MACHINE LEARNING IN ADDITIVE MANUFACTURING: A REVIEW OF LEARNING TECHNIQUES AND TASKS

J. A. Pike*†, J. Klett†, V. Kunc†, and C. E. Duty*†

*University of Tennessee, Knoxville, TN 37923
†Oak Ridge National Laboratory, Oak Ridge TN 37923

Abstract

Due to recent advances, Machine Learning (ML) has gained attention in the Additive Manufacturing (AM) community as a new way to improve parts and processes. The capability of ML to produce insights from large amounts of data by solving tasks such as classification, regression, and clustering provide possibilities to impact every step of the AM process. In the design phase, ML can optimize part design with respect to geometry, material selection, and part count. Prior to printing, process simulations can offer understanding into the how the part will be printed, and energy, time, and cost estimates of a print can be made to assist with resource planning. During printing, AM can benefit from in-situ printing optimization and quality monitoring. Lastly, ML can characterize printed parts from in-situ or ex-situ data. This article describes some of the ML learning techniques and tasks commonly employed in AM and provides examples of their use in previous works.

Introduction

Additive Manufacturing (AM) is a technology that produces parts by joining together layers of material as directed by a computer model. AM holds several advantages over traditional manufacturing methods due to its lack of tooling requirements and ability to create complex geometries that would otherwise be impossible to produce. These characteristics allow AM to accelerate progression from model to manufacture, which complements rapid prototyping and small to medium volume production. As AM has matured since its inception in the 1980's, the limitations of the technology have become better understood [1]–[3] Parts produced via AM may suffer from inaccurate geometry caused by issues such as delamination, warping, cracking, process anomalies, improper processing parameters, and many more depending on the specific AM process [4], [5]. The material properties of AM parts may be undesirable due to these reasons as well as processing limitations, material constraints, and characteristics specific to AM parts such as high porosity and anisotropic properties. Recently, there have been many attempts to address or better understand issues inherent to the AM process through Machine Learning (ML) [6]–[8]. ML is not only being used to investigate AM issues, but it can also be used in every phase of the AM lifecycle to improve the design, manufacture, and characterization of AM parts.

ML is an area of Artificial Intelligence (AI) that accomplishes complex tasks through learning from data. In recent years, technological advances and accomplishments have accelerated the development of ML. In 2013, it was discovered that ML algorithms could classify images at a level comparable to human intelligence [9] . Discoveries like this have compounded with continual

technological advances in computer technology to assist in ML development. Due to ML's reliance on large datasets to learn, the higher quantity of data produced within the last several years has been used to improve ML algorithm accuracy [10], [11]. Figure 1 shows that within the last ten years, the amount of data worldwide has increased almost 15-fold [12]. Additionally, Graphics Processing Units and higher power processors have continued to develop, which has allowed larger algorithms to be trained in shorter time periods [11]. The impacts of ML across a variety of applications have attracted the field of AM as it attempts to address some of its complex challenges.

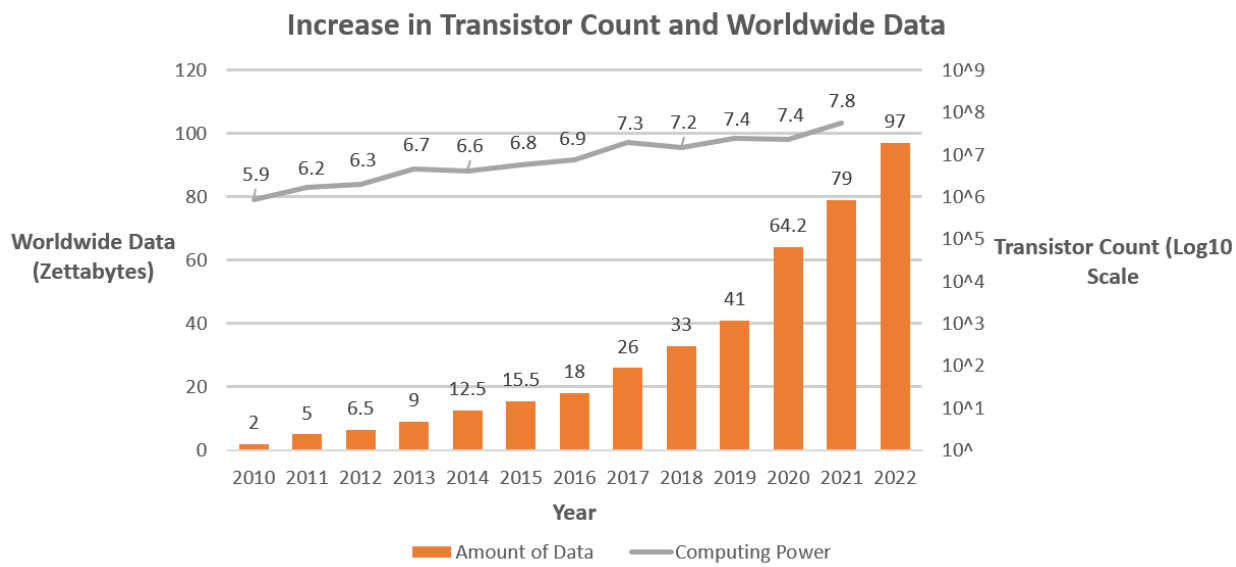

Figure 1: The increase in data and computing power within the past 12 years. Adapted from [12], [13].

ML applications in AM often target a set of difficult challenges that researchers hope to address through advanced computational techniques. The complicated interactions present within the AM process can cause poor repeatability and inconsistent parts. Even for an individual system, the underlying physics behind the AM process can be difficult to model due to the complexity of relationships between dominant variables and constant changes in material, geometry, and processing parameters. To understand some of these complex interactions, AM practitioners have turned to ML to examine the highly nonlinear relationships between the process, structure, and properties present in AM that are difficult to model using physics-based approaches.

Other areas of AM have not used ML to help solve existing challenges but to innovate in entirely new ways. New manufacturing capabilities inherent to AM present an opportunity for ML to be used in the design phase. Design for Additive Manufacturing (DfAM) has a much different set of guidelines compared to the typical Design for Manufacturing and Assembly (DfMA) [14]. AM can create geometries that are not possible with traditional manufacturing processes, which frequently results in more complex geometries or the consolidation of an assembly into a single AM part. For example, GE developed a fuel nozzle that combined 20 parts into one [15], [16]. ML has been used in several other aspects of design fostered by the complexity of parts possible through AM.

ML has been applied across the entirety of the AM lifecycle. Reviews have distinguished the phases of AM in different ways, but this paper will use the most general lifecycle structure to ensure agreement across all types of AM processing [7], [8], [17], [18]. Accordingly, the lifecycle has been broken into Design, Pre-processing, Printing, and Post-processing. A graphic displaying the different ways ML has been used in each of these phases is shown in Figure 2.

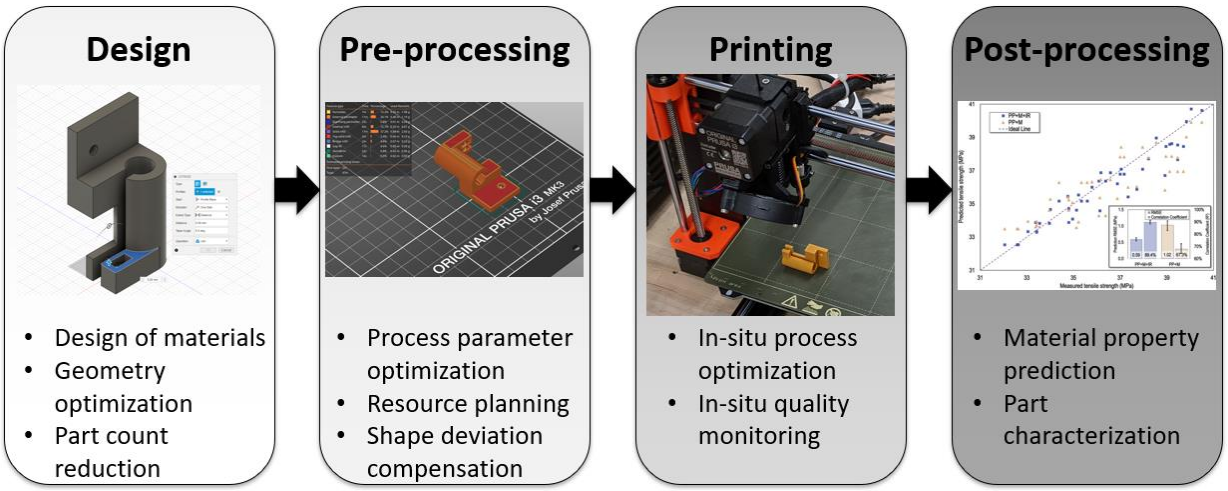

Figure 2: The 4 different phases in the AM lifecycle and the associated ways ML has been utilized in each.

