ACCSCIENCE PUBLISHING

International Journal of AI for Materials and Design

ORIGINAL RESEARCH ARTICLE

Prediction of the lack-of-fusion defect of laser powder bed fusion based on deep learning

Lidong Wang*

Institute for Systems Engineering Research, Mississippi State University, Mississippi, United States of America

(This article belongs to the *Special Issue: Applications of Deep Learning in Advanced Materials Processing*)

Abstract

Laser powder bed fusion (LPBF) is one of the additive manufacturing (AM) techniques and the most studied laser-based AM process for metals and alloys. The optimization of the laser process parameters of LPBF and the prediction of defects, for example, keyholes, cracks, and lack of fusion (LOF), are important for improving the quality of products made with LPBF. Deep learning (DL) is powerful in analyzing complex processes and predicting anomalies; however, much data is generally required for training a DL model. Experimental studies on AM (e.g., LPBF) habitually employ the design of experiments to decrease the number of experiments and save time and costs. Hence, the experimental data are not prepared for DL model creation in most situations. This paper studies the creation of a DL model on a small experimental dataset with unbalanced data and the prediction of the LOF defect of LPBF utilizing the created DL model. Data analytics is mainly conducted based on four DL methods, including Elman neural networks, Jordan neural networks, deep neural networks (DNN) with weights initialized by the deep belief network, and the regular DNN based on four algorithms: "rprop+", "rprop−", "sag," and "slr." It is shown that the regular DNN after the z-score standardization of the small dataset helps create a more accurate DL model and achieve better analytics and prediction results than the three other DL methods in this paper. The three other DL methods do not work well in the prediction of LOF based on the small dataset (with unbalanced data).

Keywords: Additive manufacturing; Laser powder bed fusion; Deep learning; Deep neural network; Defect prediction

***Corresponding author:**
Lidong Wang
(lidong@iser.msstate.edu)

Citation: Wang L. Prediction of the lack-of-fusion defect of laser powder bed fusion based on deep learning. *Int J AI Mater Design*. 2025;2(2):69-78.
doi: 10.36922/IJAMD025060005

Received: February 5, 2025

1st revised: March 27, 2025

2nd revised: May 15, 2025

3rd revised: May 29, 2025

Accepted: June 3, 2025

Published online: June 16, 2025

1. Introduction

Additive manufacturing (AM) is one of the key elements in Industry 4.0. The reliability and quality of metal AM parts or components are critical, especially for a laser powder bed fusion (LPBF) process. Melt pool defects of LPBF, for example, balling, keyholes, and lack of fusion (LOF) can compromise structural integrity.[1] LPBF has advantages, such as complex and precise parts, excellent material densification, *etc.* The laser scanning strategy is one of the significant factors affecting the quality of LPBF parts, where printing defects often occur at the path's endpoints and between the paths. The optimization of process parameters and the improvement of forming environment are

essential approaches to improving the comprehensive performance of LPBF parts.[2]

Defect modeling in an LPBF process is important. Modeling involves simulating and predicting the formation of defects and guiding the process of optimization. It helps achieve better control over LPBF parameters, including the laser power, the scanning speed, *etc.*, to reduce defects and improve the final part quality. Due to inadequate melting and bonding occurring between adjacent layers, LOF is a serious defect that compromises the overall strength and cohesion of the printed structure of LPBF.[3]

Sequentially learned random forest with enhanced sampling was presented for robust and efficient defect classification in an LPBF process.[1] A machine learning (ML) framework that combines a fuzzy logic scheme, a self-organizing map, and a tailored U-Net architecture was presented to improve the defect prediction capability of an LPBF process. The framework and methodology were employed to predict general defects, for example, keyholes and LOF, by analyzing *in situ* optical tomography data. Furthermore, a quality assurance professional was permitted to use the expert knowledge through customizable fuzzy rules.[4]

A deep learning (DL)-based approach to defect detection was proposed that uses various convolutional neural networks and transfer learning techniques to automatically segment and detect the melt pools of LPBF and porosity from microstructure images. The research demonstrated the ability to detect and segment melt pools and porosity accurately, even with a limited set of training data. It also paves the way for effective quality monitoring and quantitative evaluation of defects in LPBF.[5]

