

Universidad Internacional de la Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Predicción de la calidad de impresión en base a monitorización in-situ

Trabajo Fin de Máster

presentado por: Alberto Rodríguez Álvarez

Dirigido por: Antoni Manel Ferragut Amengual

Ciudad: Oviedo

Fecha: 22 Julio 2021

Resumen

En este trabajo se ha creado un modelo de clasificación capaz de diferenciar entre los distintos modos de fusión producidos durante la impresión de piezas metálicas en una máquina de fabricación aditiva. Como información de entrada del modelo se utilizan los datos generados por un sistema de monitorización coaxial óptico. Se ha creado un dataset propio en base a los datos de monitorización capturados durante la impresión de 40 cubos diseñados para generar los diferentes modos de fusión estudiados. Para el tratamiento de los datos, se ha desarrollado una metodología de preprocesado en base a comparativas entre diferentes técnicas. Además, se ha comparado el desempeño de diferentes técnicas de reducción de dimensionalidad entre las que se encuentran autoencoders, SVD (singular value decomposition) y extracción manual de características estadísticas. El modelo final consigue una tasa de acierto superior al 86,1 % en todas las clases, demostrando así la viabilidad del uso de este modelo para detectar la calidad de piezas impresas en SLM.

Palabras Clave: Fabricación aditiva, Reducción de dimensionalidad, Clasificación multiclase, Clasificación de series temporales, Control de calidad.

Abstract

This work presents a multilabel classification model able to distinguish between different fusion modes found during printing of metal additive manufacturing parts. The input data for the model is captured with a coaxial optical monitoring system. A dataset has been created based on 40 metal cubes were printed and monitored with different induced fusion modes. A preprocessing methodology is selected by comparing the results from different techniques. After which different dimensionality reduction techniques are applied and compared. This techniques include autoencoders, SVD (singular value decomposition) and manual extraction of statistical components. The final model achieves at least 86,1% accuracy on all classes and shows the feasibility of using this methodology to detect part quality on an SLM machine.

Key Words: Metal additive manufacturing, Dimensionality reduction, Multiclass classification, Temporal series classification, Quality assurance.

Índice de Contenidos

Resumen	I
Abstract	II
Índice de Ilustraciones	VI
Índice de Tablas	VIII
1. Introducción	1
2. Contexto y Estado del Arte	6
2.1. Fabricación aditiva	6
2.1.1. Tecnología de Powder Bed Fusion	6
2.1.2. Materiales	9
2.1.3. Parámetros de proceso en SLM	11
2.2. Monitorización	13
2.3. Inteligencia artificial en Fabricación aditiva	15
2.4. Detección de defectos y predicción de calidad en sistemas de SLM	19
2.4.1. Monitorización acústica	19
2.4.2. Monitorización óptica	20
2.5. Clasificación de series temporales	21
2.6. Aportación	23
3. Objetivos del trabajo	25
3.1. Objetivo general	25
3.2. Objetivos específicos	25

4. Metodología y Descripción detallada del experimento	27
4.1. Sistema de monitorización	27
4.2. Captura de los datos del dataset	29
4.3. Preprocesamiento de datos	34
4.3.1. Segmentación	35
4.3.2. Eliminación de los contornos	36
4.3.3. Calibración de los fotodiodos	38
4.4. Procesamiento de datos	39
4.4.1. Segmentación de los cubos	41
4.4.2. Igualación de longitudes	42
4.4.3. Reducción de dimensionalidad y extracción de características	43
4.5. Clasificación	47
5. Descripción de los resultados y Discusión	50
5.1. Impresión de los cubos	50
5.2. Caracterización de los cubos	51
5.3. Preprocesado y almacenamiento	55
5.4. Segmentación a diferentes escalas e igualación de las longitudes	56
5.5. Reducción de la dimensionalidad	57
5.5.1. Autoencoders	58
5.6. Comparación entre datasets	60
5.6.1. Comparación a la escala de los cubos	60
5.6.2. Comparación a la escala de las capas	62
5.6.3. Comparación a la escala de los vectores de escaneo	64
5.6.4. Comparación entre métodos de igualación de longitudes	65
5.6.5. Comparación entre métodos de reducción de dimensionalidad	66
5.6.6. Comparación entre técnicas de clasificación para la escala de los cubos	66
5.6.7. Comparación entre escalas	68
5.6.8. Comparación entre validaciones cruzadas	68
5.7. Modelo final y discusión	69
5.7.1. Modelo final	70
6. Conclusiones y Trabajo Futuro	76
6.1. Trabajo futuro	77

7. Bibliografía

79

Índice de Ilustraciones

2.1. Esquema de funcionamiento de una máquina de EBM [Additively, 2021]. . .	7
2.2. Esquema del funcionamiento de una máquina de SLM [Additively, 2021]. . .	8
2.3. Esquema de funcionamiento de atomizador de gas [Carpenter Additive, 2020].	10
2.4. División de los datos involucrados en fabricación aditiva por [Meng, 2020]. .	19
2.5. Taxonomía de los acercamientos para clasificación de series temporales ba- sados en aprendizaje profundo [Fawaz, 2019].	23
4.1. Esquema del tren óptico de medición [Renishaw, 2021].	29
4.2. Distribución geométrica de cubos de alto keyhole y alta falta de fusión (Elaboración propia).	33
4.3. Distribución geométrica de cubos de keyhole, fusión nominal y falta de fusión (Elaboración propia).	33
4.4. Intensidad medida por el fotodiodo MeltVIEW visible. Los datos en bruto fueron transformados a imagen utilizando la información espacial contenida en el CSV (Elaboración propia).	39
4.5. Intensidad medida por el fotodiodo MeltVIEW visible tras la calibración de los fotodiodos (Elaboración propia).	40
4.6. Visualización de la reducción de dimensionalidad en base a la descomposi- ción SVD [Albatayneh et al., 2014].	44
4.7. Esquema general de un Autoencoder [Sharma, 2021].	46
5.1. Imagen tomada durante la impresión de los cubos. Resaltados en rojo son los 2 cubos que tuvieron que ser desactivados para completar la impresión (Elaboración propia).	51
5.2. Micrografía del cubo 15 correspondiente a la clase alto keyhole (Elaboración propia).	54

5.3. Micrografía del cubo 19 correspondiente a la clase alta falta de fusión (Elaboración propia).	54
5.4. Micrografía del cubo 7 correspondiente a la clase Nominal (Elaboración propia).	55
5.5. Arquitectura modelo utilizado para la comparación de datasets en la escala del cubo (Elaboración propia).	61
5.6. Entrenamiento del mejor modelo a la escala del cubo (Elaboración propia).	62
5.7. Arquitectura modelo utilizado para la comparación de los datasets en la escala de las capas (Elaboración propia).	62
5.8. Entrenamiento del mejor modelo a la escala de las capas (Elaboración propia).	63
5.9. Arquitectura modelo utilizado para la comparación de los datasets en la escala de los vectores de escaneo (Elaboración propia).	64
5.10. Entrenamiento del mejor modelo a la escala de los vectores de escaneo (Elaboración propia).	65

Índice de Tablas

4.1. Combinaciones de potencia y velocidad. V_n se corresponde con las 3 velocidades diferentes utilizadas y P_n lo mismo para las potencias. V_n y V_n^* son aproximadamente iguales pero las velocidades V_n se obtuvieron variando la distancia entre puntos de exposición y V_n^* variando el tiempo de exposición.	31
4.2. Número de cubos impresos para cada set de parámetros.	32
4.3. Factores de corrección $N_{l,f}$ de la ecuación 4.1.	39
5.1. Densidades de los cubos impresos. HKH: alto keyhole, KH: keyhole, N: nominal, LF: falta de fusión, HLF: alta falta de fusión.	53
5.2. Combinación entre escala e igualación de longitud	57
5.3. Combinaciones entre técnicas de reducción de dimensionalidad y segmentaciones.	58
5.4. Valores de las métricas de reconstrucción MAE sobre el conjunto de test de las tres series temporales y los tres tipos de autoencoders	59
5.5. Métricas F1 de los modelos entrenado con cada uno de los datasets de la escala del cubo	61
5.6. Métricas F1 de los modelos entrenados con cada uno de los datasets de la escala de las capas	63
5.7. Métricas F1 de los modelos entrenados con cada uno de los datasets de la escala de los vectores de escaneo	64
5.8. Media de las puntuaciones F1 de validación cruzada de los datasets que se corresponden con la escala y método de igualación de longitud marcados.	65
5.9. Puntuación F1 de cada uno de los métodos aplicados al mejor dataset de la escala de los cubos.	67

5.10. Comparación de random forest con NN para las escalas de las capas y de los vectores de escaneo	68
5.11. Comparación entre las métricas F1 de los mejores datasets de cada una de las escalas.	68
5.12. Comparación del mismo modelo con validación cruzada por cubo y validación cruzada aleatoria en el dataset S9 SC	69
5.13. Matriz de confusión del modelo final sobre el conjunto de test	71
5.14. Matriz de confusión confeccionada a partir de las matrices de confusión de todos los folds de validación cruzada.	72
5.15. Matriz de confusión del modelo final sobre el conjunto de test sin keyhole	73
5.16. Matriz de confusión confeccionada a partir de las matrices de confusión de todos los folds de validación cruzada para el nuevo dataset.	74

Capítulo 1

Introducción

La fabricación aditiva, también conocida como impresión 3D, es una técnica de fabricación basada en añadir material, normalmente capa a capa, para crear una pieza siguiendo un modelo 3D [ASTM, 2015]. La primera tecnología de impresión 3D en salir al mercado fue la estereolitografía en la década de los 80. En sus inicios se concibió como una tecnología de prototipado rápido y estaba limitada al uso de plásticos y polímeros. En los años 90 se desarrollaron técnicas capaces de crear piezas metálicas, las cuales tenían un potencial industrial mucho mayor [Wohlers et al., 2020]. En los últimos años la fabricación aditiva de metal ha recibido un gran interés por parte de la comunidad científica e industrial gracias a sus múltiples virtudes, como pueden ser la alta libertad de diseño, rapidez de fabricación o alto nivel de personalización [Frazier, 2014]. Sin embargo, a pesar todas sus ventajas, la adopción de esta tecnología por parte de ciertas industrias no está siendo todo lo rápida que cabría esperar en base a sus virtudes. Uno de los principales problemas que impide que ciertos sectores adopten la fabricación aditiva de forma plena es la falta de estabilidad y repetibilidad que sufren la mayoría de tecnologías de fabricación aditiva [Meng, 2020]. Aunque se ha avanzado mucho al respecto, en la mayoría de los casos, la fiabilidad de una pieza fabricada por procesos aditivos no siempre alcanza los estándares marcados por industrias como la aeroespacial y aeronáutica.

Intentando dar solución a este problema han surgido multitud de metodologías, desde la estandarización del proceso y certificación de piezas hasta los ensayos mecánicos destructivos. En los últimos años, una de las soluciones que más fuerza está tomando, es la monitorización del proceso. Consiste en recoger datos durante el proceso de impresión con el objetivo de detectar los defectos según van surgiendo. Esta es la línea de investigación

que se seguirá en este trabajo.

En concreto se utilizarán datos provenientes de una máquina de impresión 3D de metal. Dentro de las distintas tecnologías de fabricación aditiva, esta impresora se engloba en la denominada fusión de cama de polvo por láser (SLM). En esencia el proceso se basa en extender finas capas de polvo metálico que posteriormente son fundidas de forma selectiva con un láser de alta potencia formando así una pieza con la geometría deseada. La monitorización utilizada en este trabajo consiste en un sistema de fotodiodos coaxiales que captan las emisiones ópticas derivadas del proceso de fusión por láser.

Los datos procedentes de este sistema de monitorización se corresponden con la intensidad lumínica emitida por la zona de metal fundido, medida en varios rangos de longitudes de onda diferentes. Estos datos son capturados a 100 KHz de forma nativa, pero son pre-procesados por el software del fabricante de forma que la señal no tiene una frecuencia de muestreo fija. En general los datos extraídos tienen una frecuencia del entorno a 10 KHz y están agrupados en lo que el fabricante denomina puntos de exposición. Los datos extraídos están en forma de serie temporal con una tasa de muestreo variable. Además de los datos de intensidad, el sistema está sincronizado con la posición del láser sobre la cama de polvo, por lo que los datos también se pueden explorar de forma geométrica en forma de imagen 2D o 3D. Esta característica será clave para la segmentación de los datos y permitirá etiquetarlos correctamente.

El principal objetivo de este trabajo es utilizar los datos de monitorización que se acaban de describir para determinar la calidad de impresión. La calidad de una pieza metálica impresa en 3D es un concepto complejo de definir. Existen multitud de aspectos que afectan a la calidad, desde el acabado superficial de la pieza hasta las propiedades mecánicas de la misma. Uno de los indicadores más estudiados, por su importancia, es la porosidad de la pieza. Su importancia radica en que es condición necesaria para la obtención de otras características deseables, como son unas buenas propiedades mecánicas.

Durante el proceso de fusión del polvo metálico es relativamente común, sobre todo si no se utilizan los parámetros adecuados, que se genere porosidad en la pieza. De forma simplificada, existen dos procesos principales que generan porosidad en las máquinas de SLM, los denominados “keyhole” y “lack of fusion” o falta de fusión. El primero se produce cuando se aporta una densidad de energía demasiado elevada a la cama de polvo generando un proceso de fusión inestable que puede atrapar burbujas de gas durante la solidificación del metal. El segundo mecanismo se da generalmente cuando hay una carencia de energía

depositada, lo cual puede provocar que haya zonas del interior de la pieza que no se lleguen a fundir y por tanto que se generen poros.

Dado que la densidad de la pieza está íntimamente relacionada con estos dos procesos de fusión no deseados, será el modo de fusión lo que se utilizará como etiqueta para la creación del dataset. Por modo de fusión se entienden los mecanismos que se acaban de introducir y que son los causantes de la porosidad, es decir, se distinguirá entre una fusión nominal u óptima, una fusión que produce porosidad por “keyhole” y una fusión con “lack of fusion”. Para la creación de este dataset, se va a realizar un diseño de experimentos ad-hoc, en el que se imprimirán cubos con parámetros de impresión que fueren los modos de fusión que se acaban de describir.

Desde el punto de vista de la creación del dataset, el primer paso consistirá en un preprocesamiento y limpiado de los datos de monitorización brutos. En primer lugar se deben organizar los datos por piezas ya que es esta la forma en la que están agrupadas las etiquetas. De forma nativa, los datos se extraen del software del fabricante en forma de series temporales separadas por capa y láser. En este formato, no se pueden diferenciar los datos provenientes de cada una de las piezas impresas en la misma cama de polvo. Si en cada impresión solo se imprimiese una pieza, este paso se podría omitir, pero para optimizar costes y tiempo, siempre se imprimirán varias piezas en la misma impresión. Además, en general cada pieza se imprimirá con más de un láser por lo que también se deben combinar los datos de todos los láseres.

Para realizar esta segmentación se optó por el método que se cree que menos intervención humana requiere. Se comienza pasando los datos de cada una de las capas de su formato original, en forma de serie temporal, a una representación tri-dimensional. Para ello se partirá el espacio de impresión en voxels cúbicos y se recorrerán las series temporales contando cuantos puntos de exposición caen dentro de cada voxel. De esta forma se obtiene un array de tres dimensiones con ceros en todas las posiciones sobre las que ningún láser ha incidido y un valor positivo donde si ha pasado alguno de los cuatro láser. En este proceso es de suma importancia el tamaño de los voxels, ya que si fuesen muy pequeños, las piezas no serían conexas, y si fuesen demasiado grandes, piezas distintas podrían juntarse. Una vez se tiene esta representación tridimensional de las zonas que han sido fundidas, se aplica un algoritmo de detección de componentes conexas, de forma que cada una de las componentes se corresponderá con una pieza. Por último, utilizando la máscara obtenida por el algoritmo de componentes conexas, se filtran los datos originales

creando una serie temporal por cada pieza.

Tras la segmentación de los cubos, se debe realizar un proceso de limpieza en el que se eliminarán los datos referentes a los contornos. Por norma general, los contornos de las piezas se suelen imprimir con parámetros diferentes a los utilizados en las zonas interiores de la pieza. Esto provoca que el modo de fusión obtenido en estas zonas de la pieza no sea el mismo, por lo que mantener los contornos en el dataset solo introduciría ruido.

Como último paso de preprocesamiento, se llevará a cabo un proceso de calibración de las medidas obtenidas por distintos láseres. Este paso es necesario, ya que al ser medidas tomadas por sensores equivalentes, pero no iguales, puede existir una cierta diferencia entre las medidas de uno y otro. Por ese motivo, se calcularán y aplicarán los coeficientes correctores que hacen que las medidas sean comparables entre sí.

Este preprocesamiento se realizará igual sobre todos los datos de monitorización. Sin embargo, a partir de este punto se van a explorar diferentes vías de transformación de los datos para encontrar la que mejor se ajuste a la tarea que se pretende resolver.

Las series temporales que se generan, tras el preprocesamiento descrito, tienen una longitud del orden de 25 millones de valores. Esta cantidad de datos asociada a una sola etiqueta no es factible utilizarla directamente en un modelo de aprendizaje automático, sino que se requiere algún procesamiento de los datos que disminuya el tamaño de los mismos. En este trabajo se va a implementar uno de los enfoques más utilizados en estos casos que es la reducción de dimensionalidad. Consiste en encontrar una representación de los datos que resuma lo mejor posible la información contenida en ellos pero con tamaño mucho menor. Se van a probar diferentes técnicas de reducción de dimensionalidad con el fin de encontrar cual es la más adecuada para la tarea tratada. Se utilizarán extracción manual de características, autoencoders de diferentes tipos y un método basado en la descomposición de valores singulares de matrices. En cuanto a los autoencoders, se van a implementar autoencoders basados en redes neuronales clásicas, autoencoders convolucionales y autoencoders recurrentes con LSTM.

Algunos de los métodos mencionados de reducción de dimensionalidad requieren que todas las series del dataset tengan la misma longitud por lo que también se implementarán tres métodos de igualación de longitudes diferentes, under sampling, truncado y padding.

Por otro lado, se va a explorar el aumento de la cantidad de ejemplos de entrenamiento por medio de la partición las series temporales en segmentos disjuntos. Esto permite aumentar la cantidad de datos disponibles en los dataset, que dado lo costoso de los experi-

mentos en el mundo de la fabricación aditiva, puede suponer una gran ventaja. Se probará a segmentar las series a tres niveles distintos y se estudiará cual tiene mejor desempeño.

Los tres pasos que se acaban de mencionar, partición de las series, igualación de longitudes y reducción de dimensionalidad, se combinarán entre ellos de forma que todas las combinaciones posibles sean exploradas, tras lo cual se llevará a cabo un estudio exhaustivo para determinar cuál de todas estas combinaciones produce los mejores resultados.

Una vez seleccionada la mejor metodología se procederá a crear una red neuronal de clasificación multiclase. Para ello se hará una búsqueda exhaustiva de los hiper-parámetros de la red que mejores resultados produzcan y posteriormente un estudio de los resultados obtenidos junto con comparaciones con soluciones encontradas en bibliografía.

El trabajo se va a estructurar de la siguiente manera: en el capítulo 2 se hará una revisión del estado del arte y se resumirá el contexto en el que se desarrolla el trabajo. En el capítulo 3 se presentarán los objetivos principales así como los secundarios que se pretenden abordar. En el capítulo 4 se detallará la metodología seguida en los experimentos así como las técnicas aplicadas para el tratamiento y modelado de los datos. En el capítulo 5 se presentarán los resultados obtenidos por los experimentos expuestos en el capítulo anterior así como una discusión sobre los mismos. Por último en el capítulo 6 se presentarán las conclusiones del trabajo y las posibles ampliaciones que surgen de lo descubierto durante la realización del trabajo.

Capítulo 2

Contexto y Estado del Arte

2.1. Fabricación aditiva

La ASTM define la fabricación aditiva como el proceso de unir materiales para formar objetos en base a modelos 3D, usualmente capa a capa y de forma contrapuesta a la fabricación substractiva [ASTM, 2015]. A su vez, la ASTM define 7 categorías de procesos de fabricación aditiva: Binder Jetting, Direct Energy Deposition, Material Extrusion, Material Jetting, Powder Bed Fusion, Sheet Lamination y Vat Photopolymerization. A pesar de que hay tecnologías que no encajan en ninguna de las categorías anteriores, esta es una de las clasificaciones más aceptadas dentro de la industria [Kumar, 2020] .

2.1.1. Tecnología de Powder Bed Fusion

Este trabajo se basará en el proceso denominado “Powder Bed Fusion” (PBF), que la ASTM define como el proceso de aplicar energía térmica de forma selectiva para fundir regiones de una cama de polvo [ASTM, 2015]. Dentro de los procesos de PBF existen diferentes modalidades que se distinguen principalmente en función del tipo de material, metal o polímero y la fuente de energía que utilizan, láser o haz de electrones. De esta forma se distinguen 3 procesos, fusión de metal por láser (SLM), fusión de polímero con láser (SLS) y fusión de metal por haz de electrones (EBM) [Kumar, 2020].

El proceso de PBF es intrínsecamente iterativo, primero se extiende una fina capa de polvo, para después fundirla de forma selectiva con la fuente de energía térmica. Tras esto, el proceso se repite extendiendo una nueva capa de polvo sobre la anterior. Para llevar a cabo este proceso existen una serie de sistemas básicos que todas las impresoras integran.

El primero es la fuente de energía y la forma de dirigirla. En el caso de las máquinas

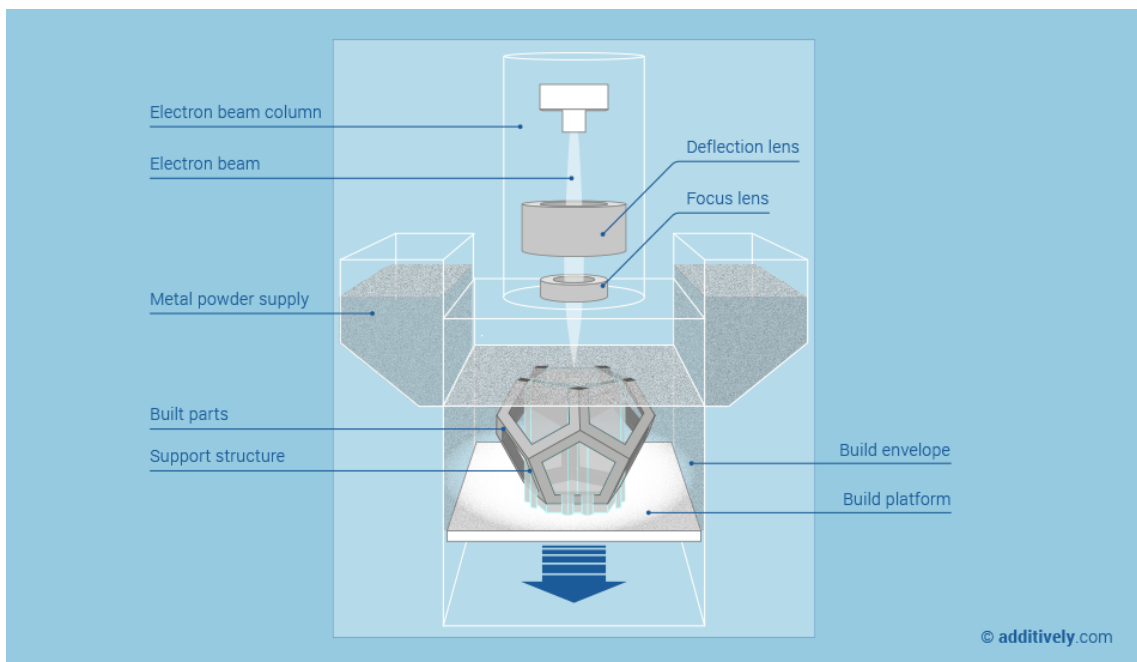


Figura 2.1: Esquema de funcionamiento de una máquina de EBM [Additively, 2021].

basadas en un haz de electrones, la fuente de electrones se coloca directamente encima de la cama de polvo apuntando hacia ella. El haz proveniente de la fuente es refinado y controlado por medio de campos magnéticos que actúan como lentes para electrones. Así es posible controlar de forma rápida y precisa el lugar en el que incide el haz. Por otro lado, las máquinas basadas en fusión por láser utilizan ópticas y espejos galvanométricos para controlar el punto donde incide el láser. Por lo general las piezas metálicas creadas con máquinas de láser superan en calidad superficial y precisión geométrica a las producidas con haz de electrones, sin embargo el proceso con haz de electrones es más rápido y produce menos tensiones residuales [Wohlers et al., 2020].

El segundo sistema necesario para los procesos basados en cama de polvo es el conocido como “recoating”. Este proceso es el encargado de extender las finas capas de polvo que luego serán fundidas. Típicamente este sistema debe ser capaz de extender una capa del orden de las $50 \mu m$ de polvo de forma homogénea en toda la extensión de la cama. Para ello existen diferentes métodos, uno de los más utilizados es el uso de una cuchilla que empuja una pequeña montaña de polvo de forma paralela a la cama dejando la altura deseada sobre la capa anterior. Otro método sustituye el filo de la cuchilla por una tira de goma blanda. De esta forma se adapta mejor a las imperfecciones de la capa anterior y el

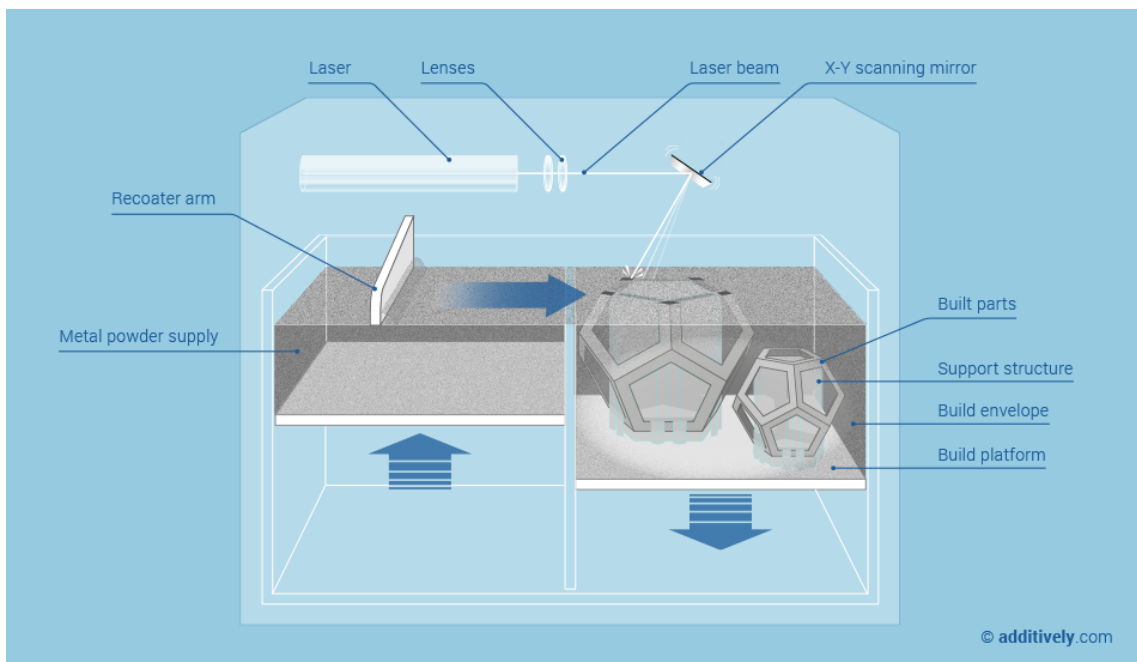


Figura 2.2: Esquema del funcionamiento de una máquina de SLM [Additively, 2021].

proceso se hace más robusto a pequeñas imperfecciones que puedan sobresalir de la capa de polvo.