The design phase of ML consists of selecting the part geometry and material for printing. In the design phase, ML has been used for geometry optimization in the form of part consolidation suggestion, automated shape deviation modeling, design of new materials, topology optimization and more [19]–[21]. The pre-processing phase involves choosing adequate printing parameters for the printing phase. ML has been leveraged in the pre-processing phase to construct process models, optimize process parameters, and plan resources [22]–[24]. The design and pre-processing phase are often combined into an interlinked iterative process whereby the results of simulations in the pre-processing phase provides feedback to the design phase for improved design choices. ML has been utilized arguably most successfully in the printing phase where in-situ data is used to monitor the quality of AM parts and optimize the printing process [25], [26]. The post-processing phase can vary widely depending on the AM process and final part requirements. Some types of AM may not require post-processing whereas others may involve downstream processes to finalize the part geometry and material properties or require extensive characterization to ensure requirement satisfaction. In the post-processing phase, the in-situ data is often used to assist in part characterization and help predict material properties of the part [27], [28].

Machine Learning Overview

To provide a clearer understanding of how and why ML is used in AM, it should be defined and distinguished from commonly used synonyms. Figure 3 provides a visual representation of the terminology discussed in this section. ML is a subset of AI that produces insights by learning from data. AI is a field of computer science that leverages computers to mimic the problem-solving and

decision-making capabilities of humans [29]. Simply put, any technology that mimics human intelligence can be thought of as AI. Within the field of AI, ML is only a sub-field defined by an ability to learn how to make decisions by analyzing data. Examples of AI that do not rely on ML are deterministic systems. In this context, deterministic means that the decisions that the AI algorithm makes are pre-programmed by a human, as opposed to self-taught from data. A good example for illustrative purposes is an Automated Teller Machine (ATM), which mimics human intelligence by carrying out a transaction that has been performed by a human in the past. If someone inserts an improper card into an ATM, and is shown an error message, the decision to reject the card will not vary as a function of the data the ATM receives. The decision-making process is hard coded into the software, and the decisions it makes are not due to learning from data.

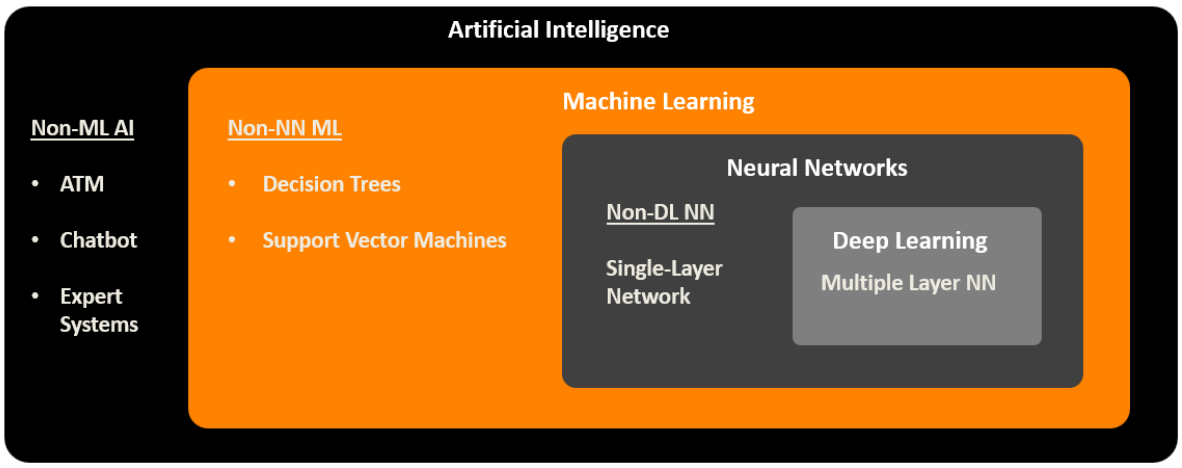

Figure 3: Venn diagram of terms related to Machine Learning.

ML is a sub-field of AI that mimics human intelligence specifically by learning through data. Across the field of ML there is a wide variety of learning techniques. In the next section, these techniques will be discussed further. Within each category of ML learning techniques, there are algorithms that rely specifically on Neural Network (NN) architectures, which are independent of the learning technique in use. Since they require extensive training, ML algorithms built on NN architectures have especially benefitted from the increases in data availability and computing power. However, the decisions made by NN's are not readily understood. Creating more explainable ML models is a separate, emerging area of ML research since difficulty in understanding decisions made by ML algorithms is a common occurrence. Given the increased frequency of this behavior with NN architectures, it is typically attempted last and considered a black-box method, especially considering ML algorithms exist which are more readily interpretable. An example of a ML algorithm that does not rely on a NN architecture is Decision Trees. A Decision Tree classifier is a type of ML algorithm that builds a model in the form of a tree structure. An example of how a Decision Tree model classifies data into 4 classes is shown in Figure 4. On the left side of the figure, a plot of the 2D input space is shown with colors corresponding to different classes. On the right side a tree-diagram is shown that explains how datapoints are categorized based on thresholds.

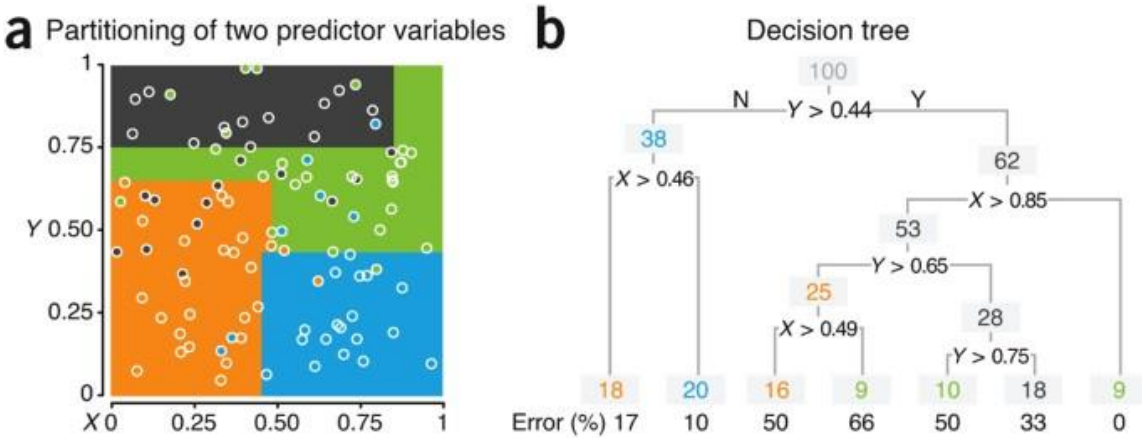

Figure 4: Graphic of classification of data by a Decision Tree. Adapted from [30].

Interpretable ML algorithms like Decision Trees provide significant utility; however, some datasets contain variables with complex, interdependent relationships that can be difficult to model using more interpretable algorithms. In these cases, ML practitioners sometimes turn to algorithms that utilize NN architectures. However, the benefit of more accurate models comes with the cost of less-interpretable solutions. NN's can be thought of simply as nonlinear function approximators. In Figure 5, the inputs to a NN are shown in orange circles on the left. These inputs are fed through a series of nonlinear functions shown by the blue circles in the middle, where each nonlinear function is referred to as a neuron. After computing the results of the non-linear functions, the outputs are given as shown in the red square on the right in the figure.