LPBF has been a widely utilized AM method for metals. *In situ,* sensing and monitoring have been an effective approach to detecting LPBF defects. A real-time defect detection system with feedback control was proposed based on the integration of 3D point cloud data processing with DL. This research demonstrated that there would be great potential for 3D point cloud-based DL in improving defect detection and quality control in LPBF.[6]

There are more hidden layers in a DL network than in a traditional artificial neural network (ANN). DL can automatically learn and discover relevant features from data or examples; therefore, it can detect patterns and trends and make predictions. Generally, much data is needed for training a DL model. A DL model can improve its performance if more data is used for model training. Overfitting and data availability are two major challenges in DL.[7] Generally, it is necessary to collect a lot of quality data (such as balanced data, not biased data, *etc.*) for DL model training; however, large amounts of data in engineering are often not available. For example, experimental studies on LPBF often use the design of experiments to reduce the number of experiments to save costs and time. Therefore, much of the experimental data of LPBF is for specific tasks; it is not prepared for ML/DL and the ML/DL model creation. The work in this paper is part of the author's effort to explore a DL model on a small experimental dataset with unbalanced data.

The main objective of this research paper is to establish DL models on a small dataset (with unbalanced data) and predict the LOF defect of LPBF based on the established DL models. The remainder of this paper is organized as follows: the second section introduces LPBF; the third section introduces data (the dataset used in this paper), data pre-processing techniques (min-max normalization and z-score standardization), and evaluation metrics for DL; the fourth section presents four DL methods utilized in this paper, including the Elman neural network, the Jordan neural network, the DNN with weights initialized by the deep belief network (DBN), and the regular DNN based on the four algorithms (rprop+, rprop−, smallest absolute gradient [sag], and smallest learning rate [slr]); the fifth section gives results and discussion; and the sixth section presents the conclusion and future research.

2. LPBF

Among the most studied laser-based AM process for metals and alloys is LPBF or selective laser melting. It is suggested that the term "LPBF" should be used according to ASTM standards. LPBF utilizes a high-power laser beam to melt the pre-defined contours selectively in subsequent layers of powder. The molten metal pool solidifies rapidly by cooling. The underlying build platform is lowered, followed by another layer of powder deposition. This cycle is repeated successively till a 3D solid object is constructed. The process is finished inside a chamber full of atmospheric gas (argon, nitrogen) to avoid oxidation.[8] Figure 1 illustrates the LPBF process. A nickel-based alloy is used in this research.

Although there is great potential for LPBF, surface and subsurface defects such as LOF and internal porosities affect the application of internal LPBF, especially where fatigue life is a major concern. The optimization of the fatigue performance of LPBF parts is substantially dependent on the manufacturing process parameters and post-processing of LPBF. It is significant to eradicate the formation of surface and subsurface critical defects (e.g., LOF) by optimizing the melting parameters to obtain materials with good fatigue performance.[9] Figure 2 shows

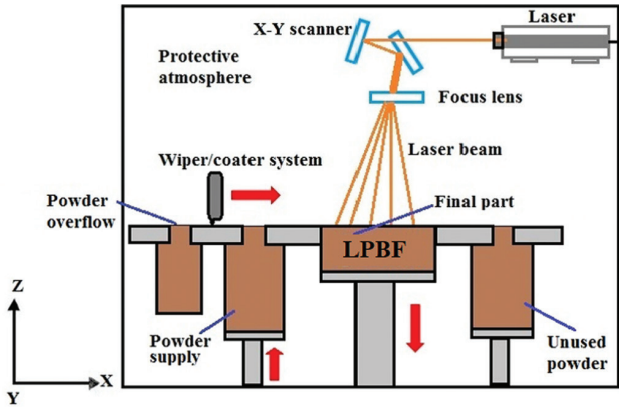

Figure 1. The laser powder bed fusion process[8]

