Machine Learning in Additive Manufacturing: A Review

Lingbin Meng[1], Brandon McWilliams[2], William Jarosinski[3], Hye-Yeong Park[4], Yeon-Gil Jung[4],

Jehyun Lee[4], Jing Zhang[1]*

1. Department of Mechanical and Energy Engineering, Indiana University-Purdue University
Indianapolis, Indianapolis, IN 46202, USA

2. CCDC Army Research Laboratory, Aberdeen Proving Ground, MD 21005, USA

3. Praxair Surface Technologies, Indianapolis, IN 46222, USA

4. Department of Materials Science and Engineering, Changwon National University, Changwon
51140, Republic of Korea

*Corresponding author: jz29@iupui.edu

Abstract

In this review article, the latest applications of machine learning (ML) in additive manufacturing (AM) field are reviewed. These applications, such as parameter optimization and anomaly detection, are classified into different types of ML tasks, including regression, classification, and clustering. The performance of various ML algorithms in these types of AM tasks are compared and evaluated. Finally, several future research directions are suggested.

Keywords: additive manufacturing; machine learning;

Nomenclature	
Abbreviation	**Meaning**
3D	Three dimensional
AE	Acoustic emission
AI	Artificial intelligence
AM	Additive manufacturing
BJ	Binder jetting
BoW	Bag of words
BP	Backpropagation
CNN	Convolutional neural network
CT	Computed tomography
CV	Cross-validation
DA	Discriminant Analysis
DBN	Deep belief network
DED	Direct energy deposition
DT	Decision tree
FFF	Fused filament fabrication
FN	False negative
FP	False positive
GP	Gaussian process
KNN	k-nearest neighbors
LOOCV	Leave-one-out cross-validation
L-PBF	Laser powder bed fusion
ME	Material extrusion
MJ	Material jetting
ML	Machine learning
NN	Neural network
PBF	Powder bed fusion
PCA	Principal component analysis
PSP	Process-structure-property
RF	Random forest
RMSE	Root mean square error
RT	Regression tree
SL	Sheet lamination
SOM	Self-organizing map
SVM	Support vector machine
TN	True negative
TP	True positive
UQ	Uncertainty Quantification

1. Introduction

Machine learning (ML), a subset of artificial intelligence (AI), has increasingly become popular in additive manufacturing (AM) research. Additive manufacturing, also known as 3D printing or rapid prototyping (RP), is defined as a group of layer-upon-layer fabrication processes controlled by a computer-aided design (CAD) model [1, 2]. Machine learning is defined as computer programming to optimize a performance criterion using example data or past experience [3]. For machine learning in additive manufacturing, besides the typical application of making predictions through data fitting, the research community is exploring new and innovative approaches to integrate ML and AI methods into AM. ML algorithms, applications, and platforms are helping AM practitioners improve product quality, optimize manufacturing process, and reduce costs.

A major challenge in current AM field is the inconsistency of the quality of the printed products, which are highly dependent on numerous processing parameters, such as printing speed and layer thickness. These process-structure-property (PSP) relationships have been discussed in many review articles [4-6]. One method to address this challenge is conducting experiments or high-fidelity simulations [7, 8] to obtain reliable data and help optimize the processing parameters, but both of them are either time-consuming or expensive, and sometimes both. Another method to ensure part quality and process reliability is the application of *in situ* monitoring systems [9], but an efficient way for defect detection using the *in situ* data such as images is needed. In both methods, there is a critical need of an effective and efficient tool for data analysis and data mining. This need is being addressed by a subset of artificial intelligence (AI) known as ML.

With a reliable training dataset, the ML models learn knowledge from the training set and make inference based on the knowledge. On one hand, the trained machine learning models can make predictions and determine the optimal processing parameters in an efficient way. On the other hand, it can also deal with *in situ* data for defect detection in real time. Some other ML applications, such as geometric deviation control, cost estimation, and quality assessment, are also reported in recent literature. In general, the ML applications can be regarded as the art of data manipulation. This capability makes ML a key aspect of Industry 4.0 [10].

Machine learning tasks can be divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning [3]. Figure 1 displays the taxonomy of ML with the corresponding applications in AM field. In supervised learning, each input datum is labeled with an output Y, and the training set consists of many input-output pairs. Each input is a vector contains all involved features, $X_1, X_2, ..., X_n$, that may affect its output. Each output can be a target classification such as quality assessment (good or bad) and the corresponding ML category is classification, or a target parameter such as porosity and tensile strength and the corresponding ML category is regression. In unsupervised learning, each input datum doesn't come with an output, and the model will study the relationship among input data. A typical application of unsupervised learning is clustering, in which all data are clustered into groups based on their similarity. Reinforcement learning, on the other hand, is learning how to map situations to actions so as to maximize a numerical reward signal [11], the applications of which include self-driving car and chess. Figure 1 illustrates some example applications in AM field with their corresponding ML categories. In AM field, most of the ML applications fall into the supervised learning category.
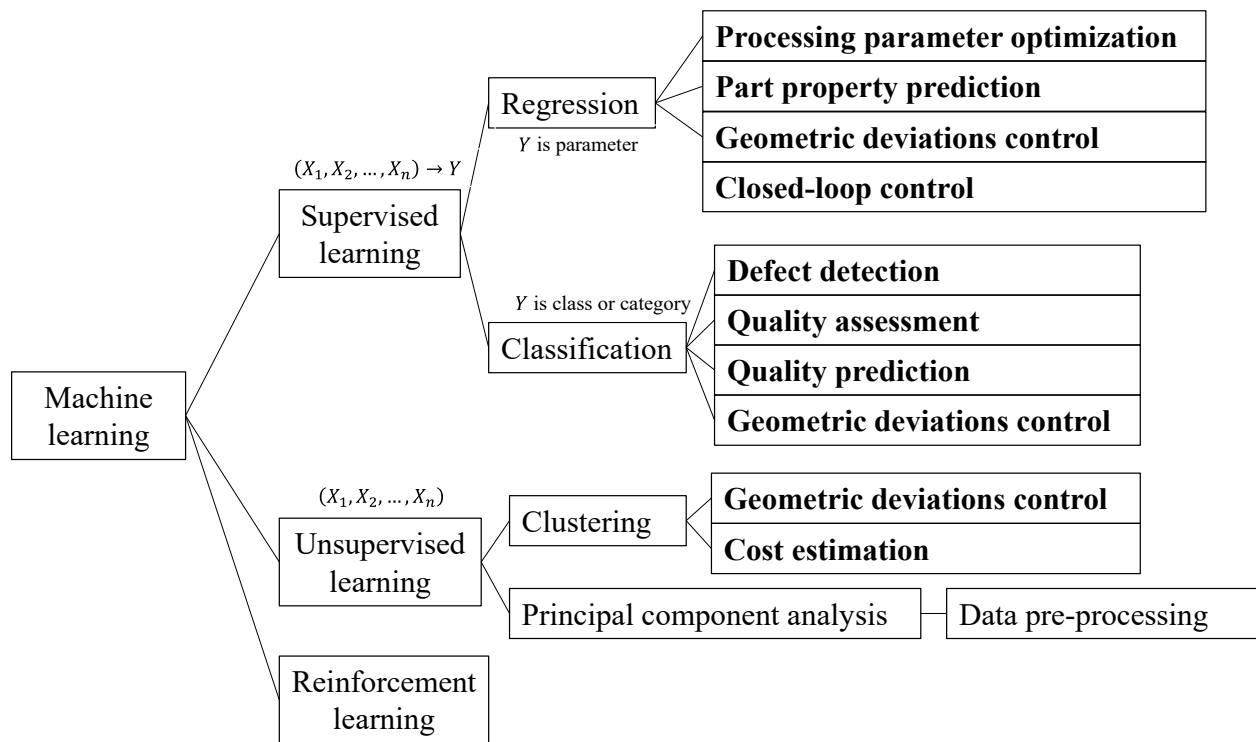
Figure 1. Taxonomy of ML applications in AM field. Text outside box is the data type. Text in bold is the ML applications in AM. $(X_1, X_2, \ldots, X_n)$ is the input vector containing all input features and Y is the output.

The objective of this review article is to present the latest applications of machine learning in AM field, and thus provide a starting point for AM practitioners and researchers who are interested in applying ML. Very recently, there are a few related review articles [12] and reports [13] available. While Ref. [12] focuses on the applications and challenges of only neural network (NN), and Ref. [13] focuses more on the data acquisition in AM field, this work focuses on providing guidance on how to generate ideas in applying ML in AM field, distinguish the type of ML tasks, and make selection of different ML models. The whole paper is organized as follows: The latest applications of ML in AM field are first reviewed from the perspective of AM in Section 2, and then classified into different categories of machine learning tasks, including supervised learning (Section 3) and unsupervised learning (Section 4). The performance of various machine learning algorithms applied in recent literature are compared and evaluated in Section 3 and Section 4. Finally, in Section 5, the work is summarized, and several future research directions are suggested.

