DEPOSITION QUALITY OPTIMIZATION OF ADDITIVE FRICTION STIR DEPOSITED ALUMINIUM ALLOY USING UNSUPERVISED MACHINE LEARNING

UDC:621.791.1:159.953
Original scientific paper https://doi.org/10.46793/adeletters.2024.3.2.2

Akshansh Mishra[1*]

[1]School of Industrial and Information Engineering, Politecnico Di Milano, Milan, Italy

Abstract:
Additive friction stir deposition (AFSD) is a promising solid-state additive manufacturing technology, but achieving continuous high deposition quality remains challenging due to complex process-structure connections. This study investigates unsupervised machine learning algorithms for mapping process parameters to deposition outcomes without requiring extensive labelled data. On experimental deposition data, algorithms including hierarchical clustering, k-means, spectral clustering, Gaussian mixtures, autoencoders, and self-organizing maps are used. The algorithms find intrinsic patterns and groupings in the multi-factor process data in an unbiased manner. With a silhouette score of 0.7618, k-means clustering performed the best, showing cohesive data clustering. Visualizations like dendrograms and trained maps shed light on the links between process parameters and deposition quality. The cluster analysis identifies process conditions that result in poor deposition quality. This highlights the ability of unsupervised approaches to capture deposition process patterns based solely on data with no prior system knowledge. This data-driven strategy has the potential to significantly improve AFSD process optimization and control, with implications for boosting industrial adoption. The unsupervised learning framework lays the groundwork for using process data to improve the quality and productivity of advanced manufacturing procedures.

ARTICLE HISTORY
Received: 28 March 2024
Revised: 6 June 2024
Accepted: 19 June 2024
Published: 30 June 2024

KEYWORDS
Additive Friction Stir Deposition, Unsupervised Machine Learning, Deposition Quality, Silhouette Score

1. INTRODUCTION

Artificial intelligence (AI) is a branch of computer science focused on developing systems that can mimic human intelligence and problem-solving abilities. They accomplish this by ingesting a plethora of data, processing it, and learning from the past in order to streamline and enhance in the future. A typical computer software would require human intervention to correct faults and improve operations. Between the invention of the term “artificial intelligence” and the 1980s, AI research experienced both significant growth and hardship [1-3]. The late 1950s and early 1960s were a moment of creation. From programming languages that are being used today to books and films that explore the concept of robotics, AI swiftly became a widespread subject.

Machine learning is an artificial intelligence (AI) subject focusing on systems that can learn from data to generate predictions or judgments without being explicitly programmed. The primary goal of AI is to create systems with human-level intelligence, and machine learning offers ways to this end. There are three types of machine learning algorithms: supervised learning, unsupervised learning, and reinforcement learning. Patterns are learned by supervised learning algorithms from labelled training data consisting of input-output pairs. Data with known class labels is used to train classification

*CONTACT: Akshansh Mishra, e-mail: akshansh.mishra@mail.polimi.it

methods such as logistic regression and decision trees. Regression algorithms are trained on input-output data samples to learn functional mapping. On the other hand, unsupervised learning discovers hidden patterns in unlabelled input data with no known output. Unsupervised tasks commonly used include clustering, dimensionality reduction, and density estimation. Through trial-and-error interactions with an environment, reinforcement learning educates AI agents to maximize rewards [4-9]. Deep learning combines neural networks and machine learning, allowing multi-layered models to extract high-level features from raw data. Combined, these machine learning techniques provide tools and procedures for developing AI systems capable of sophisticated functions such as speech and image recognition. Machine learning advancements will continue to be critical for accomplishing the overarching aims of artificial intelligence.

Additive manufacturing, often known as 3D printing, refers to procedures that layer-by-layer construct parts out of materials such as polymers, metals, or ceramics. It allows complicated shapes to be created without the limitations of standard subtractive approaches [10-12]. However, variances in final product qualities caused by factors like process parameters, scanning patterns, and material variations continue to be a significant concern. Machine learning approaches, which model the complicated interactions between these variables and part quality from data, provide intriguing answers. Chen et al. [13] studied the application of a Physics Informed Neural Network (PINN) to forecast the time-varying temperature distribution in part during laser powder bed fusion (L-PBF) manufacturing. Notably, the PINN used in the study allowed the model to be trained simply on randomly generated data. These training data are both affordable to gather and have the added benefit of improving the generalizability of the trained model. Jiang et al. [14] used an augmented machine learning technique to discover the conditions required to diminish the lack of fusion void formation in laser powder bed fusion by combining a mechanistic model with historical experimental data. Caltanissetta et al. [15] presented a technique for in-situ monitoring images captured layerwise in material extrusion AM using random forests and clustering. Du et al. [16] demonstrated that combining physics-informed machine learning, mechanistic modelling, and experimental data can reduce the occurrence of frequent faults in additive manufacturing.

This research demonstrates the first implementation of unsupervised machine learning algorithms in the additive friction stir deposition process for predicting the deposition quality. In section 2, the basic working mechanism of the implemented algorithms will be further discussed. This section emphasizes the importance of the research and hitherto research in the given filed.

2. ADDITIVE FRICTION STIR DEPOSITION PROCESS

Additive Friction Stir Deposition (AFSD) is a novel solid-state manufacturing technique that combines the concepts of friction stir welding and additive manufacturing. In AFSD, a rotating, non-consumable tool with a specially designed pin and shoulder moves across the surface of a substrate, generating frictional heat. This targeted heating weakens the material without melting, allowing for plastic deformation and flow. At the same time, feedstock material, usually metal wire or powder, is continually fed into the process zone. During the layer-by-layer deposition of AFSD deposits, the protrusions significantly agitate the plastically deformed material and produce heat. The protrusion's penetration depth into the previously formed layer varies with the thickness of the deposition layer. The interaction of the protrusions with the existing layer (or substrate in the case of the first layer) produces a different pattern of material deposition. To observe and compare the deposition patterns of these layers, slices of 1 mm, 2 mm, and 3 mm blocks were cut [17]. The cross-sections of the three blocks were then polished and etched for inspection. The results in Fig.1 show that the layers in the 1 mm block have a higher degree of intermixing than the other two. In contrast, different layers can be seen in the 3 mm block, where the layer thickness surpasses the height of the protrusions.

