

**INSTITUTO TECNOLÓGICO DE CD. JUÁREZ**  
**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



**PREDICCIÓN DEL ESTADO DE OPERACIÓN DE UNA IMPRESORA 3D**  
**MEDIANTE REDES DE SENSORES Y ALGORITMOS**  
**DE APRENDIZAJE PROFUNDO**

**PROTOCOLO DE INVESTIGACIÓN**  
**QUE PRESENTA**

**CARLOS ALBERTO RONQUILLO SALAS**

**Cd. JUÁREZ, CHIHUAHUA**

**OCTUBRE DE 2023**

## **DEDICATORIA**

Dedico este logro a las personas amadas que tengo la dicha de tener a mi lado ...  
... y también a las personas amadas que ya no están físicamente, pero están en mi corazón.

## **AGRADECIMIENTOS**

A mi Comité Tutorial, por su valiosa retroalimentación a lo largo del desarrollo de este proyecto.

A la Jefatura de la División de Estudios de Posgrado e Investigación del Tecnológico Nacional de México campus Instituto Tecnológico de Ciudad Juárez, por facilitar la impresora 3D para este proyecto.

Al Consejo Nacional de Humanidades, Ciencia y Tecnología (Conahcyt) de México, por su apoyo como becario nacional.

A todas las personas que de un modo u otro hicieron posible la consecución de este proyecto.



**C. JORGE ADOLFO PINTO SANTOS**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS**  
**DE POSGRADO E INVESTIGACIÓN**  
**P R E S E N T E.**

Por medio de la presente se hace constar que la tesis denominada "PREDICCIÓN DEL ESTADO DE OPERACIÓN DE UNA IMPRESORA 3D MEDIANTE REDES DE SENSORES Y ALGORITMOS DE APRENDIZAJE PROFUNDO", presentado por el(la) alumno(a) **C. CARLOS ALBERTO RONQUILLO SALAS** con número de control D88111388, para obtener el grado de Doctor(a) en el programa de Doctorado en Ciencias de la Ingeniería, ha sido revisada y aprobada en su forma y contenido por los suscritos, por lo que no existe ningún inconveniente para la impresión de la misma.

Se extiende la presente constancia a petición de él(la) interesado(a) y para los fines legales que a él(ella) convengan, en Ciudad Juárez, Chihuahua, a los tres días del mes de octubre del año dos mil veintitrés.

**A T E N T A M E N T E**  
**"Excelencia en Educación Tecnológica"**

ULISES MARTÍNEZ CONTRERAS  
DIRECTOR

DELIA JULIETA VALLES ROSALES  
CO-DIRECTORA

ARTURO WOCCAY PRIETO  
REVISOR

ADÁN VALLES CHÁVEZ  
REVISOR

HERNÁN DE LA GARZA GUTIÉRREZ  
REVISOR

ROSA MARÍA REYES MARTÍNEZ  
REVISORA

C.c.p. División de Estudios de Posgrado e Investigación  
Alumno (a)





**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Juárez

Ciudad Juárez, Chihuahua, **05/octubre/2023**

Oficio N°: DEPI/049/2023

Asunto: Autorización de Impresión de Tesis

**C. CARLOS ALBERTO RONQUILLO SALAS  
CANDIDATO(A) AL GRADO DE DOCTOR(A) EN  
DOCTORADO EN CIENCIAS DE LA INGENIERÍA  
P R E S E N T E.**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**PREDICCIÓN DEL ESTADO DE OPERACIÓN DE UNA IMPRESORA 3D MEDIANTE REDES DE SENSORES Y ALGORITMOS DE APRENDIZAJE PROFUNDO**" ha informado a esta División de Estudios de Posgrado e Investigación, que está de acuerdo con el trabajo presentado. Por lo anterior se le autoriza se proceda con la **IMPRESIÓN DEFINITIVA DE SU TRABAJO DE TESIS.**

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

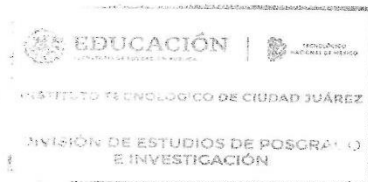
**ATENTAMENTE**

*"Excelencia en Educación Tecnológica"*

**C. JORGE ADOLFO PINTO SANTOS  
JEFE DE LA DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN**

C.c.p. Departamento de Servicios Escolares  
División de Estudios Profesionales

JAPS/dmsp



Av. Tecnológico 1340 Fracc. El Crucero C.P. 32500 Cd. Juárez, Chihuahua, Tel. 01 (656) 688-2500  
e-mail: comunicación\_y\_difusión@cdjuarez.tecnm.mx | tecnm.mx | cdjuarez.tecnm.mx



**2023  
Francisco  
VILLA**



# Instituto Tecnológico de Ciudad Juárez



División de Estudios de Posgrado e Investigación

## CARTA CESIÓN DE DERECHOS

En la Ciudad de Juárez, Chihuahua; siendo el día 16 del mes de octubre del año 2023, el (la) que suscribe **C. Carlos Alberto Ronquillo Salas** alumno(a) del Programa de **Doctorado en Ciencias de la Ingeniería**, con número de control **D88111388**, adscrito(a) a la **División de Estudios de Posgrado e Investigación**, manifiesto(a) que es el (la) autor(a) intelectual del presente trabajo de Tesis bajo la dirección del (de la, de los) **Dr. Ulises Martínez Contreras** y cede los derechos del trabajo titulado **Predicción del Estado de Operación de una Impresora 3D Mediante Redes de Sensores y Algoritmos de Aprendizaje Profundo**, al Instituto Tecnológico de Ciudad Juárez para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del (de la) autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones **carlos.rs@cdjuarez.tecnm.mx, ulises.mc@cdjuarez.tecnm.mx**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

**Carlos Alberto Ronquillo Salas**

---

Nombre y firma del alumno(a)

## RESUMEN

La Industria 4.0 comprende la integración de diversas tecnologías con el propósito de mejorar la eficiencia de los sistemas productivos y donde un elemento clave es el desarrollo de sistemas inteligentes capaces de transformar en conocimiento la gran cantidad de datos generados por controladores de máquinas y sistemas de fabricación. Una tecnología asociada con la Industria 4.0 es la manufactura aditiva o impresión 3D, la cual tiene aplicaciones prácticamente ya en todas las ramas de la industria. No obstante, las impresoras 3D sufren de limitaciones de confiabilidad ya que actualmente la mayoría de estas máquinas no poseen sistemas de monitoreo de su estado de operación y no pueden detectarse desviaciones del proceso que afectan la calidad de las piezas impresas y causan pérdida de recursos. Para abordar este problema, la recopilación de datos mediante redes de sensores y su análisis con algoritmos de aprendizaje profundo pueden ayudar a identificar problemas durante el proceso de impresión. Sin embargo, dada la gran cantidad de datos que generan las redes de sensores y la falta de una metodología estándar para analizar los datos limita su implementación. En este trabajo se propone una metodología para identificar estados de operación anormales de una impresora 3D mediante la detección de anomalías vibratorias usando unidades de medición inercial y entrenando un autocodificador mediante aprendizaje semi-supervisado con tres diferentes enfoques. Se usa la sensibilidad como métrica para evaluar el desempeño del autocodificador en la clasificación de anomalías vibratorias.

## CONTENIDO

LISTA DE TABLAS .....	IX
LISTA DE FIGURAS .....	X
1. INTRODUCCIÓN .....	1
2. PLANTEAMIENTO DEL PROBLEMA .....	2
2.1. Definición del problema .....	2
2.2. Preguntas de Investigación .....	2
2.3. Hipótesis y Variables de Investigación.....	2
2.4. Objetivos .....	3
2.4.1. Objetivo General.....	3
2.4.2. Objetivos Específicos.....	3
2.5. Justificación .....	3
2.6. Delimitaciones .....	4
3. MARCO TEÓRICO.....	5
3.1. Industria 4.0 .....	6
3.2. Manufactura Aditiva .....	6
3.3. Modelado por Deposición Fundida.....	8
3.4. Monitoreo de condiciones de operación .....	10
3.5. Aprendizaje automático .....	11
3.5.1.1. Los diferentes tipos de aprendizaje automático .....	12
3.5.1.2. Aprendizaje profundo .....	15
3.5.1.3. Redes neuronales artificiales.....	19
3.5.1.4. Autocodificadores .....	21
4. METODOLOGÍA .....	25
4.1. Materiales y métodos .....	25
4.1.1. Selección de componentes .....	25
4.2. Metodología .....	27
4.2.1. Detección de anomalías usando autocodificadores.....	28

4.2.2.	Instrumentación de la impresora.....	29
4.2.3.	Implementación del bus I <sup>2</sup> C.....	29
4.3.	Desarrollo.....	31
4.3.1.	Colecta de los datos .....	31
4.3.2.	Arquitectura del autocodificador .....	33
4.3.3.	Entrenamiento del autocodificador .....	33
4.3.3.1.	Entrenamiento con datos sin preprocesar .....	34
4.3.3.2.	Entrenamiento con datos preprocesados con VAP .....	36
4.3.3.3.	Entrenamiento con datos preprocesados con RMS.....	39
4.3.4.	Comparación del entrenamiento del autocodificador con los diferentes enfoques	41
5.	RESULTADOS .....	43
5.1.	Predicción de anomalías vibratorias .....	43
5.1.1.	Predicciones con datos sin preprocesar.....	45
5.1.2.	Predicciones con datos VAP.....	46
5.1.3.	Predicciones con datos RMS .....	48
5.2.	Comparación de las predicciones de los diferentes enfoques.....	50
6.	DISCUSIÓN Y CONCLUSIONES .....	52
6.1.	Discusión.....	52
6.2.	Conclusiones .....	52
7.	RECOMENDACIONES .....	55
8.	REFERENCIAS.....	56



## LISTA DE TABLAS

<b>Tabla 1.</b> <i>Topología del autocodificador</i> .....	33
<b>Tabla 2.</b> <i>RECM de los diferentes conjuntos con datos sin preprocesar.</i> .....	36
<b>Tabla 3.</b> <i>RECM de los diferentes conjuntos con datos VAP.</i> .....	39
<b>Tabla 4.</b> <i>RECM de los diferentes conjuntos con datos RMS</i> .....	40
<b>Tabla 5.</b> <i>Comparación de RECM normalizado sobre XTestA y las épocas.</i> .....	42
<b>Tabla 6.</b> <i>Matriz de confusión de las predicciones con datos sin preprocesar.</i> .....	45
<b>Tabla 7.</b> <i>Matriz de confusión de las predicciones con datos VAP.</i> .....	47
<b>Tabla 8.</b> <i>Matriz de confusión de las predicciones con datos RMS.</i> .....	49
<b>Tabla 9.</b> <i>Precisión y sensibilidad de los diferentes enfoques</i> .....	50

## LISTA DE FIGURAS

<b>Figura 3.1.</b> <i>Proceso de manufactura aditiva.....</i>	7
<b>Figura 3.2.</b> <i>Impresora 3D FDM Makerbot Replicator 5ª Generación™.....</i>	8
<b>Figura 3.3.</b> <i>Principales estados de operación y su relación con la calidad del producto en FDM.....</i>	9
<b>Figura 3.4.</b> <i>Los tres tipos de aprendizaje supervisado.....</i>	13
<b>Figura 3.5.</b> <i>Aprendizaje supervisado.....</i>	13
<b>Figura 3.6.</b> <i>Aprendizaje por refuerzo.....</i>	14
<b>Figura 3.7.</b> <i>Clustering o agrupamiento de datos en base a las características <math>x_1</math> y <math>x_2</math>.....</i>	15
<b>Figura 3.8.</b> <i>El aprendizaje profundo en la taxonomía de la IA.....</i>	16
<b>Figura 3.9.</b> <i>RNA simple (monocapa) vs. RNA profunda (multicapa).....</i>	18
<b>Figura 3.10.</b> <i>Modelo matemático simple de una neurona.....</i>	20
<b>Figura 3.11.</b> <i>Función de activación.....</i>	20
<b>Figura 3.12.</b> <i>Red neuronal tipo autocodificador.....</i>	22
<b>Figura 4.1.</b> <i>Unidad de medición inercial MPU6050.....</i>	25
<b>Figura 4.2.</b> <i>Estructura del bus I<sup>2</sup>C.....</i>	26
<b>Figura 4.3.</b> <i>Instrumentación de la impresora.....</i>	29
<b>Figura 4.4.</b> <i>Esquema de conexión del bus I<sup>2</sup>C.....</i>	30
<b>Figura 4.5.</b> <i>Aceleraciones X, Y, Z, condición normal.....</i>	32
<b>Figura 4.6.</b> <i>Velocidades angulares @ X, @Y, @Z, condición normal.....</i>	32
<b>Figura 4.7.</b> <i>Entrenamiento del modelo con datos sin preprocesar.....</i>	35
<b>Figura 4.8.</b> <i>Distribución del error cuadrático de las predicciones sobre XTrainN con datos sin preprocesar.....</i>	36
<b>Figura 4.9.</b> <i>Aceleraciones de la cama de impresión, condición normal.....</i>	37
<b>Figura 4.10.</b> <i>Entrenamiento del modelo con datos VAP.....</i>	38
<b>Figura 4.11.</b> <i>Distribución del error cuadrático de las predicciones sobre XTrainN con datos VAP.....</i>	38
<b>Figura 4.12.</b> <i>Aceleraciones de la cama de impresión, condición normal.....</i>	39
<b>Figura 4.13.</b> <i>Distribución del error cuadrático de las predicciones sobre XTrainN con datos RMS.....</i>	40
<b>Figura 4.14.</b> <i>Entrenamiento del modelo con datos RMS.....</i>	41
<b>Figura 5.1.</b> <i>Matriz de confusión del proyecto.....</i>	44
<b>Figura 5.2.</b> <i>Umbral para las predicciones sobre XTrainN con datos sin preprocesar..</i>	45
<b>Figura 5.3.</b> <i>Precisión y sensibilidad para diferentes umbrales con datos sin preprocesar.....</i>	46
<b>Figura 5.4.</b> <i>Umbral para las predicciones sobre XTrainN con VAP.....</i>	47

<b>Figura 5.5.</b> <i>Precisión y sensibilidad para diferentes umbrales con VAP.</i> .....	48
<b>Figura 5.6.</b> <i>Umbral para las predicciones sobre XTrainN con RMS.</i> .....	49
<b>Figura 5.7.</b> <i>Precisión y sensibilidad para diferentes umbrales con RMS.</i> .....	50

# 1. INTRODUCCIÓN

La manufactura aditiva (MA) o impresión 3D—una de las tecnologías clave de la Industria 4.0 (I4.0)—consiste en producir un objeto tridimensional por adición de capas de material adjuntas una sección transversal a la vez a partir de un modelo digital. Los procesos de MA actuales pueden utilizar una de varias tecnologías siendo una de las más populares la de fabricación de filamento fundido (FFF). La tecnología FFF hace uso de una diversidad de materiales; incluidos metales, termoplásticos, cerámica y vidrio (A. M. Crowell et al., 2018). En la actualidad la MA tiene aplicaciones prácticamente en todas las ramas de la industria— automotriz, aeroespacial, salud, etc. Por otra parte, la literatura reporta que los procesos de MA están limitados por la falta de garantía de calidad e inconsistencias entre impresiones debido a limitaciones de confiabilidad durante el proceso de impresión (Wu et al., 2016), ya que éste puede sufrir desviaciones durante la fabricación de la pieza (Huang et al., 2015). Para ayudar a corregir estos problemas, A. Crowell et al. (2018) sugiere que la recopilación de datos con redes de sensores durante el proceso de impresión y su análisis puede proporcionar información de cómo controlar el proceso. Sin embargo, las impresoras 3D actuales poseen sistemas de monitoreo y control poco sofisticados que no permiten monitorear adecuadamente su condición de operación para garantizar su desempeño, lo que ocasiona pérdida de recursos (mala calidad, material, tiempo, energía, etc.). Adicionalmente, la gran cantidad de datos que pueden generar las redes de sensores que pudieran implementarse y la falta de una metodología estándar para analizar dichos datos limita la implementación de sistemas de monitoreo y control más sofisticados en los sistemas de MA. El objetivo de este trabajo de investigación es por tanto, mediante el uso de algoritmos de aprendizaje profundo encontrar las correlaciones e interacciones entre las diversas variables de los datos de adquiridos con las unidades de medición inercial (IMU) para identificar patrones en dichos datos de alta dimensionalidad que permitan determinar la condición de operación de la impresora.

