



UNIVERSIDAD
DE MÁLAGA



ESCUELA DE INGENIERÍAS INDUSTRIALES

Departamento: Ingeniería Civil, de Materiales y Fabricación

**Área de Conocimiento: Ingeniería de los Procesos de
Fabricación**

TRABAJO FIN DE GRADO

**APLICACIÓN DE REDES NEURONALES ARTIFICIALES EN
LA OBTENCIÓN DE MODELOS PREDICTIVOS DE LA
MORFOLOGÍA DE LA VIRUTA EN EL MECANIZADO EN
SECO DE ALEACIONES LIGERAS DE USO AERONÁUTICO**

Grado en:

Ingeniería Mecánica

Autor: José Varela Agüera
Tutor: Francisco Javier Trujillo Vilches
Cotutor: Sergio Martín Béjar

MÁLAGA, octubre de 2024

Agradecimientos

Quiero agradecer de corazón a mis seres queridos, cuyo apoyo constante y ánimo han sido fundamentales. Su confianza y compañía me han dado la fuerza para superar este reto. A mis profesores, les agradezco por compartir su conocimiento, pero sobre todo por inspirarme y motivarme en el camino.

Resumen

Las aleaciones ligeras como el aluminio y el titanio, en concreto la UNS A92024 (Al-Cu), UNS A97075 (Al-Zn) y Ti6Al4V, son esenciales en la fabricación de componentes estructurales de aeronaves, donde los estándares de calidad son extremadamente altos y los márgenes de tolerancia muy estrechos. Entre los desafíos del mecanizado en seco está la adecuada evacuación de la viruta y el control de la temperatura, factores que pueden afectar la integridad del componente mecanizado.

Este Trabajo de Fin de Grado se centra en el desarrollo de modelos predictivos de parámetros geométricos de la viruta obtenida en el mecanizado en seco de aleaciones ligeras, mediante el uso de redes neuronales artificiales (RNA). Este trabajo se encuentra incluido dentro del proyecto de investigación "Sistema Experto para la Mejora de la Integridad Superficial en el Mecanizado Sostenible de Aleaciones Ligeras", del "Programa Estatal para Impulsar la Investigación Científico-Técnica y su Transferencia" del Plan Estatal de Investigación Científica, Técnica y de Innovación 2021-2023.

El objetivo principal del trabajo es aplicar técnicas de *Machine learning* para predecir parámetros de la morfología de la viruta, como la altura de pico, altura de valle, la sección, el ancho y el espesor, durante el mecanizado en seco de aleaciones de aluminio y titanio. Para ello, se utilizarán datos de investigaciones previas realizadas por el grupo de investigación iFAB "Ingeniería de Fabricación" (TEP933) de la Universidad de Málaga. La programación de las redes neuronales y su implementación se ha realizado mediante el uso de Matlab y Simulink .

El trabajo incluye una aproximación a aplicaciones recientes de la tecnología, el aprendizaje y optimización de las RNA, así como con el desarrollo de una aplicación que permita predecir los parámetros de la viruta basándose en las condiciones de corte y la aleación utilizada. Este estudio no solo contribuirá a mejorar los procesos de fabricación aeronáutica, sino que también abrirá nuevas vías para la aplicación de inteligencia artificial en el mecanizado de materiales avanzados.

Palabras clave: Mecanizado en seco, Aleaciones ligeras, Morfología de la viruta, *Machine learning*, Redes neuronales artificiales, Matlab.

Abstract

Light alloys such as aluminium and titanium, specifically UNS A92024 (Al-Cu), UNS A97075 (Al-Zn) and Ti6Al4V, are essential in the manufacture of aircraft structural components, where quality standards are extremely high and tolerances very tight. Among the challenges of dry machining are proper chip evacuation and temperature control, factors that can affect the integrity of the machined component.

This Final Degree Project focuses on the development of predictive models of geometric parameters of the chip obtained in the dry machining of light alloys, using artificial neural networks (ANN). This work is included in the research project 'Expert System for the Improvement of Surface Integrity in the Sustainable Machining of Light Alloys', of the 'State Programme to Promote Scientific-Technical Research and its Transfer' of the State Plan for Scientific, Technical and Innovation Research 2021-2023.

The main objective of the work is to apply *Machine learning* techniques to predict chip morphology parameters, such as peak height, valley height, section, width and thickness, during the dry machining of aluminium and titanium alloys. For this purpose, data from previous research carried out by the iFAB 'Manufacturing Engineering' research group (TEP933) of the University of Malaga will be used. The programming of the neural networks and their implementation has been carried out using Matlab and Simulink.

The work includes an approach to the state of the art, the learning and optimisation of the ANNs, as well as the development of an application to predict the chip parameters based on the cutting conditions and the alloy used. This study will not only contribute to improving aeronautical manufacturing processes but will also open up new avenues for the application of artificial intelligence in the machining of advanced materials.

Keywords: Dry machining, Light alloys, Chip morphology, *Machine learning*, Artificial neural networks, Matlab.

Índice general

RESUMEN.....	I
ABSTRACT	III
1. INTRODUCCIÓN	1
1.1. CONTEXTO.....	1
1.2. OBJETIVOS.....	4
1.3. ORGANIZACIÓN DE LA MEMORIA	5
2. MARCO TEÓRICO	7
2.1. ALEACIONES LIGERAS	7
2.1.1. Aleaciones de aluminio	7
2.1.2. Aleaciones de titanio	13
2.2. MECANIZADO EN SECO.....	16
2.3. FORMACIÓN DE LA VIRUTA EN PROCESOS DE MECANIZADO EN SECO	18
2.3.1. Definición de viruta	18
2.3.2. Mecánica de la formación de viruta.....	19
2.3.2.1. Herramientas de corte	19
2.3.2.2. Influencia de la geometría de la herramienta de corte en la formación de virutas y fenómenos de acumulación	20
2.3.2.3. Parámetros de corte.....	21
2.3.2.4. Evacuación de calor.....	22
2.3.2.5. Forma de la viruta	23
2.4. REDES NEURONALES ARTIFICIALES Y SU APLICACIÓN EN MECANIZADO.....	25
2.4.1. Introducción.....	25
2.4.2. ¿Qué es una red neuronal artificial?.....	26
2.4.3. Componentes clave de una red neuronal artificial	27
2.4.4. Arquitectura de las redes neuronales artificiales	27
2.4.5. Tipologías de las redes neuronales artificiales	29
2.4.6. Tipos del aprendizaje en redes neuronales artificiales.....	31
2.4.7. Proceso de Entrenamiento de una Red Neuronal Artificial.....	31
2.4.8. Funciones de activación	33
2.4.9. Algoritmos de optimización	34
2.4.10. Regularización en redes neuronales artificiales	35
2.4.10.1. Técnicas de Regularización	35
2.5. APLICACIONES DE REDES NEURONALES ARTIFICIALES EN EL MECANIZADO	37
3. METODOLOGÍA	43
3.1. INTRODUCCIÓN.....	43
3.2. MATERIALES	44
3.3. MÉTODO DE OBTENCIÓN DE LOS DATOS EXPERIMENTALES	44
3.3.1. Introducción.....	44
3.3.2. Mecanizado.....	44

3.3.3.	<i>Preparación de la viruta</i>	45
3.3.4.	<i>Medición de la viruta</i>	46
3.4.	DESARROLLO DEL MODELO DE RED NEURONAL ARTIFICIAL	48
3.3.1.	<i>Introducción</i>	48
3.3.2.	<i>Preparación de los datos</i>	49
3.3.3.	<i>Proceso de análisis ANOVA</i>	49
3.3.4.	<i>Normalización de datos</i>	50
3.3.5.	<i>Configuración de la red neuronal artificial</i>	50
3.3.5.1.	Capas de la red	50
3.3.5.2.	Neuronas	51
3.3.5.3.	Funciones de activación.....	52
3.3.5.4.	Elección del algoritmo de entrenamiento	52
3.3.5.5.	Configuración del algoritmo	53
3.4.	EVALUACIÓN Y VALIDACIÓN DEL MODELO	53
3.4.1.	<i>Evaluación de la RNA</i>	53
3.4.2.	<i>Comparación entre modelos tradicionales y RNA</i>	55
3.5.	DESARROLLO DE APLICACIÓN CON SIMULINK.....	55
4.	RESULTADOS.....	57
4.1.	INTRODUCCIÓN.....	57
4.2.	DISTRIBUCIÓN Y ANÁLISIS DE LOS DATOS EXPERIMENTALES	57
4.2.1.	<i>Altura de los picos (h_p) y de los valles (h_v)</i>	58
4.2.1.	<i>Área de sección transversal de la viruta (S) y anchura de la viruta (b)</i>	59
4.2.2.	<i>Espesor de viruta (t)</i>	61
4.3.	ANÁLISIS ANOVA DE LOS DATOS EXPERIMENTALES.....	62
4.4.	NORMALIZACIÓN DE LOS DATOS EXPERIMENTALES	63
4.5.	DESEMPEÑO Y VALIDACIÓN DE LA RED NEURONAL ARTIFICIAL	65
4.5.1.	<i>Número óptimo de neuronas en la capa oculta</i>	65
4.5.2.	<i>Histogramas de errores</i>	66
4.5.3.	<i>Gráficos de regresión</i>	68
4.5.4.	<i>Comparación de modelos predictivos</i>	74
4.6.	PREDICCIÓN CON SIMULINK.....	79
5.	CONCLUSIONES Y TRABAJOS FUTUROS	81
5.1.	CONCLUSIONES.....	81
5.2.	TRABAJOS FUTUROS.....	82
6.	BIBLIOGRAFÍA	85
ANEXOS A:	CÓDIGO DE MATLAB	89

Índice de figuras

FIGURA 1. ALEACIONES DE ALUMINIO MÁS USADAS EN LA FABRICACIÓN DE AERONAVES COMERCIALES. ADAPTADA DE [10].	11
FIGURA 2. DISTRIBUCIÓN DE MATERIALES PARA UNA SELECCIÓN DE PRODUCTOS BOEING. ADAPTADA DE [10].	12
FIGURA 3. (A) DISPOSICIÓN ORTOGONAL DE CORTE; (B) INSTANCIA INICIAL DE MECANIZADO; (C) MUESTRA PARA VARIAS PRUEBAS DE CORTE [16].	17
FIGURA 4. RELACIÓN DUREZA-RESISTENCIA DE MATERIALES DE CORTE [21].	20
FIGURA 5. MECANISMOS DE FORMACIÓN DE BUE Y BUL EN EL TORNEADO EN SECO DE LA ALEACIÓN UNS A92024 CON HERRAMIENTAS DE WC-Co Y RECUBRIMIENTO DE TiN [22].	21
FIGURA 6. DISTRIBUCIÓN DEL CALOR EN EL PROCESO DE CORTE ORTOGONAL. ADAPTADA DE [22].	22
FIGURA 7. TIPOS DE VIRUTA: (A) FRAGMENTADA; (B) CONTINUA; (C) SEGMENTADA [22].	23
FIGURA 8. TIPOLOGÍA DE VIRUTA SEGÚN ISO 3685 [8].	25
FIGURA 9. ANALOGÍA ENTRE NEURONAS BIOLÓGICAS Y NEURONAS ARTIFICIALES EN REDES NEURONALES [23].	26
FIGURA 10. CAPAS DE UNA RED NEURONAL ARTIFICIAL [24].	29
FIGURA 11. UNDERFITTING VS. OVERFITTING EN MODELOS DE APRENDIZAJE AUTOMÁTICO.	35
FIGURA 12. (A) PROCESOS DE CORTE RESPALDADOS PARA APLICACIONES DE APRENDIZAJE AUTOMÁTICO. (B) DIFERENTES TIPOS DE APLICACIONES DE APRENDIZAJE AUTOMÁTICO APLICADOS EN PROCESOS DE CORTE [25].	37
FIGURA 13. DETECCIÓN DE IMÁGENES EXPERIMENTALES MEDIANTE LA RED NEURONAL PROFUNDA (DNN) ENTRENADA CON FEM PARA EL MONITOREO BASADO EN IMÁGENES [27].	38
FIGURA 14. DIAGRAMA DE OPERACIONES UTILIZADAS PARA MODELAR LA TEMPERATURA DE CORTE EN PROCESOS DE TORNEADO DURO.	39
FIGURA 15. FLUJO DE TRABAJO EN LA METODOLOGÍA.	43
FIGURA 16. (A) TORNO HORIZONTAL UTILIZADO EN LAS PRUEBAS (PINACHO SP285R); (B) BARRAS DE PRUEBA CILÍNDRICAS [22].	45
FIGURA 17. (A) PRENSA DE MONTAJE EN CALIENTE; (B) EQUIPO DE PULIDO; (C) MUESTRAS DE VIRUTA; (D) EQUIPO DE SOM (MICROSCOPIA ÓPTICA ESTEREOSCÓPICA) [22].	46
FIGURA 18. (A) PARÁMETROS GEOMÉTRICOS LONGITUDINALES DE LA VIRUTA Y (B) PARÁMETROS GEOMÉTRICOS TRANSVERSALES DE LA VIRUTA [22].	47
FIGURA 19. ARQUITECTURA DE LA RED NEURONAL ARTIFICIAL.	51
FIGURA 20. VALORES MEDIOS DE LA ALTURA DE PICO (H_p) PARA CADA MATERIAL (A) , (B) , (C) , EN FUNCIÓN DEL AVANCE (F) Y LA VELOCIDAD DE CORTE (v_c). VALORES MEDIOS DE LA ALTURA DE VALLE (H_v) PARA CADA MATERIAL (D) , (E) , (F) , EN FUNCIÓN DEL AVANCE (F) Y LA VELOCIDAD DE CORTE (v_c).	58
FIGURA 21. VALORES MEDIOS DE LA SECCIÓN DE LA VIRUTA (S) PARA CADA MATERIAL (A) , (B) , (C) , EN FUNCIÓN DEL AVANCE (F) Y LA VELOCIDAD DE CORTE (v_c). VALORES MEDIOS DEL ANCHO DE VIRUTA (B) PARA CADA MATERIAL (D) , (E) , (F) , EN FUNCIÓN DEL AVANCE (F) Y LA VELOCIDAD DE CORTE (v_c).	60
FIGURA 22. VALORES MEDIOS DEL ESPESOR DE LA VIRUTA (T) PARA CADA MATERIAL (A) , (B) , (C) , EN FUNCIÓN DEL AVANCE (F) Y LA VELOCIDAD DE CORTE (v_c).	61
FIGURA 23. NORMALIZACIÓN DE LAS VARIABLES DE ENTRADA MATERIAL (A) , VELOCIDAD DE CORTE (B) Y AVANCE (C) .	63

FIGURA 24. NORMALIZACIÓN DE LAS VARIABLES DE SALIDA ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E)	64
FIGURA 25. VALOR DEL RMSE PARA DISTINTOS NÚMEROS DE NEURONAS EN LA CAPA OCULTA.....	65
FIGURA 26. HISTOGRAMA DE ERRORES DE ENTRENAMIENTO PARA EL CONJUNTO DE DATOS NORMALIZADO.	66
FIGURA 27. HISTOGRAMA DE ERRORES DE ENTRENAMIENTO PARA CADA VARIABLE DE SALIDA: ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E)	67
FIGURA 28. RESULTADOS DE LA REGRESIÓN: (A) DATOS DE ENTRENAMIENTO; (B) DATOS DE VALIDACIÓN; (C) TODO EL CONJUNTO DE DATOS.	69
FIGURA 29. RESULTADOS DE REGRESIÓN PARA CADA VARIABLE DE SALIDA EN EL ENTRENAMIENTO: ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E) . 70	
FIGURA 30. RESULTADOS DE REGRESIÓN PARA CADA VARIABLE DE SALIDA EN VALIDACIÓN: ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E)	71
FIGURA 31. RESULTADOS DE REGRESIÓN PARA CADA VARIABLE DE SALIDA EN CONJUNTO: ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E)	72
FIGURA 32. RESULTADOS DE REGRESIÓN PARA CADA VARIABLE DE SALIDA: ALTURA DE PICO (A) , ALTURA DE VALLE (B) , SECCIÓN DE LA VIRUTA (C) , ANCHO DE LA VIRUTA (D) Y ESPESOR DE LA VIRUTA (E)	75
FIGURA 33. COMPARACIÓN DE MODELOS CON DATOS EXPERIMENTALES PARA DIFERENTES ALEACIONES: A97075 (S (A) , T (D)); A92024 (S (B) , T (E) ; Ti6Al4V (S (C) , T (F)).	77
FIGURA 34. COMPARACIÓN DE MODELOS CON DATOS EXPERIMENTALES PARA DIFERENTES ALEACIONES: A97075 (B (A)); A92024 (B (B)); Ti6Al4V (B (C)).	78
FIGURA 35. DIAGRAMA DE SIMULINK PARA LA IMPLEMENTACIÓN DE UNA RED NEURONAL.....	79
FIGURA 36. COMPOSICIÓN DEL BLOQUE DE LA RED NEURONAL ARTIFICIAL EN SIMULINK.	79
FIGURA 37. ESTRUCTURA BÁSICA DE UN NEURONA ARTIFICIAL EN SIMULINK.	80
FIGURA 38. DIAGRAMA DE PESOS Y SESGOS DE LA CAPA OCULTA.	80

Índice de tablas

TABLA 1. CLASIFICACIÓN Y CARACTERÍSTICAS DE LAS ALEACIONES DE ALUMINIO: COMPOSICIÓN, PROPIEDADES Y APLICACIONES [9].	10
TABLA 2. COMPOSICIÓN DE LA ALEACIÓN MECANIZADA (% EN PESO).	44
TABLA 3. PARÁMETROS DE CORTE.	45
TABLA 4. ANÁLISIS ANOVA.	62
TABLA 5. RMSE Y R^2 AJUSTADO DE LA RED NEURONAL (10 NEURONAS).	69
TABLA 6. RMSE Y R^2 AJUSTADO DE LA RNA PARA EL CONJUNTO Y PARA CADA VARIABLE DE SALIDA.	73
TABLA 7. MODELOS DE REGRESIÓN.	74

1. Introducción

1.1. Contexto

El sector aeronáutico, un pilar fundamental de la economía global, abarca una variedad de actividades que van desde la fabricación de aeronaves hasta la gestión del tráfico aéreo. En el panorama contemporáneo, su importancia se extiende más allá de las fronteras nacionales, influenciando tanto a nivel mundial como europeo. A nivel global, el sector aeronáutico se distingue por su constante innovación y precisión técnica. En febrero de 2024, el sector aéreo alcanzó un hito significativo al lograr la plena recuperación del tráfico total de pasajeros, superando el umbral de 2019 en un 5,7%. Este éxito se reflejó en un crecimiento anual de los ingresos por pasajeros-kilómetro (RPK) del 21,5% interanual, así como en mejoras en el factor de ocupación de pasajeros, que se establecieron en las proximidades de los niveles prepandémicos [1].

El dinamismo del sector aeronáutico se ve impulsado no solo por las demandas del mercado, sino también por los desarrollos geopolíticos, las inversiones en defensa y, cada vez más importante, la urgencia del cambio climático. Las tensiones diplomáticas actuales han generado una mayor incertidumbre a nivel mundial y, como resultado, se está invirtiendo más dinero en defensa. Esta situación ha llevado a una intensificación de la investigación en nuevas tecnologías que no solo fortalecen la seguridad nacional, sino que también agilizan los sectores industriales, en especial el aeronáutico. Además, con el cambio climático convirtiéndose en una preocupación global, existe una creciente presión para desarrollar tecnologías más sostenibles

en la industria aeronáutica, como aviones más eficientes en el consumo de combustible y procesos de fabricación más eficientes. Este impulso hacia la sostenibilidad está moldeando la dirección de la innovación en el sector, con la necesidad de reducir las emisiones de carbono y minimizar el impacto ambiental de la aviación [2].

La importancia del sector aeronáutico se extiende también al ámbito europeo, donde se han implementado iniciativas estratégicas para fortalecer la cooperación y la innovación en el sector de la defensa. El Plan de Acción Europeo de la Defensa, lanzado por la Comisión Europea en noviembre de 2016, ha sido fundamental en este sentido. Este plan ha dado lugar al establecimiento del Fondo Europeo de Defensa (FED), que desempeña un papel crucial al apoyar la investigación y el desarrollo de tecnologías punteras y equipos interoperables. Con un presupuesto de 7.953 millones de euros para el período 2021-2027, el FED financia proyectos de investigación colaborativa que abordan las amenazas emergentes y futuras a la seguridad, promoviendo así la innovación y la participación transfronteriza de las pymes en el sector aeronáutico y de defensa europeo [3].

En el ámbito nacional, el sector aeronáutico desempeña un papel significativo en la economía y el empleo. En España, por ejemplo, el sector representa una parte importante del gasto en innovación tecnológica y del empleo nacional. Se estima que genera más de 108,000 empleos directos e indirectos y contribuye de manera sustancial al Producto Interno Bruto (PIB) nacional. Aunque actualmente representa el 0,77% del PIB nacional, se espera que este porcentaje aumente al 1% para el año 2025, reflejando el compromiso continuo de España con el crecimiento y la excelencia en el sector aeronáutico [4].

La búsqueda de la excelencia en los procesos de mecanización industrial se ha convertido en una necesidad imperante en el sector aeronáutico nacional. Ejemplos como la introducción de sistemas de control numérico computarizado (CNC) en la fabricación de componentes aeroespaciales han demostrado su impacto transformador al mejorar la precisión, reducir los tiempos de producción y optimizar los recursos utilizados. Esta innovación tecnológica no solo refuerza la competitividad de las empresas aeronáuticas, sino que también impulsa la calidad y fiabilidad de los productos fabricados, asegurando así su relevancia en el mercado global.

La mejora de la eficiencia en los procesos de fabricación es un objetivo clave en el sector aeronáutico nacional, y el precio neto de la electricidad desempeña un papel significativo en este aspecto. En España, el precio para uso industrial es notablemente inferior a la media de la zona euro, gracias al potencial de energías renovables del país. Con un costo promedio de alrededor de 0,10 euros por kilovatio-hora (kWh), comparado con los 0,18 euros/kWh de la Unión Europea, España ofrece a las empresas aeronáuticas una ventaja competitiva importante al reducir sus costos operativos y promover la eficiencia en la producción [5].

La introducción del big data, la inteligencia artificial (IA) y el Internet de las cosas (IoT) representa un punto crucial en la evolución del sector aeronáutico. Estas tecnologías emergentes están transformando radicalmente la forma en que se abordan los desafíos en la industria, ofreciendo nuevas oportunidades para mejorar la eficiencia, la calidad y la competitividad. Las soluciones de IA, por ejemplo, tienen un potencial inmenso para optimizar cada fase del proceso de fabricación. Al analizar enormes volúmenes de datos, desde los niveles de existencias hasta el rendimiento de las máquinas y los patrones de demanda, la IA puede identificar tendencias, predecir fallos y tomar decisiones más informadas en tiempo real. Esto no solo permite una producción más eficiente y precisa, sino que también abre la puerta a la personalización de productos y servicios de acuerdo con las necesidades específicas de los clientes. Además, la integración del IoT en los procesos de fabricación permite la interconexión de dispositivos y equipos, facilitando la monitorización remota, el control en tiempo real y la optimización de la eficiencia operativa. En conjunto, estas tecnologías están redefiniendo el panorama del sector aeronáutico, ofreciendo nuevas oportunidades para la innovación y el crecimiento en un mundo cada vez más digitalizado y competitivo [6].

Para situar esta investigación en el contexto adecuado, es importante reconocer que se basa en el trabajo previo realizado por otros investigadores en el campo. Se ha observado que existe un espacio limitado en la literatura existente respecto al mecanizado sostenible de aleaciones ligeras en el sector aeronáutico, especialmente en lo que respecta al empleo de técnicas de redes neuronales para mejorar la comprensión y el control de los procesos.

Dentro del ámbito aeronáutico, existe un elemento fundamental que ha impulsado la evolución y la fiabilidad de las aeronaves: las aleaciones ligeras. Entre estas, el aluminio ha destacado por su versatilidad y resistencia, siendo un componente esencial en la fabricación de estructuras aeroespaciales. La meticulosa elaboración de estas piezas, cruciales para la seguridad y eficiencia de las aeronaves, implica una serie de procesos especializados, donde el mecanizado ocupa un lugar destacado, principalmente las operaciones de torneado, taladrado y fresado. Debido a razones medioambientales, la tendencia actual es la de realizar estos procesos en seco (ausencia total de fluidos de corte). En estas condiciones, la evacuación de viruta se convierte en un factor especialmente relevante. Las altas temperaturas en el área de corte dan lugar a fenómenos de adherencia del material de la herramienta de corte, lo que dificulta la evacuación de la viruta. Además, influye en las desviaciones geométricas de la pieza mecanizada y en la continuidad y estabilidad del proceso. Por ello, el control y la monitorización del proceso de formación de viruta son aspectos a tener en cuenta en la mejora del rendimiento de estos procesos.

En este contexto se encuentra trabajando el grupo de investigación TEP933 “Ingeniería de Fabricación” de la Universidad de Málaga, en el marco del proyecto de investigación PID2021-125988OBI00, por título “SISTEMA EXPERTO PARA LA MEJORA DE LA INTEGRIDAD SUPERFICIAL

EN EL MECANIZADO SOSTENIBLE DE ALEACIONES LIGERAS”, dentro del programa “Proyectos de Generación de Conocimiento” del “Programa Estatal para Impulsar la Investigación Científico-Técnica y su Transferencia del Plan Estatal de Investigación Científica, Técnica y de Innovación 2021-2023”. Entre los objetivos de este proyecto se encuentra la aplicación de técnicas de *Machine learning*, en concreto la utilización de redes neuronales artificiales, para la obtención de modelos predictivos de distintas variables de salida del proceso en función de los parámetros de corte. Entre estas variables se encuentran distintos parámetros geométricos relacionados con la morfología de la viruta.

1.2. Objetivos

El objetivo principal de este Trabajo Fin de Grado es la utilización de redes neuronales artificiales para la obtención de modelos que permitan predecir distintos parámetros de la morfología de la viruta (factor de recalcado (ζ), sección (S) y espesor de la viruta (t), entre otros) en el mecanizado en seco de distintas aleaciones ligeras de uso aeronáutico, tales como las aleaciones de aluminio UNS A97075 (Al-Zn) y UNS A92024 (Al-Cu), así como la aleación de Titanio Ti6Al4V. Para ello se realizará una programación de la red neuronal mediante el uso de Matlab. Además, se hará uso del software Simulink para la simulación del comportamiento de la RNA. Como datos de partida se utilizarán los datos obtenidos por investigadores del grupo durante el desarrollo de diversas Tesis Doctorales. Para alcanzar el propósito de este trabajo se establecen los siguientes objetivos descritos a continuación.

Realizar el tratamiento y estudio de los datos experimentales para asegurar la coherencia entre los parámetros de entrada y los resultados correspondientes, preparando así el conjunto de datos para el entrenamiento de las redes neuronales.

Utilizar la herramienta MATLAB para diseñar y entrenar redes neuronales artificiales capaces de predecir resultados relacionados con las propiedades de la viruta a partir de los datos recopilados.

Por último, comparar los resultados obtenidos mediante modelos de RNA optimizados con otras tipologías de modelos, como el potencial o el Método de Superficie de Respuesta, para demostrar la eficacia del uso de algoritmos de RNA y la metodología propuesta.

1.3. Organización de la memoria

El trabajo comienza con una "Introducción y Contexto", donde se presenta el tema central: el mecanizado en seco de aleaciones ligeras y la implementación de redes neuronales para la predicción de parámetros de viruta. Esta sección justifica la importancia del estudio tanto en un contexto práctico como académico, estableciendo el problema y definiendo los objetivos específicos del TFG. Además, se ofrece una visión general de cómo está organizado el documento, preparando al lector para la exploración detallada que sigue.

Siguiendo la introducción, el "Marco Teórico" profundiza en los fundamentos necesarios para comprender el estudio. Aquí se abordan varios temas esenciales: las características, ventajas y desafíos del mecanizado en seco; las propiedades y usos de las aleaciones ligeras, como el aluminio y el titanio; la formación de viruta y los factores que afectan este proceso. También se introducen las redes neuronales, describiendo sus componentes básicos y cómo se aplican en el contexto de la predicción en mecanizado. Se completa esta sección con una revisión de aplicaciones recientes de las tecnologías, examinando la literatura existente sobre la predicción de parámetros de viruta y las aplicaciones de redes neuronales en el mecanizado.

La sección de "Metodología" detalla los métodos empleados en la investigación. Comienza con una introducción general a los enfoques metodológicos, seguida de la descripción de los materiales utilizados. Se explica cómo otros investigadores han obtenido los datos experimentales, la base para este estudio. Luego, se describe el desarrollo del modelo de red neuronal artificial, incluyendo el proceso de entrenamiento y ajuste del modelo. La evaluación y validación del modelo se detallan a continuación, mostrando cómo se ha probado su precisión y eficacia.

En la sección de "Resultados", se presentan y analizan los datos obtenidos. Se inicia con la distribución y análisis preliminar de los datos experimentales, seguido por un análisis ANOVA para identificar factores significativos. La normalización de los datos prepara el camino para el análisis predictivo, cuyos resultados se evalúan luego en términos de desempeño y validación de la red neuronal. Se realiza una comparación entre diferentes modelos predictivos, y se muestra la aplicación práctica de la predicción de viruta mediante Simulink.

Finalmente, el trabajo concluye con "Conclusiones y Líneas Futuras de Investigación". Esta sección sintetiza los hallazgos principales, destacando la importancia y las implicaciones del estudio. Se sugieren direcciones futuras para la investigación, enfatizando cómo los resultados pueden influir en la mejora de las técnicas de mecanizado y en el desarrollo de nuevas aplicaciones industriales.

2. Marco Teórico

2.1. Aleaciones ligeras

2.1.1. Aleaciones de aluminio

El aluminio es un metal excepcionalmente abundante en la corteza terrestre, siendo superado únicamente por el oxígeno y el silicio en términos de prevalencia. A pesar de su abundancia, no fue conocido en su forma metálica hasta tiempos relativamente recientes porque, a diferencia del hierro o el cobre, no existe en estado metálico puro en la naturaleza. Generalmente, se encuentra combinado con oxígeno formando óxido de aluminio, también conocido como alúmina, presente en arcillas y numerosos minerales.

Este metal se caracteriza por su color grisáceo y su capacidad para reflejar bien la radiación electromagnética del espectro visible y térmico. Es un excelente conductor eléctrico, con una conductividad que oscila entre 35 y 38 $m/(\Omega \text{ mm}^2)$, y térmico, con un rango de 80 a 230 $W/(m \cdot K)$. Además, el aluminio es un material ligero, con una densidad aproximada de 2700 kg/m^3 , y posee un bajo punto de fusión de 660 °C.

Desde el punto de vista mecánico, el aluminio es blando y maleable, situándose entre 2 y 4 en la escala de Mohs, y en su estado puro puede soportar una resistencia en tracción de 160-200 MPa. Su uso como elemento estructural es limitado debido a que es un metal blando. Para superar esta limitación, el aluminio se alea con otros metales, lo que mejora sus propiedades y

amplía su aplicabilidad en procesos de fundición y forja, así como en la extrusión del material y su uso en soldadura [7].

Entre las características más notables del aluminio, destacan:

- **No es tóxico:** Dada su no toxicidad, el aluminio es comúnmente utilizado para envolver y almacenar alimentos.
- **Facilidad de reciclado:** Reciclar aluminio es fundamental, ya que los costos asociados con el reciclaje de desechos de fabricación pueden ser significativos para las empresas, impactando su competitividad en el mercado.
- **Muy buenos acabados:** Se ve suave y brillante, el aluminio puede ser tratado para lograr una amplia variedad de colores y texturas.
- **Alta ductilidad:** El aluminio permite el uso de múltiples métodos de fabricación, facilitando la creación de productos variados en forma y tamaño, lo cual ayuda a mantener bajos los costos de producción. Aunque el aluminio puro es demasiado blando para algunas aplicaciones estructurales, al alearse con otros metales, sus propiedades mecánicas mejoran significativamente, permitiendo su uso en una mayor variedad de aplicaciones.
- **Material no ferromagnético:** Esta propiedad del aluminio es especialmente valiosa en aplicaciones dentro de la industria eléctrica y electrónica donde la interferencia magnética debe evitarse.
- **Resistencia a la corrosión:** Exhibe una resistencia sobresaliente contra la corrosión en una diversidad de entornos, incluyendo atmosféricos, acuosos y químicos, aunque necesita protección adicional en ambientes salinos.
- **Alta conductividad eléctrica y térmica:** Con una conductividad eléctrica que supera al cobre por peso y una excelente conductividad térmica, el aluminio es ideal para aplicaciones como conductores de alta tensión y componentes que manejan calor, como radiadores y evaporadores.
- **Relativo bajo costo:** Comparado con otros metales y aleaciones, el aluminio generalmente ofrece un costo más accesible.
- **Excelente relación peso-resistencia:** Con solo un tercio de la densidad de metales como acero, cobre o latón, ciertas aleaciones de aluminio ofrecen una resistencia mecánica incluso superior a algunos tipos de acero estructural.

Sin embargo, entre sus desventajas se encuentran:

- **Punto de fusión relativamente bajo:** Las aleaciones de aluminio se funden a aproximadamente 535°C, y aunque pueden soportar temporalmente temperaturas de hasta 300°C, su uso continuo a altas temperaturas está generalmente limitado a 150°C.

- **Módulo de elasticidad comparativamente bajo:** El módulo de elasticidad del aluminio es considerablemente inferior al de materiales competidores, siendo un 70% más bajo que el titanio y un 40% más bajo que la mayoría de los aceros estructurales.
- **Desafíos en la soldadura:** La soldadura de algunas aleaciones de aluminio puede requerir técnicas especializadas o métodos alternativos de ensamblaje, ya que no todas las aleaciones son igualmente aptas para ser soldadas de manera convencional.
- **Resistencia a la corrosión con limitaciones:** Aunque el aluminio puro y la mayoría de sus aleaciones muestran una excelente resistencia a la corrosión, algunas aleaciones específicas utilizadas en la industria aeroespacial pueden ser susceptibles a este fenómeno.

Las aleaciones de aluminio se clasifican según diferentes sistemas de nomenclatura dependiendo de la región o del organismo de normalización. En España, la normativa UNE (Una Norma Española) utiliza la letra "L" para indicar "aleación ligera", seguida de un conjunto de cuatro dígitos que especifican características particulares de la aleación. En el contexto europeo, este código es adoptado con un cambio menor, reemplazando "AA" por "EN AW" para alinearse con las normativas europeas. Por otro lado, el sistema UNS, desarrollado conjuntamente por la ASTM (American Society for Testing and Materials) y la SAE (Society of Automotive Engineers), utiliza un código que comienza con las letras "UNS" seguido de seis caracteres, compuesto por letras y varios dígitos que especifican el aleante principal, la composición exacta y, si aplica, el tratamiento térmico realizado [8].

En la siguiente tabla (Tabla 1) se presenta una clasificación detallada de diferentes series de aleaciones de aluminio, cada una con sus composiciones, propiedades mecánicas y aplicaciones principales:

Tabla 1. Clasificación y Características de las Aleaciones de Aluminio: Composición, Propiedades y Aplicaciones [9].