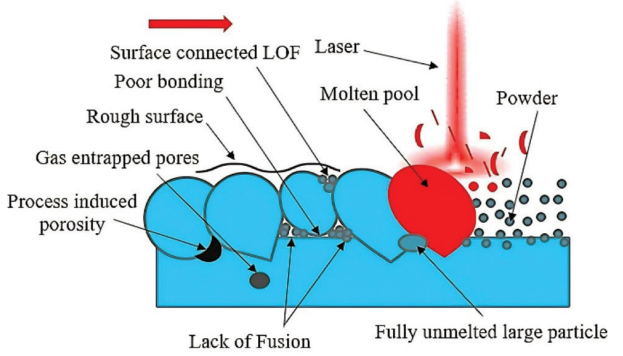

Figure 2. Possible defects and surface imperfections in the laser powder bed fusion (LPBF) process[9]
Abbreviation: LOF: Lack of fusion.

possible defects and surface imperfections in the LPBF process.

3. Data, data pre-processing techniques, and evaluation metrics for DL

The Nickel-based powder superalloy Ni-13Cr-4Al-5Ti has exceptional performance at high temperatures. The LPBF of the superalloy and associated defects were studied, and important results were obtained. The main defects include keyholes, cracks, and LOF. The process parameters include the laser power (W), the scanning speed (mm/s), the hatch space (mm), and the scanning rotation (°).[10] The dataset that was used for DL in this research was part of the experimental data regarding the main defects.[10] This paper focuses on the LOF defect. The experimental data of the LPBF of the superalloy (Ni-13Cr-4Al-5Ti) comprises 52 rows and 5 columns. LOF was utilized for DL in this research. The original experimental data were unbalanced. For example, in the column of the scanning rotation, there were five "0" values, five "45" values, thirty-two "67" values, five "90" values, and five "180" values. In the column of the scanning speed, there were 14 "1800" values; nine "1700" values and "1900" values, respectively; and five "600" values, "1000" values, "1400" values, and "2200" values, respectively. From an AM perspective, the scanning rotation degree has little effect on the LOF defect formation, as it does not significantly influence energy input. Therefore, the parameter of scanning rotation was not considered in the DL analysis in this research. Because there was unbalanced data in columns (with only 52 rows), the dataset used in this paper was small and unbalanced. In the column of LOF, "yes" was set to 1, and "no" was set to 0 for DL. There were thirty "1" values and twenty-two "0" values. The experimental data were not originally prepared for DL, though it was not necessary for DL training and testing. Table 1 shows partial experimental data of the LPBF of the superalloy.

Two common techniques for normalizing (or scaling) variables are:

- Min-max normalization: $(X - \min(X))/(\max(X) - \min(X))$
- Z-score standardization: $(X - \mu)/\sigma$

where X is the data value, μ is the mean, and σ is the standard deviation.

A random sampling with an 80–20 split or a 70–30 split on a big and quality dataset is frequently employed for DL model training and testing. A random sampling with a 60–40 split on the small dataset was conducted in this research because there were only twenty-two "0" values in total in the dataset. This means that the data for training was chosen through a random sampling of 60% of cases or examples in the dataset, and the remaining cases or examples (40%) after the sampling were used for the test. Choosing an 80–20 split or a 70–30 split on the small dataset will lead to a small number of test data and a poor performance evaluation (e.g., an unideal accuracy (*ACC*) value).

There were only two classes ("Yes" and "No," or "1" and "0") in Table 1. One class can be treated as "positive" (its value = 1) while the other can be treated as 'negative' (its value = 0). True positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*) can be expressed as follows:[11]

TP: The number of positive instances that are correctly classified as positive.

FP: The number of negative instances that are incorrectly classified as positive.

TN: The number of negative instances that are correctly classified as negative.

FN: The number of positive instances that are incorrectly classified as negative.

