12-2021

# A Machine Learning Method for the Prediction of Melt Pool Geometries Created by Laser Powder Bed Fusion

Jonathan Ciaccio
jciaccio@uno.edu

# A Machine Learning Method for the Prediction of Melt-Pool Geometries Created by Laser Powder-Bed Fusion

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Masters of Science
in
Engineering-Mechanical

by

Jonathan Ciaccio

B.S. Louisiana State University, 2016

December, 2021

## Acknowledgement

I would first like to acknowledge Dr. Uttam Chakravarty, my graduate supervisor, for encouraging and motivating me to write a thesis. I have vastly increased my knowledge of machine learning while also greatly improving my researching and report writing skills which will be invaluable moving forward into my professional career. His help and feedback were very important in writing this thesis.

I would also like to acknowledge Dr. M Shafiqur Rahman and express my deepest appreciation for his help in this paper. Throughout this past year, he has been a vital part in helping me understand how to write a proper report and as well as understanding how to do proper research. He has always been there to answer questions and provide feedback, and for that I can not thank him enough.

I would like to thank my other committee members, Dr. Martin Guillot and Dr. David Hui, for their time.

Finally, I would like to thank Md Wasi Ul Kabir with his help in machine learning, and also to Dr. Shengmin Guo and his students at LSU who conducted and shared the results of the experiments for the model validation.

# Table of Contents

# List of Figures

# List of Tables

# Abstract

A machine learning model is created to predict melt-pool geometries of Ti-6Al-4V alloy created by the laser powder bed fusion process. Data is collected through an extensive literature survey, using results from both experiments and CFD modeling. The model focuses on five key input parameters that influence melt-pool geometries: laser power, scanning speed, spot size, powder density, and powder layer thickness. The two outputs of the model are melt-pool width and melt-pool depth. The model is trained and tested by using the k fold cross-validation technique. Multiple regression models are then applied to find the model that produces the least amount of error. Verification of the ML model was achieved by comparing the model results with experimental results and CFD results given the same parameter values throughout the models. The ML model results are consistent with the experimental and CFD results.



**Keywords:** laser powder-bed fusion, machine learning, Ti-6Al-4V, regression models

# Chapter 1

# Introduction

Over the last decade, powder-bed fusion (PBF) additive manufacturing (AM) has become a wildly popular technique for creating metallic or alloy parts, especially for aerospace, automotive, biomedical, dentistry, and electronics applications [1]. In general, AM techniques save time, reduce costs, and require minimal proficiency of an operator [1]. However, the process can be defective and costly if the processing parameters are not set and optimized properly. Researchers usually perform trial and error methods using experiments and physics-based numerical modeling to determine the correct combination of the processing parameters, which takes substantial effort, time, and money [1]. This is where machine learning (ML) comes into play, by giving the scope for modeling and predicting desired outputs in a short time utilizing a very large data set of certain input parameters. When that data set is obtained and provided as an input to the model, the computer can learn and produce its results. The digital nature of the PBF process allows ML to identify and resolve the issues in manufacturing conveniently. One of the most important PBF processes is the laser PBF (L-PBF) process, where the melt-pool geometry is a significant output, showing the width and depth of penetration of the laser beam and the heat-affected zone within the workpiece. This output depends on several factors, including the material behavior, environment, and laser parameters. Among these factors, finding the optimized combination of the laser and material parameters, namely the laser power, scanning speed, spot size, layer thickness, and porosity, is extremely crucial when performing the L-PBF process effectively [2].

## 1.1 Additive Manufacturing

Additive Manufacturing, also known as 3-D printing, is a process where material parts are built in layers [3]. The design of the part is first created in a computer aided design (CAD) model and then input into a 3-D printing machine [3]. Additive manufacturing reduces material waste, saves time, and cost less compared to subtractive manufacturing, the conventional manufacturing process. With subtractive manufacturing, the process starts with a bulk piece of material, and the designed part is essentially sculpted out of that piece. Not only is significant waste produced from subtractive manufacturing but also the manufacturer must pay for the entirety of the bulk piece and cut around it. In additive manufacturing, the cost of the piece is only the material used to create it. Subtractive manufacturing requires an expert operator to the mill the desired part while additive manufacturing only requires a given CAD model and the means to input the data into the 3D printing machine.

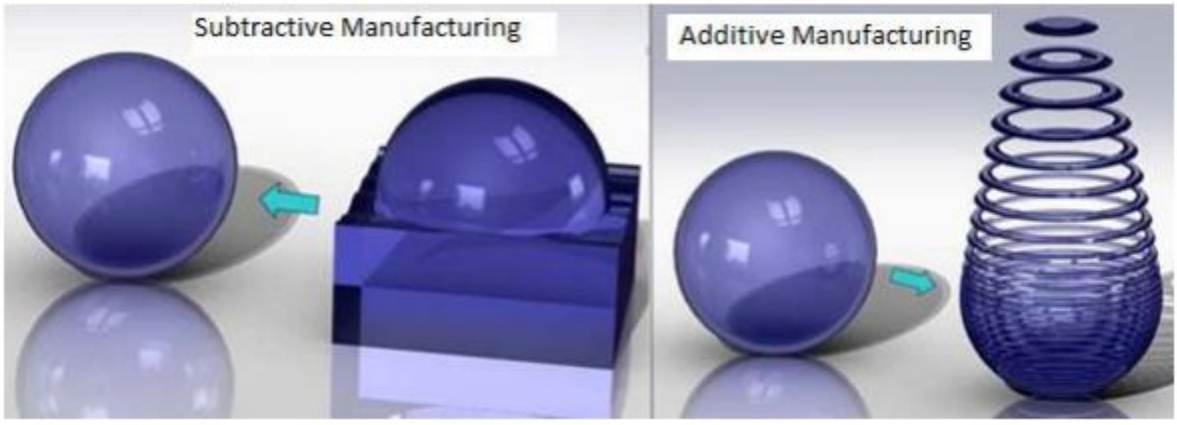


Figure 1.1: Subtractive vs Additive Manufacturing [4]

## 2.1 Laser Powder Bed Fusion

The laser powder bed-fusion process is a type of additive manufacturing that is quickly gaining popularity. During this process, a rake spreads powder from the powder reserve across a build plate [3]. The powder is melted by a laser that selectively scans across the plate based upon the given CAD model [3]. As the first layer is complete, the build platform will retreat downwards, allowing another layer of powder to be deposited by the rake [3]. This process is repeated until the desired part is created.