Serie	Elementos de Aleación (Elementos Principales en Negrita)	Resistencia a la Tracción (MPa)	Propiedades	Aplicaciones Principales
1000	Silicio (0.07 - 0.25%); Hierro (Fe, 0.07 - 0.40%); Cobre (Cu, 0.003 - 0.05%); Manganeso (0.01 - 0.05%); Magnesio (0.01 - 0.05%); Cromo (Cr, extremadamente pequeño a ninguno); Zinc (0.015 - 0.05%)	100 - 180	Resistencia a la corrosión y buena conductividad térmica.	Intercambiadores de calor y conductores.
2000	Silicio (0.20 - 1.3%); Hierro (Fe, 0.07 - 1.4%); Cobre (Cu, 1.8 - 6.8%) ; Manganeso (0.1 - 0.50%); Magnesio (0.05 - 1.80%); Cromo (Cr, 0.10%); Zinc (0.10 - 0.5%)	100 - 140	Mayor fuerza, Mayor dureza y Disminución de la resistencia a la corrosión.	Piezas aeroespaciales y de defensa.
3000	Silicio (0.20 - 1.1%); Hierro (Fe, 0.40-1.8%); Cobre (Cu, 0.05 - 2.2%); Manganeso (0.02 - 1.8%) ; Cromo (Cr, extremadamente pequeño - 0.02%); Zinc (0.05 - 1.8%)	110 - 210	Dúctil y conformable, Mayor resistencia a la corrosión.	Arquitectura y construcción.
4000	Silicio (0.20 - 21.5%) ; Hierro (Fe, 0.40-5.8%); Cobre (Cu, 0.05 - 5.4%) ; Manganeso (0.02 - 1.0%); Cromo (Cr, extremadamente pequeño - 0.02%); Zinc (0.05 - 1.8%)	100 - 260	Soldabilidad de mayor resistencia.	Alambre de soldadura.
5000	Silicio (0.08 - 0.4%); Hierro (Fe, 0.10 - 0.7%); Cobre (Cu, 0.05 - 0.35%); Manganeso (0.01 - 1.2%) ; Magnesio (0.1 - 5.2%); Zinc (0.15 - 0.8%)	160 - 420	Mayor resistencia y resistencia a la corrosión.	Aplicaciones marinas.
6000	Silicio (0.20 - 1.8%) ; Hierro (Fe, 0.15 - 1.7%); Cobre (Cu, 0.15 - 1.0%); Manganeso (0.10 - 0.5%); Cromo (0.05 - 0.40%); Zinc (0.05 - 0.25%)	100 - 260	Aleación soldable, Gran fuerza.	Arquitectura y construcción.
7000	Silicio (0.10 - 0.5%); Hierro (Fe, 0.12 - 1.4%); Cobre (Cu, 0.10 - 2.5%) ; Manganeso (0.05 - 0.8%); Cromo (0.04 - 0.35%); Zinc (0.05 - 8.2%)	150 - 305	Aleación de alta resistencia	Piezas aeroespaciales y de defensa.
8000	Silicio (0.20 - 1.1%); Hierro (Fe, 0.40-6.8%) ; Cobre (Cu, 0.05 - 2.2%) ; Manganeso (0.02 - 0.1%); Cromo (Cr, extremadamente pequeño - 0.02%); Zinc (0.05 - 1.8%)	100 - 260	Propiedades especiales	Propiedades especiales

La selección de materiales aeroespaciales es crucial en los ciclos de diseño de componentes y sistemas aeroespaciales. Afecta muchos aspectos del rendimiento de la aeronave desde la fase de diseño hasta la disposición final, incluyendo la eficiencia estructural, el rendimiento de vuelo, la carga útil, el consumo de energía, la seguridad y la fiabilidad, el costo del ciclo de vida, la reciclabilidad y la disposición final. Los requisitos críticos de los materiales estructurales aeroespaciales incluyen propiedades mecánicas, físicas y químicas, como alta resistencia, rigidez, durabilidad frente a la fatiga, tolerancia al daño; baja densidad, alta estabilidad térmica; alta resistencia a la corrosión y oxidación, así como criterios comerciales como costo, servicio y fabricabilidad. Además de cumplir con los requisitos básicos de servicio, la mejora de la eficiencia estructural en el diseño estructural aeroespacial se vuelve cada vez más crítica porque la aplicación de estructuras ligeras aporta beneficios al rendimiento de la aeronave, por ejemplo, mayor eficiencia energética, rendimiento de aceleración, carga útil, resistencia de vuelo, y reducción del costo del ciclo de vida y emisiones de gases de efecto invernadero.

Estudios previos [10] han indicado que la forma más efectiva de mejorar la eficiencia estructural es reducir la densidad (alrededor de 3 a 5 veces más efectiva en comparación con aumentar la rigidez o la resistencia), es decir, utilizando materiales ligeros. Los materiales estructurales aeroespaciales comerciales más comúnmente utilizados son aleaciones de aluminio, aleaciones de titanio, aceros de alta resistencia y compuestos, generalmente representando más del 90% del peso de los fuselajes.



Figura 1. Aleaciones de aluminio más usadas en la fabricación de aeronaves comerciales. Adaptada de [10].

a elección de una aleación específica se basa en las características que necesita la parte del avión en la que será utilizada. Como se muestra en la Figura 1, ciertas aleaciones de aluminio son preferidas para la construcción de diversas secciones de aeronaves comerciales.

Desde la década de 1920 hasta finales del siglo, las aleaciones de aluminio ligeras fueron los principales materiales estructurales de aviación (representando el 70%–80% del peso del fuselaje de la mayoría de las aeronaves civiles antes del 2000) y todavía juegan un papel importante. Desde mediados de la década de 1960 y 1970, la proporción de compuestos utilizados en estructuras aeroespaciales ha aumentado debido al desarrollo de compuestos de alto rendimiento. La siguiente Figura 2 ilustra la distribución de materiales para algunos diferentes productos de Boeing.

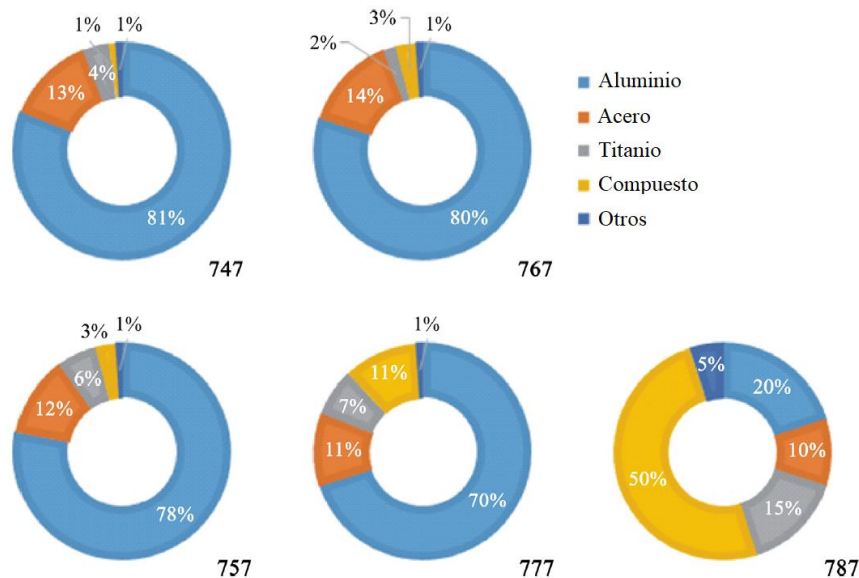


Figura 2. Distribución de materiales para una selección de productos Boeing. Adaptada de [10].

Por ejemplo, la aleación UNS A92024, que es típicamente sin tratamiento térmico o naturalmente envejecida (T3), ha sido tradicionalmente seleccionada para los fuselajes debido a su alta resistencia. Se ha logrado mejorar sus propiedades mediante un control más estricto de impurezas y aleantes, resultando en el desarrollo de variantes como las aleaciones UNS A92224, UNS A92324 y UNS A92524. Sin embargo, estos tipos tienen limitaciones en términos de resistencia a la corrosión, lo que a menudo requiere que estén revestidos. Además, la soldabilidad de estas aleaciones no es óptima, lo que lleva a que las uniones deban efectuarse mediante técnicas mecánicas. Esto ha promovido el uso de la serie 6000 (Al-Mg-Si), conocida por su mejor soldabilidad y resistencia a la corrosión, aunque sus propiedades mecánicas generales son inferiores [8].

En cuanto a la serie 7000 (Al-Zn), conocida por ofrecer la mayor resistencia mecánica de las aleaciones de aluminio, incluye variantes como UNS A97049, UNS A97050, y UNS A97075, entre otras. Estas aleaciones son altamente resistentes a la fractura, pero tienen problemas de soldabilidad que también requieren uniones mecánicas. La resistencia a la corrosión de esta serie es la más baja, especialmente cuando se trata con el método T6, lo cual aumenta el riesgo de corrosión bajo tensión (SCC). Para mitigar este riesgo, se han desarrollado aleaciones con tratamiento térmico T7, que, aunque disminuyen la resistencia mecánica, reducen la susceptibilidad a la corrosión [8].

2.1.2. Aleaciones de titanio

El titanio es un elemento metálico que destaca por tener la mayor proporción de dureza-densidad entre los metales, lo que subraya su relevancia en aplicaciones que requieren materiales robustos pero ligeros. Este metal, conocido por su fuerte resistencia y baja densidad, exhibe alta ductilidad en ambientes libres de oxígeno y un brillo blanco metálico característico. Con un punto de fusión de aproximadamente 1668 °C, el titanio es útil como metal refractario, adecuado para trabajos a altas temperaturas. Además, es paramagnético y posee baja conductividad eléctrica y térmica, lo que lo hace valioso en aplicaciones donde estas propiedades son deseables.

En términos de resistencia mecánica, las aleaciones comerciales de titanio, que típicamente contienen un 99.2% de pureza, exhiben una tensión de rotura comparable a la de las aleaciones comunes de acero, pero con una densidad considerablemente menor. A pesar de ser un 60% más denso que el aluminio, el titanio es dos veces más fuerte que la aleación de aluminio más común, 6061-T6. Algunas aleaciones avanzadas de titanio alcanzan tensiones de rotura superiores a los 1400 MPa. Sin embargo, es importante notar que el titanio pierde resistencia al calentarse por encima de los 430 °C. Aunque no es tan duro como algunos tipos de acero tratado, y su mecanizado requiere precauciones específicas para evitar uniones defectuosas, las estructuras de titanio, al igual que las de acero, tienen un límite de fatiga que asegura su durabilidad en aplicaciones prolongadas. Estas propiedades hacen del titanio un material fundamental en sectores como la aeroespacial y la biomédica, promoviendo continuamente los límites de la ingeniería moderna [11].

El titanio y sus aleaciones fueron reconocidos por primera vez por su potencial para aplicaciones de alto rendimiento, generalmente críticas en términos de peso, ya a finales de los años 1950. Sin embargo, la reactividad del titanio fundido y del titanio sólido y sus aleaciones a temperaturas elevadas fue una barrera significativa para su comercialización a gran escala, incluyendo el desarrollo de una capacidad de producción significativa. Hoy en día, estas barreras han sido en gran medida superadas, pero la reactividad intrínseca del titanio y sus aleaciones todavía requiere un procesamiento especial que añade un costo significativo al producto final,

independientemente de su uso previsto. Hay varios libros que contienen información extensa y detallada sobre el titanio y sus aleaciones [12].

Las aleaciones de titanio ofrecen múltiples ventajas en la fabricación de aeronaves y componentes de motores, que son particularmente notables frente a otros materiales más tradicionales como el acero o las aleaciones de aluminio. A continuación, se resumen las ventajas principales:

- **Ahorro de peso:** Las aleaciones de titanio presentan una alta resistencia específica, siendo aproximadamente un 50% menos densas que el acero, lo cual se traduce en un ahorro significativo de peso. Por ejemplo, el cambio de acero por titanio en el tren de aterrizaje del Boeing 777 puede ahorrar hasta 270 kg.
- **Limitaciones de espacio:** Su baja densidad comparada con el acero, pero con una resistencia específica equivalente y excelente resistencia a la corrosión, permite el uso de titanio en aplicaciones estructurales donde los componentes de aleación de aluminio serían demasiado grandes para los espacios disponibles.
- **Compatibilidad con composites:** Las aleaciones de titanio son compatibles con composites de matriz de polímero y fibra de carbono, evitando problemas de expansión térmica y corrosión electroquímica en ambientes húmedos, lo que es esencial para componentes que operan en rangos de temperatura de -55 °C a $+55\text{ °C}$.
- **Bajo módulo de elasticidad:** El módulo elástico del titanio es aproximadamente la mitad que el del acero. Esto es ventajoso en la fabricación de resortes donde se requiere menos espirales para lograr la misma capacidad de carga, reduciendo el peso hasta en un 70%.
- **Resistencia en temperaturas elevadas y criogénicas:** Las aleaciones de titanio se utilizan en secciones del motor que requieren resistencia a altas temperaturas, como los compresores de alta presión y las turbinas de baja presión, ofreciendo un rendimiento superior en temperaturas que van desde los -250 °C hasta más de 500 °C . Esto permite aplicaciones en motores de avión y cohetes que experimentan condiciones extremas de temperatura.

La aleación Ti6Al4V, también conocida como Grado 5 de titanio, es una de las aleaciones de titanio más utilizadas, especialmente en la industria aeroespacial. Aquí están algunos de sus usos principales en aeronáutica:

- **Componentes estructurales de aeronaves:** Debido a su excelente combinación de resistencia, ligereza y resistencia a la corrosión, Ti6Al4V se utiliza ampliamente en la fabricación de componentes estructurales críticos. Estos incluyen elementos del

fuselaje, marcos, soportes y otras partes estructurales que requieren materiales robustos pero ligeros.

- **Motores de aeronaves:** Esta aleación se emplea en diversas partes de los motores de aviones, incluyendo discos de compresores, palas, carcasas del motor y componentes de los sistemas de escape. Su resistencia a altas temperaturas y a la fatiga son cruciales para estas aplicaciones.
- **Tren de aterrizaje:** Las propiedades de alta resistencia y resistencia a la corrosión del Ti6Al4V lo hacen ideal para partes del tren de aterrizaje, donde estos factores son esenciales para la seguridad y durabilidad.
- **Componentes hidráulicos:** Esta aleación también se utiliza en sistemas hidráulicos dentro de la aeronave debido a su buena compatibilidad con fluidos hidráulicos y su capacidad para soportar presiones y temperaturas variadas sin corroerse.

Al evaluar la aleación Ti6Al4V, es crucial considerar sus limitaciones junto a sus ventajas. A pesar de su destacada resistencia y ligereza, este material presenta desafíos significativos:

- **Alto costo:** Ti6Al4V es considerablemente más caro que muchos otros metales utilizados en la industria, como el acero o el aluminio. Este alto costo se debe a la dificultad en su procesamiento y a la elaboración del material base, el titanio.
- **Dificultad en el mecanizado:** Aunque es más maleable que otros grados de titanio, Ti6Al4V sigue siendo difícil de mecanizar comparado con el acero y el aluminio. Requiere herramientas de corte especializadas, velocidades de corte más bajas y técnicas avanzadas para manejar el calor generado durante el mecanizado, lo que puede aumentar el tiempo y el costo de producción.
- **Sensibilidad a altas temperaturas:** Aunque Ti6Al4V puede operar de manera efectiva a temperaturas moderadamente altas, su resistencia y rigidez pueden degradarse cuando se expone a temperaturas superiores a los 430 °C. Esto limita su uso en aplicaciones que requieren resistencia a temperaturas extremadamente altas.
- **Problemas de soldadura:** Soldar Ti6Al4V puede ser desafiante debido a su alta reactividad a temperaturas elevadas. Puede absorber gases como oxígeno, nitrógeno e hidrógeno durante el proceso de soldadura, lo que puede resultar en soldaduras frágiles y defectuosas.
- **Susceptibilidad al agrietamiento por corrosión bajo tensión:** Aunque generalmente resistente a la corrosión, bajo ciertas condiciones ambientales y químicas, Ti6Al4V puede experimentar agrietamiento por corrosión bajo tensión, lo que compromete la integridad estructural de los componentes fabricados.

- **Limitada resistencia a la abrasión:** Comparado con otros materiales como algunos aceros endurecidos, Ti6Al4V tiene una menor resistencia a la abrasión, lo que puede ser una limitación en aplicaciones que requieren alta resistencia al desgaste.

En resumen, la aleación Ti6Al4V es esencial en la industria aeroespacial, ofreciendo un balance óptimo entre resistencia y ligereza. Su fiabilidad y versatilidad aseguran su presencia continua en la evolución del diseño y la funcionalidad de las aeronaves modernas, marcando estándares de excelencia en materiales avanzados [13].

2.2. Mecanizado en seco

La fabricación es un pilar fundamental en el desarrollo industrial, abarcando diversos procesos como la fundición, el conformado y, notablemente, el mecanizado. Este último es esencial para conferir la forma final, las propiedades superficiales y la precisión dimensional requeridas en las piezas fabricadas.

Tradicionalmente, el mecanizado se ha llevado a cabo mediante la eliminación de material con herramientas afiladas en forma de cuña a través de la deformación plástica y el posterior cizallamiento del material, proceso durante el cual la mayor parte de la energía se convierte en calor, afectando tanto a la herramienta como a la pieza de trabajo.

Para combatir el calor generado, se ha recurrido históricamente al uso de fluidos de corte que, si bien son efectivos en la refrigeración y la eliminación de residuos del mecanizado, conllevan considerables desventajas económicas y ecológicas. El alto consumo global de estos fluidos, llegando a superar los dos mil millones de litros [14] y la necesidad de su tratamiento y disposición adecuados plantean serios desafíos ambientales y elevan los costes de producción. Además, los riesgos para la salud de los trabajadores, derivados contacto con la piel y la inhalación de fluidos de corte cargados de microorganismos, biocidas y partículas metálicas tóxicas es una de las principales causas de enfermedades profesionales de los trabajadores industriales [15].

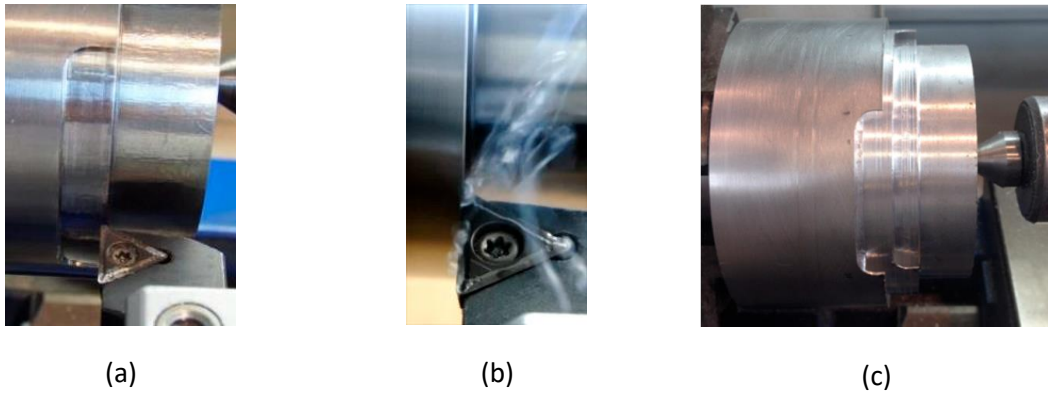


Figura 3. (a) Disposición ortogonal de corte; (b) instancia inicial de mecanizado; (c) muestra para varias pruebas de corte [16].

A la hora de aplicar el mecanizado en seco, como muestra la Figura 3, nos encontramos las siguientes desventajas técnicas:

- **Incremento de Temperatura en Herramientas y Piezas:** Las altas temperaturas generadas pueden reducir la dureza de las herramientas y alterar la precisión dimensional de las piezas trabajadas. Esto resulta en deformaciones, desafilado y fallo de las herramientas.
- **Alteración de la Integridad de la Pieza:** El calentamiento afecta la integridad y las propiedades superficiales de las piezas [17], especialmente en aleaciones de aluminio que absorben mucho calor, llevando a errores en la calidad y forma de las piezas mecanizadas [18].
- **Desafíos con Materiales Difíciles:** Mientras algunos materiales se mecanizan fácilmente en seco, otros, especialmente aquellos difíciles de cortar, presentan complicaciones significativas bajo estas condiciones debido a sus características inherentes como bajo punto de fusión y alta dilatación térmica [19].
- **Problemas de Manejo de Calor:** Una porción sustancial del calor generado durante el mecanizado se transfiere a la herramienta y a la pieza, lo cual puede afectar adversamente la operación y el resultado del mecanizado. La energía necesaria para deformar plásticamente el material de la pieza y desplazarlo sobre la cara de desprendimiento de la herramienta se transforma en gran parte en calor. Gran parte de este calor es arrastrado junto con la viruta, pero una parte se destina a calentar la herramienta y la pieza que se está mecanizando.

Por otra parte, la técnica presenta ventajas:

- **No se necesita fluido de corte:** Elimina la necesidad de aplicar y gestionar fluidos de corte, reduciendo la complejidad y la contaminación asociada con su uso y disposición.
- **Reducción de costos operativos:** Al eliminar los fluidos de corte, se reducen los costos asociados con la compra, el manejo y la eliminación de estos materiales.
- **Capacidad para mecanizar materiales convencionales:** Permite el mecanizado de una amplia variedad de materiales sin necesidad de fluidos, lo que simplifica las operaciones y reduce la preparación y limpieza.
- **Impacto ambiental mejorado:** Al no utilizar fluidos de corte, se reduce significativamente el impacto ambiental, evitando la contaminación del agua y del suelo y mejorando la sostenibilidad general del proceso.
- **Facilidad para reciclar las virutas:** Las virutas generadas en el mecanizado en seco están libres de contaminantes como los fluidos de corte, lo que facilita su reciclaje y reutilización sin procesos de limpieza adicionales.

El mecanizado en seco emerge como una técnica prometedora, alineada con los principios de fabricación sostenible. Esta técnica elimina el uso de fluidos de corte, reduciendo los riesgos para la salud y el impacto ambiental, a la vez que ofrece un potencial considerable para disminuir los costos operativos. A pesar de los desafíos que presenta en términos de calidad superficial, vida útil de la herramienta y manejo de virutas, el desarrollo continuo de nuevas tecnologías y materiales para herramientas está permitiendo superar estas barreras.

2.3. Formación de la viruta en procesos de mecanizado en seco

2.3.1. Definición de viruta

La viruta, en el contexto de la ingeniería de fabricación, constituye el residuo material generado como consecuencia de la interacción cinemática y termodinámica entre una herramienta de corte y el sustrato de trabajo durante operaciones de mecanizado. Este fenómeno, crucial para la evaluación de la eficacia de los procesos de mecanizado, no solo sirve como un indicador palpable de la materialización del corte, sino que también encapsula una serie de variables críticas inherentes al proceso, tales como las condiciones térmicas en la zona de corte, la integridad superficial de la pieza procesada, y las características mecánicas de la herramienta empleada. La formación de la viruta, por tanto, es una manifestación física de la interacción compleja entre la resistencia del material trabajado y las fuerzas aplicadas a través de la geometría optimizada de la herramienta de corte.

No obstante, un elemento crítico que influencia de manera sustancial la eficiencia operativa en los talleres de máquinas es la gestión de las virutas generadas durante el mecanizado. Configuraciones de viruta inadecuadas poseen el potencial de provocar lesiones a los operarios, además de acarrear costos adicionales atribuibles a la producción de piezas defectuosas, pérdida de tiempo de mecanizado, y retrasos en la entrega de piezas finalizadas.

Se ha demostrado empíricamente que un manejo efectivo de las virutas no solo contribuye a la fiabilidad del proceso de mecanizado, sino que también fomenta la producción de superficies mecanizadas de alta calidad y asegura tanto la seguridad operativa como la protección del equipamiento y herramientas involucradas. En tiempos recientes, el control de virutas ha adquirido una relevancia creciente, especialmente dada la tendencia hacia un entorno de mecanizado completamente automatizado.

2.3.2. Mecánica de la formación de viruta

En la mecánica de formación de la viruta, la elección del material de la herramienta y su geometría, así como los parámetros de corte y sus condiciones, son cruciales para la determinación de la viruta.

2.3.2.1. Herramientas de corte

Las herramientas de corte deben ser extremadamente duras para resistir el desgaste y la abrasión durante el mecanizado. Sin embargo, la dureza no debe comprometer la tenacidad de la herramienta, ya que también debe soportar el impacto y las cargas cíclicas sin fracturarse. Además, el material de la herramienta debe mantener su dureza a altas temperaturas, ya que el proceso de corte genera mucho calor. Materiales como el acero rápido (HSS), metales duros (WC) con o sin recubrimiento, carburo cementado, cerámicas, PCD (diamante policristalino) y CBN (nitruro de boro cúbico) son comunes por sus propiedades de dureza y resistencia a altas temperaturas, además de ofrecer resistencia a la abrasión y a la adherencia del material de la pieza sobre la herramienta. La adopción de geometrías de herramientas específicas y recubrimientos avanzados como el nitruro de boro cúbico (CBN), nitruro de boro cúbico policristalino (PCBN), diamante policristalino (PCD) y varios tipos de revestimientos como el nitruro de titanio (TiN), carbonitruro de titanio (TiCN) y nitruro de titanio y aluminio (TiAlN) mejoran significativamente la resistencia al desgaste y prolongan la vida útil de las herramientas.

El recubrimiento de WC/C (carburo de tungsteno/carbono) y diamante se recomienda para el mecanizado en seco de aleaciones de aluminio. Además, para el mecanizado de aleaciones de titanio, se ha utilizado específicamente los carburos cementados de grado recto como WC-Co (carburo de tungsteno/cobalto). Se ha hecho uso de este material en la extracción de la viruta debido a sus excelentes propiedades, incluyendo dureza, resistencia al desgaste, resistencia al calor y tenacidad, que son esenciales para los desafíos presentados por estas

aleaciones [20]. Se presenta a continuación (Figura 4) algunos de los materiales de corte más utilizados.

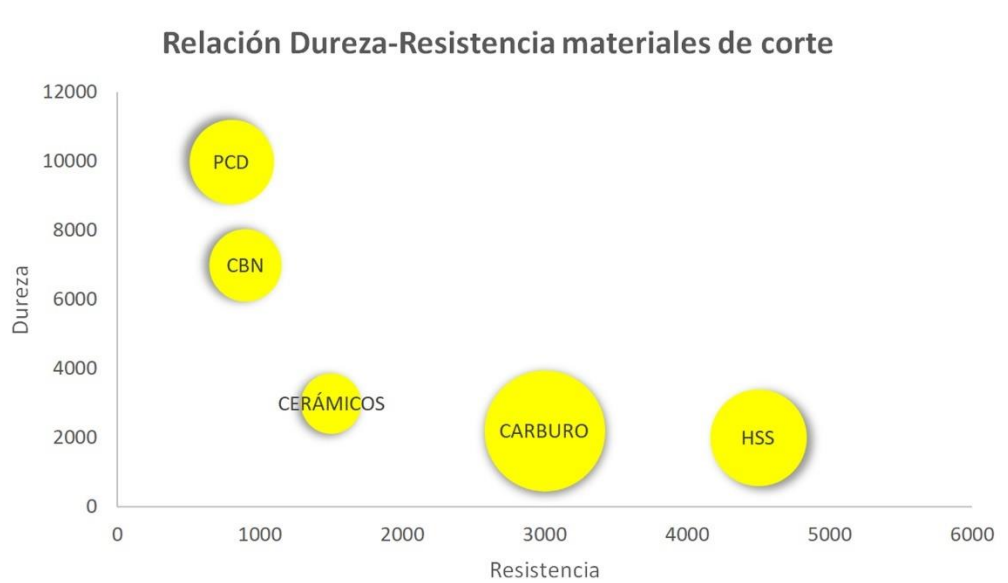


Figura 4. Relación dureza-resistencia de materiales de corte [21].

2.3.2.2. Influencia de la geometría de la herramienta de corte en la formación de virutas y fenómenos de acumulación

En el torneado, la geometría de la herramienta de corte juega un papel fundamental en la formación de la viruta y también influye significativamente en la aparición de fenómenos como la formación de capas acumuladas (BUL) y bordes acumulados (BUE). Un ángulo de ataque positivo tiende a curvar la viruta, facilitando su evacuación y es ideal para materiales blandos para evitar virutas largas y continuas; mientras que un ángulo negativo produce virutas más cortas y fragmentadas, mejorando la resistencia de la arista de corte y siendo preferible para materiales duros, ayudando así a reducir problemas de evacuación de viruta.

El ángulo de desprendimiento también juega un papel crítico al reducir la fricción entre la herramienta y la pieza. Un mayor ángulo de desprendimiento permite un corte más limpio, disminuyendo la generación de calor en la zona de contacto. Esto es crucial para evitar la formación del BUL y BUE (Figura 5). El BUL, o capa acumulada, es una capa de material de la pieza de trabajo que se fusiona temporalmente en la superficie de la herramienta bajo condiciones de alta temperatura y presión. Aunque puede proporcionar una barrera protectora contra el desgaste, su presencia irregular puede deteriorar la calidad del acabado de la pieza y afectar la precisión del mecanizado al alterar las dimensiones reales de la herramienta.

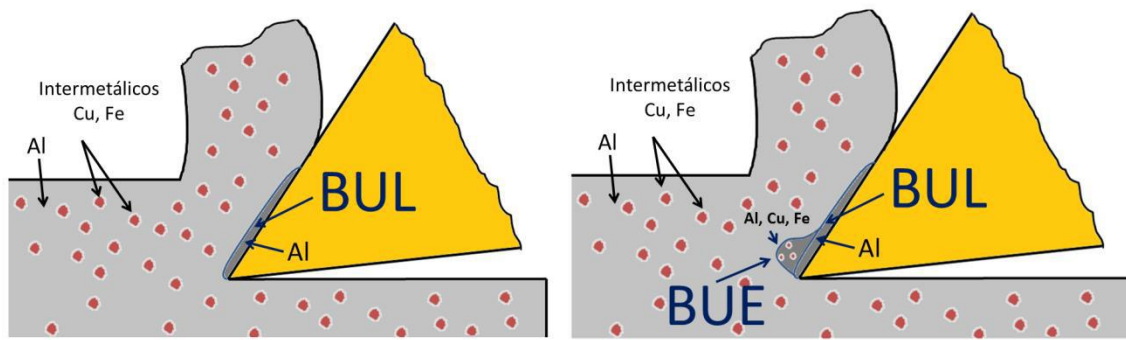


Figura 5. Mecanismos de formación de BUE y BUL en el torneado en seco de la aleación UNS A92024 con herramientas de WC-Co y recubrimiento de TiN [22].

Por otro lado, el BUE se refiere a la acumulación de material en el borde de corte de la herramienta. Este fenómeno ocurre cuando partes del material de trabajo se adhieren al borde de corte durante el mecanizado, alterando la geometría de corte efectiva. Como resultado, puede provocar un acabado superficial pobre, mayores fuerzas de corte y un aumento en el desgaste de la herramienta. La gestión eficaz de estos fenómenos depende en gran medida de un diseño adecuado de la herramienta y de los parámetros de corte, así como de una adecuada lubricación y refrigeración durante el proceso.

Además, un mayor radio en la punta de la herramienta ayuda a distribuir más uniformemente la carga de corte, lo que no solo facilita un corte más estable, sino que también reduce el estrés tanto en la herramienta como en la viruta, minimizando aún más las condiciones que favorecen la formación de BUL y BUE. Este manejo cuidadoso de la geometría de la herramienta y las condiciones de corte optimiza el proceso de mecanizado, mejorando la calidad del resultado final y extendiendo la vida útil de la herramienta.

2.3.2.3. Parámetros de corte

Los parámetros de corte como la velocidad de corte (v_c), el avance (f) y la profundidad de corte (a_p) deben ser cuidadosamente seleccionados ya que juegan un papel fundamental en la formación y características de la viruta, además de influir directamente en la eficiencia del proceso de mecanizado y la calidad del producto final.

La velocidad de corte tiene un impacto significativo en la temperatura generada durante el mecanizado. Velocidades más altas pueden mejorar el acabado superficial al reducir la formación de rebabas y proporcionar un corte más limpio, pero también aumentan la temperatura en la zona de corte. Esto puede acelerar el desgaste de la herramienta debido a la degradación térmica del material de la herramienta y de la viruta misma, lo cual puede llevar a cambios en las características de la viruta, haciéndola más frágil y difícil de manejar.

El avance, o la velocidad a la que la herramienta avanza en el material, afecta directamente el tamaño de la viruta. Un avance más rápido aumenta la tasa de remoción de material, produciendo virutas más gruesas y potencialmente más difíciles de evacuar. Esto puede empeorar el acabado superficial y aumentar el calor generado, lo cual a su vez puede influir en la formación de virutas anormales y aumentar la probabilidad de que ocurran defectos en la superficie mecanizada.

La profundidad de corte determina cuánto material se remueve en cada pasada y tiene una influencia directa en la carga aplicada a la herramienta y la máquina. Profundidades de corte más grandes pueden ser eficientes en términos de remoción de material, pero también imponen mayores fuerzas sobre la herramienta y la pieza de trabajo. Esto no solo afecta la estabilidad del proceso, sino que también puede alterar la forma en que la viruta se forma y se rompe, potencialmente conduciendo a la formación de virutas más grandes y difíciles de controlar.

2.3.2.4. Evacuación de calor

La lubricación y refrigeración juegan un papel esencial en el torneado, ayudando a reducir el calor generado y a evacuar las virutas de la zona de trabajo, lo cual prolonga la vida útil de la herramienta y mejora el acabado de la superficie. Además, la configuración de la máquina, especialmente la rigidez y estabilidad, es fundamental para mantener la precisión de las dimensiones de la pieza y minimizar las vibraciones, que pueden deteriorar la calidad del acabado y reducir la vida útil de la herramienta.

Como se ve en la Figura 6, el calor generado en el proceso de corte es transferido a los diferentes agentes que componen el mecanizado, pieza mecanizada, viruta y herramienta.

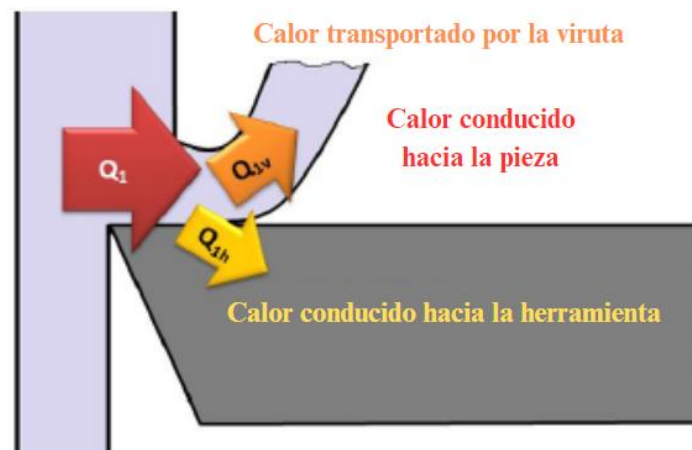


Figura 6. Distribución del calor en el proceso de corte ortogonal. Adaptada de [22].

En el mecanizado, la forma y características de la viruta varían ampliamente, dependiendo de varios factores como el material de la pieza, la geometría de la herramienta y los parámetros de corte. La comprensión de estos tipos puede ayudar a optimizar los procesos y mejorar la seguridad y eficiencia en la producción.

2.3.2.5. Forma de la viruta

La morfología de la viruta en procesos de mecanizado está influenciada principalmente por los parámetros de corte como la velocidad, el avance y la profundidad, además de la geometría de la herramienta, su posición respecto a la pieza, y las propiedades del material mecanizado.

