Key Engineering Materials
ISSN: 1662-9795, Vol. 926, pp 358-364
doi:10.4028/p-54j053
© 2022 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.

Submitted: 2021-11-30
Accepted: 2022-01-24
Online: 2022-07-22

Porosity Prediction in AM Using PBF-LB Employing Time-Series Classification

Vivek Mahato[1,2,3,4,a*], Dermot Brabazon[1,2,4,b], and Annalina Caputo[1,3,4,c]

[1] I-Form Advanced Manufacturing Research Centre, Dublin City University, Dublin, Ireland

[2] School of Mechanical & Manufacturing Engineering, Dublin City University, Dublin, Ireland

[3] School of Computing, Dublin City University, Dublin, Ireland

[4] Advanced Processing Technology Research Centre, Dublin City University, Dublin, Ireland

[a]vivek.mahato@dcu.ie , [b]dermot.brabazon@dcu.ie, [c]annalina.caputo@dcu.ie

Keywords: Additive Manufacturing, Defect Detection, Time-Series Classification, Machine Learning

Abstract. Additive Manufacturing (AM) using Powder-Bed Fusion Laser-Beam (PBF-LB) has great potential; however, it has challenges due to its sensitivity to the process parameters [1]. The availability of big data generated in AM facilitates the employment of Machine Learning (ML) tools to understand the process and have a predictive control over the production. An intelligent system like this can thus reduce material wastage and energy cost while increasing a plant's product quality and throughput. Time-series summary statistics (like mean and variance) can discard valuable discriminatory signatures embedded in raw sensor data. Therefore, special ML time-series classification (TSC) tools that can extract and utilise these signatures from the raw data are much more effective for a task like porosity prediction [1]. However, the data employed in [1] pertains to products with artificially designed pores or gaps. This study focuses on *naturally* occurring pores, rarer, and evaluates k-Nearest Neighbour (k-NN) with Dynamic Time Warping (DTW) over real-world manufacturing data to classify the porosity of individual raster scans. We believe that natural pores have more diverse signatures than artificial pores, as each pore varies in characteristics (like size and morphology).

Introduction

AM creates a product layer-by-layer using a 3-D Computer-Aided Design (CAD) file, unlike traditional subtractive processes, which creates the final product by removing excess material from a larger block. The onset of industrial revolution 4.0 drove massive research in AM. The gain in popularity of AM from academia and industry is due to its unique advantages over traditional manufacturing processes, like cheaper and accelerated prototyping, design flexibility, and environmental benefits [2]. Nevertheless, the current AM technology has its own challenges. For example, lack of reproducibility and variability in product's quality even in the same build remains a significant challenge in the widespread adoption of the technology [3].

The idea is to address the challenges in AM using ML techniques over its production data. Successful applications of ML is seen in the literature for predicting porosity with time-series pyrometer data as its input [1]. Similar to the study, we employ an AconityMINI 3D printer[5] (shown in Fig. 1). It is a Powder-Bed Fusion (PBF) system that employs the Selective Laser Melting (SLM) technique.

Powder-Bed Fusion Laser-Beam (PBF-LB) is an AM process that operates on the same fundamental principle of additively forming parts, layer upon layer, by depositing the powder and melting it. The process starts with designing the part and creating a 3-dimensional (3D) CAD file, then sliced into several 2-dimensional (2D) distinct layers. For each layer, first, the powder is fed onto the build plate, and then given the path present in the 2D file of that layer, a scanner scans the heat source (a laser beam in our case, therefore, PBF-LB) onto the powder bed. The laser melts the powder in its track, which later solidifies upon cooling. Each layer is then sequentially fused with the previous layer. The cycle of feeding the powder on the build plate and then melting and fusing it with the previous