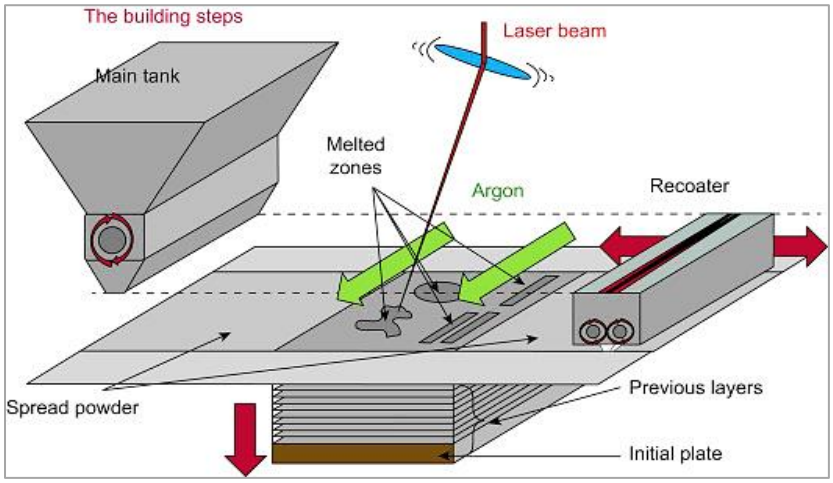


Figure 1.2: Schematic diagram of the L-PBF Process [1]

One of the most important outputs in the laser powder-bed fusion process is the geometry of the melt-pool [2]. The melt-pool geometry is indicative of the depth and width of the laser output; geometry can also show defects in the fusion process such as porosity or geometrical deviation [5]. Both of those defects along with insufficient laser penetration and spread will have a negative effect on the structural integrity of the final product. Therefore, it is vital to produce a sufficient melt-pool during the laser powder-bed fusion process in order to manufacture an

acceptable and desired material part. The melt-pool geometry is dependent on several laser and material properties, including: laser power, scanning speed, spot size, layer thickness, and powder density [2]. Optimization of these parameters will reduce cost, minimize part deficiencies, decrease surface roughness, reduce build time, and create a dense material with the correct material characteristics [6].

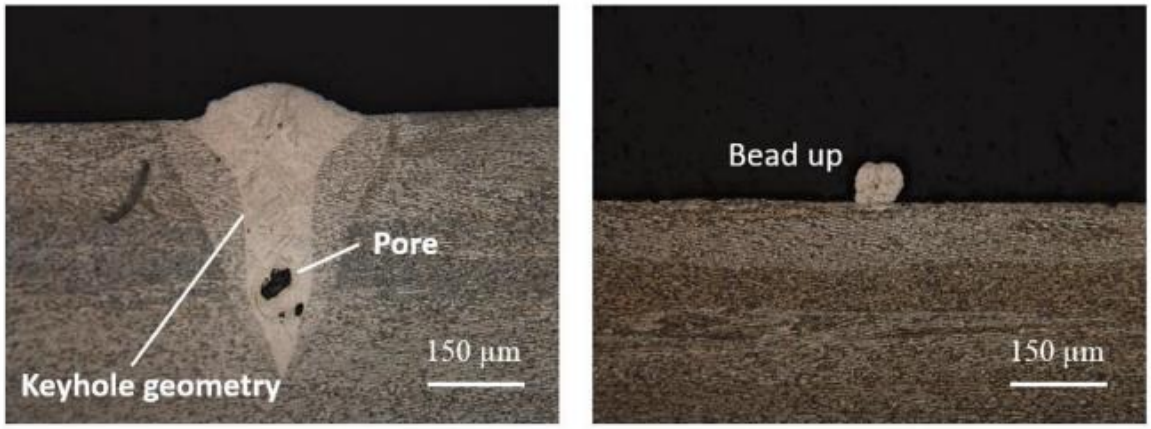


Figure 1.3: Defective Melt-Pool Geometries due to Non-optimized Process Parameters [7]

Each of the five key parameters plays a different role in creating an effective melt-pool geometry [2]. Therefore, the focus of this study is subjected to these five parameters. Optimizing the scanning speed is crucial; if the manufacturing process can be completed as fast as possible, the manufacturer will save time. However, if the scanning speed is too fast, the laser will not be able to penetrate deep enough into the powder-bed. Therefore, the manufacturer cannot operate the laser at maximum speed, but instead must find the optimum speed at which an efficient melt-pool is created. The second parameter that must be considered is power. The stronger the laser is, the more costly that laser becomes. The power of the laser must be sufficient to melt the powder layer but using a laser that has more than enough power would only be more costly. The third parameter is spot size of the laser, also known as the diameter. A smaller diameter means a more concentrated laser (more energy will be transferred deeper into the substrate); with a larger

diameter the laser will not penetrate as deep because the laser is less focused, and more energy spreads out over the surface. Layer thickness is the fourth feature to optimize. If the layer is too thin, the laser will penetrate the substrate too deeply, which wastes power and hurts the final product's structural integrity. If the layer is too thick, the laser will not penetrate the layer fully which will result in incomplete melting of the powder layer. The final parameter is the porosity of the material which is also referred to as the packing density. The porosity considered in this study is the porosity of the powder before it is melted. A more porous medium means the material is more powder than solid. Having a higher powder ratio is often preferred because the laser will be able to melt the powder more easily than if it were only solid. The data set compiled consists of these five parameters that will be the input for the ML program. The depth and width of the melt-pool geometry, collected from the same data set, will serve as the output [2].

