana, columns=encabezados, show='headings', height=10)
```

Figura 43. Creación objeto TreeView().

El siguiente paso consiste en insertar estos encabezados en la tabla, además de indicar como se quiere la alineación del texto y del ancho de las columnas.

```
for i in encabezados:
    table.heading(i, text=i)
    table.column(i, anchor="center", width=165)
```

Figura 44. Inserción de encabezados en TreeView.

Se desea tener una alternancia de colores en las filas entre gris y blanco como mejora del diseño, para ello se deben crear tags asociados a las filas de la tabla, además de realizar un parpadeo en la primera fila de color rojo simulando que es último dron que ha sido detectado.

```
table.tag_configure("evenrow", background="#f2f2f2")
table.tag_configure("oddrow", background="white")
table.tag_configure("blink", background="red")
```

Figura 45. Creación de tags para las filas del TreeView.

Se procede a insertar los datos en la tabla y con ese fin, se itera con un bucle la lista con la información de los drones. Como iteradores del bucle se utilizan; 'i' que se corresponde al índice de la fila, y 'fila' que corresponde a la tupla con los elementos, esto es la causa de utilizar la clase enumerate() que además de devolver la fila con los elementos devuelve el índice correspondiente a cada fila. A cada fila dependiendo si es par o impar se le asignan los tags "evenrow" o "oddrow" para realizar la alternancia de colores, y se van insertando al final de la tabla. Por otro parte, se salvan el primer elemento de la tabla y el primer valor del tag, que provocan el parpadeo de la primera fila que se detalla en el [4.3.5 Funciones de actualización](#):

```

for i, fila in enumerate(datos):
    tag = "evenrow" if i % 2 == 0 else "oddrow"
    item_id = table.insert('', 'end', values=fila, tags=(tag,))

    if i == 0:
        primer_item = item_id
        primer_tag_original = tag

```

Figura 46. Inserción de datos en TreeView.

4.3.5 Funciones de actualización

En este apartado se describen tres funcionalidades del sistema que se ejecutan durante la ejecución de la aplicación para a) mantener la hora del sistema, b) realizar el parpadeo de la primera de las filas de la tabla y c) mantener la ventana principal. Se crea una función que realiza el parpadeo de la primera fila de la tabla con un color rojo simulando que es el último dron que ha sido detectado como medida de aviso al usuario. De esta forma, se recoge el tag asociado a esa primera fila y se compara con el nuevo tag 'blink' que servirá para aplicar el color rojo, y se devolverá al valor de su tag original en un tiempo de 600 ms de forma que se muestre el color por defecto de la fila.

```

def parpadear_fila():
    actual = table.item(primer_item, 'tags')[0]
    nuevo = "blink" if actual != "blink" else primer_tag_original
    table.item(primer_item, tags=(nuevo,))
    ventana.after(600, parpadear_fila)

```

Figura 47. Función parpadear_fila().

La función de actualización de la hora consiste en realizar un mantenimiento de la hora actual del sistema, obteniendo este valor cada 1s y aplicando un formato para una mejor representación.

```

def actualizar_reloj():
    hora = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    reloj_var.set(hora)
    ventana.after(1000, actualizar_reloj)

```

Figura 48. Función actualizar_reloj().

Por último, se describe la función clave de Tkinter para la interfaz gráfica. Esta función mantiene la ventana abierta y activa, además de lanzar el bucle principal de eventos. Provoca que la ventana se quede en escucha de acciones como realizar click en botones, seleccionar la tabla, actualizar la hora y el cierre de la ventana:

```

ventana.mainloop()

```

Figura 49. Función mainloop().

4.3.6 Aparición de la contramedida como resultado

La función que se encarga de mostrar el resultado, es decir, de mostrar la contramedida, se ejecutará cada vez que se realice una pulsación en nuestra tabla. De esta manera se produce el evento "<<TreeviewSelect>>" al seleccionar un ítem de la tabla.

```
table.bind("<<TreeviewSelect>>", mostrar_contramedida)
```

Figura 50. Llamada a mostrar_contramedida().

Una vez se ha producido el evento, se comprueba que se haya seleccionado un ítem de la tabla para posteriormente recoger los valores correspondientes a ese ítem, en este caso, se recogen los valores correspondientes a la fila que se ha seleccionado. Antes de realizar la predicción de la contramedida se debe convertir esa tupla de valores a un DataFrame, ya que es el objeto que el modelo de ML espera, y se asegura que las columnas correspondientes a esos valores que hemos obtenido son las del modelo entrenado.

```
columnas_modelo = ["velocidad", "altura", "distancia", "amenaza", "comercial", "gps"]
```

Figura 51. Variable para las columnas que espera el modelo.

Para usar el modelo se debe haber cargado previamente procedente de la ubicación dónde se produjo su guardado en el [4.2.4 Obtención del Modelo](#).

```
modelo = joblib.load("modelo_contramedidas.pkl")
```

Figura 52. Carga del modelo ML.

Y finalmente se realiza la predicción para que sea mostrada en la ventana de la aplicación:

```
def mostrar_contramedida(event):
    item = table.focus()
    if item:
        valores = table.item(item, 'values')

        # Convertir a DataFrame para pasar al modelo
        fila_df = pd.DataFrame([valores], columns=columnas_modelo)
        prediccion = modelo.predict(fila_df)[0]

        if (prediccion == 0):
            prediccion_string = "Dron Kamikaze"
        elif (prediccion == 1):
            prediccion_string = "Jamming BB"
        elif (prediccion == 2):
            prediccion_string = "Jamming GNSS"
        elif (prediccion == 3):
            prediccion_string = "Laser"
        elif (prediccion == 4):
            prediccion_string = "PEM/MHP"
        elif (prediccion == 5):
            prediccion_string = "Spoofing"