Se han establecido diferentes criterios de clasificación de la viruta, basados en su deformación, fractura, tamaño y forma. La viruta puede presentar dos superficies distintas: una brillante, en contacto con la herramienta, y otra irregular y áspera, que no ha estado en contacto.

Se identifican principalmente tres tipos de viruta, tal y como se muestra en la Figura 7:

- **Fragmentada:** Se forma bajo alta fuerza, baja velocidad de corte, y alta fricción, resultando en superficies mecanizadas irregulares, pero es fácil de evacuar y reciclar.
- **Continua:** Resulta de la cizalladura pura en materiales dúctiles bajo condiciones controladas de corte, asociada a buenos acabados superficiales, aunque puede causar problemas como la formación de "nidos de viruta" que afectan la operación.
- **Segmentada:** Aparece típicamente en el mecanizado de materiales con baja conductividad térmica como las aleaciones de titanio, afectando negativamente el acabado superficial y acelerando el desgaste de la herramienta.

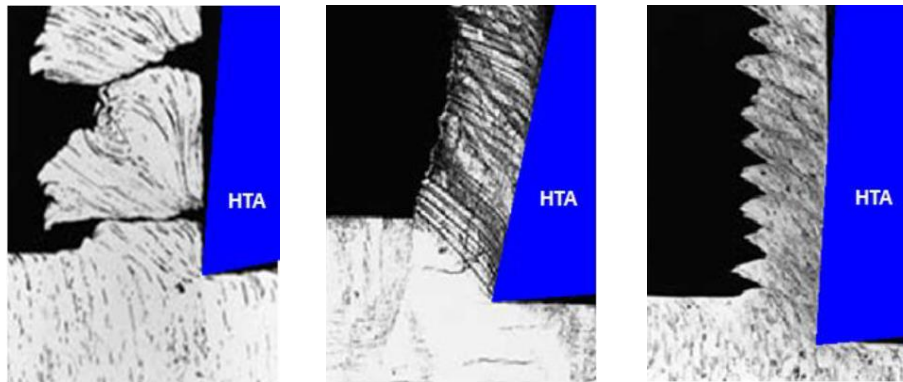


Figura 7. Tipos de viruta: (a) Fragmentada; (b) Continua; (c) Segmentada [22].

Otro criterio habitual para la clasificación de la viruta reside en la norma ISO 3685: planas, tubulares, espirales, helicoidales, helicoidales cónicas, arqueadas, fragmentadas y aciculares. Cada tipo de viruta está influenciado por las características del material, la geometría de la herramienta y las condiciones del corte.

- Las virutas **planas** se forman cuando el material mecanizado genera un flujo continuo y relativamente uniforme de material arrancado. Estas virutas suelen

ser comunes en materiales blandos o bajo condiciones de corte óptimas que permiten un desprendimiento uniforme.

- Las virutas **tubulares** presentan una forma hueca, resultando de materiales que tienden a enroscarse al ser cortados debido a una combinación de la geometría de la herramienta y la ductilidad del material. Esta forma es favorable en ciertos procesos por su fácil evacuación.
- Las **espirales** se producen cuando la viruta toma una forma enroscada con un perfil plano o cónico. Este tipo de viruta es común en materiales que permiten un corte más controlado y son típicas en operaciones donde el ángulo de corte favorece la formación de espirales suaves y controladas.
- Las virutas **helicoidales** presentan un patrón de espiral más pronunciado, similar a un resorte. Son comunes cuando la herramienta de corte posee un ángulo de desprendimiento que facilita esta forma, y son deseables por su facilidad de manejo durante el proceso de corte.
- Las **helicoidales cónicas** son similares a las helicoidales normales, pero con un perfil más estrecho y cónico. Estas virutas suelen formarse en materiales con una dureza media y bajo condiciones de corte que promueven una trayectoria de viruta más estrecha y definida.
- Las virutas **arqueadas** se caracterizan por tener una forma curva y suelta, sin la continuidad de una espiral completa. Estas virutas ocurren cuando hay variaciones menores en el corte o en materiales menos dúctiles, que no permiten un enrollamiento completo de la viruta.
- Las virutas **fragmentadas** son pequeñas piezas irregulares y quebradizas que se forman típicamente en materiales frágiles o en condiciones de corte inestables. Este tipo de viruta es común en el mecanizado de materiales duros o quebradizos, donde la viruta no puede formarse de manera continua.
- Finalmente, las virutas **aciculares** tienen una forma delgada y alargada, parecida a agujas. Este tipo de viruta es común en el mecanizado de materiales duros o cuando las condiciones de corte son tales que se generan fragmentos delgados y dispersos, que pueden ser difíciles de manejar y requieren una evacuación cuidadosa.

A continuación, se representan los tipos de virutas descritos anteriormente (Figura 8):

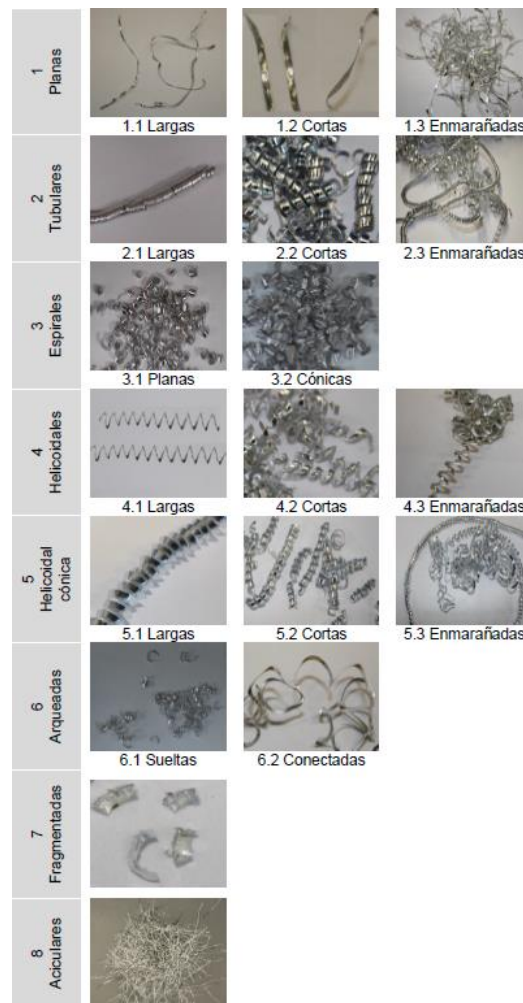


Figura 8. Tipología de viruta según ISO 3685 [8].

2.4. Redes neuronales artificiales y su aplicación en mecanizado

2.4.1. Introducción

El análisis de la viruta generada durante procesos de mecanizado proporciona *insights* críticos sobre la naturaleza del material y la eficiencia del proceso. Estudiar la viruta nos informa sobre cómo se deforma el material bajo diferentes condiciones de corte, la energía consumida en el proceso, y las propiedades físicas del material, como su ductilidad, fragilidad y conductividad térmica. Estos factores son esenciales para optimizar los parámetros de corte y mejorar la calidad del producto final.

En este contexto, el grupo de investigación iFAB “Ingeniería de Fabricación” (TEP933) de la Universidad de Málaga ha llevado a cabo estudios extensivos sobre la viruta de aleaciones ligeras como el aluminio y el titanio. A través de estas investigaciones, han desarrollado modelos

de regresión, tanto polinómicos como potenciales, que han proporcionado una base sólida para predecir y entender el comportamiento del material durante el mecanizado. Sin embargo, a pesar de los avances alcanzados con estos modelos, existe un potencial significativo para mejorar y refinar estos estudios mediante técnicas avanzadas de aprendizaje automático, como por ejemplo, el uso de las redes neuronales.

2.4.2. ¿Qué es una red neuronal artificial?

Una Red Neuronal Artificial es un modelo computacional inspirado en las redes neuronales biológicas del cerebro humano (Figura 9). Este modelo se utiliza para simular el modo en que el cerebro procesa la información, permitiendo que las máquinas aprendan y tomen decisiones de una manera que imita la cognición humana. El concepto de RNA se origina en el intento de entender los procesos cerebrales y crear sistemas que puedan realizar tareas de reconocimiento de patrones, procesamiento de lenguaje natural, y otras tareas que normalmente requieren inteligencia humana.

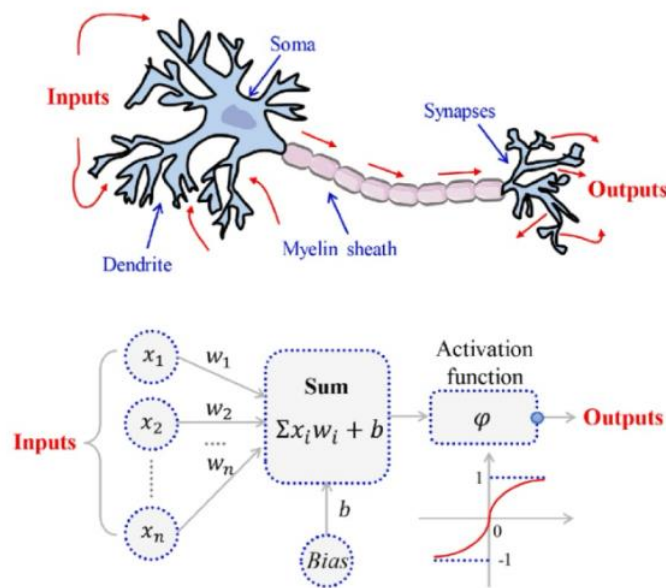


Figura 9. Analogía entre neuronas biológicas y neuronas artificiales en Redes Neuronales [23].

Las RNA fueron propuestas inicialmente en la década de 1940 por Warren McCulloch y Walter Pitts, quienes desarrollaron un modelo matemático de neuronas biológicas. Su trabajo estableció el fundamento para el desarrollo de algoritmos y arquitecturas que imitan la estructura y función del cerebro, llevando al desarrollo de técnicas de aprendizaje profundo y otras metodologías avanzadas en inteligencia artificial.

2.4.3. Componentes clave de una red neuronal artificial

Las Redes Neuronales Artificiales están compuestas por varios componentes clave que trabajan juntos para procesar y aprender de los datos:

- **Neuronas:** En una RNA, una "neurona" o unidad, es un nodo computacional que recibe y procesa señales. Cada neurona recibe entradas, las procesa, y produce una salida. La estructura de una neurona en una RNA incluye entradas (que pueden ser señales de datos o salidas de otras neuronas), una función de activación, y una salida.
- **Conexiones:** Las neuronas en una red están interconectadas por líneas que representan las sinapsis en un cerebro biológico. Estas conexiones son el medio por el cual se transmite la información de una neurona a otra.
- **Pesos:** Cada conexión entre neuronas en una RNA tiene un peso asociado. Los pesos determinan la importancia y la influencia de una señal de entrada en la señal de salida de una neurona. El ajuste de estos pesos se hace a través de un proceso iterativo conocido como **entrenamiento**, durante el cual la red aprende de los datos proporcionados.
- **Funciones de Activación:** Una función de activación en una neurona define cómo se calcula la salida a partir de las entradas sumadas y los pesos. La función de activación introduce una no linealidad en el procesamiento, permitiendo a la red aprender y modelar relaciones complejas entre los datos de entrada y salida. Ejemplos comunes de funciones de activación incluyen la sigmoidea, tanh, y ReLU (*Rectified Linear Unit*).

2.4.4. Arquitectura de las redes neuronales artificiales

Las arquitecturas de redes neuronales consisten en la organización y disposición de las capas y conexiones entre las neuronas dentro de la red. Estas arquitecturas determinan cómo se procesa la información dentro de la red, y pueden ser diseñadas para tareas específicas como clasificación, predicción, segmentación, entre otras. Las capas son los bloques fundamentales de las redes neuronales y varían según el tipo de red y la tarea a realizar.

Capa de entrada: La principal función de la capa de entrada es recibir los datos iniciales que serán procesados por la red. Esta capa no realiza ningún tipo de transformación o procesamiento de los datos; su rol es simplemente pasar la información tal como está a la siguiente capa de la red. En otras palabras, la capa de entrada actúa como un canal que introduce los datos al sistema. Las características de la capa de entrada están directamente ligadas al tipo y la dimensionalidad de los datos que se están utilizando. Cada neurona en la capa de entrada corresponde a una característica o dimensión del conjunto de datos. Por ejemplo, si el modelo está diseñado para procesar imágenes de 28x28 píxeles en escala de grises, la capa

de entrada tendría 784 neuronas ($28 \times 28 = 784$), cada una representando la intensidad de un píxel específico.

El funcionamiento de esta capa es bastante directo. Cuando los datos son introducidos en la red, cada neurona en la capa de entrada toma el valor correspondiente a una característica de los datos. En el caso de un conjunto de datos tabular, cada columna del conjunto de datos podría estar representada por una neurona diferente. En una tarea de procesamiento de lenguaje natural, cada palabra o token podría estar representado en forma de un vector que se asigna a la capa de entrada.

La capa de entrada no realiza cálculos ni aplica funciones de activación. Su propósito es puramente de transmisión: capturar los datos de entrada y transferirlos a la siguiente capa, que podría ser una capa oculta donde comienza el procesamiento real de la información.

Capas ocultas: Las capas ocultas reciben la información desde la capa de entrada o desde capas ocultas anteriores. Las capas ocultas pueden variar en número y tamaño, dependiendo del diseño de la red y de la complejidad de la tarea. Cada capa oculta está compuesta por varias neuronas, y cada neurona está conectada a las neuronas de la capa anterior y de la capa siguiente mediante pesos. Estos pesos se ajustan durante el proceso de entrenamiento de la red para minimizar el error en las predicciones.

Las funciones de activación, como ReLU, sigmoide o tanh, se aplican en las capas ocultas para introducir no linealidad en la red. Esta no linealidad es crucial para permitir que la red aprenda y modele relaciones complejas entre las variables de entrada y salida. Sin funciones de activación, la red sería simplemente una combinación lineal de los datos de entrada, lo que limitaría gravemente su capacidad para resolver problemas complejos.

El funcionamiento de las capas ocultas puede explicarse en términos de cómo las neuronas en estas capas reciben las entradas de la capa anterior, las transforman y luego envían las salidas a la siguiente capa. Cada neurona toma un conjunto de entradas, las multiplica por sus respectivos pesos, suma estos productos, añade un sesgo y luego aplica una función de activación para producir una salida. Esta salida se convierte en la entrada para las neuronas en la siguiente capa.

Durante el entrenamiento de la red, los pesos y sesgos de cada conexión en las capas ocultas se ajustan usando algoritmos de optimización como el descenso de gradiente. Este proceso se basa en minimizar una función de pérdida que mide la discrepancia entre las predicciones de la red y los valores reales de los datos de entrenamiento. A medida que el modelo se entrena, las capas ocultas se ajustan para capturar patrones cada vez más complejos y específicos en los datos.

Capa de salida: La capa de salida en una red neuronal es la encargada de generar la predicción final o el resultado basado en la información que ha sido procesada por las capas

anteriores. Es la última etapa del modelo y su función principal es transformar la información en un formato útil para la tarea específica que se está realizando, ya sea clasificación o regresión.

En los problemas de clasificación, la capa de salida produce probabilidades que indican a qué clase pertenece la entrada. Para la clasificación binaria, suele tener una sola neurona con una función de activación sigmoide, que genera un valor entre 0 y 1. En la clasificación multiclase, la capa de salida cuenta con tantas neuronas como clases posibles, y utiliza la función *softmax* para convertir las salidas en probabilidades que suman 1.

En los problemas de regresión, la capa de salida genera valores continuos, normalmente sin una función de activación o utilizando una función lineal, lo que permite que los resultados tomen cualquier valor en un rango específico. Se presenta en la siguiente figura (Figura 10) la disposición de las capas en una red neuronal:

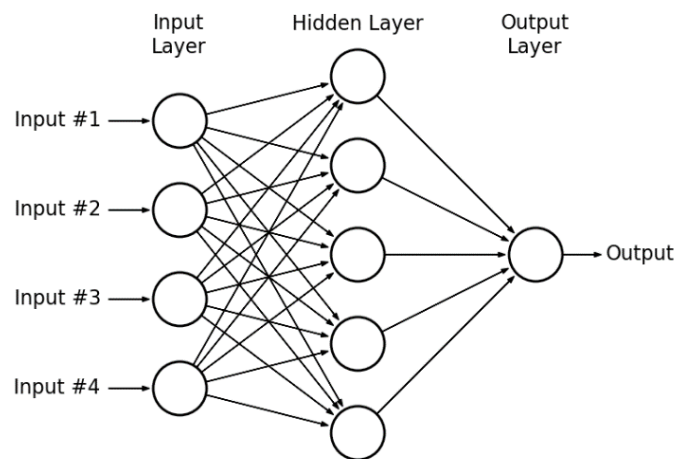


Figura 10. Capas de una Red Neuronal Artificial [24].

2.4.5. Tipologías de las redes neuronales artificiales

Las redes neuronales artificiales se clasifican en una amplia variedad de tipos, cada uno diseñado para abordar problemas específicos y optimizar el rendimiento en diferentes tareas. A continuación, se describen algunos de los tipos más importantes:

- **Redes Neuronales *Feedforward* (FNN):** son el tipo más simple y básico de redes neuronales. En estas redes, la información fluye en una sola dirección, desde la capa de entrada, a través de una o más capas ocultas, hasta la capa de salida. No existen ciclos o retroalimentaciones, lo que las hace relativamente simples de entrenar. Son utilizadas en una amplia variedad de tareas, como clasificación, regresión y predicción. Aunque hoy en día las redes más profundas y complejas han ganado popularidad, las FNN siguen siendo una base importante en el campo del aprendizaje automático.

- **Redes Neuronales Convolucionales (CNN):** son ampliamente utilizadas en tareas de visión por computadora, como el reconocimiento de imágenes, la clasificación de objetos, y la detección de patrones visuales. Estas redes están diseñadas para procesar datos en múltiples dimensiones (como imágenes), aplicando filtros (convoluciones) que extraen características relevantes a medida que la información avanza a través de las capas. Las CNN han demostrado ser extremadamente eficaces en aplicaciones que requieren la identificación de patrones espaciales en datos visuales.
- **Redes Neuronales Recurrentes (RNN):** son especialmente útiles para el procesamiento de datos secuenciales, como series temporales, texto, y secuencias de audio. A diferencia de las redes *feedforward*, las RNN tienen conexiones hacia atrás, lo que les permite mantener un "estado" o memoria de entradas anteriores, haciendo posible capturar dependencias temporales. Sin embargo, las RNN tradicionales pueden enfrentar problemas como la desaparición de gradientes, lo que limita su capacidad para aprender dependencias a largo plazo.
- **Long Short-Term Memory (LSTM):** son una mejora sobre las RNN tradicionales y están diseñadas para manejar la memoria a largo plazo de manera más efectiva. Introducen un conjunto de celdas y puertas que regulan el flujo de información dentro de la red, permitiendo que se mantenga información relevante durante largos períodos. Esto las hace especialmente adecuadas para tareas como la traducción automática, el reconocimiento de voz, y la predicción de series temporales donde las dependencias a largo plazo son cruciales.
- **Red Neuronal Residual (ResNet):** son una innovación clave en el aprendizaje profundo que permite entrenar redes extremadamente profundas. ResNet introduce conexiones residuales que permiten que las capas salten algunas conexiones, mitigando el problema del "desvanecimiento de gradientes". Estas redes han sido fundamentales para mejorar la precisión en tareas de clasificación de imágenes y han sido una base para muchas arquitecturas avanzadas en la visión por computadora.
- **Autoencoders:** son redes neuronales utilizadas para aprender representaciones comprimidas de los datos. Están compuestos por dos partes: un codificador que reduce la dimensionalidad de los datos de entrada, y un decodificador que intenta reconstruir los datos originales a partir de esta representación comprimida. Son comúnmente utilizados para la reducción de dimensionalidad, eliminación de ruido y en el preentrenamiento de redes más complejas.
- **Generative Adversarial Networks (GANs):** son un tipo revolucionario de red que consta de dos redes que compiten entre sí: una generadora y una discriminadora.

La red generadora crea datos sintéticos, mientras que la discriminadora trata de diferenciar entre datos reales y sintéticos. Las GANs han sido utilizadas para la generación de imágenes realistas, síntesis de voz, y en la creación de contenido artístico y de entretenimiento.

2.4.6. Tipos del aprendizaje en redes neuronales artificiales

Dentro de este campo, existen diversas metodologías que permiten a las máquinas aprender de diferentes formas, adaptándose a las necesidades y características específicas de los problemas que se desean resolver. Las principales categorías de aprendizaje automático son el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo.

El aprendizaje supervisado es uno de los enfoques más utilizados en el *machine learning*, especialmente en el ámbito empresarial (Según Gartner, en 2022, sigue siendo la metodología preferida entre los líderes tecnológicos). Este tipo de aprendizaje utiliza datos etiquetados, donde el algoritmo aprende de entradas y salidas históricas para predecir resultados futuros. Entre los algoritmos más utilizados en el aprendizaje supervisado se encuentran las redes neuronales (neural networks), los árboles de decisión (*decision trees*), la regresión lineal (linear regression) y las máquinas de vectores de soporte (*support vector machines* o SVM).

A diferencia del aprendizaje supervisado, el aprendizaje no supervisado no depende de datos etiquetados. En su lugar, el algoritmo busca patrones ocultos en los datos sin intervención humana. Este enfoque es útil para identificar relaciones y estructuras en datos que no están claramente definidas. Entre los algoritmos más utilizados en el aprendizaje no supervisado se encuentran los modelos de Markov ocultos (*Hidden Markov models*), k-means, la agrupación jerárquica (*hierarchical clustering*) y los modelos de mezcla gaussiana (*Gaussian mixture models*). Las aplicaciones incluyen la segmentación de clientes, la agrupación de inventarios y la identificación de asociaciones en grandes volúmenes de datos.

El aprendizaje por refuerzo es el más similar al aprendizaje humano, donde un algoritmo interactúa con su entorno y aprende a través de recompensas positivas o negativas. Este tipo de aprendizaje es particularmente potente en entornos simulados o con grandes volúmenes de datos, aunque requiere una capacidad computacional significativa. Ejemplos de aplicaciones incluyen el entrenamiento de vehículos autónomos, el control de semáforos y la robótica avanzada.

2.4.7. Proceso de Entrenamiento de una Red Neuronal Artificial

El **proceso de entrenamiento de una red neuronal** es un procedimiento iterativo que ajusta los pesos de la red para minimizar el error en las predicciones realizadas a partir de los

datos de entrada. Este proceso es fundamental para que la red neuronal aprenda a realizar tareas específicas, como clasificación o predicción.

1. **Inicialización del Modelo:** Los pesos de la red neuronal se inicializan de forma aleatoria o mediante técnicas como la inicialización Xavier o He, dependiendo de la arquitectura de la red. Esta etapa es crucial porque la elección de una mala inicialización puede afectar significativamente la eficiencia del entrenamiento.
2. **Propagación hacia Adelante (*Forward Propagation*):** En esta fase, los datos de entrada se pasan a través de la red, capa por capa, realizando operaciones matemáticas (multiplicaciones de matrices y aplicaciones de funciones de activación) para producir una salida. La salida de la red se compara con la salida deseada utilizando una función de pérdida (por ejemplo, entropía cruzada para clasificación o error cuadrático medio para regresión).
3. **Cálculo de la Pérdida:** La función de pérdida cuantifica qué tan lejos están las predicciones de la red de los valores reales. Esta métrica es la que se utiliza para ajustar los pesos de la red durante el entrenamiento.
4. **Retropropagación (*Backpropagation*):** La retropropagación es el núcleo del proceso de entrenamiento. Aquí, se calculan los gradientes de la función de pérdida con respecto a los pesos de la red usando el algoritmo de retropropagación. Estos gradientes se utilizan para actualizar los pesos en dirección opuesta al gradiente mediante un optimizador como SGD (*Stochastic Gradient Descent*), Adam, o RMSprop. El objetivo es minimizar la función de pérdida.
5. **Actualización de Pesos:** Los pesos de la red se actualizan utilizando los gradientes calculados. Técnicas como el *momentum* pueden añadirse para acelerar el proceso de convergencia y evitar quedarse atrapado en mínimos locales. En el caso de Adam, se combinan las ventajas de *momentum* y RMSprop para ajustar adaptativamente la tasa de aprendizaje durante el entrenamiento.
6. **Validación y Monitoreo:** A lo largo del entrenamiento, se suele utilizar un conjunto de validación para monitorizar el rendimiento de la red y evitar el sobreajuste. El proceso de validación permite ajustar hiperparámetros y detener el entrenamiento de forma temprana si el rendimiento en el conjunto de validación deja de mejorar (técnica conocida como "*early stopping*").
7. **Iteración:** Este ciclo de propagación hacia adelante, retropropagación y actualización de pesos se repite durante múltiples épocas. Cada época implica pasar todo el conjunto de entrenamiento a través de la red y ajustar los pesos en consecuencia. Este proceso se repite hasta que la red converge a un mínimo aceptable de la función de pérdida o hasta que se alcanza un número máximo de épocas.

2.4.8. Funciones de activación

La función de activación en una red neuronal es crucial para determinar cómo se procesan y transforman las señales de entrada a medida que pasan de una capa a otra en la red. Su función principal es introducir no linealidad en el modelo, lo que permite a la red neuronal aprender y representar relaciones complejas entre las entradas y las salidas. Sin la función de activación, una red neuronal simplemente actuaría como un modelo lineal, lo que limitaría severamente su capacidad para abordar problemas complejos. Las funciones de activación más comunes son las siguientes:

- **Función identidad:** Esta función devuelve el mismo valor que se le proporciona como entrada. No introduce ninguna no linealidad, por lo que es utilizada raramente en capas ocultas, pero puede ser útil en la capa de salida de modelos de regresión, donde se espera una salida continua:

$$f(x) = x \tag{1}$$

- **Función ReLU (*Rectified Linear Unit*):** La función ReLU activa la neurona solo si la entrada es positiva; de lo contrario, la salida es cero. Es muy popular en redes neuronales profundas debido a su simplicidad y eficiencia, ayudando a mitigar problemas de gradientes desvanecientes:

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \tag{2}$$

- **Función Tangente Hiperbólica:** La tangente hiperbólica es similar a la función sigmoide, pero sus salidas varían entre -1 y 1, lo que puede ser más útil en redes donde se necesita centrarse en valores tanto positivos como negativos:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

- **Función de Sigmoide:** La función sigmoide convierte cualquier valor de entrada en un valor entre 0 y 1, lo que es útil en problemas de clasificación binaria. Sin embargo, puede sufrir de saturación en valores extremos, lo que ralentiza el aprendizaje:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

- **Función base radial:** Esta función es utilizada en redes neuronales de función de base radial (RBF). Es particularmente útil en problemas donde las entradas están basadas en una distancia o similitud con respecto a un punto de referencia:

$$f(x) = \exp\left(-\frac{x^2}{2}\right) \tag{5}$$

2.4.9. Algoritmos de optimización

En el campo del aprendizaje automático, los algoritmos de optimización son esenciales para entrenar redes neuronales, ajustando los parámetros del modelo con el fin de minimizar la función de coste y mejorar la precisión de las predicciones.

La **función de coste** es una medida que cuantifica el error entre las predicciones del modelo y los valores reales del conjunto de datos. Minimizar esta función es crucial porque refleja directamente la capacidad del modelo para hacer predicciones precisas; un valor menor de la función de coste indica un modelo más preciso. El proceso de optimización busca ajustar los parámetros de la red de manera que la función de coste sea lo más baja posible, lo que lleva a un mejor rendimiento del modelo y a una mayor capacidad de generalización a datos nuevos y no vistos.

Entre los algoritmos de optimización, el descenso del gradiente y sus variantes son los más ampliamente utilizados para lograr este objetivo.

El **descenso del gradiente** es un método iterativo que busca encontrar el mínimo de una función de coste calculando el gradiente con respecto a los parámetros del modelo y ajustándolos en la dirección opuesta al gradiente. Este enfoque tiene varias variantes: el **descenso del gradiente en lote** utiliza todo el conjunto de datos para cada actualización, lo que puede ser lento cuando se trabaja con grandes volúmenes de datos; el **descenso del gradiente estocástico (SGD)**, en cambio, actualiza los parámetros utilizando un solo ejemplo de entrenamiento por iteración, acelerando el proceso aunque introduciendo más inestabilidad; y el **descenso del gradiente mini-batch**, que combina las ventajas de los dos anteriores al actualizar los parámetros utilizando pequeños subconjuntos de datos, logrando un equilibrio entre velocidad y estabilidad.

Para mejorar la eficacia del descenso del gradiente, se han desarrollado varias variantes. El método **Momentum** introduce un término adicional que acumula el gradiente de pasos anteriores, suavizando el proceso de optimización y ayudando al modelo a evitar quedarse atrapado en mínimos locales poco profundos. **Adagrad (Adaptive Gradient)**, por su parte, adapta la tasa de aprendizaje de forma individual para cada parámetro, lo que es especialmente útil para características escasas y de alta dimensionalidad, ajustando automáticamente la magnitud de los pasos de optimización en función del historial de gradientes.

Otra variante, **RMSprop (Root Mean Square Propagation)**, mantiene una tasa de aprendizaje estable para cada parámetro, controlando la acumulación excesiva de gradientes mediante un factor de decay, lo que permite una optimización más constante y eficiente. Finalmente, **Adam (Adaptive Moment Estimation)** combina las ventajas de *Momentum* y *RMSprop*, acumulando un promedio exponencial de los gradientes y sus cuadrados, y ajustando

dinámicamente las tasas de aprendizaje para cada parámetro. Adam se ha convertido en uno de los optimizadores más utilizados debido a su eficacia y facilidad de ajuste.

2.4.10. Regularización en redes neuronales artificiales

La regularización en redes neuronales es un conjunto de técnicas utilizadas para mejorar el rendimiento de los modelos y prevenir el sobreajuste (*overfitting*). Estas técnicas buscan equilibrar la complejidad del modelo y la capacidad de generalización a nuevos datos, lo que es crucial para asegurar que el modelo no solo aprenda bien los datos de entrenamiento, sino que también funcione correctamente con datos no vistos.

El *overfitting* ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento, capturando no solo las relaciones reales, sino también el ruido y las peculiaridades específicas de esos datos. Como resultado, aunque el modelo puede mostrar un rendimiento excelente en los datos de entrenamiento, su rendimiento en los datos de validación o en nuevos datos será pobre, porque no ha aprendido a generalizar correctamente. Por otro lado, el *underfitting*, aparece cuando un modelo es demasiado simple para capturar las relaciones subyacentes en los datos. Esto puede deberse a un modelo con una arquitectura demasiado básica o insuficiente para la tarea en cuestión, o a una falta de entrenamiento adecuada. En este caso, tanto el rendimiento en el conjunto de entrenamiento como en el conjunto de validación será bajo. La diferencia de su curva de ajuste, a nivel gráfico, se puede apreciar en la siguiente figura (Figura 11):

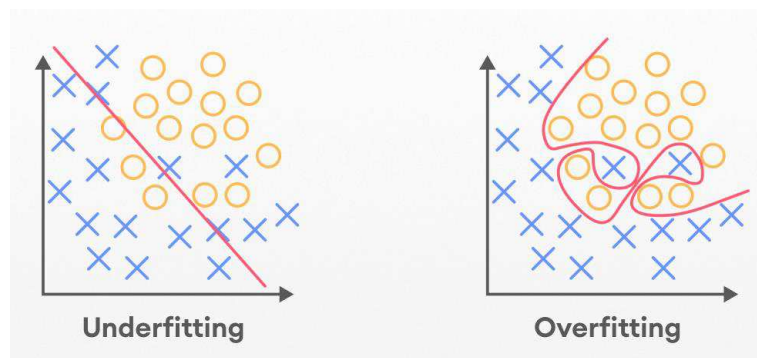


Figura 11. Underfitting vs. Overfitting en modelos de aprendizaje automático.

2.4.10.1. Técnicas de Regularización

Para mitigar el sobreajuste, se utilizan varias técnicas de regularización en redes neuronales:

1. Regularización L1 y L2:

- **L1 Regularization** (también conocida como regularización Lasso): Añade una penalización equivalente a la suma de los valores absolutos de los pesos

a la función de pérdida. Esto induce la esparsión en los pesos, lo que significa que muchos de ellos se reducen a cero, eliminando efectivamente algunas conexiones neuronales y simplificando el modelo.

- **L2 Regularization** (también conocida como regularización 'Ridge'): Añade una penalización equivalente a la suma de los cuadrados de los pesos a la función de pérdida. Esto no solo ayuda a evitar que los pesos se vuelvan excesivamente grandes, sino que también estabiliza el proceso de entrenamiento, ya que mantiene todos los pesos pequeños pero diferentes de cero.
 - **L1 y L2 combinadas** (*Regularización Elastic Net*): Combina las penalizaciones L1 y L2, beneficiándose de la esparsidad de L1 y la estabilidad de L2.
2. **Dropout**: es una técnica en la que, durante el entrenamiento, se desactivan aleatoriamente un porcentaje de neuronas en cada capa en cada paso de entrenamiento. Esto fuerza a la red a no depender de ninguna neurona en particular, promoviendo una distribución más generalizada del aprendizaje y reduciendo la co-adaptación de las neuronas. *Dropout* es particularmente efectivo en redes profundas donde el riesgo de sobreajuste es mayor.
 3. **Early Stopping**: Aunque no es una técnica de regularización en el sentido estricto, *Early Stopping* es muy utilizada para evitar el sobreajuste. Consiste en monitorizar el rendimiento del modelo en un conjunto de validación durante el entrenamiento y detener el entrenamiento cuando el rendimiento deja de mejorar. Esto previene que el modelo se entrene en exceso, capturando el ruido en los datos de entrenamiento.
 4. **Data Augmentation**: no es una técnica de regularización per se, pero está estrechamente relacionada. Involucra generar nuevas muestras de datos artificiales basadas en las existentes mediante transformaciones como rotaciones, escalado o adición de ruido. Esto aumenta el tamaño del conjunto de datos de entrenamiento y ayuda a que el modelo generalice mejor al enfrentarse a variaciones no vistas en los datos de prueba.
 5. **Batch Normalization**: normaliza las salidas de una capa antes de pasarlas a la siguiente capa, estabilizando la distribución de activaciones y permitiendo entrenar redes más profundas de manera más eficiente. Esto también tiene un efecto regularizador, ya que introduce un pequeño ruido adicional al proceso de entrenamiento.

2.5. Aplicaciones de redes neuronales artificiales en el mecanizado

La integración de redes neuronales en el mecanizado en seco está revolucionando la fabricación, combinando la inteligencia artificial con tecnologías como el Internet de las Cosas y el análisis de big data para optimizar los procesos. Este enfoque no solo mejora la eficiencia y precisión en la producción, sino que también abre nuevas posibilidades para abordar desafíos en la industria 4.0, marcando un paso significativo hacia fábricas más inteligentes y sostenibles.

Estos desafíos aumentan los estudios académicos en diferentes áreas de fabricación utilizando así técnicas innovadoras en los diferentes procesos de fabricación.