2. ML Applications in AM

ML is a data manipulation tool. Figure 2 demonstrates various types of data available to be analyzed and utilized in the PSP relation chain. The "process" term in the widely used PSP relationships is partitioned into two terms, "processing parameter" and "processing resultant data", in order to distinguish data available before the process and during the process. There are many relationships between these data, including but not limited to: (1) the processing parameters, such as extruder temperature in ME, laser power in laser powder bed fusion (L-PBF), printing speed, and layer thickness, significantly affect the structure of the printed parts, and thus dominate their quality and performance; (2) the designed shape play a crucial role in the printing cost and the geometric deviation of the printed products; (3) the *in situ* images and acoustic emission (AE) acquired by the monitoring systems are available to detect the occurrence of defect and its type in real time. Therefore, if a dataset, which consists of at least two types of

related data in the PSP relation chain, is used to train ML models, the ML models will be able to make inference based on these data. This is the general procedure to apply ML models.
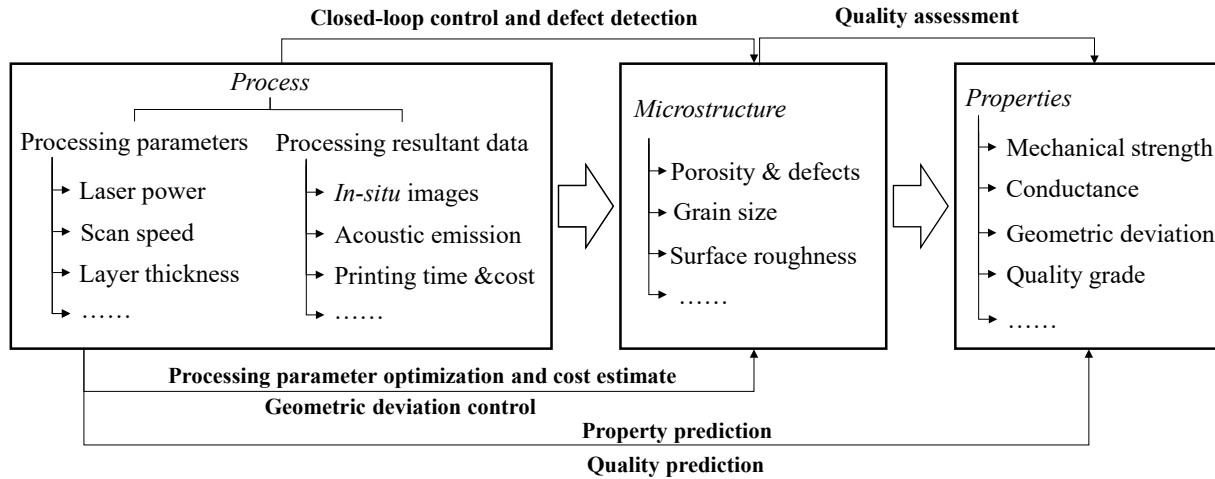

Figure 2. The process-structure-property relationship chain in additive manufacturing. Texts in the boxes represent the available data that can be used in machine learning. Bold texts represent some existing machine learning applications in additive manufacturing field. The origin and the end of each arrow represent the input and output data, respectively.

2.1. Processing parameters optimization and property prediction

For designers, the quality of a part using a certain combination of processing parameters will remain uncertain until it is finally printed. Therefore, a series of efforts, such as printing some samples and testing their performance, have to be made to ensure the part quality, which makes the design process expensive, time-consuming and dynamic. In this regard, a direct relationship between the processing parameters and part quality is strongly desirable. To this end, experiments and simulations are useful methods to help construct such a relationship, but it's impractical to obtain optimal processing parameters using the two methods when a large amount of input features is involved. ML models, on the other hand, can be applied as surrogate models to assist process optimization.

Given a series of reliable training data of the property of interest (output) at some combinations of processing parameters (input), a process map can be generated by these discrete data points using ML regression models. Figure 3(a) demonstrates a processing map of melt pool depth

(output) in terms of laser power and scan speed (input) of 316L strainless steel in L-PBF process [14]. The applications of the process map is twofold: (1) it can make predictions to the output at any combinations of input features as a surrogate model and therefore reduce the demand of experimental and computational study, and (2) it can provide the relevance of each input feature to the output so as to obtain optimal input combination. Figure 3(b) plots the uncertainty and the discrete data points used to generate the process map. The uncertainty from the ML model is part of epistemic uncertainties in uncertainty quantification (UQ) [15]. Recently, Meng and Zhang adopted the approach to develop the process design maps of two metals, 316L and 17-4 PH stainless steels [16]. Their studies show that the keyhole mode criteria need to be revised based on the specific metal composition and powder layer thickness. The process map enables designers to achieve property prediction and process optimization efficiently. Since the process map is a typical production of ML regression models, the recent applications from literature in this topic are reviewed in Section 3.1.
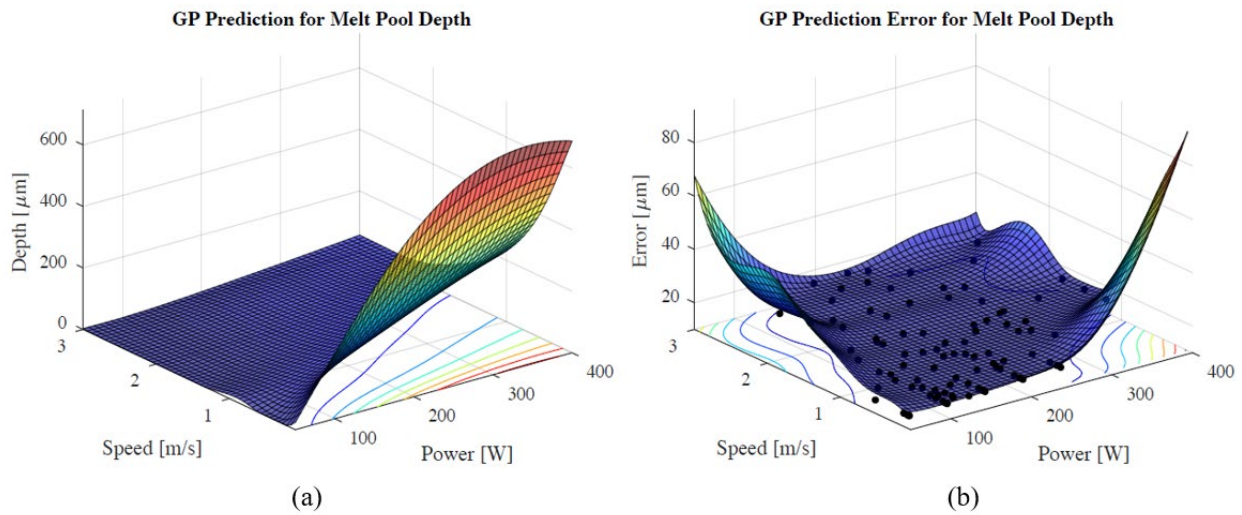

(a)   (b)

Figure 3. (a) The process map of melt pool depth in terms of laser power and scan speed of 316L strainless steel in L-PBF process, and (b)the corresponding uncertianty. The data points used to generate the process map are plotted in (b). Reprinted with permission from reference[14].

2.2. Defect detection, quality prediction and closed-loop control

The development of the *in situ* monitoring systems enables the acquisition of real time data that can be used for defect detection and closed-loop control for AM [9]. These real time data, including spectra, images, AE and computed tomography (CT), can be utilized by ML models in

several ways: (1) label these data with defect (possibly with defect types) or not by experimental results or human knowledge, and then use the labelled data to train supervised learning models for defect detection and quality prediction in real time, which is a typical application of ML classification models and will be discussed in Section 3.2; (2) conduct cluster analysis using unsupervised learning models to cluster the abnormal data so as to achieve defect detection without the labelling process, which will be discussed in Section 4.1; (3) train the ML regression models along with the data of some real-time controllable processing parameters, in order to tune these processing parameters in real time. An example of the third way is the voltage level control in MJ process by Wang *et al.* [17]. Their process control framework consists of three main parts, as demonstrated in Figure 4. First, a charge-coupled device (CCD) camera is used to capture the dynamic images for the droplet. Second, four properties (satellite, ligament, volume, and speed) of the droplet are extracted from the images to train a neural network (NN) ML model along with the current voltage. Third, the trained ML model is then used to determine the optimized voltage level and send it to the voltage adjustment system to control the droplet jetting behavior.
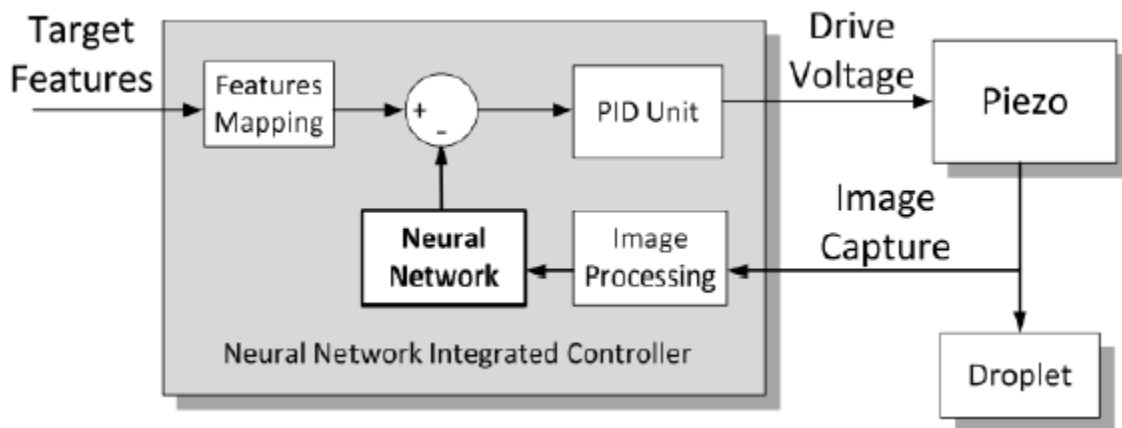

Figure 4. The closed-loop voltage control framework in MJ process. Reprinted with permission from reference [17].