## **2. PLANTEAMIENTO DEL PROBLEMA**

### **2.1. Definición del problema**

Las impresoras 3D actuales poseen sistemas de control poco sofisticados que no permiten monitorear adecuadamente su condición de operación para garantizar su desempeño, lo que ocasiona pérdida de recursos (mala calidad, material, tiempo, energía, etc.).

### **2.2. Preguntas de Investigación**

- De qué manera los algoritmos de aprendizaje profundo permitirán capturar las correlaciones e interacciones entre las diversas variables de los datos medidos con las redes de sensores instalados en una impresora 3D e identificar patrones en los mismos.
- De qué manera la identificación de patrones permitirá determinar la condición de operación de una impresora 3D.

### **2.3. Hipótesis y Variables de Investigación**

- Con el uso de redes de sensores no intrusivos y algoritmos de aprendizaje profundo será posible determinar la condición de operación de una impresora 3D.
- Los algoritmos de aprendizaje profundo permitirán capturar las correlaciones e interacciones entre las variables de los datos medidos con las redes de sensores instalados en una impresora 3D e identificar patrones en los mismos.
- Los patrones identificados permitirán determinar la condición de operación de una impresora 3D.

## **2.4. Objetivos**

### **2.4.1. Objetivo General**

Predecir la condición de operación de una impresora 3D.

### **2.4.2. Objetivos Específicos**

- Instrumentar una impresora 3D con redes de sensores no intrusivos para la adquisición de datos durante el proceso de impresión.
- Entrenar un autocodificador de arquitectura profunda para predecir la condición de operación de una impresora 3D.

## **2.5. Justificación**

Los procesos de MA están limitados por la falta de garantía de calidad e inconsistencias entre impresiones debido a limitaciones de confiabilidad durante el proceso de impresión (Wu et al., 2016), ya que éste puede sufrir desviaciones durante la fabricación de la pieza (Huang et al., 2015). Para ayudar a corregir estos problemas, A. Crowell et al. (2018) sugiere que la recopilación de datos con redes de sensores durante el proceso de impresión y su análisis puede proporcionar información de cómo controlar el proceso. Sin embargo, las impresoras 3D actuales poseen sistemas de monitoreo y control poco sofisticados que no permiten monitorear adecuadamente su condición de operación para garantizar su desempeño, lo que ocasiona pérdida de recursos (mala calidad, material, tiempo, energía, etc.). Adicionalmente, la gran cantidad de datos que pueden generar las redes de sensores que pudieran implementarse y la falta de una metodología estándar para analizar dichos datos limita la implementación de sistemas de monitoreo y control más sofisticados en los sistemas de MA. El objetivo de este trabajo de investigación es por tanto, mediante el uso de algoritmos de aprendizaje profundo encontrar las correlaciones e

interacciones entre las diversas variables de los datos de adquiridos con las unidades de medición inercial (IMU) para identificar patrones en dichos datos de alta dimensionalidad que permitan determinar la condición de operación de la impresora

## **2.6. Delimitaciones**

- a) Este trabajo se enfoca en identificar el estado normal de operación de la impresora y estados de operación anormales relacionados con fallas mecánicas del extrusor y la plataforma.
- b) Se utiliza una impresora 3D comercial marca Makerbot, modelo Replicator 5<sup>a</sup> generación.
- c) Se utiliza como material de impresión filamento de PETG.
- d) Se utiliza el software de programación Python con las librerías TensorFlow, Keras y Sklearn (local y/o en la nube).
- e) Se utilizan redes neuronales de aprendizaje profundo.

### 3. MARCO TEÓRICO

La manufactura aditiva (MA) consiste en fabricar objetos tridimensionales a partir de un diseño digital, agregando capas de material de forma sucesiva. Dependiendo del método de MA, se pueden emplear distintas tecnologías y materiales tales como metales, termoplásticos, cerámicos y vidrio. Según Crowell et al. (2018), esta técnica permite la elaboración de artículos complejos en un periodo de tiempo más reducido en comparación con los métodos convencionales. Además, destacan que su uso se ha extendido a casi todos los sectores industriales, como el automotor, aeroespacial y médico, entre otros.

Sin embargo, de acuerdo con Huang (2015) y Wu (2016), los procesos actuales de fabricación de filamento fundido (FFF) o modelado por deposición fundida (FDM) están limitados por la falta de garantía de calidad e inconsistencias entre las impresiones debido a limitaciones de confiabilidad durante el proceso de impresión, ya que algunos parámetros de la impresora pueden sufrir desviaciones durante el proceso de fabricación.

Wu et al. (2017) menciona que aunque el monitoreo de condiciones del estado de una máquina es considerado como una importante estrategia de diagnóstico y mantenimiento para asegurar la calidad del producto, actualmente la mayoría de las máquinas de MA no están equipadas con sistemas de monitoreo y control de lazo cerrado, y sin la detección en tiempo real, los fallos del proceso o averías de la máquina se producen sin previo aviso durante el proceso de fabricación y la calidad y la repetitividad de los productos finales no se puede garantizar.

Para ayudar a corregir estos problemas, Crowell et al. (2018) propone que la recopilación de datos en tiempo real durante el proceso de impresión y su análisis puede proporcionar información de cómo controlar el proceso. Sin embargo, dada la gran cantidad de datos que pueden generar las redes de sensores y la falta de una metodología estándar para analizar dicha información limita dicha implementación.



### **3.1. Industria 4.0**

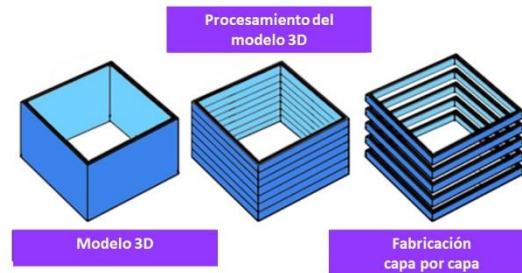
Según Ahuett-Garza y Kurfess (2018), la Industria 4.0 (I4.0), también conocida como manufactura inteligente, comprende la incorporación de varias tecnologías buscando mejorar la eficiencia y la capacidad de respuesta de un sistema productivo. Mencionan asimismo que la manufactura inteligente está estrechamente relacionada a este concepto y podría considerarse como la implementación de la I4.0 en aplicaciones tales como la trazabilidad de los productos para control de la calidad, el monitoreo del estado de operación de las máquinas y la manufactura distribuida por mencionar algunos ejemplos. Recalcan que un elemento clave de I4.0 es la acrecentada capacidad de obtener datos de los procesos de fabricación y lo que es más importante, el análisis de tendencias y comportamientos que estos datos representan están revolucionando la forma en que se toman las decisiones en la industria. El internet de las cosas (IoT), el big data, los sistemas ciberfísicos, el aprendizaje automático y la manufactura aditiva (MA) o impresión 3D son tecnologías asociadas con la I4.0, entre varias otras.

El IoT podría considerarse una de las principales tecnologías de la I4.0 gracias a la convergencia de nuevas tecnologías de sensores a precios asequibles, junto con la capacidad cada vez mayor de los sistemas de información para almacenar y analizar grandes cantidades de datos (Big Data). El concepto sistemas ciberfísicos se refiere a la unión entre el mundo físico y digital con los sistemas de analítica de datos con el fin de mejorar la eficiencia de los sistemas industriales.

### **3.2. Manufactura Aditiva**

La Asociación Americana de Ensayo de Materiales (ASTM por sus siglas en inglés), establece en la norma ASTM-F2792-12<sup>a</sup> la siguiente definición: “*La manufactura aditiva (MA) es la fabricación de un objeto tridimensional sólido a partir de un modelo digital, agregando material en capas sucesivas*” como se muestra en la Figura 3.1.

**Figura 3.1.** *Proceso de manufactura aditiva.*



*Nota.* Adaptada de Zareiyan y Khoshnevis (2017).

Los procesos de MA actuales pueden utilizar una de varias tecnologías y una variedad de materiales, incluidos metales, termoplásticos, cerámica y vidrio, y ofrecen la capacidad de diseñar y crear productos complejos que utilizan un sistema de fabricación modificable en un marco de tiempo más corto que la fabricación tradicional (Crowell et al., 2018).

De acuerdo con Ahuett-Garza y Kurfess et al. (2018), la principal ventaja del proceso de MA es la transición prácticamente directa de un modelo digital a la parte física, sin la necesidad de herramientas especiales. Mencionan que, en la actualidad, la MA está bien establecida en aplicaciones específicas caracterizadas por un alto nivel de personalización y producción de bajo volumen.

Thompson et al. (2016), enfatizan que la MA tiene el potencial de apoyar una amplia gama de actividades que incluyen entre muchas otras: manufactura, automotriz, arquitectura, aeroespacial, aplicaciones militares y medicina; en esta última Eyers y Dotchev (2010) reportan ejemplos de implantes médicos personalizados en odontología (dientes, coronas, puentes y dentaduras postizas), ortopedia (extremidades artificiales, articulaciones de rodilla y copas acetabulares) y craneomaxilofacial (mandíbulas y cráneos).

### 3.3. Modelado por Deposición Fundida

De acuerdo con Perez (2014) y Sa'ude (2013), entre las muchas técnicas de MA, el modelado por deposición fundida (FDM)—también conocido como fabricación por filamento fundido (FFF), es popular debido a su bajo costo. Señalan que la tecnología FDM es adecuada para fabricar prototipos con materiales termoplásticos como acrilonitrilo butadieno estireno (ABS), nylon, ácido poliláctico (PLA) y otros. La tecnología FDM también puede imprimir materiales en forma mixta o combinada.

**Figura 3.2.** *Impresora 3D FDM Makerbot Replicator 5ª Generación™.*



---

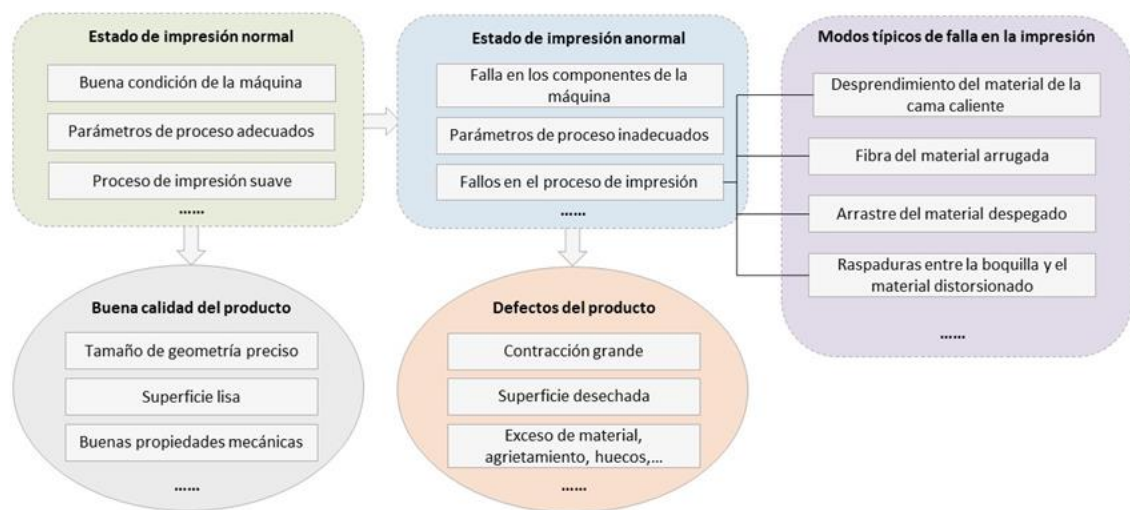
*Nota.* Tomada de Internet.

Actualmente sin embargo, las máquinas FDM comerciales tienen limitaciones en la confiabilidad del proceso y la calidad del producto ya que algunos parámetros de la máquina y de la operación pueden sufrir desviaciones durante el proceso de fabricación debido a que las condiciones de operación de las impresoras no se monitorean durante el proceso de operación (Wu et al., 2015), (Huang, 2015).

Para superar estas limitaciones, Wu et al. (2015) sugieren aumentar los niveles de inteligencia y automatización de la máquina para que las condiciones de operación de la impresora se monitorean más a fondo con sistemas de control de lazo cerrado y proponen un método no intrusivo para la identificación de los estados normales y anormales de operación (Figura 3.3) mediante el monitoreo de la impresora usando la técnica de emisión

acústica (EA) y métodos de aprendizaje automático tales como máquinas de soporte vectorial y función de base radial que se aplican para la identificación del estado, ya que afirman que aunque se han realizado una gran cantidad de trabajos de investigación para el monitoreo del producto durante la fabricación, no existía al momento de su investigación ningún estudio sobre el monitoreo de condiciones de operación de impresoras 3D FDM.

**Figura 3.3.** Principales estados de operación y su relación con la calidad del producto en FDM.



*Nota.* Adaptada de Wu, H. (2016).