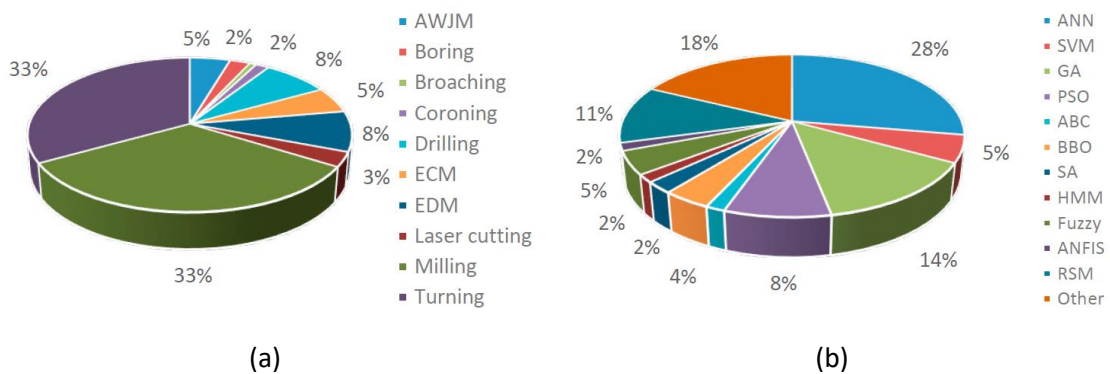


Figura 12. (a) Procesos de corte respaldados para aplicaciones de aprendizaje automático. (b) Diferentes tipos de aplicaciones de aprendizaje automático aplicados en procesos de corte [25].

En la Figura 12.a muestra varios procesos de corte como el Chorro de Agua Abrasivo (AWJM), Taladrado (Boring), Brochado (Broaching), Coronado (Coroning), Perforación (Drilling), Mecanizado Electroquímico (ECM), Mecanizado por Descarga Eléctrica (EDM), Corte por Láser (Laser cutting), Fresado (Milling) y Torneado (Turning).

En la Figura 12.b se encuentran técnicas de aprendizaje automático tales como Redes Neuronales Artificiales (ANN), Máquinas de Vectores de Soporte (SVM), Algoritmos Genéticos (GA), Optimización por Enjambre de Partículas (PSO), Algoritmo de Colonia de Abejas (ABC), Optimización de Biogeografía Basada (BBO), Simulación de Enfriamiento (SA), Método de la Harmonía (HM), Lógica Difusa (Fuzzy), Sistema de Inferencia Neuro-Difuso Adaptativo (ANFIS) y Método de Superficie de Respuesta (RSM).

D. H. Kim *et al* [26], describen MindSphere de Siemens, un avanzado sistema operativo IoT basado en la nube, facilita el monitoreo en tiempo real y el mantenimiento predictivo de maquinaria industrial a través de la integración de datos procedentes de múltiples sensores. Este sistema se potenció en 2016 con la incorporación de Watson Analytics de IBM, lo cual enriqueció sus capacidades con herramientas de análisis predictivo y aprendizaje profundo. La

mejora significativa incluye la optimización de operaciones y una reducción en tiempos de inactividad, aunque sin detalles específicos cuantitativos. Un ejemplo concreto de la eficacia de este sistema se observa en las turbinas de gas, donde más de 500 sensores integrados recopilan datos en tiempo real sobre presión, temperatura, tensión y otras variables críticas. La aplicación de inteligencia artificial en este contexto ha logrado una reducción adicional de las emisiones de óxido nitroso en un 10-15%, demostrando la potencia y precisión del análisis predictivo en operaciones industriales complejas.

T. Misaka *et al* [27] proponen una metodología que integra la simulación de elementos finitos (FEM) con redes neuronales profundas (DNN) para monitorear el proceso de torneado en máquinas. Se utilizan simulaciones FEM para predecir variables críticas como el estrés residual, que no se pueden medir directamente durante el mecanizado. Estos datos de simulación se emplean para entrenar una DNN, que luego interpreta imágenes en tiempo real obtenidas durante el torneado para estimar las condiciones de corte y las variables asociadas.

Uno de los principales beneficios de la aplicación de la IA es la capacidad mejorada para monitorear las condiciones del proceso en tiempo real, incluyendo la detección de variables difíciles de medir como el estrés efectivo y las tasas de alimentación. Además, la tecnología puede identificar anomalías durante el proceso de corte, permitiendo intervenciones tempranas para prevenir daños en la maquinaria o defectos en el producto final. A continuación (Figura 13) se detalla el proceso.

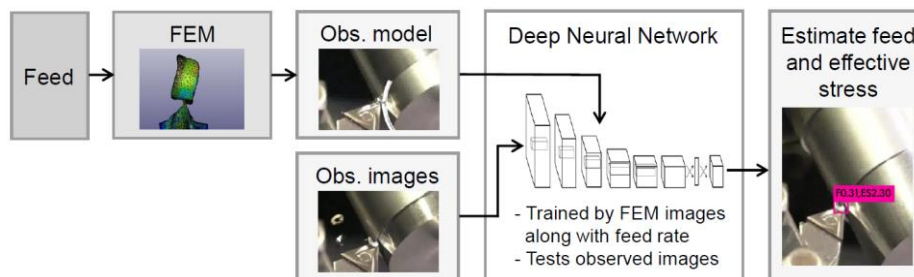


Figura 13. Detección de imágenes experimentales mediante la red neuronal profunda (DNN) entrenada con FEM para el monitoreo basado en imágenes [27].

El estudio reveló que la precisión de las redes neuronales profundas (DNN) en la detección de virutas mediante simulación por elementos finitos (FEM) varía entre el 70% y el 80%, dependiendo de la configuración del experimento. Específicamente, la precisión es mayor cuando las virutas son presentadas contra un fondo oscuro uniforme en comparación con aquellas cerca de la punta de la herramienta. Esto sugiere que el entorno visual en el que se entrenan y aplican estas redes puede influir significativamente en su eficacia.

No obstante, se enfrentaron desafíos, como detecciones incorrectas que alteraban los valores promedio de alimentación detectados, lo que subraya la necesidad de mejorar la precisión del modelo. Los histogramas de las puntuaciones de confianza de la DNN, tras procesar

imágenes experimentales, mostraron variabilidad en la precisión de las estimaciones. Este hecho refleja la necesidad de seguir refinando las técnicas de entrenamiento y selección de imágenes para mejorar la precisión general.

El documento [28] detalla cómo se realizaron las pruebas experimentales utilizando un diseño factorial completo (Figura 14), tanto en condiciones secas como con refrigerante de alta presión (HPC). Las temperaturas fueron medidas usando un termopar herramienta-pieza. Los modelos RSM desarrollados se basan en ecuaciones cuadráticas derivadas de los datos experimentales, con validaciones realizadas mediante ANOVA y el error absoluto medio porcentual (MAPE).

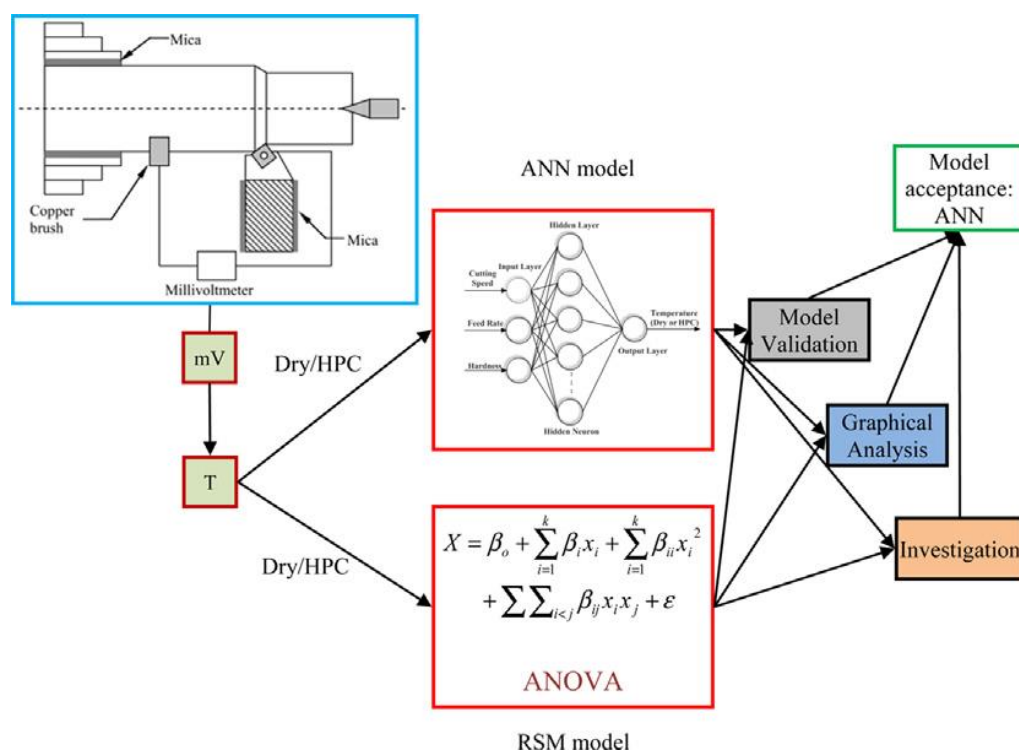


Figura 14. Diagrama de operaciones utilizadas para modelar la temperatura de corte en procesos de torneado duro.

Los modelos RNA se entrenaron con el 80% de los datos y se validaron con el 20% restante, mostrando un error de aproximadamente el 5% para los datos de prueba. Ambos modelos mostraron una alta precisión, pero el modelo RNA demostró ser más preciso.

Los resultados indican que ambos modelos son adecuados para controlar la temperatura de corte en el torneado de aceros endurecidos, con el modelo RNA ofreciendo una precisión ligeramente superior.

U. Maheshwera Reddy Paturi *et al* [29] examinan la modelación de la temperatura de corte en torneado duro bajo condiciones secas y con refrigerante de alta presión utilizando dos

métodos: la Metodología de Superficie de Respuesta (RSM) y Redes Neuronales Artificiales. En la modelación de la temperatura de corte en seco, el modelo RSM mostró un Error Porcentual Absoluto Medio de 0.78%, mientras que el modelo RNA registró un MAPE de 0.86%. A pesar de que el modelo RSM mostró un MAPE menor, este fue desarrollado y validado usando el mismo conjunto de datos, lo cual podría haber influido en los bajos valores de MAPE. En contraste, el modelo RNA, que fue entrenado y probado con diferentes conjuntos de datos, demostró una precisión razonable y una mejor capacidad de generalización, con un MAPE menor al 1%.

Entre las estructuras de red evaluadas, la configuración 3–15–1 del modelo RNA resultó ser la más precisa para predecir la temperatura de corte. Esto sugiere que el modelo RNA, aunque con un MAPE ligeramente superior, puede ser más efectivo en aplicaciones prácticas debido a su robustez y adaptabilidad en condiciones variables.

se centra en explorar las ventajas del uso de fluidos de corte en cantidad mínima (MQCF) comparado con métodos tradicionales como el enfriamiento con aire seco (DAC) y el enfriamiento por inundación (FC). El enfoque principal es evaluar cómo diferentes ambientes de mecanizado afectan la rugosidad superficial de las piezas de trabajo y cómo el uso de RNA puede modelar y predecir estas variables de forma efectiva.

En el estudio [30] se emplea una red neuronal artificial para modelar y predecir las condiciones del proceso de mecanizado utilizando la técnica de fluido de corte en cantidad mínima (MQCF). La RNA utilizada es del tipo feed-forward, caracterizada por una estructura de tres capas, que incluye una capa oculta. Aunque el número exacto de neuronas en la capa oculta no se detalla en los fragmentos proporcionados, es típico que la configuración y el tamaño de la red se ajusten según la complejidad del problema y los datos disponibles.

La red se entrena utilizando el algoritmo de retropropagación, que es común en aplicaciones de modelado por su eficacia en la minimización de errores durante el aprendizaje. Además, se menciona que la red emplea funciones de transferencia de tipo sigmooidal o tangente hiperbólica, que son estándar para este tipo de redes neuronales debido a su capacidad de manejar la no linealidad entre las variables de entrada y salida.

El entrenamiento de la RNA se realiza con un conjunto de datos que incluye diversas velocidades de corte y tasas de alimentación, ajustándose para minimizar el error cuadrático medio. Esto permite que la red genere predicciones precisas sobre el rendimiento del mecanizado, específicamente sobre la rugosidad superficial y el desgaste de herramientas, lo cual se verifica a través de un error absoluto porcentual medio que oscila entre el 1% y el 4%. Estos valores indican una alta precisión del modelo, demostrando la efectividad de las RNA para predecir y optimizar operaciones de mecanizado en condiciones reales.

Los estudios revisados han demostrado que la integración de técnicas de modelado basadas en redes neuronales artificiales (RNA) en la industria del mecanizado no solo optimiza

los procesos al reducir la necesidad de pruebas experimentales intensivas y costosas, sino que también mejora significativamente la sostenibilidad de estas operaciones. Esto se logra mediante la minimización del uso de fluidos de corte y la mejora de la calidad del producto final. Este enfoque alinea la fabricación moderna con las prácticas de producción sostenible, destacando el papel crucial de la ingeniería y la ciencia de datos en la evolución hacia fábricas más eficientes y respetuosas con el medio ambiente.

A pesar de la utilidad de las redes neuronales, se ha encontrado una notable escasez de estudios que las apliquen al análisis de la morfología de la viruta. Dada esta laguna en la investigación, se va a explorar esta aplicación, lo que representa una oportunidad significativa para futuras investigaciones. Estos estudios podrían explorar nuevas aplicaciones y optimizaciones de estos modelos predictivos en la morfología de la viruta.

3. Metodología

3.1. Introducción

A continuación, se describe la metodología experimental empleada en este Trabajo Fin de Grado. La metodología se organiza en diversas fases clave: desde la selección de los materiales y la recopilación de datos experimentales hasta el desarrollo detallado del modelo de red neuronal. Se describirá cómo se prepararon las muestras, cómo se configuró la red neuronal y cómo se realizaron las pruebas para validar el modelo. Además, se explicará el uso de Simulink para crear una aplicación práctica basada en los hallazgos. Todas estas fases se ilustran en el siguiente diagrama correspondiente (Figura 15).



Figura 15. Flujo de trabajo en la metodología.

3.2. Materiales

En el presente estudio, se desarrolló una metodología experimental detallada para investigar la caracterización morfológica de la viruta generada durante el mecanizado en seco de las aleaciones de UNS A97075 (Al-Zn) y UNS A92024 (Al-Cu) de aluminio, y UNS R56400 (Ti6Al4V) de titanio. Estas aleaciones son ampliamente utilizadas en la fabricación de componentes estructurales aeronáuticos debido a su excelente combinación de alta resistencia, ligereza y resistencia a la corrosión y fatiga. La Tabla 2 muestra la composición de cada material. Esta composición se obtuvo mediante espectroscopia de emisión atómica de arco (AES).

Tabla 2. Composición de la aleación mecanizada (% en peso).

Alloy	Zn	Mg	Cu	Cr	Si	Mn	Al
UNS A92024 (Al-Cu)	0,12	1,25	4,85	0,01	0,15	0,54	Resto
UNS A97075 (Al-Zn)	6,03	2,62	1,87	0,19	0,09	0,07	Resto
Alloy	C	Fe	N	O	Al	V	Ti
UNS R56400 (Ti6Al4V)	0,08	0,164	0,05	0,05	5,47	4,09	Resto

3.3. Método de obtención de los datos experimentales

3.3.1. Introducción

En el desarrollo de este trabajo, una parte crucial involucra la extracción de datos, la cual fue llevada a cabo por un equipo de investigadores del departamento de Ingeniería Civil, de Materiales y Fabricación de la Universidad de Málaga. Este proceso es esencial para el posterior desarrollo de modelos de predicción mediante redes neuronales que constituye el núcleo de este trabajo de fin de grado.

3.3.2. Mecanizado

El proceso de mecanizado seleccionado ha sido el cilindrado horizontal en un centro de torneado modelo Pinacho SP285R (véase la figura 16.a). Este equipo es capaz de operar sin refrigerante, lo cual es crucial para evaluar los efectos ambientales del mecanizado en seco. Se utilizaron diferentes probetas cilíndricas; de 40 a 60 mm de diámetro (Dm) y 150 mm de longitud (Lm) para las aleaciones A92024 y A97075 (Figura 16.b). Para Ti6Al4V se utilizaron probetas de 105 mm de diámetro (Dm) y 170 mm de longitud. de acuerdo con las normas establecidas en ASTM B348, para asegurar la uniformidad en las condiciones experimentales. Como herramienta se utilizó una plaquita rómbica WC-Co sin recubrimiento para A92024 y A97075 y con recubrimiento de TiCN/Al₂O₃ para Ti6Al4V, con referencias ISO DCMT 11T308F2 y TCMT 16T308-F1 respectivamente. Estas herramientas están recomendadas por el fabricante para el mecanizado de este tipo de aleaciones. Para garantizar unas condiciones iniciales uniformes, se

utilizó una plaquita nueva para cada ensayo. Todos los ensayos se realizaron en condiciones secas. Se realizaron 44 pruebas de torneado recto utilizando la combinación de parámetros de corte (avance; velocidad de corte; profundidad de corte) que se muestra en la Tabla 3. Estos valores se encuentran dentro del rango recomendado para el mecanizado de estos tipos de aleación. Es importante señalar que las aleaciones A92024 y A97075 no suelen mecanizarse a velocidades de corte bajas, de 40 y 80 m/min. Sin embargo, estos valores son habituales cuando se combina con materiales como aleaciones de Titanio o Polímeros Reforzados con Fibra de Carbono (CFRP) [16][22].

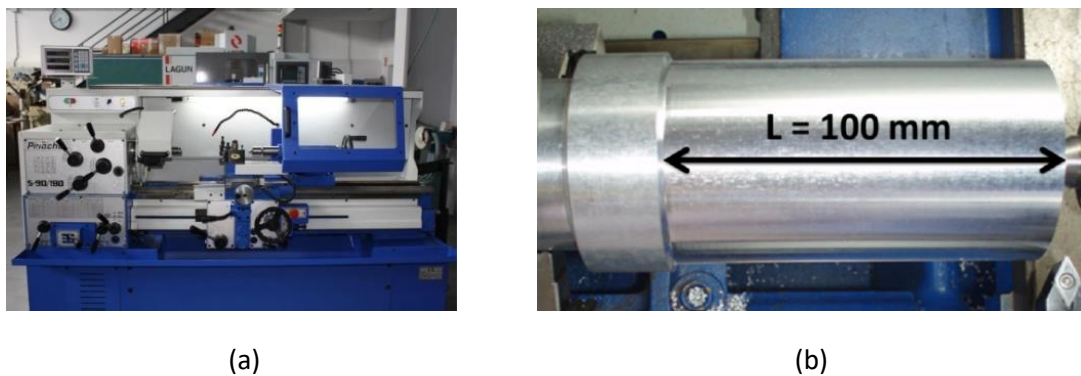


Figura 16. (a) Torno horizontal utilizado en las pruebas (Pinacho SP285R); (b) Barras de prueba cilíndricas [22].

Tabla 3. Parámetros de corte.

Material	Velocidad de corte (m/min)	Avance (mm/rev)	Profundidad de corte (mm)
A97075	40	0,05	1
	80	0,1	
A92024	150	0,2	
	200	0,3	
Ti6Al4V	30	0,05	1
	65	0,1	
	125	0,2	
		0,3	

3.3.3. Preparación de la viruta

El proceso de corte fue monitoreado utilizando una Cámara Digital SONY DCR-DVD92E, con el fin de grabar el proceso de generación de viruta y detectar cualquier problema de mecanizado. Además, la viruta generada fue recolectada, fotografiada y almacenada para

futuras observaciones. Con el objetivo de evaluar la evolución de la geometría de la viruta a lo largo de la longitud axial de mecanizado, la recolección de viruta se realizó en tres fases: al inicio, a la mitad y al final del mecanizado. Posteriormente, se prepararon muestras de viruta para su observación mediante Microscopía Óptica Estereoscópica (SOM). Para completar esto, se utilizaron técnicas metalográficas. Primero, las muestras fueron embebidas en resina (STRUERS CLAROFAST) utilizando el equipo CITOPRESS-1 STRUERS, Figura 17.a. Luego, fueron pulidas utilizando el equipo TEGRAPOL-11 STRUERS, Figura 17.b. Finalmente, se realizó un ataque químico en las muestras para facilitar la observación de los parámetros de la geometría de la viruta. Para ello, se utilizó un reactivo KELLER (95% H₂O, 2.5% HNO₃, 1.5% HCl, 1% HF).

Las muestras de viruta (Figura 17.c) fueron observadas utilizando un microscopio metalúrgico invertido EIPHOT 280 NIKON (Figura 17.d). Se adquirieron imágenes SOM de la sección longitudinal y transversal de la viruta mediante el uso de un Sistema Óptico CF Infinity (1.5× a 400×) y una Cámara Digital Eclipse. Las mediciones de los parámetros geométricos de la viruta se realizaron utilizando el software OmniMet BUEHLER [29].

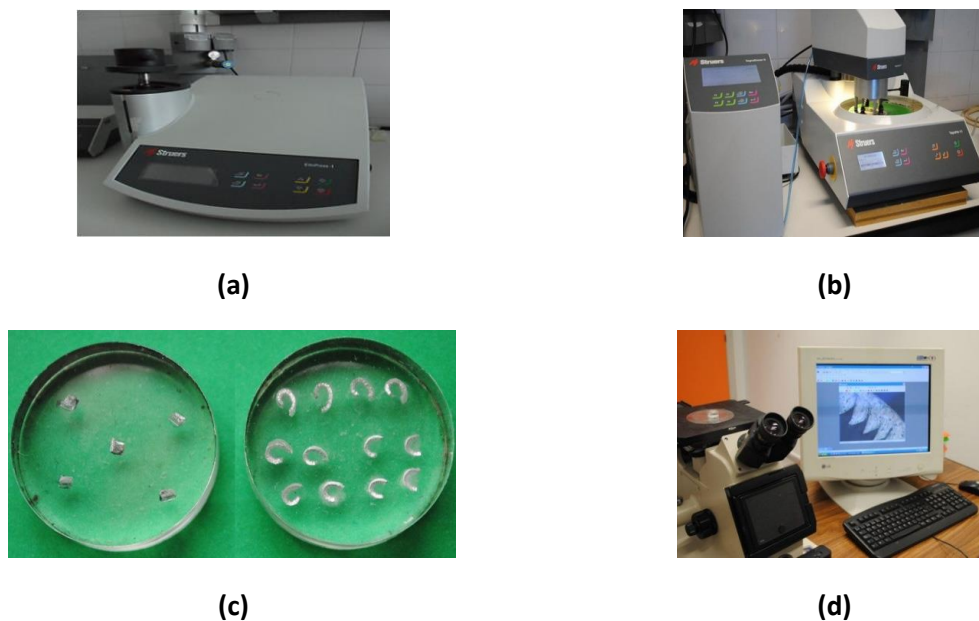


Figura 17. (a) Prensa de montaje en caliente; (b) Equipo de pulido; (c) Muestras de viruta; (d) Equipo de SOM (Microscopía Óptica Estereoscópica) [22].

3.3.4. Medición de la viruta

Por otro lado, el área de la sección transversal de la viruta puede cambiar según la posición a lo largo de la longitud de la viruta. Para asegurar la uniformidad en las comparaciones, todas las mediciones de S se realizaron a lo largo de la banda de corte. Para medir b , se trazó una línea aproximadamente paralela al lado más ancho (línea discontinua roja, Figura 18.b).

Luego, b se obtuvo trazando una línea perpendicular hasta el punto más alejado. Se analizaron un total de cinco muestras de la sección transversal, y cada parámetro geométrico se midió tres veces, resultando en un total de 15 mediciones para cada valor de avance [29].

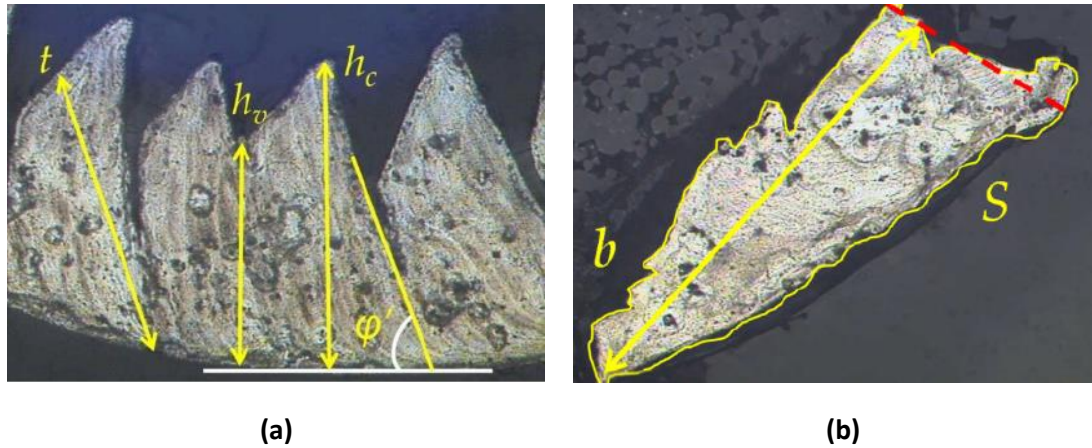


Figura 18. (a) Parámetros geométricos longitudinales de la viruta y **(b)** parámetros geométricos transversales de la viruta [22].

Esta metodología proporciona un marco exhaustivo y técnicamente avanzado para estudiar los efectos de los parámetros de mecanizado en la morfología de la viruta en una aleación crítica para aplicaciones aeronáuticas, con un enfoque en la eficiencia del proceso y la sostenibilidad ambiental.

Para analizar las propiedades geométricas de la viruta generada durante el mecanizado, se calculan varios parámetros fundamentales. Inicialmente se cuenta con los siguientes:

- Parámetros de corte:
 - Velocidad de corte (v_c) [m/min].
 - Avance (f) [mm/rev].
 - Profundidad de corte (a_p) [mm].
 - Ángulo de desprendimiento (γ) [°].
 - Ángulo de filo principal (K_r) [°].

- Propiedades geométricas de la viruta:
 - Altura de pico (h_p) [mm].
 - Altura de valle (h_v) [mm].
 - Sección de la viruta (S) [mm²].
 - Ancho de la viruta (b) [mm].
 - Ángulo de deslizamiento (ϕ) [°].

A continuación, se detallan los procedimientos para calcular el radio de segmentación (SR), el factor de recalcado (ζ) y el espesor de la viruta utilizando tres métodos diferentes (t_1, t_2, t_3).

$$SR = \frac{h_p - h_v}{h_p} \quad (6)$$

$$\zeta = \frac{\sin \varphi}{\cos(\varphi - Y)} \quad (7)$$

$$t_1 = \frac{S}{b} \quad (8)$$

$$t_2 = \frac{(f \sin K_r)}{\zeta} \quad (9)$$

$$t_3 = \frac{(h_p + h_v)}{2} \quad (10)$$

En este estudio se han evaluado numerosos parámetros de la viruta, pero se ha decidido enfocar en los parámetros experimentales, debido a que no pueden ser calculados. Entre los parámetros calculables, se ha identificado el espesor de la viruta como el más relevante, concretamente la ecuación t_3 elegida en base a lo leído en la literatura [22]. Por lo tanto, se dará prioridad a los aspectos experimentales y al espesor de la viruta, ya que proporcionan la información más crítica y precisa para los objetivos de análisis y optimización en el proceso de corte. Esta selección permite centrar los esfuerzos en los factores más significativos y obtener resultados más sólidos y aplicables.

3.4. Desarrollo del modelo de red neuronal artificial

3.3.1. Introducción

Para trabajar con los datos experimentales se ha hecho uso de Matlab, que es una herramienta ideal para la programación y análisis de redes neuronales artificiales en el contexto del mecanizado en seco de aleaciones ligeras debido a varias razones clave. Ofrece un entorno de desarrollo fácil de usar con herramientas integradas específicas para el diseño, entrenamiento y validación de RNA a través de la '*Neural Network Toolbox*', lo que simplifica la creación y optimización de modelos. Además, Matlab sobresale en la visualización de datos, crucial para analizar la morfología de la viruta y evaluar el rendimiento de los modelos RNA. Su integración con Simulink permite importar redes neuronales entrenadas directamente en un entorno de simulación, facilitando la evaluación del impacto de diferentes parámetros de corte. Matlab también proporciona capacidades avanzadas de preprocesamiento y análisis de datos,

incluyendo técnicas de normalización y análisis ANOVA, esenciales para preparar datos experimentales y garantizar la precisión del modelo predictivo. Finalmente, cuenta con una extensa comunidad de usuarios y una amplia documentación, lo que facilita la resolución de problemas y la implementación de metodologías avanzadas. Estas características hacen de Matlab una herramienta poderosa y eficiente para realizar análisis complejos y obtener resultados precisos en la predicción de la morfología de la viruta en procesos de mecanizado en seco.

A la hora de crear la red neuronal artificial en Matlab se ha seguido el siguiente orden:

- Preparación de datos.
- Análisis ANOVA.
- Normalización de datos.
- Configuración de la red neuronal artificial.

3.3.2. Preparación de los datos

La limpieza y preparación de datos es esencial para eliminar ruido y errores que puedan distorsionar el aprendizaje de la red neuronal. Datos crudos suelen contener valores atípicos, duplicados, y errores que, si no se manejan adecuadamente, pueden llevar a la red a identificar patrones incorrectos, reduciendo así la precisión del modelo. Además, la presencia de valores faltantes puede complicar el proceso de entrenamiento, por lo que técnicas de imputación o eliminación de estos valores son fundamentales.

En la preparación de datos para este estudio, se ha procedido a ordenar los datos por tipo de material, distinguiendo entre 3 materiales diferentes. Dentro de cada material, los datos se han organizado según la velocidad de corte, en orden ascendente. Para dos de los materiales (A97075 y A92024) se han considerado 4 velocidades de corte (40, 80, 150 y 200 (m/min)), mientras que para el tercer material (Ti6Al4V) se han considerado 3 velocidades de corte (30, 65 y 125 (m/min)). Dentro de cada categoría de velocidad de corte, se han agrupado los datos según 4 avances diferentes (0.05, 0.1, 0.2 y 0.3 (mm/rev)), teniendo 5 muestras de cada uno. Este proceso de organización ha resultado en un total de 220 datos.

3.3.3. Proceso de análisis ANOVA

Para evaluar la influencia de diferentes variables en la morfología de la viruta en los procesos de mecanizado en seco se procederá a realizar un análisis ANOVA (Análisis de Varianza). Este análisis estadístico permite identificar si las diferencias entre los grupos de datos son estadísticamente significativas, facilitando la comprensión de cuáles factores contribuyen más al comportamiento observado en la viruta. Mediante el ANOVA, se compararán las medias de diferentes parámetros para discernir si las variaciones en los resultados pueden atribuirse a

las variaciones en las variables independientes o si se deben al azar. Esta etapa es fundamental para validar la robustez del modelo desarrollado y para asegurar que las decisiones basadas en el modelo se sustenten en fundamentos estadísticamente validados.

3.3.4. Normalización de datos

La normalización de datos es un paso crucial en el desarrollo de modelos de redes neuronales artificiales, ya que permite ajustar los valores de las variables para que estén dentro de un rango común, generalmente entre 0 y 1 o -1 y 1, dependiendo de la técnica utilizada. Esto es fundamental porque las redes neuronales son sensibles a las magnitudes de las entradas, y trabajar con valores normalizados puede mejorar la velocidad de convergencia durante el entrenamiento y prevenir que algunas variables dominen sobre otras debido a sus escalas mayores.

Una fórmula común para la normalización es la **normalización min-max**, que ajusta los valores de una característica 'x' dentro del rango [0,1] utilizando la siguiente fórmula:

$$x_{normalizada} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (11)$$

Donde:

- 'x' es el valor original de la variable.
- ' x_{min} ' es el valor mínimo de la variable en el conjunto de datos.
- ' x_{max} ' es el valor máximo de la variable en el conjunto de datos.

Esta técnica asegura que todos los valores de las variables entren al modelo en el mismo rango, lo que facilita el proceso de aprendizaje de la red neuronal utilizada.

3.3.5. Configuración de la red neuronal artificial

3.3.5.1. Capas de la red

Capa de entrada: La capa de entrada de la red neuronal artificial está compuesta por los parámetros de corte, específicamente los materiales, la velocidad de corte y el avance.

Capas ocultas: La capa oculta de la red neuronal artificial está compuesta con una sola capa. Esta elección se fundamenta en la necesidad de minimizar la complejidad del modelo para evitar el sobreajuste y facilitar la interpretación de los resultados.

Desde un punto de vista computacional, el incremento en el número de capas y neuronas en una red conlleva un aumento significativo en la complejidad de cálculo. Esto no solo afecta el tiempo de entrenamiento, sino que puede complicar innecesariamente la convergencia del modelo en tareas donde las relaciones subyacentes no son extremadamente complejas. En el caso de la morfología de la viruta, no se exhiben patrones que justifiquen un

modelo de alta complejidad. Los experimentos preliminares han mostrado que una red con una sola capa oculta y 10 neuronas logra una convergencia adecuada y eficaz, demostrando la capacidad de este modelo simplificado para capturar las dinámicas esenciales del proceso.

Las redes más profundas y complejas son propensas a requerir una mayor cantidad de datos para entrenamiento para generalizar efectivamente sin sobreajustarse. Dado que el conjunto de datos, aunque robusto, no es extremadamente grande, un modelo más sencillo resulta más adecuado para garantizar una buena generalización a partir de los datos disponibles.

Capa de salida: La capa de salida está diseñada para predecir las propiedades de la viruta, tales como la altura de pico, la altura de valle, el ancho, la sección y el espesor de la viruta. El número de neuronas en la capa de salida corresponde al número de propiedades de la viruta que se desean predecir, en este caso 5. A continuación se muestra un diagrama de la arquitectura de la red neuronal utilizada (Figura 19):

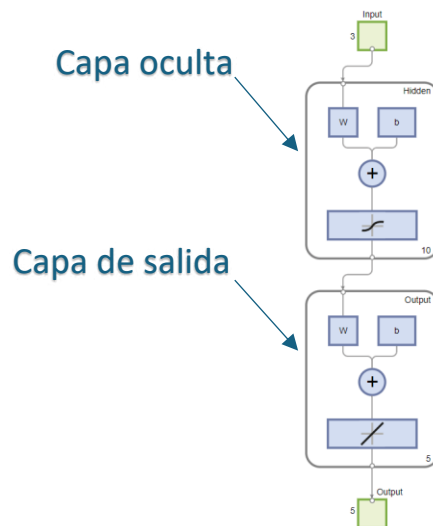


Figura 19. Arquitectura de la red neuronal artificial.

3.3.5.2. Neuronas

Para determinar el número óptimo de neuronas en la capa oculta, se implementó un método exhaustivo que comenzó con la creación de una red neuronal artificial con una sola neurona y aumentando el número de neuronas de una en una hasta llegar a 20. Para cada configuración, se calculó la Raíz del Error Cuadrático Medio (RMSE), utilizando esta métrica para evaluar el desempeño de la red debido a su capacidad de proporcionar una medida directa y comprensible de la magnitud de los errores del modelo.

Se graficaron los valores de RMSE de cada configuración para identificar visualmente el punto en el cual el incremento en el número de neuronas deja de ofrecer mejoras significativas en la precisión del modelo. Este análisis visual ayudó a identificar una red con 10 neuronas como la configuración óptima, ofreciendo un equilibrio adecuado entre la capacidad de modelado y la eficiencia computacional.

La elección de 10 neuronas se justificó además a través de gráficos de aprendizaje, los cuales mostraron que esta configuración minimizaba el riesgo de sobreajuste y subajuste, manteniendo una buena generalización del modelo en datos no vistos.