        resultado_var.set(f"La contramedida es: {prediccion_string}")
```

Figura 53. Función mostrar_contramedida().

4.4 Resultado

La solución final de la aplicación será mostrada a continuación, a través de la ejecución del programa. En ella se va a evaluar la precisión del modelo para cada una de los drones que son mostrados en pantalla.

Velocidad [m/s]	Altura [m]	Distancia [m]	Amenaza [%]	Comercial	GPS
12	96	368	24	1	1
36	20	132	62	0	0
51	10	925	89	0	0
4	35	717	27	1	1
6	30	1394	13	1	0
17	20	137	22	1	1
15	28	1309	13	1	0
22	200	1456	54	0	0
4	10	1134	31	0	0
10	80	141	22	0	0

Figura 54. HMI por defecto.

Velocidad [m/s]	Altura [m]	Distancia [m]	Amenaza [%]	Comercial	GPS
12	96	368	24	1	1
36	20	132	62	0	0
51	10	925	89	0	0
4	35	717	27	1	1
6	30	1394	13	1	0
17	20	137	22	1	1
15	28	1309	13	1	0
22	200	1456	54	0	0
4	10	1134	31	0	0
10	80	141	22	0	0

Figura 55. HMI parpadeo primera fila.

Posteriormente se muestra la contramedida automatizada al seleccionar la primera de las detecciones existentes en la tabla, como se puede observar en la [Figura 56](#), está seleccionada la primera fila de la tabla, cuya contramedida recomendada es 'Jamming GNSS':

Sistema Inteligente de Predicción de Contramedidas para Drones

S.I.P.C.D

Velocidad [m/s]	Altura [m]	Distancia [m]	Amenaza [%]	Comercial	GPS
12	96	368	24	1	1
36	20	132	62	0	0
51	10	925	89	0	0
4	35	717	27	1	1
6	30	1394	13	1	0
17	20	137	22	1	1
15	28	1309	13	1	0
22	200	1456	54	0	0
4	10	1134	31	0	0
10	80	141	22	0	0

La contramedida es: *Jamming GNSS*

2025-09-17 09:22:24

Figura 56. HMI ejemplo contramedida recomendada.

Para evaluar las contramedidas recomendadas por el Machine Learning para cada una de las detecciones de la tabla, se procede al desarrollo de la [Tabla 4](#). Estará compuesta de la contramedida automatizada y de la contramedida ideal, que realmente se espera. Tal como se puede observar para estas 10 detecciones, hemos obtenido un 100 % de precisión en nuestro modelo, lo que, comparando con la precisión obtenida en el entrenamiento del modelo de un 90 % (4.2.5 Predicciones y Evaluación del Modelo); el resultado ha sido satisfactorio.

Posición en la tabla	Contramedida recomendada	Contramedida esperada
1	Jamming GNSS	Jamming GNSS
2	PEM/MHP	PEM/MHP
3	Dron kamikaze	Dron kamikaze
4	Jamming GNSS	Jamming GNSS
5	Jamming BB	Jamming BB
6	Jamming GNSS	Jamming GNSS
7	Jamming BB	Jamming BB
8	Laser	Laser
9	Spoofing	Spoofing
10	Spoofing	Spoofing

Tabla 4. Comparación de resultados.

Se han podido reflejar las buenas características de Python para el manejo de una base de datos, del Machine Learning y de una interfaz gráfica sencilla. Esta utilización de este tipo de Inteligencia Artificial podría ser la base de un sistema más robusto el cuál se pudiera obtener una mayor precisión utilizando grandes cantidades de datos.

5. Presupuesto

En esta sección, vamos a detallar los diferentes presupuestos que son necesarios para el completo desarrollo del proyecto, donde incluimos los costes de hardware, software y de mano de obra.

5.1 Presupuesto hardware

Tanto para la redacción del proyecto como para el desarrollo el código, hemos usado nuestro ordenador personal:

Material	Unidades	Descripción	Precio/Unidad (€)	Total
Lenovo IdeaPad 14ITL6	1	Ordenador personal	799,00	799,00
Logitech M185	1	Ratón Inalámbrico	12,99	12,99

Tabla 5. Presupuesto hardware

5.2 Presupuesto software

La mayoría de los programas que hemos utilizado son de software libre, que no requieren costes de licencia, sobre estos se ha calculado un precio de coste según las horas gastadas en el desarrollo del proyecto:

Material	Version	Unidades	Precio (€)
Windows 11 Home	10.0.26100	1	145,00
Microsoft Office Hogar	2024	1	149,99
Visual Studio Code	1.101.1	1	0,00
Python	3.12.9	1	0,00

Tabla 6. Presupuesto software

5.3 Presupuesto mano de obra

En este presupuesto se tiene en cuenta el coste por mano de obra de una persona recién graduada como ingeniero de telecomunicaciones que tenga las especificaciones técnicas para el desarrollo del proyecto, esta aproximación se muestra de la siguiente manera:

Material	Precio/hora (€)	Horas (h)	Total (€)
----------	-----------------	-----------	-----------

Ingeniero de Telecomunicaciones Junior	20,00	323	6460,00	5
---	-------	-----	---------	---

Tabla 7. Presupuesto Mano de obra

5.4 Presupuesto total

El coste total del desarrollo del proyecto lo calculamos a partir de los presupuestos anteriores, en la siguiente tabla se muestra que la cantidad total que incluye todas las partidas de presupuesto es 9.156,05 € (Nueve Mil Ciento Cincuenta y Seis Euros):

Recurso	Coste
Hardware	811,99
Software	294,99
Recursos Humanos	6460,00
Total (sin IVA)	7566,98
Total (con IVA)	9156,05

Tabla 8. Presupuesto total

6. Conclusiones y líneas futuras de estudio