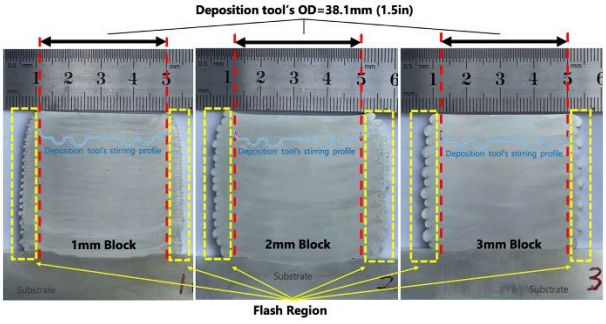

Fig. 1. Layer configurations and boundaries in polished and etched cross sections of 1 mm, 2 mm, and 3 mm blocks [17]

Aluminum alloy additive manufacturing (AM) has sparked tremendous attention due to its ability to produce complicated geometries and multilayered structures. However, the use of friction stir deposition (FSD) to create multilayered portions of AA7075-T6 aluminum alloy on an AA2024-T4 aluminum substrate is mostly unexplored. Elshaghoul et al. [18] addressed this gap by looking at the feasibility and optimization of such a method. Their research comprised a large number of tests altering feed rates (1-5 mm/min) and rotating speeds (200-1000 rpm) to discover the best deposition conditions. Key process variables like as axial deposition force and thermal cycle were painstakingly documented, and the heat input required for effective AFSD was determined.

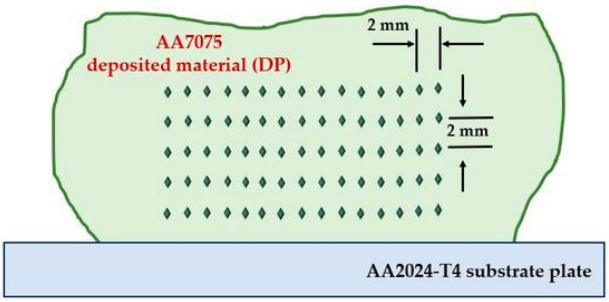

Fig. 2. Sketch of indenter positions for hardness measurements of the AA7075 DP [18]

3. OVERVIEW OF UNSUPERVISED MACHINE LEARNING ALGORITHMS USED IN PRESENT WORK

3.1 Hierarchical Clustering

Hierarchical clustering is an unusual approach in which we begin by allocating each object or example to its cluster, which is a simple answer. So, if we have *N* items, these are *N* clusters, each with only one item. The next stage is locating the two most similar clusters and combining them into one cluster. As a result, instead of *N* clusters, we now *have N-1* clusters. We repeat this method until all objects have been clustered into a single cluster of size *N*. What is remarkable about this algorithm is that it usually stops somewhere along the way, and we get something called a dendrogram, which shows us what we have merged so far at each stage, and we can have some stopping conditions. We call this agglomerative hierarchical clustering because we start with a collection of objects and agglomerate them. There is a catch: what do we mean by distance? Distance can be defined in a variety of ways. We obtain different answers depending on which linkage metrics we apply. The most common is the single linkage shown in Fig. 3, in which the distance between two clusters is equal to the shortest distance from any member of one cluster to any member of the other. We can also employ complete linkage shown in Fig. 4, meaning that the distance between any two clusters equals the highest distance between any two members.

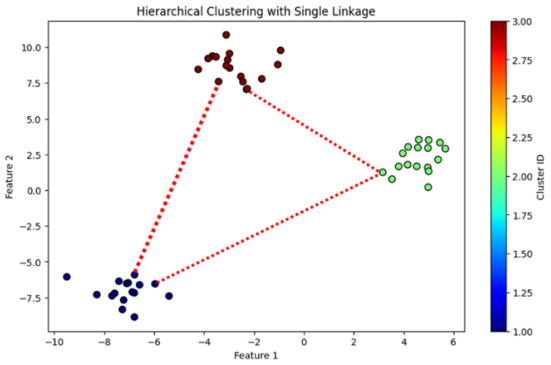

Fig. 3. Hierarchical Clustering with Single Linkage

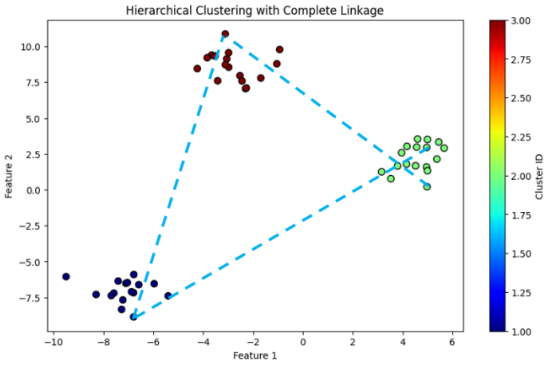

Fig. 4. Hierarchical clustering with complete linkage

When dealing with large datasets, the efficiency of k-means clustering shines. Unlike hierarchical clustering, which can be computationally expensive, k-means is more scalable and efficient. Its linear time complexity, denoted as *O(nkt)*, where *n* is the number of samples, *k* is the number of clusters, and *t* is the number of iterations, ensures a swift and effective clustering process. Hierarchical clustering, on the other hand, generates a rigid tree-based structure that restricts points from moving across groups. In contrast, k-means clustering allows points to be flexibly shuffled between clusters until convergence, enhancing its adaptability and efficiency.

3.2 k-Means Clustering

The k-means clustering algorithm is a kind of unsupervised machine learning method that is often used to arrange data points into groups based on their similarity. The central concept is to minimize total within-cluster variation, also known as inertia or the sum of squared distances between each data point and the center of its allocated cluster. k-means iteratively find the best grouping for the supplied number of clusters *k*. It works by recalculating cluster centers and reassigning data points to different clusters in order to reduce within-cluster variations between iterations until the assignments stabilize and converge, as shown in Fig. 5.