El tercer sistema presente en la mayoría de máquinas de PBF es el sistema de inertización de la cámara de proceso y de flujo de gas. Este sistema es crítico en los procesos de metal, ya que sin una atmósfera inerte el metal fundido reaccionaría químicamente con la atmósfera degradando la composición y propiedades del material. Es por esto que en SLM se sustituye la atmósfera interna por un gas inerte como puede ser el nitrógeno o el argón. En el caso de EBM, la sustitución de la atmósfera por un gas inerte no es una opción, ya que los electrones necesitan vacío para poder moverse de forma eficiente. Además de la inertización de la cámara, en SLM se suele incluir un sistema de gas inerte que genera un flujo laminar de gas paralelo a la cama de polvo. Este sistema de gas sirve para eliminar los productos generados por el proceso de fusión, generalmente gases y partículas de polvo que son propulsadas por esos gases y que podrían depositarse sobre la cama de polvo degradando su calidad.

Por último está el sistema de pistones y base de impresión que permite mover la cama de polvo en su totalidad hacia abajo para construir la siguiente capa encima. Cada vez que se va a extender una nueva capa de polvo, el sistema de pistones mueve la cama de

polvo hacia abajo justo el grosor de la siguiente capa. Este sistema simplifica enormemente el sistema de recoating que así no debe variar en altura y el tren óptico de la fuente de energía, que no necesita variar su enfoque cada vez que se cambia de capa.

2.1.2. Materiales

Un aspecto de gran importancia en todas las tecnologías de fabricación aditiva es la materia prima que se utiliza. En los procesos de fabricación aditiva de metal, se puede encontrar la materia prima en tres posibles formatos, hilo, polvo y láminas. En el caso de las tecnologías de PBF, como su propio nombre indica, se trata con polvo metálico. La producción de este polvo metálico es un proceso complejo que se explicará de forma resumida a continuación.

Actualmente, existen tres procesos principales para producir polvo metálico: atomización por plasma, atomización por gas y atomización por agua; siendo la versión de gas la más utilizada por su buen balance entre calidad y precio [Wohlers et al., 2020]. La variante de plasma es la que mejor calidad final del polvo ofrece, mientras que la atomización por agua es la más barata de las tres.

De forma simplificada, el proceso de atomización por gas consiste en romper una columna de metal fundido con un chorro de gas a alta presión. La interacción del gas con el metal fundido genera pequeñas gotas metálicas que deben solidificarse antes de llegar al fondo del atomizador donde son recogidas. El tiempo que pasa desde la creación de las gotas hasta su solidificación tiene un gran impacto en la morfología final de las mismas, ya que un tiempo muy corto generaría partículas amorfas que se han solidificado antes de alcanzar su forma más estable y un tiempo muy largo aumenta las probabilidades de impactos entre partículas sin solidificar que se quedarían pegadas formando partículas con satélites. Al igual que el proceso de impresión, el proceso de atomización también se lleva a cabo en una atmósfera inerte que protege la composición química del material.

La calidad del polvo utilizado para la impresión es un factor determinante de la calidad final de la pieza, es por esto que se presta gran atención a su evaluación. Existen multitud de factores que determinan la calidad del polvo, sin embargo los que se consideran más influyentes son: la distribución de tamaño de partículas, fluidez, factor de empaquetamiento y composición química [Alfaify, 2019].

La distribución de tamaño de partículas hace referencia a la cantidad de partículas de cada tamaño que hay en un determinado polvo. Generalmente en SLM se utilizan

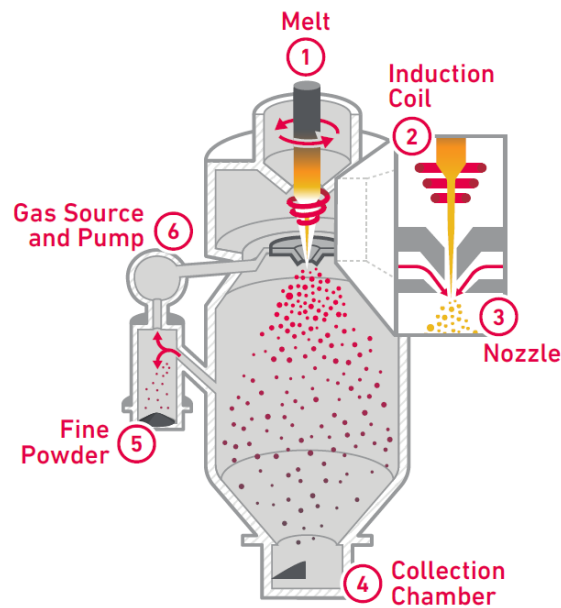


Figura 2.3: Esquema de funcionamiento de atomizador de gas [Carpenter Additive, 2020].

distribuciones de tamaño de partículas entre $20 \mu m$ y $60 \mu m$. El valor máximo debe ser siempre menor que la altura de capa que se vaya a utilizar, ya que de lo contrario se tendrían partículas más grandes que las propias capas. El valor mínimo está muy ligado a la fluidez del polvo, ya que las partículas de pequeño tamaño repercuten negativamente en la fluidez del polvo.

La fluidez del polvo se refiere a la facilidad del mismo para fluir. Esta característica es de suma importancia en los procesos de cama de polvo debido a la forma en que se extienden las capas. Si el polvo no fluye de forma adecuada, el proceso de recoating no conseguirá depositar capas homogéneas. La fluidez está afectada por muchos factores, entre los que se encuentran, la distribución de tamaño de partículas, la morfología de las partículas individuales y la humedad del polvo.

El factor de empaquetamiento hace referencia al porcentaje del espacio que es capaz de rellenar el polvo sin compactar. Este factor tiene relevancia ya que al fundir el polvo el metal resultante tendrá un volumen menor al que ocupaba el polvo, por lo que cuanto menor sea esta diferencia de volúmenes, más estabilidad tendrá el proceso.

Por último, es importante hablar de la composición química del polvo. Existen muchas aleaciones metálicas distintas disponibles para máquinas de PBF, sin embargo no todas se comportan de la misma forma, ni son capaces de producir piezas de la misma calidad,

por ello es muy importante seleccionar la composición adecuada. Según [Senvol, 2021] la lista de polvos metálicos diferentes que se pueden comprar a día de hoy ronda los 800. [Wohlers et al., 2020] considera que las familias más importantes de aleaciones son: acero de herramientas, acero inoxidable, titanio puro, aleaciones de titanio, aleaciones de aluminio, superaleaciones basadas en níquel, aleaciones cobalto-cromo, aleaciones de cobre, oro, plata, platino, paladio, tántalo. En este trabajo se tratará siempre con aceros, más concretamente con M300 un acero martensítico.

2.1.3. Parámetros de proceso en SLM

Al igual que en otros procesos de fabricación clásicos, en SLM existen múltiples parámetros de proceso que deben ser cuidadosamente ajustados para obtener la mayor calidad posible. Dado que este proceso tiene ya una cierta madurez, existe bastante consenso sobre cuales son los parámetros más relevantes del proceso [Kumar, 2020]. A continuación se hará un breve introducción de los 4 más relevantes.

Potencia del láser

La potencia del láser es probablemente el parámetro más importante, en torno al cuál se ajustan el resto de parámetros. La potencia del láser controla la cantidad de energía térmica que se le transfiere a la cama de polvo por unidad de tiempo. Por norma general, las impresoras comerciales de SLM llevan incorporados láser con potencias máximas que rondan los 400W [Metel et al., 2018]. La potencia máxima que es capaz de alcanzar una máquina suele ser el factor limitante para la obtención de mayores tasas de fabricación, ya que por lo general el resto de parámetros se pueden ajustar para acomodar una potencia de láser mayor.

Velocidad de escaneo

La velocidad de escaneo hace referencia a la rapidez con la que el haz láser se desplaza por la superficie de la cama de polvo. De los cuatro factores que se describen en esta sección, la velocidad y la potencia del láser son los dos parámetros que más influyen en la forma y tamaño que tiene la “Melt pool” (zona de metal fundido que se forma en la zona de incidencia del láser). La forma y tamaño de la “Melt pool” es un factor muy importante en los procesos de SLM y está muy relacionado con la porosidad de la pieza final. Para estudiar cómo las distintas combinaciones de potencia y velocidad afectan a la “Melt pool”

se realizan las denominadas “single tracks”. Se trata de experimentos en los que se imprimen una línea recta sobre una única capa de polvo, de forma que se elimina la dependencia con la separación entre vectores de escaneo. Por lo general se suelen distinguir tres tipos de resultados de una “single track” o modos de fusión, fusión por conducción, falta de continuidad y “Melt pool” en forma de “keyhole” [Tang et al., 2017]. El resultado ideal es la fusión por conducción, en la que la “single track” es totalmente continua y uniforme y la profundidad de la “Melt pool” es similar a su anchura. El caso de la “Melt pool” en forma de “keyhole” se produce cuando se deposita una energía excesiva por unidad de longitud. El “keyhole” se caracteriza por una profundidad de “Melt pool” considerablemente superior a la anchura de la misma. Para disminuir esta densidad de energía lineal se debe reducir la potencia del láser o aumentar la velocidad de escaneo. Por último, cuando la “single track” carece de continuidad se suele deber a una falta de densidad lineal de energía que provoca que no se funda suficiente material y que por la tensión superficial del metal se formen bolas.

Separación entre vectores de escaneo

A la hora de imprimir una pieza capa a capa, se realiza el conocido como proceso de “slicing”, que consiste en representar una pieza 3D a base de cortes paralelos entre sí. De forma análoga al “slicing”, se realiza el proceso de definición de la trayectoria de escaneo, que no es más que el camino que debe seguir el láser para cada capa. Sin entrar en mucho detalle sobre las estrategias de escaneo, por norma general las piezas se imprimen a base de líneas rectas paralelas entre sí que cubren toda la pieza. Estas líneas rectas son los denominados vectores de escaneo y la separación entre dos contiguos es el tercer factor que debe ajustarse en SLM. La distancia entre vectores suele ajustarse para que exista solape entre las zonas fundidas por cada uno de ellos. Este solape es necesario para asegurar que no haya huecos entre las zonas fundidas.

Altura de capa

La altura de capa representa el grosor de las capas de polvo. De la misma forma que la separación entre vectores debía ajustarse para que exista solape entre ellos, la altura de capa debe ajustarse en consonancia con los demás parámetros para asegurar la adherencia entre capas. La altura de capa es un parámetro muy relevante a la hora de intentar aumentar la tasa de fabricación ya que hay muchos procesos de la impresión que

suponen un alto porcentaje del tiempo de impresión y que deben repetirse para cada capa. A su vez la altura de capa determina la precisión geométrica en la dirección z de la pieza.

Caso de láser pulsado

Los cuatro parámetros descritos hasta ahora se corresponden con la variante de SLM en que se utiliza un láser continuo. En la actualidad existen multitud de impresoras que utilizan láser pulsado como fuente de energía térmica. La principal novedad que introduce el láser pulsado reside en la forma de definir la velocidad de escaneo y en la forma de recorrer los vectores de escaneo. La velocidad de escaneo para un láser pulsado, se descompone en 3 nuevos parámetros, tiempo de exposición, distancia entre puntos de exposición y tiempo entre puntos [Alfaify, 2019]. Además los sistemas con láser pulsado no recorren los vectores de escaneo de forma continua, sino que realizan una serie de puntos de exposición discretos a lo largo del vector. El tiempo de exposición controla la cantidad de tiempo que el láser expondrá cada uno de los puntos de exposición. El tiempo entre puntos se corresponde con la cantidad de tiempo que transcurre entre dos puntos de exposición durante el cual el láser está apagado y se desplaza de un punto al siguiente. Por ultimo la distancia entre puntos es, como su nombre indica, la separación entre dos puntos de exposición consecutivos.

2.2. Monitorización

A pesar de los grandes avances realizados en los últimos años en fabricación aditiva, aún existen problemas de repetibilidad en los procesos que impiden su plena adopción en sectores en los que los fallos mecánicos no son aceptables [Tapia and Elwany, 2014, Everton et al., 2016]. Uno de los métodos que se ha propuesto para solucionar este problema es la monitorización in-situ del proceso. La monitorización de variables clave junto con un adecuado tratamiento de datos debería permitir conocer la calidad final de la pieza sin necesidad de caracterizaciones o ensayos mecánicos posteriores.

Tapia et al. resume los principales métodos de monitorización para fabricación aditiva de metal por láser en [Tapia and Elwany, 2014]. En su resumen Tapia asegura que el principal factor a monitorizar en estos procesos es la temperatura, ya que está directamente relacionada con la mayoría de propiedades de la pieza final. Para medir la temperatura del proceso distingue dos métodos, las técnicas sin contacto (pirómetros) y las técnicas con

contacto (termopares).

Los pirómetros se basan en la medición de la radiación térmica emitida por un cuerpo para determinar su temperatura. En [Michalski and Michalski, 2001] se explican los fundamentos de la medición de temperatura en base a la radiación térmica emitida. Este proceso se basa en que todo cuerpo con una temperatura superior al cero absoluto radia energía en forma de radiación electromagnética. La aproximación de cuerpo negro predice la cantidad de energía radiada por un cuerpo a una temperatura concreta, así como el perfil de emisión en las diferentes longitudes de onda. Los cuerpos reales no cumplen las condiciones de cuerpo negro, por lo que las ecuaciones deben ser modificadas para predecir su comportamiento. A la diferencia entre la emisión de un material real y la emisión de un cuerpo negro a su misma temperatura se le denomina emisividad del material y es clave para poder medir de forma precisa su temperatura en base a la radiación térmica emitida.

Tapia et al. distinguen entre dos tipos de pirómetros, los fotodiodos y las cámaras termográficas. Los fotodiodos son sensores que transforman la radiación electromagnética en una señal eléctrica. Por otro lado, las cámaras termográficas son simplemente una matriz de fotodiodos que permiten reconstruir información espacial del campo de temperaturas. Las cámaras tienen una mayor utilidad que los fotodiodos en cuanto a que recogen información más rica del proceso, pero a su vez tienen la desventaja de tener que tratar una mayor cantidad de información.

Tapia et al. también hablan de las técnicas con contacto, más concretamente del uso de termopares, pero explica cómo estas técnicas no tienen la misma flexibilidad que los pirómetros por lo que no suelen ser utilizadas más que para experimentos concretos y no para monitorizar la temperatura del proceso de forma continuada.

Everton et al. en [Everton et al., 2016] también mencionan las mismas técnicas de monitorización distinguiendo entre pirómetros coaxiales y no coaxiales y añadiendo el uso de cámaras visibles para monitorizar el proceso de recoating. Un pirómetro coaxial se caracteriza por estar incluido en el tren óptico del láser de forma que su campo de visión esté siempre centrado en el punto de incidencia del láser sobre la cama de polvo. A su vez los pirómetros no coaxiales se colocan de forma completamente separada del láser y enfocan a una parte concreta o a la totalidad de la cama de polvo.

La monitorización del proceso de recoating consiste en la toma de imágenes de la cama de polvo completa antes y después de extender el polvo. De esta forma se pueden observar posibles problemas de homogeneidad en las capas que podrían irse acumulando hasta

provocar fallos en las piezas. Scime et al. en [Scime et al., 2020, Scime and Beuth, 2018a, Scime and Beuth, 2018b] describen varios métodos de monitorización y análisis de datos que permiten detectar y clasificar defectos en el proceso de recoating y fusión gracias a fotos de la cama de polvo. Estos métodos permiten detectar anomalías que se producen durante la impresión, lo que a su vez podría permitir generar lazos de control que corrijan estos errores o que en caso de no poder corregirse detengan la impresión para evitar malgastar recursos en una pieza que no pasará los controles de calidad. A su vez tener un sistema de detección de fallos suficientemente fiable sirve para generar informes de impresión que garanticen la calidad de la pieza. Para esta aplicación es muy importante que el sistema de detección tenga un bajo número de falsos negativos para que el informe de calidad sea fiable.

Otro método de monitorización in-situ encontrado en bibliografía son las emisiones acústicas [Wang, 2020]. Este método se aplica principalmente a los proceso de cama de polvo por láser y no a EBM, ya que la cámara de proceso está al vacío, lo que evita la propagación del sonido. Schevchik et al. en [Shevchik et al., 2019, Shevchik et al., 2018] proponen el uso de sensores acústicos de alta frecuencia y alta sensibilidad para distinguir los sonidos producidos durante el proceso de fusión del metal. Además proponen un método de análisis de datos que permite distinguir entre diferentes modos de fusión (“keyhole”, conducción y falta de fusión). La principal ventaja que plantea el uso de emisiones acústicas frente a los métodos ópticos es su alta resolución temporal. Los métodos ópticos más rápidos rondan los 200KHz de frecuencia de muestreo mientras que los sistemas acústicos utilizados por Schevchik et al. alcanzaban los 10MHz.

2.3. Inteligencia artificial en Fabricación aditiva

En los últimos años, la inteligencia artificial aplicada a la fabricación aditiva ha recibido gran interés por parte de la comunidad científica, ya que las técnicas basadas en datos prometen resolver algunos de los problemas que evitan que la fabricación aditiva despliegue todo su potencial [Wang, 2020, Meng, 2020, Johnson, 2020].

Existe una gran cantidad de campos dentro de la fabricación aditiva donde se ha aplicado la inteligencia artificial. Wang et al. distinguen tres dominios de aplicación diferenciados y 8 sub-dominios [Wang, 2020]. Se considera esta división como una de las mejores encontradas en la bibliografía por lo que se tomará como esquema para el resto

del apartado.

▪ **Diseño para fabricación aditiva**

- **Diseño de Materiales:** En la fabricación aditiva de metal existe una limitación en cuanto a los materiales disponibles para fabricar piezas. No todas las aleaciones desarrolladas para la fabricación convencional se pueden utilizar en los procesos aditivos. El aprendizaje automático ha sido aplicado con éxito para el diseño de nuevos materiales [Ma et al., 2018, Gu et al., 2018]. Según Wang et al. este campo de aplicación está relativamente poco explorado pero tiene un alto potencial.
- **Diseño de piezas:** Una de las mayores ventajas que supone la fabricación aditiva es la libertad de diseño que proporciona. Las técnicas aditivas permiten realizar geometrías que serían muy caras de realizar por otros métodos de fabricación o incluso imposibles por su gran complejidad. Dentro del campo del diseño para fabricación aditiva se llevan utilizando muchos años métodos de optimización topológica que aprovechan la libertad de diseño para maximizar o minimizar alguna propiedad de la pieza al mismo tiempo que se cumplen las restricciones de diseño y las condiciones de contorno del problema [Eschenauer and Olhoff, 2001]. La forma clásica de resolver este problema se basa en un método iterativo de optimización combinado con simulaciones mecánicas. Estos métodos de optimización topológica clásica son generalmente lentos por culpa de las simulaciones numéricas que realizan. Es en este punto en el que se centran la mayoría de trabajos de inteligencia artificial aplicada al diseño de piezas para fabricación aditiva; intentan sustituir la optimización topológica clásica por métodos basados en datos que aceleren considerablemente el proceso. Cang et al. proponen un método para generar piezas cuasi-óptimas en base a un algoritmo de meta-aprendizaje en un solo paso [Cang et al., 2018].

▪ **Proceso de fabricación**

- **Optimización de parámetros de proceso:** Clásicamente la optimización de parámetros de proceso se realiza por dos vías, los diseños de experimentos y las simulaciones del proceso [Meng, 2020]. Ambos métodos tienen sus inconvenientes, por un lado los diseños de experimentos suponen un proceso de prueba y error que suele ser poco eficiente y costoso. Por otro lado las simulaciones del proceso de impresión son muy costosas computacionalmente cuando se realizan

a la escala de la “Melt pool” y tienen mucha menos precisión cuando se hacen a la escala de la pieza.

En general, el aprendizaje automático aplicado a la optimización de parámetros intenta modelar la relación entre los parámetros de proceso y las propiedades finales de la pieza. Una vez obtenido un modelo fiable de la relación entre parámetros y propiedades, los modelos se pueden utilizar como modelos subrogados para encontrar la mejor combinación de parámetros de proceso. Aoyagi et al. describen una metodología para crear mapas de proceso en base a máquinas de vector soporte de clasificación binaria [Aoyagi et al., 2019].

- **Caracterización del proceso de recoating:** La homogeneidad de la capa de polvo es un factor con gran influencia en la calidad final de las piezas impresas en PBF. Por ello existen diferentes estudios en los que se estudia cómo mejorar el proceso de recoating con ayuda de la inteligencia artificial. Scime et al. crearon un sistema de detección de anomalías durante el proceso de recoating, basado en el uso de redes neuronales convolucionales para detección de objetos y segmentación [Scime et al., 2020, Scime and Beuth, 2018b, Scime and Beuth, 2018a]. Zhang et al. crearon un modelo que relacionaba los parámetros de recoating con la calidad de la capa de polvo extendida. Este modelo se creó en base a un dataset generado a partir de simulaciones numéricas del proceso de extensión de polvo. [Zhang et al., 2017]
- **Detección de defectos durante el proceso de impresión:** Los procesos de fabricación aditiva de metal tienen una gran cantidad de defectos que se producen durante el proceso de impresión, desde porosidad interna hasta fallos catastróficos de la impresión pasando por grietas o excesiva rugosidad superficial. Con el objetivo de detectar estos defectos durante el proceso de impresión se han diseñado múltiples metodologías de monitorización in-situ, descritas en el apartado anterior. Sin embargo no es suficiente con monitorizar el proceso, sino que también se necesita de un sistema de análisis de datos que sea capaz de detectar los patrones asociados a esos defectos. Es por eso que muchos investigadores recurren al aprendizaje automático en este tipo de estudios. En la próxima sección se entrará más en detalle sobre este tipo de aplicación.

▪ **Producción**

- **Planificación de la producción:** La fabricación aditiva es un proceso caro

en su estado actual. Por ello cualquier mejora en la eficiencia de la producción puede suponer un gran beneficio. Los trabajos mencionados por Wang et al. en este apartado se centran principalmente en sistemas que determinan si una pieza concreta es fabricable por una determinada tecnología aditiva [Wang, 2020]. Evitar imprimir una pieza que va a fallar por un error de diseño puede suponer un ahorro muy considerable.

- **Control de calidad:** La calidad es uno de los grandes problemas abiertos de la fabricación aditiva, por ello hay multitud de estudios que aplican técnicas de aprendizaje automático a la tarea de controlar o predecir la calidad de una pieza fabricada por métodos aditivos. Este campo de aplicación se solapa considerablemente con la detección de defectos durante el proceso de fabricación en cuanto al uso de técnicas de monitorización in-situ pero difiere ligeramente en el enfoque aplicado. Los problemas enfocados como control de calidad generalmente pretenden predecir alguna propiedad final de las piezas que se relaciona con su calidad, mientras que los problemas de detección de defectos pretenden encontrar el patrón en los datos que determinan cuándo se está produciendo un defecto concreto. A pesar de esta diferencia en enfoque, ambos campos están muy relacionados y no son tan fácilmente separables, ya que los defectos y la calidad de una pieza son inversamente proporcionales puesto que una pieza de máxima calidad no debería tener ningún defecto y al contrario, si una pieza tiene defectos su calidad no será buena. Por ello se tratarán ambos campos de forma más o menos intercambiable en la siguiente sección.
- **Seguridad de datos:** Al tratarse de un proceso intrínsecamente ligado a los diseños digitales, la fabricación aditiva es susceptible de ser atacada con el objetivo de robar la propiedad intelectual contenida en los diseños de fabricación. Generalmente este problema queda restringido al campo de la ciberseguridad informática, sin embargo en fabricación aditiva, se ha demostrado que es posible reconstruir los diseños de fabricación utilizando las señales acústicas que emiten las impresoras y algoritmos de aprendizaje automático para procesar los datos [Al Faruque et al., 2016].

Además de los campos de aplicación que se acaban de describir es muy importante tener un esquema claro de los datos involucrados en los procesos de fabricación aditiva para entender las posibles aplicaciones del aprendizaje automático a este campo. Meng et

al. hacen una clasificación de los datos involucrados que se resume en la figura 2.4.

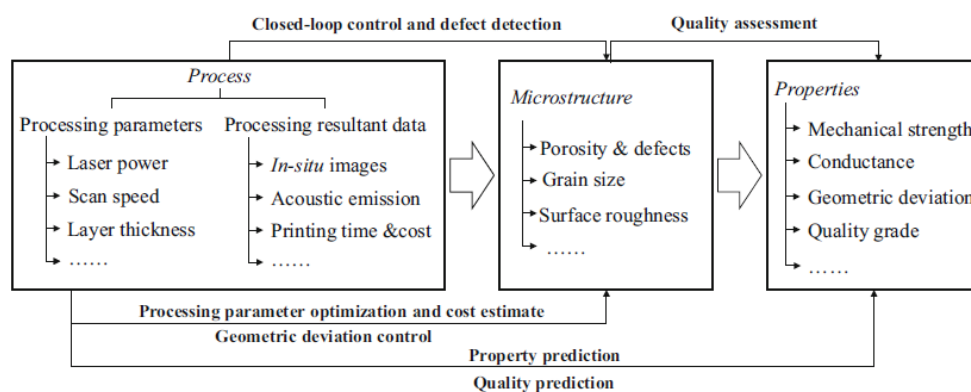


Figura 2.4: División de los datos involucrados en fabricación aditiva por [Meng, 2020].

2.4. Detección de defectos y predicción de calidad en sistemas de SLM

Como se vio en el apartado anterior, uno de los campos de aplicación del aprendizaje automático a la fabricación aditiva es la detección de defectos in-situ o enfocándolo de otra forma, la predicción de la calidad de impresión en base a monitorización in-situ. Atendiendo a la naturaleza del sistema de monitorización se puede distinguir entre enfoques basados en monitorización óptica y acústica. A pesar de que hay mucha más bibliografía que trata con sistemas ópticos, se considera que los sistemas acústicos son también interesantes ya que captan información sobre el proceso que no está disponible para los sistemas ópticos.

2.4.1. Monitorización acústica

Shevchik et al. en sus dos artículos [Shevchik et al., 2018, Shevchik et al., 2019] presentan un modelo que tomando datos de monitorización acústica de un proceso de SLM es capaz de predecir el modo de fusión que se está produciendo. Distinguen entre tres categorías de modo de fusión, baja energía ($50 J/mm^3$), condiciones óptimas ($79 J/mm^3$) y exceso de energía ($132 J/mm^3$). Los datos que utilizan se extraen del sistema de monitorización en forma de serie temporal, pero los autores realizan un preprocesamiento para transformarlo en imágenes antes de introducirlos en el modelo. El preprocesamiento consiste en una transformación ondúlica (“Wavelet transform”) que transforma la señal

en un espectrograma. El modelo que utilizan es un red convolucional espectral (SCNN) propuesta por Mathieu et al. en [Mathieu et al., 2014]. La SCNN es una modificación de una red convolucional que acelera el proceso de entrenamiento e inferencia utilizando la equivalencia entre las convoluciones en el espacio geométrico y las operaciones puntuales en el dominio de la frecuencia. Schevchik et al. utilizan la SCNN para crear un modelo de clasificación con una precisión de entre un 83 % y un 89 % en función de la clase predicha.

Ye et al. plantean un problema muy similar al trabajo anterior en el que pretenden predecir el modo de fusión en base a un sistema de monitorización acústica [Ye et al., 2018]. En este estudio a diferencia del anterior se utilizan datos de “single tracks” en lugar de datos de cubos y se distingue entre 5 niveles de energía en lugar de 3. Sin embargo, la principal diferencia radica en el modelo utilizado, Ye et al. utilizan una “Deep believe network” con y sin preprocesamiento previo de los datos. Probaron 3 tipos de preprocesamiento de los datos “Fast fourier transform” (FFT) con eliminación de ruido, FFT únicamente y ningún preprocesamiento. Los resultados de los datos sin preprocesamiento rondan el 73 % de precisión mientras que utilizando FFT con o sin eliminación de ruido la precisión sube al 95 %.