[5] https://aconity3d.com/products/aconity-mini/

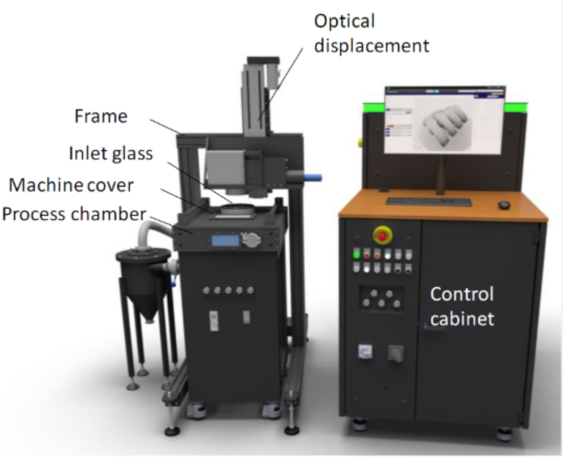


Fig. 1: Picture of the metal additive manufacturing AconityMini 3D printer [1].

layer continues until all the layers are built, therefore, creating the final product. The entire process happens in a chamber filled with an inert gas like argon. Fig. 2 illustrates the process of PBF-LB of AconityMINI.

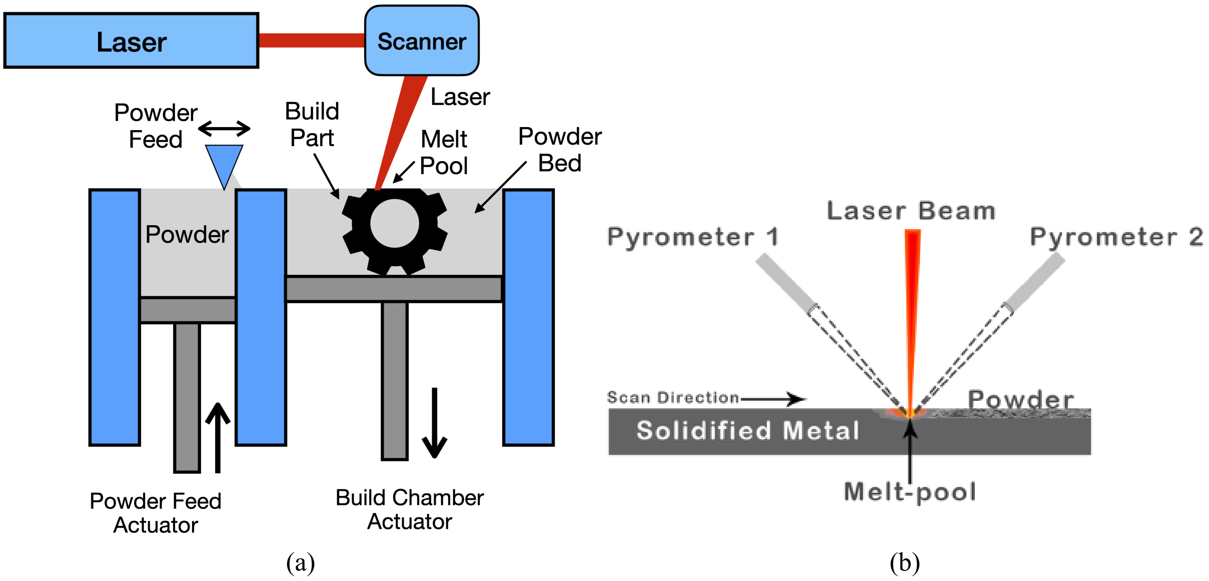


Fig. 2: (a) Block layout of AconityMini's PBF-LB manufacturing process, and (b) the laser scan melting the powder and leaving behind a solidified metal trace, and the two in-situ pyrometers capturing the melt-pool temperature [1].

The PBF-LB process reduces material wastage as the un-melted powder can be recycled for a later build. It also provides excellent active customisation, thus eliminating fixed designs. Furthermore, PBF-LB gives the flexibility of combining various materials like glass, metals, and alloys [2]. Nevertheless, PBF-LB has its disadvantages. It is susceptible to process parameters, and having a reliable set of build parameters is vital for better product quality [4]. In addition, the print process itself is relatively slow and uses high energy, and the built products need post-processing, which further adds to the time and cost [2].

On the positive side, AM produces vast amounts of data captured through in-situ sensors and quality measurement techniques. Having such an abundance of process data provides an excellent opportunity to employ ML to mine patterns and information. The discovered knowledge can then be utilised to address the challenges in the PBF-LB process and predict the product's physical properties [5, 6, 7, 8].