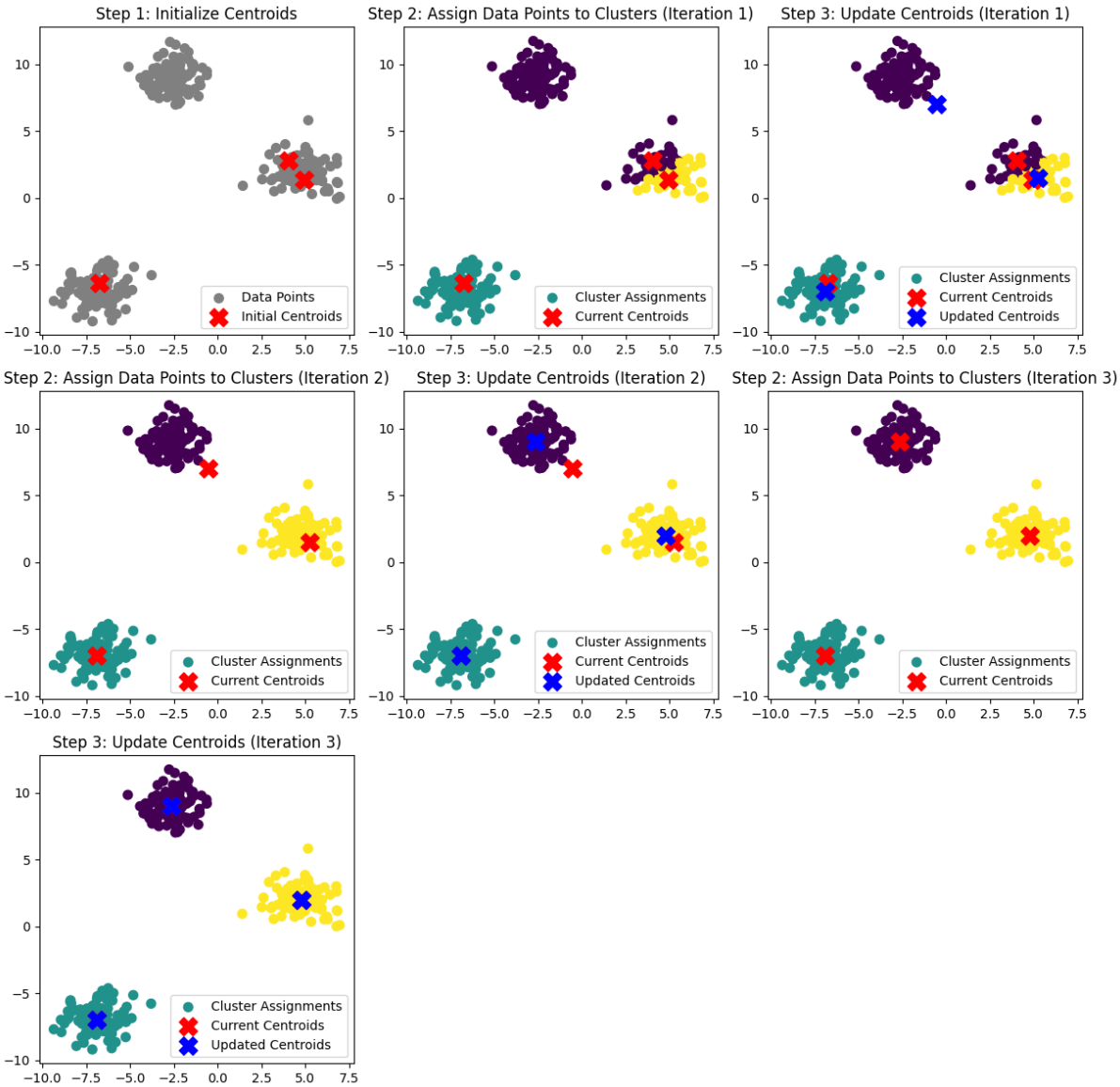

Fig. 5. Schematic showing the iterative process of k-means clustering. Points are assigned to the nearest cluster centroid, then centroids are updated based on cluster members

By selecting k data points from the dataset, the initial centroids for the clusters were determined. This selection procedure was done at random. Following that, the distance of each data point in the dataset was calculated for each of the *k* centroids. The Euclidean distance was commonly used as the distance measure for this purpose. After that, the data point was assigned to the cluster with the closest centroid. The mathematical equation shown in Equation 1 was used to create this assignment.

$$argmin_j \left\| X^{(i)} - \mu^{(j)} \right\|^2 \quad (1)$$

where $X^{(i)}$ is the *i*-th data point, $\mu^{(j)}$ is the centroid of the *j*-th cluster, and $\left\| X^{(i)} - \mu^{(j)} \right\|^2$ indicates the squared Euclidean distance.

Following the initial clustering of data points, the centroids of each cluster were updated. This update was accomplished by calculating the mean of all data points pertaining to each cluster. Equation 2 shows the mathematical representation for updating the centroid of cluster j.

$$\mu^{(j)} = \frac{1}{N^{(j)}} \sum_{i=1}^{N^{(j)}} X^{(i)} \quad (2)$$

where $N^{(j)}$ is the number of data points in cluster *j*.

The algorithm is iterated after centroid modifications. The assignment and updating procedures were repeated until one of the following conditions was met: 1) There were no longer any significant variations in the centroids, indicating convergence; 2) A preset stopping criterion, such as a maximum number of repetitions or a centroid movement threshold was met. When the algorithm finished, it produced the final clustering solution, which consisted of *k* unique clusters, each distinguished by its centroid. In addition, each data point was assigned to the cluster with the closest centroid, indicating the final cluster assignment.

3.3 Spectral Clustering

Spectral clustering, like k-means clustering, is an unsupervised machine-learning method for grouping data points based on similarities. The main distinction is that, whereas k-means clustering generates clusters by reducing distances between points in the original Euclidean space, spectral clustering first transforms the data into a new spectrum space. Creating a graph that connects similar data points and then utilizing the graph Laplacian matrix to map points into a lower-dimensional space is a common approach. Compared to k-means, spectral clustering can find more complicated, nonconvex cluster structures by operating on this altered data representation.