Otros trabajos de investigación con métodos de monitoreo no intrusivos se enfocan en la detección de la obstrucción de la boquilla del extrusor mediante el monitoreo de la corriente del motor que alimenta el filamento (Kim, 2018) o sensores de EA que detectan detonaciones (Zhensheng Yang et al., 2018) y así inferir defectos de impresión; sin embargo ninguno de estos trabajos utilizó algoritmos de aprendizaje automático para identificar patrones en los datos de los sensores.

Becker et al. (2020) reportan el uso de las señales acústicas emitidas por una impresora 3D FDM, capturadas con un micrófono cardioide y aumentadas para entrenar

una red neuronal que clasifica en 6 clases definidas para los estados de operación de la impresora.

### **3.4. Monitoreo de condiciones de operación**

Aunque el monitoreo de condiciones del estado de una máquina es considerado como una importante estrategia de diagnóstico y mantenimiento para asegurar la calidad del producto, actualmente la mayoría de las máquinas de MA no están equipadas con sistemas de monitoreo y control de lazo cerrado, y sin la detección en tiempo real, los fallos del proceso o averías de la máquina se producen sin previo aviso durante el proceso de fabricación y la calidad y la repetitividad de los productos finales no se puede garantizar (Wu et al., 2017).

Wu et al. (2017) refiere que aunque el monitoreo de condiciones del estado de una máquina es considerado como una importante estrategia de diagnóstico y mantenimiento para asegurar la calidad del producto, actualmente la mayoría de las máquinas de MA no están equipadas con sistemas de monitoreo y control de lazo cerrado, y sin la detección en tiempo real, los fallos del proceso o averías de la máquina se producen sin previo aviso durante el proceso de fabricación y la calidad y la repetitividad de los productos finales no se puede garantizar. De manera reciente se han desarrollado varios métodos para el monitoreo de los procesos de impresión 3D usando diversos tipos de sensores. Los objetivos del monitoreo y diagnóstico se pueden dividir en dos grupos principalmente: 1) monitorear los estados de operación de la máquina, y 2) garantizar la calidad de las piezas impresas (Y. Li et al., 2019). Existen varios trabajos enfocados en el monitoreo de condiciones de operación y detección de anomalías en impresoras 3D de fusión de filamento usando diversos tipos de sensores y herramientas para el análisis de los datos. En impresoras 3D tipo cartesiano, Yoon et al. utilizaron sensores de emisión acústica y sensores piezoeléctricos analizando los datos con herramientas estadísticas convencionales. Wu et al. (2017) utilizó sensores de emisión acústica y modelos de

Markov, Becker et al. (2020) uso sensores acústicos y redes neuronales de memoria a largo corto-plazo (LSTM). En impresoras 3D tipo delta, S. Zhang (2019), Long et al. (2020) y Zhe Yang et al. (2021) usaron sistemas de referencia de actitud y rumbo y variantes de autocodificadores. Li et al. (2021) utilizó acelerómetros y codificadores adversarios generativos convolucionales.

De acuerdo con Flovik (2018), la detección de anomalías (o detección de valores atípicos) es la identificación de eventos u observaciones que difieren significativamente de la mayoría de los datos. Estos datos anómalos están generalmente relacionados con algún tipo de problema, por ejemplo, defectos estructurales, mal funcionamiento del equipo, etc. En el campo de la detección de anomalías vibratorias (DAV) en particular, la mayoría de los trabajos se han concentrado en el análisis de máquinas rotativas (turbinas, engranajes, etc.), especialmente en aeronáutica, donde Vos et al. (2022) usaron datos sin procesar en el dominio temporal y redes neuronales de memoria a largo corto-plazo (LSTM) para predecir las condiciones de operación de rodamientos en aeronaves.

### **3.5. Aprendizaje automático**

Una de las ramas de la IA es el aprendizaje automático<sup>1</sup>—en inglés, *machine learning* (ML)—que se define como la aplicación y ciencia de algoritmos que dan sentido a los datos, los cuales en esta era de tecnología moderna existen en abundancia, tanto estructurados como no estructurados (Raschka, 2016, p. 2).

En el ámbito industrial, los datos producidos por máquinas, sensores y sistemas de manufactura se multiplican a un ritmo acelerado. Ante este aumento, es esencial desarrollar sistemas inteligentes que vayan más allá de los simples modelos matemáticos y estadísticos, con el fin de convertir estos vastos conjuntos de datos en conocimientos prácticos. El aprendizaje automático ofrece la capacidad de diseñar tales sistemas, brindando perspectivas cruciales para la toma de decisiones eficientes en diversas áreas

---

<sup>1</sup> También llamado aprendizaje automatizado o aprendizaje de máquinas.

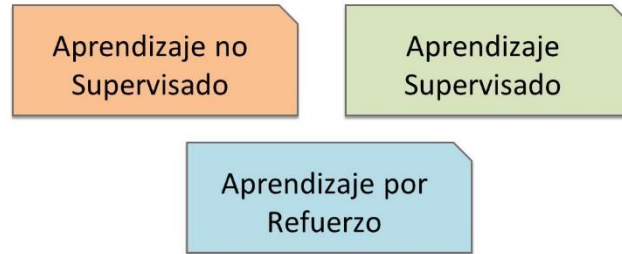
industriales. Esto es especialmente útil en situaciones como el diagnóstico de problemas, el mantenimiento anticipado, el monitoreo de condiciones y la gestión de la salud de equipos y herramientas (Larrañaga et al., 2018).

El aprendizaje automático de acuerdo con Raschka (2016), es una rama de la inteligencia artificial (IA) que se define como la aplicación y ciencia de algoritmos que permiten extraer información de los datos. Larrañaga et al. (2018), resalta que en la industria la cantidad de datos generados por controladores de máquinas, sensores, sistemas de fabricación, etc. está creciendo exponencialmente, lo que hace necesario crear sistemas inteligentes—más que modelos matemáticos y estadísticos—capaces de transformar esta gran cantidad de datos en conocimiento. El aprendizaje automático es la herramienta que permite construir dichos modelos que proporcionen información valiosa para la toma de decisiones en diferentes sectores industriales tales como diagnosticar de fallas, monitorear las condiciones de los equipos, gestionar la salud de activos, etc. LeCun et al. (2015) señala que el aprendizaje profundo—en inglés, *deep learning* (DL)—se ha convertido en un avance del aprendizaje automático para resolver problemas que requieren descubrir una estructura intrincada en datos de alta dimensionalidad. Alom et al. (2019) y Ma et al. (2018), subrayan que una diferencia clave entre el aprendizaje automático tradicional y el aprendizaje profundo es la forma en cómo se extraen las características de los datos. Esto es, los enfoques tradicionales de aprendizaje automático utilizan características de ingeniería desarrolladas por un humano mediante la aplicación de varios algoritmos de extracción de características y luego aplican los algoritmos de aprendizaje. En contraste, las redes neuronales profundas son capaces de extraer automáticamente las características relevantes de las señales sin procesar a través del procedimiento de aprendizaje por capas.

### **3.5.1.1. Los diferentes tipos de aprendizaje automático**

De acuerdo con Raschka (2016, pp. 2–6) existen tres tipos de aprendizaje automático como se muestra en la Figura 3.4: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

**Figura 3.4.** *Los tres tipos de aprendizaje supervisado.*

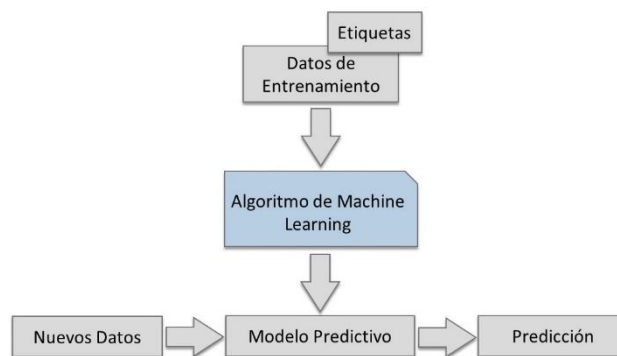


*Nota.* Adaptada de Raschka, S. (2016).

### **Aprendizaje supervisado**

El objetivo principal del aprendizaje supervisado es aprender (entrenar) un modelo a partir de un conjunto de datos etiquetados que luego permita hacer predicciones sobre datos no vistos o futuros. El término supervisado se refiere a un conjunto de muestras (datos de entrenamiento) donde las señales de salida deseadas (etiquetas) ya son conocidas.

**Figura 3.5.** *Aprendizaje supervisado.*



*Nota.* Adaptado de Raschka, S. (2016).



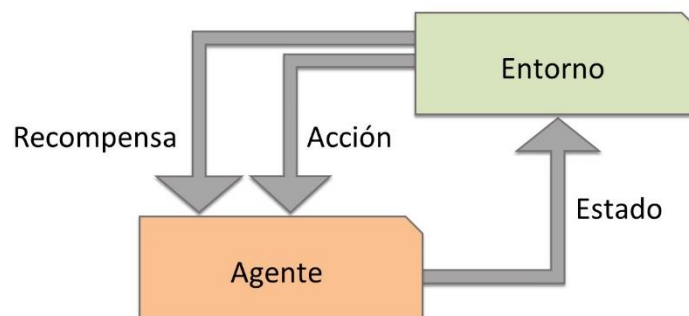
## Aprendizaje no supervisado

El aprendizaje no supervisado trata con datos no etiquetados o datos de estructura desconocida. Utilizando técnicas de aprendizaje no supervisadas, es posible explorar la estructura de los datos para extraer información significativa sin la guía de una variable de resultado conocida o función de recompensa. Una técnica para estructurar información y derivar relaciones significativas entre los datos es la agrupación en clústeres—en inglés *clustering*.

## Aprendizaje por refuerzo

En el aprendizaje por refuerzo, el objetivo es desarrollar un sistema (agente) que mejore su rendimiento en función de las interacciones con el entorno. A través de la interacción con el entorno, un agente puede utilizar el aprendizaje por refuerzo para aprender una serie de acciones que maximizan una recompensa a través de un enfoque exploratorio de prueba y error o planificación deliberativa.

**Figura 3.6.** *Aprendizaje por refuerzo.*



*Nota.* Adaptado de Raschka, S. (2016).

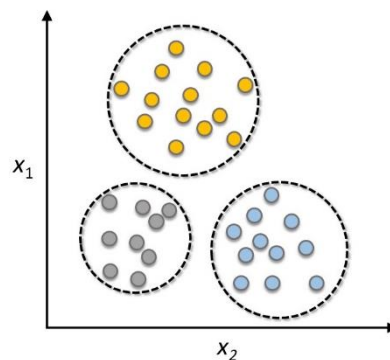
## Agrupación en clústeres

La agrupación en clústeres es una técnica exploratoria de análisis de datos que permite organizar un cúmulo de información en subgrupos significativos (agrupaciones)

sin tener ningún conocimiento previo de la membresía de grupo. Cada grupo que puede surgir durante el análisis define un grupo de objetos que comparten un cierto grado de similitud pero que son más diferentes a los objetos en otros grupos, por lo que a veces también se le denomina "clasificación no supervisada".

La Figura 3.7 muestra un agrupamiento en clústeres para organizar datos sin etiquetar en tres grupos distintos en función de la similitud de sus características  $x_1$  y  $x_2$ .

**Figura 3.7.** *Clustering o agrupamiento de datos en base a las características  $x_1$  y  $x_2$ .*



*Nota.* Adaptado de Raschka, S. (2016).

### 3.5.1.2. Aprendizaje profundo

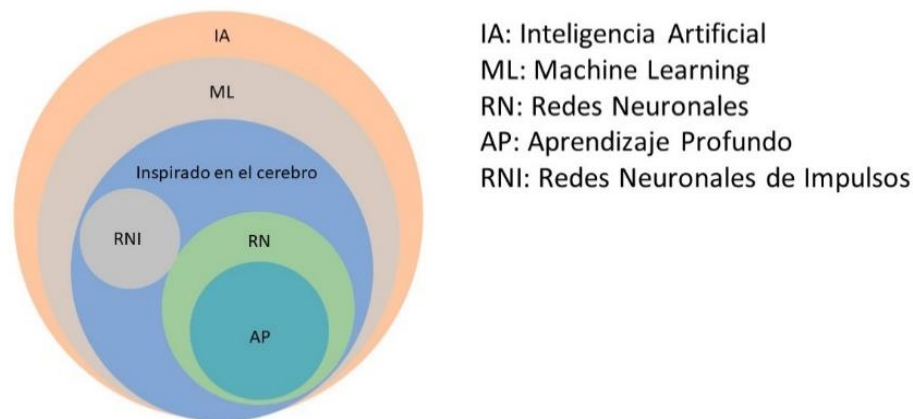
Una tendencia reciente en el aprendizaje automático es el aprendizaje profundo—en inglés *deep learning* (DL)—el cual ha surgido como un método para la detección de patrones en procesos complejos/simultáneos, utilizando como datos de entrada señales sin procesar (Ahuett-Garza y Kurfess, 2018).

De acuerdo con Alom et al. (2019), una diferencia clave entre el aprendizaje automático tradicional y el aprendizaje profundo es la forma en cómo se extraen las características de los datos, esto es, los enfoques tradicionales de aprendizaje automático utilizan características de ingeniería desarrolladas por un humano mediante la aplicación

de varios algoritmos de extracción de características y luego aplican los algoritmos de aprendizaje. Adicionalmente, a menudo se utilizan otros enfoques potenciadores en los que se aplican varios algoritmos de aprendizaje a las características de una sola tarea o conjunto de datos y se toma una decisión de acuerdo con los múltiples resultados de los diferentes algoritmos.

Por otro lado, en el caso del aprendizaje profundo, las características se aprenden automáticamente y se representan jerárquicamente en múltiples niveles. Este es el punto fuerte del aprendizaje profundo frente a los enfoques tradicionales de aprendizaje automático. La Figura 3.8 muestra la taxonomía de la IA en relación al aprendizaje profundo.

**Figura 3.8.** *El aprendizaje profundo en la taxonomía de la IA.*



*Nota.* Adaptada de Alom, M. (2019).