Por último Wasmer et al. presentan en [Wasmer et al., 2019] un método en el que usan aprendizaje por refuerzo para resolver el mismo problema presentado por Shevchik et al., utilizando los mismos datos. A pesar de usar el mismo dataset, los resultados de este modelo son peores que los obtenidos con la SCNN.

2.4.2. Monitorización óptica

Existen multitud de artículos en los que se aplican modelos de aprendizaje automático sobre datos de monitorización in-situ óptica. La mayoría utilizan cámaras infra-rojas o cámaras de alta velocidad para detectar propiedades del proceso de fusión, concretamente, geometría de la “Melt pool”, temperaturas y gradientes térmicos de la “Melt pool” y patrones de “Spatering” y “Plume”. Por “Spatering” se entienden las proyecciones no gaseosas que salen de la zona fundida y por “Plume” el gas y plasma que emite la “Melt pool” [Wang, 2020].

Sin embargo, no hay muchos estudios en los que se utilicen fotodiodos como única fuente de información. Okaro et al. presentan en [Okaro, 2019] una metodología para utilizar datos de fotodiodos coaxiales para predecir la calidad de probetas de tracción. La calidad de las probetas la definen en base a su resistencia a la tracción, realizan un pro-

ceso de binarización en el que se diferencia entre resultados aptos y no aptos en función de un umbral. De esta forma se formula el problema de predicción de calidad como un problema de clasificación binaria. Partiendo de los datos en bruto, Okaro et al. realizaron un proceso de reducción de dimensionalidad y extracción de características utilizando una descomposición de valores singulares (“Singular value decomposition”). Una vez reducida la dimensionalidad de la serie temporal de entrada aplicaron un modelo de mezcla gaussiana (“Gaussian mixture model”). Los modelos de mezcla gaussianos son modelos semi-supervisados en los que solo parte del dataset necesita estar etiquetado. Con esta metodología de trabajo, los autores son capaces de obtener una precisión del 77% en la clasificación.

En otro estudio aún sin publicar (disponible en www.preprints.com) del mismo grupo de investigación [Jayasinghe et al., 2020], se presenta una continuación del trabajo de Okaro et al. en el que usando el mismo tipo de datos con el mismo preprocesamiento aplican técnicas de aprendizaje supervisado y no supervisado a la predicción de la densidad. En la primera parte, binarizan el dataset de la misma forma que en la publicación anterior pero en este caso atendiendo a la densidad de la pieza. De esta forma vuelven a resolver un problema de clasificación binaria, pero en este caso lo plantean como un problema completamente no-supervisado (K-means y Gaussian mixture model). En una segunda parte, utilizan procesos gaussianos para generar un modelo de regresión que prediga la densidad numérica de las piezas del dataset.

2.5. Clasificación de series temporales

Aparte del estudio de las técnicas aplicadas al campo en el que se va a desarrollar el presente trabajo, es importante hacer una breve exploración de las técnicas de clasificación de series temporales existentes en la bibliografía de aprendizaje automático.

Se van a distinguir principalmente dos acercamientos a la clasificación de series temporales; enfoques desde el aprendizaje automático más clásico y enfoques desde el aprendizaje profundo. En la bibliografía consultada, se encuentran diferentes taxonomías para describir cada uno de estos campos, sin embargo, en este capítulo se van a seguir las propuestas por [Susto et al., 2018] y [Abanda et al., 2019] para aprendizaje automático clásico y [Fawaz, 2019] para aprendizaje profundo.

En primer lugar, hay que destacar que la mayoría de metodologías encontradas para

la clasificación de series temporales comienzan con un primer paso en el que se cambia la representación de los datos con los que trabajará el modelo. Salvo en las implementaciones “end-to-end” encontradas en aprendizaje profundo, ningún modelo trabaja directamente con los datos de la serie temporal sin definir una transformación por encima. Esto ya se vio en la bibliografía encontrada en el campo de la fabricación aditiva.

Respecto a los acercamientos basados en aprendizaje automático, Susto et al. diferencian dos tipos de clasificadores, los basados en características y los basados en distancias. Los métodos basados en características, aplican lo que se conocen como reducción de dimensionalidad, que se basa en generar una representación de los datos que “resuma” de la mejor manera posible la información de los datos. Es esta representación reducida con la que trabajará el modelo después. Susto et al. destacan las siguientes técnicas de reducción de dimensionalidad: generalized discriminant analysis, independent component analysis, linear discriminant analysis, principal component analysis y kernel PCA. Tras la generación de las características estos métodos pueden emplear cualquier algoritmo de clasificación.

Los métodos basados en distancias consisten en la definición de una distancia entre series temporales que se utilizará posteriormente con una técnica de clasificación basada en distancias [Susto et al., 2018]. Típicamente la técnica más utilizada en estos enfoques es k vecinos más cercanos aunque también se puedan aplicar otras [Abanda et al., 2019]. Abanda et al. presentan en [Abanda et al., 2019] una revisión de los métodos basados en distancias en la que distinguen entre técnicas que utilizan la distancia para extraer características y técnicas que generan un kernel de distancias que luego se utilizará para la clasificación. Combinando las clasificaciones de Abanda et al. y Susto et al. es posible que lo que Abanta et al. denominan métodos basados en distancias para la extracción de características, pudiera encajar mejor dentro de los métodos basados en características, ya que el modelo de clasificación trabajará con características en última instancia.

Por otro lado, Fawaz et al. proponen una taxonomía de los acercamientos basados en aprendizaje profundo que se resume en la figura 2.5.

La distinción entre modelos generativos o discriminativos expresada por Fawaz et al. se fundamenta en si existe un paso previo de entrenamiento no supervisado o si el modelo trabaja directamente con los datos. La idea detrás del paso no supervisado de los modelos generativos es la creación de una buena representación de los datos en base a técnicas que aprendan de los datos en sí, sin estar sesgados por las etiquetas de los datos. En cuanto a las técnicas discriminativas, aunque Fawaz et al. las describan cómo los métodos

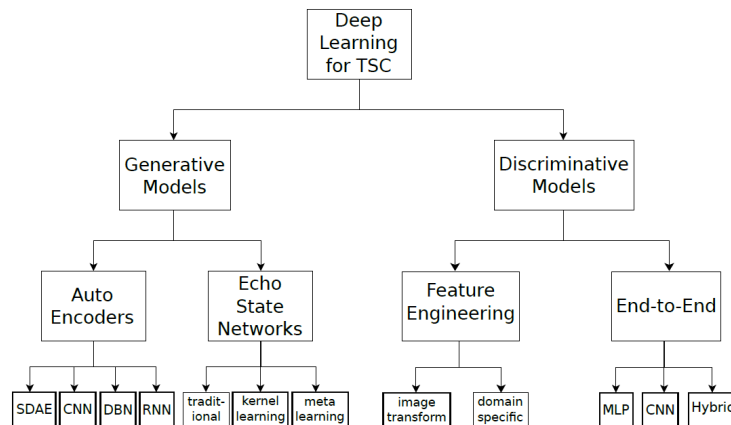


Figura 2.5: Taxonomía de los acercamientos para clasificación de series temporales basados en aprendizaje profundo [Fawaz, 2019].

que trabajan con la serie temporal en bruto, en una de sus dos ramas, la denominada ingeniería de características, los datos que llegan al modelo no son los datos en bruto, sino que igual que en las técnicas de aprendizaje automático se realiza una reducción de dimensionalidad previa al modelado. En el caso de las técnicas “end-to-end” sí que se trata de una metodología que trabaja con los datos brutos para el proceso de modelado. Excluyendo los métodos “end-to-end”, todas las técnicas descritas por Fawaz et al. siguen la misma filosofía que las técnicas basadas en características mencionadas por Susto et al., estando la diferencia en la forma en que se calculan estas representaciones y los modelos que se utilizarán posteriormente de la extracción de características.

Fawaz et al. reportan como las implementaciones de aprendizaje profundo superaran en general a las técnicas de aprendizaje automático salvo las más vanguardistas como “COTE” [Bagnall et al., 2015] o “HIVE-COTE” [Bagnall et al., 2017]. Sin embargo, Fawaz et al. resaltan que estas técnicas a pesar de ser las más precisas, carecen de escalabilidad al tratarse de grandes ensamblajes de modelos con un alto coste computacional. Por este motivo, para casos de uso en los que el tamaño de las series temporales y/o la cantidad de datos sea elevada, se recomienda el uso de modelos de aprendizaje profundo.

2.6. Aportación

En el presente trabajo se pretende abordar el problema de predicción de la calidad de una pieza impresa en una máquina de SLM equipada con fotodiodos coaxiales. El

sistema de monitorización que se utilizará es equivalente al utilizado por [Okaro, 2019, Jayasinghe et al., 2020] en sus publicaciones. En base a estos datos de monitorización se pretende resolver una tarea de clasificación multiclase. La tarea se formulará de forma equivalente a las presentadas por [Shevchik et al., 2018, Shevchik et al., 2019, Ye et al., 2018], en la que se imprimirán piezas con diferentes parámetros de proceso y diferentes densidades de energía y se intentará predecir el modo de fusión inducido en cada una de ellas.

Para la creación del modelo, se van a aplicar principalmente técnicas basadas en aprendizaje profundo. Siguiendo al taxonomía de [Fawaz, 2019] se van a explorar ambas ramas. Como representantes de los modelos generativos se van a implementar varios tipos de autoencoders, mientras que de la rama de los modelos discriminativos se utilizarán varias técnicas de reducción de dimensionalidad incluyendo la propuesta por Okaro et al. con el fin de usarla como baseline.

Capítulo 3

Objetivos del trabajo

3.1. Objetivo general

El objetivo principal del presente trabajo es la creación de un modelo de aprendizaje automático capaz de predecir la calidad de una pieza en base a los datos de monitorización tomados durante el proceso de impresión en una máquina de SLM. Concretamente se espera poder crear un modelo de clasificación multiclase, que sea capaz de distinguir entre 3 o más clases de modos de fusión utilizando la información de 3 fotodiodos coaxiales que capturan la emisiones ópticas durante el proceso de impresión. Para evaluar la consecución de este objetivo, se estudiarán las métricas de clasificación del modelo final y se pretende que el modelo tenga al menos una tasa de aciertos (accuracy) del 80%. Esta métrica se establece en base a los resultados de trabajos similares encontrados en bibliografía y simplemente se establece cómo una cota inferior para que se pueda considerar que el resultado tenga interés.

Para la consecución de estos objetivos principales se deben dar múltiples de pasos que pueden ser objetivos de por sí. Estos objetivos se detallan en el siguiente apartado.

3.2. Objetivos específicos

Entre los pasos necesarios se pueden distinguir los siguientes objetivos específicos o secundarios,

- Identificar la bibliografía previa que ha abordado problemas similares. Este paso es importante para poder evaluar si el método desarrollado mejora los métodos existentes o si ofrece algún otro tipo de ventaja.

- Estudiar la fuente de datos y la forma de extraer los datos. Este paso es importante para comprender la naturaleza de los datos así como el formato de los mismos.
- Seleccionar y evaluar el preprocesamiento de los datos. Esta será una de las partes más importantes en el trabajo dada la gran cantidad de datos que genera el sistema de monitorización. Se debe encontrar la forma de sacar el mayor rendimiento posible de los datos.
- Crear un modelo de clasificación multiclase que obtenga las métricas de acierto mencionadas.
- Evaluar de forma crítica y razonada el desempeño del modelo creado. En este objetivo también se incluye la comparación con implementaciones previas que se hayan encontrado en bibliografía.

Capítulo 4

Metodología y Descripción detallada del experimento

4.1. Sistema de monitorización

Teniendo en cuenta los objetivos presentados en el capítulo anterior, está claro que uno de los elementos clave de este trabajo va a ser el sistema de monitorización utilizado. La impresora donde se van a realizar todos los experimentos es una Renishaw AM500Q. Se trata de una impresora de L-PBF con cuatro láseres capaz de imprimir con una potencia máxima de 500W. Cada uno de los láseres tiene un conjunto de 3 fotodiodos coaxiales que monitorizan el proceso de impresión. Cada uno de estos fotodiodos mide un rango del espectro electromagnético de forma que entre los 3 cubren desde los 700nm hasta los 1700nm. Más concretamente las especificaciones técnicas de la impresora indican que el fotodiodo al que llaman “MeltVIEW visible” es sensible desde los 700nm hasta los 1040nm, el fotodiodo denominado “MeltVIEW infrarojo” desde los 1090nm hasta los 1700nm y el denominado “LaserVIEW” mide entre los 1050nm y los 1080nm. Es importante destacar que el fabricante de la máquina llama “visible” al fotodiodo que captura desde los 700nm hasta 1040nm, lo que no tiene ningún solape con el espectro visible que va desde los 400nm hasta los 700nm. Se entiende que se le da este nombre por ser el más cercano a dicho espectro. Los rangos que captura cada fotodiodo pueden parecer arbitrarios, pero teniendo en cuenta que la longitud de onda del láser es de aproximadamente 1070nm los motivos de estos rangos quedan más claros.

Además de la diferencia en rango de medición, los fotodiodos también difieren en la

procedencia de la luz que miden. El fotodiodo “LaserVIEW” está colocado de forma que mide la luz emitida por el láser antes de ser enviado a la zona de impresión, mientras que los fotodiodos de “MeltVIEW” miden la intensidad reflejada por la zona de la meltpool. En la figura 4.1 se puede ver un esquema del tren óptico que genera este comportamiento. Los fotodiodos de “MeltVIEW” son las referencias 4 y 5 mientras que “LaserVIEW” es la referencia 10.

Teniendo en cuenta las diferencias entre los fotodiodos, se puede entender que el objetivo de cada uno de ellos es diferente. Los fotodiodos de “MeltVIEW” monitorizan la radiación electromagnética emitida por la zona de metal fundido sobre la que incide el láser. Como se vio en el apartado de monitorización in-situ, este tipo de medidas está directamente relacionadas con la temperatura del proceso. Los datos recogidos de estos dos fotodiodos servirán para determinar una medida de temperatura relativa. Para tener la temperatura absoluta se tendría que realizar un proceso de calibración que se escapa del alcance de este trabajo. El hecho de tener dos fotodiodos con rangos diferentes se debe a que la diferencia de longitudes de onda los hace más sensibles a diferentes rangos de temperatura. Por otro lado, el fotodiodo “LaserVIEW” está destinado a realizar una medida cuasi directa de la intensidad del láser.

Desde el punto de vista de la recolección de datos, el fabricante proporciona una arquitectura de captación y preprocesado de los datos de la que se extraen los datos en forma de series temporales almacenadas en archivos CSV. Estos archivos se organizan internamente en lo que el fabricante denomina como puntos de exposición. Cada punto de exposición se corresponde con un pulso del láser modulado. De forma interna el sistema de captación recoge la información con una frecuencia de muestreo de 100KHz o lo que es lo mismo tomando un dato cada $10\mu s$. Los puntos de exposición en esta máquina tienen como duración mínima justamente $10\mu s$ por lo que en todos los casos el sistema de monitorización tomará 1 o más medidas durante cada punto de exposición. Para calcular el valor final que se almacena en el CSV, el sistema de preprocesamiento hace la media de todas las medidas tomadas a lo largo de la duración del punto de exposición en cuestión.

Para cada uno de los puntos de exposición el CSV recoge 7 valores:

- TimeStamp del comienzo de la exposición en μs
- Duración del punto de exposición en μs
- Posición x del punto de exposición sobre la cama de polvo en mm
- Posición y del punto de exposición sobre la cama de polvo en mm

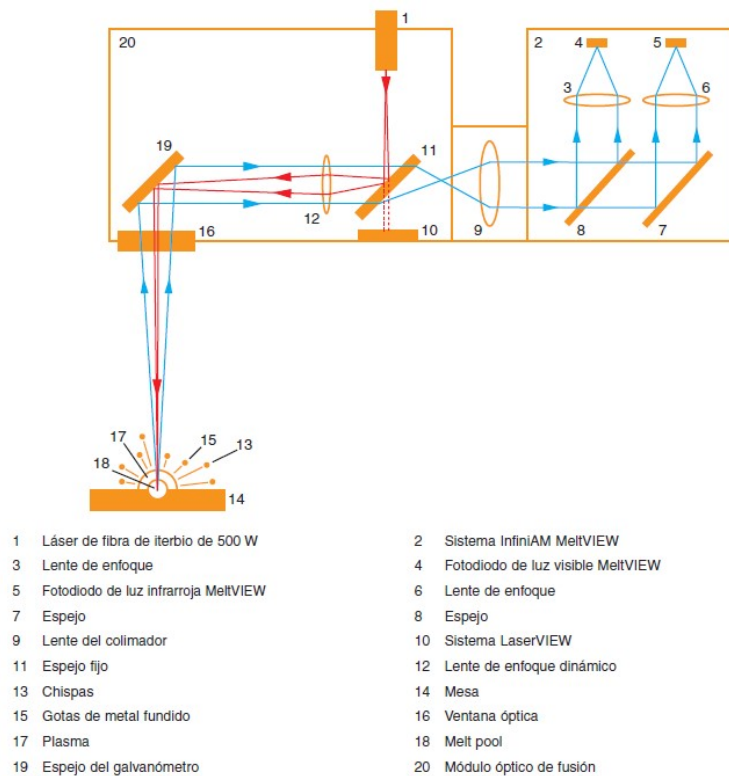


Figura 4.1: Esquema del tren óptico de medición [Renishaw, 2021].

- Intensidad media recogida por “LaserVIEW”
- Intensidad media recogida por “MeltVIEW Visible”
- Intensidad media recogida por “MeltVIEW Infra-rojo”

Las 4 primeras columnas contextualizan temporal y espacialmente los datos captados por los fotodiodos. Los datos temporales permiten conocer la duración de los pulsos del láser, el tiempo que permanece el láser apagado entre dos pulsos diferentes así como cuando se cambia de vector de escaneo. A su vez los datos de posición permiten representar los datos en forma de imágenes, así como separar la información relativa a cada una de las diferentes piezas de una impresión. Los datos de intensidad de los fotodiodos se guardan en unidades arbitrarias y toman valores enteros comprendidos entre 0 y 4095, lo que se corresponde con una discretización de 12 bits.

4.2. Captura de los datos del dataset

Con el fin de crear un dataset con datos de monitorización en el que se conozca la calidad asociada a los mismos, se recurrió a un enfoque muy similar al propuesto por

[Shevchik et al., 2018]. La idea es imprimir piezas con diferentes parámetros de proceso que produzcan modos de fusión conocidos (Keyhole, falta de fusión o fusión nominal). Estos 3 modos de fusión están asociados con muchos de los defectos producidos en fabricación aditiva de metal por lo que son unos buenos candidatos para representar la calidad del proceso de fusión. Schevchik et al. imprimieron 1 cubo variando la paramentaría a lo largo de las capas de forma que diferentes secciones del cubo tuvieran modos de fusión diferentes. Utilizaron 3 sets distintos de parámetros, uno con energía nominal ($79 J/mm^3$) otro con exceso de energía (Keyhole) ($132 J/mm^3$) y otro con falta de energía (Falta de fusión) ($50 J/mm^3$).

En este trabajo se decidió ampliar el enfoque propuesto imprimiendo más de un cubo y utilizando 18 sets de parámetros de impresión diferentes que dan lugar a 5 niveles de fusión distintos. Se imprimieron un total de 40 cubos de $20mm$ de lado distribuidos en 4 impresiones. Las camas de impresión se diseñaron minimizando la interacción entre los cubos para evitar contaminación en las medidas.

Los 18 sets de parámetros se obtuvieron variando la potencia del láser, la distancia entre puntos de exposición y el tiempo de exposición, dejando fijos el resto de parámetros. A su vez se definieron 5 rangos de densidad de energía volumétrica (VED) correspondientes a los 5 modos de fusión.

- Alto Keyhole : $165 \pm 10 J/mm^3$
- Keyhole : $110 \pm 10 J/mm^3$
- Nominal : $65 \pm 5 J/mm^3$
- Falta de fusión : $42 \pm 3 J/mm^3$
- Alta falta de fusión : $26 \pm 3 J/mm^3$

Destacar que la definición de estos parámetros se hizo en base al conocimiento experto de los investigadores que operan la impresora y no están completamente alineados con los rangos definidos por Schevick et al. Esta discrepancia se puede deber a que el proceso de traducción de densidades de energía volumétricas entre impresoras diferentes no es un problema resuelto y que en cualquier caso necesitaría multitud de especificaciones técnicas de ambas máquinas, de las que se carece.

Para la selección de los parámetros de impresión se decidió utilizar rangos de parámetros que se utilicen en la realidad sin irse a valores de potencia o de velocidad excesivamente

reducidos. De esta forma se pretende que los datos sean lo más similares posible al proceso de impresión normal. Se optó por utilizar 3 niveles de potencia combinados con 3 velocidades de escaneo diferentes de la forma que se presenta en la tabla 4.1. Al trabajar con un láser pulsado, la velocidad de escaneo se consigue variando tres parámetros, tiempo de exposición, distancia entre puntos de exposición y tiempo entre puntos de exposición. Como se dijo anteriormente en este diseño de experimentos, el tiempo entre puntos de exposición se mantendrá fijo en todos los cubos y solo se varían los otros dos parámetros. Para no complicar en exceso el experimento se dividieron los cubos en dos grupos. En el primero de los grupos, las variaciones en velocidad se obtuvieron dejando fijo el tiempo de exposición y variando la distancia entre puntos de exposición. Mientras que en el segundo grupo se procedió de manera inversa, se mantuvo fija la distancia entre puntos de exposición y se varió el tiempo de exposición. Sin más restricciones este procedimiento daría 6 velocidades de escaneo diferentes, lo que complicaría en exceso el experimento, por ello se escogieron los valores concretos de tiempo de exposición y distancia entre puntos de forma que las tres velocidades del grupo 1 coincidiesen uno a uno con las velocidades del grupo 2. Si bien esta igualdad no se consiguió de forma exacta debido a limitaciones en los valores permitidos para los parámetros de impresión, se escogieron los valores de forma que las diferencias entre velocidades que se pretendía que fuesen iguales no superase el 5%. Combinando los 3 niveles de potencia con las 3 velocidades obtenidas de 2 formas diferentes cada una se obtienen los 18 sets de parámetros diferentes mencionados al comienzo de la sección.

	Alta falta de fusión	Falta de fusión	Nominal	Keyhole	Alto keyhole
V_1			P_1	P_2	P_3
V_2		P_1	P_2	P_3	
V_3	P_1	P_2	P_3		
V_1^*			P_1	P_2	P_3
V_2^*		P_1	P_2	P_3	
V_3^*	P_1	P_2	P_3		

Tabla 4.1: Combinaciones de potencia y velocidad. V_n se corresponde con las 3 velocidades diferentes utilizadas y P_n lo mismo para las potencias. V_n y V_n^* son aproximadamente iguales pero las velocidades V_n se obtuvieron variando la distancia entre puntos de exposición y V_n^* variando el tiempo de exposición.

En la tabla 4.1 se puede observar como los distintos modos de fusión tienen diferente

número de sets de parámetros que son capaces de producirlo, esto es normal ya que se han limitado los parámetros al rango usual de funcionamiento lo que provoca que los modos de fusión más extremos solo se puedan conseguir con combinaciones muy concretas. Sin embargo para evitar tener clases desbalanceadas, el número de cubos impresos con cada set de parámetros no será el mismo. En la tabla 4.2 se presenta el número de cubos impresos con cada combinación. Cruzando la información de la tabla 4.1 y 4.2 se puede observar como cada clase tiene asociados 8 cubos con al menos dos sets de parámetros diferentes.

	P_1	P_2	P_3
V_1	1	2	4
V_2	2	2	2
V_3	4	2	1
V_1^*	1	2	4
V_2^*	2	2	2
V_3^*	4	2	1

Tabla 4.2: Número de cubos impresos para cada set de parámetros.

Además de los parámetros de impresión, se prestó especial cuidado a la colocación de los cubos en la base de impresión. Esto es muy importante ya que una colocación inadecuada podría generar correlaciones en los datos de forma artificial, debidas a interacciones entre los cubos. Para evitar interacciones térmicas los cubos deben colocarse lo más separados posible. El mejor caso sería imprimir cada cubo en una impresión individual, pero esto supondría un aumento muy grande en los costes del experimento. Buscando un punto medio, se separaron los 40 cubos en 4 impresiones distintas. Dado que los cubos con modos de fusión mas extremos tenían mayores probabilidades de fallar durante la impresión, se decidió imprimir los 16 cubos de alto keyhole y alta falta de fusión en las mismas impresiones separados del resto de cubos. De esta forma se realizaron 2 impresiones de 8 cubos y otras 2 impresiones de 12 cubos cada una.

Otra de las formas en que pueden interactuar piezas impresas en la misma cama de polvo es a través del recoater o del flujo de gas. En el caso de que varias piezas estuviesen alineadas siguiendo la dirección que recorre el recoater o el flujo del gas, estos elementos podrían transportar materia entre ambas piezas haciéndolas interactuar. En el caso de la máquina utilizada, el recoater y el flujo de gas son perpendiculares entre sí, por lo que se genera una cuadrícula en la que se debe evitar poner piezas en la misma fila o columna. En la imágenes 4.2 y 4.3 se puede ver los patrones utilizados para las camas de 8 y 12

cubos respectivamente.

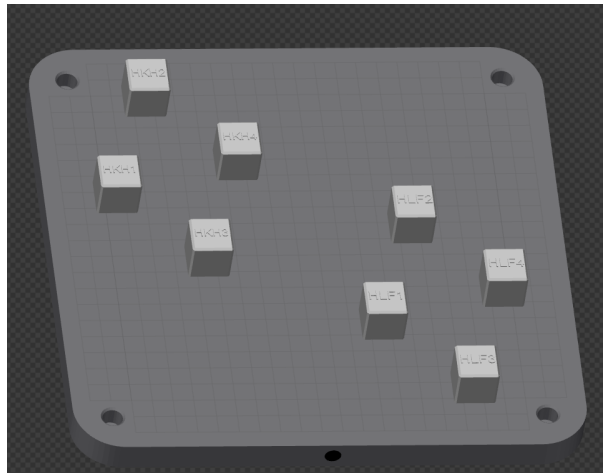


Figura 4.2: Distribución geométrica de cubos de alto keyhole y alta falta de fusión (Elaboración propia).

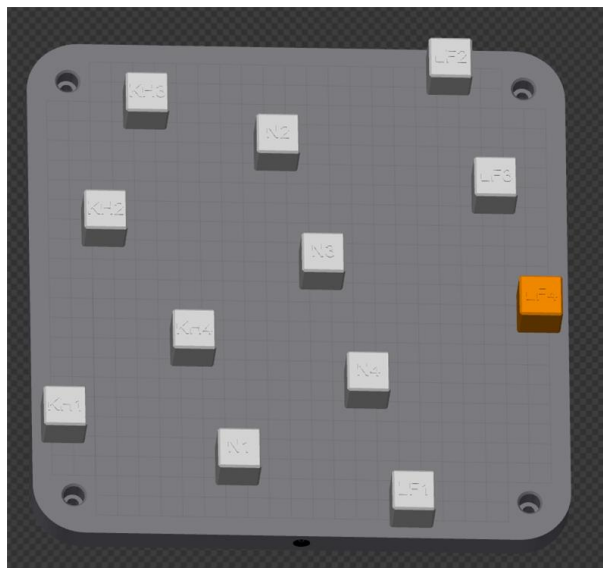


Figura 4.3: Distribución geométrica de cubos de keyhole, fusión nominal y falta de fusión (Elaboración propia).