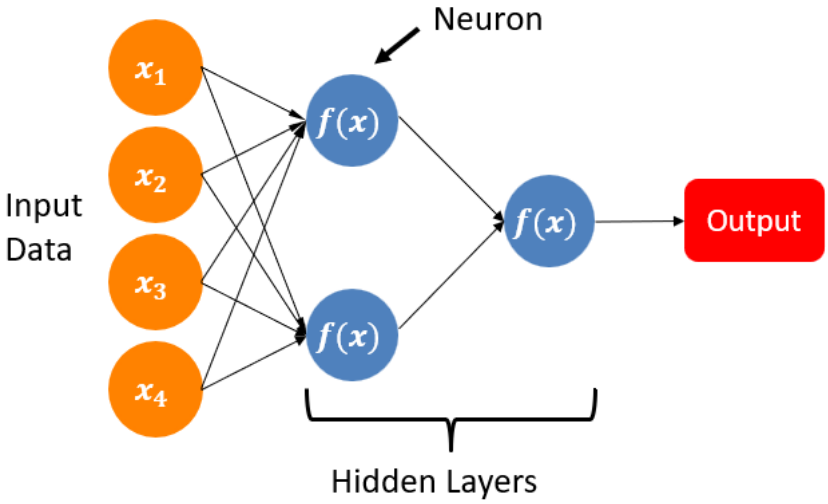

Figure 5: Diagram of a simple Neural Network.

NN architectures can be further categorized by the number of "layers" of neurons contained within a NN. In the real-world this is a rather trivial distinguishment since the vast majority of NN architectures in use consist of multiple layers. According to the "Universal Approximation Theorem," a single-layer perceptron is theoretically capable of solving any problem [31].

However, in practice, reducing the number of neurons in a layer and increasing the number of layers is usually more efficient [8]. A visual description of a single-layer perceptron (SLP) vs. a multi-layer perceptron (MLP) is shown in Figure 6. Increasing the number of layers to more than one results in creating a "deeper" network. NN architectures which use more than one layer are said to use Deep Learning (DL). The only example of a NN architecture that does not use DL is shown on the right in Figure 7 and is referred to as a SLP. Understanding the differences between AI, ML, NN architectures, and DL is very important for analyzing any paper that includes these topics. In some instances, studies will conflate these terms, so a firm understanding of their definitions and relationships to each other is important for successful analysis of papers in the field of AM that use ML.

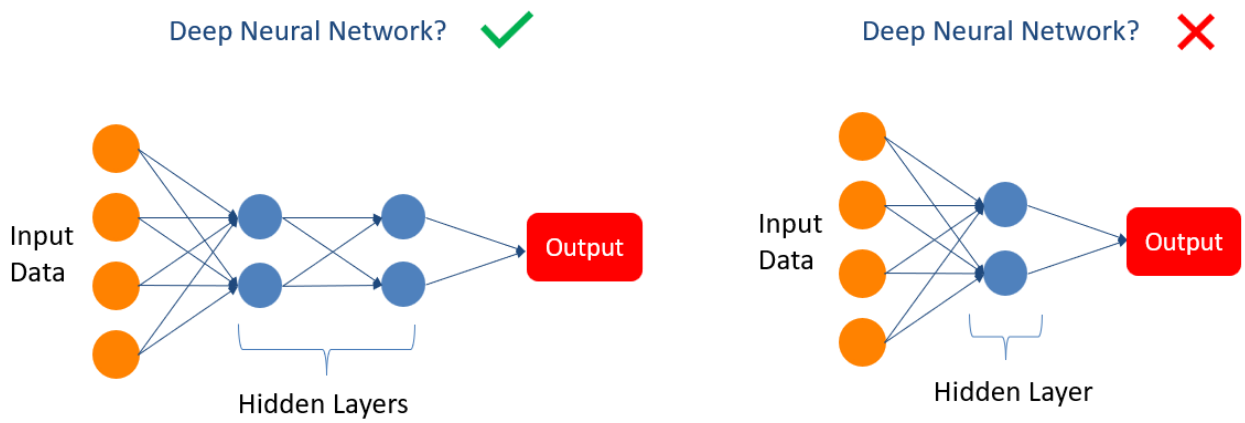

Figure 6: Comparison of NN architectures that rely on DL, MLP on the left, or do not, SLP on the right.

Machine Learning Techniques and Tasks

ML algorithms can be better understood if they are grouped by the type of learning technique used and the type of task that the algorithm aims to accomplish. Categorization of ML based on the learning techniques is a somewhat subjective task. Generally, ML can be broken down into supervised learning, unsupervised learning, and reinforcement learning. However, there are some learning techniques known as "hybrid" learning that use a combination of supervised learning and unsupervised learning that could also be included as a category. Therefore, learning techniques are oftentimes organized along a supervision spectrum extending from supervised to unsupervised learning as shown in Figure 7: Supervision spectrum displaying the relationships between categories of learning techniques within ML Since reinforcement learning and hybrid learning techniques are less commonly used in AM, this review will focus mostly on describing supervised and unsupervised learning. This study divides ML by learning techniques into 4 categories: supervised learning, unsupervised learning, hybrid learning, and reinforcement learning.

The difference between supervised and unsupervised learning techniques has to do with the training data. Supervised learning techniques have the data "labels," or model outputs, available in the training data to assist during the model training phase. Unsupervised learning techniques do not have data "labels" (outputs) available during training, so they rely on finding patterns in the input dataset. A useful example to distinguish the two is a scenario where two data

scientists, A and B, attempt to determine if a member of a community has diabetes given their health history. Data scientist A has a training dataset of 200 individuals' health history and the labels available (diabetes, no diabetes). Data scientist B has the same dataset without labels. Data scientist A will use a supervised ML algorithm, which is likely to classify people more accurately based on their health history. Data scientist B will have to use an unsupervised ML algorithm, which will likely perform worse. The decision to use supervised learning vs. unsupervised learning has to do with problem context. In the diabetes example, it is possible that an unsupervised or hybrid learning technique would be needed if labels were difficult to determine because diabetes tests were very costly. Some datasets contain labels by default, while others may not be capable of obtaining accurate labels. A common problem is that labeling datasets can be time-consuming for a human to do, which encourages use of an unsupervised learning technique.

Types of learning between the two extremes of supervised learning and unsupervised learning are often referred to as hybrid learning techniques. Hybrid learning techniques use a combination of supervised and unsupervised learning. For example, semi-supervised learning techniques train a ML algorithm with some human-labeled training data, and then use unsupervised learning to learn from the rest of the unlabeled data. Reinforcement is largely dissimilar to both supervised and unsupervised learning. Although the training process itself is considerably different, even reinforcement learning has been categorized as a type of supervised learning. This technique is less commonly used in general and especially in the field of AM, but recent studies have achieved some success utilizing this technique [33]. The placement of supervised, unsupervised, semi-supervised, and reinforcement learning on the supervision spectrum is displayed in Figure 7.

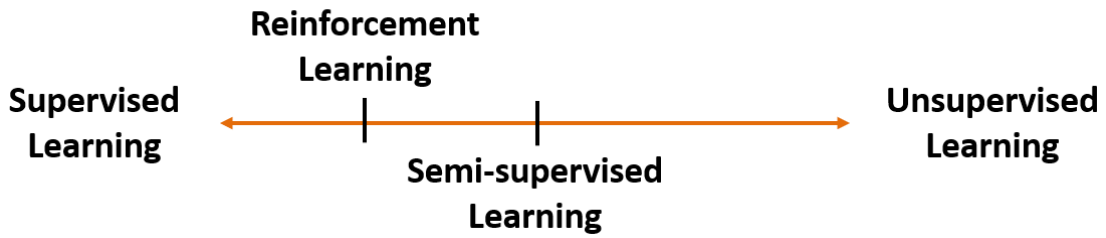

Figure 7: Supervision spectrum displaying the relationships between categories of learning techniques within ML

ML tasks refer to the problem that a ML algorithm attempts to solve (e.g., regression). As stated previously, a useful way to delineate between ML tasks is by both learning technique and output data type. ML algorithms commonly produce one of two output data types: continuous outputs or categorical outputs. Since, only supervised learning and unsupervised learning will be discussed in depth in this study, Figure 8 only breaks down categories for these two learning techniques.