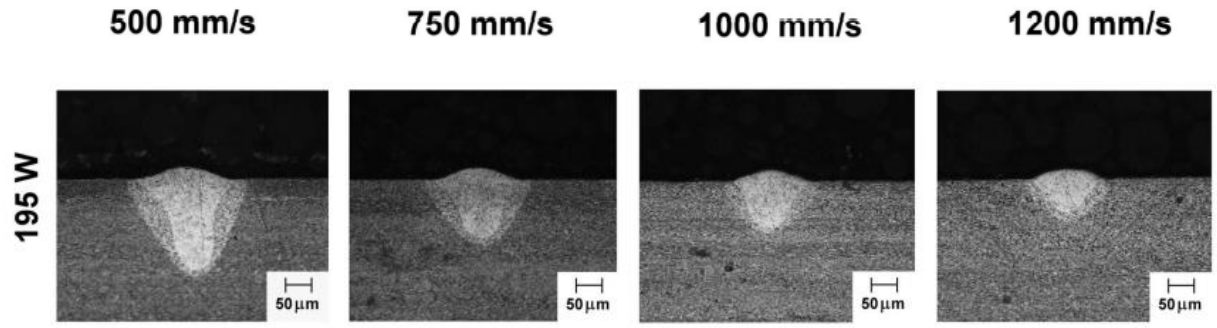


Figure 1.4: Different Melt-Pool Geometries with Laser Power of 195 W and Varying Scanning Speeds [8]

## 1.2 Machine Learning

Machine learning (ML) is a series of algorithms that are able to predict or classify data based upon previous training of similar data. Machine learning can be classified into three categories: supervised learning, unsupervised learning, and reinforcement [9]. Supervised

learning occurs when the data is labelled with specific inputs and outputs. Within the supervised learning category, two subcategories exist: regression and classification. Unsupervised learning contains unlabeled data, and the algorithm must find the patterns within the data [9]. The two most popular types of unsupervised learning are clustering and association [9]. "Reinforcement learning (RL) algorithms provide a mathematical framework for sequential decision making by autonomous agents [10]."

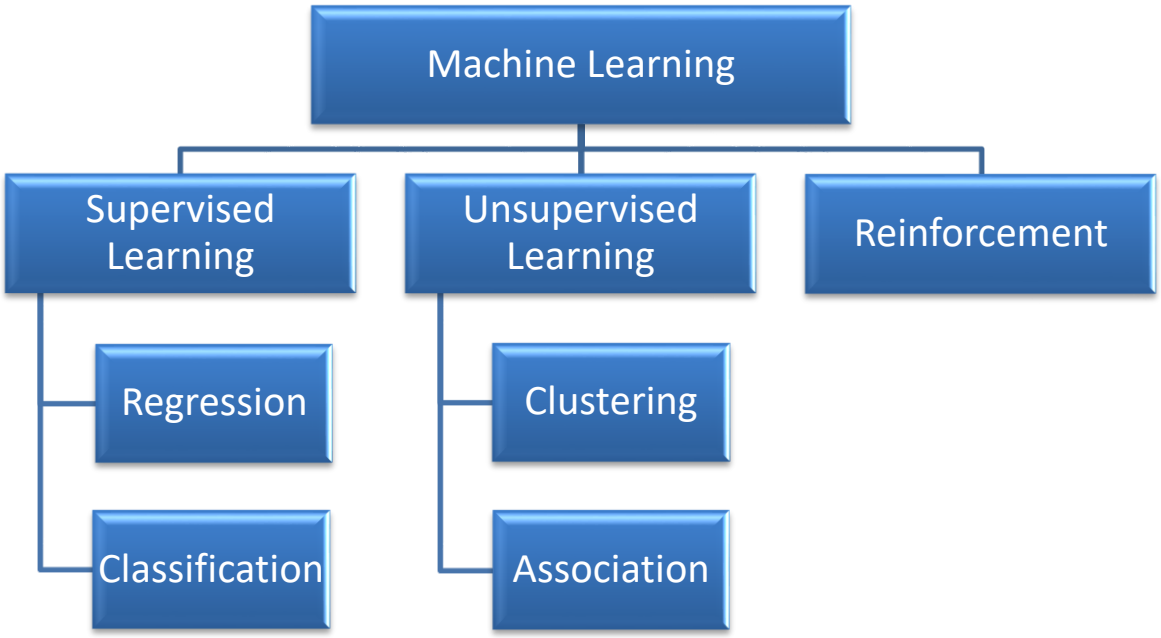


Figure 1.5: Machine Learning Family Tree

### 1.2.1 Supervised Learning

Regression analysis is a predictive modelling technique for investigating the relationship between a dependent variable (i.e., target) and independent variable(s) (i.e., predictor) [11]. Classification analysis is when the output is binary, such as a 1 or 0. The classification algorithm will be able to map the inputs into two categories. Regression can indicate the strength of impact

of multiple independent variables on a dependent variable. As a simple type of regression, linear regression relates an independent variable with a dependent variable [12]. The variables must have a linear relationship. Independent variables and dependent variables are plotted on a scatter plot [12] and a best fit line is applied to the data [12]. The best fit line produces an equation and allows one to predict the dependent variable given the independent variable [12]. The residual distance is the distance between the actual value and the predicted value. A simple linear regression model follows the equation:

$$y = \beta_o + \beta_1 x \tag{1}$$

In Eq. (1), y is the dependent variable, x is the independent variable, $B_o$ is the y-intercept, and $B_1$ is the slope of the regression line.