2.3. Geometric deviation control

Low geometric accuracy and poor surface integrity are common defects of AM parts [18]. These geometric defects impede the applications of AM in several industries, such as aerospace and medical [19]. In this regard, ML models are capable of identifying the occurrence of geometric defect, quantifying the geometric deviation, and providing guidance of geometric error

compensation. For instance, Francis *et al.* [20] developed a geometric error compensation framework for L-PBF process using convolutional neural network (CNN) ML model, shown in Figure 5. Using thermal history and some processing parameters as input and distortion as output, the trained ML model is capable of predicting distortion which is then imported reversely to the CAD model to achieve error compensation. By this means, the geometric accuracy of parts fabricated by the compensated CAD model will be significantly improved.
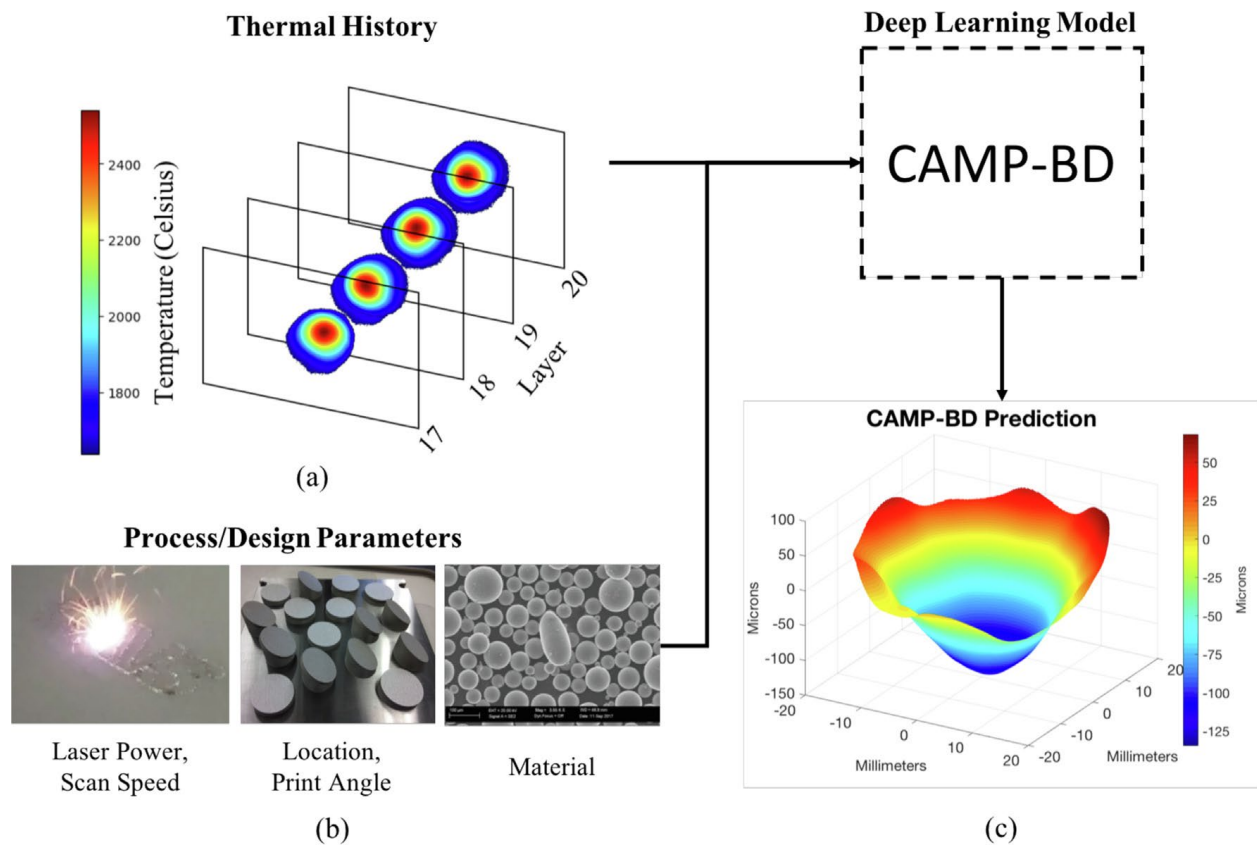

Figure 5. Procedure of geometric error compensation of Ti-6Al-4V in L-PBF process. The input data are the thermal history and some processing parameters. The output data are the distortion. Error compensation is achieved by reverse the distortion in the CAD model. CAMP-BD represents Convolutional and artificial neural network for Additive Manufacturing Prediction using Big Data. Reprinted with permission from reference [20].

2.4. Cost estimation

The printing cost and time are significant components of information shared between the manufacturers, clients and other stakeholders within the supply chain. Although they can be

roughly estimated by the volume of the designed shape, there is still a need of a more accurate and efficient tool for cost estimation. Recently, an application of cost estimation by Chan *et al.* [21] is reported. Figure 6 demonstrates the cost estimation framework they proposed: (1) a client submits a manufacturing job with a 3D model; (2) features are generated from the 3D model and form the input vector, which is then imported to the trained ML models for cost prediction based on similar jobs using clustering analysis; (3) if client prefers or the training dataset size for ML models is small, the 3D model will be forwarded to simulation models to predict the cost, which will also become training data for ML models; (4) the final predicted cost is estimated by combining the ML and simulation predictions; (5) the final prediction is forwarded to the client.
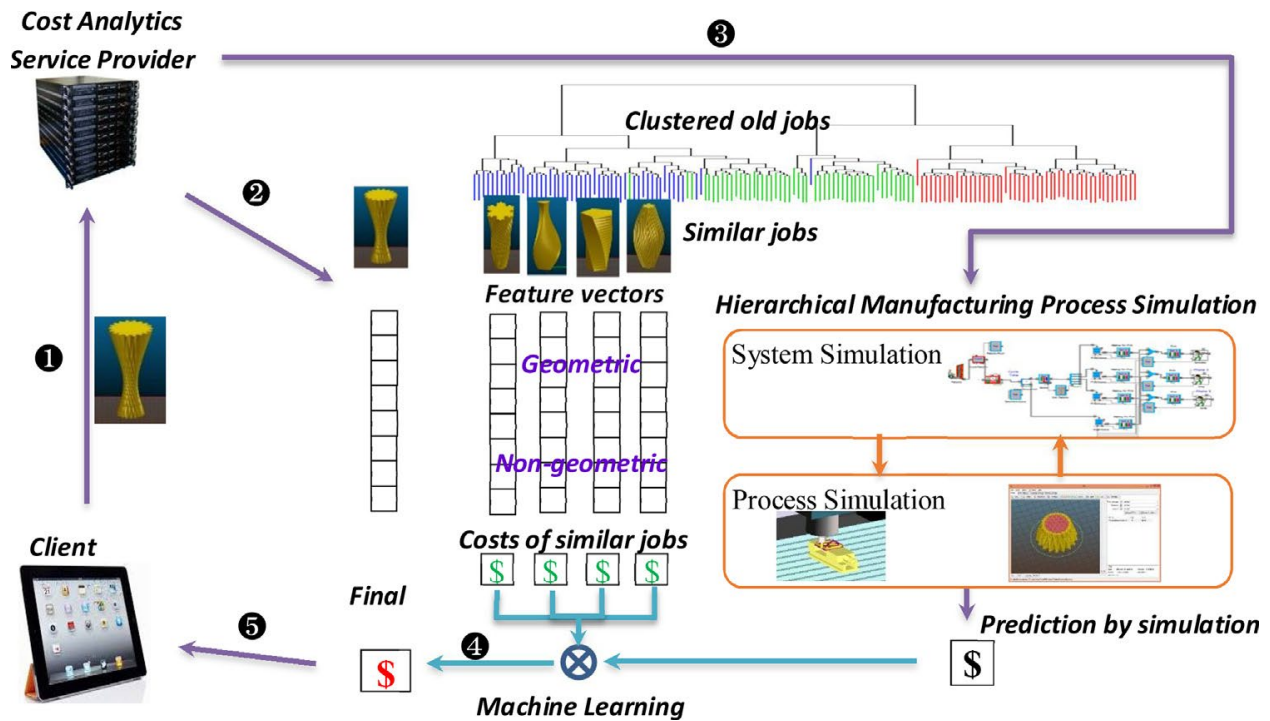

Figure 6. The cost estimation framework based on geometry and process similarities. Reprinted with permission from reference [21].

3. Supervised learning

So far, the latest applications of ML in AM have been reviewed in Section 2 from the perspective of AM. From now on, these applications are classified into different categories of ML tasks in Figure 1. This is important to make selection of ML models for some reasons: (1) even with same applications, the ML models applied may be different with different data type, such as defect detection which can be achieved by both supervised learning and unsupervised learning

models; (2) ML models tend to possess similar performance in the same categories of ML tasks using similar data type and dataset size. In this regard, Section 3 and Section 4 aim at providing guidance of making selection of data type and corresponding ML models.

An indispensable factor in applying ML in AM field is the data acquisition. A ML model requires sufficient data to make accurate predictions. The required number of training data also increases exponentially with the increasing number of input features. However, in many applications, the acquisition and labelling of data requires high experimental, computational, and/or laboring costs. Therefore, before a ML model is applied, the dimension of a ML task should be determined carefully considering the number of available data and the cost to obtain them. Section 3 and Section 4 also list many examples of ML applications in literature that can help determine the task dimension.

In supervised learning, all input data are labeled with an output. The output can be either the parameters or the corresponding ML task is regression, or classes and the corresponding ML task is classification. Since most of the ML applications in AM aim at predicting a target parameter or class, supervised learning is the major type of ML applications in AM field.

3.1. Regression

In regression tasks, the output of each input is parameters, such as porosity of the printed products, efficiency, melt pool depth, mechanical property, etc. The AM algorithm learns the relevance between the input and output parameters from the training dataset, and then makes inference from a new input to its output using the relevance it learns.