3.3.5.3. Funciones de activación

Función de activación para la capa oculta: Para la capa oculta, se utiliza la función de activación *Tansig* (Tangente Sigmoidea Hiperbólica) tal y como se observa en la figura 4 (flecha superior). La función *Tansig* varía entre -1 y 1 y se elige por su capacidad para modelar respuestas no lineales y su rango centrado en cero, lo que puede mejorar la eficiencia del aprendizaje en ciertos tipos de redes. La fórmula de la función *Tansig* es:

$$tansig(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (12)$$

Función de activación para la capa de salida: Para la capa de salida se utiliza *Purelin* (Lineal Pura), que es una función de activación lineal. Se puede ver en la figura 4, señalada con una flecha en la parte inferior del diagrama. Esta elección permite que la salida sea una representación directa y lineal de la entrada, lo cual es adecuado para problemas de regresión donde el objetivo es predecir un valor numérico continuo. La fórmula para *Purelin* es:

$$purelin(x) = x \quad (13)$$

La combinación de la función *Tansig* en las capas ocultas con *Purelin* en la capa de salida forma una estructura muy efectiva para modelos predictivos en contextos donde la salida final de la red necesita representar una gama amplia de valores reales. Esta configuración aprovecha las ventajas de ambas funciones: la no linealidad de *Tansig* para capturar complejidades en los datos y la linealidad de *Purelin* para ofrecer una predicción final precisa.

3.3.5.4. Elección del algoritmo de entrenamiento

Para el entrenamiento de la red neuronal, se ha seleccionado el algoritmo *Levenberg-Marquardt* (*trainlm*), conocido por su eficacia en la optimización de redes neuronales de tamaño pequeño a mediano. Este algoritmo combina las ventajas del método de descenso del gradiente y del método de Gauss-Newton, proporcionando una convergencia rápida y robusta. Esta combinación resulta especialmente beneficiosa en el estudio, ya que facilita una convergencia eficiente sin quedar atrapado en mínimos locales no deseados, un problema común en otros métodos de entrenamiento más convencionales.

El algoritmo *Levenberg-Marquardt* es particularmente valorado por su capacidad para minimizar el error cuadrático medio, lo cual es crucial para los objetivos de modelado. Esta característica permite una reducción significativa en las discrepancias entre las predicciones del modelo y los datos observados, asegurando un ajuste preciso a los datos experimentales. Comparado con otros algoritmos disponibles en MATLAB, *trainlm* muestra superioridad en términos de velocidad y precisión, lo que lo hace adecuado para manejar las complejidades específicas de los datos.

3.3.5.5. Configuración del algoritmo

Ajustes de Hiperparámetros: Para el ajuste de hiperparámetros como la tasa de aprendizaje, el *momentum*, el tamaño del *batch* y el número de épocas, se ha decidido utilizar los valores por defecto proporcionados por MATLAB. Esta decisión se basa en la robustez y la generalidad de estos valores predeterminados, que han demostrado ser eficaces en una amplia gama de escenarios, simplificando el proceso de configuración sin comprometer el rendimiento del modelo.

División de Datos: Antes de la creación de la red neuronal, los datos fueron divididos previamente en vectores, asignando el 80% del total de los datos al conjunto de entrenamiento y reservando el restante 20% para validación y pruebas. Para el entrenamiento de la red neuronal, se ha optado por utilizar el 100% del vector de entrenamiento. Esta estrategia maximiza la cantidad de datos disponibles para el aprendizaje, aprovechando al máximo el conjunto designado para el entrenamiento, mientras se mantiene la posibilidad de evaluar la generalización del modelo con el conjunto reservado para pruebas.

Inicialización de la Red: Los pesos de la red se inicializan aleatoriamente dentro de un rango específico que ha sido optimizado para promover una rápida convergencia. Esta estrategia de inicialización ayuda a evitar mínimos locales no óptimos al inicio del entrenamiento y facilita un proceso de aprendizaje más efectivo y eficiente.

3.4. Evaluación y validación del modelo

3.4.1. Evaluación de la RNA

Para evaluar la eficacia del modelo de red neuronal artificial en la predicción de la morfología de la viruta durante el mecanizado en seco de aleaciones ligeras, se seleccionaron varias métricas estadísticas clave. Estas métricas proporcionan una valoración cuantitativa de la precisión y la confiabilidad del modelo desarrollado.

Raíz del Error Cuadrático Medio (RMSE): El RMSE es una métrica ampliamente utilizada para medir la calidad de un modelo en tareas de regresión. Este calcula la raíz cuadrada del

promedio de los cuadrados de las diferencias entre los valores predichos por el modelo y los valores reales. Al igual que el MSE, el RMSE es efectivo para destacar errores grandes, pero a diferencia del MSE, el RMSE está en las mismas unidades que la variable de respuesta, lo que facilita su interpretación. La fórmula para calcular el RMSE es la siguiente:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (14)$$

Donde ' Y_i ' son los valores reales y ' \hat{Y}_i ' son los valores predichos por la red neuronal.

Coefficiente de Determinación (R^2): El coeficiente R^2 proporciona una medida de cuánto de la variabilidad en los datos de respuesta es explicada por el modelo. Un R^2 cercano a 1 sugiere que el modelo explica la mayoría de la variabilidad en los datos de respuesta, mientras que un valor cercano a 0 indica que el modelo no mejora las predicciones sobre la media. La fórmula para el R^2 es:

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (15)$$

Donde ' \bar{Y} ' es el promedio de los valores observados.

Coefficiente de Determinación Ajustado (R^2 Ajustado): Considerando que el R^2 puede estar sesgado a favor de modelos con más predictores, el R^2 ajustado es particularmente útil, ya que ajusta el estadístico R^2 para el número de predictores en el modelo, ofreciendo una medida más precisa del rendimiento predictivo, especialmente cuando se comparan modelos con diferentes números de predictores. La fórmula para el R^2 ajustado es:

$$R_{ajustado}^2 = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right) \quad (16)$$

Donde ' n ' es el número total de observaciones y ' p ' es el número de predictores independientes en el modelo.

Para complementar las métricas cuantitativas, se realizaron análisis visuales adicionales, incluyendo gráficos de dispersión que comparan las predicciones del modelo con los valores reales y gráficos de residuales para identificar posibles patrones de error.

Estos gráficos confirmaron que el modelo no presenta sesgos significativos y que los errores están distribuidos de manera aleatoria, lo cual es indicativo de un buen ajuste del modelo.

3.4.2. Comparación entre modelos tradicionales y RNA

Este estudio se enfoca en comparar el desempeño de una red neuronal artificial frente a modelos de regresión tradicionales ampliamente referenciados en la literatura, como son los modelos potenciales y polinómicos. La elección de estos modelos se fundamenta en su previa validación y uso extendido para la predicción de diversos fenómenos físicos en el campo del mecanizado, lo que proporciona una base comparativa robusta para evaluar las capacidades predictivas de las técnicas de aprendizaje automático más avanzadas.

Para un análisis eficiente en el estudio de mecanizado, se recolectaron cinco muestras por cada situación o conjunto de: material, velocidad de corte y avance. Se calculó la media de estas cinco muestras para obtener un único valor representativo por situación, evitando así la repetición innecesaria en los datos alimentados a la red neuronal.

Con un total de 44 datos experimentales listos para analizar (16 para las aleaciones A92024 y A97075, y 12 para el titanio Ti6Al4V), el siguiente paso es la obtención de los modelos potenciales y polinómicos en este estudio, se utilizó una herramienta de inteligencia artificial, específicamente la función 'Data Analyst' de Explora GPT, diseñada para especializarse en análisis de datos. A través de esta plataforma, se subió una hoja de Excel con datos limpios relativos a las condiciones de mecanizado. Se solicitó a la herramienta calcular los modelos potenciales y polinómicos basándose en estos datos.

La herramienta procesó la información y generó las ecuaciones correspondientes para cada modelo, incluyendo los parámetros específicos de cada ecuación. Además, proporcionó el coeficiente de determinación ajustado (R^2 ajustado) para cada uno de los modelos derivados, lo cual permitió evaluar la precisión del ajuste de cada modelo con respecto a los datos.

Para la comparación de estos tres modelos con los datos experimentales, se emplearon gráficos 3D. Esto facilitó la visualización y comparación de las predicciones de cada modelo contra los datos reales, a través de gráficas para cada variable dentro de cada material, proporcionando una comprensión más amplia y detallada del conjunto de datos.

3.5. Desarrollo de aplicación con Simulink

Finalmente, como parte del proceso de análisis y presentación de los resultados del estudio, se desarrolló una aplicación en Simulink. Esta aplicación permite a los usuarios introducir los datos de las condiciones de mecanizado y obtener automáticamente los resultados de los modelos analizados. La interfaz de Simulink facilita la visualización de cómo los diferentes modelos responden a los datos ingresados, permitiendo comparaciones directas y una evaluación inmediata de las predicciones de cada modelo.

La estructura de la aplicación se detalla a continuación, utilizando un formato sistemático y técnico adecuado para una descripción metodológica en un entorno académico o profesional:

1. **Bloques 'Constant':** A través de este bloque se introducen los datos del mecanizado: material, velocidad de corte (v_c) y el avance (f).
2. **Bloque de función (Normalización):** En la entrada de la aplicación, se encuentran los bloques encargados de normalizar las variables de entrada, en este caso, la velocidad de corte (v_c) y el avance (f).
3. **Red Neuronal - Function Fitting Neural Network:** El núcleo de la aplicación es la red neuronal diseñada específicamente para el ajuste de funciones. Este bloque de Simulink, etiquetado como 'NNet', recibe las entradas normalizadas y procesa los datos a través de la red para generar las salidas correspondientes a los parámetros del proceso de mecanizado.
4. **Bloque 'Demux':** Se utiliza para desmultiplexar, es decir, para dividir una línea de señal de entrada en múltiples líneas de salida. Este bloque es esencial cuando se desea canalizar diferentes componentes de una señal vectorial a distintos bloques o subsistemas dentro de un modelo de Simulink. Por ejemplo, en este caso se necesita extraer cada columna de la matriz de salida para desnormalizarla de manera individual.
5. **Bloques de función (Desnormalización):** Cada una de las salidas de la red neuronal, que inicialmente están en un formato normalizado, son procesadas a través de bloques de desnormalización. Estos bloques convierten los valores de salida normalizados (hp_N , hv_N , S_N , b_N , t_N) a sus escalas originales y prácticas (h_p , h_v , S , b , t), permitiendo que los resultados sean interpretados y utilizados eficazmente en aplicaciones reales.
6. **Interfaz Gráfica y Elementos de Control:** Para facilitar la interacción con el usuario, se implementan elementos de interfaz gráfica como cuadros de texto, botones y displays numéricos. Estos componentes permiten a los usuarios ingresar los parámetros de corte y visualizar los resultados desnormalizados. Los botones proporcionan funcionalidad interactiva, ejecutando cálculos y presentando los datos en los bloques de número correspondientes.

4. Resultados

4.1. Introducción

En esta sección, se presentan los resultados obtenidos mediante un análisis exhaustivo de los datos recopilados en el estudio. Se inicia con una descripción estadística visual de los datos experimentales.

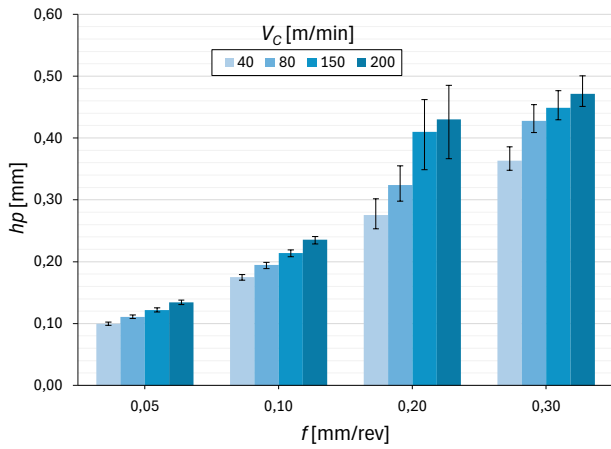
A continuación, se detalla el análisis de varianza (ANOVA) realizado para identificar diferencias significativas entre grupos de variables, lo que fundamenta la posterior creación de modelos avanzados. Antes de aplicar técnicas de aprendizaje automático, se efectuó una meticulosa normalización de las variables para garantizar uniformidad en los datos de entrada.

La sección continúa con el desarrollo de modelos potenciales, incluyendo modelos polinómicos y una red neuronal artificial. Además, se presentan comparaciones detalladas entre los modelos mediante gráficos tridimensionales que ilustran la capacidad predictiva de cada uno frente a los datos experimentales.

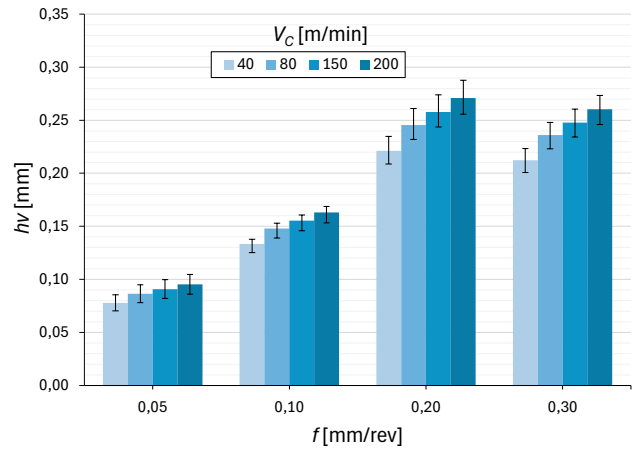
4.2. Distribución y análisis de los datos experimentales

Los datos de mecanizado se exponen a continuación. Las gráficas de barras muestran el material A97075 en azul, A92024 en verde y Ti6Al4V en morado.

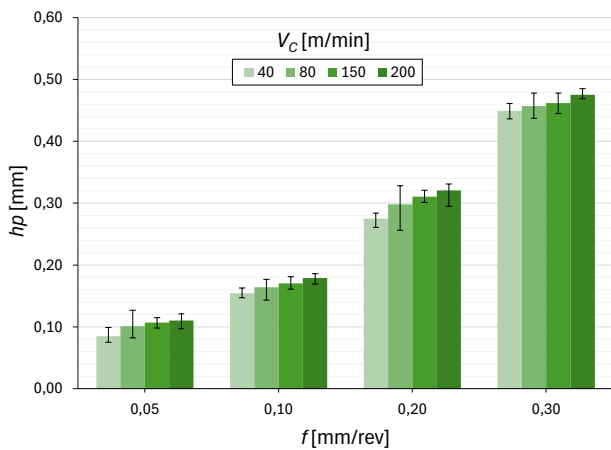
4.2.1. Altura de los picos (h_p) y de los valles (h_v)



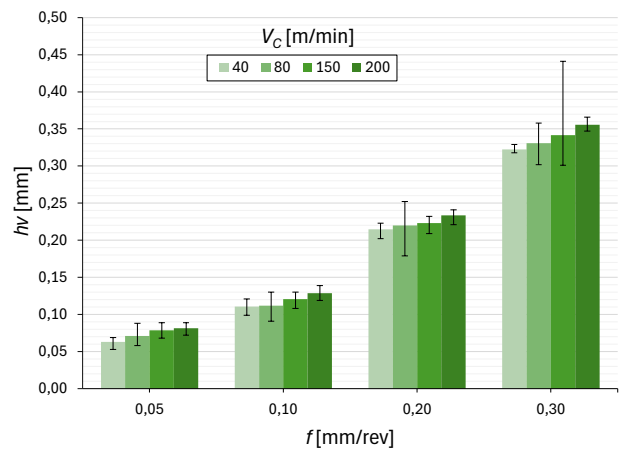
(a)



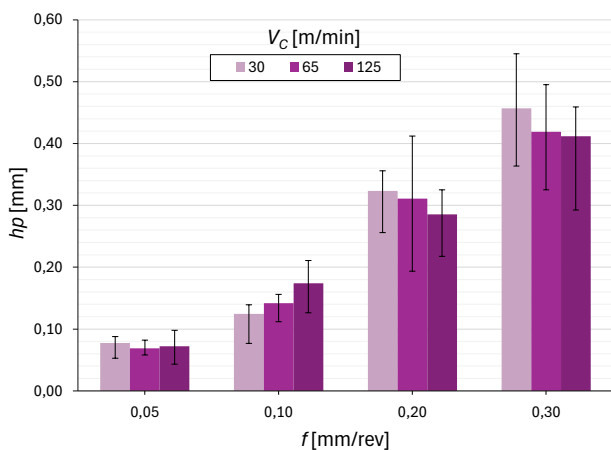
(b)



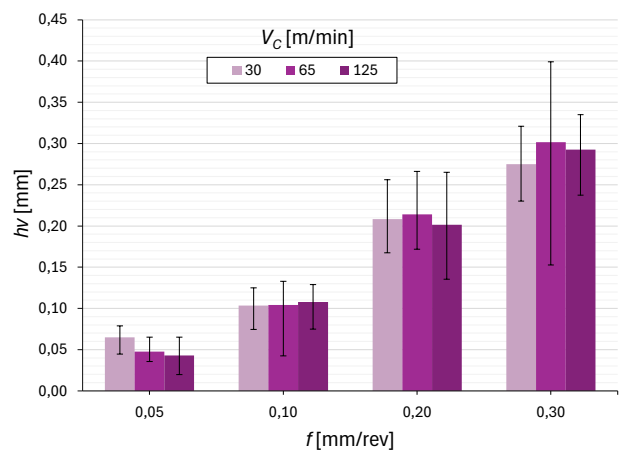
(c)



(d)



(e)



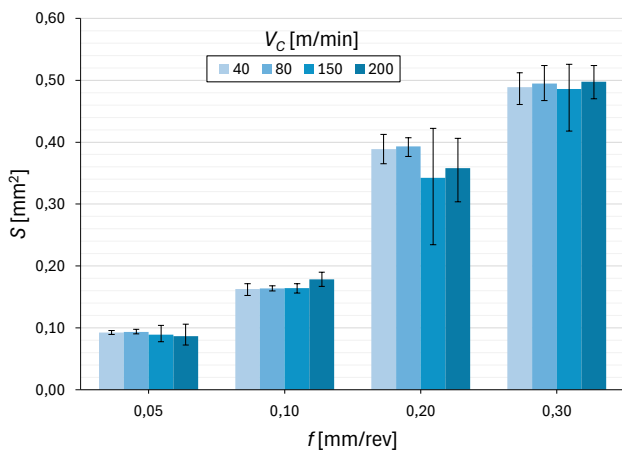
(f)

Figura 20. Valores medios de la altura de pico (h_p) para cada material (a), (b), (c), en función del avance (f) y la velocidad de corte (v_c). Valores medios de la altura de valle (h_v) para cada material (d), (e), (f), en función del avance (f) y la velocidad de corte (v_c).

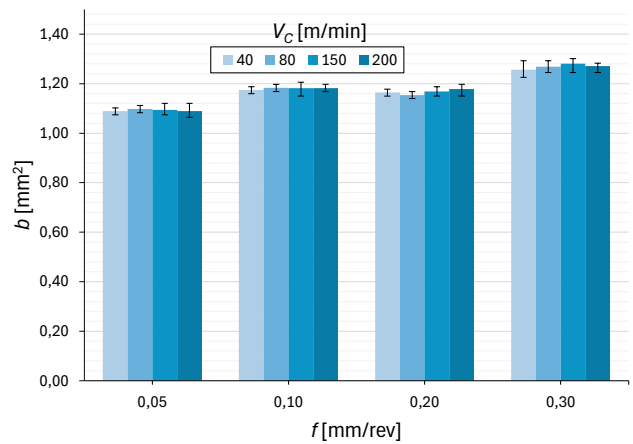
Para las aleaciones de aluminio, se observó en la Figura 20, que los valores de las alturas de la viruta (h_p y h_v) aumentan con el avance (f), siendo esta tendencia común para ambas aleaciones. En avances superiores, de 0,1 mm/r a 0,3 mm/r, el incremento porcentual es significativo. Para un avance de $f = 0,3$ mm/r, h_v es ligeramente menor en la aleación A97075, lo que resulta en una mayor diferencia de altura entre h_p y h_v en comparación con la A92024. Este comportamiento se puede explicar por la mayor capacidad de la aleación A97075 para soportar deformaciones antes de romperse, debido a su mayor plasticidad, especialmente a valores elevados de avance.

En cuanto a la aleación de titanio Ti6Al4V, se observó un aumento casi lineal en los parámetros de la viruta con la velocidad de avance, reflejando la proporcionalidad directa entre el espesor de la viruta y la velocidad de avance. A valores bajos de avance ($f = 0,10$ mm/r), h_p mostró una tendencia creciente con la velocidad de corte (v_c), mientras que a valores más altos de avance (0,20-0,30 mm/r), los valores máximos de h_p se registraron a una v_c de 30 m/min. No se observaron cambios significativos en h_v con variaciones de v_c , excepto a $f = 0,05$ mm/r, donde h_v fue ligeramente superior al disminuir v_c , presentando una tendencia opuesta a la observada a $f = 0,30$ mm/rev.

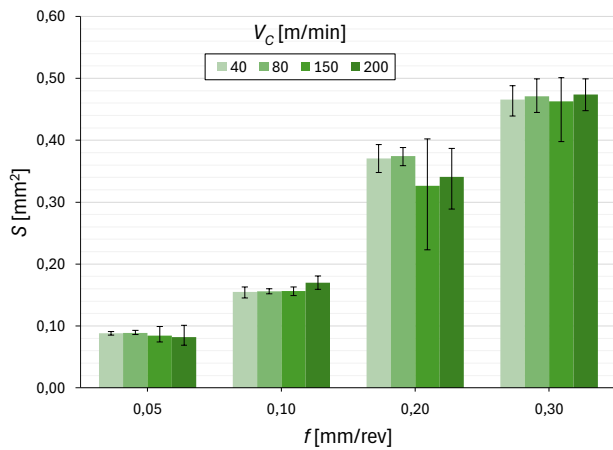
4.2.1. Área de sección transversal de la viruta (S) y anchura de la viruta (b)



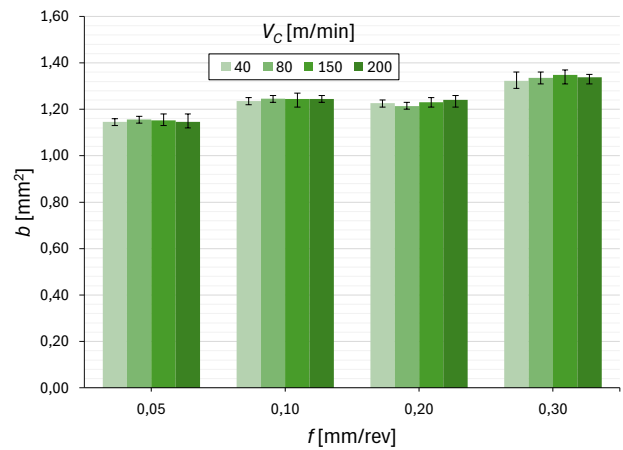
(c)



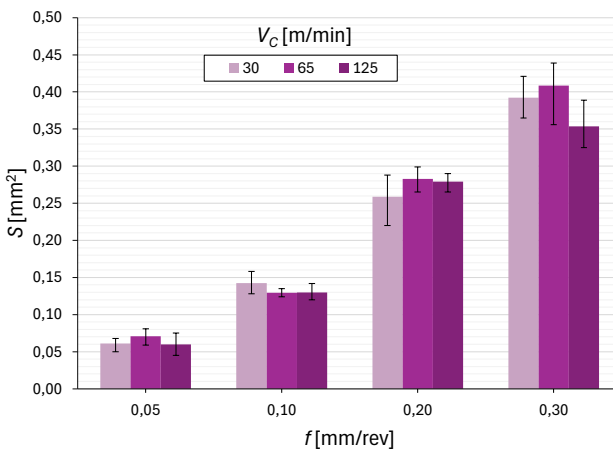
(d)



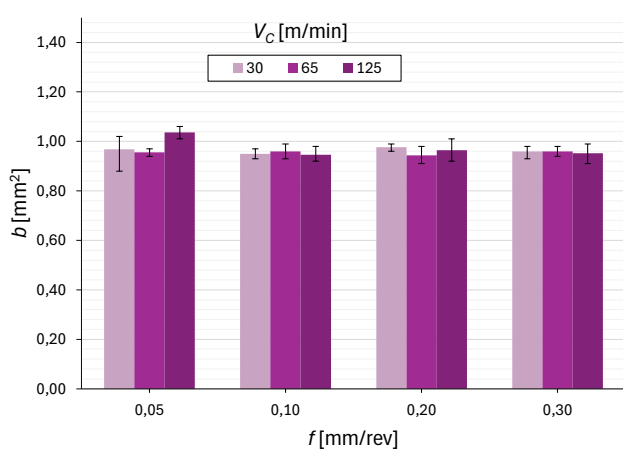
(c)



(d)



(e)



(f)

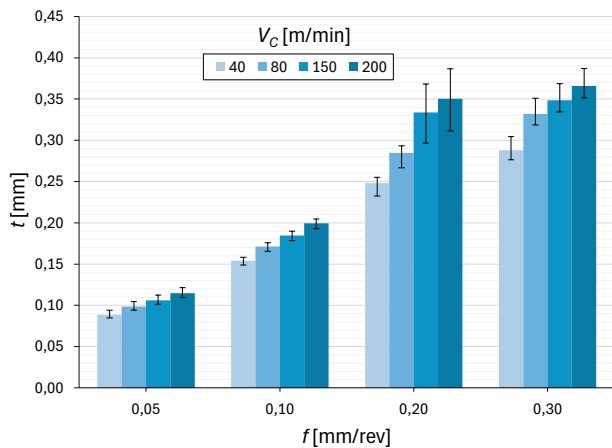
Figura 21. Valores medios de la sección de la viruta (S) para cada material (a), (b), (c), en función del avance (f) y la velocidad de corte (v_c). Valores medios del ancho de viruta (b) para cada material (d), (e), (f), en función del avance (f) y la velocidad de corte (v_c).

Se observaron en la Figura 21 tendencias específicas en la sección transversal de la viruta (S) y el ancho de la viruta (b) en función de los parámetros de corte. Para las aleaciones de aluminio, S aumentó de manera casi lineal con el avance (f), lo cual es coherente con la hipótesis de deformación plana asumida, donde se considera que b permanece constante. El valor medio de b se mantuvo casi constante al incrementar f , alineándose estrechamente con el valor teórico, con solo ligeros cambios en valores de avance más elevados. Específicamente para la aleación UNS A97075, estos cambios fueron más notables debido a una mayor deformación de la sección transversal de la viruta a mayores avances.

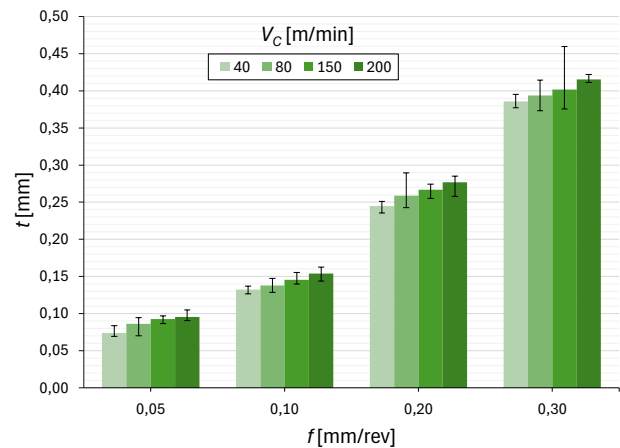
En el caso de la aleación de titanio Ti6Al4V, b tendió a permanecer constante, independientemente de los cambios en la velocidad de corte (v_c) y el avance (f), aproximándose mucho al valor teórico ($a_p = 1$ mm). Esto respalda la hipótesis de volumen constante y deformación plana durante el mecanizado. Por otro lado, el área transversal de la viruta (S)

aumentó con γ y fue independiente de v_c . Se midió a lo largo de la banda de cizalladura adiabática, estando muy próxima al producto de h_v y b . No obstante, S mostró una mayor dispersión para $f = 0,30$ mm/r, posiblemente porque el área transversal era menos homogénea y, en consecuencia, más difícil de medir a estos valores de avance.

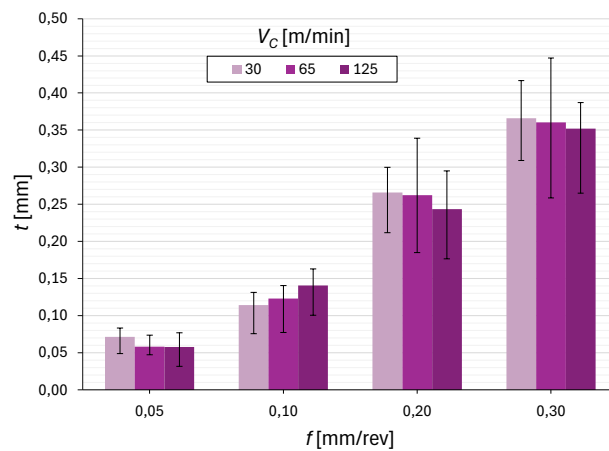
4.2.2. Espesor de viruta (t)



(a)



(b)



(c)

Figura 22. Valores medios del espesor de la viruta (t) para cada material (a), (b), (c), en función del avance (f) y la velocidad de corte (v_c).

Para todos los materiales, se observó (Figura 22) que los valores de t tienden a aumentar con el avance (f). Esta observación es coherente con la expectativa de que un mayor avance resulte en un incremento en el espesor de la viruta, debido a la mayor cantidad de material que se remueve por unidad de tiempo.

4.3. Análisis ANOVA de los datos experimentales

Para evaluar la significancia estadística de los efectos de los diferentes parámetros de corte sobre las características de la viruta, se realizó un análisis de varianza (ANOVA). Los resultados de este análisis se presentan detallando los grados de libertad (*D.F.*), la suma de cuadrados (*Sum_sq*), el valor F (*F-Value*) y la probabilidad resultante (*PR*), todos ellos parámetros estadísticos clave.

Tabla 4. Análisis ANOVA.

Variable	Source	D.F.	Sum_sq	F-Value	PR(>F)
<i>hp</i>	Vc	6	0,1182	13,9260	3,9615E-13
	f	3	3,8322	902,7560	6,8143E-113
	Interaction Vc-f	18	0,0392	1,5374	0,0807
	Residual	192	0,2717		
<i>hv</i>	Vc	6	0,0380	5,1749	5,8874E-05
	f	3	1,5682	426,6663	1,2390E-84
	Interaction Vc-f	18	0,0121	0,5495	0,9305
	Residual	192	0,2352		
<i>S</i>	Vc	6	0,1614	47,0301	3,4557E-35
	f	3	4,7580	2773,1257	9,0639E-158
	Interaction Vc-f	18	0,0683	6,6362	9,4008E-13
	Residual	192	0,1098		
<i>b</i>	Vc	6	2,5978	363,1534	6,4461E-102
	f	3	0,4287	119,8560	9,0567E-44
	Interaction Vc-f	18	0,2688	12,5272	8,2606E-24
	Residual	192	0,2289		
<i>t</i>	Vc	6	0,0713	11,1211	1,2297E-10
	f	3	2,5740	802,4575	2,5023E-108
	Interaction Vc-f	18	0,0165	0,8576	0,6304
	Residual	192	0,2053		

Los resultados de la Tabla 4 indican que tanto la velocidad de corte (v_c) como el avance (f) ejercen influencias significativas sobre todas las características de la viruta examinadas, ya que el valor de la probabilidad resultante es menor a 0,05. Cabe destacar que el avance (f) es el factor con el impacto más pronunciado, dado que la probabilidad asociada a sus efectos es extremadamente baja. No obstante, es la interacción entre ambos parámetros la que resulta crucial específicamente para la sección transversal (S) y la anchura (b) de la viruta. Esta interacción sugiere que la combinación de v_c y f afecta estas dimensiones de manera más intrincada.

4.4. Normalización de los datos experimentales

Tras el análisis inicial, como se indica en la metodología, el primer paso fue normalizar (de 0 a 1) las variables de entrada (*Material*, v_c y f) y salida (h_p , h_v , S , b , t) mediante la ecuación 11, con el fin de mejorar el rendimiento de la red neuronal, evitando los problemas de diferencia de escala en cada variable. En las Figuras 23 y 24 se muestran los histogramas con la distribución de las variables de entrada y salida utilizadas en la red neuronal.

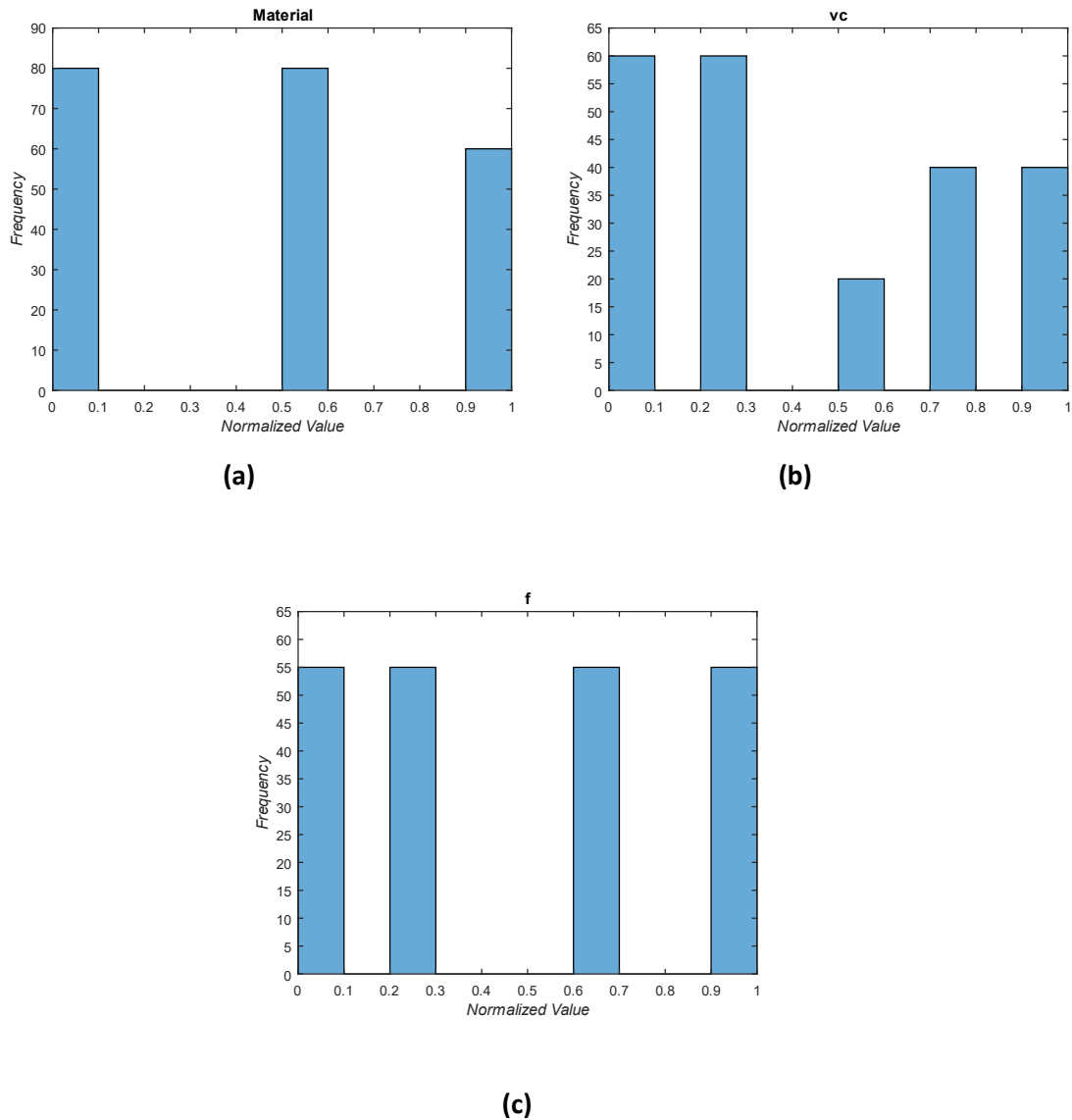


Figura 23. Normalización de las variables de entrada material (a), velocidad de corte (b) y avance (c).