LeCun et al. et al. (2015) refieren que las técnicas convencionales de aprendizaje automático estuvieron limitadas para procesar datos naturales en su forma original (sin formato o no estructurados), y que durante décadas, la construcción de un sistema de reconocimiento de patrones o aprendizaje automático requirió una ingeniería cuidadosa y una considerable experiencia en el dominio para diseñar un extractor de características que transformara los datos de su forma original en una representación interna adecuada o en un

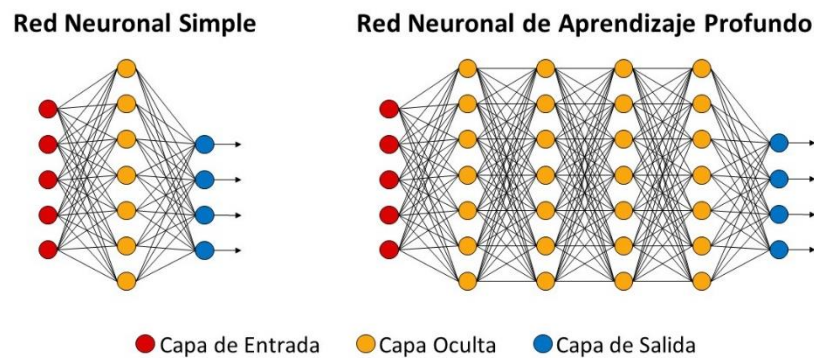
vector de características desde el cual el subsistema de aprendizaje—a menudo un clasificador, pudiera detectar o clasificar patrones en la entrada. Mencionan asimismo que en contraparte, los métodos de aprendizaje profundo son métodos con múltiples niveles de representación, obtenidos mediante la composición de módulos simples pero no lineales que transforman la representación en un nivel (comenzando con los datos de entrada sin formato) en una representación a un nivel más alto un poco más abstracto. Con la composición de suficientes transformaciones, se pueden aprender funciones muy complejas. Para las tareas de clasificación, las capas superiores de representación amplifican aspectos de la entrada que son importantes para la discriminación y suprimen variaciones irrelevantes. Recalcan que el aspecto clave del aprendizaje profundo es que estas capas de características no están diseñadas por ingenieros humanos, sino que son aprendidas de los datos mediante un procedimiento de aprendizaje de propósito general y han dado muy buenos resultados para descubrir estructuras complejas en datos de alta dimensionalidad y, por lo tanto, es aplicable a muchos dominios de la ciencia, los negocios y el gobierno. Mencionan que además el DL ha superado los registros en el reconocimiento de imágenes y el reconocimiento de voz, y a otras técnicas de aprendizaje automático para predecir la actividad molecular de potenciales fármacos, analizar los datos de aceleradores de partículas, reconstruir circuitos cerebrales y predecir los efectos de las mutaciones en ADN no codificante sobre expresión génica y enfermedades. Pero quizás lo más sorprendente es que el aprendizaje profundo ha producido resultados extremadamente prometedores para diversas tareas como la comprensión del lenguaje natural, particularmente en la clasificación de temas, el análisis de sentimientos, la respuesta a preguntas y la traducción de idiomas.

Mhaskar y Poggio (2016) revisaron los resultados de trabajos sobre arquitecturas jerárquicas para aprendizaje basado en ejemplos, que pueden explicar formalmente las condiciones bajo las cuales las redes neuronales convolucionales profundas funcionan mucho mejor en problemas de aproximación de funciones que las arquitecturas superficiales de una sola capa oculta.

Por otro lado, el bajo costo de almacenamiento de grandes conjuntos de datos y la informática de alto rendimiento han impulsado la aparición del aprendizaje profundo (Zhao et al., 2019). Con todo esto, LeCun et al. (2015) afirma que el aprendizaje profundo tendrá muchos más éxitos en el futuro cercano ya que requiere muy poca ingeniería manual y pueden aprovecharse fácilmente el aumento en la cantidad de disponibilidad de cómputo y datos.

Adicionalmente, el aprendizaje profundo se ha convertido en un avance del aprendizaje automático para resolver problemas que requieren descubrir una estructura intrincada en datos de alta dimensionalidad (LeCun et al., 2015), donde se espera que el aprendizaje profundo extraiga representaciones jerárquicas de los datos sin procesar a través de la construcción de redes neuronales profundas con múltiples capas de transformaciones no lineales.

**Figura 3.9.** *RNA simple (monocapa) vs. RNA profunda (multicapa).*



*Nota.* Adaptada de Internet.

En comparación con los modelos tradicionales de una sola capa, las redes neuronales profundas (ver Figura 3.9) son capaces de extraer automáticamente características relevantes de las señales sin procesar a través del procedimiento de aprendizaje por capas (W. Li, 2016), (Ma, Sun, & Chen, 2018).

### 3.5.1.3. Redes neuronales artificiales

Las redes neuronales artificiales (RNA) son aproximadores no lineales a la forma en que funciona el cerebro que sólo emulan en una parte muy simple su funcionamiento— por tal razón no deben compararse directamente con el cerebro ni confundir los principios que fundamentan el funcionamiento de las redes neurales artificiales y el cerebro. Debe considerarse además que las redes biológicas son generadoras de procesos neurobiológicos en que se establecen relaciones de complejidad muy alta, las cuales no pueden lograrse con redes neuronales artificiales ya sea monocapa o multicapa (Ponce-Cruz, 2010, p. 193).

#### Unidades en redes neuronales

Stuart y Norvig (2004) explican cómo están compuestas las redes neuronales. Mencionan que están constituidas de nodos o unidades (Figura 3.10) conectadas a través de conexiones dirigidas. Una conexión de la unidad  $j$  a la unidad  $i$  sirve para propagar la activación  $a_j$  de  $j$  a  $i$  y cada conexión tiene un peso numérico  $W_{j,i}$  asociado, que determina la fuerza y el signo de la conexión. Cada unidad  $i$  primero calcula una suma ponderada de sus entradas:

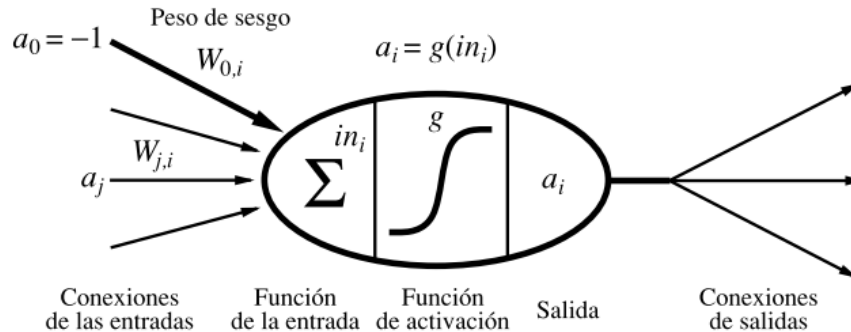
$$in_i = \sum_{j=0}^n W_{j,i} a_j \quad (3.1)$$

Luego se aplica una **función de activación**  $g$  a esta suma para producir la salida:

$$a_i = g(in_i) = g\left(\sum_{j=0}^n W_{j,i} a_j\right) \quad (3.2)$$

La función de activación  $g$  se diseña con dos objetivos: 1) que la unidad esté «activa» (cercana a +1) cuando se proporcionen las entradas «correctas» e «inactiva» (cercana a 0) cuando se den las entradas «erróneas». 2) la activación tiene que ser no *lineal*, en otro caso la red neuronal en su totalidad se colapsaría con una sencilla función lineal.

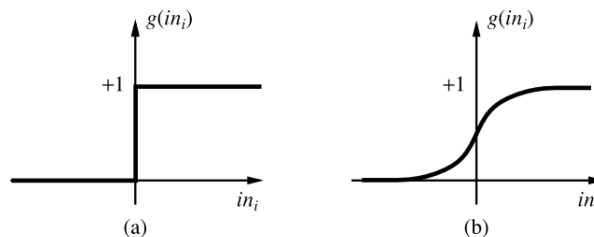
**Figura 3.10.** Modelo matemático simple de una neurona.



*Nota.* Adaptado de Stuart y Norvig (2004).

La Figura 3.11 muestra dos posibles funciones  $g$ : la función umbral y la función sigmoide (también conocida como función logística). La función sigmoide tiene la ventaja de ser diferenciable, lo que es importante para el algoritmo del aprendizaje de los pesos.

**Figura 3.11.** Función de activación.



(a) La función de activación umbral, con salida 1 cuando la entrada es positiva y 0 en otro caso. (b) La función sigmoide  $1/(1 + e^{-x})$ .

*Nota.* De Stuart y Norvig (2004).

Hay que notar que en la Figura 3.10 se ha incluido un peso de sesgo  $W_{0,i}$  conectado a una entrada fija  $a_0 = -1$ , lo que se explica a continuación.

Ambas funciones tienen un umbral (ni suave ni severo) de cero; los pesos de sesgo  $W_{0,i}$  constituyen el umbral *real* de la unidad, en el sentido de que la unidad se activa cuando

la suma de los pesos de las entradas «reales»  $\sum_{j=1}^n W_{j,i} a_j$  (es decir, incluyendo la entrada del sesgo) excede  $W_{0,i}$ .

### **Estructuras de las redes**

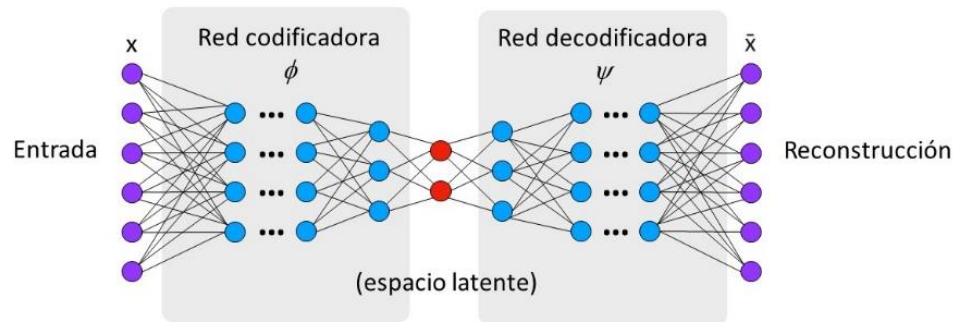
Existen dos categorías principales de estructuras de redes neuronales: acíclicas o redes con alimentación-hacia-adelante (feed-forward) y cíclicas o redes recurrentes. Una red con alimentación-hacia-delante representa una función de sus entradas actuales; de este modo, no tiene otro estado interno que no sea el de sus propios pesos. Por otro lado, una red recurrente permite que sus salidas alimenten sus propias entradas. Esto significa que los niveles de activación de la red forman un sistema dinámico que debe alcanzar un estado estable, exhibir oscilaciones o incluso un comportamiento caótico. Además, la respuesta de la red dadas unas entradas depende de su estado inicial, que dependerá de entradas previas. Por lo tanto, las redes recurrentes (a diferencia de las redes con alimentación hacia adelante) pueden tener memoria a corto plazo. Esto las hace más interesantes como modelos del cerebro, pero también más difíciles de entender

#### **3.5.1.4. Autocodificadores**

De acuerdo con López Pinaya et al. (2020), los autocodificadores son redes neuronales capaces de aprender de manera automática características de datos no etiquetados. Obtienen una representación compacta de los datos de entrada, lo que se le conoce como el espacio latente. El espacio latente se utiliza posteriormente para producir una salida que se asemeje lo máximo posible a la entrada, es decir, la red intenta replicar el dato inicial en la salida reduciendo al máximo el error. Los autocodificadores constan de dos partes: un codificador y un decodificador (Figura 3.12), cada uno con su propio conjunto de parámetros aprendibles. La función del codificador es tomar una entrada  $X$  (que puede ser cualquier tipo de datos, como imágenes 2D o 3D, audio, video o texto) y asignarlo a un espacio de codificación latente, creando un código latente  $Z$ .



**Figura 3.12.** Red neuronal tipo autocodificador.



*Nota.* Adaptada de Brunton y Kurtz (2021).

El codificador y el decodificador pueden implementarse utilizando arquitecturas personalizadas. Por ejemplo, pueden tener un diseño superficial que involucre solo una capa de entrada y una capa de salida o una estructura profunda que involucre un mayor número de capas y neuronas por capa y un tamaño de código más grande. El codificador y el decodificador pueden ser redes neuronales completamente conectadas o, si los datos vienen en un formato de cuadrícula como una imagen, pueden tener capas convolucionales y de agrupación para ser más eficientes. Al aumentar la complejidad de la arquitectura, el autocodificador podrá aprender una codificación latente más compleja.

Independientemente de la arquitectura, el objetivo principal del autocodificador es reconstruir su entrada con la mayor precisión posible. Esto se logra utilizando una función de pérdida específica—conocida como pérdida de reconstrucción—durante el entrenamiento del modelo. Esta función de pérdida suele ser el error cuadrático medio entre la salida  $\tilde{x}$  y la entrada  $x$ . Durante el entrenamiento, la función de pérdida penaliza a la red por generar salidas diferentes de las entradas.

$$\text{Función de pérdida} = \|x - \tilde{x}\|^2 \quad (3.3)$$

Brunton y Kutz (2021) recalcan que los autocodificadores ofrecen una estructura ventajosa para explotar características de baja dimensionalidad en datos de alta dimensionalidad. El autocodificador generaliza la incrustación del subespacio lineal de la descomposición de valores singulares (SVD) y el análisis de componentes principales (PCA) a una incrustación de variedad no lineal, a menudo de una dimensión inferior. Esto es, el autocodificador mapea los vectores de entrada originales de alta dimensionalidad  $x_j \in \mathbb{R}^n$  a una variable latente de baja dimensionalidad  $z_j \in \mathbb{R}^r$  y luego de vuelta al espacio de alta dimensionalidad  $\tilde{x}$ , que es técnicamente la salida  $y$ . El objetivo del autocodificador es mapear la salida a sí mismo, es decir,  $\|\tilde{x} - x\|_2 \approx 0$ . Típicamente  $r \ll n$  para la autocodificación y matemáticamente se expresa:

$$Z = \phi(X) \quad (3.4)$$

donde  $Z$  son los datos del espacio latente y  $X$  son los datos de entrada de alta dimensionalidad. Es preciso mencionar que las columnas de  $Z$  son  $z_j$  y las columnas de  $X$  son  $x_j$ . La decodificación se representa como

$$\tilde{X} = \psi(Z) \quad (3.5)$$

donde los pesos de la red neuronal se optimizan para que la salida  $\tilde{X}$  esté lo más cerca posible de la entrada,

$$\underset{\theta}{\operatorname{argmin}} \|X - \tilde{X}\|_2^2 = \underset{\theta}{\operatorname{argmin}} \|X - f_{\theta}(X)\|_2^2 \quad (3.6)$$

donde  $\theta$  son los pesos del autocodificador  $f_{\theta}(x) = \psi(\phi(x))$ . La dimensión del espacio latente  $z$  a menudo está determinada por el ajuste de los hiperparámetros. Por lo tanto,  $z$  se hace lo más pequeño posible hasta que el rendimiento del autocodificador comienza a disminuir.