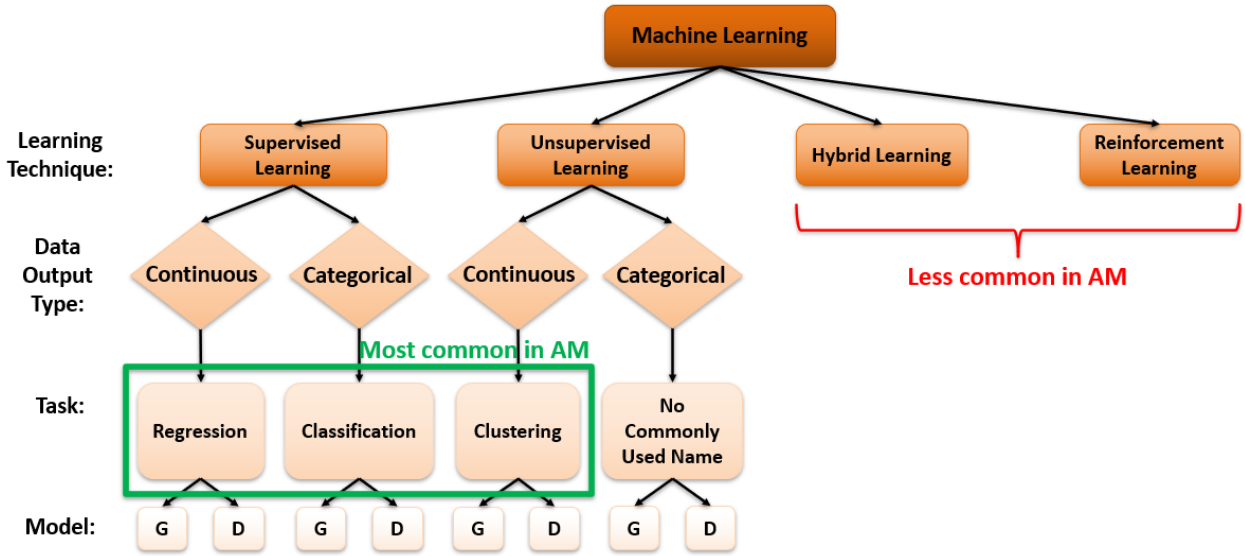

Figure 8: Categorization of ML by most common learning techniques.

For applications of ML in AM, the three most common ML tasks are supervised techniques that determine continuous or categorical outputs (classification or regression) and unsupervised techniques that determine categorical outputs (clustering). Supervised techniques to determine categorical outputs are referred to as classification. Supervised techniques that produce continuous outputs are called regression. Lastly, unsupervised techniques that produce categorical outputs are referred to as clustering. Unsupervised learning techniques that determine continuous outputs are also widely implemented using models such as Principal Component Analysis (PCA), but they are mostly used for data pre-processing instead of providing final outputs [34]. This topic will be covered briefly but will not be a point of emphasis. The next section includes a discussion on these ML tasks and provides specific examples of how they have been applied in AM.

Tasks describe the type of problem that a ML algorithm attempts to solve, but different models can be implemented for a given task. The underlying math of different ML models varies, but a clear distinction between models is the ability of a model to sample new datapoints. Generative models learn the distribution of data while discriminative models learn the boundaries in the data. This means a generative model can sample new datapoints, but a discriminative model cannot. The distinction between different types of models within a class are represented by "G" for generative and "D" for discriminative in Figure 8: Categorization of ML by most common learning techniques. This difference in types of models holds importance based on problem context. For example, if a dataset has many missing values, then employing a generative model might improve the accuracy of the data imputed in place of the missing value. The description of the 3 most common ML tasks used in AM in the next section will only delineate by task and will not further describe the most common types of models.

Common Applications of ML in AM

Supervised Learning:

In this subsection, regression and classification will be briefly described, and then works in the field of AM that accomplish these tasks will be discussed.

Continuous Output Tasks (Regression):

When the term regression is used, typically the speaker is referring specifically to supervised regression since unsupervised regression is not a commonly used term. Supervised regression can be very simple or complex depending on the ML algorithm used. For example, using a least-squares linear fit in Excel to approximate Young's Modulus is simple, but using a NN architecture to produce predictions of tensile strength of an AM part relies on a more complex method.

An example of a ML regression algorithm is Support Vector Regression (SVR) models [34]. The resulting function of a SVR model with default parameters approximated to fit 500 noisy datapoints from a sine function is shown in red in Figure 9. This problem consists of one independent and dependent variable, so it is easy to visualize. However, much more complex models with dimensions above three are more difficult or impossible to visualize. Fitting a model of the right waveform is much more difficult for high-dimensional datasets, so oftentimes ML models are used. In general, what distinguishes statistics models from ML models, is that ML models are built on less restrictive assumptions. This can cause issues such as overfitting, but it allows extremely complex relationships between variables to be modeled to produce higher accuracy results [34]. There is a hazy line between statistical models and ML models. In fact, it can be argued that there is no need to discern since ML models are formulated using statistics. However, the difference between the extremes of fitting a line in a spreadsheet vs. training a neural network has engendered the term ML, which has most often been used to describe models that rely on fewer assumptions and are applied to large datasets.

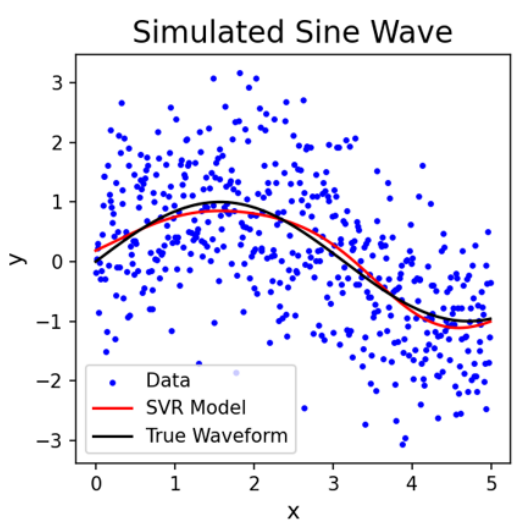

Figure 9: Example of how well a SVR model can fit a waveform using the default hyperparameters from Sci-kit Learn [35].

In Figure 10, a ML algorithm known as a Multi-Layer Perceptron (MLP) was used to predict the tensile strength of a part made from a Fused Filament Fabrication (FFF) system [36]. The left side of the figure shows the inputs to the MLP consist of process parameters such as material properties, heater temperatures, etc. and in-situ measurements such as melt temperature and vibrations. The block labeled "LSTM" in the input stands for Long Short-Term Memory (LSTM). LSTM is a NN-based architecture that was used in this paper to pre-process the in-situ measurements to improve results by correlating datapoints sampled in a certain timeframe [37]. The middle of the figure shows the MLP algorithm, which acts as the regression model. The output from the MLP, the strength prediction, is shown on the right side of the figure. The MLP was trained and tested using 144 ASTM samples that were printed using different process settings and materials. After the 144 samples were printed, tensile tests were conducted for each sample to measure each sample's yield stress. The yield stress for the 144 samples were the labels/output for the dataset. The MLP was trained using 100 of the samples associated data, and then 44 were used to test the accuracy of the MLP model's tensile strength prediction. The MLP attained an R^2 value of 90% correlation between predictions and measurements of tensile strengths based on the 44 test datapoints.