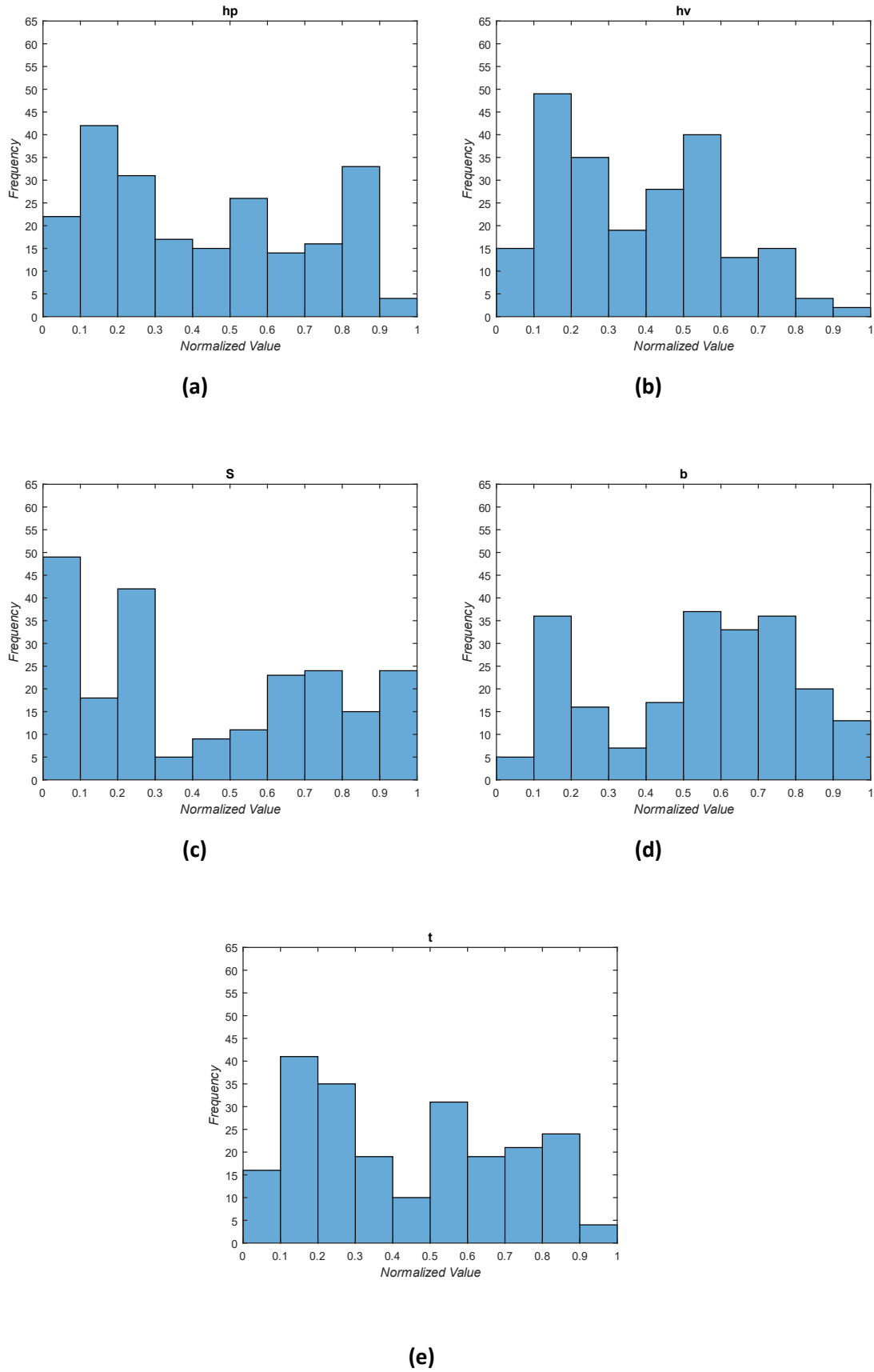


Figura 24. Normalización de las variables de salida altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).

4.5. Desempeño y validación de la red neuronal artificial

4.5.1. Número óptimo de neuronas en la capa oculta

Para determinar el número óptimo de neuronas en la capa oculta, se llevaron a cabo experimentos variando su cantidad en diferentes configuraciones de redes neuronales. La gráfica (Figura 25) muestran los datos del Error Cuadrático Medio Raíz, utilizado para evaluar la precisión de las salidas y la adecuación de la regresión lineal que describe los errores en la Red Neuronal Artificial. En estas gráficas, las líneas azules representan la evaluación de estos parámetros con los datos de entrenamiento de la red, mientras que las líneas rojas indican los resultados obtenidos al probar la red con datos no vistos anteriormente.

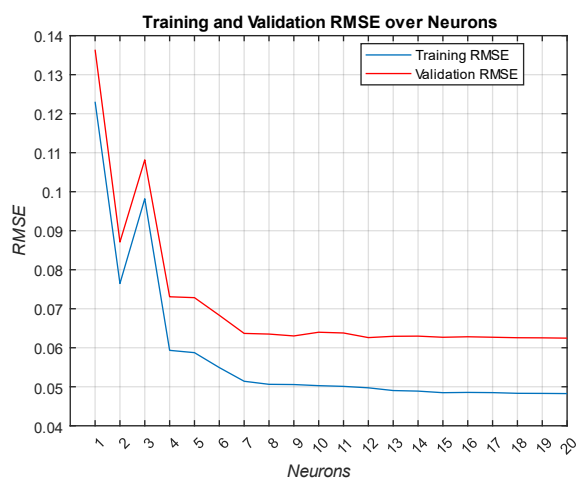


Figura 25. Valor del RMSE para distintos números de neuronas en la capa oculta.

La gráfica analizada muestra que el RMSE disminuye al aumentar el número de neuronas en la capa oculta de la red neuronal, lo que sugiere una mejora en el rendimiento del modelo con un mayor número de neuronas. Se observa un alto error inicial cuando hay pocas neuronas, indicativo de *underfitting*, donde el modelo es demasiado simple para capturar la complejidad de los datos. Se ha optado por utilizar 10 neuronas en la capa oculta, ya que, al incrementar el número de neuronas, el RMSE tiende a estabilizarse.

Esto indica que más allá de cierto punto, añadir neuronas adicionales no mejora significativamente el desempeño en datos no vistos, evitando así el *overfitting*. Este número proporciona un equilibrio adecuado, permitiendo que el modelo aprenda las características esenciales sin sobreajustarse a los ruidos o detalles superfluos de los datos de entrenamiento, demostrando una configuración bien ajustada que maximiza la capacidad de generalización del modelo.

4.5.2. Histogramas de errores

Avanzando directamente al análisis, los histogramas de error nos permiten observar la distribución de los errores cometidos por el modelo de red neuronal con 10 neuronas en la capa oculta. Esta visualización es crucial para identificar rápidamente la eficacia del modelo y determinar las áreas que requieren ajustes. A continuación, se examinan estos histogramas para evaluar el desempeño general del modelo en términos de precisión y consistencia de las predicciones.

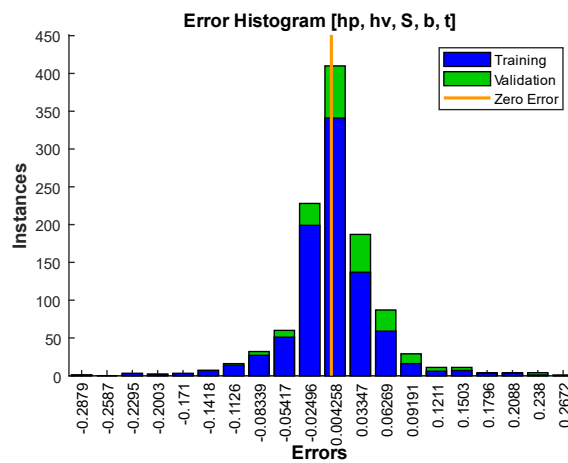
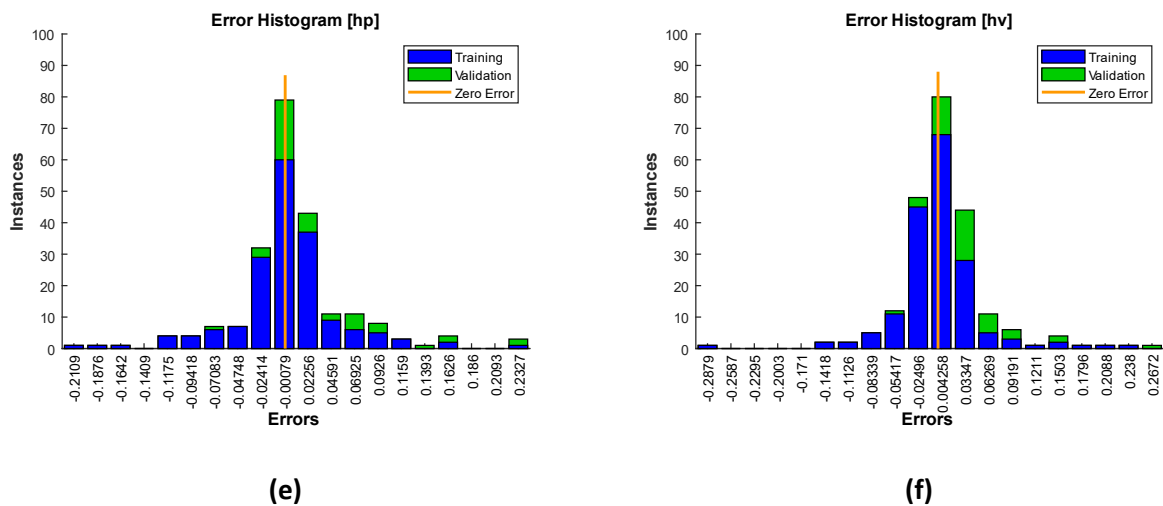


Figura 26. Histograma de errores de entrenamiento para el conjunto de datos normalizado.



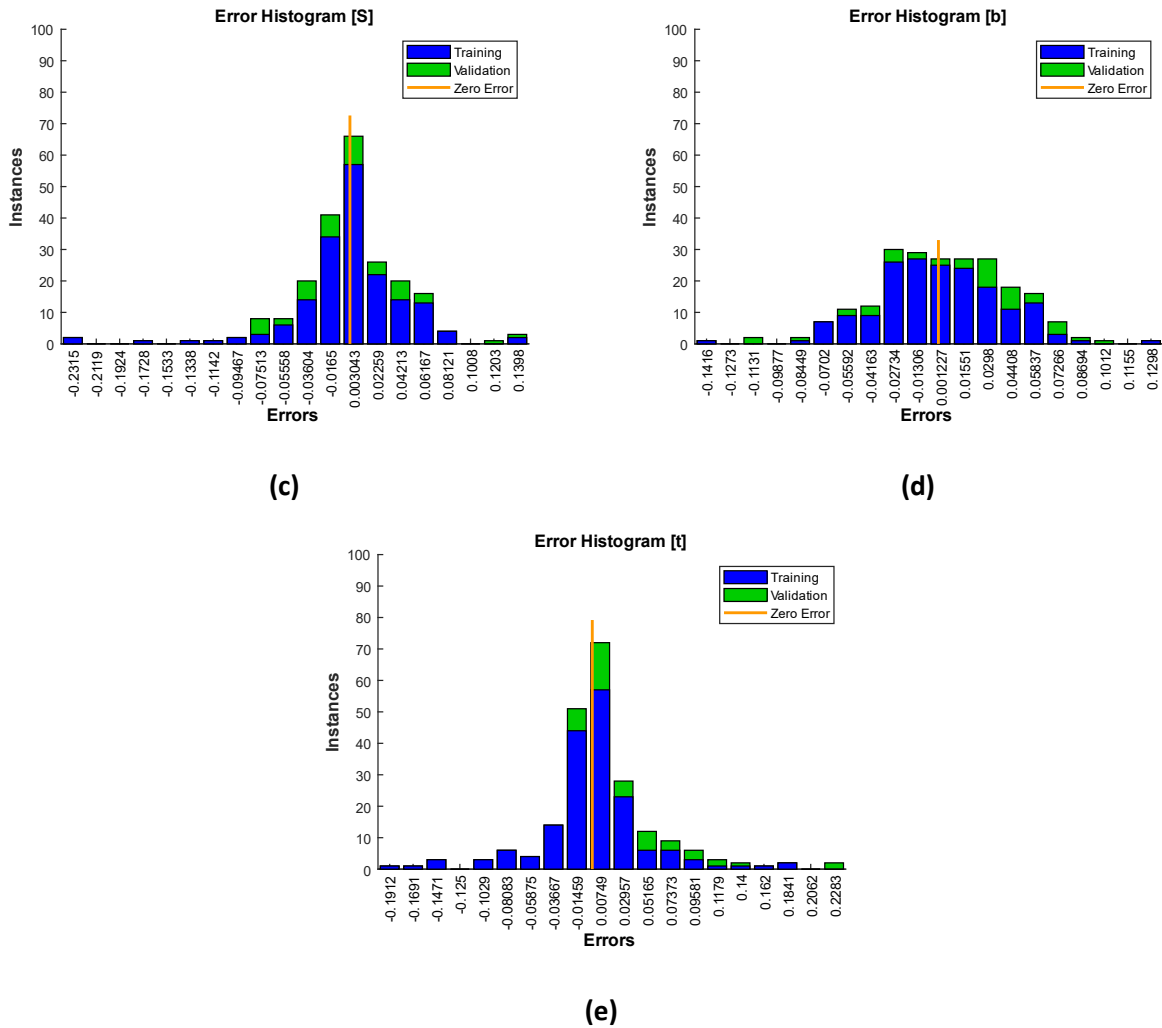
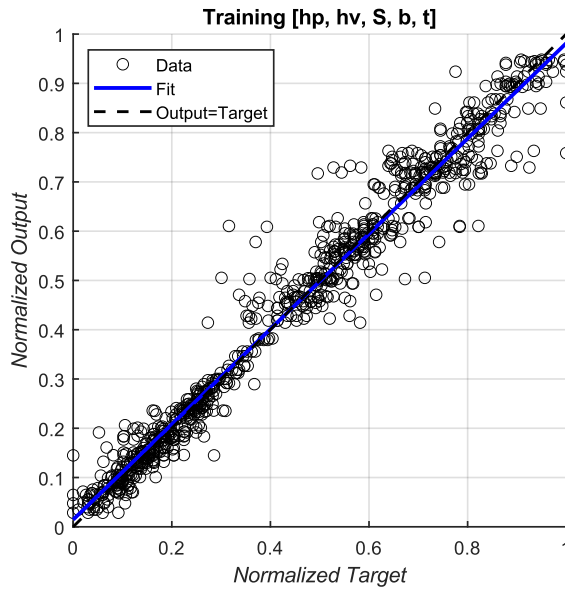


Figura 27. Histograma de errores de entrenamiento para cada variable de salida: altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).

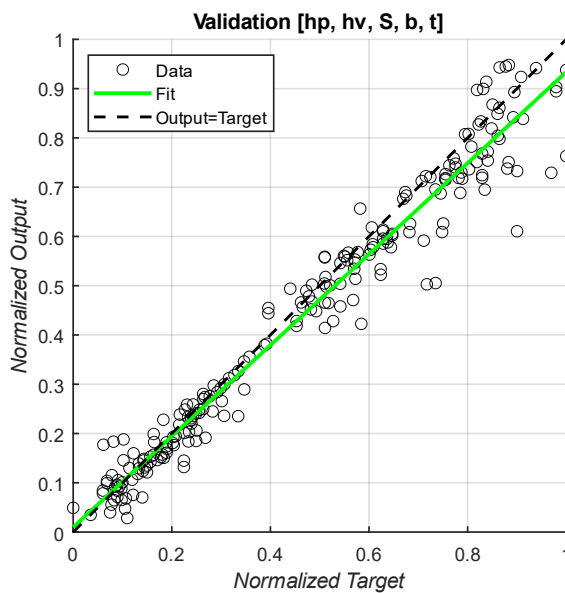
En la Figura 26 se observa el histograma de errores para todas las variables. En los de la Figura 27 se muestran la distribución de los errores para cinco variables diferentes (h_p , h_v , S , b , g). En todos los casos, la tendencia central está cerca de 0, indicando predicciones precisas sin sesgo significativo. Las distribuciones son simétricas y se asemejan a una normal, con la mayoría de los errores concentrados dentro de ± 0.05 , lo que sugiere desviaciones estándar bajas y alta precisión en las predicciones. Por ejemplo, para la variable h_p , la mayoría de los errores se encuentran entre -0.05 y 0.05, con una leve dispersión hasta ± 0.2 . Los histogramas de h_v y S presentan un comportamiento similar, con errores principalmente entre -0.04 y 0.04. En el caso de b , la dispersión de los errores es ligeramente mayor, abarcando desde -0.06 hasta 0.06. Para t , los errores también se centran en ± 0.05 , aunque algunos se extienden hasta ± 0.2 . Hay pocas desviaciones extremas, indicando que los modelos son robustos. La consistencia entre los errores de entrenamiento y validación en todas las gráficas sugiere que el modelo generaliza bien a datos no vistos.

4.5.3. Gráficos de regresión

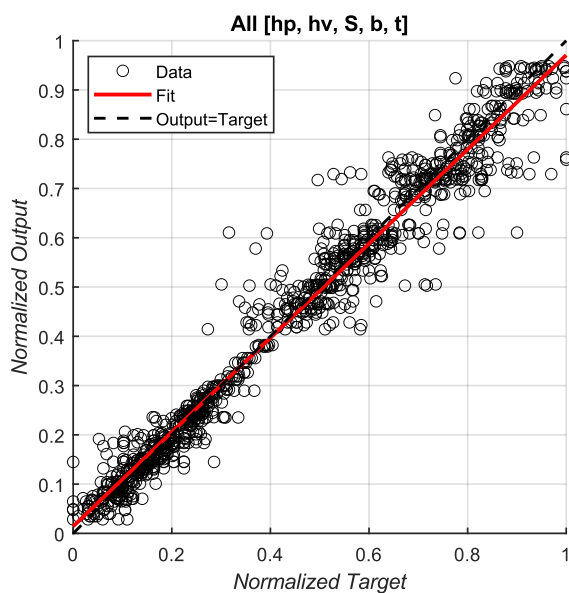
El siguiente paso consiste en explorar los modelos de regresión (Figura 28) derivados de la red neuronal. Estos modelos de regresión permitirán entender cómo las variables de entrada se relacionan con las predicciones de salida y cuál es el impacto cuantitativo de cada variable en el modelo. A continuación, se exploran los resultados obtenidos.



(a)



(b)



(c)

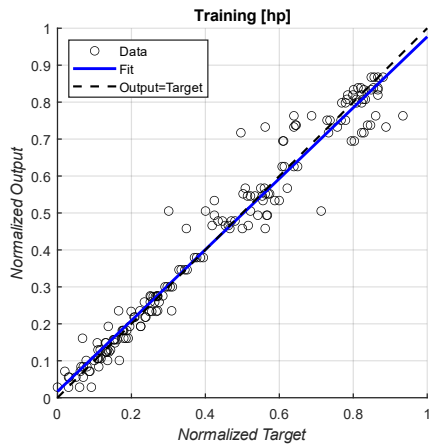
Figura 28. Resultados de la regresión: (a) Datos de entrenamiento; (b) Datos de validación; (c) Todo el conjunto de datos.

Tabla 5. RMSE y R^2 Ajustado de la red neuronal (10 neuronas).

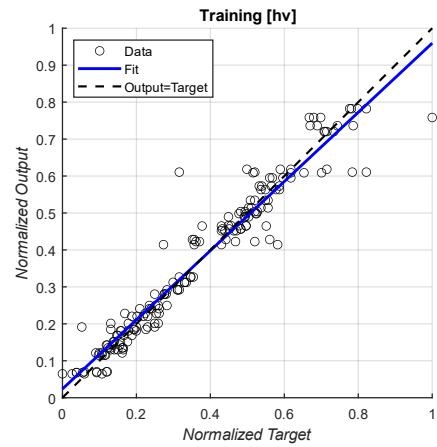
	RMSE			R2Adj		
	Training	Validation	ALL	Training	Validation	ALL
ALL	0,0496	0,0625	0,0524	0,9735	0,9474	0,9676

Como se observa en la tabla 4, el valor de R^2 ajustado en la fase de entrenamiento es 0,9735, cercano al valor obtenido en la fase de validación, que es de 0,9474. En el conjunto total de datos, el R^2 ajustado es de 0,9676, lo que indica un ajuste bastante razonable.

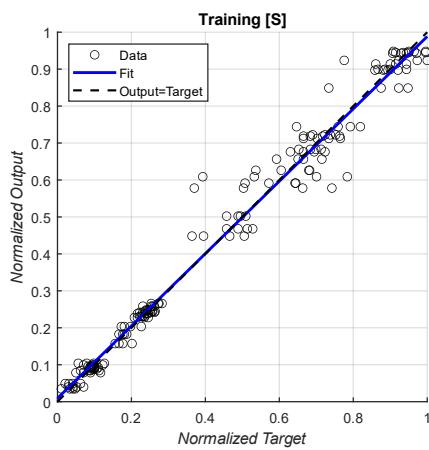
Las gráficas de regresión (Figura 28) presentadas ilustran una notable alineación de los puntos de datos con la línea ideal, indicando que la red neuronal logra una excelente correspondencia entre las salidas predichas y los valores objetivo. La precisión del modelo destaca por la proximidad de la mayoría de los puntos a la línea diagonal, con una pendiente cercana a 1 y una intersección mínima. A lo largo de las gráficas de entrenamiento (a), validación (b) y combinadas (c), la variabilidad entre las predicciones y los valores reales es consistentemente baja, evidenciando una capacidad de generalización robusta. Estos resultados técnicos confirman que el modelo es fiable y mantiene un alto grado de precisión en diferentes conjuntos de datos, sin exhibir síntomas de sobreajuste o subajuste.



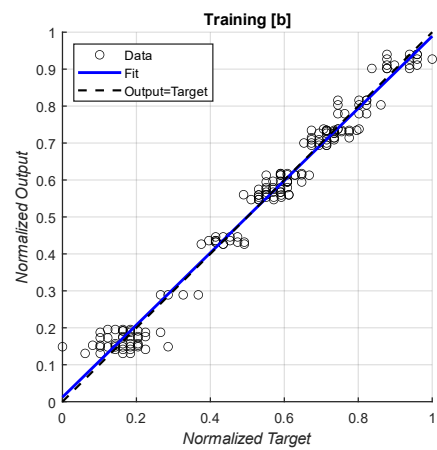
(g)



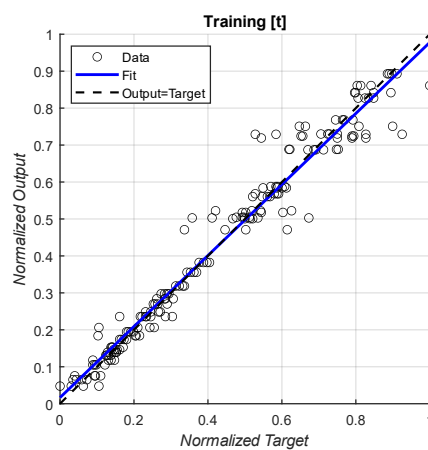
(h)



(c)



(d)



(e)

Figura 29. Resultados de regresión para cada variable de salida en el entrenamiento: altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).

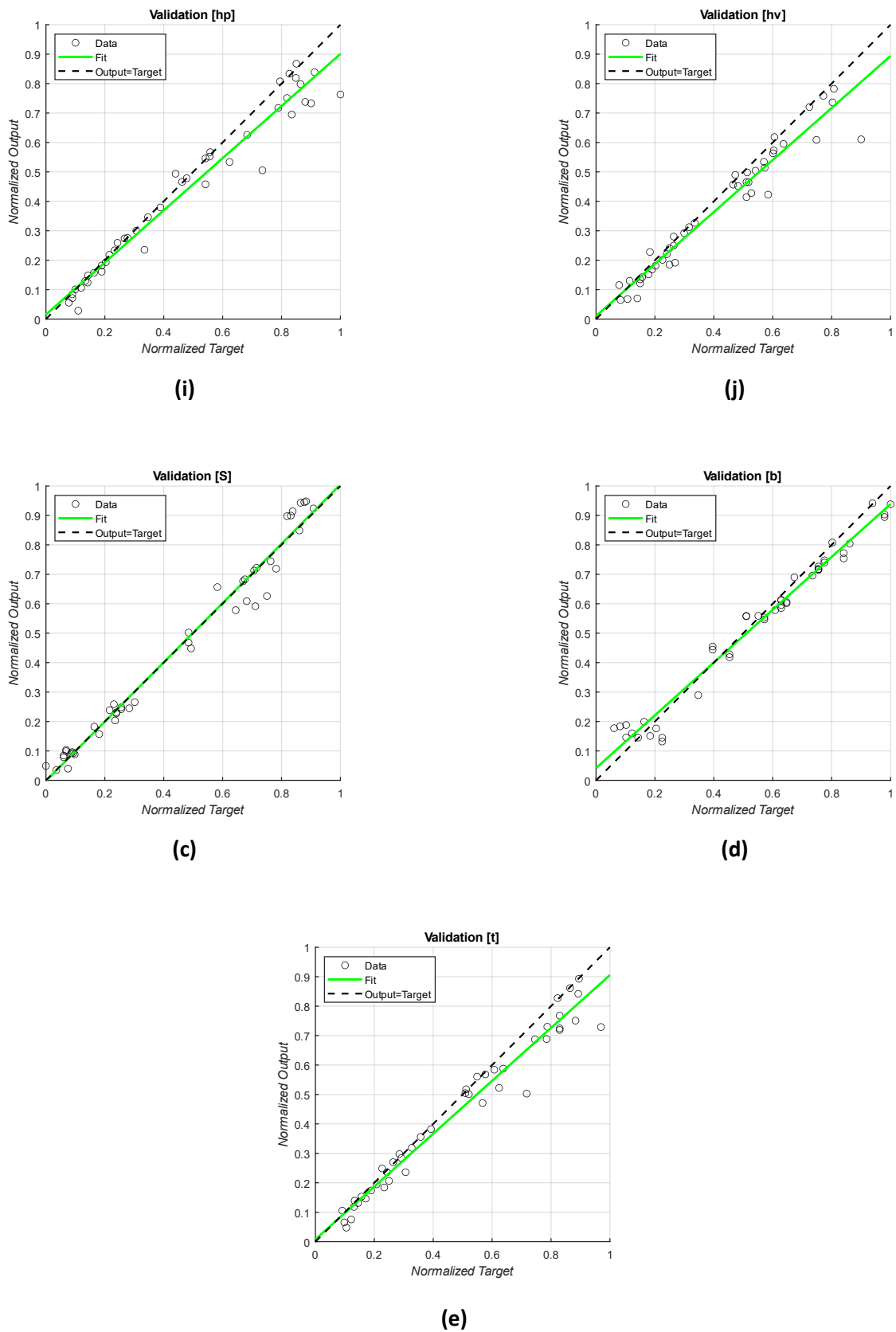
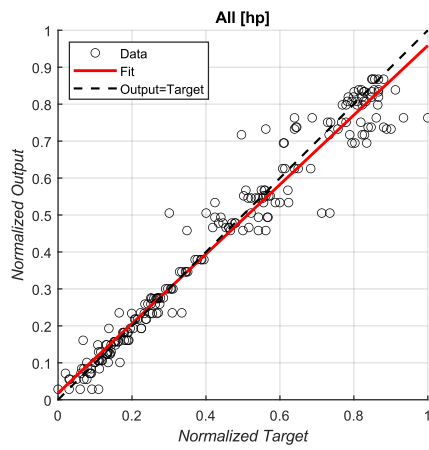
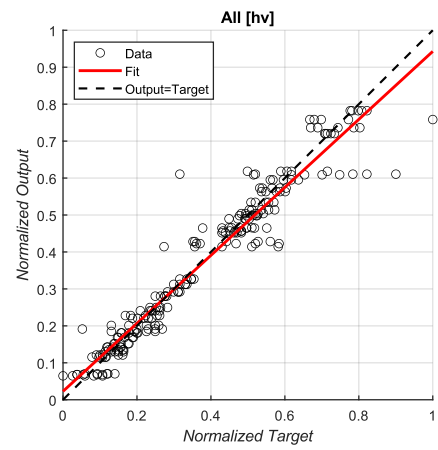


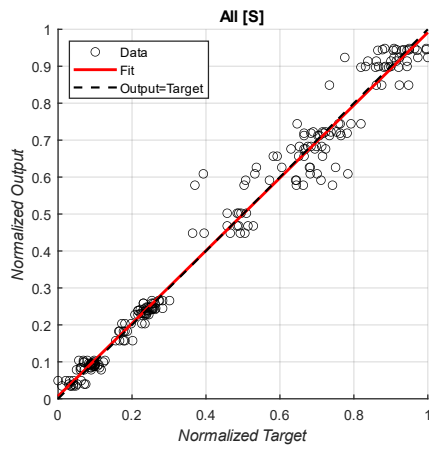
Figura 30. Resultados de regresión para cada variable de salida en validación: altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).



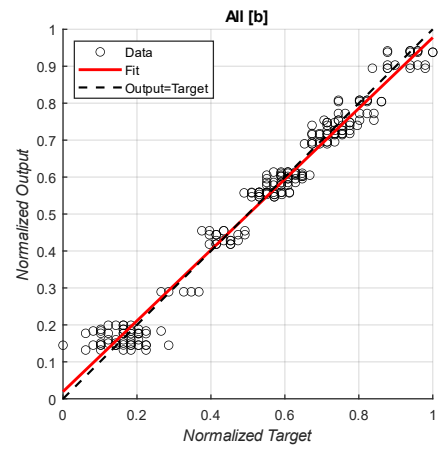
(k)



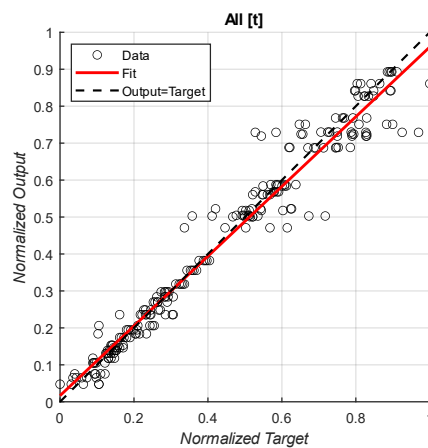
(l)



(c)



(d)



(e)

Figura 31. Resultados de regresión para cada variable de salida en conjunto: altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).

Tabla 6. RMSE y R^2 Ajustado de la RNA para el conjunto y para cada variable de salida.

	RMSE			R2Adj		
	Training	Validation	ALL	Training	Validation	ALL
<i>ALL</i>	0,0496	0,0625	0,0524	0,9735	0,9474	0,9676
<i>hp</i>	0,0544	0,0743	0,0589	0,9596	0,9335	0,9545
<i>hv</i>	0,0539	0,0655	0,0564	0,9362	0,9138	0,9324
<i>S</i>	0,0476	0,0474	0,0476	0,9774	0,9753	0,9772
<i>b</i>	0,0384	0,0518	0,0415	0,9768	0,9631	0,9740
<i>t</i>	0,0518	0,0693	0,0557	0,9606	0,9363	0,9560

En la tabla 6 se aprecia que a variable con mejor ajuste es *S* ($R^2Adj = 0,9772$), después le siguen *b* ($R^2Adj = 0,9740$), *t* ($R^2Adj = 0,9560$), *hp* ($R^2Adj = 0,9545$) y *hv* ($R^2Adj = 0,9324$). Sin embargo, el nivel de ajuste presentado por todas las variables puede considerarse correcto.

En las gráficas de entrenamiento, se observa una alineación cercana de los puntos de datos con la línea de ajuste. En particular, la Figura 29.a, muestra una correlación casi perfecta con la línea ideal, lo que indica un error mínimo y una captura de la variabilidad de los datos por parte del modelo. Las demás variables en las gráficas (Figura 29.b, Figura 29.c, Figura .d, Figura 29.e) siguen una tendencia similar, con una densidad de puntos que subraya una varianza muy baja y un ajuste preciso. Esto refleja no solo un aprendizaje efectivo sino también una buena retención de información sin caer en sobreajuste.

Durante la fase de validación, los gráficos muestran una ligera dispersión aumentada en comparación con las fases de entrenamiento, una indicación típica de la confrontación del modelo con nuevos datos. Sin embargo, las líneas de ajuste (Figura 30.b, Figura 30.c, Figura 30.d, Figura 30.e) siguen muy próximas a la línea ideal, lo que sugiere que el modelo generaliza adecuadamente a pesar de la variabilidad inevitable introducida por los datos no vistos durante el entrenamiento. Esta dispersión, aunque más notable, sigue siendo moderada, lo cual es indicativo de que el modelo se mantiene consistente y no muestra señales claras de *overfitting*.

Las representaciones que combinan todos los datos (Figura 31) de entrenamiento y validación muestran consistencia en el rendimiento del modelo. La alineación de los puntos con la línea de ajuste en cada una de las variables de salida, refleja una calibración efectiva del modelo a lo largo de diferentes conjuntos de datos, destacando su capacidad para realizar predicciones precisas en una amplia gama de condiciones operativas.

4.5.4. Comparación de modelos predictivos

En la metodología experimental se detalló el origen y la preparación de los datos utilizados. A continuación (Tabla 7), se presentarán las ecuaciones específicas de los modelos de regresión polinómicos y potenciales, junto con sus respectivos valores de $R^2_{ajustado}$, para evaluar cuantitativamente la capacidad predictiva y la precisión de cada modelo en el conjunto de datos.

Tabla 7. Modelos de regresión.

Modelo	Variable	Ecuación	R2Adj
Potencial	hp	$hp = 1,743 \times Vc^{-0,055} \times f^{0,963}$	0,9528
	hv	$hv = 0,833 \times Vc^{-0,016} \times f^{0,924}$	0,9096
	S	$S = 1,399 \times Vc^{-0,030} \times f^{0,960}$	0,9461
	b	$b = 0,908 \times Vc^{0,008} \times f^{-0,014}$	0,1756
	t	$t = 1,264 \times Vc^{-0,026} \times f^{0,946}$	0,9483
Polinómico	hp	$hp = -0,0091 + (1,7728 \times f) + (0,0003 \times Vc \times f) - (1,2281 \times f^2)$	0,9598
	hv	$hv = -0,0053 + (1,4003 \times f) + (0,0002 \times Vc \times f) - (1,5746 \times f^2)$	0,9169
	S	$S = -0,0151 - (0,0003 \times Vc) + (2,0689 \times f) + (0,0005 \times Vc \times f) - (1,6943 \times f^2)$	0,9509
	b	$b = 1,0745 - (0,0002 \times Vc) - (0,0085 \times f) + (0,0017 \times Vc \times f) + (0,6936 \times f^2)$	0,1918
	t	$t = -0,0070 + (1,5850 \times f) + (0,0003 \times Vc \times f) - (1,3956 \times f^2)$	0,9676
Red Neuronal	hp		0,9545
	hv		0,9324
	S		0,9772
	b		0,9740
	t		0,9560

En los modelos potenciales, el exponente de v_c (x) es mucho menor que el exponente de f (y) para todas las desviaciones geométricas. Esto sugiere que f tiene una influencia mayor que v_c . La influencia es mayor en h_p ($y = 0,964$), menor en h_v ($y = -0,014$) y mucho menor en b ($y = -0,014$). En todos los casos menos en b , el exponente es positivo, lo que indica un aumento de la desviación geométrica con f . El modelo presenta un ajuste razonable para todos los parámetros excepto para b , que el nivel de ajustes es muy bajo para ambos modelos, potencial y polinómico ($R^2_{adj} = 0,1756$ y $0,1918$, respectivamente). En definitiva, el análisis sugiere que estas desviaciones se ven más afectadas por f , y tienden a aumentar a medida que f aumenta.