Table 1. Partial experimental data of laser powder bed fusion (selective laser melting)[10]

No.	Power (W)	Speed (mm/s)	Hatch space (mm)	Lack of fusion
1	200	1000	0.06	Yes (1)
2	200	1800	0.12	Yes (1)
3	240	2200	0.03	Yes (1)
4	240	1800	0.15	Yes (1)
5	270	1700	0.05	Yes (1)
6	270	1800	0.07	Yes (1)
7	280	2200	0.06	Yes (1)
8	280	600	0.09	Yes (1)
9	280	1000	0.12	Yes (1)
10	290	1800	0.05	Yes (1)
11	290	1900	0.06	Yes (1)
12	320	1400	0.03	Yes (1)
13	320	1800	0.06	Yes (1)
14	360	1000	0.03	Yes (1)
15	360	2200	0.12	Yes (1)
16	200	600	0.03	No (0)
17	240	1000	0.09	No (0)
18	270	1900	0.06	No (0)
19	270	1700	0.07	No (0)
20	280	1900	0.05	No (0)
21	280	1800	0.07	No (0)
22	290	1900	0.05	No (0)
23	290	1700	0.06	No (0)
24	290	1700	0.07	No (0)
25	320	600	0.12	No (0)
26	320	1000	0.15	No (0)

The *ACC*, false positive rate (*FPR*), and false negative rate (*FNR*) are utilized as measures for the classification and the performance of DL models in this paper. They can be calculated as follows.[12-14]

$$ACC = (TP + TN)/(TP + FP + TN + FN) \quad \text{(I)}$$

$$FPR = (FP)/(FP + TN) \quad \text{(II)}$$

$$FNR = (FN)/(FN + TP) \quad \text{(III)}$$

The value of $(TP + FP + TN + FN)$ is equal to the total number of instances in the testing data of the dataset.

Traditional ML methods (e.g., traditional ANN) have been used to predict the LOF defect. The results were not satisfying, which is the expected situation due to the dataset characteristics (small and unbalanced, see Table 1). Generally, this kind of data is also inappropriate for DL and the DL model creation. However, DL is generally more powerful (e.g., in handling complex data) than traditional ML methods. The objective of this research is to explore DL models on a small experimental dataset with unbalanced data, predict the LOF defect using the created DL models, and improve the modeling and prediction performance (according to the *ACC*, *FPR*, and *FNR*).

4. DL methods

4.1. The Elman neural network and the Jordan neural network

Both the Elman neural network and the Jordan neural network are recurrent neural networks (RNNs). There are one or more context layers in the Elman neural network, and the number of neurons in the context layer is the same as the number of neurons in the hidden layer. In addition, the context layer neurons are completely connected to all the neurons in the hidden layer. The Jordan neural network is similar to the Elman neural network. The only difference is that the context neurons in the Jordan neural network are fed from the output layer instead of the hidden layer.[15] The Elman neural network and the Jordan neural network are expressed as follows:[16,17]

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad \text{for the Elman neural network} \quad \text{(IV)}$$

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h) \quad \text{for the Jordan neural network} \quad \text{(V)}$$

$$y_t = \sigma_y(W_y h_t + b_y) \quad \text{(VI)}$$

where x_t is the input vector, and the input vector $V = (V_1, V_2, \ldots, V_p)$ in this paper; h_t is the hidden layer vector; and y_t is the output vector. W, *U,* and *b* are the parameter matrices and vectors. σh and σy are the activation functions.

4.2. The deep neural network (DNN) with weights initialized by the DBN

The DNN with weights initialized by the DBN means the DNN with the initial values of its weights that are set employing learned features from a pre-trained DBN. This technique is called DNN-DBN in this paper. DBN is a composition of restricted Boltzmann machines (RBMs). The procedures of the DNN with weights initialized by the DBN are: (1) training a DBN, (2) extracting the learned weights after the training of the DBN is completed, (3) initializing the DNN, and (4) performing "fine-tune" with supervised learning.[18]

DBN is employed to determine the weights, biases, and other parameters of the initial DNN. This technique does better than the only DNN-used technique in most situations.[19] The training technique for the RBMs is named

contrastive divergence.[20] Pre-training and fine-tuning are implemented while training a DBN. The following is specific information regarding the methodology and related algorithms.[21,22] The network energy is expressed as $E(v,h)$.