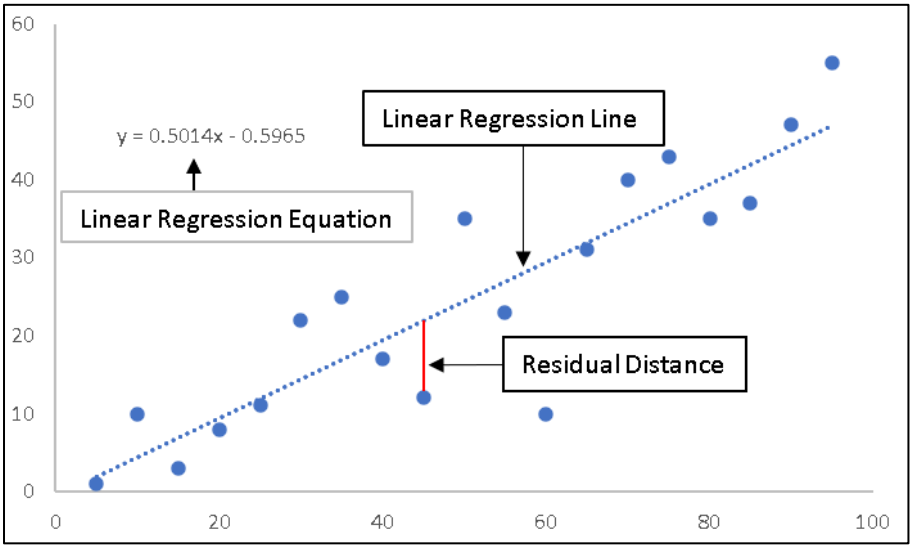


Figure 1.6: Linear Regression Line

A multivariate linear regression model has more than one independent variable and one dependent variable. The multivariate linear regression model follows the following equation [12]:

$$y = \beta_0 + \beta_1 x_1 + \ldots \beta_n x_n \quad (2)$$

Another type of regression is polynomial regression, which is used when the data set follows more of a parabolic shape. Polynomial regression is a variation of linear regression where the relationship between the independent and dependent variables can be a polynomial to the *n*th degree [13]. The equation for polynomial regression is as follows [13]:

$$y = B_0 + B_1 x_1 + B_2 x_2^2 + B_3 x_3^3 + B_n x_n^n \quad (3)$$

where *n* is the degree of polynomial.

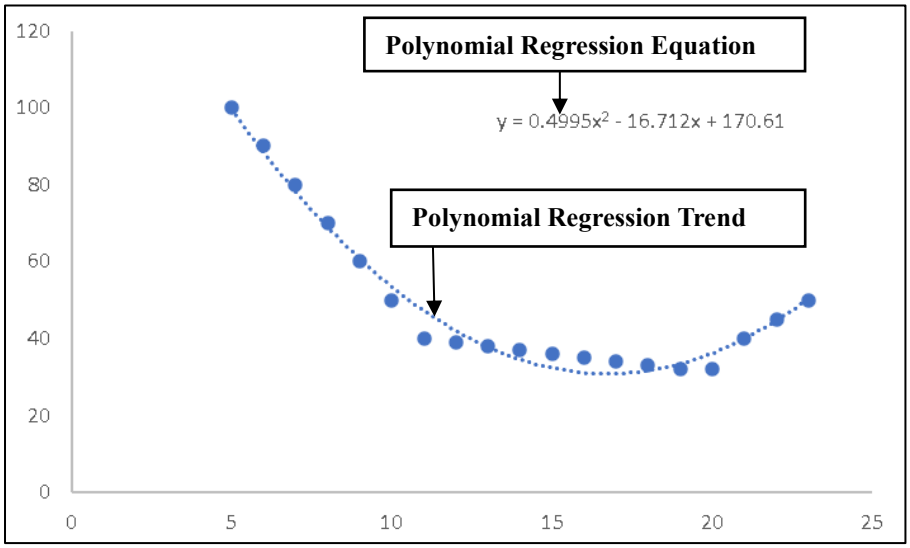


Figure 1.7: Polynomial Regression Trend Curve

A type of classification model is logistic regression. Although it is called logistic regression, it is technically a classification model. Logistic regression uses a logistic function to relate independent variables to dependent variables using a Bernoulli conditional distribution

[11] and can be binomial, ordinal, or multinomial. The outcome of an input data being 1 has a probability, P, while the outcome of an input data being 0 has a probability of 1-P. The equations for logistic regression are as follows [11]:

$$Odds = \ln\left(\frac{P}{1-P}\right) \tag{4}$$

$$logit(P) = \ln\left(\frac{P}{1-P}\right) = Bo + B_1X_1 + B_2X_2 + BnXn \tag{5}$$

### 1.2.2 Unsupervised Learning

Unsupervised learning includes many different types of methods, but the most common are association and clustering. Clustering is an algorithm that groups or "clusters" patterns in data based upon their likeness to one another [14]. In supervised learning the data is in labelled patterns; the algorithm learns these patterns from training and labels new patterns in testing [14]. Clustering labels the data itself and must group them based upon similarities [14]. Clustering is used in many aspects of data analysis including pattern classification, image segmentation, document retrieval and data mining [14]. Association is a series of rules that contain "if-then statements" in order to identify patterns in data [15]. Three attributes determine the strength of an association rule: support, confidence, and lift [16]. Support is defined as the frequency of an item [16]. Confidence is defined as the probability that *Y* also happens given that *X* happens [16]. Lift is defined as the support divided by the probability of *Y* times the probability of *X* given no association between them [16]. Association algorithms are most commonly used in market basket analysis of customer transactions [15].