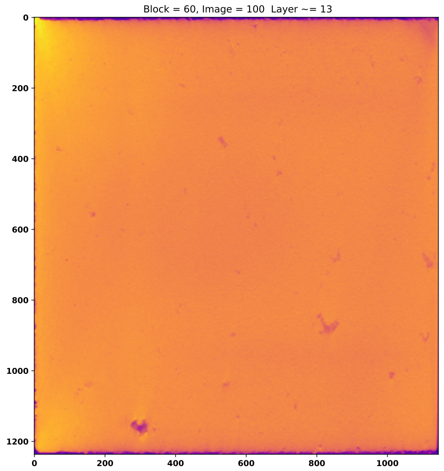


Fig. 3: A slice from the μ-CT scan representing a layer of NiTi block printed with 60 micron thickness.

Data

We built a batch of 10 mm $\times$ 10 mm $\times$ 10 mm Nickel-Titanium (NiTi) metal blocks using the AconityMINI system with 30, 60, and 90 microns layer thicknesses. These blocks were printed without any designed pores or cavities, and therefore any pores in the blocks were naturally generated from the process (see Fig. 3).

Two in-situ pyrometers measure the reflected light from the melt-pool area by KLEIBER Infrared GmbH[6], which detects the heat emission light in the range of 1500 to 1700 nm, representing the melt-pool temperature. The light is split into two paths through optical filters and transmitted via optical fibre cables to the pyrometers. Similar to the study by Mahato et al. [1], the scanner and the pyrometers are configured to cover x and y values (for each layer) in the range of 0 to 32,768 bit covering an area of 400$\times$400 mm, which results in a calibration value of 81.92 bit/mm. In addition, the frequency of the sensors is set to 100 kHz.

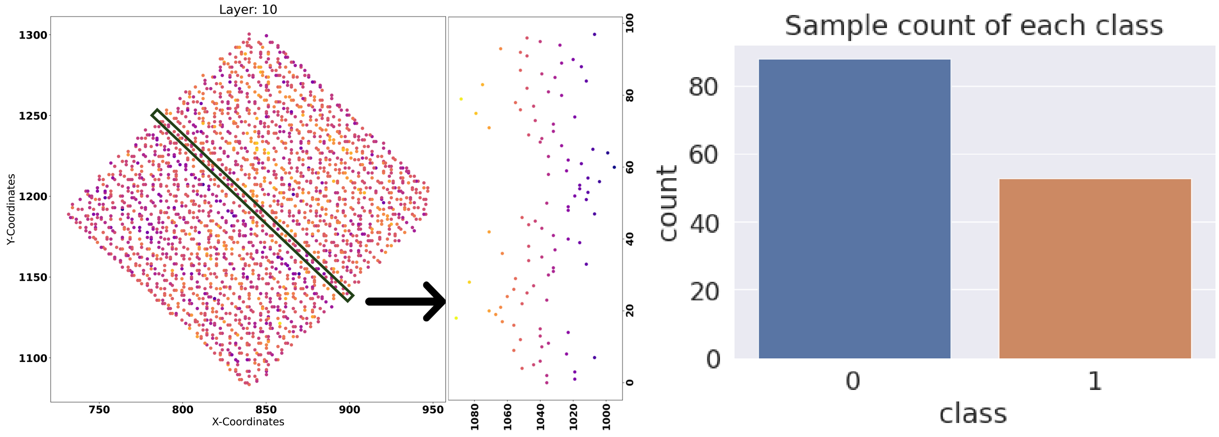
Each layer's pyrometer data is divided into individual raster scans (see Fig. 4a). We then truncate the data around a large visible pore to eliminate noise and magnify the patterns present in the data. The raster scans that pass through one of our located large pores are labelled as porous. Furthermore, the rest of the raster scans are carefully labelled as non-porous only if they do not pass through any visible pore. The class distribution of the resultant dataset is illustrated in Fig. 4b, where 0 and 1 correspond to non-porous and porous samples, respectively. The negative class (i.e. 0) contains 88 samples, and the positive class (i.e. 1) contains 53 samples, which results in a slight imbalance in the class distribution in the dataset.

k-Nearest Neighbour

The data from the sensor are inherently a time-series. Our idea is to employ the raw time-series data because summary time-series statistics (like mean and variation) can potentially lose informative discriminatory signatures embedded in them [9, 10]. Furthermore, tabular ML models like Decision Trees treat each vector in the time-series independently; hence, it is necessary to employ time-series ML tools that consider the sequential information of the series.