3.1.1. Regression applications in AM

The major functionality of ML regression models in AM field is the generation of process map, which has been discussed in Section 2.1. Therefore, processing parameters optimization and property prediction will be the two major applications of ML regression models. In addition, since the targets in geometric deviation control and cost estimation are all parameters, they may also be the applications of ML regression models.

3.1.2. Regression models assessment in AM applications

Table 1 shows the recent regression applications along with the ML models in AM field. According to Table 1, the two major ML models for regression tasks applied in AM field in recent literature are neural network (NN) and Gaussian process (GP).

Table 1. ML regression applications in AM.

Applications	Inputs	Outputs	Models
Geometric deviations control [22]	Shape parameters	Shape deviation parameters	Gaussian process (GP)
Processing parameters optimization [14]	Laser power and scan speed	Melt pool depth	GP
Processing parameters optimization [23]	Laser power and scan speed	Porosity	GP
Processing parameters optimization [24]	Laser power and scan speed	Melt pool depth	Regression tree (RT), GP
Processing parameters optimization [16]	Laser power and scan speed	Melt pool depth and width depth ratio	GP
Trace geometry prediction [25]	Laser power and scan speed, and powder feeding rate	Deposited trace cross-section geometrical parameter	NN
Processing parameters optimization [26]	Layer thickness, layer power, hatch spacing, laser speed, interval time, surroundings temperature, and scanning mode	Shrinkage ratio	NN
Property prediction [27]	Material property, extruder temperature, printing speed, layer thickness	Tensile strength	Support vector regression, random forest (RF), recurrent NN
Property prediction [28]	108 input features including extruder temperature, printing speed, and layer thickness	Surface roughness	RF, AdaBoost, RT, support vector regression (SVR), Ridge regression, NN,

			and ensemble of them
Thermal history prediction [29]	Toolpath feature, the time of deposition, closest distance to the boundary of the build, layer height, laser intensity, and laser state	Thermal history	Recurrent NN
Real-time composition monitoring [30]	Spectral line-intensity-ratio and spectral integrated intensity	Element composition	SVR, NN,
Geometric deviations control [20]	Thermal history, laser power and scan speed, printing location and angle, and material	Distortion	CNN
Geometric deviations control [31]	Deformed note locations	Original note locations	NN
Closed-loop control [17]	Droplet features (satellite, ligament, volume, and speed)	Voltage level	NN

Artificial neural networks, inspired by biological neural systems, are computing systems consist of massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations [32]. All the "neural network" or "NN" in this paper refer to artificial neural networks, instead of biological neural systems. A typical neural network contains an input layer, one or more hidden layers and one or more output layers. Each layer is made of numerous neurons. The information of each neuron is propagated to the next layer based on it's weight. A NN will be categorized to recurrent NN when the propagation of its neurons forms cycles, and feedforward NN otherwise. During training, the weight of each neuron is optimized by the learning rule as soon as a new observation is imported into the NN. The most popular learning rule for NN is the backpropagation (BP) algorithm [33], which adjusts the weights based on the gradient descent. However, due to the strong learning ability of BP algorithm, NN usually suffers from overfitting issue (More discussions in Section 3.3), which can be alleviated by either early stopping method or regularization [34, 35]. For more knowledge about NN, refer to Ref. [36].

Caiazzo *et al.* [25] applied BP-NN for trace geometry prediction with RMSE of around 5% using 30 tranining data. Rong-Ji *et al.* [26] tested the performance of BP-NN with 5 to 10 hidden neurons and their results exposed the trend that more hidden neurons tend to make better predictions. Zhang *et al.* [27] used recurrent NN in ME process to predict the tensile strength of

the printed products and the RMSE was around 2%. Figure 7 [27] illustrates the NN they constructed: during training (bottom), the output of each input combination are progagating backward in the NN to adjust the relavance of each input feature, and after training (top), new combinations of input features are propagating forward to predict tensile strength. Their result exposed that recurrent NN has outperformed the random forest and support vector regression algorithms in this application. Overall, NN displayed excellent performance in regression tasks, but also required tuning a series of hyperparameters such as number of hidden neurons and layers [12].
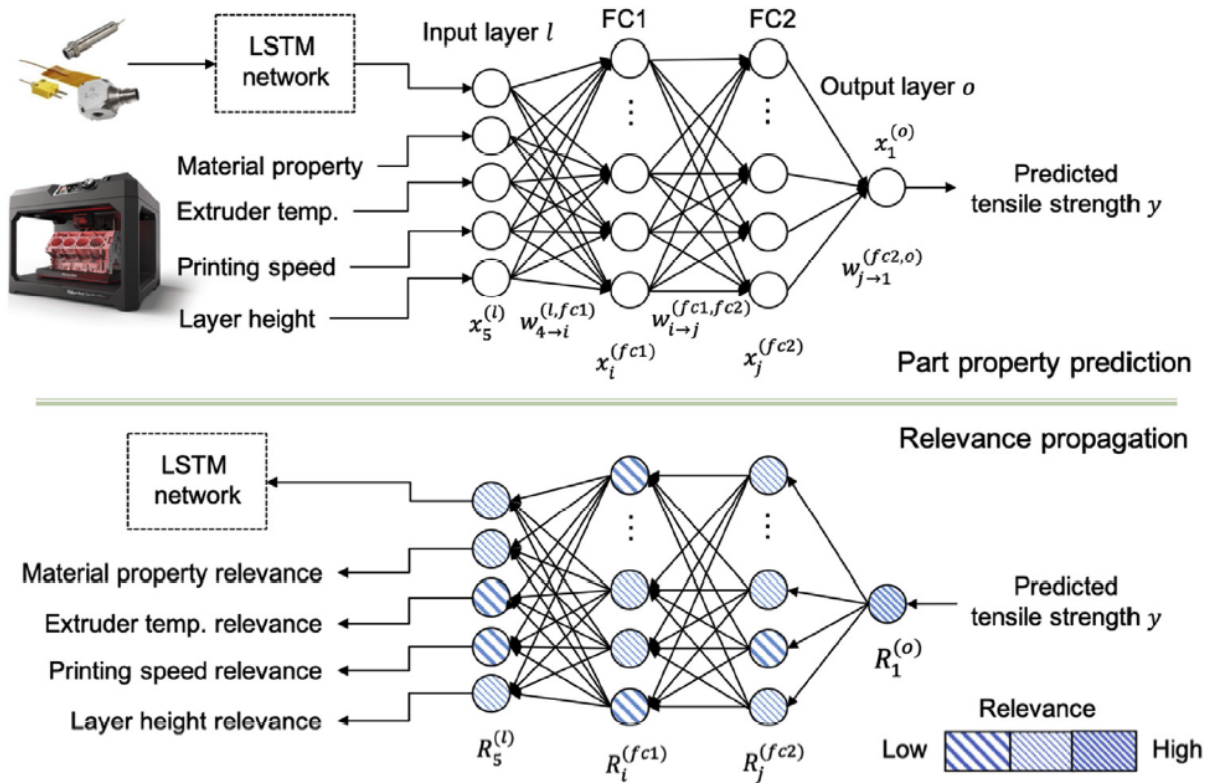

Figure 7. Layer-wise relevance propagation through the trained neural network for polylactic acid (PLA) in fused filament fabrication (FFF): porpagation forward for prediction of tensile strength (top) and propagation backward for training the relevance of each input feature (bottom). Reprinted with permission from reference[27].

A Gaussian process is defined as a collection of random variables, any finite number of which has a joint Gaussian distribution [37]. Similar to the mean value and variance in Gaussian distribution, a GP is completely specified by a mean function $m(X)$ and a covariance function

$C(X, X^*)$, where X is the input vector containing all input features. The covariance function is defined by a single or a combination of kernel functions and is critical to the performance of GP as it captures the spatial dependence between two different locations, X and X^*. The selection of kernel functions should be based on practical applications and has been discussed in Ref. [37].

Tapia *et al.* [14, 23] applied GP to make predictions of porosity and melt pool depth in L-PBF process. The process map of melt pool depth in terms of laser power and scan speed with the corresponding uncentainty are plotted in Figure 3. GP shows excellent regression performance in noisy environment and with limited number of training data. The GP can be used calibrate a convinient criterion [14] to aviod porous formation due to keyhole mode [38] in L-PBF process. An exmaple of this calibration is demonstrated in Ref. [16], where the process maps of 316L and 17-4 PH stainless steels are generated by GP model using experimental dataset and used to compare anaginst and calibrate the normalized enthalpy criterion [14]. Zhu *et al.* [22] applied GP to make predictions on shape deviation and the RMSE is around 3% using 75% of the whole dataset as the training set. Overall, GP is efficient and effective in regression tasks with a few input features and a small dataset. GP may lose its efficiency when the number of involved input features is large or the size of the training dataset is too large, due to the high computational costs in performing matrix inversion.

In general, both GP and NN are capable of handling regression tasks in AM field. NN is more complicated than GP and requires more knowledge to tune the hyperparameters. The selection of ML models should be based on the complexity of the training dataset (i.e. the number of training data points and input features). For low complexity tasks, GP is recommended. For high complexity tasks, NN is recommended. The application of an ensemble of multiple algorithms [28] (including NN) are also reported, which predict more accurately than NN and can be regarded as an alternative.

3.2. Classification

In classification tasks, the output of each input is a class or a category, such as different defect types or quality assessment grade. Similar to regression tasks, the ML models learn how to make classification from the training set, and then use the knowledge to classify new input.