Como último punto, una vez realizada esta investigación, reflejaremos las conclusiones y posibles líneas futuras que puede tomar este proyecto. Podemos afirmar que los objetivos planteados en el comienzo del proyecto han sido alcanzados exitosamente. Tal y como hemos podido observar durante la realización del mismo, el uso del Machine Learning sobre una ristra de drones detectados permitió automatizar el proceso de predecir cuál es la contramedida para

Respecto al controlador/usuario del sistema como hemos comentado, le facilita las tareas, además de que en situaciones críticas donde la amenaza de un dron sea muy elevada permite que no se dependa de la velocidad de ejecución de acciones por parte del controlador, estas son: la reacción a la detección del dron que supone una amenaza, y, la capacidad para determinar la mejor contramedida. Por supuesto que este sistema puede funcionar sin apoyo humano, pero es conveniente que ante sistemas basados en IA se realice una monitorización por parte del humano.

El sistema ofrece una alta precisión, tal como hemos analizado en el apartado Resultados [7], y comprobábamos como nos daba la solución más eficiente para cada dron nuevo detectado en el sistema, por lo que es un producto altamente confiable con un HMI sencillo e intuitivo. Como líneas futuras de este proyecto, sería interesante formar un sistema completo junto con un proceso de detección de drones para identificar que drones representan una amenaza real y cuáles son drones autorizados. Por otro lado, otra opción a desarrollar sería la capaz de integrar un sistema de procesamiento de señales que sea capaz de activar de forma automática la contramedida que ha indicado el sistema previamente. Estaríamos contando con un sistema completamente autónomo de detección de drones, análisis de contramedida a aplicar, y ejecución de la contramedida.

Citas y Referencias

- [1] DEDrone, “Counter-UAS White Paper”. [En línea]. Disponible en: <https://es.dedrone.com/white-papers/counter-uas>. [Accedido: 15-feb-2025].
- [2] AviationGroup, “Drones: ¿Quién los inventó?” [En línea]. Disponible en: <https://www.aviationgroup.es/actualidad/drones-quien-invento/>. [Accedido: 15-feb-2025].
- [3] AESA, “Normativa de UAS, Drones”. [En línea]. Disponible en: <https://www.seguridadaerea.gob.es/es/ambitos/drones/normativa-de-uas-drones>. [Accedido: 15-feb-2025].
- [4] Grupo UAS, “Placa identificativa para drones.” [En línea]. Disponible en: <https://grupo-uas.com/placa-identificativa-dron/>. [Accedido: 26-jun-2025].
- [5] Reglamento de Ejecución (UE) 2019/947, *Diario Oficial de la Unión Europea*. [En línea]. Disponible en: <https://www.boe.es/doue/2019/152/L00045-00071.pdf>. [Accedido: 26-jun-2025].
- [6] Novodrone, “Tipos de drones.” [En línea]. Disponible en: <https://novodrone.com/content/13-tipos-dedrones>. [Accedido: en 15-feb-2025].
- [7] J. Gomiz, “Tipos de drones.” [En línea]. Disponible en: <https://javiergomiz.com/tipos-drones/>. [Accedido: 15-feb-2025].
- [8] E. Escribano, “El nuevo dron de Parrot permite aprender y disfrutar del aeromodelismo”, *El Español*, 21-jun-2017. [En línea]. Disponible en: https://www.elespanol.com/omicrono/tecnologia/20170621/nuevo-dron-parrot-permite-aprender-disfrutar-aeromodelismo/225478735_0.html. [Accedido: 26-jun-2025].
- [9] Space, “DJI Mavic 4 Pro Drone Review.” [En línea]. Disponible en: <https://www.space.com/technology/drones/dji-mavic-4-pro-drone-review>. [Accedido: 26-jun-2025].
- [10] Military Drones (China), “50kg Crawler Type Unmanned Ground Vehicle.” [En línea]. Disponible en: <https://es.militarydrones.org.cn/50kg-crawler-type-unmanned-ground-vehicle-p00307p1.html>. [Accedido: 26-jun-2025].
- [11] El Vuelo del Drone, “Chasing M2 ROV Submarino.” [En línea]. Disponible en: https://elvuelodeldrone.com/drones-profesionales/drones-acuaticos/chasing-m2-rov-submarino/?attribute_modelo=Chasing%20M2%20cable%20200m.%20con%20maleta%20y%20brazo%20rob%C3%B3tico. [Accedido: 26-jun-2025].
- [12] Wedio, “Encuentra las 5 mejores cámaras para drones en 2025.” [En línea]. Disponible en: <https://www.wedio.com/es/guias-de-compra/mejor-camara-drone?srsltid=AfmBOopmFKU3owFftEXhjeZ9HQ305TjvbBX-Gutps-a7GjwBchzwwmG>. [Accedido: 21-jun-2025].
- [13] Drone Racing League, “Sitio web oficial.” [En línea]. Disponible en: <https://www.drl.io/>. [Accedido: 28-jun-2025].
- [14] RPASDrones, “¿Cuál es la velocidad máxima registrada por un dron?” [En línea]. Disponible en: <https://rpas-drones.com/velocidad-maxima-registrada-por-un-dron/>. [Accedido: 30-jun-2025].

- [15] Jouav, "Heavy Lift Drone Motors Guide." [En línea]. Disponible en: <https://www.jouav.com/blog/heavy-lift-drone-motors.html>. [Accedido: 26-jun-2025].
- [16] AdvDron, "Como elegir MOTORES Brushless para DRONES FPV - Guía Completa", *YouTube*, 15-mar-2024. [En línea]. Disponible en: <https://www.youtube.com/watch?v=AijAE5Nv3Ak>. [Accedido: 26-jun-2025].