Fig. 6 shows the working mechanism of spectral clustering. The first step is to create a similarity matrix using pairwise Euclidean distances between data points. This matrix contains pairwise similarities and serves as the foundation for spectral analysis. An affinity matrix is built to transition from similarity to graph representation. The similarity matrix is subjected to an exponential decay algorithm to ensure data point connectivity. For consistency, diagonal elements are set to zero. Following that is the development of the Laplacian matrix, which is vital in spectral clustering. To obtain the Laplacian matrix, subtract the degree matrix from the affinity matrix, representing node degrees in the graph. This matrix represents the connections between data points. Basically, there are two standard methods to compute the Laplacian matrix for graphs: Unnormalized Laplacian, shown in Equation 3, and Normalized Laplacian, shown in Equation 4:

$$L = D - W \quad (3)$$

where are: *D* is the degree matrix (a diagonal matrix indicating the total of each node's edge weights) and *W* is the adjacency matrix (the similarity matrix);

$$L_{norm} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (4)$$

where are: I is the identity matrix and $D^{-\frac{1}{2}}$ is the inverse square root of the degree matrix.

The Laplacian matrix is decomposed into eigenvectors and eigenvalues via eigenvector decomposition. In this scenario, the first two eigenvectors are chosen, thus lowering the dimensionality of the data while keeping important information. The fourth graphic depicts the eigenvector decomposition, with blue and maroon dots representing the reduced dimensions. The final phase uses Spectral Clustering to find inherent clusters in the reduced data. The precomputed affinity matrix is used to perform spectral clustering with two clusters as the target.

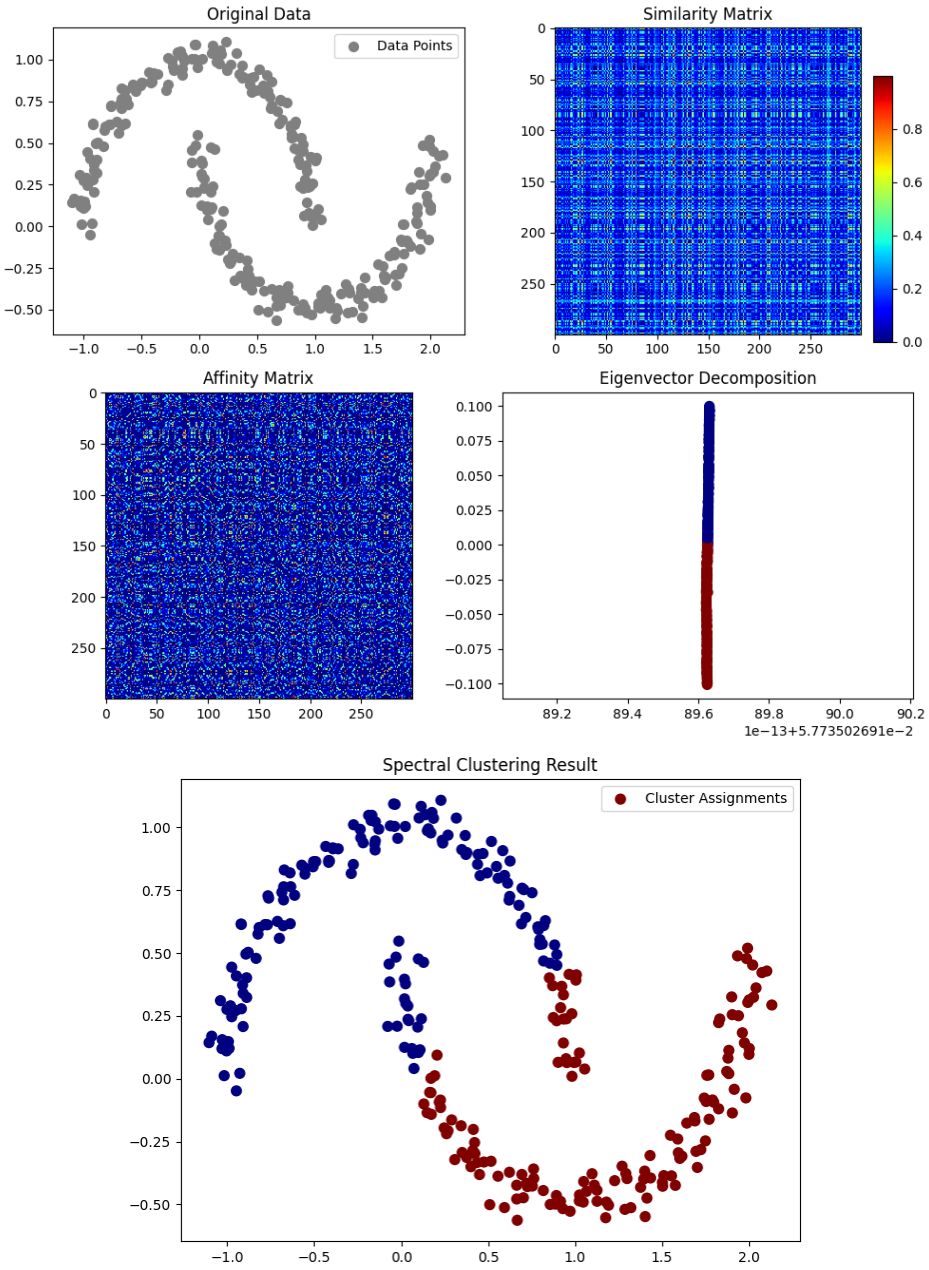

Fig. 6. Steps involved in spectral clustering - similarity matrix construction, graph representation, eigen decomposition and clustering

3.4 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The DBSCAN algorithm involves the setting of two critical parameters: ε and the minimal number of points (MinPts) within the ε-neighbourhood. MinPts sets the lowest data points required to form a dense region or cluster. In contrast, Epsilon specifies the maximum distance between data points to be considered neighbours. DBSCAN begins by randomly selecting a data point from the dataset shown in Fig. 7. It then examines this point's ε-neighbourhood to see if it contains at least MinPts data points (including the point itself). The chosen point is designated as a core point if this requirement is met. Clusters have core points that serve as beginning sites for cluster expansion. DBSCAN then searches for density-reachable points after detecting core points. A data point is density-reachable from another if a path links them (each within ε-distance of the next), starting from a core point. This path only goes through core points.