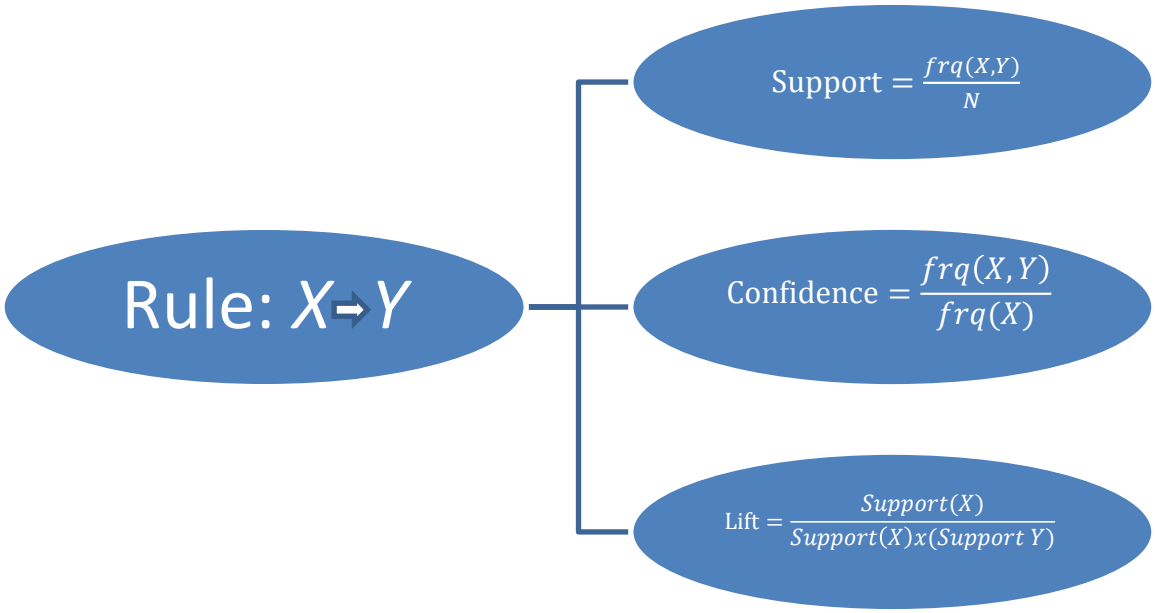


Figure 1.8: Association Attributes [16]

### 1.2.3 Reinforcement Learning

Reinforcement learning is a type of machine learning that involves an agent, an environment, and a reward. The agent produces an action in a certain environment that results in a reward, and cumulative rewards should be maximized [17]. The trial-and-error process is used to achieve learning when the agent interacts with the environment [17]. Reinforcement learning learns by consequences, the program is rewarded for choosing the right action and punished for choosing the wrong action [18]. When compared to supervised and unsupervised learning, reinforcement learning is more analogous to the human intellect [10]. In supervised learning, the data is classified as what is rewarded behavior and what is punished behavior; in reinforcement learning, the environment's response only indicates when the action is rewarded or punished [10]. Reinforcement learning is utilized in a multitude of fields including cloud computing, robotics, recommender systems, inventory control, and vehicular traffic management [10].

## 1.3 Motivation of the Work

In order to produce efficient and geometrically consistent melt pools in laser powder bed fusion, manufacturers must conduct experiments using many different laser and material parameters. These experiments become extremely costly when done repeatedly while also needing plenty of time to run the actual experiments. Another option is to create a CFD model of the laser powder bed fusion process. This requires an operator who is technically proficient in creating CFD models. CFD models also require a great deal of time to generate and can be computationally expensive. In order to eliminate the costs of conducting experiments and generating a CFD model while simultaneously reducing the time it requires to run both of these processes, a machine learning model is created to predict melt-pool geometries in laser powder bed fusion. The model created does not need a proficient operator who is skilled in additive manufacturing or generating CFD models; the operator only needs to be able to input data into the existing code. This machine learning model data results are specific to the material Ti-6Al-4V; however, the model is extremely versatile and can be used for any material that is used in additive manufacturing. The operator will have to collect melt-pool geometries from those specific materials used in laser powder bed fusion and apply that data to the code. The machine learning model is easy to operate and produces accurate results.

# Chapter 2

# Literature Review

The purpose of this literature review is to familiarize the reader with the many different methods of machine learning and how they are implemented in studying melt-pool geometries. Some types of machine learning methods are also described in detail.

Mondal et *al.* [19] created a low-cost surrogate model using the gaussian process and Bayesian optimization in order to estimate process parameter evolution throughout the time of the process [19]. A gaussian process is a stochastic process for a finite set of random variables under a multivariate jointly gaussian distribution [19]. Unlike polynomial regression, gaussian processes do not assume default functional associations between inputs and outputs [19]. Gaussian processes excel in modeling nonlinear functions with the predictions made by a posterior probability distribution [19]. The structure of the model contained three significant steps: "evaluation of the thermal field using an experimentally validated 3D analytically melt-pool evolution model which serves as a source of a data for formulating a gaussian process surrogate," creation of gaussian processes with adaptable kernel structures, under a tight budget formulation of a Bayesian optimization model including a gaussian process surrogate intended on solving global optimization problems [18]. Optimization refers to solving for the maximation of the target function [19]. The equation is as follows:

$$x* = argmax f(x) \tag{6}$$

however, if $f$ is unknown, a gradient free optimization method must be used. Bayesian optimization is a gradient free method which inputs predictions into a surrogate model for active learning by finding the global optima of the target function [19].

Yang et *al.* [20] created a convolutional neural network that can provide accurate classification of melt-pool geometries of real time laser powder bed fusion in a closed loop control. Convolutional neural network is an artificial neural network that uses multi-layer feedforward propagation for data processing of 2D images [20]. A conventional convolutional neural network consists of nested convolutional and pooling layers in the beginning followed by connected layers at the end [20].