Let $p(v)$ be the probability of a visible vector, and it is described as follows:

$$p(v) = \frac{1}{Z}\sum_h e^{-E(v,h)} \quad \text{(VII)}$$

where $Z = \sum \exp(-E(v,h))$ is the partition function.

To train an RBM, weights are updated as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot \frac{\partial log(p(v))}{\partial w_{ij}} \quad \text{(VIII)}$$

4.3. Regular DNNs

In the DNN learning,[23,24] each training tuple can be handled in two steps:[23] Propagating inputs forward and backpropagating the error. For an input vector $V = (V1, V2,\ldots, V_p)$, each hidden layer transforms its inputs from the layer to the next layer by applying an affine transform and a nonlinear mapping as follows:

$$z^{(1)} = V \quad \text{(IX)}$$

$$y_j^{(l)} = \sum_{j=1}^{N^{(l)}} w_{ij}^{(l)} z_i^{(l-1)} + \theta_j^{(l)} \left(l = 2, 3, \ldots, L\right) \quad \text{(X)}$$

$$z_j^{(l)} = f(y_j^{(l)}) \quad \text{(XI)}$$

where L is the number of layers; $N^{(l)}$, $\theta_j^{(l)}$, and $w_{ij}^{(l)}$ are the number of nodes in the l^{th} layer, the bias of the node j in the l^{th} layer, and the weight of the connection from node i in the previous layer to node j of the l^{th} layer, respectively; and f is the activation function (nonlinear).

In this paper, the Elman neural network, the Jordan neural networks, the DNN with weights initialized by the DBN, and regular DNN based on various algorithms were employed to establish DL models and implement the prediction of the LOF of LPBF because all of the DL methods achieved good *ACC*, *FNR*, and *FPR* when large and quality databases such as "spambase" (https://archive.ics.uci.edu/ml/datasets/Spambase) were employed in the author's past research work.

Four algorithms, "rprop+", "rprop−", "sag," and "slr," were used in this paper. "rprop+" and "rprop−" refer to the resilient back-propagation with and without weight backtracking, respectively. "sag" and "slr" induce the usage of the modified globally convergent algorithm globally resilient backpropagation (grprop), adjusting the learning rate associated with the sag or the slr.

The calculation of *ACC*, *FPR*, and *FNR* (Equations I, II, and III) in this paper is based on the confusion matrix and the components in the matrix. The confusion matrix (*CM*) is given as follows:

$$CM = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix} \quad \text{(XII)}$$

5. Results and discussion

5.1. Results of the Elman neural network and the Jordan neural network

Both the Elman neural network and the Jordan neural network are RNNs. An Elman neural network can be thought of as a model that is constantly unfolding as a sequence of predictors. A Jordan neural network uses a context layer to process sequential data.[15] The dataset (Table 1) used in this paper is experimental data that can be regarded as a sequential dataset over time.

Tables 2 and 3 show part of the results of the Elman neural network and the Jordan neural network after min-max normalization and z-score standardization on the dataset, respectively. It was shown that both the Elman and Jordan neural networks did not work well in establishing DL models on a small dataset (with unbalanced data) and predicting the LOF of the LPBF. Most of the *ACC* values were low, and most of the *FPR* values, as well as most of the *FNR* values, were high or somewhat high. The reason is the small dataset and the unbalanced data in the dataset. The structures of context layers show the number of nodes in the context layers. For instance, c(8) indicates that eight nodes are in the context layer.