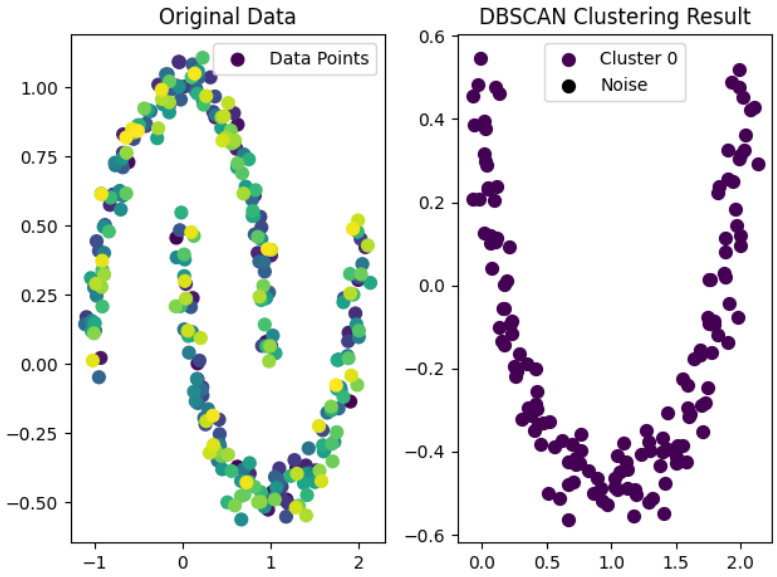

Fig. 7. DBSCAN identifies core, border, and noise points where a cluster forms from density-connected core points

3.5 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) shown in Fig.8 is a probabilistic model in which data is assumed to be generated by a finite number of Gaussian distributions with unknown parameters. The goal is to discover the Gaussian distribution parameters that best fit the data. Equation 5 describes the probability density function (PDF) of a univariate Gaussian distribution:

$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5)$$

where are: x is the random variable, μ is the mean of the distribution and σ is the standard deviation of the distribution. Instead of using a single Gaussian distribution, a GMM has a mixture of *K* Gaussian distributions. Equation 6 describes the probability density function of a GMM:

$$P(x) = \sum_{k=1}^{K} \pi_k . \mathcal{N}(x|\mu_k, \sigma_k) \quad (6)$$

where: $P(x)$ is the overall probability density function of the GMM, π_k is the weight or mixing coefficient of the *k*-th Gaussian component, and $\mathcal{N}(x|\mu_k, \sigma_k)$ is the PDF of the *k*-th Gaussian component with mean μ_k and standard deviation σ_k.

3.6 Autoencoders

Autoencoders are unsupervised artificial neural networks trained to reconstruct their own inputs. It is made up of two parts: an encoder and a decoder. The encoder reduces the size of the input data to a lower-dimensional representation known as the code. It has hidden layers that generate this code by performing non-linear modifications on the data. The decoder then attempts to reconstruct the original input from the code alone, as shown in Fig. 9. It also has hidden layers that return the code to its original input space. The encoder, represented as *f(x)*, transfers the input data *x* to a lower-dimensional representation *z*. This is commonly accomplished by employing a neural network with one or more hidden layers. Equation 7 denotes the encoder function:

$$z = f(x). \quad (7)$$

The decoder, represented by *g(z)*, takes the encoded representation z and attempts to recreate the original input *x'*. Like the encoder, the decoder can be built using a neural network. The decoder function can be written as Equation 8:

$$x' = g(z). \quad (8)$$

To train the autoencoder, a loss function is designed to assess the difference between the input x and the reconstructed output x'. Depending on the nature of the data, standard loss functions include Mean Squared Error (MSE) and Binary Cross-Entropy. The loss function is defined in Equation 9:

$$L(x, x') = MSE(x, x'). \quad (9)$$

where $MSE(x, x')$ is the Mean Squared Error between x and x'. The autoencoder seeks to minimize the loss function during training by modifying its parameters (weights and biases) using techniques such as gradient descent. The

optimization problem can be stated as Equation 10.

$$min_\theta \frac{1}{N}\sum_{i=1}^{N} L\left(x_i, g\left(f(x_i)\right)\right) \quad (10)$$

θ represents the parameters of both the encoder and decoder and *N* represents the number of training samples. The lower-dimensional representation *z* is also known as "latent space." It extracts key features and patterns from the input data. Autoencoders can develop helpful representations for many downstream tasks by lowering the dimensionality of the data.