Tras las impresiones, se realizará un proceso de caracterización en el que medirán las densidades de los cubos y se tomarán imágenes con el microscopio para verificar la cantidad y tipo de porosidad producida en cada uno de los cubos. De esta forma se validará la suposición sobre los modos de fusión inducidos en cada cubo.

4.3. Preprocesamiento de datos

Como ya se ha descrito en el apartado anterior sobre el sistema de monitorización, los datos de los fotodiodos se extraen del software del fabricante en forma de CSV. Para cada impresión se generan cientos de archivos CSV que contienen toda la información de la impresión separada por capa y láser. Dado que la estrategia de impresión escogida involucra imprimir todas las piezas con todos los láseres, la información relativa a cada uno de los cubos se encuentra distribuida a lo largo de la mayoría de los CSV de la impresión. Por ello uno de los primeros pasos de preprocesamiento será segmentar los datos en función de la pieza a la que correspondan.

El siguiente paso consiste en eliminar la información relativa a los contornos de las piezas. Los contornos no se suelen imprimir con los mismos parámetros que las zonas internas de la pieza. Esto está pensado para mejorar el acabado superficial de las piezas. En este caso además servirá para poder medir la densidad de cubos altamente porosos por el método de Arquímedes. Al estar impresos con una parametría diferente los datos tomados durante la impresión de los contornos deben eliminarse para evitar introducir ruido en el dataset.

Por último deben igualarse los datos medidos con láseres distintos. A pesar de utilizar la misma parametría para los 4 láseres, las intensidades medidas por los fotodiodos de distintos láseres no son iguales. Esta discrepancia puede tener dos orígenes. Por un lado, podría ser que los diferentes láseres emitiesen diferentes cantidades de energía cuando su controlador les requiere una misma potencia. Este sería un problema grave que causaría problemas en las propiedades mecánicas de las piezas. En estudios previos realizados con la máquina no se han observado diferencias estadísticamente relevantes entre los láseres, por lo que este posible primer origen se descarta. El segundo posible origen de esta discrepancia sería la medida de los fotodiodos. A pesar de que sean sensores iguales, diferencias en la calibración o en la fabricación de los sensores o componentes ópticos puede provocar que se midan intensidades distintas aún observando la misma fuente. Esta es la explicación que se considera más factible, por lo que se llevará a cabo un proceso de corrección de las medidas de los diferentes fotodiodos para equipararlos entre ellos.

4.3.1. Segmentación

Como se explicó anteriormente, los datos según se extraen del software del fabricante están divididos por láser y capa de impresión. Esta división de los datos no es la que mejor se ajusta al problema que se quiere resolver en este trabajo, ya que las etiquetas del dataset se corresponden con los cubos y no con las capas o los láseres. A pesar de que no sea la forma deseada de organizar los datos, la división entre láseres es muy útil para poder llevar a cabo la calibración de los fotodiodos en etapas posteriores del preprocesado, por lo que esta información se seguirá almacenando.

A grandes rasgos, se encontraron 3 métodos diferentes de segmentar los datos. Los métodos varían en el nivel de intervención humana requerida para llevarse a cabo y su coste computacional. En esencia, todos los métodos se basan en filtrar los puntos de exposición en función de sus coordenadas x e y sobre la cama de polvo. La diferencia está en la forma en que se decide a que pieza corresponden unas determinadas coordenadas.

- La primera forma consiste en filtrar los datos en base a “bounding boxes” definidas a mano. Esta es la forma más rápida pero que más interacción humana requiere ya que el investigador debe definir cada una de las cajas que engloba cada pieza. Con esta información el programa solo debe recorrer todos los archivos correspondientes a la impresión seleccionando los datos que quedan dentro de las cajas y juntándolos todos en un nuevo DataFrame. Para este trabajo este método podría haber sido suficiente ya que al segmentar cubos en posiciones bien diferenciadas no habría mayor problema. Sin embargo en caso de tratar de segmentar piezas con geometrías más complicadas o en posiciones mucho más cercanas, esta solución corre el riesgo de no segmentar correctamente los datos y mezclar datos de piezas diferentes.
- La segunda alternativa consiste en transformar los datos a una representación 3D de la impresión y utilizando un algoritmo de detección de componentes conexas sacar las máscaras que se corresponden con cada pieza. Este es sin duda el método más autónomo de todos los contemplados, pero también es el más costoso computacionalmente. Para transformar los CSVs en una representación tridimensional se genera una partición del espacio de impresión en forma de voxels cúbicos cuya información se guarda en una array tridimensional. Para rellenar dicho array se recorren todos los CSVs de la impresión y se va guardando un 1 en las posiciones del array para las cuales exista uno o más puntos de exposición que caigan sobre el voxel correspon-

diente. Tras acabar este proceso, se tiene un array binario donde si se ha escogido correctamente el tamaño de los voxels, cada pieza será una componente conexas en el array. El tamaño del voxel es de suma importancia ya que de ser demasiado grande, podrían llegar a juntarse piezas, pero de ser más pequeño que la separación entre puntos o entre vectores de escaneo, una única pieza podría tener más de una componente conexas. Por ello el valor del tamaño de voxel ideal será aproximadamente de $250\mu m$ que se corresponde por exceso con el tamaño de la zona fundida por cada punto de exposición. Es este proceso de generación del array binario lo que más coste computacional supone y hace que este método sea varios ordenes de magnitud más lento que el anterior, pero a cambio, este método es completamente autónomo y no corre riesgo de mezclar datos de piezas.

- El tercer método es una combinación de los dos anteriores. En lugar de generar una representación tridimensional de la pieza, el investigador debe seleccionar una capa que considere adecuada de la que se generará un array bidimensional binario de forma análoga al método anterior. Utilizando un algoritmo de componentes conexas sobre ese array se generara una máscara bidimensional que representa la extensión de cada una de las piezas. Usando esta máscara se pueden filtrar los datos para obtener la segmentación deseada suponiendo que la máscara es válida para todas las capas. Este método es un punto medio entre los métodos anteriores teniendo un coste computacional varios ordenes de magnitud menor que el anterior pero a cambio requiere que se seleccione una capa adecuada para representar al resto. Esta aproximación es perfectamente asumible para piezas con sección constante o monótona, pero falla rápidamente con geometrías arbitrarias.

Dada la simplicidad geométrica de las piezas con las que se trata en este trabajo se optó por el tercer método.

4.3.2. Eliminación de los contornos

Como ya se explicó en la introducción de esta sección, la parametría de los contornos es diferente a la del interior de la pieza y por tanto, se deben eliminar los datos correspondientes a los mismos para evitar introducir ruido en el dataset. El principal problema radica en la forma de identificar qué datos se corresponden con los contornos. En un principio se consideraron dos acercamientos totalmente diferentes para resolver el problema:

- La primera solución consiste en usar el mismo sistema de generación del array binario utilizado en el apartado anterior junto con técnicas de morfología matemática para así determinar el borde exterior de las piezas. Posteriormente usando la máscara del borde, eliminar todos los datos con los que se corresponda. El principal problema de este algoritmo es la imprecisión del método, ya que en función del tamaño de los voxels, se cogerían muchos más datos que los estrictamente necesarios. Por esto y por que no ofrecía una mejora sustancial en cuanto a coste computacional, se descartó esta implementación.
- El segundo método se basa en el tiempo que tarda el láser en saltar entre los puntos de exposición y la geometría de los contornos. Cuando el láser se desplaza entre dos puntos de exposición que están dentro del mismo vector de escaneo el láser está apagado el tiempo que se fija en el parámetro de tiempo entre puntos de exposición que típicamente está en el orden de los $50\mu s$, mientras que cuando el láser pasa del último punto de un vector al primero del siguiente, el láser permanecerá apagado alrededor de $1000\mu s$ como poco. Este comportamiento permite analizar los datos de monitorización y separarlos en vectores de escaneo. Para calcular el tiempo que el láser está apagado entre dos puntos de exposición, simplemente se restan el timestamp de comienzo de ambos puntos y a esto se le resta la duración del primero de los puntos. Con esto simplemente hay que encontrar aquellos saltos que hayan durado más de $1000\mu s$ y etiquetar todos los datos entre estos saltos largos con la misma etiqueta de vector de escaneo. Una vez segmentados los datos en vectores de escaneo se puede diferenciar entre vectores de escaneo internos y externos en función de su longitud total comparada la distancia entre inicio y final. Esto es posible gracias a que los contornos se realizan de manera continua a lo largo de todo el perímetro de la pieza en lugar de en forma de rectas aisladas. Gracias a esta forma de ejecutar los contornos, los vectores de escaneo correspondientes a contornos tendrán una gran longitud pero una distancia entre puntos inicial y final muy pequeña en comparación. De esta forma se pueden identificar todos los datos correspondientes a los contornos y eliminarlos sin correr el riesgo de eliminar datos del interior de la pieza.

Al final se decidió implementar el segundo método, por su precisión a la hora de separar los datos, y porque la segmentación de los datos en vectores de escaneo se utilizará más adelante a la hora de particionar los cubos.

4.3.3. Calibración de los fotodiodos

Durante el desarrollo de este trabajo se detectaron discrepancias entre las medidas de los fotodiodos correspondientes a cada uno de los láseres. La suposición es que si se imprime una pieza con dos o más láseres, pero que todos ellos utilicen los mismos parámetros de impresión, los datos de monitorización haciendo la media a lo largo de toda la pieza se esperaba que fueran muy parecidos si no iguales. Esto debería ser cierto suponiendo que no exista ningún sesgo que haga que un láser tienda a hacer más partes con vectores de escaneo cortos, o tenga más prevalencia en capas más altas en las que se haya más calor acumulado. En general los algoritmos de asignación de láseres tienden a minimizar este tipo de desviaciones y al tratarse de desviaciones aleatorias deberían desaparecer al promediar una cantidad suficiente de piezas o capas. Analizando los datos de varias impresiones, se determinó que la diferencia entre los fotodiodos era consistente en todas las impresiones, por lo que se decidió corregir la medidas. En la imagen 4.4 se puede observar como las zonas realizadas por diferentes láseres tienen intensidades muy diferentes aún habiendo sido impresas con los mismos parámetros.

Con el objetivo de solucionar este problema se decidió calcular los factores correctores que normalizarían la medida de los fotodiodos. Se debe hallar los $N_{l,f}$ de tal forma que:

$$\sum_{j \in P_1^{in}} x_{j,f} = N_{l,f} \sum_{i \in P_l^{in}} x_{i,f} \quad (4.1)$$

Siendo $x_{i,f}$ la medida del fotodiodo f en el punto de exposición i y P_l^{in} todos los puntos de exposición del interior de la pieza P expuestos con el láser l . Destacar que la elección del láser 1 como referencia para la normalización es completamente arbitraria y cualquier otro hubiese válido.

Para calcular las correcciones más precisas posibles se debe escoger una pieza suficientemente grande como para que todos los láseres tengan una presencia equitativa y los sesgos se puedan mitigar por la propia aleatoriedad de la estrategia de escaneo. Dado que los cubos impresos para este trabajo no cumplen esas condiciones se aprovecharon datos recogidos con anterioridad en los que se imprimió una base solida de grandes dimensiones en la que se usaron los cuatro láseres, ver figuras 4.4, 4.5. Usando la ecuación 4.1 se calcularon los coeficientes de corrección que se presentan en la tabla 4.3.

Tras aplicar la corrección a la misma pieza de la imagen 4.4 se obtiene la figura 4.5. Si bien las diferencias entre los láseres no han desaparecido por completo, si se han atenuado

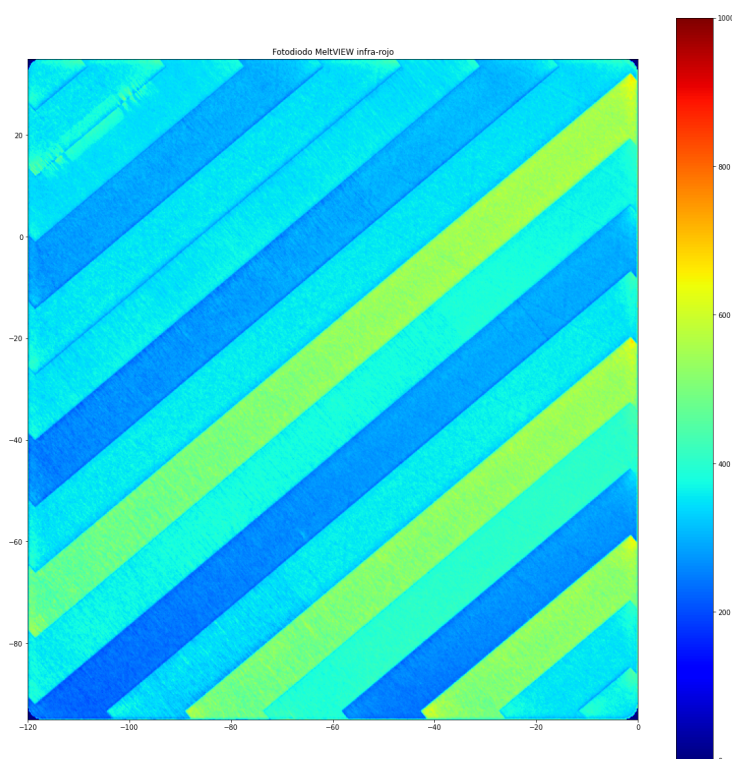


Figura 4.4: Intensidad medida por el fotodiodo MeltVIEW visible. Los datos en bruto fueron transformados a imagen utilizando la información espacial contenida en el CSV (Elaboración propia).

	LaserVIEW	MeltVIEW visible	MeltVIEW infra-rojo
1	1	1	1
2	0.83	0.65	1.54
3	1	0.96	1.49
4	1.01	0.92	1.79

Tabla 4.3: Factores de corrección $N_{l,f}$ de la ecuación 4.1.

hasta el punto en que podrían deberse a la variabilidad intrínseca del proceso.

4.4. Procesamiento de datos

Tras la fase de preprocesamiento se tiene un dataframe para cada uno de los cubos impresos conteniendo todos sus puntos de exposición. En función de la distancia entre puntos de exposición usada para cada cubo, su dataframe correspondiente contiene entre 26 y 63 millones de puntos de exposición. En general, no es factible ni eficiente utilizar directa-

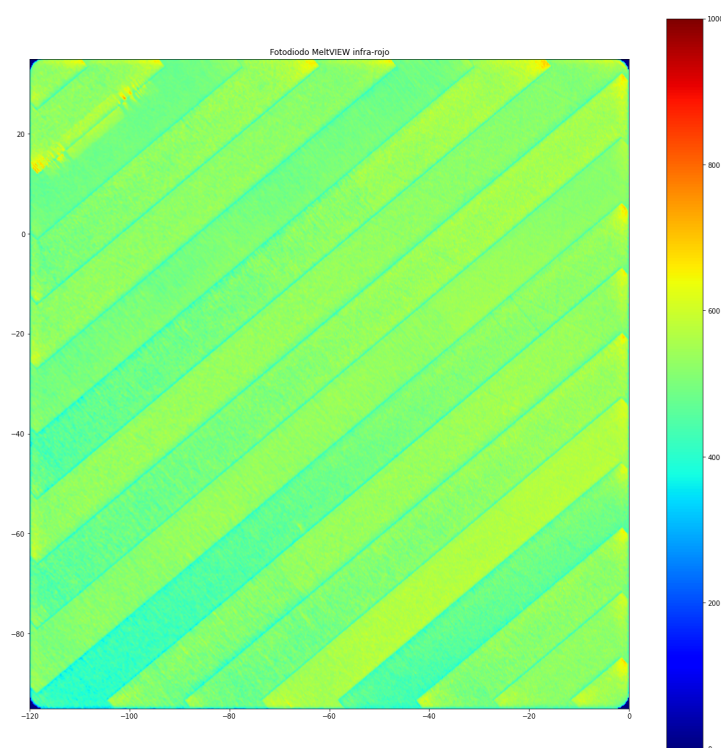


Figura 4.5: Intensidad medida por el fotodiodo MeltVIEW visible tras la calibración de los fotodiodos (Elaboración propia).

mente esta cantidad de información, sino que hay que realizar algún tipo de procesamiento previo al modelado que reduzca el tamaño de los datos de entrada.

Okaro [Okaro, 2019] se enfrenta a este mismo problema combinado dos técnicas. Primero eliminan el 90% de los datos quedándose con los datos de 1 de cada 10 capas y segundo agrupa todo el dataset en una matriz sobre la que aplican una descomposición de valores singulares (SVD) para extraer una representación reducida de los datos.

Shevchik et al. [Shevchik et al., 2019] también resuelven un problema similar. En su trabajo imprimen un único cubo con diferentes parámetros a diferentes alturas. Al solo tener un cubo su dataset final se compone de segmentos de la serie temporal del cubo lo que además de reducir la longitud de la serie aumenta el número de ejemplos en el dataset.

En este trabajo se van a explorar diferentes acercamientos al problema y se compararán los resultados de cada uno. En primer lugar, al igual que Shevchik et al., se van a partir los datos correspondientes a cada cubo a varias escalas. En un segundo paso se tomarán las series temporales resultantes y se aplicarán diferentes técnicas de reducción de dimensionalidad y extracción de características para reducir su tamaño.

4.4.1. Segmentación de los cubos

La primera decisión que cabe tomar es si las etiquetas que se van a asignar a los datos de monitorización tienen sentido a diferentes escalas de la serie temporal.

En el caso de estar trabajando con propiedades mecánicas que se midiesen de forma experimental sobre la pieza impresa en su totalidad (densidad, resistencia a la tracción, etc), la respuesta probablemente sería que las etiquetas solo tienen sentido a escalas similares a las de la pieza completa. Esto se debe principalmente a que al ensayar la pieza completa se está realizando una medida que resume de forma estadística las propiedades locales de toda la pieza. En el caso de la densidad es fácil ver como para un valor de densidad global dado, puede haber zonas de escala reducida con valores de densidad que difieran significativamente de la global pero que al promediarse todas den la densidad global. Es por esto, que las propiedades medidas macroscópicamente pueden no transferirse correctamente a escalas mucho más pequeñas.

Sin embargo en el caso que se trata en este trabajo, las etiquetas se asignan a partir de parámetros de impresión, que son constantes a lo largo de todas las partes de las piezas, por lo que sí puede tener sentido asignárselas a segmentos reducidos de la serie temporal completa. Aun siendo así, es recomendable no ir a una escala demasiado cercana al punto de exposición ya que los datos podrían verse afectados por el ruido de la medida. Es por eso que la escala más pequeña que se considerará en este trabajo son ventanas de 100 puntos de exposición.

Una de las principales ventajas que plantea partir cada cubo en varias series temporales es que se multiplica considerablemente el número de valores en el dataset, lo que puede permitir el uso de técnicas con mayor requerimiento de datos. Teniendo todo esto en cuenta se decidió probar las siguientes tres escalas:

- **Escala de cubo completo:** Utilizando todos los datos correspondientes a cada cubo para generar un dato en el dataset se obtiene un baseline que servirá para verificar la hipótesis sobre la validez de las etiquetas a diferentes escalas.
- **Escala de capas completas:** Las capas de impresión son el siguiente escalón natural por debajo de la pieza completa. A pesar de que las etiquetas se cree que siguen teniendo sentido a esta escala, se decidió eliminar las primeras y últimas capas de cada cubo con el fin de eliminar estados transitivos de fusión que pudieran introducir ruido al dataset. Cada cubo cuenta con 440 capas de las cuales solo se utilizarán desde

la 50 hasta la 420. Las 50 primeras capas se eliminan por corresponderse en su gran mayoría con los soportes de la pieza, mientras que las últimas 20 se eliminan ya que en las capas superiores del cubo se imprimieron los nombres de los mismos lo que genera geometrías mucho más complejas.

- **Escala de vectores de exposición:** Dentro de cada capa de impresión, la siguiente unidad serían los vectores de escaneo. Sin embargo en lugar de optar por los vectores de escaneo como tal, se decidió coger segmentos de 100 puntos de exposición dentro de cada vector de escaneo. Esta decisión se tomó dado que la longitud de los vectores es muy variable, yendo desde los 10 puntos de exposición hasta los 400 aproximadamente. El algoritmo utilizado desprecia aquellos vectores de escaneo que sean más cortos de 100 puntos porque no se considera que tengan la suficiente cantidad de información como para ser estadísticamente relevantes y aquellos vectores de mayor longitud se partirán de forma que se extraigan la mayor cantidad de ventanas de 100 puntos sin solape entre ellas. Con esta forma de partir cada capa, se obtienen aproximadamente 550 ventanas de 100 valores por cada capa.

4.4.2. Igualación de longitudes

El siguiente paso es la reducción de dimensionalidad de las series temporales obtenidas, sin embargo algunas de las técnicas que se utilizarán solo son capaces de tratar con series de longitud constante. De las tres escalas descritas arriba, solo la más pequeña genera series de longitud constante, mientras que las otras dos su longitud depende de los parámetros de impresión del cubo al que correspondan. Para solucionar este problema se van a probar 3 métodos diferentes de igualación de longitudes:

- **Down Sample:** Consiste en eliminar datos de forma equitativa a lo largo de la serie temporal. Con este propósito se diseñó un algoritmo que elimina datos equiespaciados en tres pasadas sucesivas dejando la serie temporal lo más cerca posible de la longitud final deseada. En cada una de las pasadas se elimina 1 de cada n datos, lo que equivale a multiplicar la longitud de la serie por $\frac{n-1}{n}$. Este problema se puede definir como un problema de minimización 4.2 y resolver con una búsqueda exhaustiva. Los datos sobrantes se truncan del final de la serie. Este método se diseñó con la idea de minimizar la cantidad de datos truncados a la vez que intentando eliminar los datos de la forma más uniforme posible a lo largo de la serie. Utilizando este

algoritmo y suponiendo que la serie inicialmente no sea más de 4 veces más larga que la longitud deseada, nunca deberán truncarse más de un 2% de los datos.

$$a \leq b \leq c \in [2, \dots, 100], L_f = L_i \left(\frac{a-1}{a} \cdot \frac{b-1}{b} \cdot \frac{c-1}{c} \right), L_f \geq L_d \quad \min_{a,b,c} L_d - L_f \quad (4.2)$$

- **Truncado:** Con este método simplemente se eliminarán el mismo número de datos de la parte inicial de la serie como de la final para dejar la serie de la longitud deseada. Este método es mucho más rápido que el anterior, pero pierde mucha información del principio y del final de la serie.
- **Padding:** Consiste en añadir ceros en la cola de la serie hasta que la serie tenga la longitud deseada.

Con el primer y segundo método todas las series se igualarán a la longitud de la serie más corta mientras que con el tercer método se igualarán todas con la serie más larga.

4.4.3. Reducción de dimensionalidad y extracción de características

Una de las principales formas de clasificar secuencias de datos o series temporales es la reducción de dimensionalidad [Fawaz, 2019], que consiste en “resumir” la información de la serie en un vector de dimensiones reducidas que pueda utilizarse fácilmente como entrada para algoritmos de aprendizaje automático [Xing et al., 2010]. Con este objetivo se van a implementar tres técnicas diferentes para tratar de encontrar la mejor representación de los datos para resolver el problema de clasificación.

- **SVD:** Esta es la técnica utilizada por Okaro et al. en su publicación [Okaro, 2019]. Se pretende utilizar como baseline para ver si otras de las propuestas mejoran su rendimiento. La idea principal detrás de SVD (singular value decomposition) es aprovechar la descomposición de una matriz cualquiera A , que en este caso será la matriz formada por todo el dataset, en el producto de tres matrices U , Σ , V de la siguiente forma:

$$A = U\Sigma V^T \quad (4.3)$$

Donde Σ es una matriz diagonal y U y V dos matrices ortogonales. Suponiendo que A es una matriz $m \times n$, U será $m \times m$, Σ será una matriz diagonal $n \times n$ y V será $n \times n$. De forma intrínseca esta descomposición no tiene ningún efecto de reducción dimensional, pero si se restringen los tamaños de las matrices de la descomposición

como se puede ver en la imagen 4.6, se obtiene una aproximación de la matriz A que conserva la mayor parte de la información. Usando el producto de ΣV^T en su forma reducida, se consigue una nueva representación de los datos que conserva la mayor cantidad de información posible y ha pasado de tener m dimensiones a tener K . Dado que se debe representar el dataset entero en forma de matriz rectangular, ésta es una de las técnicas que requiere que todas las series temporales del dataset tengan la misma longitud.

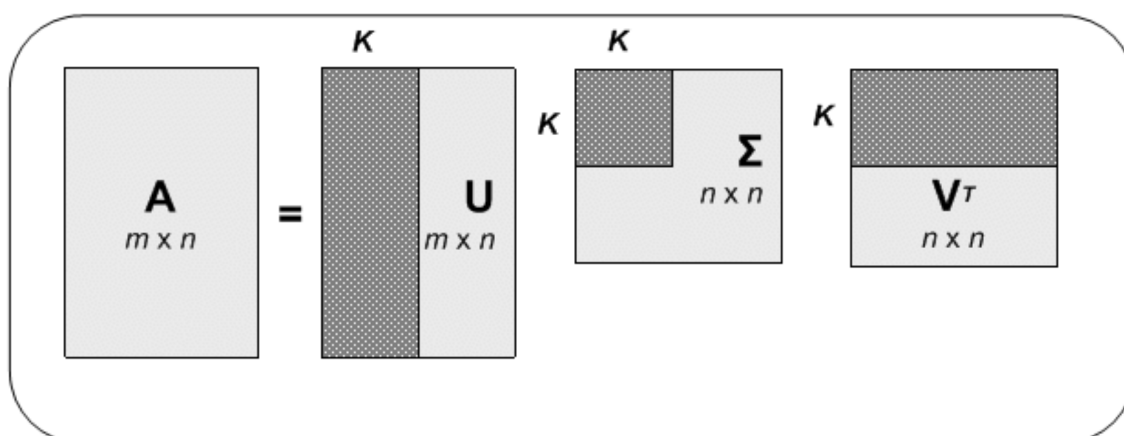


Figura 4.6: Visualización de la reducción de dimensionalidad en base a la descomposición SVD [Albatayneh et al., 2014].

Destacar que la reducción de dimensionalidad llevada a cabo con SVD es muy similar a la que se realiza al aplicar PCA (principal component analysis) que es una técnica mucho más común en literatura.

- Características estadísticas:** Otro método para obtener un vector de dimensiones reducidas a partir de una serie temporal es la extracción manual de características estadísticas. Existe multitud de características que se pueden utilizar y por lo general no existe un método estandarizado para seleccionar las mejores. La selección de las mejores características es el principal problema de esta técnica y el principal motivo por el que se suele recurrir a la extracción de características aprendidas por un modelo de aprendizaje automático. En este trabajo se van a utilizar las características descritas por [Jones et al., 2014] en su trabajo sobre la detección de anomalías en series temporales. En su trabajo Jones et al. utilizan un conjunto de 7 métricas estadísticas que resumen el comportamiento de la serie temporal. Estas

características son:

1. Media.
 2. Desviación estándar.
 3. Media de la diferencia absoluta entre valores de puntos contiguos $|z(t) - z(t+1)|$.
 4. Número de veces que la serie cruza la media dividido por la longitud de la serie
 5. Porcentaje de diferencias positivas entre valores de puntos contiguos
 6. Porcentaje de diferencias nulas entre valores de puntos contiguos
 7. Longitud media de las series de valores estrictamente positivos en la diferencia entre valores de puntos contiguos dividido por la longitud de la serie completa.
- **Autoencoders:** Como se mencionó en el apartado anterior, otra de las principales técnicas para la extracción de características es el uso de modelos de aprendizaje automático que aprenden una nueva representación de los datos en un espacio latente de dimensiones reducidas. Dentro de los métodos para aprender representaciones, los autoencoders son de las técnicas más utilizadas.