Desde un punto de vista matemático, el autocodificador proporciona un mapeo como se ilustra en la Figura 3.12 de modo que:

$$\phi: X \rightarrow Z \quad (3.7)$$

$$\psi: Z \rightarrow \tilde{X} \quad (3.8)$$

donde la entrada  $x \in X \subseteq \mathbb{R}^n$  y la salida  $z \in Z \subseteq \mathbb{R}^r$  se definen en espacios de alta y baja dimensionalidad respectivamente. La optimización de la red neuronal resultante se formula en torno a la función de pérdida

$$\underset{\phi, \psi}{\operatorname{argmin}} = \|X - (\psi \circ \phi)X\|. \quad (3.9)$$

## 4. METODOLOGÍA

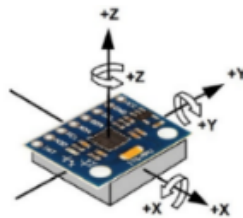
### 4.1. Materiales y métodos

#### 4.1.1. Selección de componentes

##### Unidad de medición inercial MPU6050

El sensor MPU6050 es un sensor inercial de bajo costo. Es un sistema microelectromecánico (MEMS) que puede ser utilizado para medir aceleraciones lineales y velocidades angulares. El uso combinado del acelerómetro de 3 ejes y giroscopio de 3 ejes integrado permite determinar el movimiento de un cuerpo en un espacio tridimensional. Posee adicionalmente una interfaz mediante bus I<sup>2</sup>C para la escritura y lectura de los registros del dispositivo, operando hasta una frecuencia de 400 kHz. (Fedorov et al., 2015).

**Figura 4.1.** *Unidad de medición inercial MPU6050.*



---

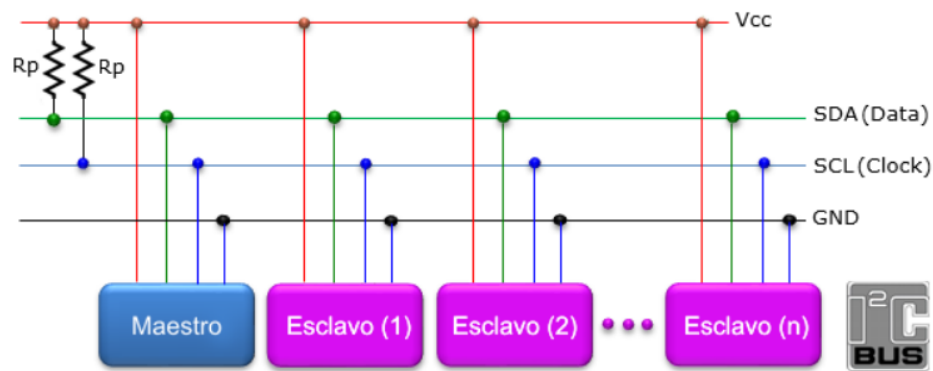
*Nota.* Tomada de Internet.

##### Bus serial I<sup>2</sup>C

El bus serial I<sup>2</sup>C (del inglés Inter Integrated Circuit), es un bus maestro/esclavo con un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos tales como sensores capaces de recolectar datos del mundo real. En la Figura

4.2 se muestra la estructura del bus I<sup>2</sup>C, el cual maneja 3 conexiones: SCL (del inglés System Clock), es la línea de los pulsos de reloj que sincronizan el sistema; SDA (del inglés System Data), es la línea por la que se mueven los datos entre los dispositivos; GND (del inglés Ground), común de la interconexión entre todos los dispositivos conectados al bus. Las líneas SDA y SCL son del tipo drenaje abierto por lo que se deben polarizar en estado alto por medio de resistencias “pull-up”. Adicionalmente, debe tomarse en cuenta que entre más datos se envíen por el bus mayor será la saturación de éste y por lo tanto es necesario calcular el valor adecuado de las resistencias y así garantizar una comunicación sin interrupciones (*Descripción Y Funcionamiento Del Bus I2C*).

**Figura 4.2.** Estructura del bus I<sup>2</sup>C.



*Nota.* Tomada de Internet.

## Python

De acuerdo con Nagpal y Gabrani (2019), ha existido una tradición entre los científicos y desarrolladores de utilizar lenguajes compilados como C, C++ y LISP para aplicaciones científicas y análisis de datos. Sin embargo, en los últimos años el uso de muchos lenguajes compilados está disminuyendo, dando paso al surgimiento de entornos interpretados como Octave, MATLAB y R. En la actualidad, el lenguaje de programación

Python está surgiendo como una alternativa a MATLAB, R y otros entornos relacionados y se ha vuelto popular por varias razones, entre ellas, es simple y tiene una sintaxis que es como pseudocódigo; su modularidad; su diseño orientado a objetos; sus capacidades de creación de perfiles, portabilidad, la posibilidad de realizar pruebas y auto documentación; y la presencia de una librería numérica que permite el almacenamiento efectivo y el manejo de enormes cantidades de información numérica. Con lo anterior y aunado al poder y la capacidad de realizar programación paralela y las capacidades de interfaz, Python representa un buen entorno para desarrollar ciencia computacional. Adicionalmente, hay muchas herramientas disponibles para facilitar esta integración. Por ejemplo, muchos entornos o marcos de trabajo basados en Python han mejorado la forma en que se programan muchas aplicaciones científicas. Tal es el caso de TensorFlow, PyTorch, Keras, Sklearn, etc., que han revolucionado el campo del aprendizaje profundo.

Stancin (2019) mencionan que Sklearn es la mejor biblioteca en el campo del aprendizaje automático ya que tiene una documentación muy buena e intuitiva con muchos ejemplos, además de una amplia gama de algoritmos implementados. Recomiendan asimismo el uso de PyTorch o Keras para la creación rápida de prototipos y TensorFlow para proyectos que exigen mucha personalización.

## **4.2. Metodología**

Algunos problemas durante el proceso de impresión de la pieza, tales como pandeo, defectos en las capas iniciales y defectos en las capas finales entre otros, producen interferencias entre la boquilla (la cual forma parte del sistema extrusor) y la pieza asentada en la cama de impresión. Esto genera que el sistema extrusor y la cama de impresión presenten vibraciones anormales, inclusive la interferencia puede ser tal que cause la pérdida de referencia del sistema extrusor. Asimismo, la vibración anormal debida a desgaste mecánico, falta de lubricación, etc. puede reflejarse en desplazamientos erráticos del sistema extrusor y de la cama de impresión. Adicionalmente de acuerdo con D. Zhang

et al. (2019), bajo ciertas condiciones durante el proceso de impresión, si la frecuencia de vibración de la boquilla/extrusor está cerca de la frecuencia natural de la impresora, se puede producir resonancia en la impresora, lo que afectará seriamente la precisión de la impresión e incluso provocará defectos en la pieza impresa.

Para abordar estos problemas se propone un método de detección de anomalías vibratorias bajo el enfoque del aprendizaje automático semi-supervisado. Este enfoque combina el aprendizaje supervisado y el aprendizaje no supervisado y el cual es menos dependiente de datos etiquetados. La idea es utilizar un autocodificador (Goodfellow et al., 2016) para aprender el comportamiento normal de la impresora y luego usarlo para detectar anomalías.

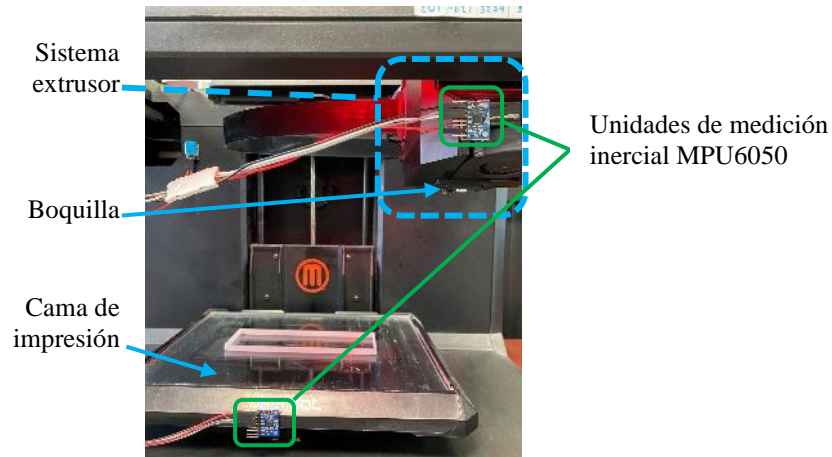
#### **4.2.1. Detección de anomalías usando autocodificadores**

Según Borghesi et al. (2019), en fechas recientes los autocodificadores han sido aplicados bajo el paradigma del aprendizaje semi-supervisado, especialmente para la detección de anomalías en diversas áreas tales como supercomputadoras, equipos de manufactura y plantas industriales. La idea central es entrenar el autocodificador para que aprenda el comportamiento normal de los equipos con datos etiquetados y, después del entrenamiento, usarlo para identificar condiciones anormales en base a la magnitud del error de reconstrucción. La premisa es que si el autocodificador aprende las correlaciones entre el conjunto de mediciones (características) que describen el estado normal del equipo bajo análisis, entonces puede notar cambios en estas correlaciones que indiquen un estado anormal. En condiciones normales de funcionamiento, estas características están vinculadas por relaciones específicas (es decir, variables físicas y químicas). La hipótesis es que estas correlaciones se verán perturbadas si el sistema entra en un estado anómalo.

### 4.2.2. Instrumentación de la impresora

En la Figura 4.3 se muestra la impresora 3D Makerbot Replicator 5ª generación la cual se instrumentó con dos unidades de medición inercial MPU6050 (InvenSense Inc., 2013), una en el extrusor y una en la cama de impresión. Las unidades de medición inercial pueden medir tres aceleraciones lineales y tres velocidades angulares y escribir los registros a un dispositivo mediante un bus I<sup>2</sup>C a una frecuencia de hasta 400 kHz (Fedorov et al., 2015).

**Figura 4.3.** *Instrumentación de la impresora.*



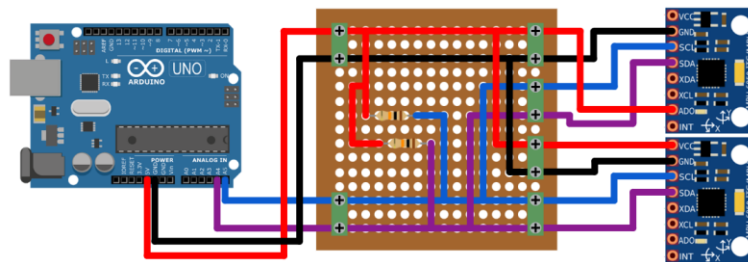
### 4.2.3. Implementación del bus I<sup>2</sup>C

Las unidades de medición inercial MPU6050 están conectadas al módulo I<sup>2</sup>C del microcontrolador Arduino UNO como se muestra en la Figura 4.4. Los dispositivos MPU6050 actúan como esclavos y el Arduino UNO como maestro.



Cabe mencionar que el sensor MPU6050 cuenta con resistencias pull-up para la conexión de un solo sensor al bus serial, pero al añadir otro sensor al bus, ambas resistencias son insuficientes y se requiere añadir resistencias externas. En este caso, para la conexión de dos dispositivos MPU6050 con Arduino UNO se utilizaron resistencias de 10 k $\Omega$  tanto para la línea SCL como para la línea SDA. Sin embargo, esto no implica una generalización para cualquier circuito ya que en otros circuitos se puede presentar una mayor o menor saturación del bus I<sup>2</sup>C según la cantidad de datos que se estén compartiendo a través del bus.

**Figura 4.4.** Esquema de conexión del bus I<sup>2</sup>C.



*Nota.* Imágenes tomadas de internet. Dibujo del circuito: Esquivel, I. (2022).

### **4.3. Desarrollo**

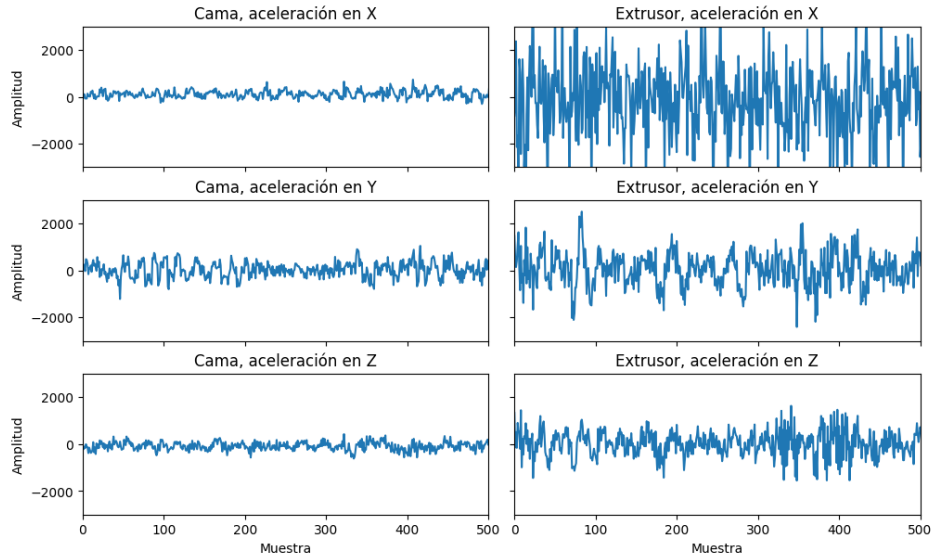
En esta sección se describe la manera en que se llevó a cabo la colecta de los datos y el entrenamiento del autocodificador utilizando diferentes enfoques.

#### **4.3.1. Colecta de los datos**

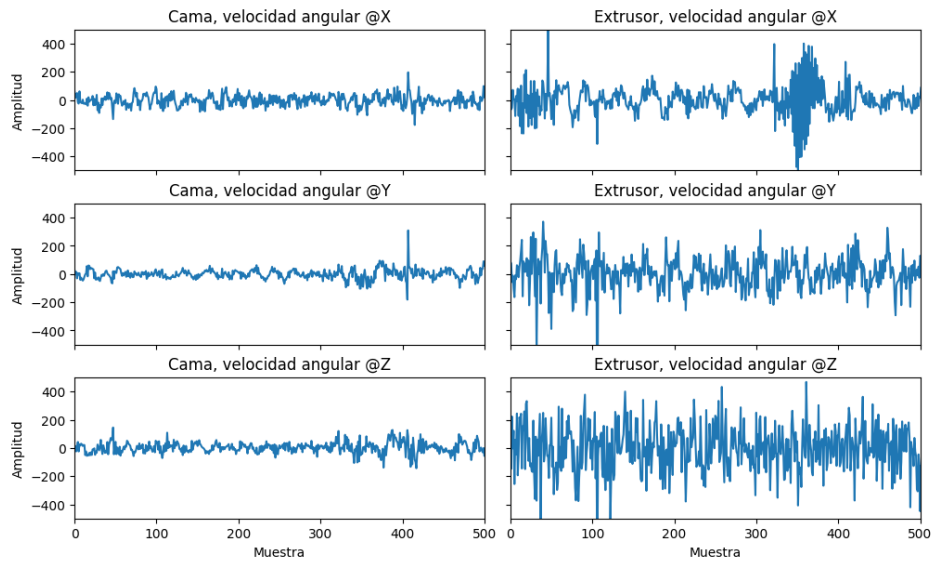
Se recolectaron datos de 12 variables durante trabajos de impresión de diversas piezas. Esto es, tres aceleraciones lineales X, Y, Z y tres velocidades angulares @X, @Y, @Z, tanto de la plataforma como del sistema extrusor. La lectura de las 12 variables se realizó a una frecuencia de 20 Hz. La velocidad de avance del extrusor se definió en 4 mm/s con lo que las lecturas se obtuvieron cada 0.2 mm de recorrido del extrusor.