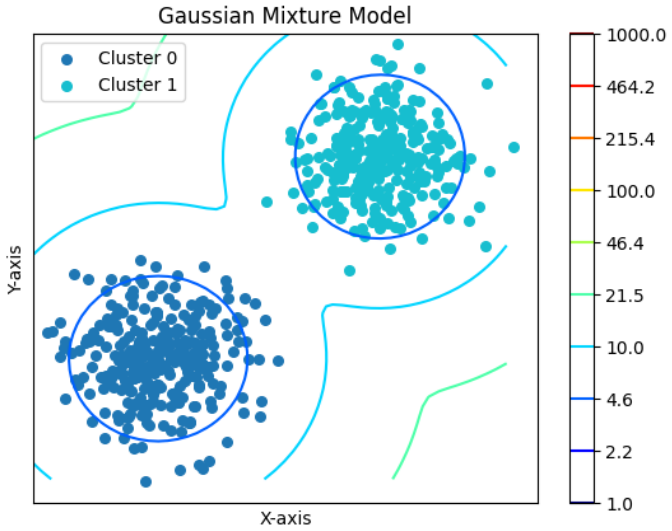

Fig. 8. Gaussian mixture model with mixture components

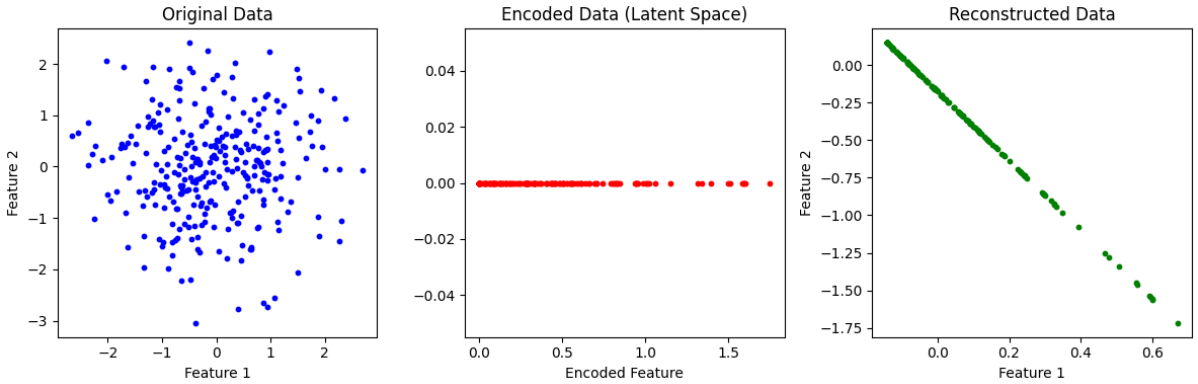

Fig. 9. Autoencoder architecture - encoder compresses input to lower-dimensional code, decoder reconstructs input

3.7 Self-Organized Maps (SOMs)

A Self-Organizing Map (SOM) is an unsupervised artificial neural network that organizes the mapping of high-dimensional input data onto a two-dimensional map. It is made up of an input layer that is completely linked to a two-dimensional grid of neurons known as the Kohonen layer. Each Kohonen neuron has a weight vector that corresponds to the dimensions of the input data. To begin the SOM training process, a grid of neurons in a two-dimensional or sometimes higher-dimensional lattice is generated. Each neuron has a weight vector that corresponds to the dimensionality of the input data. Initially, these weights are allocated random values. The SOM computes the similarity between each input data point *x* and the weight vectors of all neurons in the grid, often using the Euclidean distance metric. The Best Matching Unit (BMU), denoted as b as denoted in Equation 11, is the neuron whose weight vector is most similar to the features of the input data point.

$$b = \arg min_i \|x - w_i\| \quad (11)$$

The SOM then updates the weights of the BMU and its neighbours to improve their likeness to the input data point. The degree of adjustment is determined by a neighbourhood function, which is frequently Gaussian or rectangular and decreases with distance as shown in Equation 12:

$$h_{ib} = exp\left(-\frac{\|i-b\|^2}{2\sigma^2}\right) \quad (12)$$

where is: σ is the neighbourhood radius that decreases over time. The BMU and its neighbours' weights are modified to correspond more closely with the input data point. The following is the weight update rule shown in Equation 13:

$$w_i(t+1) = w_i(t) + \alpha(t).h_{ib}(t).(x - w_i(t)) \quad (13)$$

where are: $w_i(t)$ is the weight vector of neuron i at time t, $\alpha(t)$is the learning rate that decreases over time, $h_{ib}(t)$ is the neighbourhood function at time t and x is the input data point. The SOM repeats the phases of competition, cooperation, and adaptation for a predetermined number of iterations or until convergence is reached. Following training, the SOM returns a lower-dimensional representation of the input data, with related data points mapped to neighbouring neurons. SOM visualization frequently entails showing the weights of neurons in the same grid structure, using techniques such as heat maps shown in Fig. 10.

Fig. 10. SOM grid showing cluster assignments of input data after training convergence. Related inputs clustered together

4. MATERIALS AND METHODS

The Additive Friction Stir Deposition (AFSD) technique, which is based on friction stir welding principles, operates by layering material from a consumable rod or powder feedstock onto a substrate using a rotating non-consumable tool [19-21]. Friction heating and adiabatic shear heating from the rotating tool at the contact interfaces with the substrate and feedstock are the primary mechanisms underlying AFSD. This produces temperatures that are typically between 0.5 and 0.9 times the melting point of the material, enough to soften but not melt it. The tool's rotational motion causes a stirring action that plastically deforms the softened feedstock and intimately mixes it with the surface layers of the substrate. Because of the high temperatures and intense plastic deformation, metallic bonding between the deposited material and the substrate is possible via dynamic recrystallization and diffusion. As the tool moves across the substrate, new feedstock is constantly placed into its path and goes through the same sequence of heating, deformation, and bonding with the previous layer. The deposit is built up cyclically layer by layer to form a near-net shape additive part that is metallurgically linked to the substrate. The heat profile and material deformation encountered during AFSD are greatly influenced by process parameters such as tool rotation and traverse speed.

The data were collected from the research work of Chaudhary et al. [22]. The data were prepared in CSV format where Rotational Rate (RPM), Travel Speed (mm/min), Deposition Material Flow Rate (mm^3/min), and Tool Diameter (mm) were the input parameters and deposition quality is the output parameter. The data were further imported to

Google Colaboratory (Colab) which is a Jupyter notebook environment hosted in the cloud that has recently gained traction as a valuable tool for machine learning research and applied data science applications. Colab enables users to develop and run Python code entirely in the browser, without the need to install any software locally. It makes GPUs and TPUs available for free for rapid model training and inference. Colab notebooks can make use of pre-installed data science libraries such as NumPy, Pandas, Matplotlib, TensorFlow, PyTorch, and Keras.