Mahato et al. [1] use k-NN with DTW as their TSC model for the task of pore vs non-pore classification task in their study. For consistency, we employ the same architecture in our study but for naturally occurring pores.

[6] https://www.kleiberinfrared.com/index.php/en.html



(a) (b)

Fig. 4: (a): (left) Pyrometer data from a layer of an AM Ni-Ti block with natural pores; a single raster scan is then highlighted. (b): Sample size of each class in the dataset, 0 is non-porous samples, and 1 is porous samples.

When it comes to classification or regression, Nearest Neighbours (NN) searching algorithms are found to be popular in ML literature with successful applications in various fields due to their simplicity and adaptability [11]. They are based on the assumption that similar samples are closer to each other than unlike samples in a d-dimensional space. NN algorithms render the versatility to formulate or leverage existing distance measures to compute the “nearness” of two samples. In addition, NN algorithms provide proof of their predictions by showcasing the located neighbours, making the algorithms interpretable [11].

Given a set of n points in a d-dimensional space, $X \subset E^d$, and given a query point $q \in E^d$, the idea is to extract a set K of k points in X that has minimum distance to q. First, the NNs are fetched by computing the similarity or distance between a query (a newly laid raster scan) and the set of training samples (previously laid and labelled raster scans) using the measure of choice (like DTW). Then the algorithm assigns the most frequent class in the query's neighbourhood (consisting of k-closest neighbour) to the query. Hence, distance or similarity measure choice is crucial as it directly corresponds to the model's performance.

Dynamic Time Warping. When computing distance between two time-series data, Euclidean distance might not be the best metric to employ. Firstly, Euclidean distance needs time-series data to be of equal sizes; consequently, it fails when the data is of unequal lengths. Secondly, Euclidean distance computes the distance between two time-series datum by linearly matching their data points (Fig. 5(a)). Therefore, even if two similar time-series data are only slightly displaced in the time-axis, the Euclidean distance between them is considerable (Fig. 5(b)). The aforementioned problem is illustrated in Fig. 5. DTW approaches this misalignment by giving us the flexibility of mapping the data points in a non-linear manner (Fig. 5(c)) [1].

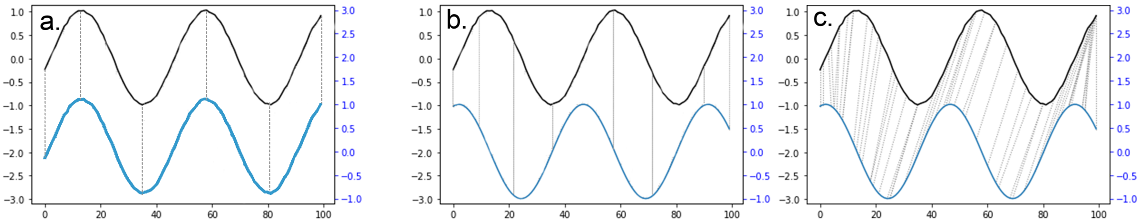


Fig. 5: (a) The two time-series are similar with no displacement on time-axis, therefore, Euclidean distance is small. (b) The two similar time-series are displaced in time, hence, Euclidean distance is large. (c) DTW non-linearly maps the two time-series in (b) [1].