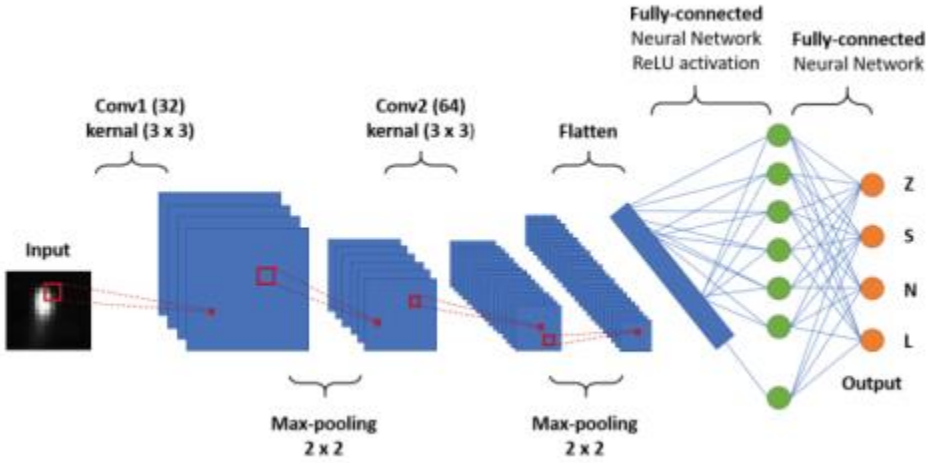


Figure 2.1: Yang et *al.* Convolutional Neural Network Structure [19]

The network of Yang et *al.* was a four-class classification model. The four classes included: no melt-pool (Z), small melt-pool (S), normal melt-pool (N), and larger melt-pool (L). Four examples of each melt-pool image are shown below.

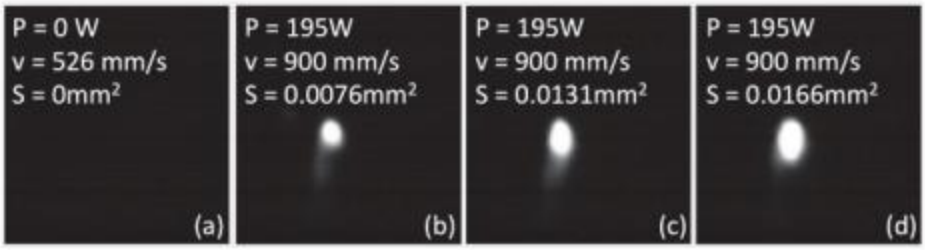


Figure 2.2: Melt-pool image classification from left to right: no melt pool (a), small melt pool (b), normal melt-pool (c), larger melt-pool (d) [20]

The training data consisted of 2763 melt-pool images while the testing data consisted of 2926 melt-pool images. Overall accuracy of the test data was 90.84% with 268 misclassifications [20].

Lee et *al.* [21] employed a data analytics method to calculate multi-physics-based phenomena in additive manufacturing. They utilized machine learning to predict melt-pool geometries in the laser powder bed fusion process [21]. The database consisted of LPBF-processed single track nickel Alloy 625 and nickel Alloy 718 powders [21]. The machine learning models' accuracy increased via the data correlation analysis. Two types of correlation analysis were used, maximal information coefficient (MIC) and Pearson's correlation coefficient (POC). POC can detect and model non-linear relationships while MIC is used to characterize linear relationships. Six different machine learning models predicted the melt-pool geometries: Bayesian ridge regression, kernel ridge regression, linear regression, nearest neighbors regression, random forest regression, and support vector machine. With the implementation of 5-fold cross-validation, the accuracy of the machine learning models was assessed by the coefficient of determination (R^2), which was calculated among the actual and predicted values as a function of the features associated with both analytic correlation approaches [21]. "The results demonstrated the data analytics approach prioritizes key materials/ process parameters of

the melt-pool formation and facilitates the accurate prediction of melt-pool geometries for the additive manufacturing process optimization [21]."

Code surrogates were created by Kamath and Fan [22] for the prediction of melt-pool geometries. The purpose of the study was to show that an accurate code surrogate can be created using a small data set [22]. Data was generated by two models, the Eagar-Tsai model and the Verhaeghe model [22]. The Eagar-Tsai model consists of a Gaussian laser beam scanning across a 3D flat plate [22]. The temperature distribution of the laser beam is used to calculate the melt-pool characteristics as a function of four input parameters: laser power, scanning speed, spot size, absorptivity of the powder [22]. The model is computationally cheap and runs in about one minute [22]. The Verhaeghe model has a greater number of parameters while being computationally more expensive [22]. The model takes into consideration multiple physical phenomena involved in laser powder bed fusion [22]. It takes around one to three hours to run [22]. Three data sets were input into five machine learning models consisting of 461, 100, and 41 points of data [22]. The five types of machine learning models were trained and tested: nearest-neighbor method, regression trees, multivariate adaptive regression splines, support vector regression, and gaussian process [22]. The four input variables were laser power, laser speed, beam size, and absorptivity of the material as a fraction while the 3 target variables were melt-pool width, melt-pool depth, and length [22]. While the support vector regression and regression tree performed the worst, the remaining three methods performed accurately with the gaussian process performing the best [22].

Scime and Beuth [23] categorized melt-pool geometries during testing using the machine learning method support vector machine which is a binary classifier [23]. The purpose of the study was to identify defective melt-pool geometries that contained balling or keyhole effects.

The data set was created by collecting images of the melt-pool geometry during laser powder bed fusion of Inconel 718 alloy by a megapixel Photron FASTCAM Mini AX200 high speed camera [23]. During training, bag of words, an unsupervised machine learning technique, separated the melt-pool images captured by the high-speed camera [23]. The initial captured images are in a Eulerian frame of reference and make it difficult to differentiate the melt-pool geometries; therefore, Bag of words is applied to convert the images into a coaxial reference frame [23].