5. RESULTS AND DISCUSSION

Fig. 11 shows the dendrogram obtained from the implementation of the Hierarchical clustering algorithm. Fig. 12 shows the SOM grid with the cluster assignments. Fig. 11 depicts the hierarchical clustering dendrogram, which depicts the hierarchy of cluster mergers at each level. The vertical line length reflects the distance between the merged clusters; shorter lines imply greater similarity. This representation reveals the underlying structure of the data. However, as mentioned in the paper, hierarchical clustering can be computationally expensive for big datasets. The trained self-organizing map in Fig. 12 projects the high-dimensional deposition data into a two-dimensional lattice. The map colors represent each data point's cluster assignment. Similar input data is mapped to neighbouring map locations in this visualization technique, displaying relationships in a low-dimensional representation. However, the map's fixed structure can limit adaptability.

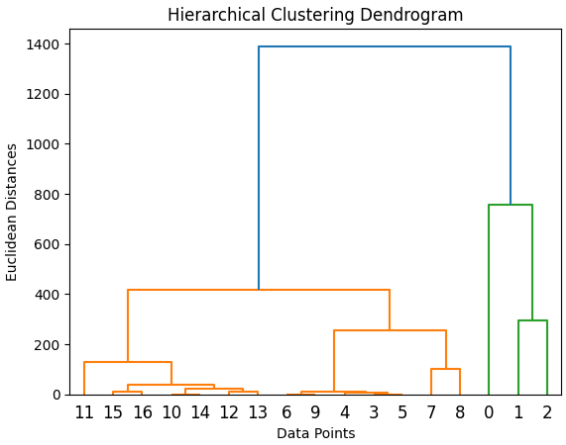

Fig. 11. Dendogram from hierarchical clustering of deposition data showing cluster merging hierarchy

The performance of the implemented algorithms was evaluated based on the Silhouette Score. It indicates how similar an object is to its own cluster in comparison to other clusters. The silhouette score runs from -1 to 1, with a higher value indicating that samples are more closely related to their own cluster and less closely related to nearby clusters. The silhouette score $s(i)$ for sample i can be calculated using Equation 14:

$$s(i) = \frac{(b(i)-a(i))}{\max(a(i),b(i))} \quad (14)$$

where are: $a(i)$ is the mean distance between sample I and all other points in its cluster and $b(i)$ is the mean distance between sample I and all points in the nearest neighboring cluster. A high silhouette score near 1 suggests good clustering, with samples close to their cluster and far from other clusters. Scores close to zero suggest overlapping clusters. Negative scores often imply that clusters have been assigned erroneously. Table 1 compares the silhouette scores of the various unsupervised learning methods investigated in this study. The silhouette score is a quantitative indicator that measures how well clustering methods split data into distinct categories. A higher score implies more coherent and well-separated groups. The table shows that k-means clustering achieved the most incredible silhouette score of 0.7618, beating the other procedures significantly. This indicates that k-means was the most effective at detecting inherent categories in the deposition data.

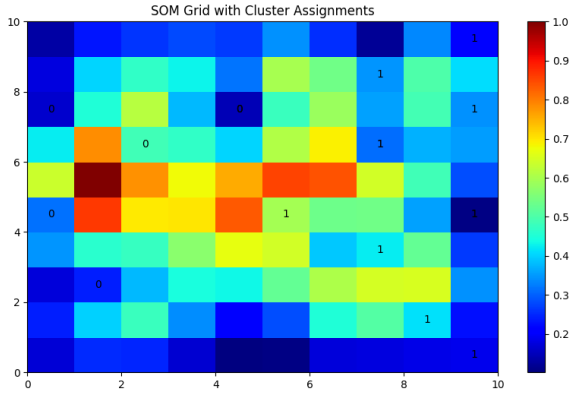

Fig. 12. Trained self-organizing map projecting deposition data into a 2D lattice, (map colors show cluster assignments)

Table 1. Comparison of the algorithms on the basis of Silhouette Score

Algorithms	Silhouette Score
Hierarchical Clustering	0.5986
k-Means Clustering	0.7618
Spectral Clustering	0.2756
DBSCAN	0.4320
Gaussian Mixture Model	0.2947
Autoencoders	0.6964
Self-Organized Maps	0.3088

The cluster coherence and separation achieved by k-means are most likely the result of two major reasons. First, k-means is optimized to decrease within-cluster variation, attracting points to the cluster centroid. Second, it reassigns points between clusters iteratively to improve coherence. Unlike hierarchical clustering, which has a fixed tree structure, k-means allows for the rescheduling of cluster assignments. These algorithmic improvements enable k-means to locate non-convex groupings more effectively. The poor performance of approaches such as spectral clustering and Gaussian mixture models demonstrates the challenge of capturing structure in complex deposition data. For this application, k-means clustering provides superior unsupervised modeling of the correlations between process parameters and deposition quality outcomes based on the silhouette evaluation.

6. CONCLUSION

This study has unveiled the significant role of unsupervised machine learning techniques in modelling the intricate interactions between process factors and deposition quality in additive friction stir deposition. The algorithms, including k-means clustering, hierarchical clustering, autoencoders, and self-organizing maps, have successfully revealed internal patterns and connections in the multivariate deposition data without the need for labelled training examples. Among these, k-means clustering has excelled in identifying coherent groupings of process conditions associated with lower deposition quality. The insights from the cluster analysis can be instrumental in process optimization to prevent poor quality outputs. Moreover, the unsupervised learning approach offers high-fidelity prediction of deposition results based solely on process variables, significantly reducing the need for costly trial-and-error research. While this study demonstrates the potential of unsupervised learning for deposition modelling, several areas require further exploration. More complex neural network topologies could potentially detect nonlinear correlations in data. Semi-supervised strategies that utilize sparsely labelled data could enhance model accuracy. The learning algorithms should also be tested on a wider range of materials and deposition processes. A crucial next step is to deploy the models for real-time process control. The data-driven technique could be integrated with physics-based process models for hybrid modelling.

Conflicts of Interest

The author declares no conflict of interest.