5.2. Results of the DNN with weights initialized by the DBN

Tables 4 and 5 list part of the results of the DNN with weights initialized by the DBN after min-max normalization and z-score standardization are employed, respectively. It was shown that DNN-DBN did not work well in establishing DL models on a small dataset (with unbalanced data) and predicting the LOF of LPBF. There are the input layer, the output layer, and two or three hidden layers in this technique. The performance (according to the *ACC*, *FPR*, and *FNR*) of the established DL models was not good due to the small dataset and the unbalanced data in the dataset. The structures of hidden layers indicate the number of nodes in the hidden layers. For instance, c(8, 6, 4) indicates that there are three hidden layers, and the number of nodes

Table 2. Data analytics based on the Elman neural network and the Jordan neural network after data normalization (min-max normalization)

Algorithms or methods	Structures (context layers)	*FPR* (%)	*FNR* (%)	*ACC* (%)
Elman	c(3)	25.00	38.46	66.67
	c(8)	100.00	15.38	52.38
Jordan	c(3)	37.50	46.15	57.14
	c(8)	37.50	30.77	66.67
	c(12)	62.50	30.77	57.14

Abbreviations: ACC: Accuracy; FNR: False negative rate; FPR: False positive rate.

Table 3. Data analytics based on the Elman neural network and the Jordan neural network after z-score standardization

Algorithms or methods	Structures (context layers)	*FPR* (%)	*FNR* (%)	*ACC* (%)
Elman	c(3)	25.00	38.46	66.67
	c(8)	87.50	15.38	57.14
Jordan	c(3)	37.50	46.15	57.14
	c(8)	87.50	15.38	57.14
	c(12)	50.00	23.08	66.67

Abbreviations: ACC: Accuracy; FNR: False negative rate; FPR: False positive rate.

Table 4. Data analytics based on the DNN-DBN after data normalization (min-max normalization)

Algorithms or methods	Structures (hidden layers)	*FPR* (%)	*FNR* (%)	*ACC* (%)
DNN-DBN	c(10, 4)	n/a	n/a	n/a
	c(12, 6)	n/a	n/a	n/a
	c(8, 6, 4)	n/a	n/a	n/a

Abbreviations: ACC: Accuracy; DBN: Deep belief network; DNN: Deep neural network; FNR: False negative rate; FPR: False positive rate.

Table 5. Data analytics based on the DNN-DBN after z-score standardization

Algorithms or methods	Structures (hidden layers)	*FPR* (%)	*FNR* (%)	*ACC* (%)
DNN-DBN	c(10, 4)	62.50	38.46	52.38
	c(12, 6)	62.50	38.46	52.38
	c(8, 6, 4)	50.00	46.15	52.38

Abbreviations: ACC: Accuracy; DBN: Deep belief network; DNN: Deep neural network; FNR: False negative rate; FPR: False positive rate.

in the three hidden layers is 8, 6, and 4, respectively. In Table 4, n/a means that no normal result is obtained after the min-max normalization is applied, and structure c(10, 4), c(12, 6), or c(8, 6, 4) is utilized.

5.3. Results of the regular DNN

Part of the results of the regular DNN established on the small dataset with unbalanced data of the LPBF are listed in Table 6 (after min-max normalization) and Table 7 (after z-score standardization). Table 6 shows that the regular DNN after min-max normalization can obtain better results than the DNN-DBN. After z-score standardization was employed, the regular DNN could achieve the best results among the DL methods if a suitable algorithm was used. Most of the relevant *FPR* and *FNR* values are much lower than those of the three DL methods (Elman, Jordan, and DNN-DBN). Data analytics based on traditional ANN was also conducted, and the results are shown in Table 8 for comparison with the results of various DL methods. It was demonstrated that some DL methods, such as Elman, Jordan, and DNN-DBN, might obtain worse results than traditional ANN due to insufficient training and test data if a small dataset was used.

Table 6. Data analytics based on the regular DNN after data normalization (min-max normalization)

Structures (hidden layers)	Algorithms	*FPR* (%)	FNR (%)	*ACC* (%)
c(8, 4)	rprop−	25.00	46.15	61.90
c(8, 5)	rprop−	12.50	53.85	61.90
c(8, 6, 3)	rprop+ or sag	12.50	53.85	61.90

Abbreviations: ACC: Accuracy; DNN: Deep neural network; FNR: False negative rate; FPR: False positive rate

Table 7. Data analytics based on the regular DNN after z-score standardization

Structures (hidden layers)	Algorithms	*FPR* (%)	*FNR* (%)	*ACC* (%)
c(7, 4)	slr	25.00	15.38	80.95
c(8, 4)	slr	37.50	7.69	80.95
c(8, 5)	rprop+	12.50	23.08	80.95
c(8, 6)	rprop-	25.00	23.08	76.19
c(8, 6, 3)	rprop- or sag	0.00	61.54	61.90

Abbreviations: ACC: Accuracy; DNN: Deep neural network; FNR: False negative rate; FPR: False positive rate; slr: Smallest learning rate.