Akbari et *al.* [24] designed a two feed forward back propagation neural networks with eleven and fourteen neurons to predict melt-pool geometries of Ti-6Al-4V alloy. A neural network has three main features: structure that exhibits connections between layers and neurons, a learning algorithm that updates the weighs on the connections, and a transfer function [24].

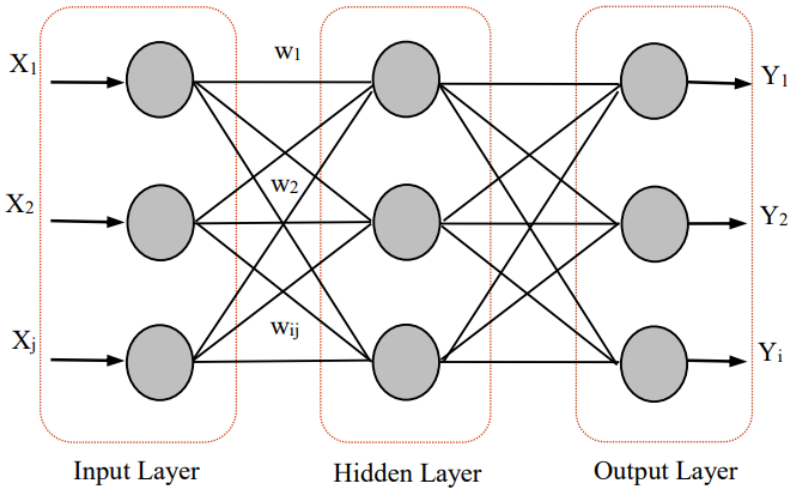


Figure 2.3: Structure of a Neural Network [24]

The input layer of a neural network is multiplied by the weights which are randomly selected [24]. The layer output is calculated by the equation below [24]:

$$Y = f(\sum X_j W_{ij} + b) \quad (7)$$

"Where $W_{ij}$ is the weight of the connection bewtweeen each neuron (j) in the input layer and each neuron (i) in the hidden layer and between hidden layer and output layer. $X_j$ is the value of the input (j) at the input layer, f is the transfer function and b is the bias [24]." The tan-sigmoid transfer function was used the hidden and output layers. Tan-sigmoid function equation is:

$$f(x) = \frac{1}{1+e^{-x}} \quad (7)$$

Akbari et *al.* implemented the mean square error in order to reduce the error. The results showed a mean square error of .079 and .063 for each neural network and agreed with experimental and numerical data.

# Chapter 3

# Material and Methodology

## 3.1 Material

The material used in the laser powder bed fusion process is Ti-6Al-4V alloy. Ti-6Al-4V is chosen because it is commonly used in many industries, especially the aerospace, dental, and biomedical industries [25]. It is a high entropy material, can withstand very high temperatures, and has a large strength to weight ratio [25]. However, Ti-6Al-4V is an expensive alloy compared to other leading industry metals such as stainless steel and carbon steel; therefore, it is imperative to ensure that the L-PBF process is conducted correctly the first time [2]. A correct combination of the laser power, scanning speed, spot size, layer thickness, and powder porosity is required to reduce or eliminate material wastage, part defects, and time delays in the L-PBF process [2].

Table 3.1: Comparison of Material Properties of Ti-6Al-4V and 316L Stainless Steel [26-28]

| Material Property | Ti-6Al-4V | 316L Stainless Steel |
|---|---|---|
| Hardness (Brinell) | 379 | 217 |
| Tensile Strength Ultimate (MPa) | 1170 | 485 |
| Tensile Strength Yield (MPa) | 1100 | 170 |
| Melting Point (°C) | 1604-1660 | 1390-1440 |
| Shear Modulus (GPa) | 44 | 82 |
| Price ($) of Metal Sheet (.05"x24"x36") | 1733.63 | 221.31 |

As seen from Table 3.1, the hardness, ultimate tensile strength, yield tensile strength, and range of melting points of Ti-6Al-4V are much greater than the 316L stainless steel. However, stainless steel has a greater shear modulus value. The price of a sheet of Ti-6Al-4V is almost eight times as much as a sheet of 316L stainless steel.

## 3.2 Machine Learning Methods

Multiple regression models were trained and tested with the data set, and each model implements a different method to predict the outputs. Every model used will not be described in detail (there is a total of nineteen different models); however, one of the worst performing models and one of the best performing models will be discussed in this section.

### 3.2.1 Linear Regression

Linear regression relates an independent variable to a dependent variable. When all the data is plotted on a scatter plot, the trend line will follow the equation as referenced before:

$$y_p = \beta_o + \beta_1 x \tag{8}$$

This line is representative of the predicted values. In order to find the most accurate trend line, the variables $B_o$ and $B_1$ must be optimized, meaning finding the best values of those variables that will give the lowest error. An error function is a measurement of the distance between the predicted value and the actual value, which is referred to as the residual distance. There are many types of error functions used to minimize the residual distance; one common method is the mean square error. Starting with the mean square error equation [29]:

$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_p)^2}{n} \tag{9}$$

where $y_i$ is the actual value and $y_p$ is the predicted value. Substituting (8) into (9) produces [29]:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \beta_o + \beta_1 x)^2}{n} \quad (10)$$

Next, gradient descent will be calculated which will find the minimum of the error function. First, values of $B_o$ and $B_1$ will start at zero [29]. The partial derivative of the mean square error function will be calculated with respect to $B_o$ and $B_1$[29]:

$$d_{B_0} = -\frac{2}{n}\sum_{i=1}^{n}(y_i + y_p) \quad (11)$$

$$d_{B_1} = -\frac{2}{n}\sum_{i=1}^{n} B_1(y_i + y_p) \quad (12)$$

Next, current value of $dB_o$ and $dB_1$ will be updated with the following equations where L is the learning rate; the learning rate is the rate at which the $B_1$ changes, also known as the step size [29].

$$new\ B_o = B_o - L\left(d_{B_o}\right) \quad (13)$$

$$new\ B_1 = B_1 - L(d_{B_1}) \quad (14)$$

This is done until the error loss function is very small.