3.2.1. Classification applications in AM

In AM field, there are various classes with different criteria that can be used to distinguish part quality, such as defect and non-defect, quality is good or bad, quality grade assessment on a scale of 1 to 10 to quantify the quality, etc. If a ML model is trained by some classification examples at different input settings, it will be able to make classification to new input henceforth. Therefore, ML classification models can be used in AM field in three main aspects: (1) it can use *in situ* data, such as images and AE, to make predictions of defects so as to help defect detection in real time, (2) it can predict the part quality at different processing parameters, and (3) it can assist quality assessment using the geometric information of printed parts. As the geometric deviation can be described by several types, such as translation and rotation, ML classification models can also achieve geometric deviation control.

3.2.2. Classification performance assessment method

An assessment method is necessary to quantify the performance of a classification model. Classification tasks can be further divided into two subgroups: (1) binary problems, in which only two categories are involved, and (2) multiclass problems, in which at least three categories are involved. The performance of ML algorithms in classification tasks is usually assessed by precision, recall, or F_1 score in binary problems, and accuracy in multiclass problems.

Table 2 displays the confusion matrix of binary classification problems. Precision is defined as $\frac{TP}{TP+FP}$ and represents the ability of a model to identify only the relevant instances, whereas recall is defined as $\frac{TP}{TP+FN}$ and represents the ability of a model to find all the relevant instances. As there is usually a trade-off between precision and recall, F_1 score is defined as $2 \times \frac{Precision \times Recall}{Precision + Recall}$ and represents the overall performance of a model. The range of F_1 score is from 0 to 1, and the larger the F_1 score, the better the performance. Accuracy is defined as the total number of correct predictions over all predictions, or $\frac{TP+TN}{TP+TN+FP+FN}$ in binary problems, but it may not be appropriate in binary problems when the number of positive and negative samples is imbalanced.

Table 2. Confusion matrix of binary classification problems.

Prediction \ Ground truth	Positive	Negative
Positive	True positive (TP)	False positive (FP)
Negative	False negative (FN)	True negative (TN)

3.2.3. Classification models assessment in AM applications

Table 3 shows recent classification applications along with the ML models in AM field. Typical ML algorithms for classification tasks are decision trees (DT), support vector machines (SVM), and convolutional neural networks (CNN).

Table 3. ML classification applications in AM.

Applications	Inputs	Outputs	Models
Design feature recommendation [39]	Target components	Recommended AM feature	Support vector machines (SVM), dendrogram
Defect detection [40]	Regions of interest of spatters, plume and melt pool	Class 1,2 or 3	SVM, convolutional neural networks (CNN)
Defect detection [41, 42]	*In situ* images	Defect type	Bag of words (BoW), CNN
Defect detection [43]	Spectral intensity graph	Defect or not	SVM
Defect detection [44]	Melt pool characteristics	Porous or not	Decision trees (DT), k-nearest neighbors (KNN), SVM, Discriminant Analysis (DA)

Quality assessment [45]	Dimensional variation	Infill classes	KNN, naive Bayes, NN, SVM, DT
Quality prediction [46]	Energy density, particle distribution and surface morphology	Quality: Good or bad	SVM
Defect detection [47]	AE	Defect type	Deep belief network (DBN), SVM, NN
Geometric deviation control [48]	Voxel grid	Deformation type	CNN
Defect detection [49]	25 thermal features	Porosity label and normalized porosity size	KNN, NN, Self-organizing error-driven neural networks (SOEDNN)
Defect detection [50]	CT image layers	Defect or not	SVM
Defect detection [51]	In situ images	Defect or not	CNN
Quality prediction [52]	AE	Poor, medium or high quality	Spectral convolutional neural networks (SCNN)

Decision trees [53] are a type of common ML algorithm for classification tasks. Compared with NN, decision trees are more interpretable. Khanzadeh *et al.* [44] and Tootooni *et al.* [45] applied multiple ML models including DT for defect detection and quality assessment, respectively. In both articles, DT shows medium performance among many classifiers. Overall, DT is a relatively simple method and is capable of dealing with classification tasks in AM field. Though it may not perform the best, it is recommended as a contrast when applying other models to better show the performance of other models.

Support vector machine is designed to deal with binary classification problems [54], but it can also be generalized to multiclass problems [55]. In binary problems, as each input-output pair in training set consists of a high dimensional input vector containing all input features and a target category as output, SVM uses a hyperplane in the high dimensional space to partition the two

groups. According to Table 3, SVM is a very popular classifier in AM applications. In the comparison of multiple classifiers [44, 45], SVM shows comparable performance with other algorithms.

While SVM is good at handling inputs consist of only parameters or classes, it can also be applied in image-based problems. Figure 8 [44] demonstrates a procedure using images as input for defect detection of Ti-6Al-4V in L-PBF process. For each thermal image labeled with either porous or not porous, some geometric features are extracted from the image and used to train the ML models. Zhang *et al.* [40] applied SVM for defect detection using *in situ* images as input. In their article, though CNN performs better (92.8% accuracy), SVM shows 90.1% accuracy in this three-group classification task. Ye *et al.* [47] applied SVM for defect detection using AE as input, which also requires a feature extraction procedure like images. In this binary classification problem, SVM (98.01% accuracy) outperformed the deep belief network (95.87%). Gobert *et al.* [50] applied SVM for defect detection using CT image layers as input, and the F_1 score (refer to Section 2.2.2) of their optimized SVM model is 0.62. Overall, SVM is a great alternative in classification problems.
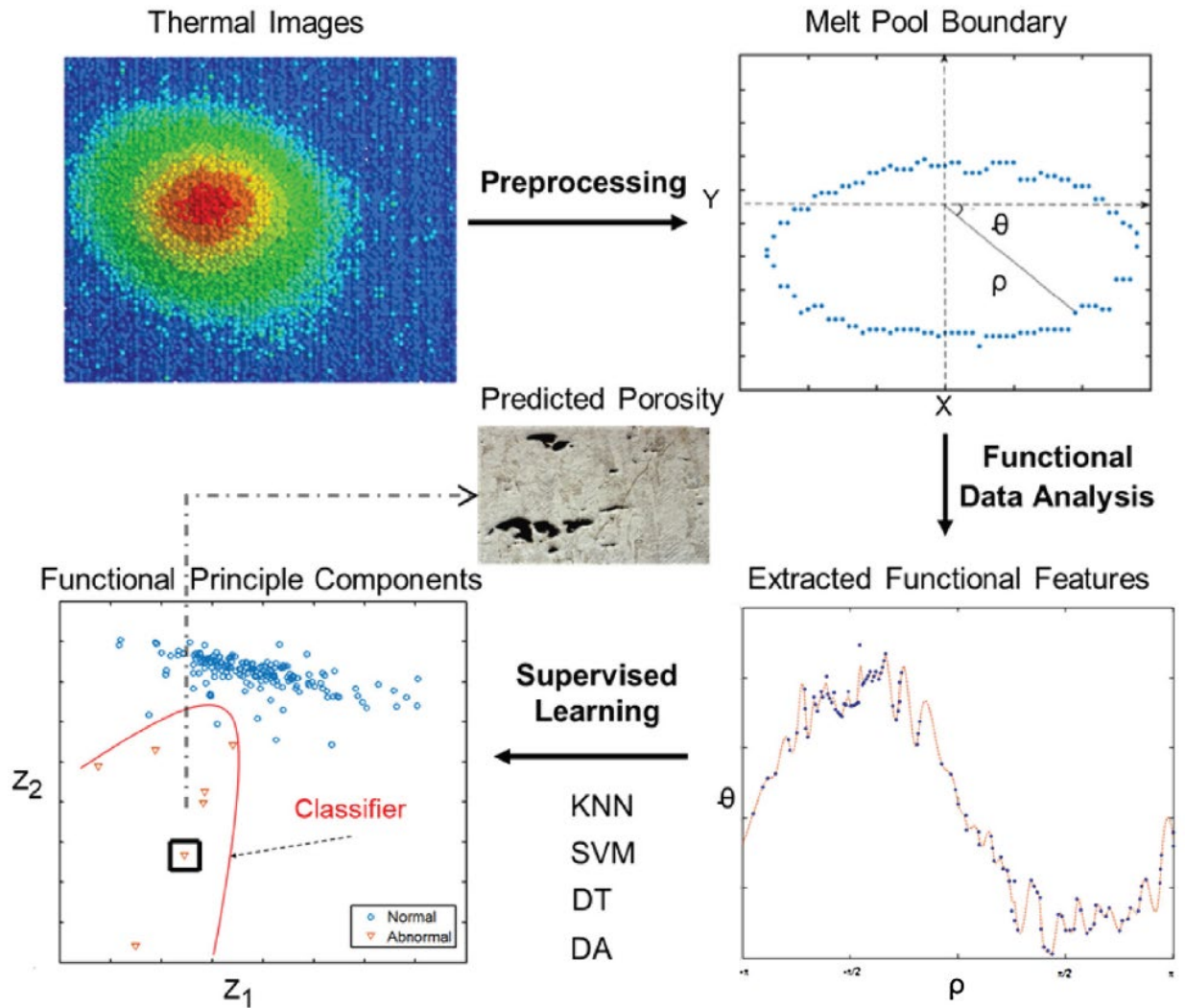
Figure 8. The procedure from thermal images (input) to porosity predictions (output) of Ti-6Al-4V in L-PBF process. Some geometric features are extracted from the thermal images to train the ML models, which can then classify whether the printed product is porous (abnormal) or not (normal). Reprinted with permission from reference [44].