- [17] M.ua, "Xiaomi Yi Erida Drone Description." [En línea]. Disponible en: <https://m.ua/desc/xiaomi-yi-erida/>. [Accedido: 26-jun-2025].
- [18] Amazon Brasil, "DJI Mavic Pro." [En línea]. Disponible en: <https://www.amazon.com.br/DJI-CP-PT-000500-Mavic-Pro/dp/B01LZ8QTSU>. [Accedido: 26-jun-2025].
- [19] Amazon México, "Yuneec Collision Avoidance Hexacopter." [En línea]. Disponible en: <https://www.amazon.com.mx/Yuneec-Collision-Avoidance-Hexacopter-Controller/dp/B01AT4BKFU?th=1>. [Accedido: 26-jun-2025].
- [20] AeroExpo, "Multirotor UAV Frame - Gryphon Dynamics." [En línea]. Disponible en: <https://www.aeroexpo.online/es/prod/gryphon-dynamics/product-185461-80809.html>. [Accedido: 26-jun-2025].
- [21] HLD Store, "MX860 Multirotor - T-Drones." [En línea]. Disponible en: <https://www.hldstore.com/multirrotor/t-drones/mx860/>. [Accedido: 26-jun-2025].
- [22] Real Decreto 1036/2017, de 15 de diciembre, por el que se regula la utilización civil de las aeronaves pilotadas por control remoto, y se modifican los R.D. 552/2014 y 57/2002, BOE-A-2017-15721, *Boletín Oficial del Estado*, núm. 316, 29-dic-2017, pp. 129609-129641. [En línea]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-2017-15721>. [Accedido: 28-jun-2025].
- [23] Grupo UAS, "Tipos de Drones." [En línea]. Disponible en: <https://grupo-uas.com/tipos-de-drones/>. [Accedido: 27-jun-2025].
- [24] RPAS Drones, "¿Cómo detectar un dron?" [En línea]. Disponible en: <https://rpas-drones.com/como-detectar-un-dron-metodos-tecnologias/>. [Accedido: 26-jun-2025].
- [25] J. P. Zeitoun Moralez, "El uso del radar pasivo en la defensa aérea," *Pucará Defensa*, 6-sep-2024. [En línea]. Disponible en: <https://www.pucara.org/post/el-uso-del-radar-pasivo-en-la-defensa-a%C3%A9rea>. [Accedido: 26-jun-2025].
- [26] C-UAS Hub, "Radar FS pasivo con DVB-S para la detección de drones." [En línea]. Disponible en: <https://cuashub.com/es/contenido/radar-fs-pasivo-con-dvb-s-para-la-deteccion-de-drones/>. [Accedido: 26-jun-2025].
- [27] Universidad Virtual de Quilmes, "Unidad 2 - Teledetección." [En línea]. Disponible en: <https://static.uvq.edu.ar/mdm/teledeteccion/unidad-2.html>. [Accedido: 26-jun-2025].
- [28] DEDrone, "Extensiones: Cámaras PTZ." [En línea]. Disponible en: <https://es.dedrone.com/products/drone-detection/extensions/ptz-cameras>. [Accedido: 26-jun-2025].
- [29] HGH Infrared, "Sensores acústicos y UAVs." [En línea]. Disponible en: <https://hgh-infrared.com/es/deteccion-y-seguimiento-de-los-drones-y>

- [uavs/#:~:text=Los%20sensores%20ac%C3%Basticos%20y%20los,la%20detecci%C3%B3n%20de%20los%20UAVs. \[Accedido: 26-jun-2025\].](#)
- [30] C-UAS Hub, “Detección y seguimiento de drones mediante señales de identificación RF.” [En línea]. Disponible en: <https://cuashub.com/es/contenido/deteccion-y-seguimiento-de-drones-mediante-senales-de-identificacion-rf/>. [Accedido: 26-jun-2025].
- [31] HGH Infrared Systems, “SPYNEL IR Camera: A Proven Solution for Drone Swarms / Small UAVs Detection & Tracking,” nov-2020. [En línea]. Disponible en: https://hgh-infrared.com/wp-content/uploads/2020/11/HGH_CaseStudy_Spynel_DroneUAVDetection.pdf. [Accedido: 26-jun-2025].
- [32] D-Fend Solutions, “Evaluación de tecnologías antidron.” [En línea]. Disponible en: <https://d-fendsolutions.com/es/evaluacion-y-comparacion-de-las-tecnologias-de-deteccion-antidron-c-uas/>. [Accedido: 26-jun-2025].
- [33] SentryCS, “The Rise of Non-State Drone Threats.” [En línea]. Disponible en: <https://sentrycs.com/es/the-counter-drone-blog/the-rise-of-the-non-state-drone-threats/>. [Accedido: 26-jun-2025].
- [34] AGE2, “Drones y ciberseguridad.” [En línea]. Disponible en: <https://www.age2.es/noticias/drones-representan-una-amenaza-para-la-ciberseguridad/>. [Accedido: 26-jun-2025].
- [35] SeguriLatam, “Contramedidas de drones,” 18-nov-2024. [En línea]. Disponible en: https://www.segurilatam.com/tecnologias-y-servicios/drones/contramedidas-de-drones-sistemas-modelo-general-y-desarrollo_20241118.html. [Accedido: 26-jun-2025].
- [36] D-Fend Solutions, “Tecnologías de mitigación antidron.” [En línea]. Disponible en: <https://d-fendsolutions.com/es/evaluacion-y-comparacion-de-las-tecnologias-de-mitigacion-antidron-c-uas/>. [Accedido: 26-jun-2025].
- [37] A. Mitrofanov, “Palisade for Ukraine: Armas rusas de precisión de largo alcance que ya se están utilizando o podrían utilizarse potencialmente en la zona NVO,” *TopWar (Military Review)*, 05-jul-2023. [En línea]. Disponible en: <https://es.topwar.ru/220677-chastokol-dlja-ukrainy-rossijskoe-vysokotochnoe-oruzhie-bolshoj-dalnosti-kotoroe-uzhe-primenjaetsja-ili-potencialno-mozhet-byt-primeneno-v-zone-svo.html>. [Accedido: 26-jun-2025].
- [38] GitHub Pages, “IAARBook - Aprendizaje automático.” [En línea]. Disponible en: <https://iaarbook.github.io/ML/>. [Accedido: 26-jun-2025].