REFERENCES

[1] S. Gu, M. Choi, H. Park, S. Jeong, J. Doh, S.-i. Park, Application of artificial intelligence in additive manufacturing. *JMST Advances*, 5, 2023: 93-104. https://doi.org/10.1007/s42791-023-00057-7

[2] A. Baghbani, T. Choudhury, S. Costa, J. Reiner, Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. *Earth-Science Reviews*, 228, 2022: 103991. https://doi.org/10.1016/j.earscirev.2022.103991

[3] O. Mypati, A. Mukherjee, D. Mishra, S.K. Pal, P.P. Chakrabarti, A. Pal, A critical review on applications of artificial intelligence in manufacturing. *Artificial Intelligence Review*, 56, 2023: 661-768. https://doi.org/10.1007/s10462-023-10535-y

[4] C.J.C. Nocheseda, M.F.A. Santos, A.H. Espera, R.C. Advincula, 3D digital manufacturing technologies, materials, and artificial intelligence in art. *MRS Communications*, 13, 2023: 1102-1118. https://doi.org/10.1557/s43579-023-00489-1

[5] T. Jiang, J.L. Gradus, A.J. Rosellini, Supervised machine learning: a brief primer. *Behavior Therapy*, 51(5), 2020: 675-687. https://doi.org/10.1016/j.beth.2020.05.002

[6] S. Uddin, A. Khan, M.E. Hossain, M.A. Moni, Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19, 2019: 281. https://doi.org/10.1186/s12911-019-1004-8

[7] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, A.J. Aljaaf, A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and Unsupervised Learning for Data Science*, 2020: 3-21. https://doi.org/10.1007/978-3-030-22475-2_1

[8] W. Choi, R.C. Advincula, H.F. Wu, Y. Jiang, Artificial intelligence and machine learning in the design and additive manufacturing of responsive composites. *MRS Communications*, 13, 2023: 714-724. https://doi.org/10.1557/s43579-023-00473-9

[9] S.U. Saeed, Y. Fu, V. Stavrinides, Z.M. Baum, Q. Yang, M. Rusu, R.E. Fan, G.A. Sonn, J.A. Noble,

D.C. Barratt, Y. Hu, Image quality assessment for machine learning tasks using meta-reinforcement learning. *Medical Image Analysis*, 78, 2022: 102427. https://doi.org/10.1016/j.media.2022.102427

[10] E. Maleki, S. Bagherifard, A.H. Astaraee, S. Sgarbazzini, M. Bandini, M. Guagliano, Application of gradient severe shot peening as a novel mechanical surface treatment on fatigue behavior of additively manufactured AlSi10Mg. *Materials Science and Engineering: A*, 881, 2023: 145397. https://doi.org/10.1016/j.msea.2023.145397

[11] R. Ghanavati, E. Lannunziata, E. Norouzi, S. Bagherifard, L. Iuliano, A. Saboori, Design and development of SS316L-IN718 functionally graded materials via laser powder bed fusion. *Materials Letters*, 349, 2023: 134793. https://doi.org/10.1016/j.matlet.2023.134793

[12] L. Greco, F. Buccino, Z. Xu, L. Vergani, F. Berto, M. Guagliano, N. Razavi, S. Bagherifard, Design and analysis of energy absorbent bioinspired lattice structures. *Journal of Bionic Engineering*, 20, 2023: 1670-1686. https://doi.org/10.1007/s42235-023-00358-6

[13] J. Chen, J. Pierce G. Williams, T.W. Simpson, N. Meisel, S. Prabha Narra, C. McComb, Accelerating thermal simulations in additive manufacturing by training physics-informed neural networks with randomly-synthesized data. *Journal of Computing and Information Science in Engineering*, 24(1), 2024: 011004. https://doi.org/10.1115/1.4062852

[14] M. Jiang, T. Mukherjee, Y. Du, T. DebRoy, Superior printed parts using history and augmented machine learning. *npj computational materials*, 8, 12022: 184. https://doi.org/10.1038/s41524-022-00866-9

[15] F. Caltanissetta, L. Bertoli, B.M. Colosimo, In-situ Monitoring of Image Texturing via Random Forests and Clustering with applications to Additive Manufacturing. *IISE Transactions*, 2023: 1-15. https://doi.org/10.1080/24725854.2023.2257255

[16] Y. Du, T. Mukherjee, T. DebRoy, Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects. *Applied Materials Today,* 24, 2021: 101123. https://doi.org/10.1016/j.apmt.2021.101123

[17]H. Ghadimi, M. Talachian, H. Ding, S. Emanet, S. Guo, The Effects of Layer Thickness on the Mechanical Properties of Additive Friction Stir Deposition-Fabricated Aluminum Alloy 6061 Parts. *Metals*, 14(1), 2024: 101. https://doi.org/10.3390/met14010101

[18]Y.G.Y. Elshaghoul, M.M. El-Sayed Seleman, A. Bakkar, S.A. Elnekhaily, I. Albaijan, M.M.Z. Ahmed, A. Abdel-Samad, R. Reda, Additive Friction Stir Deposition of AA7075-T6 Alloy: Impact of Process Parameters on the Microstructures and Properties of the Continuously Deposited Multilayered Parts. *Applied Sciences*, 13(18), 2023: 10255. https://doi.org/10.3390/app131810255

[19] H.Z. Yu, R.S. Mishra, Additive friction stir deposition: a deformation processing route to metal additive manufacturing. *Materials Research Letters*, 9(2), 2021: 71-83. https://doi.org/10.1080/21663831.2020.1847211

[20] P. Agrawal, R.S. Haridas, P. Agrawal, R.S. Mishra, Deformation based additive manufacturing of a metastable high entropy alloy via Additive friction stir deposition. *Additive Manufacturing*, 60(A), 2022: 103282. https://doi.org/10.1016/j.addma.2022.103282

[21] V. Gopan, K.L.D. Wins, A. Surendran, Innovative potential of additive friction stir deposition among current laser based metal additive manufacturing processes: A review. *CIRP Journal of Manufacturing Science and Technology*, 32, 2021: 228-248. https://doi.org/10.1016/j.cirpj.2020.12.004

[22] B. Chaudhary, N.K. Jain, J. Murugesan, Development of friction stir powder deposition process for repairing of aerospace-grade aluminum alloys. *CIRP Journal of Manufacturing Science and Technology*, 38, 2022: 252-267. https://doi.org/10.1016/j.cirpj.2022.04.016