Neural network tends to be the most popular algorithm in classification problems. While normal NN is usually applied in problems with input consists of only parameters and classes, a special type of NN, known as convolutional neural network (CNN), is designed to handle problems with images and AE [56]. Scime *et al.* [41, 42] applied mutli-scale CNN for defect detection using *in situ* images and the overall, anomaly detection, and anomaly differentiation accuracies are 97%, 85%, and 93%, respectively. The multi-scale CNN they implemented is demonstrated in Figure

9. The information of images is propagated in the NN using convolution. Shen *et al.* [48] applied CNN for geometric error compensation using voxel grid as geometric input feature and got an overall F_1 score (refer to Section 2.2.1) of 0.95. Overall, NN is a complex but strong model among the existing algorithms for classification tasks in AM field. NN is applicable in most classification tasks.
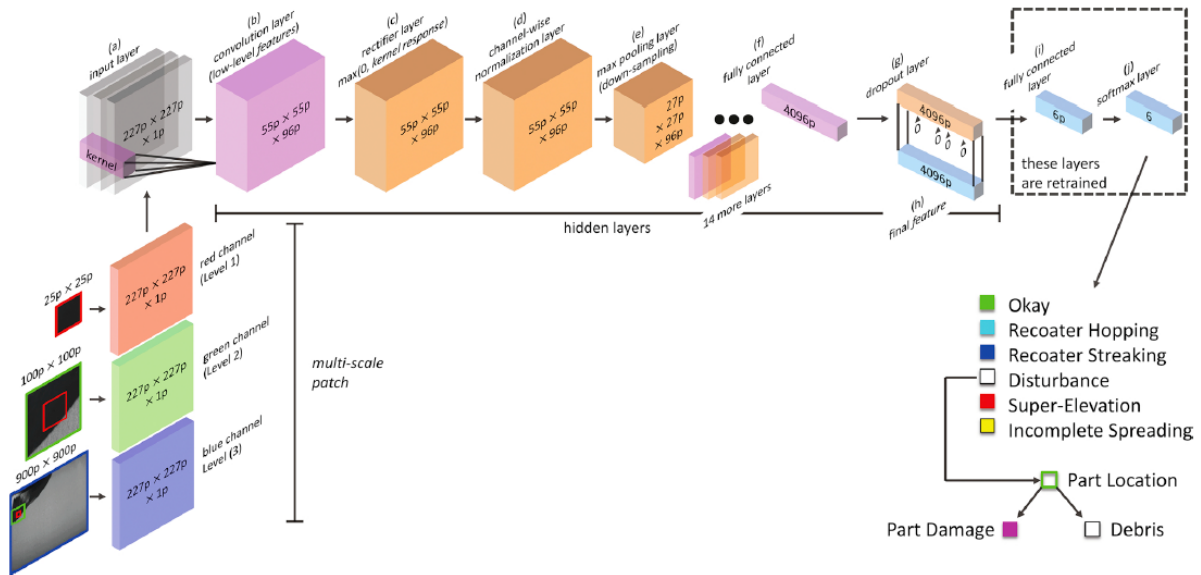

Figure 9. The flowchart of multi-scale CNN in defect detection in L-PBF process using multiple materials. Images labeled with okay or 5 types of defect are used to train multi-scale CNN. The information of images is propagated in the multi-scale CNN using convolution and finally used to classify the type of defect. Reprinted with permission from reference[42].

In general, the selection of different classification models should be based on the type of input features. While most of classifiers including DT, KNN, NN, SVM can deal with common parametric input problems in AM, SVM and CNN are recommended for images or AE based problems.

3.3. Overfitting issue and solutions

ML models learn knowledge from training data, and then use the knowledge to make predictions. Therefore, if the training data is used for testing the performance of ML models, the models tend to make perfect predictions in these training data, which seems good but may trap in the

overfitting issue. Figure 10 demonstrates an example of this situation in melt pool depth predictions using GP in L-PBF process. It reflects the fact that validation should never be done with the training dataset. Another example of this situation is the 100% F_1 score of NN in Ref. [45].
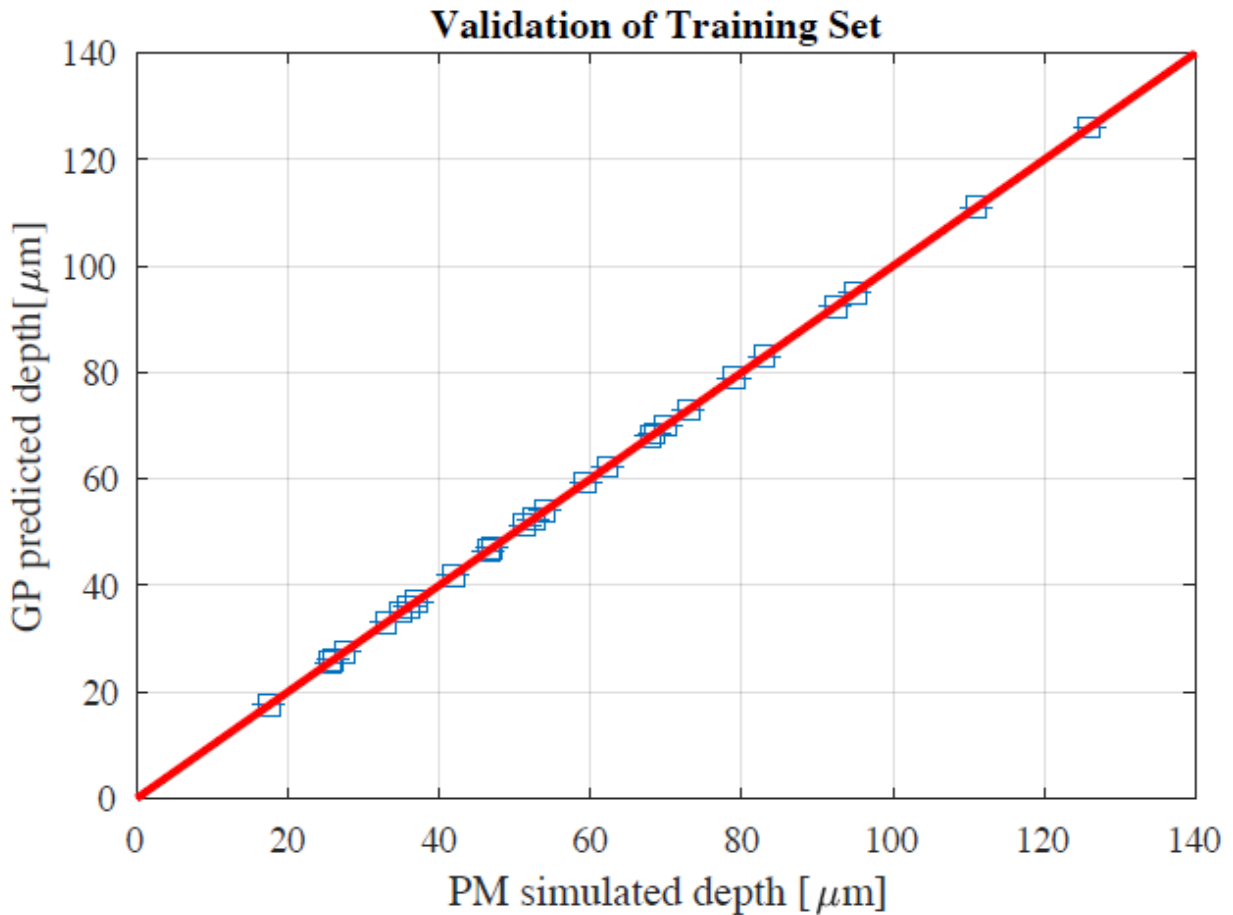

Figure 10. Validation plot of GP in prediction of melt pool depth in terms of laser power and speed of 316L strainless steel in L-PBF process. GP predictions are comparing against its training dataset. The closer from each point to the ideal $y = x$ line, the more accurate the prediction is. In this plot, all points are exactly lying on the ideal line without any uncertainty, indicating that the GP perfectly represents the data in the training set, and also the fact that validation should never be done with the training dataset. Reprinted with permission from reference[14].

Overfitting is a phenomenon that the model adjusts itself to fit the training dataset too exactly. In other words, with decreasing training error, the prediction error for future observations tends to increase. This is a common issue in supervised learning and should be avoided by some means. Three popular methods to help detect and avoid the overfitting issue in AM field are the hold-out method, the k-fold cross-validation method, and the regularization method.

The hold-out method, also known as data splitting, is a simple method to monitor overfitting. It partitions the whole dataset into two subsets, training set and testing set. The training set is used to train the model and the testing set is used to test the performance of the model. By this means, data in the testing set will not be used to train the model and is useful to test the performance of the model and whether overfitting occurs. An appropriate size of the training set is usually around 70% of the whole dataset. However, this method has a main drawback: it will further reduce the size of the training dataset when initially the number of data points is limited, which is the common situation in additive manufacturing as the cost, consumed time, and human labor to obtain each data point is usually high. This method is commonly applied in most of the applications mentioned above.

The k-fold cross-validation (CV) method is an iterative procedure which can monitor the overfitting issue and enhance the utilization of data. It partitions the whole dataset into k subsets of roughly same size. In each iteration, one subset is left out as the testing set and all other subsets are used to train the model. The iteration is repeated until all subsets have been left out once. A special case of this method is $k = n$, n being the number of data points, which is also called n-fold cross-validation method or leave-one-out cross-validation (LOOCV). Compared with the simple hold-out method, the CV method alleviates the common issue of the limited size of dataset in AM field.

The regularization method is a process which discourages the ML model to become too complex by adding information during training [57, 58]. In general, the goal of a ML model is to minimize the loss function:

$$E = \frac{1}{m}\sum_{k=1}^{m} E_k, \tag{1}$$

where E is the accumulative error, m is the number of training data points, and E_k is the error at each training data point. However, if noise exists in the training data, the ML model will also learn the noise using Eq. (1) and tend to overfit. To avoid this situation, the regularization method adds a term to the loss function, to penalize the complexity of the model. A commonly used regularization method is called L_2 regularization [59], which encourages the sum of the squares of the parameters to be small. For example, the loss function using L_2 regularization for neural network is:

$$E = \frac{1}{m}\sum_{k=1}^{m} E_k + (1-\lambda)\sum_i w_i^2, \tag{1}$$

where $\lambda \in (0,1)$ is the tuning parameter that determines how much penalty is added to the model complexity, and w_i is the weight of each neuron. The tuning parameter should be carefully selected and is usually estimated by cross-validation [57, 58].