Un autoencoder es un tipo de red neuronal que se entrena de forma auto-supervisada. La red neuronal se entrena con técnicas de aprendizaje supervisado, pero los datos no necesitan estar etiquetados. Esto se debe a que el autoencoder intenta reconstruir en su salida lo mismo que se le pasó como entrada, por lo que la etiqueta con la que se compara la salida es la propia entrada.

La red neuronal de un autoencoder tiene dos partes diferenciadas, una red de entrada llamada codificador y una de salida llamada decodificador. El codificador se encarga de pasar los datos de entrada al espacio latente de representación reducida, mientras que el decodificador reconstruye los datos de entrada desde ese espacio latente. Aunque se hable de dos redes diferenciadas, durante el entrenamiento ambas redes están conectadas y se entrenan como una sola. Es una vez que el entrenamiento ha terminado, cuando se separan ambas redes y se descarta el decodificador ya que la parte interesante para esta aplicación es el codificador.

Este tipo de redes neuronales funcionan muy bien para obtener representaciones de datos ya que se está obligando a la red a reducir el tamaño de los datos para que puedan pasar por un cuello de botella a la vez que se debe conservar toda la información relevante de los mismos para obtener una reconstrucción lo más parecida

posible a la entrada. En la figura 4.7 se presenta un esquema de un autoencoder básico donde se puede distinguir el codificador, el decodificador, el cuello de botella así como el vector que representará a los datos.

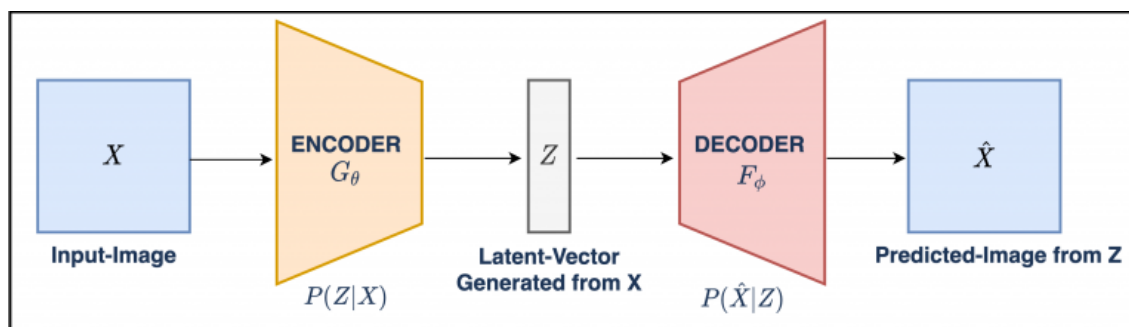


Figura 4.7: Esquema general de un Autoencoder [Sharma, 2021].

Existen muchas variantes de autoencoders. En este trabajo se van a implementar y testear tres implementaciones diferentes. Primero se probará un autoencoder clásico con una red neuronal densa. A pesar que las redes densas no sean las más utilizadas para el tratamiento de secuencias de datos, se quiere probar esta implementación como punto de partida para ver cuánto se puede mejorar el resultado con las otras implementaciones. En segundo lugar se implementará un autoencoder convolucional de 1 dimensión y por último se construirá un autoencoder recurrente con LSTMs.

Para el entrenamiento de los autoencoders, se separarán los datos en conjuntos de entrenamiento y validación y se escogerá la combinación de parámetros que mejor rendimiento tenga en el conjunto de validación. La métrica que se utilizará para el entrenamiento y la evaluación va a ser el error medio absoluto (MAE). No se utilizará un conjunto de test ni se llevará a cabo validación cruzada ya que no se necesita una métrica de la capacidad de generalización de los autoencoders, sino que simplemente se seleccionará el que mejor rendimiento tenga sobre el conjunto de datos en cuestión.

Destacar que tras experimentar con los autoencoders y la reducción de dimensionalidad con SVD se decidió utilizar un espacio latente para ambos casos de 5 dimensiones para cada una de las series temporales. Este número no se quiso reducir más, a pesar de que la variabilidad explicada por un PCA en la quinta componente ronda el 10^{-3} , para que fuese una comparación razonable con las características estadísticas seleccionadas a mano que tienen un espacio latente de 7 dimensiones.

4.5. Clasificación

Una vez llevada a cabo la extracción de características se obtienen datasets con entre 15 y 21 variables y entre 36 y 7 millones de ejemplos en función de las combinaciones del procesamiento de los datos. La escala a la que se consideran los datos es lo que va a marcar más diferencia en el tamaño de cada uno de los datasets ya que en cada uno de los cambios de escala se aumenta en un par de ordenes de magnitud el tamaño del dataset. El objetivo de la creación de todos estos datasets, es compararlos para encontrar la mejor combinación de pasos de preprocesamiento. Para evaluar el desempeño de los datasets, se entrenarán modelos de clasificación con cada uno y se tomarán las métricas del modelo como indicadores de la calidad de los datasets.

Para los datasets más grandes (escala de vectores de escaneo y escala de capas) la única técnica de clasificación utilizada va a ser una red neuronal densa de clasificación multiclase. Se considera que esta es una de las técnicas más adecuada debido a la gran cantidad de datos. Para los datasets de la escala del cubo completo, se van a probar varias técnicas. La primera será también una red neuronal de clasificación multiclase que permita comparar con las otras dos escalas. Después se experimentará con técnicas más adecuadas para el tamaño del dataset, Random forest, SVM y KNN.

Con el fin de comparar entre los diferentes modelos y los diferentes datasets es muy importante establecer una metodología de evaluación clara y robusta. En primer lugar es importante establecer cómo se van a seleccionar los conjuntos de entrenamiento, test y validación. Para todas las comparaciones, se va a utilizar la técnica de validación cruzada con $k = 8$, por lo que no se requiere la creación de un conjunto de validación. Para el conjunto de test, se van a seleccionar 10 cubos al azar que serán los mismos para todos los modelos. Es importante destacar que a la escala de los cubos la única opción es coger cubos enteros para el conjunto de test y cada uno de los folds de validación, pero a las otras dos escalas se debe tomar la decisión de hacerlo por cubos completos o de forma aleatoria. En este trabajo se van a coger cubos completos para los “folds” de validación y el conjunto de test ya que se considera que de no hacerse así, las similitudes entre los datos extraídos del mismo cubo podrían sobre-estimar la capacidad de generalización de los modelos. Con el fin de comprobar esta hipótesis se llevará a cabo un pequeño experimento en el que se compararán ambos métodos. En cuanto a la métrica de evaluación se utilizará la puntuación F1 media en las 5 clases. Se escogió la puntuación F1 ya que tiene buen

comportamiento cuando las clases están desbalanceadas, que es muy probable que sea el caso con los “folds” de validación.

Teniendo clara la metodología de evaluación, se van a realizar las siguientes comparaciones bajo las condiciones indicadas:

- **Mejor dataset a la escala de cubo completo:** Para esta comparación se va a crear una red neuronal básica que se entrenará por separado con cada uno de los datasets. La arquitectura de la red se seleccionará haciendo una búsqueda de hiper-parámetros tomando la media de la validación cruzada en todos los datasets como métrica a maximizar. De esta forma se espera que la arquitectura no favorezca a ninguno de los datasets en particular. En cuanto al entrenamiento, todos los datasets partirán de la misma arquitectura con la misma inicialización de los pesos. De los entrenamientos con cada uno de los datasets se recogerá la puntuación F1 de los datos de entrenamiento y validación cruzada haciendo la media sobre los 8 folds, y la puntuación F1 de los datos de test predichos con un nuevo modelo igual que los anteriores, pero entrenado con todos los datos utilizados en la validación cruzada (entrenamiento + validación).
- **Mejor dataset a la escala de capa completa:** Se procederá de la misma forma que en el caso anterior. Destacar que como se dijo antes los conjunto de test y validación se separarán tomando cubos completos. Matizando el proceso de búsqueda de hiper-parámetros tanto para esta comparación cómo la anterior, se probarán diferentes tamaños de red en cuanto a número de neuronas y de capas, se probarán diferentes optimizadores para el entrenamiento, se experimentará con capas “dropout” para controlar el “overfitting” así como con regularización L1 y L2. Para las activaciones se usarán “relu” para las capas intermedias y “softmax” para la capa final.
- **Mejor dataset a la escala de vectores de escaneo:** Se procederá de la misma forma que en las dos comparaciones anteriores.
- **Mejor método de igualación de longitudes:** Para llevar a cabo esta comparación se reutilizarán los datos de las comparaciones de la escala del cubo y de la capa completa (en la escala de los vectores de escaneo no hizo falta igualar las longitudes por lo que no hay datos útiles para esta comparación). Dado que los modelos entre-

nados a la escala del cubo no tienen porque tener la misma arquitectura que los de la capa completa, se realizarán dos comparaciones una para cada una de las escalas.

- **Mejor reducción de dimensionalidad:** Al igual que en la comparación anterior, se aprovecharán los modelos entrenados en las tres primeras comparaciones para determinar cual de todas las reducciones de dimensionalidad tiene un mejor rendimiento a cada una de las tres escalas.
- **Mejor modelo para la escala de los cubos:** Para esta comparación, se seleccionará el mejor de los datasets de la escala del cubo, y se entrenarán lo mejores modelos posibles utilizando las siguientes técnicas: Red neuronal densa, Random forest, Máquina de vector soporte y K vecinos más cercanos. Se seguirá el mismo procedimiento de validación y test que en anteriores comparaciones.
- **Mejor escala:** Tras determinar cual es la mejor reducción de dimensionalidad y la mejor forma de igualar la longitudes de las series si aplica, se compararán los datos del mejor dataset para cada una de las escalas, evaluando así cual de las tres escalas tiene mejor rendimiento.

Capítulo 5

Descripción de los resultados y Discusión

En el capítulo anterior se han detallado los pasos seguidos para el desarrollo del experimento; en este capítulo se van a detallar y discutir los resultados en cada una de las fases mencionadas.

5.1. Impresión de los cubos

De las 4 camas de impresión empleadas para este trabajo, la mitad se imprimieron sin problemas en el primer intento, mientras que las otras dos impresiones fallaron la primera vez. En el segundo intento se consiguieron imprimir las dos camas faltantes, pero hubo que desactivar 2 cubos en cada una de las camas para poder terminarlas. Sorprendentemente las dos impresiones fallidas se corresponden con las impresiones que contenían los cubos con parámetros de impresión más cercanos a los parámetros estándar, mientras que las camas con los cubos más extremos fueron las que no dieron problemas.

Tras analizar las imágenes tomadas durante el proceso, se observó que los cubos problemáticos fueron los mismos en todas las impresiones. En la imagen 5.1 se pueden ver los cubos problemáticos justo antes de desactivarlos. Se puede ver como su superficie es significativamente más rugosa que la de los demás cubos. Esta rugosidad fue la causante del fallo del primer intento ya que provocó la parada del recoater. En todos los casos se tratan de los dos primeros cubos con modo de fusión keyhole. Ambos cubos tenían la misma parametría en cada una de las impresiones y estaban colocados en la parte izquierda de la cama. A pesar de que los parámetros de impresión estaban diseñados para generar

defectología en las piezas, no se esperaban fallos catastróficos que impidiesen completar los cubos y menos teniendo en cuenta que los cubos de alto Keyhole se imprimieron sin problema alguno aun teniendo parámetros mucho más problemáticos.

La principal hipótesis que se maneja es que los fallos no se deben únicamente a los parámetros de impresión sino que es una combinación entre la colocación de los cubos dentro de la cama de polvo y los parámetros utilizados. Si se intentasen imprimir nuevamente estos cubos en otra zona de la cama de impresión, con menos cubos a su alrededor, las impresiones probablemente resultasen exitosas. Sin embargo en el marco de este trabajo no fue posible realizar estas impresiones por falta de tiempo y disponibilidad de la máquina.

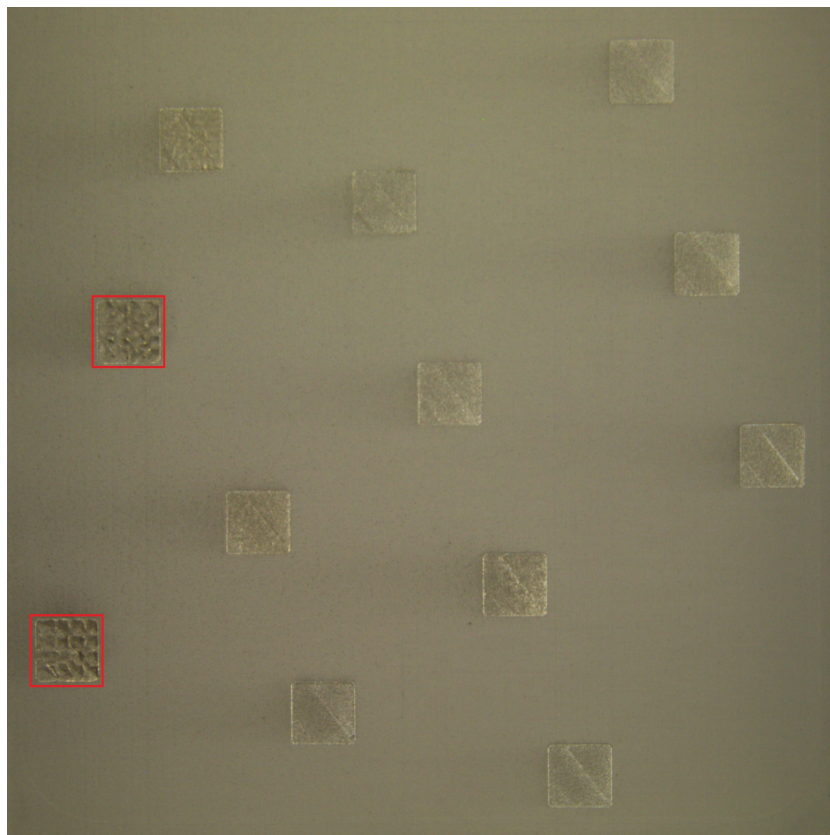


Figura 5.1: Imagen tomada durante la impresión de los cubos. Resaltados en rojo son los 2 cubos que tuvieron que ser desactivados para completar la impresión (Elaboración propia).

5.2. Caracterización de los cubos

Tras la impresión, los cubos fueron caracterizados con el fin de determinar si el modo de fusión que se pretendía conseguir fue efectivamente inducido. Para ello se midieron las

densidades de los cubos por medio del método de Arquímedes y se tomaron imágenes al microscopio de los cubos cortados. La determinación de densidad por Arquímedes consiste en medir el peso de los cubos dentro y fuera de un fluido de densidad conocida. En base a la diferencia de pesos y la densidad del fluido se puede determinar la densidad del objeto sumergido. Para evitar que el fluido penetre en los poros exteriores de la pieza, se aplica una laca que sella la porosidad superficial.

Los resultados obtenidos confirman para casi todos los cubos el modo de fusión pretendido. Sin embargo hay 4 cubos que no obtuvieron la densidad que cabría esperar según su modo de fusión teórico. En la tabla 5.1 se presentan los resultados obtenidos.

Es importante destacar que la porosidad inducida por la falta de fusión es mucho mayor que la producida por keyhole, por lo que es normal que los cubos de keyhole estén más cerca en cuanto a densidad de la parametría normal que los de falta de fusión. Aun teniendo esto en cuenta, se puede ver como los cubos 3 y 4 que se corresponden con keyhole tienen una densidad comparable con los cubos de parametría nominal y a su vez los cubos 5 y 8 que se imprimieron con parámetros nominales tienen una densidad más parecida a los de falta de fusión. Con el fin de ser lo más fieles posibles a la realidad, se decidió cambiar la etiqueta de los cubos mencionados a las clases que les corresponden por densidad. Por tanto, los cubos 3 y 4 pasan a pertenecer a la clase nominal y los cubos 5 y 8 a la clase de falta de fusión.

Un efecto secundario de este problema y de los fallos en la impresión de los cubos 1, 2, 21 y 22, es que ahora las clases se encuentran desbalanceadas. Dado que no hay posibilidad de imprimir más cubos para esa clase, se optó por continuar sin modificar nada más. Cuando se evalúe el desempeño del modelo final, se prestará especial atención a la clase keyhole que es la más afectada por el desbalanceo.

En cuanto a las micrografías, hay que destacar que se solicitaron al mismo tiempo que los ensayos de densidad, por lo que no se pudieron tener en cuenta estos resultados para seleccionar los cubos de mayor interés. A pesar de esto, sí que se tienen imágenes de la clase alto keyhole 5.2, alta falta de fusión 5.3 y nominal 5.4, en las que se ve claramente el tipo de porosidad inducida. Se puede ver cómo el alto keyhole ha generado poros esféricos y no excesivamente abundantes, mientras que alta falta de fusión ha generado una gran cantidad de poros de aspecto irregular.

Cubo	Impresion	Modo de fusión	Energía [J/mm^3]	Densidad [%]	Incertidumbre [%]
1	1	KH	112	X	X
2	1	KH	112	X	X
3	1	KH	95.2	99.1	0.2
4	1	KH	95.2	99.0	0.3
5	1	N	65.9	97.1	0.6
6	1	N	66.7	99.4	0.1
7	1	N	66.7	98.4	0.6
8	1	N	62.9	97.8	0.2
9	1	LF	39.2	93.5	0.3
10	1	LF	39.2	94.5	0.1
11	1	LF	44	95.5	0.3
12	1	LF	44	95.2	0.1
13	2	HKH	160	98.3	0.1
14	2	HKH	160	98.4	0.1
15	2	HKH	160	98.3	0.3
16	2	HKH	160	98.4	0.3
17	2	HLF	25.9	80.3	0.2
18	2	HLF	25.9	80.4	0.2
19	2	HLF	25.9	79.3	0.2
20	2	HLF	25.9	80.5	0.1
21	3	KH	112	X	X
22	3	KH	112	X	X
23	3	KH	100	98.2	0.3
24	3	KH	100	98.5	0.1
25	3	N	70	98.5	0.2
26	3	N	66.7	99.0	0.3
27	3	N	66.7	99.2	0.1
28	3	N	66	98.5	0.1
29	3	LF	41.7	95.4	0.1
30	3	LF	41.7	96.2	0.1
31	3	LF	44	98.1	0.1
32	3	LF	44	98.3	0.1
33	4	HKH	168	97.4	0.3
34	4	HKH	168	97.1	0.2
35	4	HKH	168	97.5	0.2
36	4	HKH	168	97.1	0.2
37	4	HLF	27.5	82.4	0.5
38	4	HLF	27.5	82.9	0.1
39	4	HLF	27.5	81.7	0.2
40	4	HLF	27.5	82.5	0.1

Tabla 5.1: Densidades de los cubos impresos. HKH: alto keyhole, KH: keyhole, N: nominal, LF: falta de fusión, HLF: alta falta de fusión.

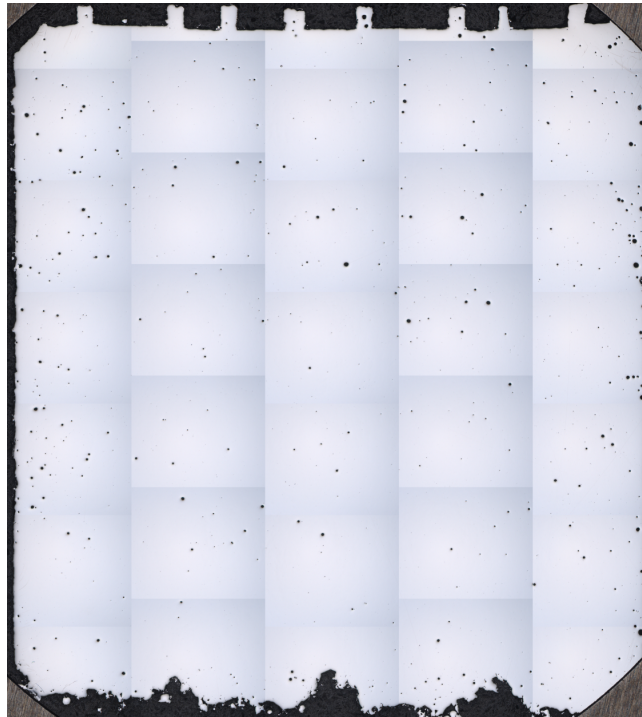


Figura 5.2: Micrografía del cubo 15 correspondiente a la clase alto keyhole (Elaboración propia).

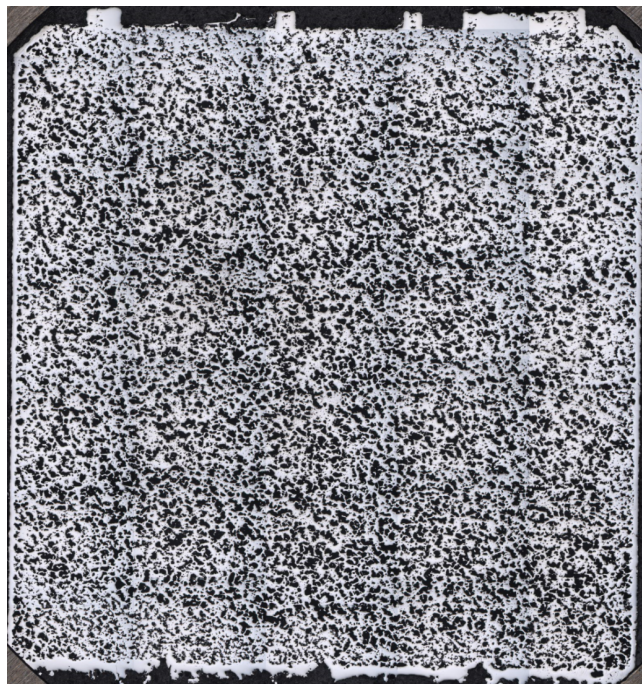


Figura 5.3: Micrografía del cubo 19 correspondiente a la clase alta falta de fusión (Elaboración propia).

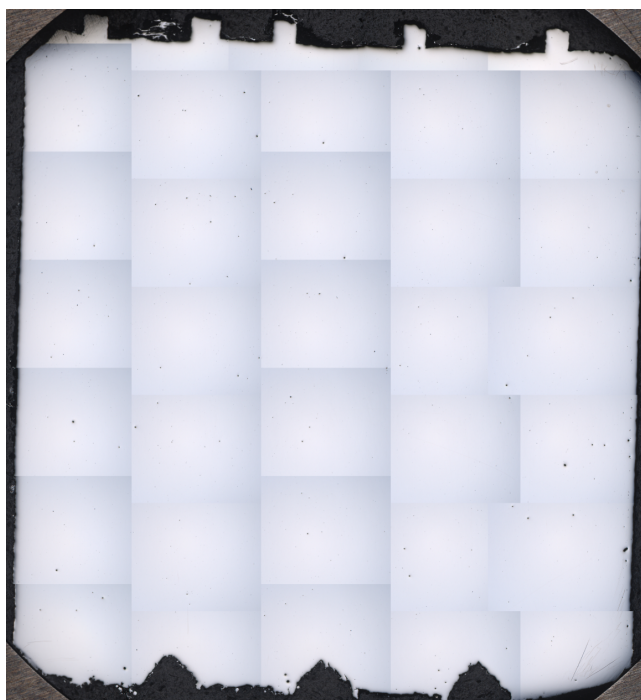


Figura 5.4: Micrografía del cubo 7 correspondiente a la clase Nominal (Elaboración propia).

5.3. Preprocesado y almacenamiento

Como se describe en el capítulo anterior, tras la recolección de los datos, se llevó a cabo el preprocesamiento de los mismos. Este proceso se realizó con éxito tal y como se describe en el capítulo anterior sin ningún tipo de contratiempo más allá del almacenamiento requerido para realizarlo .

De partida, los datos en bruto extraídos del software del fabricante ocupaban un total de 51GB. Tras llevar a cabo el primer paso de preprocesamiento de separación de la información en cubos y añadir a qué láser y capa pertenecía cada punto de exposición, los datos pasaron a ocupar 75GB. En el siguiente paso, en el que se eliminan los datos de los contornos, cabría esperar una reducción en el tamaño de los datos, pero esto no fue así ya que para llevar a cabo la segmentación por vectores de escaneo, se requirió calcular multitud de nuevas variables auxiliares. Por ello los datos pasaron a ocupar 150GB. La calibración de los fotodiodos no modificó el tamaño del conjunto de datos ya que simplemente variaba valores sin añadir nada nuevo. Destacar que el conjunto de datos ha llegado a ocupar 150GB debido a todas las variables auxiliares que se ha necesitado almacenar hasta este punto; en las siguientes partes del proceso estas variables ya no son necesarias por lo que

se pasará de 150GB a 22GB en total.

Tras el preprocesamiento y antes de la reducción de dimensionalidad que reducirá considerablemente el tamaño del conjunto de datos, se crearon múltiples versiones del dataset que combinaban diferentes segmentaciones de los datos con diferentes formas de igualar la longitud de las series, dando lugar a un total de 9 combinaciones distintas. Cada uno de estos 9 datasets contiene gran parte de la información del conjunto de datos total, por lo que el total de datos subió a 272GB. Destacar que no todos los conjuntos ocupan lo mismo. Los conjuntos que utilizan padding para la igualación de longitudes ocupan más del doble que los que truncan los datos, mientras que los que no aplican igualación se quedan en un punto intermedio.

Tras todos estos pasos, se aplicaron las diferentes técnicas de reducción de dimensionalidad. Combinando los 9 conjuntos de datos del paso anterior con las 5 técnicas de reducción de dimensionalidad seleccionadas se obtuvieron 18 datasets diferentes que ocuparon en total 6GB. Destacar que solo se obtuvieron 18 datasets ya que no todas las técnicas de reducción de dimensionalidad pudieron aplicarse a todas las segmentaciones. Estas limitaciones se comentarán con más detalle en el siguiente apartado.

Dada la naturaleza cambiante del proyecto, y el tiempo necesario para recalcular cada uno de los pasos, se decidió almacenar todos los pasos intermedios, por lo que se almacenaron un total de 600GB para el proyecto. Esta cantidad excedía el almacenamiento disponible en el ordenador de ejecución por lo que hubo que ampliarla a mitad del proyecto.

5.4. Segmentación a diferentes escalas e igualación de las longitudes

Como se explicó en el capítulo anterior, algunas de las técnicas de reducción de la dimensionalidad que se van a implementar en siguientes pasos requieren que todas las series del dataset tengan la misma longitud. Concretamente lo requieren todas menos la extracción de características estadísticas. Por ello cada una de las escalas de segmentación se combinará con todas las técnicas de igualaciones de longitud posibles incluida la no igualación. En la tabla 5.2 se recogen estas combinaciones junto con los nombres que se le dio a cada una.

	Under Sampling	Truncado	Padding	No igualación
Escala del cubo	S1	S2	S3	S4
Escala de capas	S5	S6	S7	S8
Escala vectores	NA	NA	NA	S9

Tabla 5.2: Combinación entre escala e igualación de longitud

Los datos segmentados a la escala de los vectores de escaneo no se les aplica ninguna técnica de igualación ya que la forma de segmentarlos ya produce este efecto.

Cada una de estas segmentaciones tiene un tamaño diferente:

- S1 \rightarrow [25000000, 36, 3]
- S2 \rightarrow [25000000, 36, 3]
- S3 \rightarrow [62000000, 36, 3]
- S4 \rightarrow [25000000 - 62000000, 36, 3]
- S5 \rightarrow [60000, 13536, 3]
- S6 \rightarrow [60000, 13536, 3]
- S7 \rightarrow [150000, 13536, 3]
- S8 \rightarrow [60000 - 150000, 13536, 3]
- S9 \rightarrow [100, 7177627, 3]

El primero de los números expresa la longitud de cada una de las series, el segundo el número de ejemplos en el dataset y el tercero hace referencia a que se tienen 3 series temporales para cada uno.