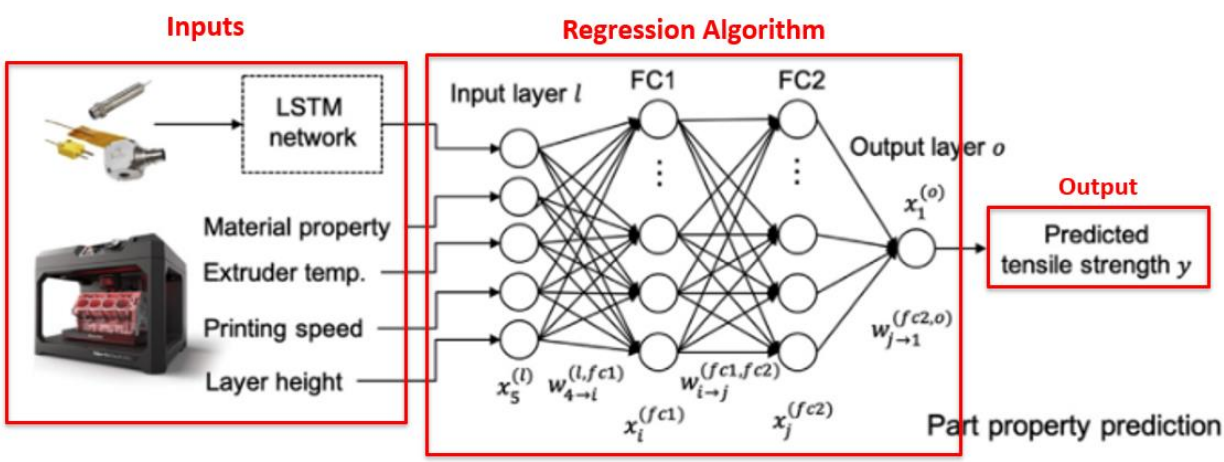

Figure 10: Diagram of regression algorithm, MLP. Adapted from [36].

Categorical Output Tasks (Classification):

Another type of supervised task is classification. Classification algorithms attempt to assign classes to input datapoints. For example, images could be classified into two different classes: dogs vs. cats. Figure 11 shows how two ML algorithms classify datapoints into categories A and B [38]. On the left side, an algorithm known as Quadratic Discriminant Analysis (QDA) is used, and on the right side the k-Nearest Neighbors (kNN) algorithm is used [34]. In Figure 11, the horizontal and vertical axes correspond to independent variables, and the colors indicate the class of the datapoint. The dashed and solid lines are decision boundaries which partition the discrete classes in the input space. This figure provides another example of how ML models built on fewer assumptions tend to produce higher accuracy results. Clearly, the decision boundary formed by kNN produces a decision boundary that adheres more closely to the true decision boundary than QDA. In the 2-dimensional space, choosing a correct decision boundary that

separates the classes is a simple task for a human operator. However, for high-dimensional inputs visualization is difficult. This difficulty generates the need for ML classification algorithms capable of learning complex decision boundaries in high-dimensional spaces that do not rely on strict assumptions. As stated in the regression sub-section, enforcing less restrictive assumptions improves accuracy, but this comes with the potential cost of overfitting. Figure 11 shows how the QDA classification produces a poor decision boundary since it is based on a fundamental assumption that the decision boundary has a quadratic shape. On the other hand, kNN relies on less strict assumptions that allow it to produce a more complex decision boundary which improves classification accuracy. For a visualization of how the ML decision boundaries evolve as more datapoints are used to train the ML algorithms, the authors refer you to a video from the original author or the presentation for this article for a side-by-side comparison [38], [39].

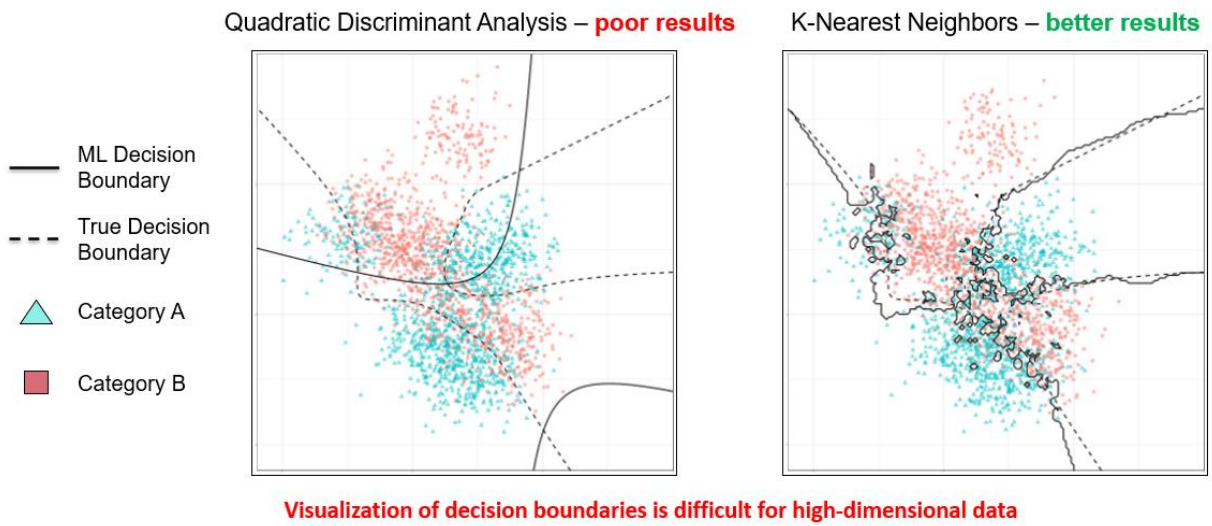

Figure 11: Classification of a dataset performed by QDA on the left and kNN on the right. Adapted from [38].

An example of classification performed in the context of AM is the classification of layers into low, medium, and high degrees of porosity [40]. The inputs to the model, shown in the top of Figure 12, are acoustic emissions from a Selective Laser Melting (SLM) printed layer measured with fiber-Bragg Grating sensors. After pre-processing, the acoustic signals were fed to a spectral Convolutional Neural Network (CNN), which is shown in the bottom of the figure. The spectral CNN used the acoustic emission from a layer as input and classified the signal as belonging to a layer with low, medium, or high porosity. The possible classes corresponding to the outputs of the model are shown in the left of the figure. Initially the classification of classes of porosity was agnostic of the percent porosity corresponding to each class; however, afterward the porosity of each class was characterized. This makes it a classification task even though after the model was trained continuous outputs were assigned to the 3 classes.

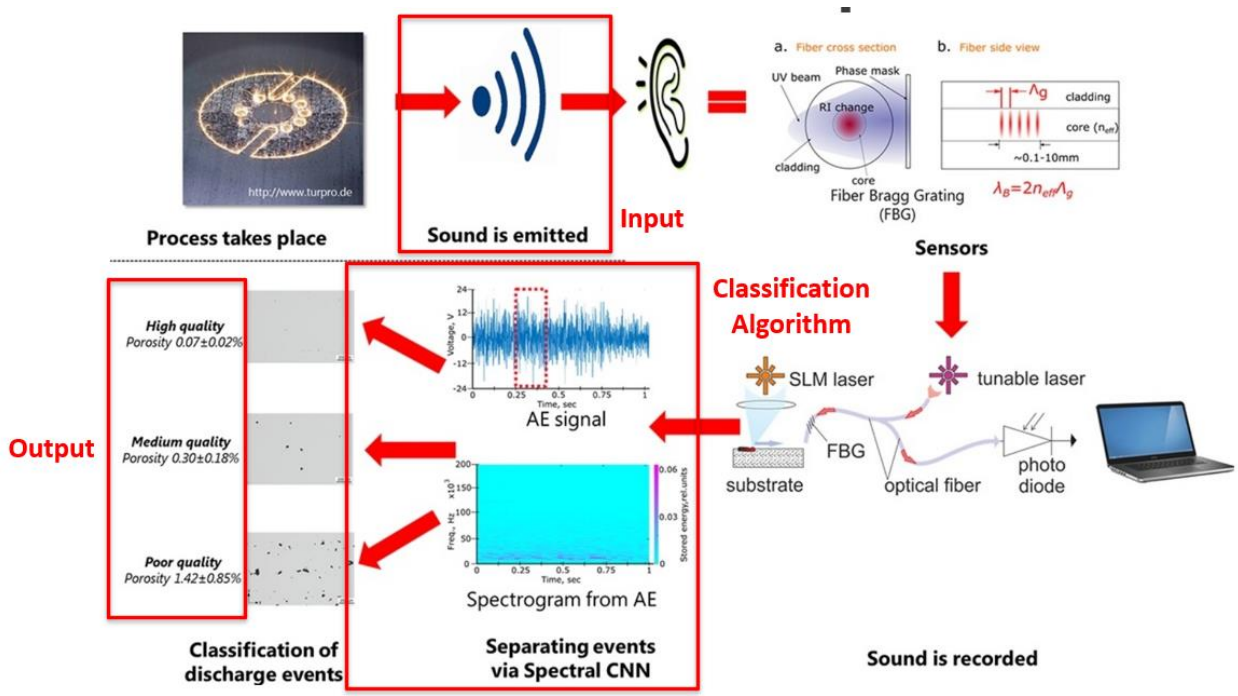

Figure 12: Graphic of a classification algorithm used for classifying percent porosity of SLM layers. Adapted from [40].