En la Figura 4.5 y Figura 4.6 se muestran extractos de las aceleraciones y de las velocidades angulares de la cama de impresión y del extrusor. Se puede observar que existen diferencias en la amplitud de las señales medidas en cada uno de los 3 ejes. Esto se debe a que durante el proceso de impresión la cama permanece mayormente estática y el extrusor se encuentra en movimiento.

**Figura 4.5.** Aceleraciones X, Y, Z, condición normal.



**Figura 4.6.** Velocidades angulares @ X, @Y, @Z, condición normal.



### 4.3.2. Arquitectura del autocodificador

La topología del autocodificador se muestra en la Tabla 1. Los 12 nodos de entrada y los 12 nodos de salida del autocodificador corresponden con las 12 variables medidas (6 aceleraciones y 6 velocidades angulares). Es una red neuronal totalmente conectada con 5 capas ocultas. Las capas 1 y 5 poseen 10 neuronas cada una; las capas 2 y 4 poseen 6 neuronas cada una. El espacio latente—capa central—posee 4 neuronas. La red presenta 456 parámetros aprendibles.

**Tabla 1.** *Topología del autocodificador*

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 12)]	0
dense (Dense)	(None, 10)	130
dense_1 (Dense)	(None, 6)	66
dense_2 (Dense)	(None, 4)	28
dense_3 (Dense)	(None, 6)	30
dense_4 (Dense)	(None, 10)	70
dense_5 (Dense)	(None, 12)	132

=====  
Total params: 456  
Trainable params: 456  
Non-trainable params: 0  
=====

*Nota.* Generada en Google Colab.

### 4.3.3. Entrenamiento del autocodificador

El autocodificador se entrenó de forma semi-supervisada, esto es, se entrenó exclusivamente con datos de operación normal de la impresora. Se entrenó bajo 3 enfoques distintos, esto es, se entrenó con datos sin preprocesar (datos crudos) y con datos preprocesados con VAP y RMS. Esto con la finalidad de evaluar si existen diferencias sustanciales en el desempeño del autocodificador al realizar las predicciones.

La idea central es que al entrenar el autocodificador, el error de reconstrucción—esto es, la diferencia entre el dato de entrada y el dato reconstruido—se minimice. Si el autocodificador se entrena únicamente con datos de condición normal  $X_{Train}^N$ , éste aprenderá las relaciones de ese conjunto de entrada y minimizará el error en base a esos datos. Una vez entrenado el autocodificador, si se le presentan datos de condición anormal  $X_{Test}^A$ , el error de reconstrucción deberá ser sustancialmente mayor.

Los datos se normalizaron entre 0 y 1 y se dividieron 80% para el conjunto de entrenamiento  $X_{Train}^N$  y 20% para el conjunto de prueba durante el entrenamiento  $X_{Test}^N$ . Se usó tamaño de lote 32, 100 épocas, optimizador ADAM y como función de pérdida el error cuadrático medio (ECM).

Para efectos de comparar el error de los diferentes conjuntos una vez entrenado el autocodificador, se normalizó el *RMSE* de cada conjunto dividiendo por el *RMSE* de  $X_{Train}^N$ . De esta manera, el *RMSE* normalizado de  $X_{Train}^N = 1$  es la referencia para una reconstrucción ideal y cuanto mayor sea el *RMSE* normalizado de un conjunto dado, esto indicará un mayor error de reconstrucción de ese conjunto (Borghesi et al., 2019), lo que significa que el autocodificador es capaz de diferenciar mejor los datos normales de los anormales.

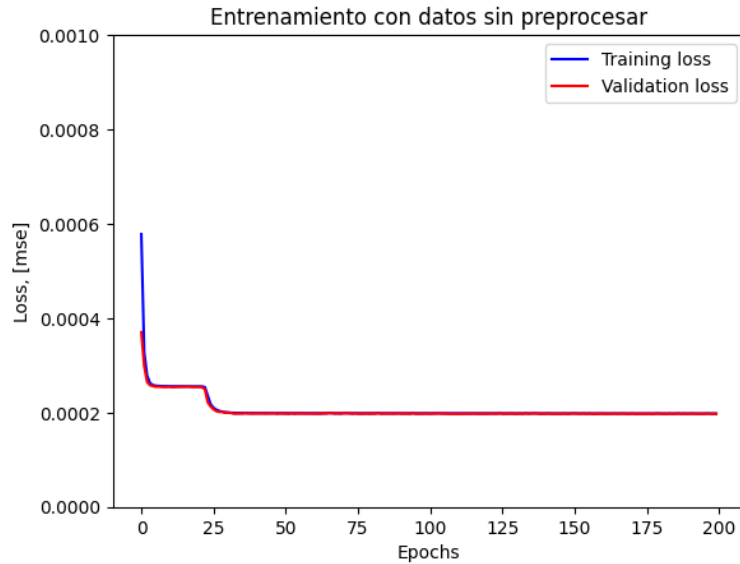
#### **4.3.3.1. Entrenamiento con datos sin preprocesar**

En este enfoque se entrenó el autocodificador con los datos sin preprocesar, esto es, se introdujeron a la red neuronal los datos medidos—datos crudos—de los 12 canales (Figura 4.5 y Figura 4.6). Los datos se normalizaron entre 0 y 1 para garantizar la convergencia del algoritmo.

La Figura 4.7 muestra la evolución del proceso de entrenamiento del autocodificador con datos sin preprocesar. Podemos remarcar que el modelo tiene un buen desempeño y se logra un buen ajuste ya que durante el entrenamiento el error del conjunto de entrenamiento (training loss) y el error del conjunto de validación (validation loss),

ambos decrecen y eventualmente se estabilizan en un punto específico, en este caso que después de aproximadamente 30 épocas—donde las curvas se vuelven asintóticas.

**Figura 4.7.** *Entrenamiento del modelo con datos sin preprocesar.*

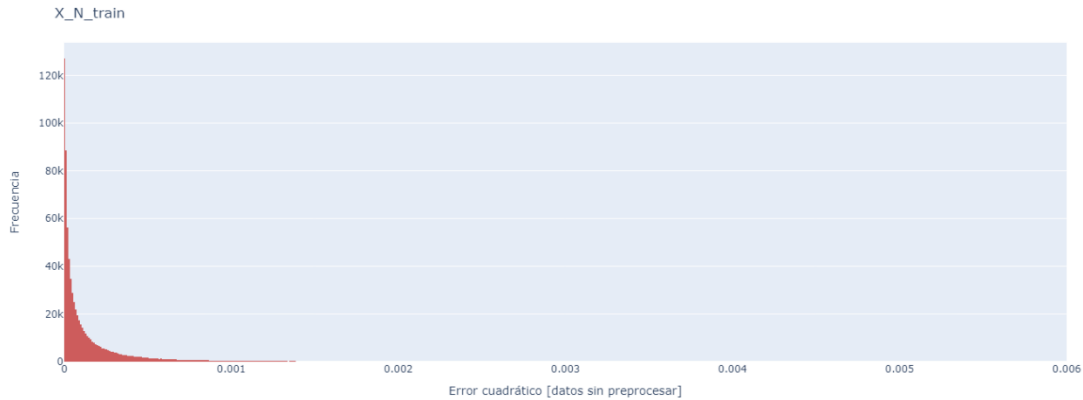


La Figura 4.8 muestra la distribución del error cuadrático de las reconstrucciones con datos sin preprocesar sobre el conjunto de entrenamiento de condición normal  $X_{Train}^N$ . Esto es, una vez entrenado el autcodificador—ajustados los pesos—se introducen nuevamente todos los datos del conjunto  $X_{Train}^N$  para ver el error de reconstrucción de cada dato.

En una reconstrucción ideal el error debería ser cero. En este caso podemos ver que el error se concentra por debajo de 0.0015. Esto valida la hipótesis de que si el autcodificador aprende adecuadamente las relaciones entre los datos de condición normal ( $X_{Train}^N$ ), y después del entrenamiento se le presentan datos pertenecientes a una condición

similar ( $X_{Test}^N$ ), el autocodificador puede reconstruir el dato con el mínimo error. Por el contrario, si son datos de condición anormal ( $X_{Test}^A$ ), el error de reconstrucción es mayor.

**Figura 4.8.** Distribución del error cuadrático de las predicciones sobre  $X_{Train}^N$  con datos sin preprocesar.



Como se observa en la Tabla 2, la RECM normalizada de los datos de prueba de condición anormal  $X_{Test}^A$  es 4.84 veces mayor que la RECM normalizada de los datos del conjunto  $X_{Train}^N$  y del conjunto  $X_{Test}^N$ .

**Tabla 2.** RECM de los diferentes conjuntos con datos sin preprocesar.

	$X_{Train}^N$	$X_{Test}^N$	$X_{Test}^A$
RECM	0.013	0.013	0.063
RECM normalizada	1	1	4.84

#### 4.3.3.2. Entrenamiento con datos preprocesados con VAP

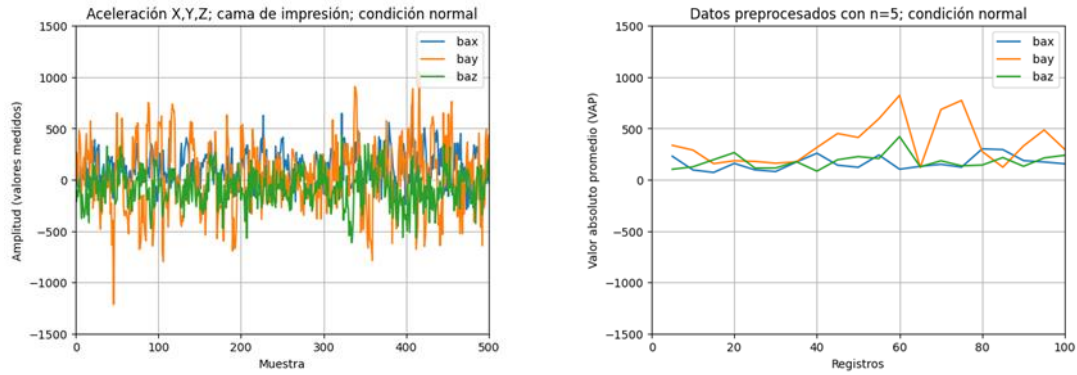
Para evaluar el impacto del preprocesamiento—reducción del ruido— en los datos, las lecturas en el dominio temporal se preprocesaron usando el valor absoluto promedio (VAP) con un tamaño de ventana de 5. Esto es, por cada 5 datos de cada una de las 12 señales medidas se generó un nuevo registro para crear el conjunto de condición normal  $X^N$  que se usó para entrenar el autocodificador.

Para el cálculo del VAP se utilizó la siguiente fórmula:

$$VAP(x_{input}) = \frac{1}{n} \sum_{i=1}^n |x_i|, \text{ donde } n \text{ es el tamaño de ventana.} \quad (4.1)$$

En la Figura 4.9 a) y b) se muestra un extracto de los datos de las aceleraciones X, Y, Z de la cama de impresión sin preprocesar y preprocesados respectivamente. Puede notarse que con VAP obtenemos un suavizado de las señales, así como también únicamente valores positivos.

**Figura 4.9.** Aceleraciones de la cama de impresión, condición normal.



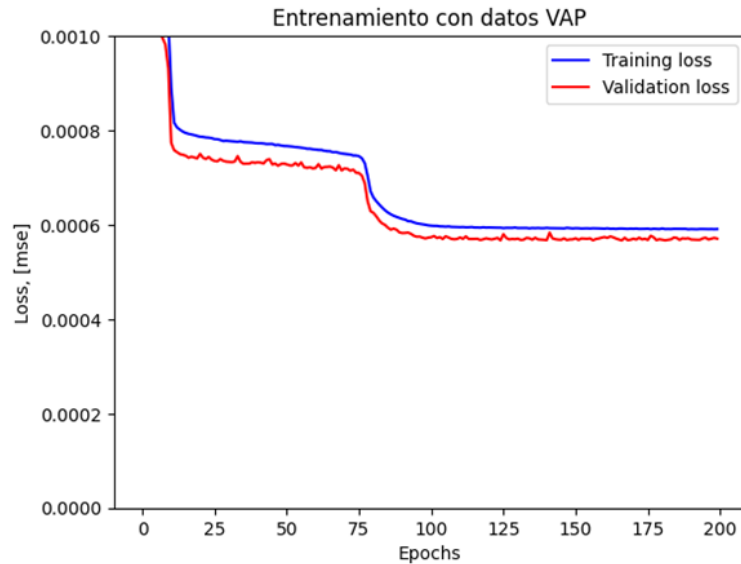
a) Datos sin preprocesar

b) Datos preprocesados con VAP

La Figura 4.10 muestra la evolución del proceso de entrenamiento del autocodificador con datos VAP. En este caso podemos decir que el modelo también tiene un buen desempeño—al igual que con los datos sin preprocesar—ya que se logra un buen ajuste, volviéndose asintóticas las curvas después de aproximadamente 100 épocas.