5.5. Reducción de la dimensionalidad

Tras la segmentación de los datos y la igualación de las longitudes es el turno de la reducción de dimensionalidad o extracción de características. Como ya se mencionó en el capítulo anterior, se van a implementar 5 técnicas distintas con el fin de encontrar la más adecuada para el problema que se está tratando. Cada una de ellas se va a combinar con varios de los datasets generados en el apartado anterior: S1-S9. Solo una de las 5

técnicas de reducción de dimensionalidad consideradas es capaz de trabajar sobre datasets con tamaños de series variables, la extracción de características estadísticas. Teniendo esto en cuenta, las técnicas que requieren una longitud constante solo se podrán aplicar sobre los datasets que cumplan esta propiedad, por razones obvias; sin embargo la técnica de extracción manual de características se aplicará solo a las segmentaciones que no tienen aplicada la igualación de longitudes a pesar de que también pueda ser aplicada al resto. Esto se decidió hacer así ya que al no aplicar ninguna igualación de longitud, el método tendrá acceso a los datos menos adulterados.

En la tabla 5.3 se recogen las combinaciones entre los datasets segmentados S1-S9 y las técnicas de reducción de dimensionalidad aplicadas sobre cada uno. NA indica que no se intentó aplicar esa técnica con esa segmentación. Los × indican que se intentó aplicar la técnica pero no se pudo, mientras que los ✓ significa que se aplicó satisfactoriamente.

	SVD	NN AE	CNN AE	RNN AE	SC
S1	✓	×	×	×	NA
S2	✓	×	×	×	NA
S3	✓	×	×	×	NA
S4	NA	NA	NA	NA	✓
S5	✓	✓	✓	×	NA
S6	✓	✓	✓	×	NA
S7	✓	✓	×	×	NA
S8	NA	NA	NA	NA	✓
S9	✓	✓	✓	✓	✓

Tabla 5.3: Combinaciones entre técnicas de reducción de dimensionalidad y segmentaciones.

5.5.1. Autoencoders

Como se puede observar en la tabla 5.3 las únicas técnicas de reducción de dimensionalidad que no se pudieron implementar como se tenía planeado fueron algunos autoencoders. Esto se debió en la mayoría de los casos a problemas de memoria a la hora de inicializar y entrenar los modelos.

En el caso de las segmentaciones S1, S2 y S3 los problemas de memoria son muy marcados ya que al tener cada una de las series como poco 25 millones de elementos, el tamaño de los modelos se vuelve excesivamente grande y al intentar entrenarlos, el sistema

se queda sin memoria. Teniendo acceso a un sistema con mayor cantidad de memoria es posible que este acercamiento fuese viable, pero como se verá en la comparación entre datasets, muy probablemente no compensase.

En el caso de S7 con el autoencoder convolucional, el entrenamiento de la red nunca llegó a fallar por falta de memoria del sistema, sin embargo la inicialización del modelo tomaba un tiempo que no se consideró razonable. Tras varios intentos, el más largo de ellos de 30 horas, el modelo nunca llegó a comenzar el entrenamiento y se decidió darlo como fallido.

Respecto a los autoencoders recurrentes de la escala de las capas, el entrenamiento si se pudo comenzar en el caso de S5 y S6, pero al cabo de pocas épocas de entrenamiento los valores de loss divergían, dando lugar a NaN. Se cree que esto es debido al problema de “exploding gradient”. Para solucionarlo se aplicó truncado de gradientes y disminución del learning rate, sin éxito, y consecuentemente se marcó también como fallido.

Destacar que antes de llegar a la comparación de los diferentes datasets en función de su capacidad de clasificación, se puede hacer un breve análisis de cuál de los tres tipos de autoencoders dio mejores métricas de reconstrucción. Este análisis solo se puede llevar a cabo en la segmentación S9 ya que es la única sobre la que se pudieron crear los 3 tipos de autoencoders. En la tabla 5.4 se muestran las métricas de error de reconstrucción de los tres autoencoders para cada una de las series temporales.

	NN	CNN	RNN
láser	0.0002	0.0010	0.0009
Visible	0.0315	0.0244	0.0214
Infra-rojo	0.0109	0.0081	0.0069

Tabla 5.4: Valores de las métricas de reconstrucción MAE sobre el conjunto de test de las tres series temporales y los tres tipos de autoencoders

Haciendo la media de los datos de la tabla se ve claramente como el que mejor resultado obtiene es la RNN, a pesar de que la NN sea la mejor en el caso de la serie temporal del láser. Hay que destacar que por la arquitectura de las redes, el autoencoder NN son en realidad 3 autoencoders, uno por cada serie temporal, mientras que los autoencoders RNN y CNN son un único modelo para las 3 series. Este hecho se cree que puede haber permitido que el autoencoder NN sea el que mejor rendimiento tuvo en la serie del láser mientras que el peor en el resto.

5.6. Comparación entre datasets

Tras la creación de los 18 datasets definitivos, se procede a compararlos entre sí en busca del más adecuado para la tarea de clasificación. Como se detalló en el capítulo anterior, la mayoría de modelos utilizados para las comparaciones van a ser redes neuronales densas de clasificación multiclase. Para que las comparativas sean lo más representativas posibles todos los datasets que pertenezcan a la misma escala se compararán utilizando modelos con la misma arquitectura e inicialización. Entre diferentes escalas se consideró que no era conveniente utilizar la misma arquitectura, ya que al aumentar el número de datos en el dataset se hará posible entrenar modelos más grandes y profundos sin que el “overfitting” sea un problema. Por ello, solo se compararán datasets de diferentes escalas en la comparación entre escalas en la que se intentará comparar los mejores modelos posibles de cada una de las escalas.

En cuanto al conjunto de test, se seleccionaron de forma aleatoria 10 cubos que se utilizarán para la evaluación de todos los modelos que se entrenen en este trabajo. Los cubos escogidos son: 8, 9, 11, 14, 15, 16, 23, 28, 36 y 37. A pesar de hacerse de forma aleatoria, se restringió la selección a conjuntos de test que tuviesen al menos un cubo de cada una de las clases, para que sean representativos del dataset completo. En concreto el conjunto de test escogido contiene, 1 cubo con parametría nominal, 1 cubo con parametría de keyhole, 1 cubo con parametría de alta falta de fusión, 3 cubos de falta de fusión y 4 cubos de alto keyhole.

5.6.1. Comparación a la escala de los cubos

En este apartado se comparan los cuatro datasets de la escala de los cubos. Para ello se entrenaron cuatro modelos iguales con cada uno de los datasets y se recopilaron las puntuaciones F1 para el conjunto de entrenamiento, test y media de validación cruzada.

Para determinar la arquitectura y parámetros del modelo se realizó una búsqueda de hiper-parámetros intentando maximizar la media de las puntuaciones en la validación cruzada de los 4 datasets. La arquitectura final seleccionada consiste en dos capas ocultas de 50 y 25 neuronas respectivamente con activaciones relu, dos capas dropout con un 25 % de probabilidad después de cada una de las capas ocultas. La capa de entrada tiene 15 neuronas para los tres primeros datasets y 21 para el cuarto. Como salida se utiliza una capa de 5 neuronas con activación softmax para la clasificación multiclase. En cuanto al

entrenamiento se utilizó como optimizador Adam con “learning rate” de 10^{-3} , 500 épocas de entrenamiento y un “batch size” de 8. En la figura 5.5 se presenta un resumen de la arquitectura.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 50)	800
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 25)	1275
dropout_1 (Dropout)	(None, 25)	0
dense_2 (Dense)	(None, 5)	130
Total params: 2,205		
Trainable params: 2,205		
Non-trainable params: 0		

Figura 5.5: Arquitectura modelo utilizado para la comparación de datasets en la escala del cubo (Elaboración propia).

Los resultados obtenidos se recogen en la tabla 5.5. Como se puede observar, de los 4 datasets, el que mejor resultado ha obtenido es el que combina la segmentación S4 con la extracción manual de características estadísticas. Supera a los otros datasets en los tres conjuntos de datos, entrenamiento, validación y test, con bastante margen.

Segmentacion	Reducción	Train	K-fold	Test
S1	SVD	0.864	0.567	0.600
S2	SVD	0.828	0.467	0.700
S3	SVD	0.921	0.581	0.500
S4	SC	0.992	0.656	0.900

Tabla 5.5: Métricas F1 de los modelos entrenado con cada uno de los datasets de la escala del cubo

En la figura 5.6 se presenta el historial de entrenamiento de la arquitectura mencionada con el dataset S4. Este entrenamiento se hizo con todos los datos de entrenamiento y validación cruzada y usando el conjunto de test como “validación”. En el caso de los otros datasets se observó mucho más overfitting a partir de las 200 épocas, pero ni utilizando “early stopping” o mayor dropout se consiguieron mejores resultados que con S4 y la características estadísticas.

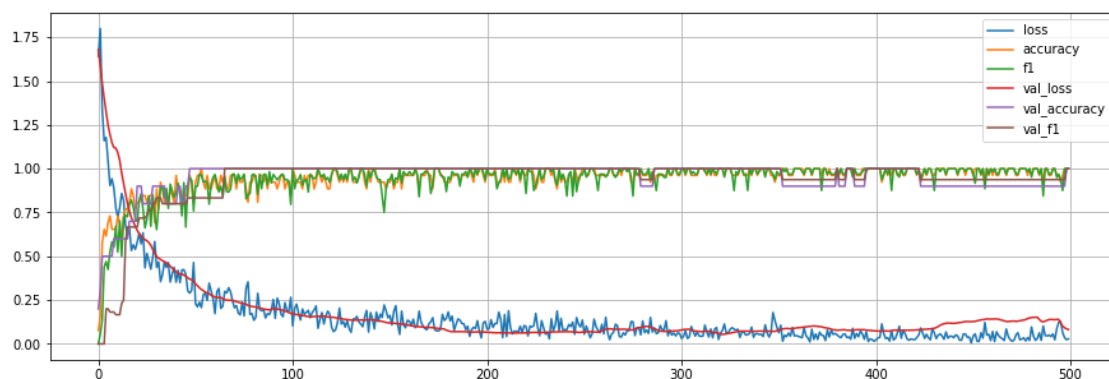


Figura 5.6: Entrenamiento del mejor modelo a la escala del cubo (Elaboración propia).

5.6.2. Comparación a la escala de las capas

Siguiendo el mismo procedimiento de la comparación anterior, se estudiaron los datasets de la escala de las capas. El modelo final utilizado tras la búsqueda de hiper-parámetros se presenta en la figura 5.7. Es muy similar al utilizado en la comparación anterior pero aumentando el tamaño de la red. Las capas ocultas pasan a ser de 200 y 100 neuronas respectivamente, la probabilidad de dropout pasa a ser del 50%, el batch size de 1024, las épocas de entrenamiento 200 y el resto de parámetros de entrenamiento y arquitectura se dejan igual.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 200)	3200
dropout (Dropout)	(None, 200)	0
dense_1 (Dense)	(None, 100)	20100
dropout_1 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 5)	505
Total params: 23,805		
Trainable params: 23,805		
Non-trainable params: 0		

Figura 5.7: Arquitectura modelo utilizado para la comparación de los datasets en la escala de las capas (Elaboración propia).

Los resultados obtenidos por los modelos se encuentran en la tabla 5.6. Los resultados son muy similares a los obtenidos en la escala de los cubos, el mejor de los datasets

Segmentacion	Reducción	Train	K-fold	Test
S5	NN AE	0.9766	0.7453	0.9997
S5	CNN AE	0.9613	0.7278	0.9988
S5	SVD	0.9858	0.8035	0.9904
S6	NN AE	0.9783	0.7960	0.9991
S6	CNN AE	0.9573	0.8001	0.9985
S6	SVD	0.9876	0.8058	0.9941
S7	NN AE	0.9999	0.5976	0.9997
S7	SVD	0.9999	0.5493	0.9994
S8	SC	1.000	0.8105	0.9830

Tabla 5.6: Métricas F1 de los modelos entrenados con cada uno de los datasets de la escala de las capas

en entrenamiento y validación cruzada es el dataset creado con la segmentación S8 y la extracción de características estadísticas. En este caso este mismo dataset es el peor en el conjunto de test con una puntuación F1 de 0,9830 mientras que el resto llegan hasta los 0,9997. Sin embargo, a pesar de no ser el mejor en test, el propósito del conjunto de test no es la selección del mejor dataset, sino tener una estimación de la capacidad de generalización del modelo por lo que se seleccionó S8 como mejor dataset en base a la validación cruzada.

En la figura 5.8 se presenta nuevamente la evolución de las métricas durante el entrenamiento con el mejor dataset. En este caso no sucede como en la escala anterior, y ninguno de los datasets exhibe un overfitting excesivo.

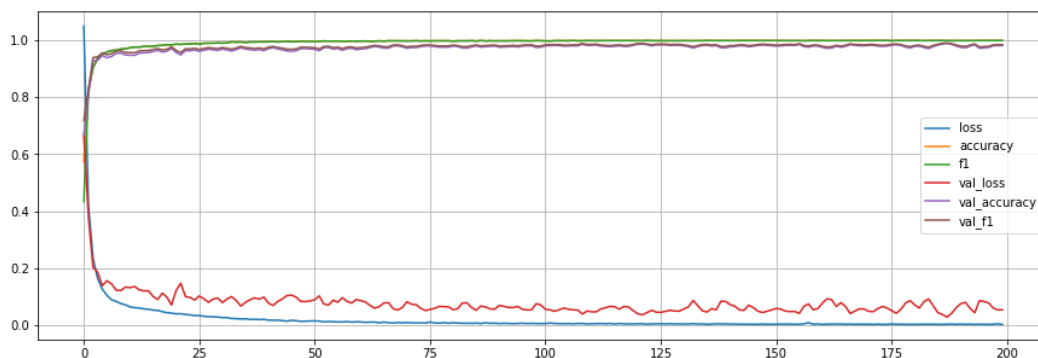


Figura 5.8: Entrenamiento del mejor modelo a la escala de las capas (Elaboración propia).

5.6.3. Comparación a la escala de los vectores de escaneo

Nuevamente se sigue el mismo procedimiento de las comparaciones anteriores. El modelo seleccionado pasa a tener tres capas ocultas de 1000, 500, 100 neuronas respectivamente, dropout se baja nuevamente al 25 %, el batch size pasa a ser de 8192, se entrenan durante 20 épocas y el resto de parámetros se mantienen igual.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1000)	16000
dropout (Dropout)	(None, 1000)	0
dense_1 (Dense)	(None, 500)	500500
dropout_1 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 100)	50100
dropout_2 (Dropout)	(None, 100)	0
dense_3 (Dense)	(None, 5)	505
Total params: 567,105		
Trainable params: 567,105		
Non-trainable params: 0		

Figura 5.9: Arquitectura modelo utilizado para la comparación de los datasets en la escala de los vectores de escaneo (Elaboración propia).

Segmentacion	Reducción	Train	K-fold	Test
S9	NN AE	0.9801	0.8205	0.9431
S9	CNN AE	0.9777	0.8205	0.9380
S9	RNN AE	0.9786	0.8227	0.9351
S9	SVD	0.9821	0.8240	0.9501
S9	SC	0.9804	0.8332	0.9567

Tabla 5.7: Métricas F1 de los modelos entrenados con cada uno de los datasets de la escala de los vectores de escaneo

Los resultados de la tabla 5.7 muestran como nuevamente el mejor dataset en el conjunto de validación cruzada es el creado con las características estadísticas. En esta comparación los resultados son muy similares entre todos los datasets, pero aún así el último dataset es el que mejores métricas obtuvo.

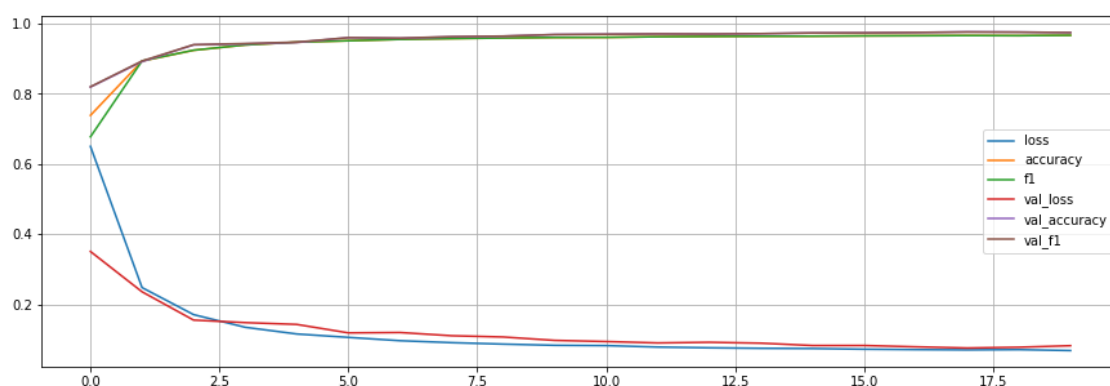


Figura 5.10: Entrenamiento del mejor modelo a la escala de los vectores de escaneo (Elaboración propia).

5.6.4. Comparación entre métodos de igualación de longitudes

Para realizar esta comparación no se entrenaron nuevos modelos, sino que se aprovecharon los datos de las dos primeras comparaciones. Observando los datos de la tabla 5.5, se puede observar como under-sampling (S1) y padding (S3) tienen un rendimiento en validación cruzada muy similar entre sí, mientras que truncado (S2) es considerablemente peor. En cambio, en la tabla 5.6, haciendo la media de los modelos con el mismo tipo de igualación de longitud, se observa que truncar (S6) es superior a under-sampling (S5) y este a su vez es superior a padding (S7). Los datos de este análisis se compilan en la tabla 5.8. Cada posición de la tabla corresponde con el valor medio de puntuación F1 de validación cruzada de todos los datasets que usasen esa técnica de igualación de longitudes a la escala marcada.

	Cubo	Capa
Under Sample	0.567	0.759
Truncado	0.467	0.800
Padding	0.581	0.573

Tabla 5.8: Media de las puntuaciones F1 de validación cruzada de los datasets que se corresponden con la escala y método de igualación de longitud marcados.

Los datos de la escala de los cubos y la escala de las capas no concuerdan entre sí a la hora de decidir cual es el mejor método. Esto puede deberse a que la técnica de igualación de longitud tiene cierta interacción con la escala y por tanto puede ser mejor utilizar una

diferente para cada escala, aunque también puede deberse a la incertidumbre experimental. De todas las formas, no se cree que se pueda sacar una conclusión significativa de los datos analizados.

5.6.5. Comparación entre métodos de reducción de dimensionalidad

Al igual que en la comparación anterior, no se entrenaron nuevos modelos para llevar a cabo esta comparación, sino que se usaron los datos de la segunda y tercera comparación.

De forma general, viendo los resultados de las tres primeras comparaciones, está claro que el mejor método para el problema que se trata en este trabajo es la extracción de características estadísticas elegidas a mano. Sin embargo hasta ahora no se ha estudiado si hay diferencias entre el resto de métodos aplicados.

La escala ideal para llevar a cabo esta comparación es la escala de los vectores de escaneo, ya que es la única que tiene todos los métodos de reducción de dimensionalidad y además no se aplicaron métodos de igualación de longitud que podría confundir el análisis. Sin embargo, las diferencias entre datasets en esta escala son demasiado pequeñas como para considerarlas representativas. Por lo tanto se deberá recurrir a los datos de la comparación entre datasets a la escala de las capas para sacar alguna conclusión.

Observando los datos de la tabla 5.6 se puede ver como en todos los casos, menos para la segmentación S7, el segundo mejor método es SVD seguido por NN AE y cerrando con CNN AE. En el caso de S7, NN AE supera a SVD y CNN AE no pudo realizarse. En S9, SVD también obtiene mejores resultados que los tres métodos de autoencoders, por lo que se puede decir con cierta seguridad que este método es el segundo mejor de los probados.

En cuanto a los autoencoders, al igual que la comparación de igualación de longitudes, se considera que los resultados no son suficientemente significativos como para dar una conclusión definitiva que diferencie entre todos ellos.

5.6.6. Comparación entre técnicas de clasificación para la escala de los cubos

De los resultados expuestos hasta ahora se puede ver como los peores resultados obtenidos son los de la escala de los cubos. Esto posiblemente se deba a que las redes neuronales no son el modelo más indicado cuando el dataset no es suficientemente grande. Para explorar esta posibilidad se probaron otros algoritmos de clasificación sobre el mejor dataset de la escala de los cubos y sus resultados se comparan a continuación.

Se probaron 3 métodos sin contar las redes neuronales: random forest, máquinas de vector soporte y k vecinos más cercanos. Para cada uno de ellos se realizó una búsqueda de hiper-parámetros por el método “grid search”. El modelo de random forest utilizado tiene, 50 estimadores, utiliza el criterio “gini”, cada árbol de decisión tiene una profundidad máxima de 50 “splits” y el resto de parámetros estándar de Scikit-learn. Para el modelo de máquina de vector soporte, los parámetros seleccionados fueron, $C = 1000$, kernel “rbf”, gamma automática y el resto de parámetro los estándar de Scikit-learn. Por último para el modelo de k vecinos más cercanos se usaron, 1 solo vecino y como métrica de distancia, la distancia Manhattan.

Los resultados obtenidos se recogen en la tabla 5.9.

	Train	K-fold	Test
NN	0.992	0.656	0.900
RF	1.000	0.840	1.000
SVM	0.937	0.626	0.649
KNN	1.000	0.833	1.000

Tabla 5.9: Puntuación F1 de cada uno de los métodos aplicados al mejor dataset de la escala de los cubos.

Se puede observar como los mejores resultados en validación cruzada los obtienen random forest y k vecinos más cercanos, mientras que la red neuronal y la máquina de vector soporte tienen un rendimiento mucho peor pero similar entre ambos métodos.

Como se sospechaba, a esta escala, la red neuronal no es el mejor modelo posible debido probablemente a la baja cantidad de datos disponibles.

Observando estos resultados cabe preguntarse si random forest también podría tener mejores resultados para el resto de escalas. Por ello, se entrenaron modelos de random forest para cada uno de los mejores datasets de las dos escalas restantes. Los resultados se comparan con el rendimiento de la red neuronal en la tabla 5.10.

Los resultados de la comparación muestran como en la escala de las capas, random forest es mejor que la red neuronal, pero al pasar a la escala de los vectores de escaneo la red neuronal obtiene resultados mucho mejores. Por ello se considera que la red neuronal es el modelo superior al trabajar con el dataset S9 SC.

		Train	K-fold	Test
S8 SC	NN	1.0000	0.8105	0.9830
S8 SC	RF	1.0000	0.8274	0.9426
S9 SC	NN	0.9804	0.8332	0.9567
S9 SC	RF	1.0000	0.7642	0.8542

Tabla 5.10: Comparación de random forest con NN para las escalas de las capas y de los vectores de escaneo

5.6.7. Comparación entre escalas

Por último queda comparar cual de las tres escalas tuvo un mejor rendimiento. Para ello en la tabla 5.11 se recopilan las puntuaciones F1 de los mejores modelos para cada una de las escalas. Se puede apreciar como las escalas de las capas y los vectores de escaneo superan con creces a la escala de los cubos obteniendo como poco una mejora del 20% en la puntuación F1 de la validación cruzada. Comparando ahora la escala de las capas y la de los vectores se encuentra que la diferencia ya no es tan pronunciada, aun así la escala de los vectores de escaneo obtiene una puntuación ligeramente mejor.

Segmentacion	Reducción	Train	K-fold	Test
S4	SC	0.9920	0.6560	0.9000
S8	SC	0.9999	0.8105	0.9830
S9	SC	0.9804	0.8332	0.9567

Tabla 5.11: Comparación entre las métricas F1 de los mejores datasets de cada una de las escalas.

5.6.8. Comparación entre validaciones cruzadas

Como se comentaba en el capítulo de metodología del trabajo, se considera que el proceso de validación cruzada debe realizarse tomando cubos completos incluso en las escalas en las que se trabaja con trozos de cubo más pequeños. En [Shevchik et al., 2019] y [Shevchik et al., 2018] el proceso de evaluación de los modelos no se hace de esta forma ya que en esos trabajos los datos provienen todos del mismo cubo. Se cree que la forma en que llevaron a cabo de validación shevchik et al. sobrestima considerablemente las capacidades de generalización del modelo. Para validar esta hipótesis, se entrenaron dos

modelos idénticos sobre el dataset S9 con características estadísticas, con la única diferencia entre ambos en la forma en que se realiza la validación cruzada. Los datos se recogen en la tabla 5.12.

	Train	K-fold	Test
Validación por cubo	0.9804	0.8332	0.9567
Validación aleatoria	0.9784	0.9784	0.9573

Tabla 5.12: Comparación del mismo modelo con validación cruzada por cubo y validación cruzada aleatoria en el dataset S9 SC

Como se puede observar, los valores de entrenamiento y test son casi iguales mientras que la validación cruzada es mucho mejor en la validación cruzada aleatoria. Con estos datos, se cree validada la hipótesis de que la validación cruzada aleatoria sobrestima considerablemente los resultados al entrenar y validar sobre datos del mismo cubo.

5.7. Modelo final y discusión

Los datos expuestos hasta este punto ponen de manifiesto cómo el mejor enfoque de todos los implementados es el uso de características estadísticas seleccionadas a mano. Esto va en contra de las tendencias actuales del campo, en las que el uso de representaciones aprendidas está tomando una gran relevancia para la clasificación de series temporales. Esta contradicción puede deberse simplemente a que el conjunto de características extraídas de [Jones et al., 2014] son especialmente efectivas para el problema tratado. En las escalas con menor cantidad de datos también es posible que el tamaño del dataset jugase un rol importante que no permitiese crear representaciones aprendidas de calidad. Esta hipótesis tiene cierto respaldo de los datos, ya que según se estudiaban escalas con mayor cantidad de datos en el dataset, menor era la diferencia entre el método manual y los autoencoders.

La comparación con el método de SVD propuesto para este tipo de problemas por [Okaro, 2019] y [Jayasinghe et al., 2020] es también importante ya que indica si el método desarrollado mejora soluciones ya existentes. En este aspecto, SVD estuvo consistentemente por encima de los autoencoders pero al mismo tiempo siempre estuvo por debajo de la selección manual de características. Por tanto se considera que la solución de preprocesamiento encontrada es superior a la existente en bibliografía para este problema.

Respecto a la igualación de longitudes, dado que no se obtuvo una diferencia significa-

tiva entre los métodos se considera que la mejor alternativa es el truncado por ser la más sencilla de implementar y que menor cantidad de recursos computacionales consume.

Por otro lado, es posible que los autoencoders hubiesen tenido mejores resultados si se hubiesen acoplado a la red neuronal de clasificación y se hubiera permitido la modificación de sus pesos durante el entrenamiento. De esta forma se hubiera hecho “fine-tuning” al encoder para mejorar su rendimiento en el problema concreto. Esta alternativa se considera interesante, por lo que se incluirá en el apartado de trabajos futuros al no ser posible incluirla en el presente trabajo por falta de tiempo.

En referencia a la obtención de los datos, se considera que existen ciertas limitaciones en el dataset construido. El principal fallo que se ha producido es el desbalanceo de las clases. Concretamente, la clase referente a keyhole tiene muchos menos datos que otras clases como puede ser falta de fusión. Este fallo no se cree que se deba a fallos metodológicos del trabajo sino a problemas sobrevenidos durante la ejecución de los experimentos. Concretamente, aún no se tiene una explicación para el fallo de impresión de los 4 cubos de la clase keyhole, pero se cree que se solucionaría simplemente imprimiendo más cubos de esa clase en una cama de polvo separada.