- [39] SQLite, “Sitio oficial.” [En línea]. Disponible en: <http://www.sqlite.org>. [Accedido: 26-jun-2025].
- [40] Arsys, “¿Qué es Visual Studio Code y cuáles son sus ventajas?” 17-jul-2025. [En línea]. Disponible en: <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>. [Accedido: 26-jun-2025].
- [41] Amazon Web Services, “¿Qué es Python?” [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/python/>. [Accedido: 26-jun-2025].

- [42] Wikipedia, "Historia de Python." [En línea]. Disponible en: https://es.wikipedia.org/wiki/Historia_de_Python. [Accedido: 26-jun-2025].
- [43] NeoTeo, "Peregreen V2: El dron más rápido del mundo," 21-abr-2024. [En línea]. Disponible en: <https://www.neoteo.com/peregreen-v2-el-dron-mas-rapido-del-mundo>. [Accedido: 26-jun-2025].
- [44] IBM Think, "¿Qué es un árbol de decisión?" 2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/decision-trees>. [Accedido: 26-jun-2025].
- [45] Scikit-Learn, "sklearn.tree.DecisionTreeClassifier." [En línea]. Disponible en: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> [Accedido: 26-jun-2025].
- [46] UAV LATAM, "Uso militar de drones: amenazas y mitigaciones." [En línea]. Disponible en: <https://uavlatam.com/uso-militar-de-drones-amenazas-y-como-mitigarlas/>. [Accedido: 26-jun-2025].
- [47] AESA, "Listado de fabricantes y modelos de UAS." [En línea]. Disponible en: <https://www.seguridadeaerea.gob.es/es/ambitos/drones/requisitos-de-uas-drones/listado-de-fabricantes-y-modelos-de-uas>. [Accedido: 26-jun-2025].
- [48] EU Drone Port, "Lista de drones certificados." [En línea]. Disponible en: <https://eudroneport.com/es/blog/lista-de-drones-certificados/>. [Accedido: 26-jun-2025].
- [49] DJI Agricultura, "Página principal." [En línea]. Disponible en: <https://ag.dji.com/es>. [Accedido: 26-jun-2025].
- [50] DEDRONE, "Software de detección de drones." [En línea]. Disponible en: <https://es.dedrone.com/products/drone-detection-software#:~:text=%C2%BFQu%C3%A9%20es%20un%20sistema%20de,aspecto%20de%20una%20soluci%C3%B3n%20integral>. [Accedido: 26-jun-2025].
- [51] NQDefense, "Sistema de detección y contramedida de drones." [En línea]. Disponible en: <https://www.nqdefense.com/es/que-es-el-sistema-de-deteccion-y-contramedida-de-drones-y-cuales-son-sus-caracteristicas/>. [Accedido: 26-jun-2025].
- [52] DEDRONE, "Detecting and Neutralizing Drone Threats." [En línea]. Disponible en: <https://es.dedrone.com/white-papers/detecting-and-neutralizing-drone-threats>. [Accedido: 26-jun-2025].
- [53] HGH Infrared, "Detección y seguimiento de drones y UAVs." [En línea]. Disponible en: <https://hgh-infrared.com/es/deteccion-y-seguimiento-de-los-drones-y-uavs/>. [Accedido: 26-jun-2025].
- [54] SeguriLatam, "Sensores infrarrojos: tipos y funciones," 27-dic-2023. [En línea]. Disponible en: https://www.segurilatam.com/actualidad/sensores-infrarrojos-tipos-funciones_20231227.html. [Accedido: 26-jun-2025].
- [55] C-UAS Hub, "Ucrania utiliza sensores acústicos para rastrear drones." [En línea]. Disponible en: <https://cuashub.com/es/contenido/ucrania-utiliza-sensores-acusticos->

[para-rastrear-la-llegada-de-drones/](#). [Accedido: 26-jun-2025].

[56] El Confidencial, “Nuevos drones e IA en Ucrania”, 12-feb-2025. [En línea]. Disponible en: https://www.elconfidencial.com/tecnologia/novaceno/2025-02-12/ucrania-nuevos-drones-inteligencia-artificial-sensores-acusticos_4063339/. [Accedido: 26-jun-2025].

[57] IPTC-UPM, “Nuevos sensores radar para protección antidron.” [En línea]. Disponible en: <https://iptc.upm.es/nuevos-sensores-radar-para-la-proteccion-antidron/>. [Accedido: 26-jun-2025].

[58] NQDefense, “Sistema de contramedidas de drones.” [En línea]. Disponible en: <https://www.nqdefense.com/es/que-es-el-sistema-de-contramedidas-de-drones/>. [Accedido: 26-jun-2025].

[59] YouTube, “Evaluación y comparación de tecnologías de mitigación contra drones (C-UAS).” [En línea]. Disponible en: <https://www.youtube.com/watch?v=ssCwawuVXyw&t=423s>. [Accedido: 26-jun-2025].

[60] PhotoSpecialist, “Sitio web oficial.” [En línea]. Disponible en: <https://www.photospecialist.es/>. [Accedido: 26-jun-2025].

[61] Potensic, “Sitio web oficial.” [En línea]. Disponible en: <https://www.potensic.com/>. [Accedido: 26-jun-2025].

[62] TYX Toys, “Sitio web oficial.” [En línea]. Disponible en: <https://www.tyxtoys.com/>. [Accedido: 26-jun-2025].

[63] AgEagle Aerial Systems, “Sitio web oficial.” [En línea]. Disponible en: <https://ageagle.com/es/>. [Accedido: 26-jun-2025].

[64] Wingtra, “Sitio web oficial.” [En línea]. Disponible en: https://wingtra.com/es/?srsltid=AfmBOogRc0Av4tJvIrMTxz4yuy-_mkGN-YGfJfnAjcJyKiQmc-lgyqVu. [Accedido: 26-jun-2025].

[65] Delair Aero, “Sitio web oficial.” [En línea]. Disponible en: <https://delair.aero/>. [Accedido: 26-jun-2025].