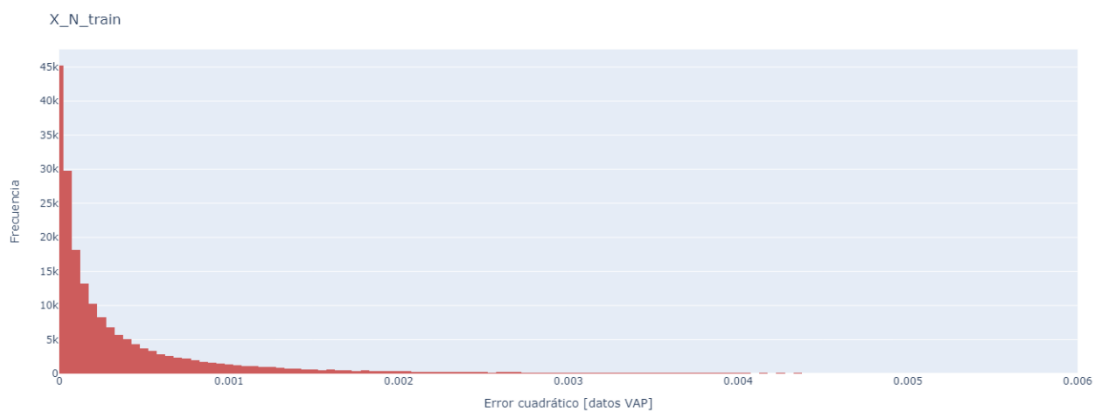


**Figura 4.10.** *Entrenamiento del modelo con datos VAP.*



La Figura 4.11 muestra la distribución del error cuadrático de las reconstrucciones sobre el conjunto de entrenamiento de condición normal  $X_{Train}^N$  con datos VAP.

**Figura 4.11.** *Distribución del error cuadrático de las predicciones sobre  $X_{Train}^N$  con datos VAP.*



Como se aprecia en la Tabla 3, la RECM normalizada de los datos de prueba de condición anormal  $X_{Test}^A$  es 5.70 veces mayor que la RECM normalizada de los datos del conjunto  $X_{Train}^N$  y del conjunto  $X_{Test}^N$ .

**Tabla 3.** RECM de los diferentes conjuntos con datos VAP.

	$X_{Train}^N$	$X_{Test}^N$	$X_{Test}^A$
RECM	0.024	0.024	0.137
RECM normalizada	1	1	5.70

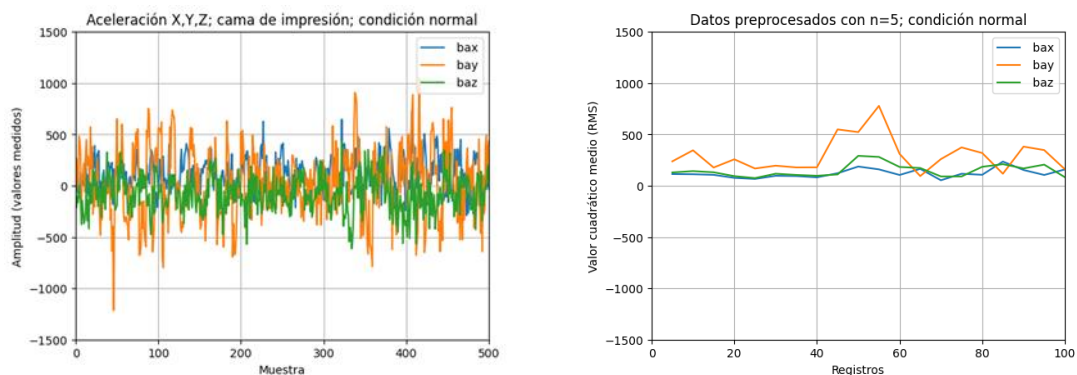
### 4.3.3.3. Entrenamiento con datos preprocesados con RMS

Para el cálculo del valor cuadrático medio (RMS) se utilizó la siguiente fórmula:

$$RMS(x_{in}) = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}, \text{ donde } n \text{ es el tamaño de ventana.} \quad (4.2)$$

En la Figura 4.12 a) y b) se muestra un extracto de los datos sin preprocesar y preprocesados respectivamente de las aceleraciones X, Y, Z de la cama de impresión. Puede notarse que con RMS obtenemos también un suavizado de las señales y únicamente valores positivos.

**Figura 4.12.** Aceleraciones de la cama de impresión, condición normal.



a) Datos sin preprocesar

b) Datos preprocesados con RMS

La Figura 4.13 muestra la distribución del error cuadrático de las predicciones sobre el conjunto de entrenamiento de condición normal  $X_{Train}^N$  con registros RMS.

**Figura 4.13.** Distribución del error cuadrático de las predicciones sobre  $X_{Train}^N$  con datos RMS.



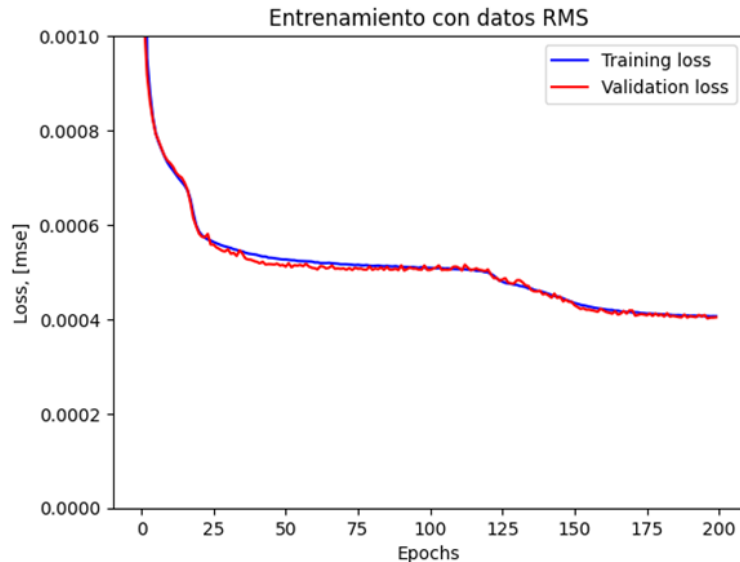
La Tabla 4 muestra que la RECM normalizada de los datos de prueba de condición anormal  $X_{Test}^A$  es 5.150 veces mayor que la RECM normalizada de los datos del conjunto  $X_{Train}^N$  y del conjunto  $X_{Test}^N$ .

**Tabla 4.** RECM de los diferentes conjuntos con datos RMS.

	$X_{Train}^N$	$X_{Test}^N$	$X_{Test}^A$
RECM	0.020	0.020	0.103
RECM normalizado	1	1	5.150

La Figura 4.14 muestra la evolución del proceso de entrenamiento del autocodificador con datos RMS. En este caso podemos decir que el modelo también tiene un buen desempeño—al igual que con los datos sin preprocesar y datos VAP—ya que se logra un buen ajuste, volviéndose asintóticas las curvas después de aproximadamente 200 épocas.

**Figura 4.14.** *Entrenamiento del modelo con datos RMS.*



#### **4.3.4. Comparación del entrenamiento del autocodificador con los diferentes enfoques**

En la Tabla 5 se muestran los valores obtenidos de RECM normalizado de las predicciones sobre el conjunto de datos de condición anormal ( $X_{Test}^A$ ) de los diferentes enfoques que se ensayaron—datos sin preprocesar, datos VAP y datos RMS. Es preciso recordar que entre mayor sea el RECM normalizado (ver sección 4.3.3) esto significa que el autocodificador puede diferenciar mejor entre datos de condición normal y anormal. Se muestran asimismo el número de épocas donde se alcanzó el comportamiento asintótico para cada enfoque.

**Tabla 5.** Comparación de RECM normalizado sobre  $X_{Test}^A$  y las épocas.

	Datos sin preprocesar	Datos VAP	Datos RMS
RECM normalizado	4.84	5.70	5.15
Épocas	30	100	200

En este caso el entrenamiento con datos VAP es el que presenta la mejor condición para reconocer datos anómalos con un RECM normalizado de 5.70.

## 5. RESULTADOS

En esta sección se presentan los resultados de las predicciones de los diferentes enfoques, esto es, con datos sin preprocesar, datos VAP y datos RMS. Al hablar de predicciones se hace referencia al resultado de la clasificación que se obtiene—dato de condición normal o anormal—al presentarle a la red neuronal datos no vistos anteriormente.

### 5.1. Predicción de anomalías vibratorias

Para la predicción de anomalías se utilizó el ECM que se define como:

$$ECM = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (5.1)$$

donde  $x_i$  es el dato original y  $\hat{x}_i$  es el dato reconstruido.

Si el ECM de la predicción  $\hat{x}$  supera un umbral definido, entonces ese dato se clasifica como anomalía. Se optó por definir el umbral de una manera simple a partir de la distribución del error de los datos de entrenamiento de condición normal  $X_{Train}^N$ .

Para evaluar cuantitativamente el desempeño del método propuesto se usó la precisión y sensibilidad de las predicciones sobre el conjunto de prueba  $X_{Test}$ . El conjunto  $X_{Test}$  contiene las 2 clases de datos, esto es, registros de condición normal y registros de condición anormal.

La precisión nos indica la tasa de verdaderos negativos (VN) clasificados correctamente por el algoritmo y se define como:

$$Precisión = \frac{VP}{VP + FP} \quad (5.2)$$

Cuanto más cercana la precisión a uno, esto indica que los registros de condición normal son en su mayoría clasificados correctamente. En este caso se definió como VN a un dato de condición normal clasificado como normal (ver la Matriz de Confusión en la Figura 5.1).

Por otro lado, la sensibilidad nos indica la tasa de verdaderos positivos (VP) clasificados correctamente por el algoritmo y se define como:

$$Sensibilidad = \frac{VP}{VP + FN} \quad ( 5.3 )$$

Cuanto más cercana la sensibilidad a uno, esto indica que las anomalías son en su mayoría clasificadas correctamente. En este caso se definió como VP a las anomalías clasificadas correctamente como anomalías (ver Figura 5.1).

**Figura 5.1.** *Matriz de confusión del proyecto.*

---

		Predicción	
		Normal	Anormal
Real	Normal	Verdadero negativo (VN): normal clasificado como normal	Falso positivo (FP) o error tipo1: normal clasificado como anormal
	Anormal	Falso negativo (FN) o error tipo 2: anormal clasificado como normal	Verdadero positivo (VP): anormal clasificado como anormal

---

### 5.1.1. Predicciones con datos sin preprocesar

En la Figura 5.2 observamos que el error de reconstrucción está concentrado por debajo de 0.001, por lo que se usó ese valor como umbral para las predicciones sobre nuevos datos.

**Figura 5.2.** Umbral para las predicciones sobre  $X_{Train}^N$  con datos sin preprocesar.



La matriz de confusión de las predicciones con datos sin preprocesar se muestra en la Tabla 6. Podemos observar que 337 registros anormales fueron clasificados correctamente y 93 incorrectamente, obteniendo una *Sensibilidad* de 0.784, lo que representa que el 78.4 % del total de las anomalías son clasificados correctamente.

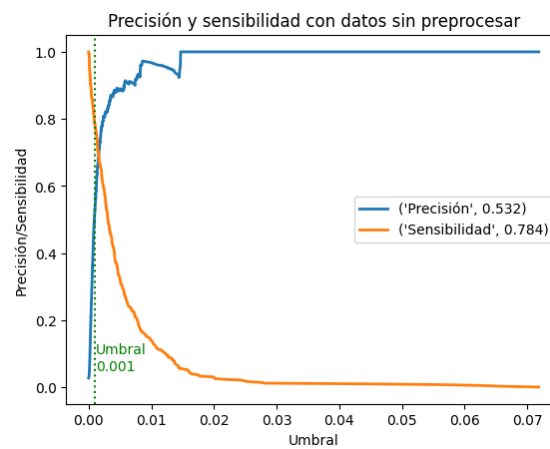
**Tabla 6.** Matriz de confusión de las predicciones con datos sin preprocesar.

		Predicción	
		Normal	Anormal
Real	Normal	15,162	297
	Anormal	93	337



En la Figura 5.3 se muestran las curvas de precisión y sensibilidad para diferentes umbrales para datos sin preprocesar. Observamos que dependiendo del umbral que se seleccione se obtendrá un cierto valor de sensibilidad y precisión en las predicciones. En este caso nos interesa lograr una sensibilidad—dato anormal clasificado correctamente—lo más cercana a 1 pero sin penalizar demasiado la precisión—dato normal clasificado correctamente.

**Figura 5.3.** *Precisión y sensibilidad para diferentes umbrales con datos sin preprocesar.*



### 5.1.2. Predicciones con datos VAP

En la Figura 5.4 observamos que el error de reconstrucción está concentrado por debajo de 0.0035, por lo que se usó ese valor como umbral para las predicciones sobre nuevos datos.

**Figura 5.4.** Umbral para las predicciones sobre  $X_{Train}^N$  con VAP



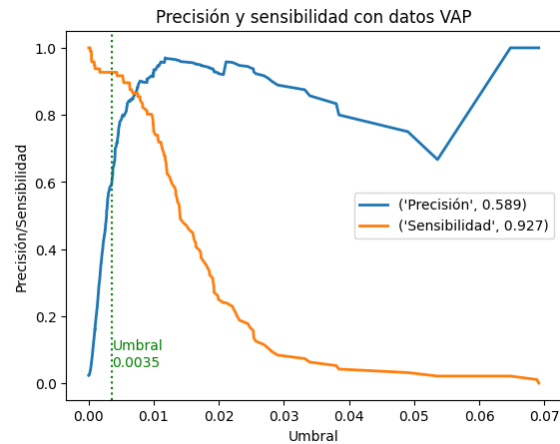
La matriz de confusión de las predicciones con datos VAP se muestra en la **Tabla 7**. En este caso 89 registros anormales fueron clasificados correctamente y 7 incorrectamente, obteniendo una *Sensibilidad* de 0.927, lo que representa que el 92.7 % del total de las anomalías son clasificados correctamente.

**Tabla 7.** Matriz de confusión de las predicciones con datos VAP.

		Predicción	
		Normal	Anormal
Real	Normal	4,014	62
	Anormal	7	89

En la **Figura 5.5** se muestran las curvas de precisión y sensibilidad con datos VAP.

**Figura 5.5.** *Precisión y sensibilidad para diferentes umbrales con VAP.*



---

### 5.1.3. Predicciones con datos RMS

La Figura 5.6 muestra que el error de reconstrucción con datos RMS está concentrado por debajo de 0.0035, por lo que se usó ese valor como umbral para las predicciones sobre nuevos datos.

**Figura 5.6.** Umbral para las predicciones sobre  $X_{Train}^N$  con RMS.



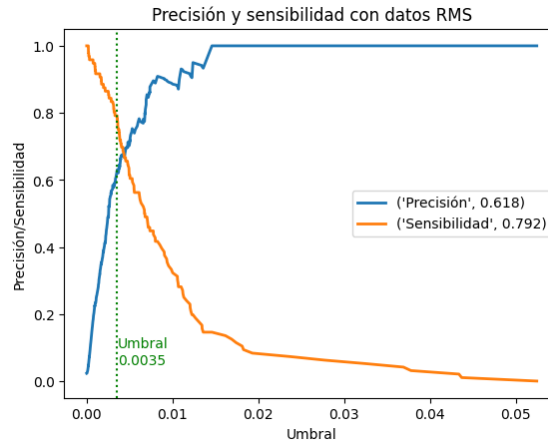
La matriz de confusión de las predicciones con datos RMS se muestra en la Tabla 8. En este caso 76 registros anormales fueron clasificados correctamente y 20 incorrectamente, obteniendo una *Sensibilidad* de 0.792, lo que representa que el 79.2 % del total de las anomalías son clasificadas correctamente.