Unsupervised:

Like supervised learning, unsupervised learning can be broken down by the type of data output produced by the model. Unsupervised techniques that produce categorical outputs are commonly referred to as clustering, but unsupervised techniques that produce continuous outputs do not have a specific term. As mentioned previously, unsupervised learning does not have the outputs available during training. Therefore, unsupervised learning techniques rely on recognizing patterns in the input data. As in the supervised learning sub-section, this sub-section will discuss unsupervised continuous-output tasks then shift to unsupervised categorical-output tasks.

Continuous Output Tasks:

Unsupervised approaches that produce continuous outputs do not have a commonly used term in the field of ML. Methods stemming from this type of ML technique are dimensionality reduction, association methods, or correlation methods [34]. This type of ML task has less emphasis than regression, since the outputs are usually available when attempting to produce continuous outputs from a model, which in turn means an unsupervised learning technique is not needed. One of the most widely used ML algorithms in unsupervised continuous-output techniques is a dimensionality reduction method known as Principal Component Analysis (PCA) [34]. Dimensionality reduction attempts to compress the many features, or variables, of input data into fewer features that retain the nearly same amount of information as the input data. PCA and other dimensionality reduction methods are used for speeding up computation time when training models, improving model performance, assisting in visualizations, and reducing noise in a dataset. PCA use is wide-ranging across all fields of ML, due to it meeting demands for problems consistent with all ML models.

Categorical Output (Clustering) Tasks:

Unlike unsupervised approaches that produce continuous variables, supervised learning techniques that produce categorical variables have a commonly used term known as "clustering." Clustering involves partitioning datapoints in the input space into different classes, or "clusters." since the outputs are unknown. In some cases, the number of clusters is known ahead of time, or in other cases it is estimated.

To provide a useful depiction of how clustering works, the K-means clustering algorithm will be described. Figure 13 shows three steps during training of the K-means clustering algorithm. The color of each region indicates the input space that belongs to each cluster of datapoints, the boundaries of the colored regions correspond to decision boundaries, and an "X" corresponds to a cluster center. The plots depict the input space that contains 4 locations with a high density of datapoints referred to as "clusters." The leftmost plot depicts the random assignment of cluster centers at the beginning of training. The middle plot shows the K-means clustering algorithm updating the cluster center location based on the calculation of the centroid of the datapoints in each. The final plot on the right shows that the cluster centers have been correctly identified by the end of training. In this example, the number of clusters has either been determined or accurately guessed. However, the number of clusters may be unknown, which results in using an algorithm that estimates the number of cluster centers or in reduced clustering performance.

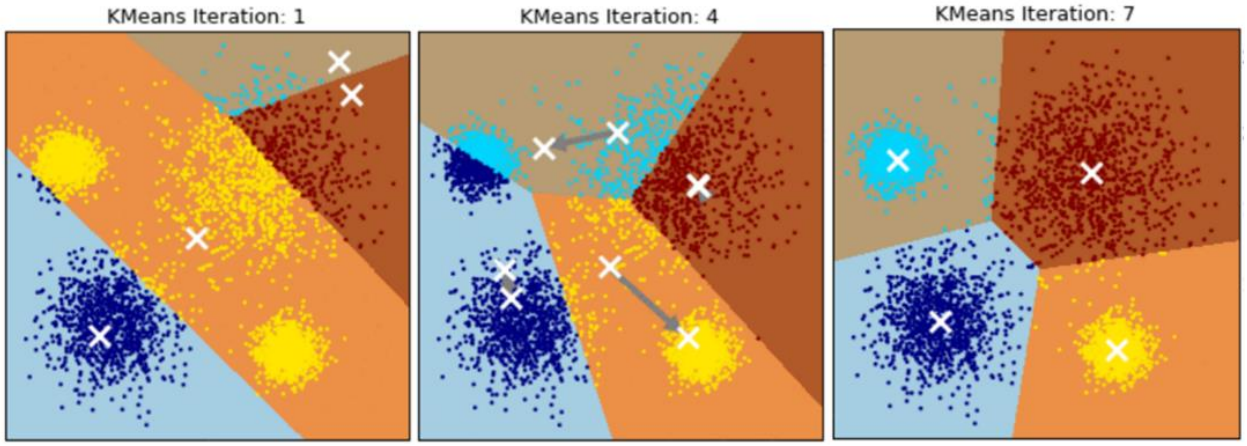

Figure 13: K-means clustering algorithm during training. Adapted from [41].

An example of clustering performed in AM is clustering of similar part geometries [42]. In this study, a multi-step process is used to estimate the cost of printing parts with an FFF system. The cost of printing a part relies on the amount of material used, energy costs, and time required to print. All of these can be estimated after a part has been printed using simple measurements techniques; however, estimating the cost before printing is difficult due to the uncertainty in forecasting the quantities of mass, energy, and time needed. In steps 1-3 in Figure 18, a clustering algorithm is used to group similar geometries. The input to the clustering algorithm was a computer-aided design (CAD) model, and the output was the cluster assignment (class of geometry). Using one regression algorithm to estimate the cost of a printed part from a CAD

geometry can be difficult since geometry drastically affects the price of the print. To circumvent this issue, the authors in [42] built multiple regression models for each one of the clusters, as shown in the figure below with the images labeled "similar jobs". Therefore, the cost prediction algorithm first categorized the geometry with clustering and then used the regression model that has been trained on similar geometries to estimate the cost of the print for a given CAD model. This improved the overall accuracy of the cost estimation, since each cost estimate used a regression algorithm trained on similar CAD geometries to estimate the cost.

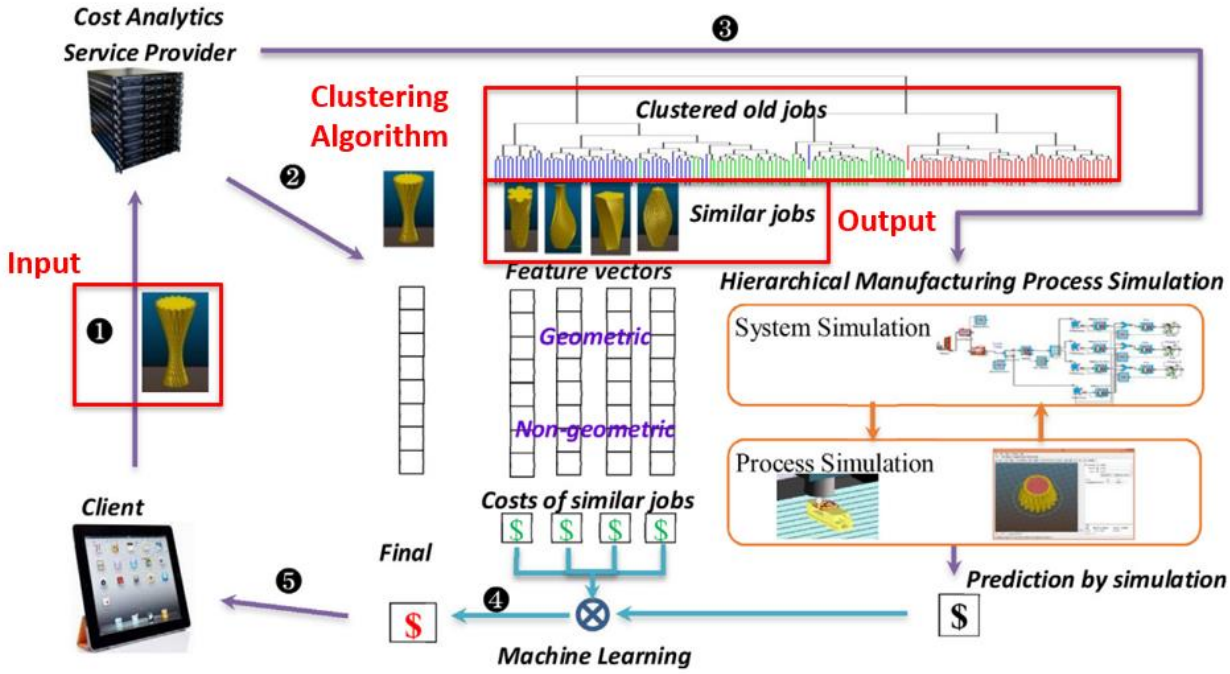

Figure 14: Diagram of the print cost prediction algorithm used in [42]. The red boxes call out where the clustering algorithm is used and the input and output to the clustering algorithm. Adapted from [42]