Table 8. Data analytics based on traditional ANN

Data pre-processing	Structures (number of nodes in the hidden layer)	*FPR* (%)	*FNR* (%)	*ACC* (%)
Min-max normalization	c(6)	0.00	53.85	66.67
	c(10)	12.50	53.85	61.90
	c(14)	0.00	53.85	66.67
z-score standardization	c(6)	75.00	69.23	28.57
	c(10)	12.50	38.46	71.43
	c(14)	25.00	53.85	57.14

Abbreviations: ACC: Accuracy; ANN: Artificial neural network; FNR: False negative rate; FPR: False positive rate.

6. Conclusion and future research

Data analytics based on four DL techniques (the Elman neural network, the Jordan neural network, the DNN-DBN, and regular DNN based on the four algorithms ("rprop+", "rprop-", "sag", and "slr") were conducted on a small dataset with unbalanced data of the LPBF. After z-score standardization was employed, the regular DNN could obtain a better *ACC* than the three other DL techniques (Elman, Jordan, and the DNN-DBN), while most of the relevant *FPR* and *FNR* values were much lower than those of the three DL techniques. Future research should consider DL-based data analytics and defect prediction of other defects (e.g., keyholes and cracks) while considering the material properties of LPBF. This will improve the DL model's applicability and enable a more comprehensive defect classification in LPBF processes. DL-based data analytics on more AM datasets (small or big) and the prediction of related defects are also directions for future work.

Acknowledgments

The author thanks the support from Mississippi State University, Mississippi, USA.

Funding

None.

Conflict of interest

The author declares no conflicts of interest.

Author contributions

This is a single-authored article.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

The dataset in this paper is available from the author of reference[10] upon appropriate request.

References

1. Raihan AS, Harper A, Era IZ, *et al*. A data-efficient sequential learning framework for melt pool defect classification in laser powder bed fusion. *J Manuf Processes*. 2025;145:201-210.

   doi: 10.1016/j.jmapro.2025.03.118

2. Ni C, Zhu J, Zhang B, *et al*. Recent advance in laser powder bed fusion of Ti-6Al-4V alloys: Microstructure, mechanical properties and machinability. *Virtual Phys Prototyp*. 2025;20(1):e2446952.

   doi: 10.1080/17452759.2024.2446952

3. Nabavi SF, Dalir H, Farshidianfar A. A comprehensive review of recent advances in laser powder bed fusion characteristics modeling: Metallurgical and defects. *Int J Adv Manuf Technol*. 2024;132(5):2233-2269.

   doi: 10.1007/s00170-024-13491-1

4. Ero O, Taherkhani K, Hemmati Y, Toyserkani E. An integrated fuzzy logic and machine learning platform for porosity detection using optical tomography imaging during laser powder bed fusion. *Int J Extrem Manuf*. 2024;6(6):065601.

   doi: 10.1088/2631-7990/ad65cd

5. Gu Z, Mani Krishna KV, Parsazadeh M, *et al*. Deep learning-based melt pool and porosity detection in components fabricated by laser powder bed fusion. *Prog Addit Manuf*. 2025;10(1):53-70.

   doi: 10.1007/s40964-024-00603-2

6. Zhao J, Yang Z, Chen Q, *et al*. Real-time detection of powder bed defects in laser powder bed fusion using deep learning on 3D point clouds. *Virtual Phys Prototyp*. 202531;20(1):e2449171.

   doi: 10.1080/17452759.2024.2449171

7. Pouyanfar S, Sadiq S, Yan Y, *et al*. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput Surv* (*CSUR*). 2018;51(5):1-36.