Many algorithms, due to their learning mechanisms or strong learning ability, tend to overfit the training data. For instance, a characteristic of GP is that it will pass through all training points in regression tasks (Figure 10). For another instance, Hornik *et al.* [60] has shown that multilayer feedforward NN can approximate any function to any desired degree of accuracy, provided sufficiently many hidden units are available. Such strong learning ability of NN makes it likely overfit. To counter overfitting, many applications using above methods are reported in literature, such as the 10-fold [14] and *n*-fold cross-validation method for GP [16, 24, 61], and L2 regularization for NN [35]. Some algorithms also have their own methods against overfitting, such as the dropout method [62] for NN. Overall, the applications of one or more methods to monitor and avoid overfitting issue is necessary to make the ML model robust.

4. Unsupervised learning

In unsupervised learning, all data are not labeled with an output. The most common task in unsupervised learning is clustering analysis, in which the data are separated into groups based on their similarity. Another main type of unsupervised learning is principal component analysis (PCA), which converts a dataset with possibly correlated variables into a set of values of linearly uncorrelated variables called principal components by orthogonal transformation.

4.1. Clustering analysis in AM

In clustering analysis, all data are separated into groups based on their similarity. In general, a clustering analysis usually requires a large dataset size. However, the dataset size in AM field is usually limited, which impedes the application of clustering analysis. Therefore, only a few applications of clustering analysis in AM are reported recently.

In the cost estimation framework proposed by Chan *et al.*, the ML models applied are the least absolute selection and shrinkage operator (LASSO) and elastic net (EN) models. Another typical ML model for clustering analysis is the self-organizing map (SOM), which is a type of NN for unsupervised learning. Recently, an application for geometric accuracy analysis using SOM by Khanzadeh *et al.* [63] is reported. Using SOM, millions of data of geometric deviation are separated into clusters, and the overall geometric accuracy of the part fabricated using each combination of processing parameters can then be assessed: the more clusters that appear, the more types of deviations in terms of direction and magnitude it has. It should be noted that the same dataset in Ref. [63] is also used in Ref. [45] for quality assessment using supervised learning. This reveals that one dataset can have multiple applications in machine learning.

Khanzadeh *et al.* [64] also applied their SOM model for defect detection of Ti-6Al-4V in DED process. Under the assumptions that (1) an abnormal melt pool has low correlation with others, and (2) the percentage of abnormal melt pool is much smaller compared with normal melt pools, the data representing the melt pool temperature distribution characteristics are clustered and therefore the cluster with low correlation to all others is considered anomaly and porosity tends to occur at the corresponding locations. Another recent application of clustering analysis for defect detection using SOM is conducted by Wu *et al.* [65]. Features are extracted from AE signals acquired by AE sensors and then imported into SOM for clustering, as demonstrated in Figure 11. As the AE signals in the abnormal (failure) cases tend to be different to the ones in normal cases, the cluster of abnormal cases can be extracted from the normal cluster, and thus the defect detection is achieved. These two applications for defect detection in AM are a main functionality of clustering analysis known as anomaly detection. Comparing with the supervised learning-based defect detection, this method possesses a significant advantage that it doesn't require human interaction to label data. Therefore, clustering analysis can be a strong alternative for defect detection.
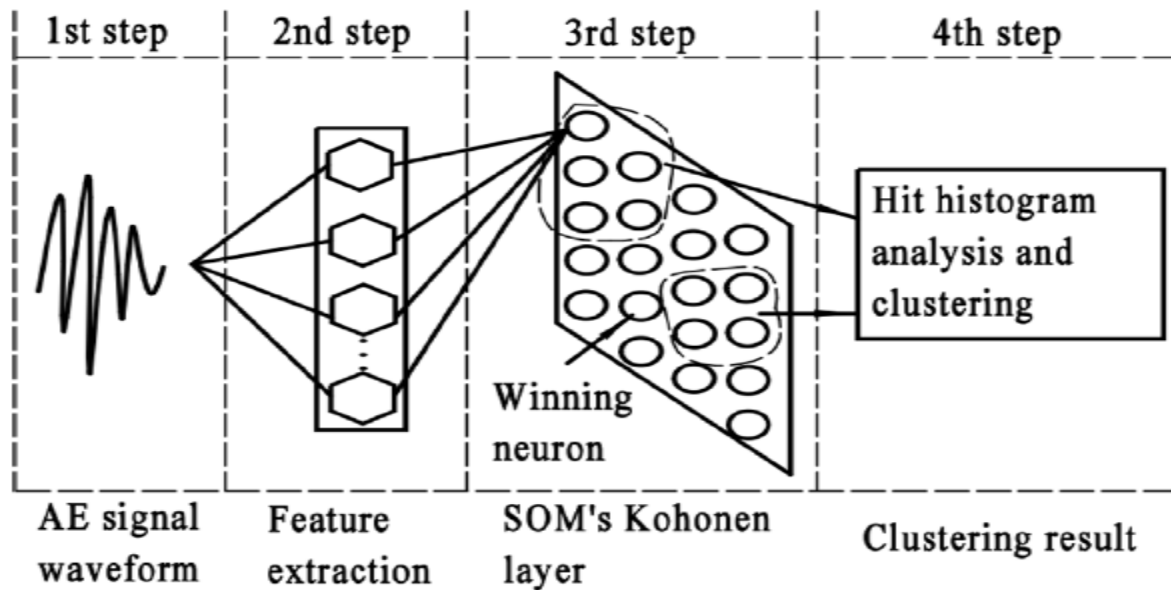
Figure 11. Procedure of a clustering analysis of AE signals of acrylonitrile butadiene styrene (ABS) in FFF process. Reprinted with permission from reference [65].

4.2. Principal component analysis in AM

Sometimes the number of features in a dataset is very large, especially when the input data type is image. In this case, to avoid the problem being too complicated, PCA is usually applied as a data pre-processing tool in AM to reduce the number of features so as to simplify the data. Khanzadeh *et al.* [44] applied PCA to simplify the features from melt pool characteristics and finally obtained nine principal components that account for almost 99.52% of variation in the data. Yang *et al.* [66] also demonstrates how to extract geometric features for energy consumption estimation in mask image projection stereolithography using PCA. In an application using *in situ* images as input data [40], the PCA increases the accuracy of SVM from 89.6% to 90.1% using 33 input features extracted from the image. However, in their 17-feature case, PCA is reported to weaken the performance of SVM, which indicate that PCA may also have negative effects on the performance of the coupled model, since too many features are eliminated and too much information is lost. Overall, when dealing with image-based problems, PCA is a great alternative to simplify the data.

5. Concluding remarks and future research directions

26

In this review article, the latest applications of ML in AM field are reviewed in terms of the type of learning tasks: supervised learning and unsupervised learning. For each specific types of tasks, including regression, classification, clustering and PCA, the corresponding applications and some popular algorithms are discussed, and the performance of some popular algorithms are assessed. The following is the recommended future research directions:

- While ML has been developing for several decades, the applications of ML in AM field have only been discovered for several years. These applications span processing parameters optimization, property prediction, defect detection, geometric deviation control, quality prediction and assessment, etc. Firstly, ML models can learn the relevance between the processing parameters and property using existing data, so as to provide guidance of optimizing these processing parameters. Secondly, ML models can predict the geometric deviation based on the designed geometry after training and provide guidance of geometric error compensation. Thirdly, ML models are good at dealing with *in situ* images and acoustic emission during printing and detecting defect formation in real time. However, the available data that can be extracted from the processing parameter-process-microstructure-property map have not been fully utilized. In this regard, exploiting more data acquisition methods, exploring more ML applications and developing better algorithms will be the main research directions in this infancy research field.

- A missing but useful functionality in supervised learning in recent literature is active learning. In AM field, labelling the output of each input data point is usually expensive in terms of the consumed time, cost and human labor, because it requires conducting an experiment or a simulation at each input setting to make this observation. Active learning is a method that can alleviate this issue. In recent literature, the common procedure in ML models is acquiring enough input-output pairs first and then using them to train ML models without further query of labelling new data. On the contrary, the procedure in active learning is that the ML models can make query interactively for labelling new data during training so as to maximize its performance. By this means, the ML models may use fewer data points to achieve better performance. Therefore, active learning is strongly recommended in the case that a dataset to be used to train the ML model has not been acquired.

- Another potential research field is the uncertainty quantification (UQ), which is critical for a robust design. The uncertainty in AM field has been reviewed in Ref. [15]. In regression tasks, ML models like GP provide not only the mean value at a certain input as the prediction of its output, but also standard deviation which represents the uncertainty at that point. Also, in classification tasks, ML models will also provide confidence when they make a classification. These uncertainties are part of the epistemic uncertainty and have not been utilized in recent literature. In addition, a typical UQ procedure [67] may require hundreds of data points, which is impractical to obtain from experiments or simulations. In this regard, a ML-based surrogate model is very helpful in obtaining the required data and increasing the efficiency of the UQ procedure. Overall, UQ in ML applications in AM field is a good research direction that has not been investigated in depth.

References

[1] ASTM F42. Resource document. https://www.astm.org/COMMIT/SUBCOMMIT/F42.htm. Accessed 28 February 2020

[2] I. Gibson, D. W. Rosen, and B. Stucker: *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, 2nd ed. (New York, NY: Springer, 2015), pp. 1-18.