Al comparar los ajustes obtenidos en estos modelos con el obtenido para la RNA, se observa que se consiguió un mejor ajuste utilizando la RNA para las variables de salida analizadas:

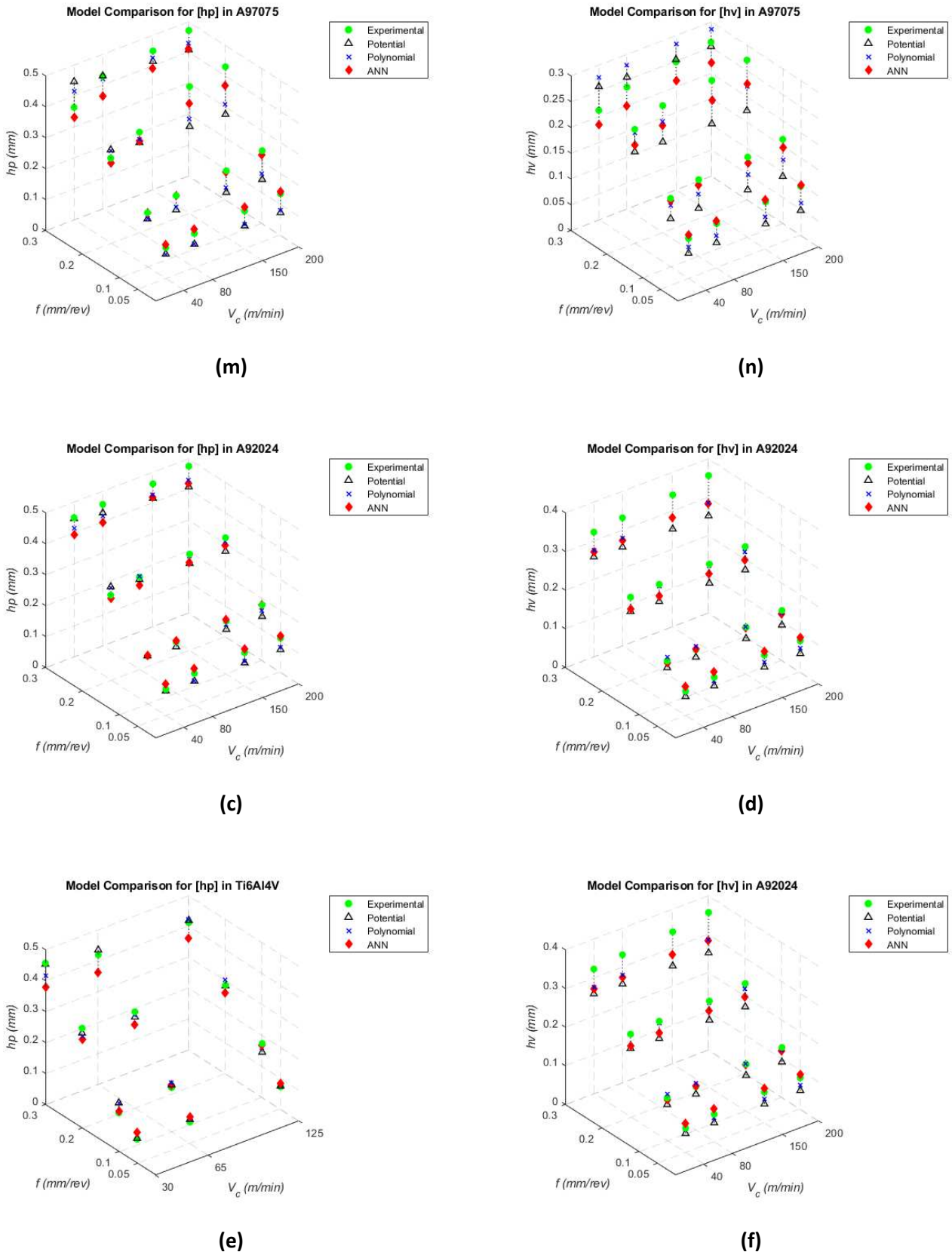


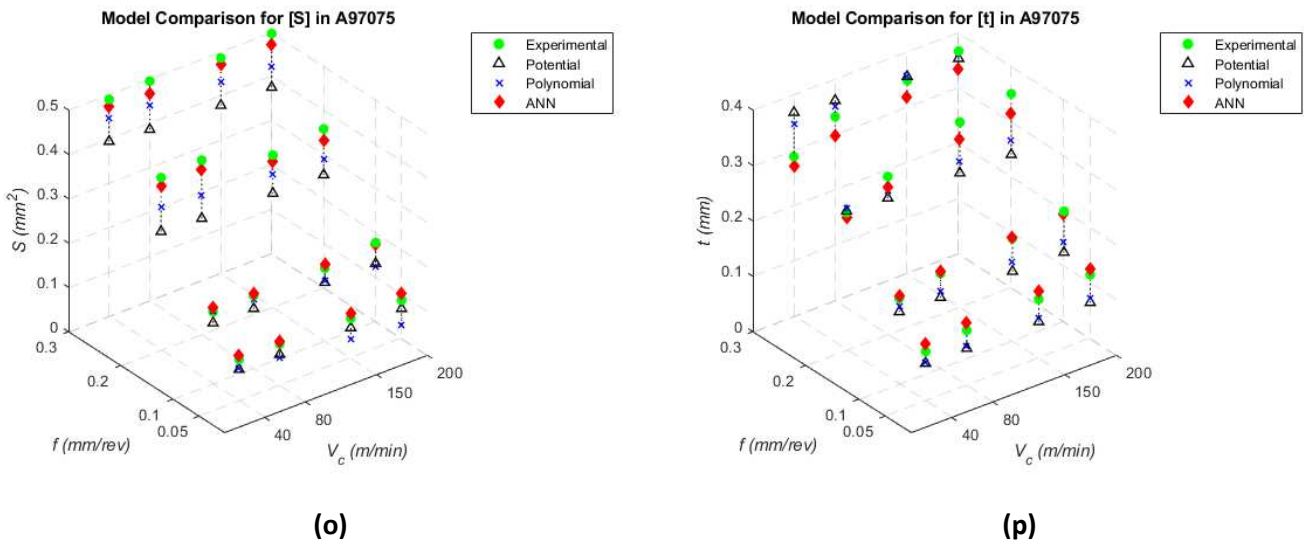
Figura 32. Resultados de regresión para cada variable de salida: altura de pico (a), altura de valle (b), sección de la viruta (c), ancho de la viruta (d) y espesor de la viruta (e).

En las gráficas presentadas (Figura 32), se observa un mejor desempeño del modelo de Red Neuronal Artificial en comparación con los modelos polinomiales y de potencial. En particular, cuando el avance f es de 0,05 mm/rev, todos los modelos coinciden adecuadamente con los datos experimentales. Sin embargo, a medida que f aumenta hasta 0,3 mm/rev, la variabilidad en las predicciones de los modelos se hace más evidente.

Para el caso específico de h_p en la aleación A97075, el modelo de RNA demuestra ser el más preciso, especialmente para velocidades de corte v_c de 150 y 200 mm/min y avances f de 0,2 y 0,3 mm/rev, con una variación en la predicción de alrededor de $\pm 0,06$ mm. Un comportamiento similar se observa para la aleación A92024, donde la mayor variabilidad también se manifiesta a un avance de 0,3 mm/rev, alcanzando valores de $\pm 0,05$ y $\pm 0,06$ mm.

En cuanto al parámetro h_v para la aleación A97075, el modelo de RNA mantiene el mejor ajuste, alcanzando su mayor alteración de aproximadamente $\pm 0,04$ mm en velocidades de corte avanzadas (150 y 200 mm/min) y avances de 0,2 y 0,3 mm/rev. Para la aleación A92024 en h_v , las fluctuaciones más significativas se registran a un avance de 0,3 mm/rev, con un valor aproximado de $\pm 0,07$ mm.

Es notable que los comportamientos para la aleación de titanio T6Al4V difieren de aquellos observados en las aleaciones de aluminio. A avances más altos, los modelos polinomiales y de potencial tienden a adaptarse mejor que el modelo de RNA, resaltando diferencias en la respuesta de los materiales bajo condiciones de corte similares.



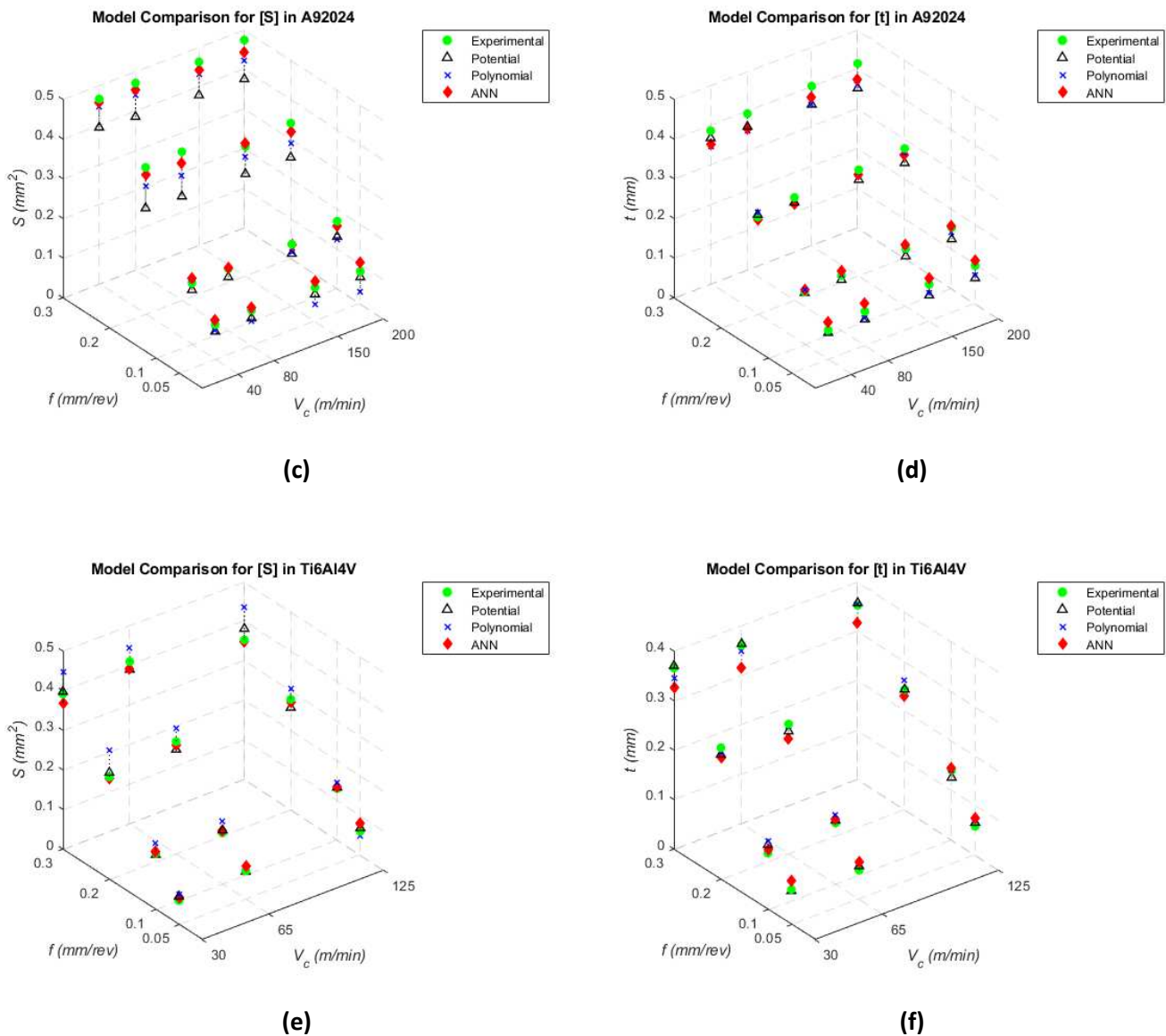


Figura 33. Comparación de modelos con datos experimentales para diferentes aleaciones: A97075 (*S* (a), *t* (d)); A92024 (*S* (b), *t* (e)); Ti6Al4V (*S* (c), *t* (f)).

Se destaca en la Figura 33 la notable adaptación del modelo de Red Neuronal Artificial a los datos experimentales en todas las aleaciones para la métrica de la sección de la viruta (*S*). Este modelo muestra una precisión superior en todos los rangos de mecanizado evaluados, con una dispersión máxima de $\pm 0,03$ en la aleación A92024 para una velocidad de corte v_c de 150 mm/min y un avance f de 0,3 mm/rev. En cuanto al espesor de la viruta (*t*), se observa un patrón de comportamiento parecido al de la sección *S*, con una coherencia comparativa entre los tres modelos evaluados. La desviación más significativa se registra como $\pm 0,04$ para un avance de 0,3 mm/rev en los materiales A92024 y titanio, indicando una consistencia entre diferentes condiciones de mecanizado y tipos de aleación.

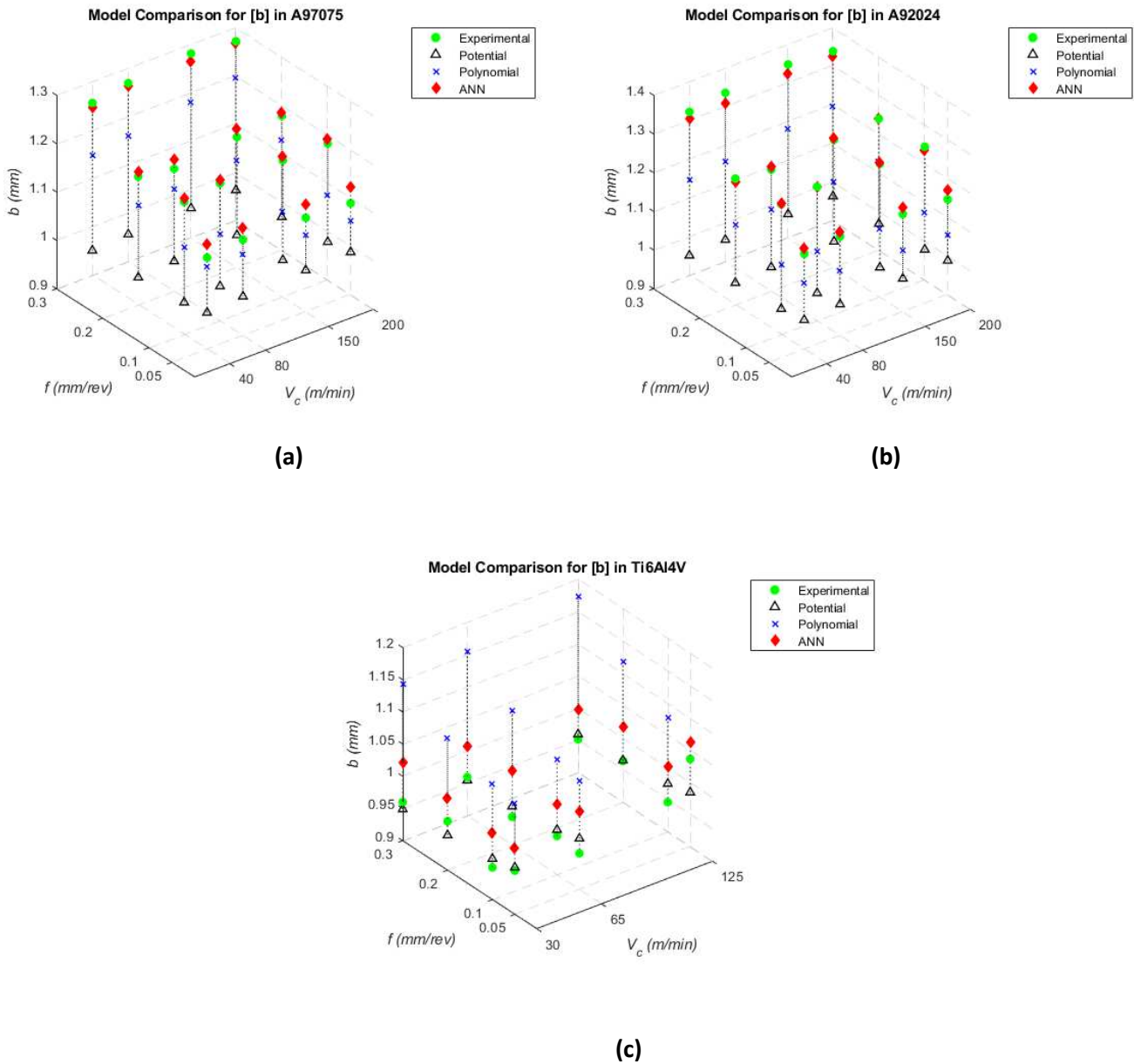


Figura 34. Comparación de modelos con datos experimentales para diferentes aleaciones: A97075 (b (a)); A92024 (b (b)); Ti6Al4V (b (c)).

En la Figura 34, se evidencia que los modelos potenciales y polinómicos exhiben un R^2 ajustado bajo, reflejando una considerable dispersión en las predicciones comparadas con los valores experimentales, lo cual puede estar relacionado con la dependencia del ancho de la viruta respecto a la profundidad de corte. Por otro lado, la Red Neuronal Artificial muestra un ajuste más preciso en las aleaciones de aluminio, mientras que en la aleación de titanio se observan mayores desviaciones.

Específicamente, las variaciones más pronunciadas, de aproximadamente $\pm 0,07$, se registran a una velocidad de corte (v_c) de 65 mm/min en la aleación de titanio, indicando una dispersión irregular en sus valores.

4.6. Predicción con Simulink

Después de desarrollar los modelos de redes neuronales artificiales, se ha procedido a integrarlos en el entorno de Simulink para exponer los datos y el proceso de una manera más gráfica. Esta aplicación, construida utilizando Simulink, proporciona una interfaz accesible y eficiente para el modelo.

La Figura 35 muestra la estructura del bloque de Red Neuronal Artificial en Simulink. Los parámetros de entrada se normalizan en un bloque de función y luego se envían al bloque de la red neuronal. Tras procesar los datos normalizados, se obtienen los resultados, que posteriormente se desnormalizan en bloques específicos para mostrar los valores finales.

La Figura 36 expone los detalles internos del bloque de Red Neuronal Artificial, incluyendo la normalización interna y las dos capas, oculta y de salida.

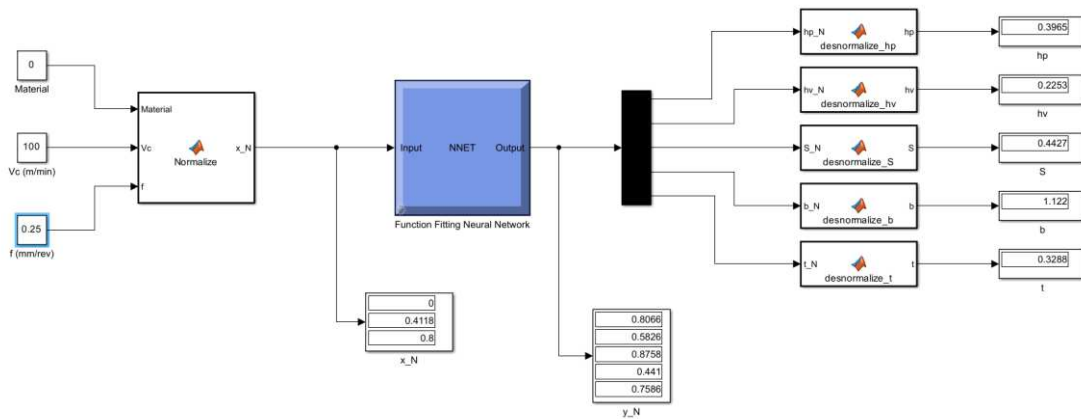


Figura 35. Diagrama de Simulink para la Implementación de una Red Neuronal.

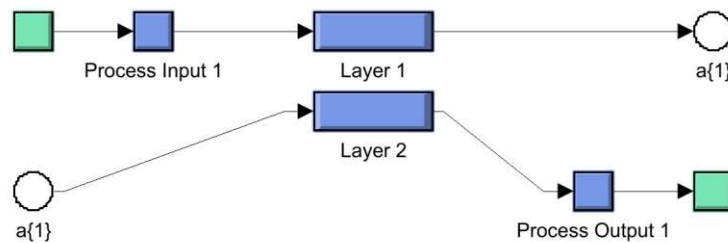


Figura 36. Composición del bloque de la Red Neuronal Artificial en Simulink.

La Figura 37 ilustra cómo están implementadas las neuronas artificiales en Simulink, destacando los sesgos y la función de activación empleada. Finalmente, la Figura 38 exhibe las conexiones de cada neurona en la capa oculta.

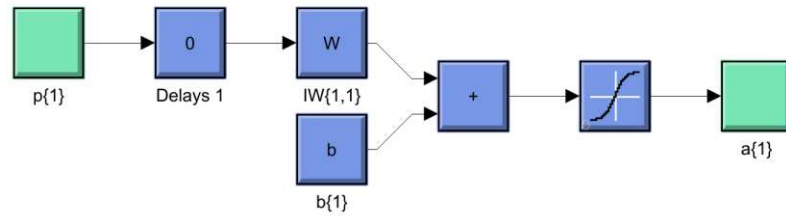


Figura 37. Estructura básica de un Neurona Artificial en Simulink.

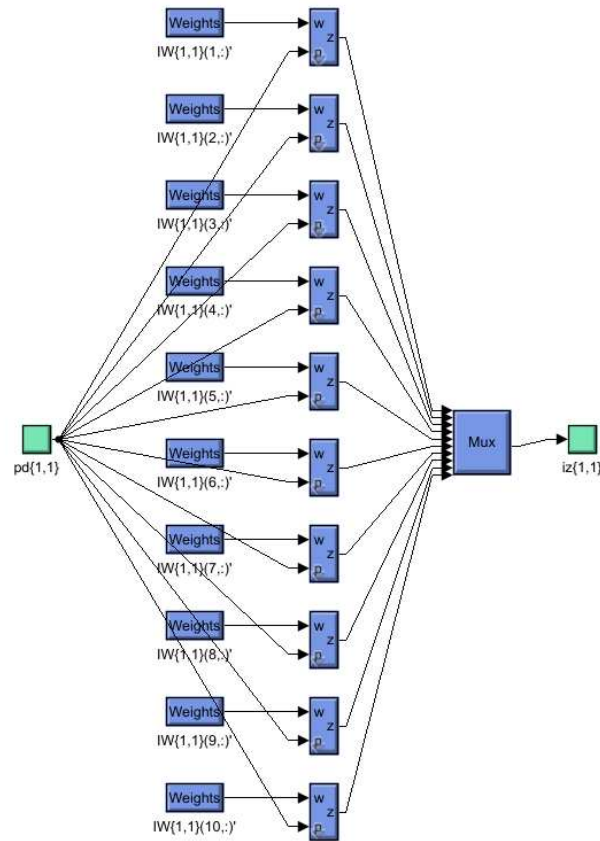


Figura 38. Diagrama de Pesos y Sesgos de la capa oculta.

5. Conclusiones y trabajos futuros

5.1. Conclusiones

El presente trabajo ha evaluado la efectividad de las redes neuronales artificiales en la predicción de parámetros geométricos relacionados con la morfología de la viruta durante el mecanizado en seco de aleaciones ligeras, comparándolas con modelos de regresión polinómicos y potenciales. Los resultados obtenidos permiten extraer varias conclusiones clave sobre la capacidad predictiva de los diferentes enfoques y sus aplicaciones en la industria aeronáutica.

En primer lugar, los modelos de regresión polinómicos y potenciales muestran que la influencia del avance f en las desviaciones geométricas es significativamente mayor que la velocidad de corte v_c , especialmente en la altura de pico h_p y la altura de valle h_v . Sin embargo, estos modelos presentan limitaciones en su capacidad predictiva, especialmente para el parámetro de ancho de viruta b , donde se observaron bajos valores de R^2 ajustado (0.1756 y 0.1918), lo que indica una baja precisión en la predicción de este parámetro.

Al comparar estos modelos con los resultados obtenidos mediante la RNA, se observa que la red neuronal ofrece un mejor ajuste para todas las variables analizadas. En particular, el modelo de RNA mostró una mayor precisión en la predicción de h_p y h_v en las aleaciones de aluminio A97075 y A92024, especialmente a mayores velocidades de corte y avances. La RNA fue capaz de reducir la variabilidad en las predicciones, manteniendo una variación en torno a

± 0.06 mm para h_p y ± 0.04 mm para h_v , lo que supera significativamente a los modelos polinómicos y potenciales.

En cuanto al titanio Ti6Al4V, la RNA también demostró un desempeño superior en la predicción de la sección de la viruta S y el espesor de la viruta t , aunque se observó que los modelos tradicionales tendían a adaptarse mejor a altas velocidades de corte, lo que resalta las diferencias en el comportamiento del material bajo condiciones de mecanizado similares.

Además, en la predicción del ancho de la viruta b , la RNA mostró un ajuste más preciso en las aleaciones de aluminio, aunque presentó mayores desviaciones en la aleación de titanio, con variaciones de hasta ± 0.07 mm a bajas velocidades de corte. Esto sugiere que, si bien la RNA es una herramienta potente, la complejidad del comportamiento del titanio bajo ciertas condiciones de corte requiere una optimización adicional de los modelos predictivos.

En conclusión, los resultados obtenidos validan la eficacia de las redes neuronales artificiales para predecir con mayor precisión las características geométricas de la viruta en el mecanizado en seco de aleaciones ligeras. Esta herramienta no solo mejora la exactitud de las predicciones en comparación con los modelos de regresión tradicionales, sino que también proporciona una base sólida para su aplicación en la optimización de procesos de mecanizado en la industria aeronáutica, donde el control de las desviaciones geométricas es crítico para garantizar la calidad y fiabilidad de los componentes fabricados.

5.2. Trabajos futuros

Aunque este estudio ha demostrado la eficacia de las redes neuronales artificiales para predecir parámetros geométricos clave en el mecanizado en seco de aleaciones ligeras, existen varias áreas en las que se podría profundizar para optimizar y expandir los resultados obtenidos. Una dirección futura importante sería la diversificación y ampliación del conjunto de datos experimentales utilizados para entrenar la RNA. Recoger datos bajo diferentes condiciones de corte, así como incluir nuevas aleaciones y variaciones en los parámetros de mecanizado, podría mejorar la capacidad del modelo para generalizar y predecir con mayor precisión en un rango más amplio de escenarios industriales.

Otra línea de investigación prometedora sería la implementación de arquitecturas de redes neuronales más complejas, como redes neuronales profundas que incluyan múltiples capas ocultas. Esta evolución permitiría al modelo capturar patrones más intrincados y relaciones no lineales entre las variables, lo que podría resultar en una mejora significativa en la predicción de desviaciones geométricas y otras características críticas del proceso de mecanizado.

Además, sería beneficioso considerar la incorporación de parámetros adicionales en el modelo, como el desgaste de la herramienta, la temperatura de corte, y las propiedades tribológicas del material. Estos factores podrían proporcionar una visión más holística del proceso de mecanizado y aumentar la precisión del modelo en la predicción de comportamientos complejos, particularmente en condiciones de operación extremas.

En términos de aplicabilidad, extender la metodología a otros procesos de mecanizado, como el torneado, el fresado y la rectificación, podría ofrecer una comprensión más integral de cómo los parámetros de corte influyen en la calidad superficial en diferentes contextos de fabricación. Esto no solo mejoraría la versatilidad del modelo, sino que también ampliaría su utilidad en una variedad de aplicaciones industriales, incluyendo aquellas donde la precisión geométrica es crítica.

Por último, integrar las RNA desarrolladas en sistemas de control en tiempo real para la optimización automatizada del proceso de mecanizado podría representar un avance significativo en la industria. Este enfoque permitiría ajustar dinámicamente los parámetros de corte en respuesta a las condiciones operativas, mejorando la eficiencia y la calidad del producto final. Este desarrollo también podría ser particularmente útil en la implementación de procesos de fabricación más sostenibles y eficientes, alineándose con las tendencias actuales de la industria.

6. Bibliografía

- [1] K. E. Y. Points, “Air Passenger Market Analysis,” no. December, pp. 1–4, 2013.
- [2] IATA, “Heightened Policy Uncertainty,” pp. 1–5, 2024.
- [3] Ministerio de Defensa, “El Plan de Acción Europeo de la Defensa,” pp. 1–13, 2017.
- [4] C. y T. Ministerio de Industria and TEDAE, “Agenda sectorial de la Industria Aeronáutica,” pp. 1–37, 2018.
- [5] MINTUR, “Precio Neto De La Electricidad Para Uso Doméstico Y Uso Industrial,” 2023.
- [6] PAE, “AGENDA ESTRATÉGICA DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN EN AERONÁUTICA 2019 – 2030,” vol. 7, no. 3, p. 6, 2021.
- [7] “Aluminio - Wikipedia, la enciclopedia libre,” 2023. [Online]. Available: <https://es.wikipedia.org/wiki/Aluminio>. [Accessed: 01-Sep-2024].
- [8] A. E. Paramétrico Del Mecanizado Seco, F. Javier Trujillo Vilches, and D. D. Lorenzo Sevilla Hurtado Mariano Marcos Bárcena, “Universidad De Málaga Departamento De Ingeniería Civil, De Materiales Y Fabricación Tesis Doctoral,” 2013.
- [9] W. A. Alloys, “International Alloy Designations and Chemical Composition Limits for Wrought Aluminum and Wrought Aluminum Alloys,” *Alum. Assoc. Arlington, Virginia*, vol. ISSN: 2377, no. August, p. 34, 2018.
- [10] L. Zhu, N. Li, and P. R. N. Childs, “Light-weighting in aerospace component and system design,” *Propuls. Power Res.*, vol. 7, no. 2, pp. 103–119, 2018.
- [11] “Titanio - Wikipedia, la enciclopedia libre.” [Online]. Available:

- <https://es.wikipedia.org/wiki/Titanio>. [Accessed: 01-Sep-2024].
- [12] J. C. Williams and R. R. Boyer, “Opportunities and issues in the application of titanium alloys for aerospace components,” *Metals (Basel)*, vol. 10, no. 6, 2020.
- [13] Z. Liu, B. He, T. Lyu, and Y. Zou, “A Review on Additive Manufacturing of Titanium Alloys for Aerospace Applications: Directed Energy Deposition and Beyond Ti6Al4V,” *Jom*, vol. 73, no. 6, pp. 1804–1818, 2021.
- [14] S. J. Skerlos, K. F. Hayes, A. F. Clarens, and F. Zhao, “Current advances in sustainable Metalworking Fluids research,” *Int. J. Sustain. Manuf.*, vol. 1, no. 1–2, pp. 180–202, 2008.
- [15] E. O. Bennett, “Water based cutting fluids and human health,” *Tribol. Int.*, vol. 16, no. 3, pp. 133–136, 1983.
- [16] Y. S. Hernández, F. J. T. Vilches, C. B. Gamboa, and L. S. Hurtado, “Experimental parametric relationships for chip geometry in dry machining of the Ti6Al4V alloy,” *Materials (Basel)*, vol. 10, no. 7, 2018.
- [17] F. Klocke, S. Gierlings, M. Brockmann, and D. Veselovac, “Influence of temperature on surface integrity for typical machining processes in aero engine manufacture,” *Procedia Eng.*, vol. 19, pp. 203–208, 2011.
- [18] J. F. Kelly and M. G. Cotterell, “Minimal lubrication machining of aluminium alloys,” *J. Mater. Process. Technol.*, vol. 120, no. 1–3, pp. 327–334, 2002.
- [19] A. Thakur and S. Gangopadhyay, “Dry machining of nickel-based super alloy as a sustainable alternative using TiN/TiAlN coated tool,” *J. Clean. Prod.*, vol. 129, pp. 256–268, 2016.
- [20] G. S. Goindi and P. Sarkar, “Dry machining: A step towards sustainable machining – Challenges and future directions,” *J. Clean. Prod.*, vol. 165, pp. 1557–1571, 2017.
- [21] “XIMA Herramientas de Corte Industrial Insertos de Carburo.” [Online]. Available: <https://ximacorte.com/productos.html>. [Accessed: 01-Sep-2024].
- [22] F. J. T. Vilches, L. S. Hurtado, F. M. Fernández, and C. B. Gamboa, “Analysis of the chip geometry in dry machining of aeronautical aluminum alloys,” *Appl. Sci.*, vol. 7, no. 2, 2017.
- [23] A. Baba, “Neural networks from biological to artificial and vice versa,” *BioSystems*, vol. 235, 2024.
- [24] “El Perceptrón Multicapa - Introducción.” [Online]. Available: https://juansensio.com/blog/023_mlp_backprop. [Accessed: 02-Sep-2024].
- [25] A. du Preez and G. A. Oosthuizen, “Machine learning in cutting processes as enabler for smart sustainable manufacturing,” *Procedia Manuf.*, vol. 33, pp. 810–817, 2019.
- [26] D. H. Kim *et al.*, “Smart Machining Process Using Machine learning: A Review and Perspective on Machining Industry,” *Int. J. Precis. Eng. Manuf. - Green*

- Technol.*, vol. 5, no. 4, pp. 555–568, 2018.
- [27] T. Misaka, J. Herwan, I. Ogura, and Y. Furukawa, “Turning Process Monitoring with Deep Neural Network Trained by FEM Simulation,” *Procedia CIRP*, vol. 104, pp. 376–380, 2021.
- [28] M. Mia and N. R. Dhar, “Response surface and neural network based predictive models of cutting temperature in hard turning,” *J. Adv. Res.*, vol. 7, no. 6, pp. 1035–1044, 2016.
- [29] U. Maheshwera Reddy Paturi, H. Devarasetti, and S. Kumar Reddy Narala, “Application of Regression and Artificial Neural Network Analysis in Modelling of Surface Roughness in Hard Turning of AISI 52100 Steel,” *Mater. Today Proc.*, vol. 5, no. 2, pp. 4766–4777, 2018.
- [30] M. R. Sankar *et al.*, “Experimental study and artificial neural network modeling of machining with minimum quantity cutting fluid,” *Mater. Today Proc.*, vol. 18, pp. 4921–4931, 2019.
- [31] Y. Sanchez, F. J. Trujillo, L. Sevilla, M. Batista, and M. Marcos, “Técnicas Metalográficas para la Caracterización Morfológica de la Viruta en el Mecanizado de Aleaciones Ligeras de Uso Aeronáutico,” *Técnica y Tecnol. Mec. para la Ind. Aeronáutica*, vol. 23, pp. 14–22, 2014.