   doi: 10.1145/3234150

8. Narasimharaju SR, Zeng W, See TL, Zhu Z, Scott P, Jiang X, Lou S. A comprehensive review on laser powder bed fusion of steels: Processing, microstructure, defects and control methods, mechanical properties, current challenges and future trends. *J Manuf Processes*. 2022;75:375-414.

   doi: 10.1016/j.jmapro.2021.12.033

9. Ganta MG, Kurek M. Influence of post-processing methods on the fatigue performance of materials produced by selective laser melting (SLM). *Int J Adv Manuf Technol*. 2025;9:1-32.

   doi: 10.1007/s00170-024-14920-x

10. Wang GW. *Microstructure and Mechanical Properties of Oxide Dispersion Strengthened Nickel-Based Superalloys by Laser Additive Manufacturing*. [Dissertation, Zhongnan University, China]; 2023. Available from: https://www.cnki.net [Last accessed on 2025 May 28].

11. Bramer M. *Data for Data Mining. Principles of Data Mining*. London: Springer; 2016. p. 9-19.

    doi: 10.1007/978-1-4471-7307-6

12. Alabdulwahab S, Moon B. Feature selection methods simultaneously improve the detection accuracy and model building time of machine learning classifiers. *Symmetry*. 2020;12(9):1424.

    doi: 10.3390/sym12091424

13. Maseer ZK, Yusof R, Bahaman N, Mostafa SA, Foozy CF. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access*. 2021;9:22351-22370.

    doi: 10.1109/ACCESS.2021.3056614

14. Mohammed JZ, Wagner M. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge: Cambridge University Press; 2014.

15. Lewis ND. *Deep Learning Made Easy with R. A Gentle Introduction for Data Science*. South Carolina: CreateSpace Independent Publishing Platform; 2016.

16. Elman JL. Finding structure in time. *Cogn Sci*. 1990;14(2):179-211.

    doi: 10.1207/s15516709cog1402_1

17. Jordan MI. Serial order: A parallel distributed processing approach. *Adv Psychol*. 1997;121:471-495.

18. Jang H, Plis SM, Calhoun VD, Lee JH. Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: Evaluation using sensorimotor tasks. *Neuroimage*. 2017;145:314-328.

    doi: 10.1016/j.neuroimage.2016.04.003

19. Ghasemi F, Mehridehnavi A, Fassihi A, Pérez-Sánchez H. Deep neural network in QSAR studies using deep belief network. *Appl Soft Comput*. 2018;62:251-258.

    doi: 10.1016/j.asoc.2017.09.040

20. Hinton GE. Training products of experts by minimizing contrastive divergence. *Neural Comput*. 2002;14:1771-800.

    doi: 10.1162/089976602760128018

21. Hinton G. A practical guide to training restricted Boltzmann

machines. *Momentum*. 2010;9(1):926.

22. Fischer A, Igel C. Training restricted Boltzmann machines: An introduction. *Pattern Recognit*. 2014;47(1):25-39.

    doi: 10.1016/j.patcog.2013.05.025

23. Hann J, Pei J, Kamber M. *Data Mining: Concepts and Techniques*. Netherlands: Elsevier; 2011.

24. Chung H, Lee SJ, Park JG. Deep Neural Network using Trainable Activation Functions. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE; 2016. p. 348-352.

    doi: 10.1109/IJCNN.2016.7727219

Appendix

Table A1. Acronyms

AM	Additive manufacturing
ANN	Artificial neural network
ACC	Accuracy
CD	Contrastive divergence
CNN	Convolutional neural network
CT	Computed tomography
DBN	Deep belief network
DED	Directed energy deposition
DL	Deep learning
DNN	Deep neural network
DOE	Design of experiments
FN	False negative
FNR	False negative rate
FP	False positive
FPR	False positive rate
LOF	Lack of fusion
LPBF	Laser powder bed fusion
ML	Machine learning
MPC	Model predictive control
RBM	Restricted Boltzmann machine
RNNs	Recurrent neural networks
SLM	Selective laser melting
TN	True negative
TP	True positive