[3] E. Alpaydin, *Introduction to machine learning*, 3rd ed. (London, England: The MIT Press, 2009), pp. 3.

[4] W. E. Frazier, *J. Mater. Eng. Perform.,vol*. 23, 1917 (2014).

[5] W. J. Sames, F. List, S. Pannala, R. R. Dehoff, and S. S. Babu, *Int. Mater. Rev.,vol*. 61, 315 (2016).

[6] Y. Zhang, L. Wu, X. Guo, S. Kane, Y. Deng, Y.-G. Jung, J.-H. Lee., and J. Zhang, *J. Mater. Eng. Perform.,vol*. 27, 1 (2018).

[7] M. M. Francois, A. Sun, W. E. King, N. J. Herson, D. Tourret, C. A. Bronkhorst, N. N. Carlson, C. K. Newman, T. S. Haut, J. Bakosi, J. W. Gibbs, V. Livescu, W. Vander, A. Scott, A. J. Clarke, M. W. Schraad, T. Blacker, H. Lim, T. Rodgers, S. Owen, F. Abdeljawad, J. Madison, A. T. Anderson, J. -L. Fattebert, R. M. Ferencz, N. E. Hodge, S. A. Khairallah, and O. Walton, "Modeling of additive manufacturing processes for metals: Challenges and opportunities," *Curr. Opin. Solid State Mater. Sci.,vol*. 21, 198 (2017).

[8] M. Markl and C. Körner, *Annu. Rev. Mater. Res.,vol*. 46, 93 (2016).

[9] S. K. Everton, M. Hirsch, P. Stravroulakis, R. K. Leach, and A. T. Clare, *Mater. Des.,vol*. 95, 431 (2016).

[10] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, *Bus. Inf. Syst. Eng.,vol*. 6, 239 (2014).

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. (London, England: The MIT Press, 2018), pp. 1-3.

[12] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, *Eng.,*vol. *5, 721 (*2019).

[13] National Academies of Sciences, Engineering, and Medicine. *Data-Driven Modeling for Additive Manufacturing of Metals: Proceedings of a Workshop*, 1st ed. (Washington, DC: The National Academies Press, 2019).

[14] G. Tapia, S. Khairallah, M. Matthews, W. E. King, and A. Elwany, *Int. J. Adv. Manuf. Tech.,vol*. 94, 3591 (2018).

[15] Z. Hu and S. Mahadevan, *The Int. J. Adv. Manuf. Tech.,vol*. 93, 2855 (2017).

[16] L. Meng and J. Zhang, *JOM-J. Min. Met. Mat. S.,vol*. 72, 420 (2020).

[17] T. Wang, T.-H. Kwok, C. Zhou, and S. Vader, *J. Manuf. Syst.,vol*. 47, 83 (2018).

[18] M. Grasso and B. M. Colosimo, *Meas. Sci. Technol.,vol*. 28, 044005 (2017).

[19] M. Mahesh, Y. Wong, J. Fuh, and H. Loh, *Rapid Prototyping J.,vol*. 10, 123 (2004).

[20] J. Francis and L. Bian, *Manuf. Lett.,vol*. 20, 10 (2019).

[21] S. L. Chan, Y. Lu, and Y. Wang, *J. Manuf. Syst.,vol*. 46, 115 (2018).

[22] Z. Zhu, N. Anwer, Q. Huang, and L. Mathieu, *CIRP Ann.,vol*. 67, 157 (2018).

[23] G. Tapia, A. Elwany, and H. Sang, *Addit. Manuf.,vol*. 12, 282 (2016).

[24] C. Kamath, *The Int. J. Adv. Manuf. Tech.,vol*. 86, 1659 (2016).

[25] F. Caiazzo and A. Caggiano, *Mat.,vol*. 11, 444 (2018).

[26] W. Rong-Ji, L. Xin-hua, W. Qing-ding, and W. Lingling, *The Int. J. Adv. Manuf. Tech.,vol*. 42, 1035 (2009).

[27] J. Zhang, P. Wang, and R. X. Gao, *Comput. Ind.,vol*. 107, 11 (2019).

[28] Z. Li, Z. Zhang, J. Shi, and D. Wu, *Robot. Cim-Int. Manuf.,vol*. 57, 488 (2019).

[29] M. Mozaffar, A. Paul, R. Al-Bahrani, S. Wolff, A. Choudhary, A. Agrawal, K. Ehmanna, J. Cao, *Manuf. Lett.,vol*. 18, 35 (2018).

[30] L. Song, W. Huang, X. Han, and J. Mazumder, *IEEE T. Ind. Electron.,vol*. 64, 633 (2016).

[31] S. Chowdhury and A. Sarn, ASME Int. Manuf. Sci. Eng. Conf., Proc. (2016) doi.org/10.1115/MSEC2016-8784.

[32] T. Kohonen, *Neural Networks,vol*. 1, 3 (1988).

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Cogn. Model.,vol*. 5, 1 (1988).

[34] A. R. Barron, *Nato. Adv. Sci. I. C-Mat.,*vol. 335, 561 (1991).

[35] F. Girosi, M. Jones, and T. Poggio, *Neural Comput.,vol*. 7, 219 (1995).

[36] S. Haykin, *Neural networks: a comprehensive foundation*, 1st ed. (Upper Saddle River, NJ: Prentice Hall PTR, 1994).

[37] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, 2nd ed. (London, England: MIT Press, 2006).

[38] R. Rai, J. Elmer, T. Palmer, and T. DebRoy, *J. Phys. D: Appl. Phys.,vol*. 40, 5753 (2007).

[39] X. Yao, S. K. Moon, and G. Bi, *Rapid Prototyping J.,vol*. 23, 983 (2017).

[40] Y. Zhang, G. S. Hong, D. Ye, K. Zhu, and J. Y. Fuh, *Mater. Des.,vol*. 156, 458 (2018).

[41] L. Scime and J. Beuth, *Addit. Manuf.,vol*. 19, 114 (2018).

[42] L. Scime and J. Beuth, *Addit. Manuf.,vol*. 24, 273 (2018).

[43] J. Mazumder, *Proc. CIRP,vol*. 36, 187 (2015).

[44] M. Khanzadeh, S. Chowdhury, M. Marufuzzaman, M. A. Tschopp, and L. Bian, *J. Manuf. Syst.,vol*. 47, 69 (2018).

[45] M. S. Tootooni, A. Dsouza, R. Donovan, P. K. Rao, Z. J. Kong, and P. Borgesen, *J. Eng. Ind.,vol*. 139, 091005 (2017).

[46] K. Aoyagi, H. Wang, H. Sudo, and A. Chiba, *Addit. Manuf.,vol*. 27, 353 (2019).

[47] D. Ye, G. S. Hong, Y. Zhang, K. Zhu, and J. Y. H. Fuh, *The Int. J. Adv. Manuf. Tech.,*vol. *96,* 1 (2018).

[48] Z. Shen, X. Shang, M. Zhao, X. Dong, G. Xiong, and F.-Y. Wang, "A Learning-Based Framework for Error Compensation in 3-D Printing," *IEEE T. Cybernetics,*vol. 49, 4042 (2019).

[49] R. Jafari-Marandi, M. Khanzadeh, W. Tian, B. Smith, and L. Bian, *J. Manuf. Syst.,vol*. 51, 29 (2019).

[50] C. Gobert, E. W. Reutzel, J. Petrich, A. R. Nassar, and S. Phoha, *Addit. Manuf.,vol*. 21, 517 (2018).

[51] A. Caggiano, J. Zhang, V. Alfieri, F. Caiazzo, R. Gao, and R. Teti, *CIRP Ann.,*vol. *68, 451 (*2019).

[52] S. A. Shevchik, C. Kenel, C. Leinenbach, and K. Wasmer, *Addit. Manuf.,vol*. 21, 598 (2018).

[53] J. R. Quinlan, *Mach. Learn.,vol*. 1, 81 (1986).

[54] C. Cortes and V. Vapnik, *Mach. Learn.,vol*. 20, 273 (1995).

[55] C.-W. Hsu and C.-J. Lin, *IEEE T. Neural Networ.,vol*. 13, 415 (2002).

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Adv. Neur. In.*, 1097 (2012).

[57] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*, 1st ed. (Berlin, Germany: Springer Science+Business Media, 2011).

[58] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, 1st ed. (New York, NY: Springer Science+Business Media, 2013)

[59] A. Y. Ng, *Proceedings of the twenty-first international conference on Machine learning*, 78 (2004).

[60] K. Hornik, M. Stinchcombe, and H. White, *Neural Networks,vol.* 2, 359 (1989).

[61] E. Popova, T. M. Rodgers, X. Gong, A. Cecen, J. D. Madison, and S. R. Kalidindi, *Integrating Materials and Manufacturing Innovation,vol.* 6, 54 (2017).

[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *J. Mach. Learn. Res.,vol.* 15, 1929 (2014).

[63] M. Khanzadeh, P. Rao, R. Jafari-Marandi, B. K. Smith, M. A. Tschopp, and L. Bian, *J. Eng. Ind.,vol.* 140, 031011 (2018).

[64] M. Khanzadeh, S. Chowdhury, M. A. Tschopp, H. R. Doude, M. Marufuzzaman, and L. Bian, *IISE Transactions,vol.* 51, 437-455 (2019).

[65] H. Wu, Z. Yu, and Y. Wang, *Measurement,vol.* 136, 445 (2019).

[66] Y. Yang, M. He, and L. Li, *Proc. CIRP,vol.* 80, 741 (2019).

[67] Z. Wang, P. Liu, Y. Xiao, X. Cui, Z. Hu, and L. Chen, *J. Eng. Ind.,vol.* 141, 081004 (2019).