To summarize the descriptions of the three main ML tasks in AM, three review papers were surveyed to determine the distribution of these tasks versus others [7], [17], [18]. Figure 15 provides a pie chart for each phase of the AM lifecycle. Since some sources use multiple ML algorithms in their approaches, some papers have appeared in more than one of the four ML tasks specified. The "Other" category in blue refers to studies that use more complex or custom methods of applying ML that do not fit neatly into the three tasks described in this work. Examples of methods within this category are optimization algorithms or unsupervised learning algorithms that produce continuous outputs. The remaining three categories are regression, classification, and clustering and correspond to different shades of orange. In the four pie charts, the "Other" category consists of less than a quarter of the papers surveyed. This indicates that the three machine learning tasks discussed in this article are sufficient for understanding most papers in the field of AM that use ML.

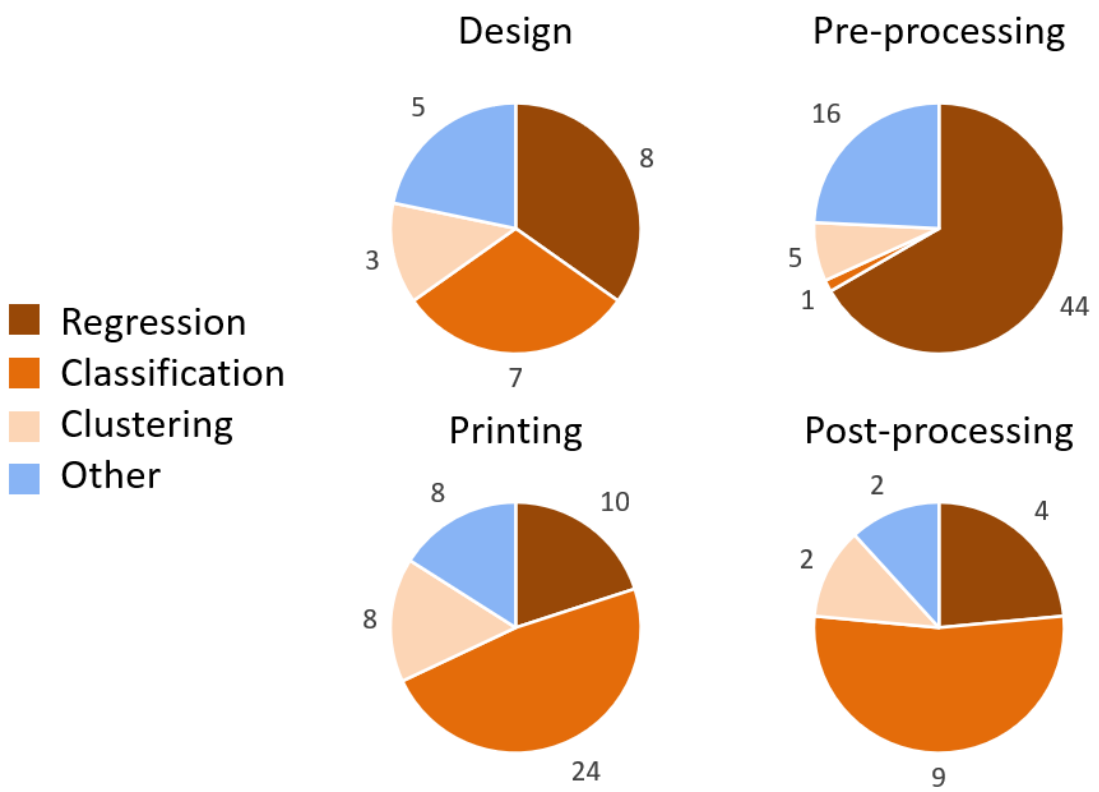

Figure 15: Distribution of the 3 ML tasks used in AM in comparison to "Other" lesser common tasks. The 156 sources analyzed to produce this figure are from 3 reviews of ML in AM [7], [17], [18]

Although previous reviews have found most papers that use ML techniques are categorized under the printing phase, this article chooses to group articles differently. Many papers use in-situ data to train ML models, but many of the models are not used during the printing process. For example, in-situ data in combination with ML is often used to optimize printing parameters in the pre-processing phase, but the ML model is not used in real-time [43]. Thus, the definitions used in this paper influences the quantity of papers in each AM phase. The imbalance in number of papers within each AM phase has been explored in other papers and is not a point of emphasis for this survey [7].

Conclusion

ML has been used to solve a wide variety of problems and optimize many phases in the AM lifecycle. As ML develops and becomes better understood, opportunities to optimize AM processes will continue to be explored. This study has covered the phases of AM where ML has been applied, and described the background, function, and instances of the three most solved ML tasks in the field of AM: classification, regression, and clustering. The current applications of ML in AM have been described, and examples of ML's use in AM have been incorporated using specific studies from the field. The claim of the importance of these three ML tasks has been demonstrated using a survey of literature that reviewed ML uses in AM.

Acknowledgements

This research was sponsored by the Bredesen Center for Interdisciplinary Research and Graduate Education. Research also sponsored by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Advanced Manufacturing Office, under contract DE-AC05-00OR22725 with UT-Battelle, LLC.

References

[1] W. Gao *et al.*, "The status, challenges, and future of additive manufacturing in engineering," *Computer-Aided Design*, vol. 69, pp. 65–89, Dec. 2015, doi: 10.1016/j.cad.2015.04.001.

[2] M. Attaran, "The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing," *Business Horizons*, vol. 60, no. 5, pp. 677–688, Sep. 2017, doi: 10.1016/j.bushor.2017.05.011.

[3] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Q. Nguyen, and D. Hui, "Additive manufacturing (3D printing): A review of materials, methods, applications and challenges," *Composites Part B: Engineering*, vol. 143, pp. 172–196, Jun. 2018, doi: 10.1016/j.compositesb.2018.02.012.

[4] B. N. Turner and S. A. Gold, "A review of melt extrusion additive manufacturing processes: II. Materials, dimensional accuracy, and surface roughness," *Rapid Prototyping Journal*, vol. 21, no. 3, pp. 250–261, Jan. 2015, doi: 10.1108/RPJ-02-2013-0017.

[5] F. H. Kim and S. P. Moylan, "Literature review of metal additive manufacturing defects," National Institute of Standards and Technology, Gaithersburg, MD, NIST AMS 100-16, May 2018. doi: 10.6028/NIST.AMS.100-16.

[6] Z. Jin, Z. Zhang, K. Demir, and G. X. Gu, "Machine Learning for Advanced Additive Manufacturing," *Matter*, vol. 3, no. 5, pp. 1541–1556, Nov. 2020, doi: 10.1016/j.matt.2020.08.023.

[7] J. Qin *et al.*, "Research and application of machine learning for additive manufacturing," *Additive Manufacturing*, vol. 52, p. 102691, Apr. 2022, doi: 10.1016/j.addma.2022.102691.

[8] C. Wang, X. P. Tan, S. B. Tor, and C. S. Lim, "Machine learning in additive manufacturing: State-of-the-art and perspectives," *Additive Manufacturing*, vol. 36, p. 101538, Dec. 2020, doi: 10.1016/j.addma.2020.101538.