Anexos A: Código de Matlab

1.- Importación de los datos de Excel

```
clear all
```

```
clc
```

```
% Importamos los datos de la hoja excel colocada en el directorio C:\Users\usuario\Documents\MATLAB
```

```
data = readmatrix ("General.xlsx");
```

2.- Definición los datos de entrada x (Vc,f) y de salida (hp, hv, S, b, t)

```
x = data (:,1:3); % Extraemos todas las filas y las columnas
```

```
y = data (:,7:19);
```

```
Material = round(x(:,1),3); % Definimos la variable Material como la primera columna de x, redondeado a 3 decimales
```

```
Vc = round(x(:,2),3); % Definimos la variable Vc como la segunda columna de x, redondeado a 3 decimales
```

```
f = round(x(:,3),3); % Definimos la variable f como la tercera columna de x, redondeado a 3 decimales
```

```
hp = round(y(:,1),3); % Definimos la variable hp como la primera columna de y, redondeado a 3 decimales
```

```
hv = round(y(:,2),3); % Definimos la variable hv como la segunda columna de y, redondeado a 3 decimales
```

```
S = round(y(:,6),3); % Definimos la variable S como la sexta columna de y, redondeado a 3 decimales
```

```
b = round(y(:,7),3); % Definimos la variable b como la septima columna de y, redondeado a 3 decimales
```

```
t = round(y(:,13),3); % Definimos la variable t como la decimotercera columna de y, redondeado a 3 decimales
```

3.- Normalización de datos de entrada y salida (0,1)

```
Material_n = (Material-min(Material))/(max(Material)-min(Material)); % Normalización de Material para que sus valores estén comprendidos entre 0 y 1
```

```
Vc_n= (Vc-min(Vc))/(max(Vc)-min(Vc)); % Normalización de Vc para que sus valores estén comprendidos entre 0 y 1
```

```
f_n= (f-min(f))/(max(f)-min(f)); % Normalización de f para que sus valores estén comprendidos entre 0 y 1
```

```
hp_n= (hp-min(hp))/(max(hp)-min(hp)); % Normalización de hp para que sus valores estén comprendidos entre 0 y 1
```

```
hv_n= (hv-min(hv))/(max(hv)-min(hv)); % Normalización de hv para que sus valores estén comprendidos entre 0 y 1
```

```
S_n= (S-min(S))/(max(S)-min(S)); % Normalización de S para que sus valores estén comprendidos entre 0 y 1
```

```
b_n= (b-min(b))/(max(b)-min(b)); % Normalización de b para que sus valores estén comprendidos entre 0 y 1
```

```
t_n= (t-min(t))/(max(t)-min(t)); % Normalización de t para que sus valores estén comprendidos entre 0 y 1
```

4.- Histogramas de las variables normalizadas

```
figure
```

```
histogram(Material_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);
```

```
title('Material');
```

```
xlabel('Normalized Value', 'FontAngle', 'italic');
```

```
ylabel('Frequency', 'FontAngle', 'italic');
```

```
ylim([0 90]); % Establece los límites del eje y
```

```
yticks(0:10:90); % Establece los ticks del eje y
```

```
xlim([0 1]); % Establece los límites del eje x
```

```
xticks(0:0.1:1); % Establece los ticks del eje x
```

```
figure;
```

```
histogram(Vc_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);
```

```
title('vc');
```

```
xlabel('Normalized Value', 'FontAngle', 'italic');
```

```
ylabel('Frequency', 'FontAngle', 'italic');
```

```
ylim([0 65]);
```

```
yticks(0:5:65);
```

Anexos A. Código de Matlab

```
xlim([0 1]);  
xticks(0:0.1:1);
```

```
figure;  
histogram(f_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);  
title('f');  
xlabel('Normalized Value', 'FontAngle', 'italic');  
ylabel('Frequency', 'FontAngle', 'italic');  
ylim([0 65]);  
yticks(0:5:65);  
xlim([0 1]);  
xticks(0:0.1:1);
```

```
figure;  
histogram(hp_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);  
title('hp');  
xlabel('Normalized Value', 'FontAngle', 'italic');  
ylabel('Frequency', 'FontAngle', 'italic');  
ylim([0 65]);  
yticks(0:5:65);  
xlim([0 1]);  
xticks(0:0.1:1);
```

```
figure;  
histogram(hv_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);  
title('hv');  
xlabel('Normalized Value', 'FontAngle', 'italic');  
ylabel('Frequency', 'FontAngle', 'italic');  
ylim([0 65]);  
yticks(0:5:65);  
xlim([0 1]);  
xticks(0:0.1:1);
```

```
figure;  
histogram(S_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);
```

```

title('S');
xlabel('Normalized Value', 'FontAngle', 'italic');
ylabel('Frequency', 'FontAngle', 'italic');
ylim([0 65]);
yticks(0:5:65);
xlim([0 1]);
xticks(0:0.1:1);

figure;
histogram(b_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);
title('b');
xlabel('Normalized Value', 'FontAngle', 'italic');
ylabel('Frequency', 'FontAngle', 'italic');
ylim([0 65]);
yticks(0:5:65);
xlim([0 1]);
xticks(0:0.1:1);

figure;
histogram(t_n, 'BinEdges', 0:0.1:1, 'Normalization', 'count', 'FaceColor', [0 0.4470 0.7410]);
title('t');
xlabel('Normalized Value', 'FontAngle', 'italic');
ylabel('Frequency', 'FontAngle', 'italic');
ylim([0 65]);
yticks(0:5:65);
xlim([0 1]);
xticks(0:0.1:1);

```

6.- Definición de los datos de entrenamiento y de validación

X = [Material_n, Vc_n, f_n]; % crear la matriz de datos de entrada a la ANN, invirtiendo filas por columnas

Y = [hp_n, hv_n, S_n, b_n, t_n]; % crear la matriz de datos de salida a la ANN, invirtiendo filas por columnas

trainIndices = [1:4, 6:9, 11:14, 16:19, 21:24, 26:29, 31:34, 36:39, 41:44, 46:49, ...

51:54, 56:59, 61:64, 66:69, 71:74, 76:79, 81:84, 86:89, 91:94, 96:99, ...

Anexos A. Código de Matlab

```
101:104, 106:109, 111:114, 116:119, 121:124, 126:129, 131:134, 136:139, ...
141:144, 146:149, 151:154, 156:159, 161:164, 166:169, 171:174, 176:179, ...
181:184, 186:189, 191:194, 196:199, 201:204, 206:209, 211:214, 216:219]; % Índices para
entrenamiento
valIndices = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, ...
100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, 170, ...
175, 180, 185, 190, 195, 200, 205, 210, 215, 220]; % Índices para validación
trainInput = X(:, trainIndices); %Selecciona los valores de entrada para entrenamiento
trainTarget = Y(:,trainIndices); %Selecciona los valores de para entrenamiento
valInput = X(:, valIndices); %Selecciona los valores de entrada para validación
valTarget = Y(:, valIndices); %Selecciona los valores de entrada de para validación
```

6.- Definición de la Red neuronal

```
for j = 1:20
    HiddenLayerSize = j; % Definición del número de neuronas en la capa oculta
    net = fitnet(HiddenLayerSize); % Definición de la ANN, indicando el número de neuronas en la capa
oculta
    net.divideParam.trainRatio = 100/100; % Selección del % de los datos de entrada para el
entrenamiento de la ANN
    net.divideParam.valRatio = 0/100; % Selección del % de los datos de entrada para la validación de la
ANN
    net.divideParam.testRatio = 0/100; % Selección del % de los datos de entrada para el testeo de la ANN
    net.trainParam.showWindow = false; % Ocultar la ventana de entrenamiento
    [net,tr]= train(net,trainInput,trainTarget); % Ejecución del entrenamiento de la red neuronal y
almacenamiento de datos en tr para el conjunto de datos preseleccionados

    trainSim = net(trainInput); % Valores simulados con los datos de entrada de entrenamiento
    valSim = sim(net,valInput); % Valores simulados con los datos de entrada de validación

    ALLSim = [trainSim valSim]; % Valores simulados con los datos de entrada de validación ordenados
(training + validation)
    ALLTarget = [trainTarget valTarget]; % Valores experimentales ordenados (training + validation)

    %Cálculo de la raíz del error cuadrático medio RMSE de los valores normalizados

    All_RMSE_Tr(j) = rmse (trainSim,trainTarget,"all"); % Error cuadrático medio en la fase de
entrenamiento Normalizado para todos los datos
```

```
All_RMSE_Val(j) = rmse (valSim,valTarget,"all"); % Error cuadrático medio en la fase de validación
Normalizado para todos los datos
```

```
All_RMSE_ALL(j) = rmse (ALLSim,ALLTarget,"all"); % Error cuadrático medio Normalizado (training +
validation) para todos los datos
```

```
end
```

```
figure; % Abre otra nueva ventana de figura
```

```
plot(1:20, All_RMSE_Tr); % Grafica los datos
```

```
hold on;
```

```
grid on;
```

```
xticks(1:1:20); % Establece las marcas del eje X de 1 en 1
```

```
% Añadir otra línea
```

```
plot(1:20, All_RMSE_Val, 'r'); % Añade otra serie de datos
```

```
% Añadir la leyenda
```

```
legend('Training RMSE', 'Validation RMSE', 'Location', 'best');
```

```
% Añadir etiquetas y título
```

```
xlabel('Neurons', 'FontAngle', 'italic'); % Etiqueta del eje X
```

```
ylabel('RMSE', 'FontAngle', 'italic'); % Etiqueta del eje Y
```

```
title('Training and Validation RMSE over Neurons'); % Título del gráfico2
```

```
hold off; % No más elementos se añadirán al gráfico
```

8.- Gráficas de regresión

```
% Visualización de las gráficas de regresión para la fase de entrenamiento
```

```
trainSim = sim(net,trainInput); % Valores simulados con los datos de entrada de entrenamiento
```

```
valSim = sim(net, valInput); % Valores simulados con los datos de entrada de validación
```

```
ALLSim = [trainSim valSim]; % Valores simulados con los datos de entrada de validación ordenados
(training + validation)
```

```
ALLTarget = [trainTarget valTarget]; % Valores experimentales ordenados (training + validation)
```

```
% Training
```

Anexos A. Código de Matlab

```
figure;  
plotScatterWithRegression('Training [hp, hv, S, b, t]', trainTarget, trainSim, 'b-');
```

% Validation

```
figure;  
plotScatterWithRegression('Validation [hp, hv, S, b, t]', valTarget, valSim, 'g-');
```

% ALL

```
figure;  
plotScatterWithRegression('All [hp, hv, S, b, t]', ALLTarget, ALLSim, 'r-');
```

% Definición vector parámetros

```
params = {'hp', 'hv', 'S', 'b', 't'};
```

% Iteración para la generación de las gráficas de regresión y parámetros simulados, valores individuales

```
for i = 1:numel(params)
```

% Almacenar los valores simulados y experimentales para el parámetro

```
eval([params{i} '_TrainSim = trainSim(i,:);'])
```

```
eval([params{i} '_TrainTarget = trainTarget(i,:);'])
```

```
eval([params{i} '_ValSim = valSim(i,:);'])
```

```
eval([params{i} '_ValTarget = valTarget(i,:);'])
```

```
eval([params{i} '_ALLSim = ALLSim(i,:);'])
```

```
eval([params{i} '_ALLTarget = ALLTarget(i,:);']);
```

%Training

```
figure;
```

```
plotScatterWithRegression(['Training [' params{i} ']'], eval([params{i} '_TrainTarget']), eval([params{i} '_TrainSim']), 'b-');
```

% Validation

```
figure;
```

```
plotScatterWithRegression(['Validation [' params{i} ']', eval([params{i} '_ValTarget']), eval([params{i} '_ValSim']), 'g-');
```

```
% ALL
```

```
figure;
```

```
plotScatterWithRegression(['All [' params{i} ']', eval([params{i} '_ALLTarget']), eval([params{i} '_ALLSim']), 'r-');
```

```
end
```

```
% Definición de las predicciones y objetivos para cada parámetro normalizado
```

```
trainSims = {hp_TrainSim, hv_TrainSim, S_TrainSim, b_TrainSim, t_TrainSim};
```

```
trainTargets = {hp_TrainTarget, hv_TrainTarget, S_TrainTarget, b_TrainTarget, t_TrainTarget};
```

```
valSims = {hp_ValSim, hv_ValSim, S_ValSim, b_ValSim, t_ValSim};
```

```
valTargets = {hp_ValTarget, hv_ValTarget, S_ValTarget, b_ValTarget, t_ValTarget};
```

```
ALLSims = {hp_ALLSim, hv_ALLSim, S_ALLSim, b_ALLSim, t_ALLSim};
```

```
ALLTargets = {hp_ALLTarget, hv_ALLTarget, S_ALLTarget, b_ALLTarget, t_ALLTarget};
```

9.- Histograma de errores

```
figure;
```

```
ploterrhist(gsubtract(trainTarget,trainSim),"Training",gsubtract(valTarget,valSim), "Validation")
```

```
% Cambia el título del histograma
```

```
titulo = 'Error Histogram [hp, hv, S, b, t]';
```

```
title(titulo);
```

```
figure;
```

```
ploterrhist(gsubtract(hp_TrainTarget,hp_TrainSim),"Training",gsubtract(hp_ValTarget,hp_ValSim), "Validation")
```

```
titulo = 'Error Histogram [hp]';
```

```
title(titulo);
```

```
ylim([0 100]);
```

```
yticks(0:10:100);
```

```
figure;
```

```
ploterrhist(gsubtract(hv_TrainTarget,hv_TrainSim),"Training",gsubtract(hv_ValTarget,hv_ValSim), "Validation")
```

Anexos A. Código de Matlab

```
titulo = 'Error Histogram [hv]';
title(titulo);
ylim([0 100]);
yticks(0:10:100);

figure;
ploterrhist(gsubtract(S_TrainTarget,S_TrainSim),"Training",gsubtract(S_ValTarget,S_ValSim),
"Validation")
titulo = 'Error Histogram [S]';
title(titulo);
ylim([0 100]);
yticks(0:10:100);

figure;
ploterrhist(gsubtract(b_TrainTarget,b_TrainSim),"Training",gsubtract(b_ValTarget,b_ValSim),
"Validation")
titulo = 'Error Histogram [b]';
title(titulo);
ylim([0 100]);
yticks(0:10:100);

figure;
ploterrhist(gsubtract(t_TrainTarget,t_TrainSim),"Training",gsubtract(t_ValTarget,t_ValSim), "Validation")
titulo = 'Error Histogram [t]';
title(titulo);
ylim([0 100]);
yticks(0:10:100);
```

10.- Cálculo de RMSE, R2 y R2adj de los valores normalizados

%Cálculo de la raíz del error cuadrático medio RMSE de los valores normalizados

All_RMSE_Tr = rmse (trainSim,trainTarget,"all"); % Error cuadrático medio en la fase de entrenamiento
Normalizado para todos los datos

All_RMSE_Val = rmse (valSim,valTarget,"all"); % Error cuadrático medio en la fase de validación
Normalizado para todos los datos

```
All_RMSE_ALL = rmse (ALLSim,ALLTarget,"all"); % Error cuadrático medio Normalizado (training + validation) para todos los datos
```

```
%Cálculo del R2 y R2 ajustado para el conjunto de datos
```

```
All_R2_Tr = 1 - sum((trainTarget - trainSim).^2)/sum((trainTarget - mean(trainTarget')).^2);
```

```
All_R2_Val = 1 - sum((valTarget - valSim).^2)/sum((valTarget - mean(valTarget')).^2);
```

```
All_R2_ALL = 1 - sum((ALLTarget - ALLSim).^2)/sum((ALLTarget - mean(ALLTarget')).^2);
```

```
ntr=numel(trainTarget); % número de datos totales utilizados en simulación
```

```
nval=numel(valTarget); % número de datos totales utilizados en entrenamiento
```

```
ntotal=numel(ALLSim); % número de datos totales utilizados
```

```
nvariables=2; % número de variables de entrada independientes
```

```
All_R2Adj_Tr = 1-(1-All_R2_Tr)*(ntr-1)/(ntr-nvariables-1);
```

```
All_R2Adj_Val = 1-(1-All_R2_Val)*(nval-1)/(nval-nvariables-1);
```

```
All_R2Adj_ALL = 1-(1-All_R2_ALL)*(ntotal-1)/(ntotal-nvariables-1);
```

```
% Itera sobre cada parámetro normalizado
```

```
for i = 1:numel(params)
```

```
    % Calcula métricas para el parámetro actual
```

```
    [RMSE, R2, R2Adj] = calculateParameter(trainSims{i}, trainTargets{i}, valSims{i}, valTargets{i}, ALLSims{i}, ALLTargets{i});
```

```
    % Almacena los resultados en variables con el formato deseado
```

```
    eval([params{i}, '_RMSE_Tr = RMSE(1);']);
```

```
    eval([params{i}, '_RMSE_Val = RMSE(2);']);
```

```
    eval([params{i}, '_RMSE_ALL = RMSE(3);']);
```

```
    eval([params{i}, '_R2_Tr = R2(1);']);
```

```
    eval([params{i}, '_R2_Val = R2(2);']);
```

```
    eval([params{i}, '_R2_ALL = R2(3);']);
```

```
    % Calcula R^2 ajustado para los datos normalizados
```

```
    eval([params{i}, '_R2Adj_Tr = R2Adj(1);']);
```

```
    eval([params{i}, '_R2Adj_Val = R2Adj(2);']);
```

Anexos A. Código de Matlab

```
eval([params{i}, '_R2Adj_ALL = R2Adj(3);']);  
end
```

11.- Tabla de datos de rendimiento de la red, parámetros normalizados

RMSE y R2

% Definir nombres de archivos y hojas

```
archivoExcel = 'resultados_red_neuronal.xlsx';
```

```
hojaRMSE = 'RMSE';
```

```
hojaR2Ajustado = 'R2_Ajustado';
```

% Sección RMSE

```
RMSE_Tr = [All_RMSE_Tr, hp_RMSE_Tr, hv_RMSE_Tr, S_RMSE_Tr, b_RMSE_Tr, t_RMSE_Tr];
```

```
RMSE_Val = [All_RMSE_Val, hp_RMSE_Val, hv_RMSE_Val, S_RMSE_Val, b_RMSE_Val, t_RMSE_Val];
```

```
RMSE_ALL = [All_RMSE_ALL, hp_RMSE_ALL, hv_RMSE_ALL, S_RMSE_ALL, b_RMSE_ALL, t_RMSE_ALL];
```

% Crear tabla

```
datosTablaRMSE = table(RMSE_Tr, RMSE_Val, RMSE_ALL, 'VariableNames', {'Training', 'Validation',  
'ALL'}, 'RowNames', {'[hp, hv, S, b, t]', '[hp]', '[hv]', '[S]', '[b]', '[t]'});
```

% Mostrar la tabla

```
disp(datosTablaRMSE);
```

% Crear la tabla utilizando uitable

```
f = figure;
```

```
posicionTabla = [10, 200, 520, 175];
```

```
t = uitable(f, 'Data', datosTablaRMSE{:, :}, 'ColumnName', datosTablaRMSE.Properties.VariableNames,  
'Position', posicionTabla, 'rowname', {'[hp, hv, S, b, t]', '[hp]', '[hv]', '[S]', '[b]', '[t]'});
```

```
titulo = 'RMSE';
```

```
annotation('textbox', [0, 0.9, 0.2, 0.1], 'String', titulo, 'EdgeColor', 'none', 'HorizontalAlignment', 'center',  
'FontSize', 12);
```

% Guardar tabla RMSE en Excel

```
writetable(datosTablaRMSE, archivoExcel, 'Sheet', hojaRMSE, 'WriteRowNames', true);
```

% Sección R² Ajustado

```
R2Adj_Tr = [All_R2Adj_Tr, hp_R2Adj_Tr, hv_R2Adj_Tr, S_R2Adj_Tr, b_R2Adj_Tr, t_R2Adj_Tr];
```

```
R2Adj_Val = [All_R2Adj_Val, hp_R2Adj_Val, hv_R2Adj_Val, S_R2Adj_Val, b_R2Adj_Val, t_R2Adj_Val];
```

```
R2Adj_ALL = [All_R2Adj_ALL, hp_R2Adj_ALL, hv_R2Adj_ALL, S_R2Adj_ALL, b_R2Adj_ALL, t_R2Adj_ALL];
```

```
% Crear tabla
```

```
datosTablaR2Adj = table(R2Adj_Tr', R2Adj_Val', R2Adj_ALL', 'VariableNames', {'Training', 'Validation', 'ALL'}, 'RowNames', {'[hp, hv, S, b, t]', '[hp]', '[hv]', '[S]', '[b]', '[t]'});
```

```
% Mostrar la tabla
```

```
disp(datosTablaR2Adj);
```

```
% Crear la tabla utilizando uitable
```

```
f = figure;
```

```
posicionTabla = [10, 200, 520, 175];
```

```
t = uitable(f, 'Data', datosTablaR2Adj{:, :}, 'ColumnName', datosTablaR2Adj.Properties.VariableNames, 'Position', posicionTabla, 'rowname', {'[hp, hv, S, b, t]', '[hp]', '[hv]', '[S]', '[b]', '[t]'});
```

```
titulo = 'R2 Ajustado';
```

```
annotation('textbox', [0, 0.9, 0.2, 0.1], 'String', titulo, 'EdgeColor', 'none', 'HorizontalAlignment', 'center', 'FontSize', 12);
```

```
% Guardar tabla R2 Ajustado en Excel
```

```
writetable(datosTablaR2Adj, archivoExcel, 'Sheet', hojaR2Ajustado, 'WriteRowNames', true);
```

12.- Gráficos 3D comparativos

```
% Importar Datos desde Excel para cada parámetro A97075
```

```
for p = 1:length(params)
```

```
    paramName = params{p};
```

```
    % Importar Datos desde Excel
```

```
    data = readmatrix("A97075_graficas_comparativas.xlsx", 'Sheet', paramName);
```

```
    % Si los datos están bien separados y limpios, puedes convertir directamente
```

```
    Vc = data(:,1);
```

```
    f = data(:,2);
```

```
    Experimental = data(:,3);
```

```
    Potencial = data(:,4);
```

```
    Polinomico = data(:,5);
```

```
    Red_neuronal = data(:,6);
```

Anexos A. Código de Matlab

```
% Crear gráfico 3D
figure;
hold on; % Mantiene la gráfica actual para superponer nuevos gráficos
grid on; % Activa la cuadrícula
title(['Model Comparison for [' paramName '] in A97075']);

view(3); % Ajusta la vista a 3D

% Configuración de las líneas de la cuadrícula
ax = gca;
ax.GridLineStyle = '--'; % Estilo de línea para la cuadrícula principal

% Definir marcadores y colores
markers = {'o', '^', 'x', 'd'}; % Círculo, triángulo hacia arriba, cruz diagonal, diamante
colors = {'g', 'k', 'b', 'r'}; % Verde, negro, azul, rojo
groups = {'Experimental', 'Potential', 'Polynomial', 'ANN'};
dataSets = {Experimental, Potencial, Polinómico, Red_neuronal};

% Tamaño de los marcadores
markerSize = 30; % Ajustar este valor para cambiar el tamaño de los marcadores

% Crear handles para la leyenda
legendHandles = gobjects(1, length(dataSets));

% Iterar a través de cada grupo y dibujar puntos y líneas
for i = 1:length(dataSets)
    % Configurar el color de relleno (sin relleno para el triángulo)
    if strcmp(markers{i}, '^')
        scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', 'none', 'DisplayName', groups{i});
        % Configurar el grosor del borde de los marcadores
        set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario
        % Guardar los handles para la leyenda
        legendHandles(i) = scatter3Handle;
    else
```

```

scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', colors{i}, 'DisplayName', groups{i});

% Configurar el grosor del borde de los marcadores
set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario

% Guardar los handles para la leyenda
legendHandles(i) = scatter3Handle;

end

end

% Conectar puntos entre los diferentes grupos secuencialmente
for j = 1:length(Vc)
    for k = 1:(length(dataSets) - 1)
        plot3([Vc(j) Vc(j)], [f(j) f(j)], [dataSets{k}(j) dataSets{k+1}(j)], ':', 'LineWidth', 0.5, 'Color', 'k');
    end
end

% Configurar ticks de los ejes X e Y
ax.XTick = unique(Vc);
ax.YTick = unique(f);

% Añadir etiquetas y título
xlabel('V_c (m/min)', 'FontAngle', 'italic');
ylabel('f (mm/rev)', 'FontAngle', 'italic');
if paramName == 'S'
    zlabel(strcat(paramName, ' (mm^{2})'), 'FontAngle', 'italic');
else
    zlabel(strcat(paramName, ' (mm)'), 'FontAngle', 'italic');
end

% Añadir leyenda sin las líneas discontinuas
legend(legendHandles, 'Location', 'northeastoutside');

hold off; % Liberar el gráfico para futuras operaciones
end

```

Anexos A. Código de Matlab

```
% Importar Datos desde Excel para cada parámetro A92024
for p = 1:length(params)
    paramName = params{p};

    % Importar Datos desde Excel
    data = readmatrix("A92024_graficas_comparativas.xlsx", 'Sheet', paramName);

    % Si los datos están bien separados y limpios, puedes convertir directamente
    Vc = data(:,1);
    f = data(:,2);
    Experimental = data(:,3);
    Potencial = data(:,4);
    Polinomico = data(:,5);
    Red_neuronal = data(:,6);

    % Crear gráfico 3D
    figure;
    hold on; % Mantiene la gráfica actual para superponer nuevos gráficos
    grid on; % Activa la cuadrícula
    title(['Model Comparison for [' paramName '] in A92024']);

    view(3); % Ajusta la vista a 3D

    % Configuración de las líneas de la cuadrícula
    ax = gca;
    ax.GridLineStyle = '--'; % Estilo de línea para la cuadrícula principal

    % Definir marcadores y colores
    markers = {'o', '^', 'x', 'd'}; % Círculo, triángulo hacia arriba, cruz diagonal, diamante
    colors = {'g', 'k', 'b', 'r'}; % Verde, negro, azul, rojo
    groups = {'Experimental', 'Potential', 'Polynomial', 'ANN'};
    dataSets = {Experimental, Potencial, Polinomico, Red_neuronal};

    % Tamaño de los marcadores
```

```

markerSize = 30; % Ajustar este valor para cambiar el tamaño de los marcadores

% Crear handles para la leyenda
legendHandles = gobjects(1, length(dataSets));

% Iterar a través de cada grupo y dibujar puntos y líneas
for i = 1:length(dataSets)
    % Configurar el color de relleno (sin relleno para el triángulo)
    if strcmp(markers{i}, '^')
        scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', 'none', 'DisplayName', groups{i});
        % Configurar el grosor del borde de los marcadores
        set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario
        % Guardar los handles para la leyenda
        legendHandles(i) = scatter3Handle;
    else
        scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', colors{i}, 'DisplayName', groups{i});
        % Configurar el grosor del borde de los marcadores
        set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario
        % Guardar los handles para la leyenda
        legendHandles(i) = scatter3Handle;
    end
end

% Conectar puntos entre los diferentes grupos secuencialmente
for j = 1:length(Vc)
    for k = 1:(length(dataSets) - 1)
        plot3([Vc(j) Vc(j)], [f(j) f(j)], [dataSets{k}(j) dataSets{k+1}(j)], ':', 'LineWidth', 0.5, 'Color', 'k');
    end
end

% Configurar ticks de los ejes X e Y
ax.XTick = unique(Vc);

```

Anexos A. Código de Matlab

```
ax.YTick = unique(f);

% Añadir etiquetas y título
xlabel('V_c (m/min)', 'FontAngle', 'italic');
ylabel('f (mm/rev)', 'FontAngle', 'italic');
if paramName == 'S'
    xlabel(strcat(paramName, ' (mm^{2})'), 'FontAngle', 'italic');
else
    xlabel(strcat(paramName, ' (mm)'), 'FontAngle', 'italic');
end

% Añadir leyenda sin las líneas discontinuas
legend(legendHandles, 'Location', 'northeastoutside');

hold off; % Liberar el gráfico para futuras operaciones
end

% Importar Datos desde Excel para cada parámetro
for p = 1:length(params)
    paramName = params{p};

    % Importar Datos desde Excel
    data = readmatrix("Ti6Al4V_graficas_comparativas.xlsx", 'Sheet', paramName);

    % Si los datos están bien separados y limpios, puedes convertir directamente
    Vc = data(:,1);
    f = data(:,2);
    Experimental = data(:,3);
    Potencial = data(:,4);
    Polinomico = data(:,5);
    Red_neuronal = data(:,6);

    % Crear gráfico 3D
    figure;
    hold on; % Mantiene la gráfica actual para superponer nuevos gráficos
    grid on; % Activa la cuadrícula
```

```

title(['Model Comparison for [' paramName '] in Ti6Al4V']);

view(3); % Ajusta la vista a 3D

% Configuración de las líneas de la cuadrícula
ax = gca;
ax.GridLineStyle = '--'; % Estilo de línea para la cuadrícula principal

% Definir marcadores y colores
markers = {'o', '^', 'x', 'd'}; % Círculo, triángulo hacia arriba, cruz diagonal, diamante
colors = {'g', 'k', 'b', 'r'}; % Verde, negro, azul, rojo
groups = {'Experimental', 'Potential', 'Polynomial', 'ANN'};
dataSets = {Experimental, Potencial, Polinomico, Red_neuronal};

% Tamaño de los marcadores
markerSize = 30; % Ajustar este valor para cambiar el tamaño de los marcadores

% Crear handles para la leyenda
legendHandles = gobjects(1, length(dataSets));

% Iterar a través de cada grupo y dibujar puntos y líneas
for i = 1:length(dataSets)
    % Configurar el color de relleno (sin relleno para el triángulo)
    if strcmp(markers{i}, '^')
        scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', 'none', 'DisplayName', groups{i});
        % Configurar el grosor del borde de los marcadores
        set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario
        % Guardar los handles para la leyenda
        legendHandles(i) = scatter3Handle;
    else
        scatter3Handle = scatter3(Vc, f, dataSets{i}, markerSize, 'Marker', markers{i},
'MarkerEdgeColor', colors{i}, 'MarkerFaceColor', colors{i}, 'DisplayName', groups{i});
        % Configurar el grosor del borde de los marcadores
        set(scatter3Handle, 'LineWidth', 1); % Ajusta este valor según sea necesario
        % Guardar los handles para la leyenda

```

Anexos A. Código de Matlab

```
        legendHandles(i) = scatter3Handle;
    end
end
% Conectar puntos entre los diferentes grupos secuencialmente
for j = 1:length(Vc)
    for k = 1:(length(dataSets) - 1)
        plot3([Vc(j) Vc(j)], [f(j) f(j)], [dataSets{k}(j) dataSets{k+1}(j)], ':', 'LineWidth', 0.5, 'Color', 'k');
    end
end

% Configurar ticks de los ejes X e Y
ax.XTick = unique(Vc);
ax.YTick = unique(f);

% Añadir etiquetas y título
xlabel('V_c (m/min)', 'FontAngle', 'italic');
ylabel('f (mm/rev)', 'FontAngle', 'italic');
if paramName == 'S'
    xlabel(strcat(paramName, ' (mm^{2})'), 'FontAngle', 'italic');
else
    xlabel(strcat(paramName, ' (mm)'), 'FontAngle', 'italic');
end

% Añadir leyenda sin las líneas discontinuas
legend(legendHandles, 'Location', 'northeastoutside');

hold off; % Liberar el gráfico para futuras operaciones
end
```

13.- Análisis ANOVA

% Convertir variables de factores a categóricas y redondear

Vc = categorical(round(x(:,2),3)); % Convertimos y redondeamos la primera columna de x

f = categorical(round(x(:,3),3)); % Convertimos y redondeamos la segunda columna de x

```
% Nombres de las variables de salida
```

```
output_vars = {'hp', 'hv', 'S', 'b', 't'};
```

```
% Realizar ANOVA de dos vías para cada variable de salida
```

```
results = cell(length(output_vars), 1); % Almacenar los resultados de cada ANOVA
```

```
for i = 1:length(output_vars)
```

```
    response = round(y(:,i),3); % Redondear la variable de salida actual
```

```
    % Ejecutar ANOVA de dos vías
```

```
    [p, tbl, stats] = anovan(response, {Vc, f}, 'model', 'interaction', 'varnames', {'Vc', 'f'}, 'display', 'off');
```

```
    % Guardar resultados en un cell array
```

```
    results{i} = struct('Variable', output_vars{i}, 'Table', tbl, 'PValues', p, 'Stats', stats);
```

```
% Opcional: Mostrar tabla de ANOVA en la consola
```

```
    fprintf('\nResultados de ANOVA para %s:\n', output_vars{i});
```

```
    disp(tbl);
```

```
end
```

FUNCIONES

```
function plotScatterWithRegression(titleText, target, sim, color)
```

```
    scatter(target, sim, 'k');
```

```
    title(titleText);
```

```
    xlabel('Normalized Target', 'FontAngle', 'italic');
```

```
    ylabel('Normalized Output', 'FontAngle', 'italic');
```

```
    grid on;
```

```
    axis equal;
```

```
    xlim([0, 1]);
```

```
    ylim([0, 1]);
```

```
    legend('Data', 'Location', 'northwest');
```

```
    hold on;
```

```
    % Añadir la línea de regresión
```

```
    p = polyfit(target, sim, 1);
```

```
    x_fit = linspace(0, 1, 100);
```

Anexos A. Código de Matlab

```
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, color, 'LineWidth', 2, 'DisplayName', 'Fit');
% Añadir la línea de ajuste al 100%
plot([0, 1], [0, 1], 'k--', 'LineWidth', 1.5, 'DisplayName', 'Output=Target');
end

function [RMSE, R2, R2Adj] = calculateParameter(trainSim, trainTarget, valSim, valTarget, ALLSim,
ALLTarget)

% Calcula RMSE (raiz del error cuadrático medio) para la fase de entrenamiento
RMSE_Tr = rmse(trainSim, trainTarget, "all");

% Calcula RMSE (raiz del error cuadrático medio) para la fase de validación
RMSE_Val = rmse(valSim, valTarget, "all");

% Calcula RMSE (raiz del error cuadrático medio) para todos los datos
RMSE_ALL = rmse(ALLSim, ALLTarget, "all");

% Calcula R^2 para la fase de entrenamiento
R2_Tr = 1 - sum((trainTarget - trainSim).^2) / sum((trainTarget - mean(trainTarget')).^2);

% Calcula R^2 para la fase de validación
R2_Val = 1 - sum((valTarget - valSim).^2) / sum((valTarget - mean(valTarget')).^2);

% Calcula R^2 para todos los datos
R2_ALL = 1 - sum((ALLTarget - ALLSim).^2) / sum((ALLTarget - mean(ALLTarget')).^2);

% Número de datos y variables
ntr = numel(trainTarget); % número de datos totales utilizados en simulación
nval = numel(valTarget); % número de datos totales utilizados en entrenamiento
ntotal = numel(ALLSim); % número de datos totales utilizados
nvariables = 2; % número de variables de entrada independientes

% Calcula R^2 ajustado
R2Adj_Tr = 1 - (1 - R2_Tr) * (ntr - 1) / (ntr - nvariables - 1);
R2Adj_Val = 1 - (1 - R2_Val) * (nval - 1) / (nval - nvariables - 1);
```

```
R2Adj_ALL = 1 - (1 - R2_ALL) * (ntotal - 1) / (ntotal - nvariables - 1);
```

```
% Devuelve los resultados
```

```
RMSE = [RMSE_Tr, RMSE_Val, RMSE_ALL];
```

```
R2 = [R2_Tr, R2_Val, R2_ALL];
```

```
R2Adj = [R2Adj_Tr, R2Adj_Val, R2Adj_ALL];
```

```
end
```