Let q and s be two time-series with sizes u and v respectively, and define $\delta_{DTW}(i,j)$ as the DTW distance between $q(1:i)$ and $s(1:j)$, with the mapping path starting from $(1,1)$ to (i,j). Let the notation L^p represent the norm of order p, and initial condition be $\delta_{DTW}(1,1) = \delta_{L^p}(q_1, s_1)$. Then, recursively calculate $\delta_{DTW}(i,j)$ using Eq. 1; on completion, the minimum distance is at $\delta_{DTW}(u,v)$.

$$\delta_{DTW}(i,j) = \delta_{L^p}(q_i, s_j) + min \begin{cases} \delta_{DTW}(i-1,j) & \text{insertion} \\ \delta_{DTW}(i-1,j-1) & \text{match} \\ \delta_{DTW}(i,j-1) & \text{deletion} \end{cases} \tag{1}$$

In short, DTW constructs a cost matrix and then determines the shortest path through the grid. DTW algorithm is $O(t_1, t_2)$ as it requires searching through the matrix, making the complexity $O(t^2)$ in the time-series' average length. This causes the DTW algorithm to be computationally expensive. A global restraint like Sakoe-Chiba [12] is applied to prune unwanted deviation to obtain the optimal path while improving the time and memory complexity of the algorithm.

Evaluation

We split the dataset described in the "Data" section into train and test sets of sizes 105 and 36, respectively. For obtaining the best set of parameters of the k-NN-DTW model, we employ GridSearchCV from the Scikit-learn[7] library over a wide range of parameter settings. For the cross-validation mechanism, we employ RepeatedStratifiedKFold from Scikit-learn with the number of splits and repeats both set to 5 (therefore, a total of 25 iterations). The system is only exposed to the train data for hyper-parameter optimisation. Fig. 6 illustrates the model accuracies across the training instances (i.e. iterations). Here, we see that the best parameters (k-Neighbours and Sakoe-Chiba radius of 2 and 1, respectively) render a mean accuracy of the k-NN-DTW model of around *69%* on training data.

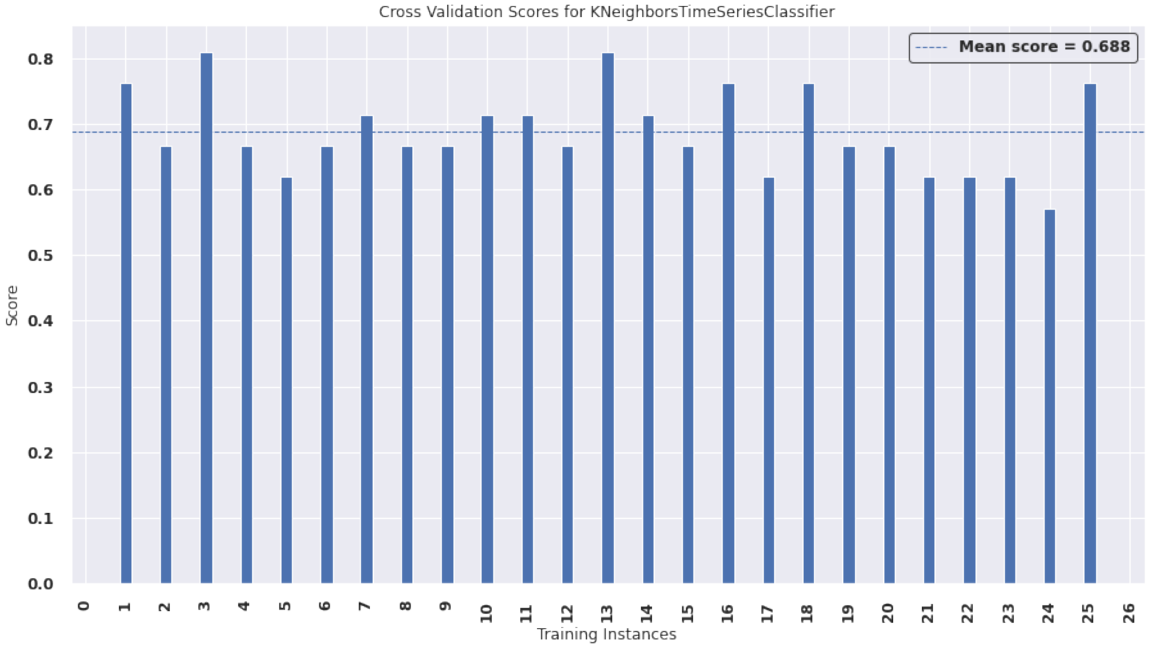


Fig. 6: Accuracy scores of k-NN-DTW model trained with best set of parameters across cross-validation folds over training data. Here, the mean cross-validated accuracy is around 69%.