**Tabla 8.** Matriz de confusión de las predicciones con datos RMS.

		Predicción	
		Normal	Anormal
Real	Normal	4,029	47
	Anormal	20	76

En la Figura 5.7 se muestran las curvas de precisión y sensibilidad con datos RMS.

**Figura 5.7.** *Precisión y sensibilidad para diferentes umbrales con RMS.*



## 5.2. Comparación de las predicciones de los diferentes enfoques

En la Tabla 9 se resumen la precisión y sensibilidad obtenidas para los diferentes enfoques. Como se mencionó en la sección 5.1, estos parámetros nos indican el porcentaje de predicciones clasificados correctamente por el algoritmo. La precisión indica la tasa de verdaderos negativos—dato normal clasificado correctamente, mientras que la sensibilidad indica la tasa de verdaderos positivos— dato anormal clasificado correctamente.

**Tabla 9.** *Precisión y sensibilidad de los diferentes enfoques.*

	Datos sin preprocesar	Datos VAP	Datos RMS
Precisión	0.532	0.589	0.618
Sensibilidad	0.784	0.927	0.792

Es preciso recalcar que en este trabajo se dio prioridad a la sensibilidad sobre la precisión ya que el interés es lograr que las anomalías sean detectadas correctamente la mayor parte de las veces.

## 6. DISCUSIÓN Y CONCLUSIONES

### 6.1. Discusión

Respecto al desempeño del algoritmo para clasificar anomalías vibratorias, dado que se da prioridad a la detección de la condición anormal operación de la impresora, el porcentaje de falsos positivos o error tipo 1—condición normal clasificada como anormal—puede ser alto. Sin embargo, este problema pudiera acometerse mejorando la sintonización de los hiperparámetros y agregando alguna técnica de regularización a la red neuronal.

El conjunto de datos de estado anormal de operación  $X_{Test}^A$  utilizado para probar el desempeño del autocodificador es relativamente pequeño. Esto debido a la dificultad para reproducir fallas en la impresora. Sin embargo, la base de datos de anomalías se irá incrementando y adicionalmente, el autocodificador permitirá identificar otras anomalías difíciles de detectar por el usuario de manera observacional.

### 6.2. Conclusiones

En este trabajo se presentó una metodología para predecir anomalías vibratorias en una impresora 3D de fusión de filamento y determinar si dicho equipo opera en condición normal o anormal. Se analizaron datos de vibración y velocidad angular obtenidos mediante unidades de medición inercial y usando redes neuronales, específicamente un autocodificador de arquitectura profunda, se analizaron los datos en el dominio temporal.

Se demostró que los autocodificadores ofrecen ventajas para la detección de anomalías respecto a otros algoritmos de aprendizaje automático, siendo una de las principales ventajas el que pueden entrenarse de forma semi-supervisada, esto es, con datos de condición de operación normal exclusivamente. Esto facilita su implementación en

escenarios industriales, ya que en dichos escenarios generalmente se dispone mayormente de datos de condición normal de operación de los equipos y en mucho menor medida de datos de condición anormal de operación.

Se demostró asimismo que con el uso de autocodificadores de arquitectura profunda es innecesario un preprocesamiento extensivo de los datos antes de alimentarlos al autocodificador—sea para remover ruido, extraer componentes principales (Wu et al., 2017) o transformar los datos al dominio frecuencial (Long et al., 2020), (D. Zhang et al., 2019). Se comprobó inclusive que se pueden usar los datos en el dominio temporal sin preprocesar y aun así obtener un buen desempeño del algoritmo en las predicciones, habiendo aun lugar para mejoras.

Adicionalmente se demostró que los sensores y procesadores de bajo costo cumplen satisfactoriamente para la adquisición de señales vibratorias, lo que representa una alternativa al uso de equipos costosos (Wu et al., 2017) y muy sofisticados que eventualmente pueden resultar más caros que la propia impresora. De igual manera, el uso de software libre para la adquisición y análisis de los datos evitó depender de programas propietarios costosos, aprovechando el entorno de desarrollo en la nube con Google Colab, siendo innecesario utilizar equipo de cómputo de alta gama para el entrenamiento del autocodificador.

Por otro lado, se comprobó que la inteligencia artificial provee herramientas poderosas y fáciles de implementar que permiten sin mayor problema manejar grandes cantidades de datos—inclusive datos sin preprocesar para remover ruido, por ejemplo—y convertirlos en información en un proceso mucho más rápido y sencillo que con el uso de herramientas estadísticas tradicionales—redes bayesianas, por ejemplo. Esto resulta de gran valía para la toma de decisiones y acciones de una manera rápida y efectiva, sirviéndose de las tendencias previstas por el algoritmo y de esta manera limitar el impacto de los fallos en los equipos productivos.

Por último, esta metodología puede hacerse extensible a una gran cantidad de áreas industriales en las que actualmente resulta complicado hacer análisis vibracional, ya sea



porque los equipos son demasiado costosos, o porque se requiere personal altamente especializado para el manejo tradicional de los datos y su interpretación, o ambos.

## **7. RECOMENDACIONES**

Como trabajo futuro se recomienda la optimización de los hiperparámetros del autocodificador y agregar más capas ocultas y más neuronas a dichas capas y al espacio latente, así como la implementación de alguna técnica de regularización tal como la dilución (Dropout) para mejorar el desempeño del modelo propuesto.

Se recomienda asimismo el despliegue del sistema con cómputo de borde para la predicción de fallas en tiempo real.

## 8. REFERENCIAS

- Ahuett-Garza, H. y Kurfess, T. (2018). A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing. *Manufacturing Letters*, 15, 60–63. <https://doi.org/10.1016/j.mfglet.2018.02.011>
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., van Essen, B. C., Awwal, A. A. S. y Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3), 292. <https://doi.org/10.3390/electronics8030292>
- Becker, P., Roth, C., Roennau, A. y Dillmann, R. (2020). Acoustic Anomaly Detection in Additive Manufacturing with Long Short-Term Memory Neural Networks, 921–926.
- Borghesi, A., Bartolini, A., Lombardi, M., Milano, M. y Benini, L. (2019). Anomaly Detection Using Autoencoders in High Performance Computing Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 9428–9433. <https://doi.org/10.1609/aaai.v33i01.33019428>
- Brunton, S. y Kutz, N. (2021). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (2nd.). Cambridge University Press.
- Crowell, A. M., Cramer, S. A., Lopez, J. P., Rashka, E. K., Reuss, R. C., Wordsworth, V. K. y & Lewin, G. C. (2018). Development of a Quality Assurance Tool for Additive Manufacturing: University of Virginia, Charlottesville, Virginia, USA, 27 April 2018. *Systems and Information Engineering Design Symposium (SIEDS)*. <http://ieeexplore.ieee.org/servlet/opac?punumber=8370613>
- Descripción y funcionamiento del bus I2C*. <https://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>
- Eyers, D. y Dotchev, K. (2010). Technology review for mass customisation using rapid manufacturing. *Assembly Automation*, 30(1), 39–46. <https://doi.org/10.1108/01445151011016055>
- Fedorov, D., Ivoylov, A., Zhmud, V. y Trubin, V. (2015). Using of Measuring System MPU6050 for the Determination of the Angular Velocities and Linear Accelerations. *Automatics & Software Engineering*.(No. 1 (11)), 76–81.
- Flovik, V. (2018). *How to use machine learning for anomaly detection and condition monitoring* [Online]. Available: <http://towardsdatascience.com>
- Goodfellow, I., Bengio, Y. y Courville, A. (2016). *Deep learning*. MIT press.
- Huang, T. (2015). Quality control for fused deposition modeling based additive manufacturing: Current research and future trends.
- InvenSense Inc. (2013). *MPU6050 Product specification*. <https://invensense.tdk.com>

- Kim, C. (2018). Unobtrusive In Situ Diagnostics of Filament-Fed Material Extrusion Additive Manufacturing. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 8, 1469–1476.
- Larrañaga, P., Atienza, D., Diaz-Rozo, J., Ogbechie, A., Puerto-Santana, C. E. y Bielza, C. (2018). *Industrial Applications of Machine Learning. Chapman and Hall/CRC Data Mining and Knowledge Discovery Ser.* Chapman and Hall/CRC. <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5613453>
- LeCun, Y., Bengio, Y. y Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Li, Cabrera, D., Fernando, S., René-Vinicio, S., Cerrada, M., Long, J. y Valente de Oliveira, J. (2021). Fusing convolutional generative adversarial encoders for 3D printer fault detection with only normal condition signals. *Mechanical Systems and Signal Processing*, 147, 1–18. <https://doi.org/10.1016/j.ymsp.2020.107108>
- Li, W. (2016). Feature denoising and nearestfarthest distance preserving projection for machine fault diagnosis. *IEEE Transactions on Industrial Informatics*, 12, 393–404.
- Li, Y., Zhao, W., Li, Q., Wang, T. y Wang, G. (2019). In-Situ Monitoring and Diagnosing for Fused Filament Fabrication Process Based on Vibration Sensors. *Sensors (Basel, Switzerland)*, 19(11). <https://doi.org/10.3390/s19112589>
- Long, J., Sun, Z., Li, C., Hong, Y., Bai, Y. y Zhang, S. (2020). A Novel Sparse Echo Autoencoder Network for Data-Driven Fault Diagnosis of Delta 3-D Printers. *IEEE Transactions on Instrumentation and Measurement*, 69(3), 683–692. <https://doi.org/10.1109/TIM.2019.2905752>
- López Pinaya, W., Vieira, S., Garcia-Dias, R. y Mechelli, A. (2020). *Machine learning: Chapter 11. Autoencoders*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-815739-8.00011-0>
- Ma, M., Sun, C. y Chen, X. (2018). Deep Coupling Autoencoder for Fault Diagnosis with Multimodal Sensory Data. *IEEE Transactions on Industrial Informatics*, PP.
- Mhaskar, H. y Poggio, T. (2016, 10 de agosto). *Deep vs. shallow networks : An approximation theory perspective*. <http://arxiv.org/pdf/1608.03287v1>
- Nagpal, A. y Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications, 140–145. <https://doi.org/10.1109/AICAI.2019.8701341>
- Perez, A. (2014). Fracture surface analysis of 3d-printed tensile specimens of novel ABS-based materials, 343–353.
- Ponce-Cruz, P. (2010). *Inteligencia Artificial. Con Aplicaciones a la Ingeniería* (1a Edición). Alfaomega.

- Raschka, S. (2016). *Python machine learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics. Community experience distilled*. Packt Publishing open source.
- Sa'ude, N. (2013). Dynamic mechanical properties of copper-abs composites for FDM feedstock. *Int J Eng Res Appl* 3, 1257–1263.
- Stancin, I. (2019). An overview and comparison of free Python libraries for data mining and big data analysis, 977–982. <https://doi.org/10.23919/MIPRO.2019.8757088>
- Stuart, R. y Norvig, P. (2004). *Inteligencia Artificial. Un enfoque moderno* (2a. Edición). Pearson.
- Thompson, M. K., Moroni, G., Vaneker, T., Fadel, G., Campbell, R. I., Gibson, I., Bernard, A., Schulz, J., Graf, P., Ahuja, B. y Martina, F. (2016). Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints. *CIRP Annals*, 65(2), 737–760. <https://doi.org/10.1016/j.cirp.2016.05.004>
- Vos, K., Peng, Z., Jenkins, C., Shahriar, M. R., Borghesani, P. y Wang, W. (2022). Vibration-based anomaly detection using LSTM/SVM approaches. *Mechanical Systems and Signal Processing*, 169, 108752. <https://doi.org/10.1016/j.ymsp.2021.108752>
- Wu, H. (2016). A New Approach for Online Monitoring of Additive Manufacturing Based on Acoustic Emission. Publicación en línea avanzada. <https://doi.org/10.1115/MSEC2016-8551>
- Wu, H., Wang, Y. y Yu, Z. (2015). In situ monitoring of FDM machine condition via acoustic emission. *The International Journal of Advanced Manufacturing Technology*, 209(15–16), 5593. <https://doi.org/10.1007/s00170-015-7809-4>
- Wu, H., Yu, Z. y Wang, Y. (2017). Real-time FDM machine condition monitoring and diagnosis based on acoustic emission and hidden semi-Markov model. *The International Journal of Advanced Manufacturing Technology*, 90(5-8), 2027–2036. <https://doi.org/10.1007/s00170-016-9548-6>
- Yang, Z [Zhe], Gjorgjevikj, D., Long, J., Zi, Y., Zhang, S. y Li, C. (2021). Sparse Autoencoder-based Multi-head Deep Neural Networks for Machinery Fault Diagnostics with Detection of Novelties. *Chinese Journal of Mechanical Engineering*, 34(1). <https://doi.org/10.1186/s10033-021-00569-0>
- Yang, Z [Zhensheng], Jin, L., Yan, Y. y Mei, Y. (2018). Filament Breakage Monitoring in Fused Deposition Modeling Using Acoustic Emission Technique. *Sensors (Basel, Switzerland)*, 18(3). <https://doi.org/10.3390/s18030749>
- Yoon, J., He, D. y Van Hecke, B. (2014). A PHM Approach to Additive Manufacturing Equipment Health Monitoring, Fault Diagnosis, and Quality Control.

- Zareiyan, B. y Khoshnevis, B. (2017). Effects of interlocking on interlayer adhesion and strength of structures in 3D printing of concrete. *Automation in Construction*, 83, 212–221. <https://doi.org/10.1016/j.autcon.2017.08.019>
- Zhang, D., Li, Z., Qin, S. y Han, S. (2019). Optimization of Vibration Characteristics of Fused Deposition Modeling Color 3D Printer Based on Modal and Power Spectrum Method. *Applied Sciences*, 9(19), 4154. <https://doi.org/10.3390/app9194154>
- Zhang, S. (2019). Dynamic condition monitoring for 3D printers by using error fusion of multiple sparse auto-encoders. *Computers in Industry*, 105, 164–176. <https://doi.org/10.1016/j.compind.2018.12.004>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. y Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, 213–237. <https://doi.org/10.1016/j.ymsp.2018.05.050>