[9] D. Gershgorn, "The data that transformed AI research—and possibly the world," *Quartz*, Jul. 26, 2017. https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/ (accessed Aug. 18, 2022).

[10] R. Alfred, "The rise of machine learning for big data analytics," Oct. 2016, pp. 1–1. doi: 10.1109/ICSITech.2016.7852593.

[11] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.

[12] U. Hack, "What's the real story behind the explosive growth of data?," *Redgate*, Sep. 08, 2021. https://www.red-gate.com/blog/database-development/whats-the-real-story-behind-the-explosive-growth-of-data (accessed Aug. 18, 2022).

[13] K. Rupp, "Microprocessor Trend Data." Aug. 23, 2022. Accessed: Aug. 25, 2022. [Online]. Available: https://github.com/karlrupp/microprocessor-trend-data

[14] J. Jiang, Y. Xiong, Z. Zhang, and D. W. Rosen, "Machine learning integrated design for additive manufacturing," *J Intell Manuf*, vol. 33, no. 4, pp. 1073–1086, Apr. 2022, doi: 10.1007/s10845-020-01715-6.
[15] E. D. Herderick, "Progress in Additive Manufacturing," *JOM*, vol. 67, no. 3, pp. 580–581, Mar. 2015, doi: 10.1007/s11837-015-1323-x.
[16] "2013 - GE Capital - Additive - Manufacturing - Fall - 2013 | PDF | 3 D Printing | General Electric," *Scribd*. https://www.scribd.com/document/235540644/2013-GE-Capital-Additive-Manufacturing-Fall-2013 (accessed Aug. 26, 2022).
[17] L. Meng *et al.*, "Machine Learning in Additive Manufacturing: A Review," *JOM*, vol. 72, no. 6, pp. 2363–2377, Jun. 2020, doi: 10.1007/s11837-020-04155-y.
[18] S. A. Razvi, S. C. Feng, A. N. Narayanan, Y.-T. Lee, and P. Witherell, "A Review Of Machine Learning Applications In Additive Manufacturing," *NIST*, Aug. 2019, Accessed: Jul. 11, 2022. [Online]. Available: https://www.nist.gov/publications/review-machine-learning-applications-additive-manufacturing
[19] R. de Souza Borges Ferreira, A. Sabbaghi, and Q. Huang, "Automated Geometric Shape Deviation Modeling for Additive Manufacturing Systems via Bayesian Neural Networks," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 584–598, Apr. 2020, doi: 10.1109/TASE.2019.2936821.
[20] X. Yao, S. K. Moon, and G. Bi, "A hybrid machine learning approach for additive manufacturing design feature recommendation," *Rapid Prototyping Journal*, vol. 23, no. 6, pp. 983–997, Jan. 2017, doi: 10.1108/RPJ-03-2016-0041.
[21] H. Ko, P. Witherell, Y. Lu, S. Kim, and D. W. Rosen, "Machine learning and knowledge graph based design rule construction for additive manufacturing," *Additive Manufacturing*, vol. 37, p. 101620, Jan. 2021, doi: 10.1016/j.addma.2020.101620.
[22] J. Qin, Y. Liu, R. Grosvenor, F. Lacan, and Z. Jiang, "Deep learning-driven particle swarm optimisation for additive manufacturing energy optimisation," *Journal of Cleaner Production*, vol. 245, p. 118702, Feb. 2020, doi: 10.1016/j.jclepro.2019.118702.
[23] Z. Gan *et al.*, "Data-Driven Microstructure and Microhardness Design in Additive Manufacturing Using a Self-Organizing Map," *Engineering*, vol. 5, no. 4, pp. 730–735, Aug. 2019, doi: 10.1016/j.eng.2019.03.014.
[24] W. Tian, J. Ma, and M. Alizadeh, "Energy consumption optimization with geometric accuracy consideration for fused filament fabrication processes," *Int J Adv Manuf Technol*, vol. 103, no. 5, pp. 3223–3233, Aug. 2019, doi: 10.1007/s00170-019-03683-5.
[25] P. Becker, C. Roth, A. Roennau, and R. Dillmann, "Acoustic Anomaly Detection in Additive Manufacturing with Long Short-Term Memory Neural Networks," in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, Apr. 2020, pp. 921–926. doi: 10.1109/ICIEA49774.2020.9102002.
[26] "(PDF) Real-Time Composition Monitoring Using Support Vector Regression of Laser-Induced Plasma for Laser Additive Manufacturing." https://www.researchgate.net/publication/309056067_Real-Time_Composition_Monitoring_Using_Support_Vector_Regression_of_Laser-Induced_Plasma_for_Laser_Additive_Manufacturing (accessed Aug. 26, 2022).
[27] M. Kusano *et al.*, "Tensile properties prediction by multiple linear regression analysis for selective laser melted and post heat-treated Ti-6Al-4V with microstructural quantification," *Materials Science and Engineering: A*, vol. 787, p. 139549, Jun. 2020, doi: 10.1016/j.msea.2020.139549.

[28] Y. Zhang, G. S. Hong, D. Ye, K. Zhu, and J. Y. H. Fuh, "Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring," *Materials & Design*, vol. 156, pp. 458–469, Oct. 2018, doi: 10.1016/j.matdes.2018.07.002.
[29] "What is Artificial Intelligence (AI)? | IBM." https://www.ibm.com/cloud/learn/what-is-artificial-intelligence (accessed Aug. 18, 2022).
[30] M. Krzywinski and N. Altman, "Classification and regression trees," *Nature Methods*, vol. 14, no. 8, Art. no. 8, Aug. 2017, doi: 10.1038/nmeth.4370.
[31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989, doi: 10.1016/0893-6080(89)90020-8.
[32] A. Geron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]." https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (accessed Aug. 18, 2022).
[33] "Using artificial intelligence to control digital manufacturing," *MIT News | Massachusetts Institute of Technology*. https://news.mit.edu/2022/artificial-intelligence-3-d-printing-0802 (accessed Aug. 18, 2022).
[34] F. Sohil, M. U. Sohali, and J. Shabbir, "An introduction to statistical learning with applications in R: by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, $41.98, eISBN: 978-1-4614-7137-7," *Statistical Theory and Related Fields*, vol. 6, no. 1, pp. 87–87, Jan. 2022, doi: 10.1080/24754269.2021.1980261.
[35] "1.4. Support Vector Machines," *scikit-learn*. https://scikit-learn/stable/modules/svm.html (accessed Aug. 18, 2022).
[36] J. Zhang, P. Wang, and R. X. Gao, "Deep learning-based tensile strength prediction in fused deposition modeling," *Computers in Industry*, vol. 107, pp. 11–21, May 2019, doi: 10.1016/j.compind.2019.01.011.
[37] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997, doi: 10.1162/neco.1997.9.8.1735.
[38] R. Holbrook, "Learning Decision Boundaries." Aug. 16, 2022. Accessed: Aug. 18, 2022. [Online]. Available: https://github.com/ryanholbrook/decision-boundaries-animations
[39] J. Pike, "Research | Jay Pike," *Jpike*. https://jaypike12.wixsite.com/jpike/research (accessed Aug. 26, 2022).
[40] S. A. Shevchik, C. Kenel, C. Leinenbach, and K. Wasmer, "Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks," *Additive Manufacturing*, vol. 21, pp. 598–604, May 2018, doi: 10.1016/j.addma.2017.11.012.
[41] D. Sheehan, "Clustering with Scikit with GIFs," *dashee87.github.io*, May 09, 2017. https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/ (accessed Aug. 18, 2022).
[42] S. Chan, Y. Lu, and Y. Wang, "Data-driven cost estimation for additive manufacturing in cybermanufacturing," *Journal of Manufacturing Systems*, vol. 46, pp. 115–126, Jan. 2018, doi: 10.1016/j.jmsy.2017.12.001.
[43] S. Mondal, D. Gwynn, A. Ray, and A. Basak, "Investigation of Melt Pool Geometry Control in Additive Manufacturing Using Hybrid Modeling," *Metals*, vol. 10, no. 5, Art. no. 5, May 2020, doi: 10.3390/met10050683.