Finally, we trained the model on training data supplied with the best parameters obtained in the previous step and evaluated it on the held-out test data. As a result, the model predicts the classes of

[7] https://scikit-learn.org/stable/

the held-out samples with an accuracy of *65%*. Although the model is reasonably good in classifying natural pores with an accuracy of *65%*, the resulting accuracy is relatively poor than the artificial pores model (*92%*) presented in the study [1]. The difference in performance is due to the classification task over naturally occurring pores being more challenging than the artificially designed pores.

Conclusion and Future Work

In this paper, we examine the performance of the k-NN-DTW model over AM dataset of metal blocks having natural pores. The following are our observations and our ideas for future work:

- In the study involving artificial cavities, k-NN-DTW achieves a high accuracy of *92%*, but when it comes to natural pores, the accuracy drops to *65%*.
- As stated previously, each natural pore has distinct characteristics, area, and morphology, unlike an artificial pore with fixed properties and well-defined shapes and forms. Therefore, if not accompanied by greater sample size for each category of the pore, such diversity has a detrimental effect on the model's performance for the prediction task.
- Most real-world data suffers from class imbalance, and so does AM datasets. Therefore, a more robust approach needs to be examined that address this issue. The idea is to either collect more data or use ML tools like SMOTE from the imbalanced-learn[8] library to upsample the minority class (i.e. porous) or use cost-sensitive learning, which gives higher weightage to the minority class.
- To properly evaluate the models over the classification task on the imbalanced AM dataset, a more robust metric like F1-score or ROC-AUC needs to be employed.

Acknowledgements

This publication has emanated from research supported by a research grant from Science Foundation Ireland (SFI) under grant number 16/RC/3872 and is co-funded under the European Regional Development Fund.

References

[1] V. Mahato, M. A. Obeidi, D. Brabazon, and P. Cunningham, "Detecting voids in 3d printing using melt pool time series data," *Journal of Intelligent Manufacturing*, 2020.

[2] M. R. Nichols, "The pros & cons of powder bed fusion," *Metal Powder Report*, vol. 76, no. 2, pp. 90–91, 2021.

[3] S. S. Razvi, S. C. Feng, A. Narayanan, Y.-T. T. Lee, and P. Witherell, "A review of machine learning applications in additive manufacturing," *Volume 1: 39th Computers and Information in Engineering Conference*, 2019.

[4] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, "Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives," *Engineering*, 2019.

[5] Z. Zhang and K. Friedrich, "Artificial neural networks applied to polymer composites: a review," *Composites Science and Technology*, vol. 63, no. 14, pp. 2029–2044, 2003.

[8] https://imbalanced-learn.org/stable/index.html

[6] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science," *APL Materials*, vol. 4, no. 5, p. 53208, 2016.

[7] Z. Li, Z. Zhang, J. Shi, and D. Wu, "Prediction of surface roughness in extrusion-based additive manufacturing with machine learning," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 488–495, 2019.

[8] G. Pilania, C. Wang, X. Jiang, S. Rajasekaran, and R. Ramprasad, "Accelerating materials property predictions using machine learning," *Scientific Reports*, vol. 3, no. 1, 2013.

[9] V. Mahato and P. Cunningham, "A case-study on the impact of dynamic time warping in time series regression," *3nd ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, vol. abs/2010.05270, 2018.

[10] V. Mahato, M. A. Obeidi, D. Brabazon, and P. Cunningham, "An evaluation of classification methods for 3d printing time-series data," *International Federation of Automatic Control, World Congress*, vol. abs/2010.00903, 2020.

[11] G. H. Chen and D. Shah, "Explaining the success of nearest neighbor methods in prediction," *Foundations and Trends® in Machine Learning*, vol. 10, no. 5-6, pp. 337–588, 2018.

[12] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, February 1978.