5.7.1. Modelo final

Teniendo en cuenta todo lo aprendido en las comparaciones anteriores se procedió a crear el modelo final sobre el dataset S9 SC. Vista la comparación entre random forest y red neuronal, se decidió crear una red neuronal de clasificación multiclase. La arquitectura se creó en base a una búsqueda exhaustiva de hiper-parámetros en la que se compararon 9 arquitecturas diferentes con más y menos neuronas y capas. La arquitectura final tiene tres capas ocultas de 1000, 500 y 100 neuronas respectivamente. Todas las activaciones son relus exceptuando la de la última capa, que tiene 5 neuronas con activación softmax. Entre cada una de las capas ocultas se incluyó una capa dropout con probabilidad de desactivación del 50% con el fin de controlar el overfitting.

Para el entrenamiento se siguió utilizando el optimizador Adam con learning rate de 10^{-3} , batch size de 8192 y 40 épocas de entrenamiento.

En cuanto a la validación cruzada, se decidió mejorar la precisión del método usando una especie de “leave-one-out cross validation”. En cada uno de los folds se incluirá únicamente un cubo como conjunto de validación. De esta forma se tendrá la mejor estimación posible del conjunto de validación aunque a costa de un mayor tiempo de ejecución. No

se trata de “leave-one-out cross validation” estricta, ya que eso implicaría utilizar cada dato individual para crear un fold. El conjunto de test sigue siendo el mismo que se lleva utilizado durante todo el trabajo.

Los resultados obtenidos son bastante similares a los que ya se habían obtenido en las comparaciones anteriores:

- **Puntuación F1 entrenamiento:** 0,9730
- **Puntuación F1 validación cruzada:** 0,8431
- **Puntuación F1 test:** 0,9699

	Predicción				
	HLF	LF	N	KH	HKH
HLF	128567	2484	0	0	0
LF	25293	783136	178	0	0
N	0	42960	88088	0	0
KH	0	1	5014	126022	5
HKH	0	253	440	0	1350000

Tabla 5.13: Matriz de confusión del modelo final sobre el conjunto de test

En la tabla 5.13 se recoge la matriz de confusión del modelo sobre los datos de test. Observando la diagonal principal se puede ver como la mayoría de predicciones se han realizado correctamente, sobre todo para las clases más extremas que tienen tasas de errores bastante bajas. Observando las matrices triangular inferior y superior, se puede ver como en general el modelo cuando falla, lo hace a una clase contigua, lo cual es muy deseable. También se puede ver como el modelo tiende a subestimar la energía, ya que hay muchas más instancias de errores a la clase inferior que a la clase superior.

Por otro lado, como se comentaba en la discusión el hecho de que el dataset tenga muy pocos ejemplos de la clase de keyhole claramente está generando problemas en el modelo. Se puede ver como el modelo tiende a no predecir keyhole. Esto se ve claramente en las predicciones de la clase de alto keyhole, que se esperaría que tuviese cierto número de fallos que predijesen keyhole, pero como se puede ver en la tabla esto no es así. Sin embargo esto no es porque alto keyhole no se confunda con otras clases, ya que hay errores en las clases nominal y falta de fusión. El modelo está confundiendo antes alto keyhole con falta de fusión que con keyhole. Esto claramente no es deseable.

Para obtener más información sobre los errores del modelo y sobre todo para averiguar por qué el modelo tiene un desempeño mucho peor en validación cruzada que en test, se recogieron las matrices de confusión sobre el conjunto de validación en cada uno de los pasos de la validación cruzada. De esta forma se pudo crear la tabla 5.14 con todos los cubos de entrenamiento en la que se muestra con qué clases se confunde cada uno de ellos cuando no se ha introducido este cubo en el conjunto de entrenamiento.

Cubo	Clase	HLF	LF	N	KH	HKH	Accuracy
3	N	0	72	406491	15	0	0.9998
4	N	0	71	406495	11	0	0.9998
5	LF	0	0	121512	0	58	0
6	N	0	918	279520	27	0	0.9966
7	N	0	173	280290	0	0	0.9994
10	LF	1	120791	781	0	0	0.9936
12	LF	280381	84	4	0	0	0.0003
13	HKH	0	23	3442	17	403096	0.9914
17	HLF	120137	1438	1	0	0	0.9882
18	HLF	116065	5512	0	0	0	0.9547
19	HLF	120226	1349	0	0	0	0.9889
20	HLF	120055	1520	0	0	0	0.9875
24	KH	0	0	130908	0	133	0
25	N	0	130820	0	0	224	0
26	N	0	11	130633	402	0	0.9969
27	N	0	18	129697	1329	0	0.9897
29	LF	0	130695	346	0	0	0.9974
30	LF	2	130592	451	0	0	0.9965
31	LF	15	131028	2	0	0	0.9999
32	LF	9	131034	1	0	0	0.9999
33	HKH	0	0	2	0	131042	0.9999
34	HKH	0	3	0	0	131037	0.9999
35	HKH	0	2	24	0	131016	0.9998
38	HLF	119036	12010	2	0	0	0.9083
39	HLF	128216	2829	0	0	0	0.9784
40	HLF	128425	2616	0	0	0	0.9800

Tabla 5.14: Matriz de confusión confeccionada a partir de las matrices de confusión de todos los folds de validación cruzada.

Se puede observar como la tasa de acierto de casi todos los cubos está por encima del 99% salvo en 4 cubos en los que el la tasa de acierto es casi del 0%. Haciendo una cuenta

rápida suponiendo que el modelo acertase 22 de los 26 cubos con un 100% de acierto y los otros 4 con un acierto del 0% se obtendría una tasa de acierto en validación cruzada de 84,61%. Este valor es muy cercano al valor obtenido y es muy probable que no se pueda superar hasta que no se resuelva el problema de esos 4 cubos.

No se tiene una explicación fiable para los fallos de 3 de los 4 cubos, sin embargo, para el cubo 24 si que se descubrió el motivo. Tras los fallos de impresión y los cambios de etiqueta por las densidades de los cubos, la clase de keyhole se quedó únicamente con dos cubos, 23 y 24. El cubo 23 fue seleccionado para el conjunto de test, ya que se forzó a que hubiese al menos 1 cubo de cada clase en el conjunto de test, dejando al cubo 24 como único representante de la clase en el conjunto de entrenamiento. Habiendo un solo cubo de la clase keyhole en el conjunto de entrenamiento, al llevar a cabo la validación cruzada en el paso del cubo 24, el entrenamiento se lleva a cabo sin datos de la clase keyhole lo que imposibilita que pueda aprenderla.

Teniendo en cuenta todos los problemas ocasionados por la clase keyhole se decidió eliminarla para comprobar los resultados en ese caso. Los dos cubos restantes de keyhole se etiquetaron con clase nominal al igual que se hizo con los cubos 3 y 4 y se procedió a repetir el entrenamiento.

Los resultados con el nuevo dataset fueron muy similares en entrenamiento y test, pero significativamente mejores en validación cruzada:

- **Puntuación F1 entrenamiento:** 0,9723
- **Puntuación F1 validación cruzada:** 0,8754
- **Puntuación F1 test:** 0,9768

	Predicción			
	HLF	LF	N	HKH
HLF	128039	3012	0	0
LF	18955	789520	132	0
N	0	36230	225854	6
HKH	0	229	491	1349973

Tabla 5.15: Matriz de confusión del modelo final sobre el conjunto de test sin keyhole

Observando la matriz de confusión sobre el conjunto de test 5.15 se puede observar como el comportamiento del modelo ha mejorado considerablemente. Sigue tendiendo a

subestimar la clase, pero al tener las clases mejor balanceadas ya no hay comportamientos extraños en cuanto a los errores y las clases contiguas. La única clase que sigue teniendo un comportamiento no demasiado deseable respecto a las confusiones con clases no contiguas, es la de alto keyhole, que sigue confundándose a veces con falta de fusión.

Por otro lado, analizando las métricas de cada una de las clases por separado, se puede ver como la clase que peor se predice es la nominal. Concretamente se alcanzó una tasa de aciertos del 86,2%.

Cubo	Clase	HLF	LF	N	HKH	Accuracy
3	N	0	80	406498	0	0.9998
4	N	0	84	406493	0	0.9998
5	LF	0	0	121523	47	0.0000
6	N	0	919	279546	0	0.9967
7	N	0	140	280323	0	0.9995
10	LF	8	120979	586	0	0.9951
12	LF	280281	183	5	0	0.0007
13	HKH	0	22	4959	401597	0.9877
17	HLF	120100	1475	1	0	0.9879
18	HLF	116708	4869	0	0	0.9600
19	HLF	120137	1438	0	0	0.9882
20	HLF	119952	1623	0	0	0.9867
24	N	0	1	130395	645	0.9951
25	N	0	130835	0	209	0.0000
26	N	0	17	130611	433	0.9966
27	N	0	15	131029	0	0.9999
29	LF	0	130669	372	0	0.9972
30	LF	1	130611	433	0	0.9967
31	LF	11	131027	7	0	0.9999
32	LF	6	131037	1	0	0.9999
33	HKH	0	0	0	131044	1.0000
34	HKH	1	1	0	131038	1.0000
35	HKH	0	2	26	131014	0.9998
38	HLF	120784	10262	2	0	0.9217
39	HLF	127945	3100	0	0	0.9763
40	HLF	128507	2534	0	0	0.9807

Tabla 5.16: Matriz de confusión confeccionada a partir de las matrices de confusión de todos los folds de validación cruzada para el nuevo dataset.

Al igual que antes, se recoge en la tabla 5.16 las confusiones durante el proceso de validación cruzada. Esta tabla es muy similar a la anterior, con la principal diferencia que el cubo 24 ahora se predice correctamente. Eso si, los otros 3 cubos problemáticos (5, 12, 25) siguen sin predecirse correctamente.

El cubo 5 pertenecía originalmente a la clase nominal, pero que fue cambiado de clase a falta de fusión por sus valores de densidad no aceptables para la clase nominal, sin embargo, el modelo lo ha clasificado como clase nominal al 99,99%. El cubo 12 pertenece a la clase de falta de fusión sin embargo el modelo lo clasifica casi totalmente como alta falta de fusión. Por último el cubo 25 está en la situación contraria al 5, pertenece a la clase nominal y por densidad estuvo a punto de cambiarse a la clase de falta de fusión que es la que predice el modelo. Estos errores del modelo sugieren que podría ser útil revisar las definiciones de las clases para asegurar su consistencia. Esto se propondrá como trabajo futuro ya que no da tiempo a tratarlo en el presente trabajo.

Cabe destacar que a pesar de estos 3 cubos problemáticos, el resto del dataset funciona como se esperaba. Podría parecer que el modelo tiene demasiados errores entre clases contiguas, pero esto era esperable al trabajar en una escala tan local. Al trabajar con segmentos de cubo tan pequeños, cualquier pequeña fluctuación térmica en el proceso puede provocar condiciones de fusión ligeramente diferentes a las del cubo completo. Si estas fluctuaciones son suficientemente grandes es normal y deseable que el modelo prediga esas partes del cubo como que tienen un modo de fusión de otra clase contigua, porque en el fondo la tiene. Por ello se considera que el rendimiento final del modelo es bueno.

Comparar el modelo obtenido con otras publicaciones es complicado ya que no hay ninguna que resuelva exactamente la misma tarea. En [Okaro, 2019] se utiliza el mismo tipo de datos, pero la tarea a resolver no es la misma. Sin embargo en el procesamiento de los datos si que se superó su metodología para el problema que se trata en este trabajo, lo cual no quiere decir que la metodología propuesta aquí, superase a la suya para su tarea. En [Shevchik et al., 2018] sí que resuelven la misma tarea pero partiendo de un tipo de monitorización distinta, emisiones acústicas, que podría considerarse menos directa y por tanto más difícil. El clasificador presentado en este trabajo si que supera el 85,66% de accuracy que reportan shevchik et al. en su trabajo.

Capítulo 6

Conclusiones y Trabajo Futuro

En este trabajo se pretendía dar solución a uno de los problemas que lastra la fabricación aditiva; la falta de repetibilidad y garantías sobre de la calidad de las piezas. Para intentar resolver este problema, se ha creado un modelo de clasificación capaz de diferenciar entre 4 modos de fusión diferentes en base a los datos de monitorización in-situ del proceso, con una tasa de acierto mayor al 86,2%. Los modos de fusión estudiados están estrechamente relacionados con la calidad de impresión de las piezas, ya que 3 de ellos están asociados con los principales mecanismos de creación de porosidad en piezas impresas. La porosidad es a su vez uno de los principales indicadores de la calidad de impresión. Por ello se considera que el modelo creado ayuda a asegurar la calidad de las piezas impresas, mitigando así la posible incertidumbre del proceso de impresión.

Con la creación de este modelo se ha cumplido con el objetivo principal de este trabajo. Sin embargo, en el capítulo 3 se plantearon una serie de objetivos parciales que se discuten a continuación:

- En primer lugar se pretendía identificar aquellos trabajos en los que se abordasen problemas similares. Si bien no se encontró ningún trabajo que abordase el mismo problema, se encontraron otros que tratan con el mismo tipo de fuente de datos para resolver una tarea similar, así como trabajos que usaban una fuente diferente de datos, pero que sí que resolvían la misma tarea de aprendizaje automático. De entre toda la bibliografía consultada no se tiene constancia de ningún trabajo con las mismas características que el presente; por ello se considera que este objetivo ha sido cumplido al encontrar los trabajos más parecidos disponibles. La comparación con los mismos no es directa por la falta de homogeneidad, pero se considera que el

trabajo presentado supera en la metodología de preprocesado a [Okaro, 2019] y en los resultados de clasificación a [Shevchik et al., 2018].

- En segundo lugar se pretendía estudiar la fuente de datos. A este respecto, se ha hecho un estudio exhaustivo y suficiente de la fuente de datos, identificando todas las variables involucradas, así como la naturaleza física de las mismas.
- En tercer lugar se pretendía seleccionar y evaluar diferentes métodos de procesamiento de los datos. En primer lugar se han seleccionado las técnicas de preprocesamiento y extracción de características más adecuadas en base a las técnicas encontradas en bibliografía. Posteriormente se ha llevado a cabo una exhaustiva comparación entre diferentes técnicas de preprocesamiento así como de diferentes técnicas de reducción de dimensionalidad. El resultado de esta comparación ha sido la selección de la extracción de características estadísticas propuesta por [Jones et al., 2014] así como la segmentación de los datos en ventanas de 100 valores de longitud.
- En cuarto lugar se planteaba la creación del modelo de clasificación que superase el 80 % de clasificación, que como se adelantaba al comienzo de las conclusiones se ha conseguido utilizando una red neuronal de clasificación multiclase.
- Por último se planteaba la evaluación detallada del desempeño del modelo. A este respecto, a partir del análisis de las puntuaciones F1 en validación cruzada y test, así como de las matrices de confusión, se identificaron aquellas clases con peor desempeño, así como los cubos que impedían que el modelo mejorase sus métricas.

A pesar de que todos los objetivos planteados han sido cumplidos, el trabajo deja líneas de investigación abiertas que deberían explorarse en siguientes iteraciones.

6.1. Trabajo futuro

- Una de las primeras dificultades encontradas durante el desarrollo del presente trabajo y que debería tratar de solucionarse en trabajos futuros, son los problemas de cantidad de datos encontrados con la clase keyhole. Originalmente se diseñaron 8 cubos para esta clase al igual que para el resto, pero durante la impresión fallaron 4 de ellos y tras la caracterización se vio que otros dos no tenían suficiente porosidad como para calificarlos como keyhole. Finalmente se decidió eliminar la clase e incluir

los dos cubos restantes en la clase nominal, ya que parecía ser la más cercana. Sin embargo se considera que esta solución es únicamente válida en el contexto de este trabajo dadas las limitaciones de tiempo y que en futuras iteraciones se deberían generar más datos de esta clase para balancear correctamente el dataset.

- Aparte de los problemas encontrados con la clase keyhole, también se detectaron ciertas inconsistencias con algunos cubos concretos de otras clases. Esto sugiere que si bien el dataset en su totalidad parece ser coherente, quizás haya determinadas parametrías y clases que habría que depurar en futuras revisiones del dataset.
- Otra posible expansión mencionada en el cuerpo del trabajo es la combinación de los autoencoders dentro de la arquitectura del clasificador de forma que se obtenga una solución “end-to-end”. Una vez entrenado el autoencoder, podría crearse un modelo de clasificación sobre el que se llevase a cabo “transfer learning” del encoder a las primeras capas del modelo. De esta forma se espera que la representación aprendida por el encoder pueda ser depurada para adecuarse más a la tarea concreta que se intenta resolver.
- Por último se considera muy interesante la implementación de este modelo en un entorno de producción real, en el que pudieran validarse la precisión y calidad de las predicciones generadas fuera del ambiente de laboratorio.

Capítulo 7

Bibliografía

- [Abanda et al., 2019] Abanda, A., Mori, U., and Lozano, J. A. (2019). A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412.
- [Additively, 2021] Additively (2021). <https://www.additively.com/en/learn-about/3d-printing-technologies>. Accedido 26/01/2021.
- [Al Faruque et al., 2016] Al Faruque, M. A., Chhetri, S. R., Canedo, A., and Wan, J. (2016). Acoustic Side-Channel Attacks on Additive Manufacturing Systems. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, Vienna, Austria. IEEE.
- [Albatayneh et al., 2014] Albatayneh, N., Imran, K., and Chua, F.-F. (2014). A semantic content-based forum recommender system architecture based on content-based filtering and latent semantic analysis. volume 287, pages 369–378.
- [Alfaify, 2019] Alfaify, A. Y. M. (2019). The Effect of Changing Particle Size Distribution and Layer Thickness on the Density of Parts Manufactured Using the Laser Powder Bed Fusion Process. page 209.
- [Aoyagi et al., 2019] Aoyagi, K., Wang, H., Sudo, H., and Chiba, A. (2019). Simple method to construct process maps for additive manufacturing using a support vector machine. *Additive Manufacturing*, 27:353–362.
- [ASTM, 2015] ASTM, I. (2015). Astm52900-15 standard terminology for additive manufacturing—general principles—terminology. *ASTM International, West Conshohocken, PA*, 3(4):5.

- [Bagnall et al., 2017] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31(3):606–660.
- [Bagnall et al., 2015] Bagnall, A., Lines, J., Hills, J., and Bostrom, A. (2015). Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.
- [Cang et al., 2018] Cang, R., Yao, H., and Ren, Y. (2018). One-Shot Generation of Near-Optimal Topology through Theory-Driven Machine Learning. *arXiv:1807.10787 [cs, stat]*. arXiv: 1807.10787.
- [Carpenter Additive, 2020] Carpenter Additive (2020). Comparison of plasma and electrode inert gas atomized powder. Technical report.
- [Eschenauer and Olhoff, 2001] Eschenauer, H. A. and Olhoff, N. (2001). Topology optimization of continuum structures: A review*. *Applied Mechanics Reviews*, 54(4):331–390.
- [Everton et al., 2016] Everton, S. K., Hirsch, M., Stravroulakis, P., Leach, R. K., and Clare, A. T. (2016). Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, 95:431–445.
- [Fawaz, 2019] Fawaz, H. I. (2019). Deep learning for time series classification: a review. page 44.
- [Frazier, 2014] Frazier, W. E. (2014). Metal Additive Manufacturing: A Review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928.
- [Gu et al., 2018] Gu, G. X., Chen, C.-T., and Buehler, M. J. (2018). De novo composite design based on machine learning algorithm. *Extreme Mechanics Letters*, 18:19–28.
- [Jayasinghe et al., 2020] Jayasinghe, S., Paoletti, P., Sutcliffe, C., Dardis, J., Jones, N., and Green, P. (2020). Automatic Quality Assessments of Laser Powder Bed Fusion Builds from Photodiode Sensor Measurements. Publisher: Preprints.
- [Johnson, 2020] Johnson, N. S. (2020). Invited Review__ Machine Learning for Materials Developments in Metals Additive Manufacturing. page 44.

- [Jones et al., 2014] Jones, M., Nikovski, D., Imamura, M., and Hirata, T. (2014). Anomaly detection in real-valued multidimensional time series. In *International Conference on Bigdata/Socialcom/Cybersecurity. Stanford University, ASE. Citeseer*. Citeseer.
- [Kumar, 2020] Kumar, S. (2020). *Additive Manufacturing Processes*. Springer International Publishing, Cham.
- [Ma et al., 2018] Ma, W., Cheng, F., and Liu, Y. (2018). Deep-Learning-Enabled On-Demand Design of Chiral Metamaterials. *ACS Nano*, 12(6):6326–6334. Publisher: American Chemical Society.
- [Mathieu et al., 2014] Mathieu, M., Henaff, M., and LeCun, Y. (2014). Fast Training of Convolutional Networks through FFTs. *arXiv:1312.5851 [cs]*. arXiv: 1312.5851.
- [Meng, 2020] Meng, L. (2020). Machine Learning in Additive Manufacturing: A Review. page 15.
- [Metel et al., 2018] Metel, A., Stebulyanin, M., Fedorov, S., and Okunkova, A. (2018). Power Density Distribution for Laser Additive Manufacturing (SLM): Potential, Fundamentals and Advanced Applications. *Technologies*, 7(1):5.
- [Michalski and Michalski, 2001] Michalski, L. and Michalski, L., editors (2001). *Temperature measurement*. J. Wiley, Chichester ; New York, 2nd ed edition.
- [Okaro, 2019] Okaro, I. A. (2019). Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing*, page 12.
- [Renishaw, 2021] Renishaw (2021). Infiniam spectral. <https://www.renishaw.es/es/infiniam-spectral--42310>.
- [Scime and Beuth, 2018a] Scime, L. and Beuth, J. (2018a). Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing*, 19:114–126.
- [Scime and Beuth, 2018b] Scime, L. and Beuth, J. (2018b). A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 24:273–286.
- [Scime et al., 2020] Scime, L., Siddel, D., Baird, S., and Paquit, V. (2020). Layer-wise anomaly detection and classification for powder bed additive manufacturing processes:

- A machine-agnostic algorithm for real-time pixel-wise semantic segmentation. *Additive Manufacturing*, 36:101453.
- [Senvol, 2021] Senvol (2021). <http://senvol.com/material-search/>. Accedido 26/04/2021.
- [Sharma, 2021] Sharma, A. (2021). <https://learnopencv.com/variational-autoencoder-in-tensorflow/>. Accedido 26/06/2021.
- [Shevchik et al., 2018] Shevchik, S. A., Kenel, C., Leinenbach, C., and Wasmer, K. (2018). Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks. *Additive Manufacturing*, 21:598–604.
- [Shevchik et al., 2019] Shevchik, S. A., Masinelli, G., Kenel, C., Leinenbach, C., and Wasmer, K. (2019). Deep Learning for In Situ and Real-Time Quality Monitoring in Additive Manufacturing Using Acoustic Emission. *IEEE Transactions on Industrial Informatics*, 15(9):5194–5203. Conference Name: IEEE Transactions on Industrial Informatics.
- [Susto et al., 2018] Susto, G. A., Cenedese, A., and Terzi, M. (2018). Time-series classification methods: Review and applications to power systems data. *Big data application in power systems*, pages 179–220.
- [Tang et al., 2017] Tang, M., Pistorius, P. C., and Beuth, J. L. (2017). Prediction of lack-of-fusion porosity for powder bed fusion. *Additive Manufacturing*, 14:39–48.
- [Tapia and Elwany, 2014] Tapia, G. and Elwany, A. (2014). A Review on Process Monitoring and Control in Metal-Based Additive Manufacturing. *Journal of Manufacturing Science and Engineering*, 136(060801).
- [Wang, 2020] Wang, C. (2020). Machine learning in additive manufacturing__ State-of-the-art and perspectives. *Additive Manufacturing*, page 20.
- [Wasmer et al., 2019] Wasmer, K., Le-Quang, T., Meylan, B., and Shevchik, S. A. (2019). In Situ Quality Monitoring in AM Using Acoustic Emission: A Reinforcement Learning Approach. *Journal of Materials Engineering and Performance*, 28(2):666–672.
- [Wohlers et al., 2020] Wohlers, T., Associates (Firm), W., Campbell, R., Diegel, O., Huff, R., and Kowen, J. (2020). *Wohlers Report 2020: 3D Printing and Additive Manufacturing State of the Industry*. Wohlers Associates.

- [Xing et al., 2010] Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48.
- [Ye et al., 2018] Ye, D., Hong, G. S., Zhang, Y., Zhu, K., and Fuh, J. Y. H. (2018). Defect detection in selective laser melting technology by acoustic signals with deep belief networks. *The International Journal of Advanced Manufacturing Technology*, 96(5):2791–2801.
- [Zhang et al., 2017] Zhang, W., Mehta, A., Desai, P. S., and Iii, C. F. H. (2017). Machine learning enabled powder spreading process map for metal additive manufacturing (AM). pages 1235–1249.

Quality prediction of additive manufacturing parts via in-situ monitoring

Alberto Rodríguez Álvarez

Universidad Internacional de la Rioja, Logroño (España)

July 22 2021

ABSTRACT

One of the main problems faced by metal additive manufacturing techniques is the lack of process reliability and quality assurance. This work presents a multiclass classification model able to distinguish between different fusion modes found during printing of metal additive manufacturing parts. Fusion modes are closely related to part quality, which enables the use of this model as a quality assurance technique. The input data for the model is captured with a coaxial optical monitoring system integrated in an L-PBF printer. A dataset is created based on 40 metal cubes specially design to induce the desired fusion modes. A data processing pipeline is created based on tests and comparisons among the many different alternatives. Different dimensionality reduction techniques are applied and compared in search for the best performing one for the task. These techniques include autoencoders, SVD (singular value decomposition) and manual extraction of statistical components. The final model achieves at least 86.1% accuracy on all classes and shows the feasibility of using this methodology to detect part quality on a L-PBF machine.

I. INTRODUCTION

Additive manufacturing (AM) is described as the use of a computer aided design to construct an object in a layer by layer fashion [1]. At first it was designed as a polymer technology, but later metal printing technologies were developed which had a much higher industrial potential [2]. AM has been received in the last decades with great excitement by the scientific and industrial community due to its unique properties, such as its never before seen freedom of design, manufacturing times and personalization capabilities [3]. Nevertheless this excitement has also found some bottlenecks that have set back its adoption in some industries. One such problem is its low repeatability and difficulty to assure the quality of printed parts[4]. Aiming to address these problems many methodologies have been developed, from standardization of the process, to part certification, but one of these methodologies has gotten a lot of traction in recent years; defect detection through process monitoring. It is right in this line of research where this work can be included.

All the experiments for this work have been conducted on a commercial Laser powder bed fusion

system which has been fitted with an in-situ monitoring suite. The monitoring suite consists of three coaxial photodiodes for each laser, that capture the electromagnetic emissions produced by the fusion process. The data is captured at a 100KHz sampling rate which is then preprocessed by the manufacturer's software and output at a variable rate in the form of three time series.

The objective of this study is to create a classifier able to distinguish among the different fusion modes that take place during printing. These fusion modes are closely related with the mechanisms that produce porosity on metal additively manufactured parts. Porosity is one of the most important indicators of printing quality, so with this model it should be possible to create a system that predicts part quality while printing.

In order to train the described model, a dataset was created. A series of 40 cubes with different process parameters were printed and monitored. The process parameters of each cube were deliberately tuned to produce all the desired fusion modes.

Before any work could be done on this data, it had to be cleaned and correctly organized. Three steps were performed to this end. Firstly, data had



KEY WORDS

Metal additive manufacturing, Dimensionality reduction, Multiclass classification, Temporal series classification, Quality assurance.

to be organized by cube, as that is the way labels are assigned. Then data from the contours had to be eliminated, as they are printed with different parameters which do not produce the desired fusion modes. And lastly, as the printer used has four lasers, data from each laser had to be calibrated so that they yielded comparable results.