[66] Objectif Drone, “Sitio web oficial.” [En línea]. Disponible en: <https://www.objectifdrone.net/>. [Accedido: 26-jun-2025].

[67] Aerobotics, “Sitio web oficial.” [En línea]. Disponible en: <https://www.aerobotics.com/?locale=es>. [Accedido: 26-jun-2025].

[68] Cavok UAS, “Sitio web oficial.” [En línea]. Disponible en: <https://www.aeroexpo.online/es/prod/cavok-uas/product-187991-78365.html>. [Accedido: 26-jun-2025].

[69] Manual Pandas Python, Aprende con Alf. [En línea]. Disponible en: <https://aprendeconalf.es/docencia/python/manual/pandas/#la-clase-de-objetos-dataframe>. [Accedido: 26-jun-2025].

[70] Scikit-Learn, “sklearn.preprocessing.LabelEncoder.” [En línea]. Disponible en:

<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>.
[Accedido: 26-jun-2025].

[71] TkDocs, “Tkinter Tutorial.” [En línea]. Disponible en:
<https://tkdocs.com/tutorial/index.html>. [Accedido: 26-jun-2025].

[72] Python Docs, “tkinter.ttk.Treeview.” [En línea]. Disponible en:
<https://docs.python.org/es/3.10/library/tkinter.ttk.html#treeview>. [Accedido: 26-jun-2025].

[73] S. Raschka, *Python Machine Learning*. Packt Publishing, 2015.

[74] M. Owens, *The Definitive Guide to SQLite*. Apress, mayo 2006.

Anexo 1.

En esta sección se muestran los datos extraídos de la base de datos, para su posterior manipulación.

Mostrando DF de la BBDD

id	velocidad	altura	distancia	amenaza	comercial	gps	modelo	peso	contramedida	
0	1	5.0	80.0	220.0	2.0	Si	No	DJI MINI SE 2	249.0	Jamming BB
1	2	8.0	20.0	1400.0	14.0	Si	No	DJI MINI 3	249.0	Jamming BB
2	3	11.0	40.0	160.0	1.0	Si	No	DJI MINI 3 PRO	249.0	Jamming BB
3	4	2.0	60.0	1280.0	12.0	Si	No	DJI MINI 4 PRO	249.0	Jamming BB
4	5	12.0	85.0	200.0	2.0	Si	No	ATOM	249.0	Jamming BB
5	6	3.0	10.0	1100.0	11.0	Si	No	ATOM LT	249.0	Jamming BB
6	7	5.0	80.0	120.0	1.0	Si	No	ATOM SE	249.0	Jamming BB
7	8	3.0	200.0	1800.0	18.0	Si	No	WIPKVIEW T28	130.0	Jamming BB
8	9	4.0	50.0	150.0	1.0	Si	No	WIPKVIEW B12	249.0	Jamming BB
9	10	5.0	27.0	1750.0	17.0	Si	No	WIPKVIEW T26	138.0	Jamming BB
10	11	12.0	20.0	12.0	0.0	Si	No	DJI NEO	135.0	Jamming BB
11	12	3.0	10.0	2100.0	21.0	Si	No	DJI MAVIC 3 CLASSIC	895.0	Jamming BB
12	13	5.0	80.0	120.0	1.0	Si	No	DJI AIR 2S	595.0	Jamming BB
13	14	15.0	200.0	2000.0	20.0	Si	No	DJI AIR 3	720.0	Jamming BB
14	15	10.0	190.0	795.0	7.0	Si	No	DJI AVATA 2	377.0	Jamming BB
15	16	5.0	27.0	1628.0	16.0	Si	No	SENSEFLY EBEE X	1600.0	Jamming BB
16	17	12.0	20.0	452.0	4.0	Si	No	SENSEFLY EBEE CLASSIC	690.0	Jamming BB
17	18	16.0	80.0	1920.0	39.0	Si	Si	DJI Mavic 3 PRO	958.0	Jamming GNSS
18	19	13.0	78.0	847.0	29.0	Si	Si	DJI MATRICE M30	3998.0	Jamming GNSS
19	20	11.0	96.0	350.0	24.0	Si	Si	DJI MAVIC 3M	950.0	Jamming GNSS
20	21	10.0	35.0	722.0	27.0	Si	Si	DJI MAVIC 3T	920.0	Jamming GNSS
21	22	12.0	20.0	120.0	22.0	Si	Si	DJI MAVIC 3E	915.0	Jamming GNSS
22	23	2.0	10.0	100.0	21.0	Si	Si	EVO II v3	1191.0	Jamming GNSS
23	24	3.0	10.0	2000.0	40.0	Si	Si	ABZ INNOVATION M12	13600.0	Jamming GNSS
24	25	5.0	80.0	120.0	22.0	Si	Si	X11	1500.0	Jamming GNSS
25	26	5.0	200.0	1970.0	39.0	Si	Si	WINGTRAONE GENLL	4500.0	Jamming GNSS
26	27	10.0	186.0	65.0	0.0	Si	No	DJI MATRICE 350 RTK	9200.0	Jamming GNSS
27	28	7.0	27.0	455.0	25.0	Si	Si	DJI INSPIRE 3	4430.0	Jamming GNSS
28	29	8.0	20.0	1722.0	37.0	Si	Si	TRINITY F90+	5000.0	Jamming GNSS
29	30	3.0	10.0	1100.0	11.0	Si	No	CHRONO	3900.0	Jamming BB
30	31	5.0	80.0	1867.0	18.0	Si	No	CHRONOS MINI+	2200.0	Jamming BB
31	32	19.0	30.0	1400.0	14.0	Si	No	AGRI X	15000.0	Jamming BB
32	33	16.0	28.0	1300.0	13.0	Si	No	SPRAY -L	15000.0	Jamming BB

Figura 57. Datos extraídos BBDD 1.

33	34	17.0	27.0	500.0	5.0	Si	No	SPRAY -S	9000.0	Jamming BB
34	35	12.0	20.0	12.0	0.0	Si	No	CK 23VE	20000.0	Jamming BB
35	36	3.0	10.0	1100.0	31.0	Si	Si	ARES A1	21700.0	Jamming GNSS
36	37	12.0	20.0	12.0	21.0	Si	Si	UX11	1400.0	Jamming GNSS
37	38	3.0	10.0	1100.0	11.0	Si	No	FIREHOUND	1210.0	Jamming BB
38	39	9.0	80.0	120.0	22.0	Si	Si	GUARDIAN EO-IR VRGU23C	1520.0	Jamming GNSS
39	40	14.0	170.0	1500.0	15.0	Si	No	CK280X8	18000.0	Jamming BB
40	41	5.0	80.0	750.0	7.0	Si	No	CK9	5000.0	Jamming BB
41	42	12.0	20.0	112.0	1.0	Si	No	Walkera F210	370.0	Jamming BB
42	43	3.0	10.0	900.0	29.0	Si	Si	Autel Evo 2	1191.0	Jamming GNSS
43	44	9.0	200.0	1433.0	34.0	Si	Si	Bayraktar TB2	50000.0	Jamming GNSS
44	45	7.0	100.0	350.0	24.0	Si	Si	Parrot Anafi	500.0	Jamming GNSS
45	46	6.0	70.0	220.0	23.0	No	No	-	0.0	Spoofing
46	47	9.0	30.0	1400.0	34.0	No	No	-	0.0	Spoofing
47	48	12.0	50.0	160.0	22.0	No	No	-	0.0	Spoofing
48	49	3.0	70.0	1280.0	33.0	No	No	-	0.0	Spoofing
49	50	13.0	75.0	200.0	22.0	No	No	-	0.0	Spoofing
50	51	4.0	20.0	1100.0	31.0	No	No	-	0.0	Spoofing
51	52	6.0	90.0	120.0	22.0	No	No	-	0.0	Spoofing
52	53	7.0	200.0	1807.0	38.0	No	No	-	0.0	Spoofing
53	54	5.0	60.0	150.0	22.0	No	No	-	0.0	Spoofing
54	55	6.0	37.0	1750.0	37.0	No	No	-	0.0	Spoofing
55	56	13.0	30.0	12.0	21.0	No	No	-	0.0	Spoofing
56	57	4.0	24.0	1944.0	39.0	No	No	-	0.0	Spoofing
57	58	6.0	76.0	120.0	22.0	No	No	-	0.0	Spoofing
58	59	16.0	198.0	2000.0	40.0	No	No	-	0.0	Spoofing
59	60	11.0	182.0	795.0	28.0	No	No	-	0.0	Spoofing
60	61	6.0	27.0	1606.0	36.0	No	No	-	0.0	Spoofing
61	62	13.0	20.0	467.0	25.0	No	No	-	0.0	Spoofing
62	63	4.0	10.0	1114.0	31.0	No	No	-	0.0	Spoofing
63	64	6.0	80.0	133.0	22.0	No	No	-	0.0	Spoofing
64	65	6.0	200.0	1987.0	39.0	No	No	-	0.0	Spoofing
65	66	11.0	186.0	78.0	21.0	No	No	-	0.0	Spoofing
66	67	9.0	20.0	1733.0	37.0	No	No	-	0.0	Spoofing
67	68	3.0	10.0	106.0	22.0	No	No	-	0.0	Spoofing

Figura 58. Datos extraídos BBDD 2.

68	69	17.0	80.0	1541.0	35.0	No	No	-	0.0	Spoofing
69	70	22.0	198.0	1000.0	50.0	No	No	-	0.0	Laser
70	71	22.0	146.0	1100.0	51.0	No	No	-	0.0	Laser
71	72	34.0	200.0	600.0	46.0	No	No	-	0.0	Laser
72	73	26.0	163.0	1200.0	52.0	No	No	-	0.0	Laser
73	74	28.0	180.0	875.0	49.0	No	No	-	0.0	Laser
74	75	31.0	89.0	468.0	45.0	No	No	-	0.0	Laser
75	76	33.0	126.0	1365.0	53.0	No	No	-	0.0	Laser
76	77	20.0	120.0	575.0	26.0	No	No	-	0.0	Laser
77	78	34.0	200.0	600.0	46.0	No	No	-	0.0	Laser
78	79	34.0	193.0	1275.0	53.0	No	No	-	0.0	Laser
79	80	27.0	122.0	613.0	46.0	No	No	-	0.0	Laser
80	81	27.0	180.0	1900.0	59.0	No	No	-	0.0	Laser
81	82	20.0	159.0	455.0	25.0	No	No	-	0.0	Laser
82	83	25.0	40.0	789.0	48.0	No	No	-	0.0	Laser
83	84	29.0	120.0	164.0	42.0	No	No	-	0.0	Laser
84	85	22.0	194.0	1000.0	50.0	No	No	-	0.0	Laser
85	86	22.0	144.0	1100.0	51.0	No	No	-	0.0	Laser
86	87	34.0	200.0	600.0	46.0	No	No	-	0.0	Laser
87	88	26.0	163.0	1200.0	52.0	No	No	-	0.0	Laser
88	89	28.0	180.0	875.0	49.0	No	No	-	0.0	Laser
89	90	31.0	89.0	468.0	45.0	No	No	-	0.0	Laser
90	91	33.0	187.0	1365.0	53.0	No	No	-	0.0	Laser
91	92	20.0	120.0	575.0	26.0	No	No	-	0.0	Laser
92	93	34.0	200.0	600.0	46.0	No	No	-	0.0	Laser
93	94	8.0	27.0	461.0	25.0	No	No	-	0.0	Laser
94	95	34.0	179.0	1275.0	53.0	No	No	-	0.0	PEM/MHP
95	96	27.0	122.0	613.0	46.0	No	No	-	0.0	PEM/MHP
96	97	27.0	180.0	1900.0	59.0	No	No	-	0.0	PEM/MHP
97	98	20.0	159.0	455.0	25.0	No	No	-	0.0	PEM/MHP
98	99	25.0	40.0	789.0	48.0	No	No	-	0.0	PEM/MHP
99	100	29.0	120.0	164.0	42.0	No	No	-	0.0	PEM/MHP
100	101	22.0	166.0	1000.0	50.0	No	No	-	0.0	PEM/MHP
101	102	22.0	140.0	1100.0	51.0	No	No	-	0.0	PEM/MHP
102	103	34.0	200.0	600.0	46.0	No	No	-	0.0	PEM/MHP

Figura 59. Datos extraídos BBDD 3.

103	104	26.0	163.0	1200.0	52.0	No	No	-	0.0	PEM/MHP
104	105	28.0	180.0	875.0	49.0	No	No	-	0.0	PEM/MHP
105	106	31.0	89.0	468.0	45.0	No	No	-	0.0	PEM/MHP
106	107	33.0	125.0	1365.0	53.0	No	No	-	0.0	PEM/MHP
107	108	20.0	120.0	575.0	26.0	No	No	-	0.0	PEM/MHP
108	109	34.0	200.0	600.0	46.0	No	No	-	0.0	PEM/MHP
109	110	34.0	175.0	1275.0	53.0	No	No	-	0.0	PEM/MHP
110	111	27.0	122.0	613.0	46.0	No	No	-	0.0	PEM/MHP
111	112	27.0	122.0	613.0	46.0	No	No	-	0.0	PEM/MHP
112	113	27.0	180.0	1900.0	59.0	No	No	-	0.0	PEM/MHP
113	114	20.0	159.0	455.0	25.0	No	No	-	0.0	PEM/MHP
114	115	25.0	40.0	789.0	48.0	No	No	-	0.0	PEM/MHP
115	116	29.0	120.0	164.0	42.0	No	No	-	0.0	PEM/MHP
116	117	22.0	200.0	1000.0	50.0	No	No	-	0.0	PEM/MHP
117	118	22.0	150.0	1100.0	51.0	No	No	-	0.0	PEM/MHP
118	119	34.0	200.0	600.0	46.0	No	No	-	0.0	PEM/MHP
119	120	26.0	263.0	1200.0	52.0	No	No	-	0.0	PEM/MHP
120	121	51.0	163.0	970.0	90.0	No	No	-	0.0	Dron Kamikaze
121	122	52.0	77.0	669.0	87.0	No	No	-	0.0	Dron Kamikaze
122	123	52.0	195.0	1568.0	95.0	No	No	-	0.0	Dron Kamikaze
123	124	51.0	110.0	470.0	85.0	No	No	-	0.0	Dron Kamikaze
124	125	53.0	199.0	572.0	86.0	No	No	-	0.0	Dron Kamikaze
125	126	51.0	175.0	1231.0	92.0	No	No	-	0.0	Dron Kamikaze
126	127	52.0	166.0	510.0	85.0	No	No	-	0.0	Dron Kamikaze
127	128	52.0	180.0	1901.0	99.0	No	No	-	0.0	Dron Kamikaze
128	129	51.0	159.0	456.0	85.0	No	No	-	0.0	Dron Kamikaze
129	130	53.0	40.0	788.0	88.0	No	No	-	0.0	Dron Kamikaze
130	131	52.0	120.0	166.0	82.0	No	No	-	0.0	Dron Kamikaze
131	132	51.0	80.0	2000.0	100.0	No	No	-	0.0	Dron Kamikaze
132	133	51.0	78.0	867.0	89.0	No	No	-	0.0	Dron Kamikaze
133	134	51.0	96.0	364.0	84.0	No	No	-	0.0	Dron Kamikaze
134	135	53.0	35.0	719.0	87.0	No	No	-	0.0	Dron Kamikaze
135	136	52.0	44.0	133.0	82.0	No	No	-	0.0	Dron Kamikaze
136	137	53.0	89.0	1091.0	91.0	No	No	-	0.0	Dron Kamikaze

Figura 60. Datos extraídos BBDD 4.

137	138	53.0	182.0	1931.0	99.0	No	No	-	0.0	Dron Kamikaze
138	139	51.0	196.0	1298.0	93.0	No	No	-	0.0	Dron Kamikaze
139	140	51.0	105.0	1312.0	93.0	No	No	-	0.0	Dron Kamikaze
140	141	51.0	117.0	513.0	85.0	No	No	-	0.0	Dron Kamikaze
141	142	52.0	77.0	121.0	82.0	No	No	-	0.0	Dron Kamikaze
142	143	53.0	34.0	1122.0	91.0	No	No	-	0.0	Dron Kamikaze
143	144	52.0	60.0	117.0	82.0	No	No	-	0.0	Dron Kamikaze
144	145	51.0	76.0	1234.0	92.0	No	No	-	0.0	Dron Kamikaze
145	146	53.0	82.0	147.0	82.0	No	No	-	0.0	Dron Kamikaze
146	147	51.0	142.0	1476.0	95.0	No	No	-	0.0	Dron Kamikaze
147	148	52.0	78.0	848.0	89.0	No	No	-	0.0	Dron Kamikaze
148	149	14.0	78.0	862.0	29.0	No	No	-	0.0	Dron Kamikaze

Figura 61. Datos extraídos BBDD 5.

Anexo 2.