One of the main challenges that was tackled in this study is the creation of a processing methodology that would extract the most value out of the data. For this, different techniques were implemented and compared against each other over their performance in the classification task. The first step in the processing pipeline consisted of separating the data into individual samples. Three different scales were tested, cube scale, layer scale, and hatch scale. After that, four distinct length-equalization techniques were implemented. This step is not optional, as some of the techniques in the following steps require all samples of the dataset to be the same length. Lastly, five dimensionality reduction methods were tested.

After all the different combinations among all processing steps had been implemented a total of 18 datasets were compared to find the one that yielded the best classification performance. Once this dataset had been selected it was used to create a fine-tuned model which was then compared to other solutions found in literature.

The present study is organized as follows. In chapter 2 the state of the art relating to this study's field is reviewed. In chapter 3, the objectives for this study are stated. After that, in chapter 4, all the experiments are detailed and its results reviewed in chapter 5. Finally in chapter 6 conclusions are drawn and future work is proposed.

II. STATE OF THE ART

A Additive manufacturing

The ASTM defines additive manufacturing as the process of adding material to form an object following a 3D design, usually layer by layer [1]. The ASTM also defines 7 additive manufacturing technologies; Binder Jetting, Direct Energy Deposition, Material Extrusion, Material Jetting, Powder Bed Fusion, Sheet Lamination and Vat Photopolymerization.

This work is centered around the process of Powder Bed Fusion, more concretely, Laser Powder Bed Fusion (L-PBF) of steel. This technology uses a laser source to selectively melt thin layers of metal powder laid on top of each other. These kind of

printers have three main modules that enable the printing process. The first one is the power source and the way it is directed to the powder bed. In the case of L-PBF systems, the power source is a high energy laser which is pointed to the desired region of the powder bed through a series of lenses and mirror galvanometers. This system ensures that the energy is deposited at the right time in the right place. The second module is in charge of the powder bed. This system is composed of a recoater which spreads a thin layer of powder over the top of the last one, and a mechanism that lowers the powder bed before each recoating process. The layer height is determined by how much the powder bed is lowered. The recoater has many forms, but the most common is a blade that pushes a pile of powder over the lowered powder bed leaving the right amount of powder behind. The last system is in charge of the atmospheric control and gas flow. It is really important to have control over the atmospheric composition inside the build chamber as metals oxidize very quickly at high temperatures. The gas flow system creates a laminar flow of inert gas over the powder bed that carries away any by-products of the fusion process.

As most manufacturing processes, L-PBF systems have many parameters that have to be tuned in order to achieve the desired outcome. Given the relative maturity of the process, there is some consensus about the most important process parameters, which according to Kumar et al. are; laser power, scanning speed, hatch distance and layer height [5]. Laser power and layer height are very straightforward to understand, one controls how much energy the laser deposits per unit of time while the other controls the amount of powder that constitutes a layer. Scanning speed refers to how fast the laser spot moves across the powder bed surface. In combination with the laser power it can control how much energy is deposited per unit of length. Hatch distance is closely related to how each layer is printed. To create a 3D part with a laser point it is necessary to divide the part into layers first, and then into scanning vectors. In each layer, the path followed by the laser is normally formed by many straight lines that cover the entire part. The separation between these lines is what is called Hatch distance. It is important to note, that the described parameters are the ones used in continuous waved lasers. The printer used for this study uses a modulated laser which changes some of the parameters. With a modulated laser, instead of making each hatch line as a continuous line where the laser is never turned off, the hatches are created as many aligned exposure points. This means that instead

of selecting a speed at which the laser travels, three new parameters have to be selected; exposure time, point distance and jump time. The first one determines how long will the laser be turned on at each exposure point. The second one refers to the distance between two consecutive exposure points. And lastly, the jump time selects how long the laser will be off between two consecutive exposure points.

B Monitoring

Even with some of the latest advances in metal additive manufacturing, stability and quality assurance are still lagging behind some other manufacturing technologies [6, 7]. One of the methods that aims to solve the quality assurance problem, is in-situ process monitoring. These techniques enable the extraction of key process indicators that can be used to estimate the quality of a part during printing. According to Tapia et al., the main magnitude to monitor in L-PBF is temperature [6]. For this purpose Tapia et al. distinguish two families of methods, contact measurement with thermocouples and non-contact measurements with pyrometers. Tapia et al. remark that non-contact measurements are the most used due to their flexibility. In the pyrometer category they distinguish between cameras and photodiodes. Everton et al. makes a similar classification where they separate the pyrometer family into coaxial and non-coaxial methods and add visible cameras to monitor the recoating process [7]. On top of all these techniques, Wang et al. add acoustic emissions monitoring to the classification [8].

The monitoring system used for this study consists of multiple coaxial photodiodes that register the thermal emissions of the melted area (meltpool).

C Artificial intelligence applied to additive manufacturing

As in many other fields before additive manufacturing, artificial intelligence, and more specifically machine learning, has started to be combined with additive manufacturing to try to improve the processes and everything that surrounds them. In [8, 4, 9] the different application fields and approaches are reviewed in detail. For this study the main focus is what Wang et al. denotes as defect detection and quality prediction. In these kind of approaches, the aim is to create a model that is able to predict when some kind of defect has occurred or is occurring. The data used for these kinds of models is almost always obtained from in-situ monitoring systems which provide the model with the required insight into the printing process. Depending on the type

of monitoring systems used they can be classified as acoustic monitoring or optical monitoring. Even though optical monitoring is much more prevalent, acoustic monitoring approaches have great potential as they can access data that optical sensors can not.

C.1 Acoustic monitoring

Shevchik et al. in [10, 11] present a model capable of distinguishing three different fusion modes on an L-PBF printer. They printed one cube with three fusion modes at different layers and were able to create a deep neural network that predicts them based on small windows of the time series data coming from a fiber-bragg sensor. They preprocessed the data using a wavelet transform which converts the time series into an spectrogram (2D image) which is then classified by a convolutional neural network achieving between 83% and 89% accuracy.

Ye et al. in [12] present a similar study, but instead of printing cubes they use data from single tracks. They achieve a 95% accuracy using Fast Fourier transform combined with a deep belief network.

Lastly, Wasmer et al. present in [13] a new approach to classify the data collected by [11, 10]. They propose a reinforcement learning methodology which results in worse performance than the one presented by shevchik et al.

C.2 Optical monitoring

There are many studies that use optical monitoring for quality assurance or defect detection. Most of them make use of cameras in different configurations to extract information from the melting and the recoating process [14, 15, 16]. For details on works that use cameras to monitor the melting process Wang et al. make a great summary in its review [8].

Nevertheless, there is no abundance of studies that only use photodiodes to monitor the melting process. Okaro et al. [17] present a model able to classify into “OK” and “Not OK” traction probes using only photodiode data. They use singular value decomposition as a dimensionality reduction method and then a Gaussian mixture model for the classification. With these methods they achieve a 77% accuracy. Jayasighe et al. [18] present in their pre-print study a regression model able to predict part density from the photodiode data.

To the author’s knowledge [17, 18] are the only two published studies that make use of a similar monitoring system to the one that has been used for the present study.

D Classification of time series

Apart from the bibliography related to the application of machine learning to additive manufacturing, it is important to review the existing bibliography on time series classification, as this is the type of data and problem that has been tackled in this study.

Combining the classifications of this field given by [19, 20, 21], the field can be separated into classical machine learning and deep learning approaches.

The classical ML approaches can be split into feature-based and distance-based. The feature-based approaches make use of feature engineering or dimensionality reduction to transform the time series into tabular data that can be used by most classification algorithms. The distance-based methods define a way of measuring the distance among time series in order to use algorithms based on distances such as k-nearest neighbors.

The deep learning approaches can be split into generative and discriminative. The generative methods are very similar to the dimensionality reduction approaches in classical ML but the features are learned in a self-supervised manner. Discriminative models are split into end-to-end models, which will use the raw time series to make a prediction, and feature engineering combined with deep models for classification.

III. OBJECTIVES

The main objective for this study is the creation of a multiclass classification model able to distinguish between fusion modes produced on an L-PBF printer using only data from a coaxial photodiode monitoring system with an accuracy greater than 80%. This metric has been chosen based on the results found in literature, and it is considered that any less than that would not be enough to compete against other solutions.

On top of the creation of the model, a comparison of data processing pipelines is to be performed in order to find the best suited for this problem.

IV. METHODOLOGY AND EXPERIMENTS

One of the key parts of this work is the monitoring system. It consists of a total of 12 photodiodes divided into four groups of three, corresponding to each of the four lasers. Two of the photodiodes are

used to measure the optical emissions of the melt-pool while the others monitor the emissions of the laser. The first two measure from 700nm to 1050nm and from 1090nm to 1700nm respectively while the other measures only the light from the 1070nm laser.

The native sampling rate of the system is 100Khz but the data extracted from the manufacturer's software has a variable sample rate, as it organizes the data into the exposure points that the modulate laser produces. In every exposure point there is at least one native data point, and if there are more than one, the average is taken.

The data is extracted from the manufacturer's software as CSV files that contain a row for each exposure point and 7 columns. The CSVs are segmented by layer and laser. The last three columns correspond to the measurements of the three photodiodes of that laser, while the other 4 columns contextualize the data in time and space by indicating the starting time, duration, x and y position of each exposure point.

A Data acquisition

The creation of the dataset followed a similar approach to the one proposed in [11]. The idea is to use process parameters that deliberately generate the different types of fusion modes that will be used as labels. For this study 18 sets of parameters were selected that generate 5 different levels of fusion modes; nominal fusion, lack of fusion, keyhole, high lack of fusion and high keyhole. To achieve the different combinations, power, point distance and exposure time were varied to obtain the following volumetric energy densities:

- High keyhole: $165 \pm 10 J/mm^3$
- Keyhole: $110 \pm 10 J/mm^3$
- Nominal: $65 \pm 5 J/mm^3$
- Lack of fusion: $42 \pm 3 J/mm^3$
- High lack of fusion: $26 \pm 3 J/mm^3$

A total of 40 cubes were printed. Each of the five classes has eight cubes with at least two different sets of parameters. The cubes were separated into four printing jobs in order to optimize their separation on the build plate and the amount of printing jobs required. The distribution of the cubes on the build plate was specially design to minimize thermal interaction caused by cube proximity and physical interaction due to the recoater and gas flow systems.

B Preprocessing

After the data collection phase, three preprocessing steps were performed.

Firstly, the data was reorganized. Natively the

data is obtained in hundreds of CSV files separated by layer and laser used, but for this study the data had to be separated by the cube to which they belong. To accomplish this, a method to determine the cube to which each exposure points belongs, had to be developed. Different alternatives were studied but the following was selected due to computational expense and automation.

The first step of the process is to determine the location of each cube. For this, the data is transformed from time series format into a binary array using the x and y position of each exposure point. The binary array is then analyzed with a connected components algorithm that creates a mask of each cube. This mask is then used to filter the CSVs and separate the data from each cube.

The second preprocessing step consisted of eliminating the data corresponding to the contours of the cubes. Part contours are not printed using the same parameters as the infill, so its data must be removed in order to avoid introducing noise into the dataset. As in the first preprocessing step, the main problem to solve, was determining which data points corresponded to the contours. This problem was solved using some insight of the inner workings of the laser movement. While doing a hatch, the laser stays off in between exposure points the amount dictated by the jump time parameter, but when jumping in between hatches, the laser stays off for much longer, 100 times more at least. This enabled the separation of the data into hatches, which then enabled the classification of each hatch as infill or contour. This classification was done by comparing the length of the hatch with the distance between the first and last point of the hatch. For infill hatches, these two distances should be very similar, but for contour hatches, the length should be much bigger than the other distance.

Lastly the third preprocessing step was motivated by the discovery that the measurements from each laser did not line up as expected. When printing a part with the same parameters, it is expected that the measurements from each photodiode would be the same on average. This was found not to be the case. So, to mitigate this problem, a correction of the data was performed. The data used to calculate the corrections was extracted from an old print job that had a massive part in it. This job was selected as the corrections were expected to be more robust the bigger the part. Calculating the average response of each laser in each layer, the correction coefficients were calculated.

C Data transformation

As seen in the state of the art analysis of time series classification techniques, most methods require the time series to be transformed into a more manageable form, which can be used by classification algorithms. To this end, three steps of data transformation were performed. Firstly, to increase the amount of training examples in the dataset, each cube was segmented into smaller chunks. Secondly, in preparation for the application of dimensionality reduction and feature extraction techniques, the length of the time series was equalized. And lastly, a dimensionality reduction step was performed.

For each of the three steps described above, different alternatives were tested. For the segmentation, three scales were selected: cube scale, layer scale, and hatch scale. In the cube scale, no segmentation was performed. At the layer scale, each dataset sample corresponds to the data from one layer. Not all layers were considered, as some of the lower and upper layers of the cubes were considered to contain transitory states which might not correspond well with the label of the whole cube. Lastly, at the hatch scale, instead of making each hatch one data sample, it was considered better to use 100 long windows contained inside each hatch without overlap. In this way all hatches shorter than 100 points were discarded, and hatches longer than 100 points might enable the extraction of more than one window. This variation from the other two scales was considered as very short hatches might not be statistically relevant to assign them with the cube's label.

After the segmentation, the scales which did not have a uniform series length needed to be equalized. To this end, 4 methods were tested, uniform down sampling, truncation, 0 padding at the end and no length-equalization. The no equalization option was included because one of the feature extraction techniques selected does not require series to have the same length.

Lastly, for the key part of the transformation; the dimensionality reduction or feature extraction, five different techniques were tested. Three types of autoencoders, SVD dimensionality reduction and manual feature extraction. The SVD technique is the same used in [17] and was included as a way to compare the proposed methodology with others in the literature. For the manual feature extraction, 7 features were selected. These features had been proposed in [22] for time series anomaly detection. The autoencoders tested were: a vanilla neural network, a convolutional neural network, and a recurrent neural network with LSTMs. The vanilla NN

was selected as a benchmark while the other two were expected to perform better.

D Datasets comparison

After having created one dataset with each of the viable combinations from the previous section, the next step was evaluating their performance of the classification task at hand.

To evaluate the performance of the datasets, the idea was to train a classification model with each one, and use the classification performance of the model as a metric of the quality of the dataset. For it to be a fair comparison, the models had to be as similar as possible.

All the datasets from the same scale were used to train a classification neural network with the same architecture and initialization for all of them. Using the same architecture for different scales was considered to be inadequate as the dataset size changes drastically with the scale, which changes the optimal size of the network.

For the training, validation and testing data split strategy, it was considered that the best way would be to use the same testing data for all models and use 8-fold cross validation during training. As for the way to separate the data, it was decided that even at the scales where it was possible to mix cubes, whole cubes would be used to separate the test and validation data. This is considered to be very important as including data from the same cube in training and validation seems to give over optimistic metrics. The test set was selected at random but assuring that at least one cube of each class was present. And lastly, to evaluate the performance, the global F1 score was used.

Using this methodology, the following comparison were made:

- Best dataset at the cube scale: As described above, all the datasets from this scale were used to train a NN. The architecture of the network was the same for all datasets, and for its selection a hyper-parameter search was performed. The network selected was the one with the better mean F1 score across all datasets.
- Best dataset at the layer scale: All the datasets at the layer scale were compared following the same methodology as the cube scale comparison.
- Best dataset at the hatch scale. Same methodology as the two previous comparisons but for the hatch scale datasets.
- Best length-equalization method: Without training any more models, the data from the cube and layer scale was compared to see which

method was best. The hatch scale models weren't used as they already had the same length for all series.

- Best dimensionality reduction technique: Using the the data from the previous comparisons, the best dimensionality reduction technique was identified.
- Best classification algorithm at cube scale: The cube scale datasets are the smallest of all. This makes them the least suitable for a NN classifier. Taking this into account, random forest, support vector machine, and k-nearest neighbours were tested.
- Best scale: Once all the best dataset from each scale was selected, they were compared to find the best of the three.
- Random cross validation vs cube-based cross validation: As mentioned before, it is believed that mixing cubes in testing and validation might overestimate the generalization capabilities of the model. This hypothesis was tested training one model with each method.

E Final model

Once the best dataset was identified, it was used to create a fine-tuned neural network classifier. The results from this model were analyzed using its accuracy, F1 score, and confusion matrices. Finally the model's performance was compared with similar solutions found in the literature.

V. RESULTS AND DISCUSSION

A Cube printing

Out of the four planned printing jobs, two were completed successfully, while the other two failed on the first try but were successful on the second, after some modification. Four of the eight cubes of the keyhole class failed during printing due to an excess of surface roughness causing the two failed printing beds. In order to complete the print jobs, these 4 cubes had to be eliminated from the build job.

This failure, is considered to be due to the proximity of the cubes to the edge of the buildplate and proximity to other cubes and not due to faulty parameters. This theory is reinforced by the fact that the cubes corresponding to high keyhole were printed without any problems in a less populated build job.

B Cube characterization

Once the cubes were printed, they were characterized in order to validate the fusion mode that was going to be assign to them. The characterization consisted of a density measurement via the Arquimedes method and micrographs.

Both measurements validated most of the cubes. But, there were four cubes that had to be re-labeled into other classes as they produced a different level of porosity than expected. Two cubes from the key-hole class had to be changed to the nominal class as they had very low levels of porosity, which is what would be expected in the nominal fusion mode. Also, two cubes from the nominal class had to be changed into lack of fusion, as they exhibit higher levels of amorphous porosity than expected.

C Dimensionality reduction

All the data preprocessing steps and the two first steps from the data transformation pipeline were performed as planned, but in the dimensionality reduction step, some of the techniques did not work as expected.

More concretely, some of the combinations between scales and autoencoders failed due to memory issues. At the cube scale, none of the three autoencoder types worked. This probably was caused by the size of the time series. At this scale, the shortest time series, has 25 million time steps. This means that the smallest vanilla autoencoder with 5 dimensions in the latent space, would have over 250 million parameters. For this reason, the autoencoders at the cube scale were abandoned as a dimensionality reduction technique.

At the layer scale, only the recurrent and one convolutional autoencoders failed. In hindsight the RNN autoencoders could had been expected to fail because at this scale, the series have at least 60,000 time steps, which makes the network too deep to be trained.

Lastly, at the hatch scale, no method failed, thanks to the smaller size of the time series.

D Dataset comparison

After all the transformation steps, a total of 18 different datasets were created. As described in the methodology, all the datasets were used to train a neural network classifier. The results from all these tests are summarized in table 1.

From these results it can clearly be seen how with cross validation at all three scales the best combination is to perform no length-equalization combined with the hand picked features. At the cube

Scale	Length	Dim	Train	8-fold	Test
Cube	DS	SVD	0.864	0.567	0.600
Cube	TR	SVD	0.828	0.467	0.700
Cube	PD	SVD	0.921	0.581	0.500
Cube	NO	SC	0.992	0.656	0.900
Layer	DS	NN	0.977	0.745	0.999
Layer	DS	CNN	0.961	0.728	0.999
Layer	DS	SVD	0.986	0.804	0.990
Layer	TR	NN	0.978	0.796	0.999
Layer	TR	CNN	0.957	0.800	0.999
Layer	TR	SVD	0.988	0.806	0.994
Layer	PD	NN	0.999	0.598	0.999
Layer	PD	SVD	0.999	0.550	0.999
Layer	NO	SC	1.000	0.811	0.983
Hatch	NO	NN	0.980	0.821	0.943
Hatch	NO	CNN	0.978	0.821	0.938
Hatch	NO	RNN	0.979	0.823	0.935
Hatch	NO	SVD	0.982	0.824	0.950
Hatch	NO	SC	0.980	0.833	0.957

Table 1: F1 scores for the models trained with all the combinations of transformation steps. Length-equalization column: DS down sampling, TR truncate, PD zero padding, NO no length-equalization.

scale, the difference is very notable, but as the scale shrinks so does the difference.

Comparing the results from the length-equalization techniques, the results are less clear. At the cube scale, zero padding and down sampling seem to have the edge, while at the layer scale, truncating seems to be the best option. Given this results, no definitive conclusion can be extracted.

Looking at the results from the different dimensionality reduction methods, it is clear that the seven features proposed in [22] are the clear winner. Looking at the rest, it seems like SVD takes second place in most cases. The only combination where an autoencoder surpassed SVD was at the layer scale with zero padding. These results suggest that the autoencoders might not be the best techniques for this problem, and given that they are much more computationally expensive, the decision to use the hand picked features is clear.

Finally comparing the three different scales, it is clear that the hatch scale wins over the other two, as all of its models surpass the best models of each of the other two scales.

As mentioned in the methodology, a NN classifier might not be the best technique when dealing with small datasets like the ones at the cube level. To see if other methods would be better suited, some tests were carried out. Using the best dataset from the cube scale, a random forest model, a SVM model

and a KNN model were trained. Random forest achieved 0.840 F1 score in cross validation making it the best of the tested models. To check if the final model should be a random forest, the best dataset from the hatch scale was also tested with these techniques, but the results were much less impressive, 0.764 F1 score.

Lastly, the comparison between random cross validation and cube-based cross validation, was carried out with the best dataset from the hatch scale. The F1 score went from 0.833 with cube-based cross validation to 0.978 with random cross validation. This confirms the theory that the random cross validation over-estimates the predictive power of the model.

E Final model

Based on all the results from the last section, the final model was trained using the dataset created at the hatch scale with the hand picked features. To define the architecture of the model, a hyperparameter search was performed. The final model has three hidden layers with 1000, 500 and 100 neurons respectively. The output layer has 5 units with softmax activation while the rest of the layers use relu. In between the hidden layers, dropout layers with a probability of 50% were introduced to control the overfitting. The network was initialized using glorot, the optimizer used was Adam with a learning rate of 10^{-3} , batch size was set to 8192 and the network was trained for 40 epochs.

The test set used was the same as the one used for all the models in this study, and cross validation was changed from 8 folds to 26.

After training the model and reviewing its confusion matrices, a flaw in the dataset was identified. The keyhole class only had 2 cubes left, as 4 of them couldn't be printed and 2 more had to be changed to another class. This means there is one cube in the test set and one in the train-cross validation set. The only problem with that, is that in the cases that the only keyhole cube from the train set is put in the cross validation fold, the model is tested on a class that has not been trained for. This hurts the scores of the model. For this reason, the keyhole class was removed, and the model retrained.

The final model achieves 0.9723 F1 score in training, 0.8754 F1 score in cross validation and 0.9768 F1 score in testing. The confusion matrix is presented in table 2. The confusion matrix shows how the model tends to underestimate the energy density of the samples and how nearly all its misses go to neighbor classes.

	Prediction			
	HLF	LF	N	HKH
HLF	128039	3012	0	0
LF	18955	789520	132	0
N	0	36230	225854	6
HKH	0	229	491	1349973

Table 2: Confusion matrix of the final model over the test set after the removal of the keyhole class.

VI. CONCLUSIONS AND FUTURE WORK

This research aimed to create a model able to predict the quality of metal additive manufacturing parts. This objective has been fulfilled as the model created can distinguish between 4 fusion modes closely related to part quality with an accuracy in excess of 86.2%.

On top of that, the comparison between data transformation pipelines has found a methodology that outperforms the methodologies found in literature. Concretely, the hand picked feature extraction was better on all tests to the SVD dimensionality reduction proposed in [17, 18].

The model created achieves at least 86.2% accuracy in all classes which outperforms the results presented in [11, 10] where they solved the same problem but using different monitoring capabilities.

A Future work

- One of the main problems that arose during the development of this study was the lack of keyhole data. This should be solved in future revisions of the project in order to create a balanced dataset.
- Apart from obtaining more data for the keyhole class, it might be necessary to re-define the boundaries of other classes in order to make them more robust. A lot of confusion was found from the nominal class blending into the lack of fusion class.
- Although the results from the autoencoders were underwhelming, combining them with the classification network to create an end-to-end model, might yield better results.
- Lastly, the model has been tested in a development environment, but to really know its industrial potential, it should be tested in a production environment. Implementing and testing the model in such an environment seems to be the logical next step for the project.

References

- [1] Iso ASTM. Astm52900-15 standard terminology for additive manufacturing—general principles—terminology. *ASTM International, West Conshohocken, PA*, 3(4):5, 2015.
- [2] T. Wohlers, Wohlers Associates (Firm), R.I. Campbell, O. Diegel, R. Huff, and J. Kowen. *Wohlers Report 2020: 3D Printing and Additive Manufacturing State of the Industry*. Wohlers Associates, 2020.
- [3] William E. Frazier. Metal Additive Manufacturing: A Review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928, June 2014.
- [4] Lingbin Meng. Machine Learning in Additive Manufacturing: A Review. page 15, 2020.
- [5] Sanjay Kumar. *Additive Manufacturing Processes*. Springer International Publishing, Cham, 2020.
- [6] Gustavo Tapia and Alaa Elwany. A Review on Process Monitoring and Control in Metal-Based Additive Manufacturing. *Journal of Manufacturing Science and Engineering*, 136(060801), October 2014.
- [7] Sarah K. Everton, Matthias Hirsch, Petros Stravroulakis, Richard K. Leach, and Adam T. Clare. Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, 95:431–445, April 2016.
- [8] C Wang. Machine learning in additive manufacturing_ State-of-the-art and perspectives. *Additive Manufacturing*, page 20, 2020.
- [9] N S Johnson. Invited Review_ Machine Learning for Materials Developments in Metals Additive Manufacturing. page 44, 2020.
- [10] S. A. Shevchik, G. Masinelli, C. Kenel, C. Leinenbach, and K. Wasmer. Deep Learning for In Situ and Real-Time Quality Monitoring in Additive Manufacturing Using Acoustic Emission. *IEEE Transactions on Industrial Informatics*, 15(9):5194–5203, September 2019. Conference Name: IEEE Transactions on Industrial Informatics.
- [11] S. A. Shevchik, C. Kenel, C. Leinenbach, and K. Wasmer. Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks. *Additive Manufacturing*, 21:598–604, May 2018.
- [12] Dongsen Ye, Geok Soon Hong, Yingjie Zhang, Kunpeng Zhu, and Jerry Ying Hsi Fuh. Defect detection in selective laser melting technology by acoustic signals with deep belief networks. *The International Journal of Advanced Manufacturing Technology*, 96(5):2791–2801, May 2018.
- [13] K. Wasmer, T. Le-Quang, B. Meylan, and S. A. Shevchik. In Situ Quality Monitoring in AM Using Acoustic Emission: A Reinforcement Learning Approach. *Journal of Materials Engineering and Performance*, 28(2):666–672, February 2019.
- [14] Luke Scime and Jack Beuth. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing*, 19:114–126, January 2018.
- [15] Luke Scime and Jack Beuth. A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 24:273–286, December 2018.
- [16] Luke Scime, Derek Siddel, Seth Baird, and Vincent Paquit. Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation. *Additive Manufacturing*, 36:101453, December 2020.
- [17] Ikenna A Okaro. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing*, page 12, 2019.
- [18] Sarini Jayasinghe, Paolo Paoletti, Chris Sutcliffe, John Dardis, Nick Jones, and Peter Green. Automatic Quality Assessments of Laser Powder Bed Fusion Builds from Photodiode Sensor Measurements. April 2020. Publisher: Preprints.
- [19] Gian Antonio Susto, Angelo Cenedese, and Matteo Terzi. Time-series classification methods: Review and applications to power systems data. *Big data application in power systems*, pages 179–220, 2018.

- [20] Amaia Abanda, Usue Mori, and Jose A Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, 2019.
- [21] Hassan Ismail Fawaz. Deep learning for time series classification: a review. page 44, 2019.
- [22] Michael Jones, Daniel Nikovski, Makoto Imamura, and Takahisa Hirata. Anomaly detection in real-valued multidimensional time series. In *International Conference on Big-data/Socialcom/Cybersecurity. Stanford University, ASE. Citeseer*. Citeseer, 2014.