En esta sección se muestran los datos que han sido seleccionados para el Machine Learning, corresponden con los mismos datos que los extraídos de la base de datos pero que han sido manipulados para que el modelo ML los pueda procesar.

Valores seleccionados y transformados de la base de datos:

	velocidad	altura	distancia	amenaza	comercial	gps
0	5.0	80.0	220.0	2.0	1	0
1	8.0	20.0	1400.0	14.0	1	0
2	11.0	40.0	160.0	1.0	1	0
3	2.0	60.0	1280.0	12.0	1	0
4	12.0	85.0	200.0	2.0	1	0
5	3.0	10.0	1100.0	11.0	1	0
6	5.0	80.0	120.0	1.0	1	0
7	3.0	200.0	1800.0	18.0	1	0
8	4.0	50.0	150.0	1.0	1	0
9	5.0	27.0	1750.0	17.0	1	0
10	12.0	20.0	12.0	0.0	1	0
11	3.0	10.0	2100.0	21.0	1	0
12	5.0	80.0	120.0	1.0	1	0
13	15.0	200.0	2000.0	20.0	1	0
14	10.0	190.0	795.0	7.0	1	0
15	5.0	27.0	1628.0	16.0	1	0
16	12.0	20.0	452.0	4.0	1	0
17	16.0	80.0	1920.0	39.0	1	1
18	13.0	78.0	847.0	29.0	1	1
19	11.0	96.0	350.0	24.0	1	1
20	10.0	35.0	722.0	27.0	1	1
21	12.0	20.0	120.0	22.0	1	1
22	2.0	10.0	100.0	21.0	1	1
23	3.0	10.0	2000.0	40.0	1	1
24	5.0	80.0	120.0	22.0	1	1
25	5.0	200.0	1970.0	39.0	1	1
26	10.0	186.0	65.0	0.0	1	0
27	7.0	27.0	455.0	25.0	1	1
28	8.0	20.0	1722.0	37.0	1	1
29	3.0	10.0	1100.0	11.0	1	0
30	5.0	80.0	1867.0	18.0	1	0
31	19.0	30.0	1400.0	14.0	1	0
32	16.0	28.0	1300.0	13.0	1	0

Figura 62. Datos seleccionados para ML 1.

33	17.0	27.0	500.0	5.0	1	0
34	12.0	20.0	12.0	0.0	1	0
35	3.0	10.0	1100.0	31.0	1	1
36	12.0	20.0	12.0	21.0	1	1
37	3.0	10.0	1100.0	11.0	1	0
38	9.0	80.0	120.0	22.0	1	1
39	14.0	170.0	1500.0	15.0	1	0
40	5.0	80.0	750.0	7.0	1	0
41	12.0	20.0	112.0	1.0	1	0
42	3.0	10.0	900.0	29.0	1	1
43	9.0	200.0	1433.0	34.0	1	1
44	7.0	100.0	350.0	24.0	1	1
45	6.0	70.0	220.0	23.0	0	0
46	9.0	30.0	1400.0	34.0	0	0
47	12.0	50.0	160.0	22.0	0	0
48	3.0	70.0	1280.0	33.0	0	0
49	13.0	75.0	200.0	22.0	0	0
50	4.0	20.0	1100.0	31.0	0	0
51	6.0	90.0	120.0	22.0	0	0
52	7.0	200.0	1807.0	38.0	0	0
53	5.0	60.0	150.0	22.0	0	0
54	6.0	37.0	1750.0	37.0	0	0
55	13.0	30.0	12.0	21.0	0	0
56	4.0	24.0	1944.0	39.0	0	0
57	6.0	76.0	120.0	22.0	0	0
58	16.0	198.0	2000.0	40.0	0	0
59	11.0	182.0	795.0	28.0	0	0
60	6.0	27.0	1606.0	36.0	0	0
61	13.0	20.0	467.0	25.0	0	0
62	4.0	10.0	1114.0	31.0	0	0
63	6.0	80.0	133.0	22.0	0	0
64	6.0	200.0	1987.0	39.0	0	0
65	11.0	186.0	78.0	21.0	0	0
66	9.0	20.0	1733.0	37.0	0	0
67	3.0	10.0	106.0	22.0	0	0
68	17.0	80.0	1541.0	35.0	0	0

Figura 63. Datos seleccionados para ML 2.

69	22.0	198.0	1000.0	50.0	0	0
70	22.0	146.0	1100.0	51.0	0	0
71	34.0	200.0	600.0	46.0	0	0
72	26.0	163.0	1200.0	52.0	0	0
73	28.0	180.0	875.0	49.0	0	0
74	31.0	89.0	468.0	45.0	0	0
75	33.0	126.0	1365.0	53.0	0	0
76	20.0	120.0	575.0	26.0	0	0
77	34.0	200.0	600.0	46.0	0	0
78	34.0	193.0	1275.0	53.0	0	0
79	27.0	122.0	613.0	46.0	0	0
80	27.0	180.0	1900.0	59.0	0	0
81	20.0	159.0	455.0	25.0	0	0
82	25.0	40.0	789.0	48.0	0	0
83	29.0	120.0	164.0	42.0	0	0
84	22.0	194.0	1000.0	50.0	0	0
85	22.0	144.0	1100.0	51.0	0	0
86	34.0	200.0	600.0	46.0	0	0
87	26.0	163.0	1200.0	52.0	0	0
88	28.0	180.0	875.0	49.0	0	0
89	31.0	89.0	468.0	45.0	0	0
90	33.0	187.0	1365.0	53.0	0	0
91	20.0	120.0	575.0	26.0	0	0
92	34.0	200.0	600.0	46.0	0	0
93	8.0	27.0	461.0	25.0	0	0
94	34.0	179.0	1275.0	53.0	0	0
95	27.0	122.0	613.0	46.0	0	0
96	27.0	180.0	1900.0	59.0	0	0
97	20.0	159.0	455.0	25.0	0	0
98	25.0	40.0	789.0	48.0	0	0
99	29.0	120.0	164.0	42.0	0	0
100	22.0	166.0	1000.0	50.0	0	0
101	22.0	140.0	1100.0	51.0	0	0
102	34.0	200.0	600.0	46.0	0	0
103	26.0	163.0	1200.0	52.0	0	0
104	28.0	180.0	875.0	49.0	0	0

Figura 64. Datos seleccionados para ML 3.

105	31.0	89.0	468.0	45.0	0	0
106	33.0	125.0	1365.0	53.0	0	0
107	20.0	120.0	575.0	26.0	0	0
108	34.0	200.0	600.0	46.0	0	0
109	34.0	175.0	1275.0	53.0	0	0
110	27.0	122.0	613.0	46.0	0	0
111	27.0	122.0	613.0	46.0	0	0
112	27.0	180.0	1900.0	59.0	0	0
113	20.0	159.0	455.0	25.0	0	0
114	25.0	40.0	789.0	48.0	0	0
115	29.0	120.0	164.0	42.0	0	0
116	22.0	200.0	1000.0	50.0	0	0
117	22.0	150.0	1100.0	51.0	0	0
118	34.0	200.0	600.0	46.0	0	0
119	26.0	263.0	1200.0	52.0	0	0
120	51.0	163.0	970.0	90.0	0	0
121	52.0	77.0	669.0	87.0	0	0
122	52.0	195.0	1568.0	95.0	0	0
123	51.0	110.0	470.0	85.0	0	0
124	53.0	199.0	572.0	86.0	0	0
125	51.0	175.0	1231.0	92.0	0	0
126	52.0	166.0	510.0	85.0	0	0
127	52.0	180.0	1901.0	99.0	0	0
128	51.0	159.0	456.0	85.0	0	0
129	53.0	40.0	788.0	88.0	0	0
130	52.0	120.0	166.0	82.0	0	0
131	51.0	80.0	2000.0	100.0	0	0
132	51.0	78.0	867.0	89.0	0	0
133	51.0	96.0	364.0	84.0	0	0
134	53.0	35.0	719.0	87.0	0	0
135	52.0	44.0	133.0	82.0	0	0
136	53.0	89.0	1091.0	91.0	0	0
137	53.0	182.0	1931.0	99.0	0	0
138	51.0	196.0	1298.0	93.0	0	0
139	51.0	105.0	1312.0	93.0	0	0
140	51.0	117.0	513.0	85.0	0	0

Figura 65. Datos seleccionados para ML 4.

141	52.0	77.0	121.0	82.0	0	0
142	53.0	34.0	1122.0	91.0	0	0
143	52.0	60.0	117.0	82.0	0	0
144	51.0	76.0	1234.0	92.0	0	0
145	53.0	82.0	147.0	82.0	0	0
146	51.0	142.0	1476.0	95.0	0	0
147	52.0	78.0	848.0	89.0	0	0
148	14.0	78.0	862.0	29.0	0	0

Figura 66. Datos seleccionados para ML 5.