### 3.2.2 Random Forest

The random forest model consists of an ensemble of decision trees which are structured like a flowchart of rules that lead to a prediction [30]. Data is input into the root node of the tree, and the algorithm splits the data into smaller groups until the data cannot be split any further or it reaches a rule that stops the splitting [30]. Decision trees can be regressive or categorical. Random forest is a collection of many decision trees, where each tree is created on random

subsets of variables and observations [30]. The randomness of the variables and observations creates a more accurate prediction [30]. “The random predictors are used at each split in the tree which de-correlate the trees forming the forest [30].”

### 3.3 Error Calculations

Error calculations measure the residual distance between the actual values and predicted values in regression, thus exhibiting the accuracy of the model [2]. Four error functions were tested to find the least amount of error: the mean absolute error, the root mean square error, relative error percentage, and root relative squared error. The mean absolute error (MAE) formula calculates the error between actual values and predicted values. The formula is

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{16}$$

with *y* as the predicted value, *x* as the actual value, and *n* as the total number of data points. The root-mean-square-error (RMSE) or the root-mean-square-deviation (RSMD) measures the standard deviation of the distances between the actual values and the predicted values (residuals). The formula is given below:

$$RSMD = \sqrt{\frac{\sum_{i=1}^{N} (y_i - x_i)^2}{n}} \tag{17}$$

with *y* as the predicted value, *x* as the actual value, and *n* as the total number of data points. Relative absolute error (RAE) percentage calculates the absolute error divided by the actual

value. The absolute error is the absolute value of the difference between the predicted value and actual value. The formula is

$$Absolute\ Error = |y_i - x_i| \tag{18}$$

And

$$RAE = \frac{Absolute\ error}{x_i} \times 100 \tag{19}$$

with *y* as the predicted value and *x* as the known value. The root relative squared error formula is

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{20}$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{21}$$

with *y* as the predicted value, *x* as the actual value, and *n* as the total number of data points [2].

### 3.4 Preparation of Data Set

The data set is compiled from multiple sources, from both experimental data and simulated data for single track laser scans. Dilip et *al*. [8] ran experiments on Ti-6Al-4V researching the effects laser power and scan speed had on melt-pool width, microstructures, and porosity. Gong et *al*. [7] performed experiments on Ti-6Al-4V using a range of laser power and speed values to

produce melt pools. These melt pools were then examined to find the optimum hatch spacing in the fabrication of metallic test sheets [7]. The topology of the test sheets was analyzed to find the ideal parameters [7]. Solyemez [31] created a numerical model that would simulate melt-pool widths and depths created by a wide range of laser powers and scan speeds from experiments [31]. Kusuma [32] performed experiments with varying laser powers and scans speed to examine the effect these parameters have on melt-pool width and melt-pool depth.

After interpreting their data and collecting feedback from the authors, the data set is prepared in the form of an Excel file having 667 rows and 7 columns of variables, five of the columns are features ($x_1$ to $x_5$) and two are target variables ($y_1$ and $y_2$) [2]. However, two separate models are created, one targeting the five input parameters' influence on the melt-pool width and the other focusing on the five input parameters' influence on the melt-pool depth [2]. The input parameters include laser power, $P$ (in W), scanning speed, $v_s$ (in mm/s), spot size, $\Phi$ (in $\mu$m), layer thickness, $l_t$ (in $\mu$m), and powder porosity, $\varphi$ (%). The output parameters are melt-pool width, $w$ and melt-pool depth, $d$ (both in $\mu$m). This model is trained twice, once for the melt pool-width and once for the melt-pool depth. The maximum and minimum values of the variables in the data set are shown in Table 3.2 [2].

Table 3.2: Maximum and Minimum Values in the Data Set

| **Param.*** | $P$**(w)** | $v_s$ **(mm/s)** | $\Phi$ **(**$\mu$**m)** | $l_t$ **(**$\mu$**m)** | $\varphi$**(%)** | $w$ **(**$\mu$**m)** | $d$**(**$\mu$**m)** |
|---|---|---|---|---|---|---|---|
| Max | 400 | 3200 | 100 | 70 | 50 | 412 | 980 |
| Min | 50 | 100 | 58 | 30 | 0 | 45 | 5 |

***Param. = Parameters**

The data is separated into two groups, a training group and a testing group. Approximately 80% of the data are categorized into the training group while 20% are organized into the test group for 5-fold cross-validation while 90% of the data is split into the training group and 10% is split into the testing group during 10-fold cross-validation. The entire data set is shuffled before being presented to the regression models in order to prevent any biasness. For the training data set, the model is given the inputs as well as the outputs. It creates a scatter plot and a regression line using the log-structured merge-tree [2]. Once the percent error is calculated and the model is considered accurate, the model is given the test data. The test data only includes the inputs; the model predicts the outputs based upon the regression line it previously calculated. The predicted values are measured against the actual values to see how accurate the model is [2].

### 3.5 K-Fold Cross-validation

In order to test the accuracy of the model, the *k*-fold cross-validation method is used. With this method, the original sample data are shuffled and randomly partitioned into *k* equal parts [2]. Afterwards, the model will choose one set of data as the validation set and the other sets (the remaining *k*–1) as the training sets. This is done *k* times, with each specific set of data serving as the validation set exactly once [3]. When the model has completed its cross-validations, the results are averaged to give an accuracy of the model [3]. This study used a 5-fold cross-validation model as well as a 10-fold cross-validation model and compared which one was more accurate [2].


Figure 3.1: Representation of 5-Fold Cross-Validation [33]

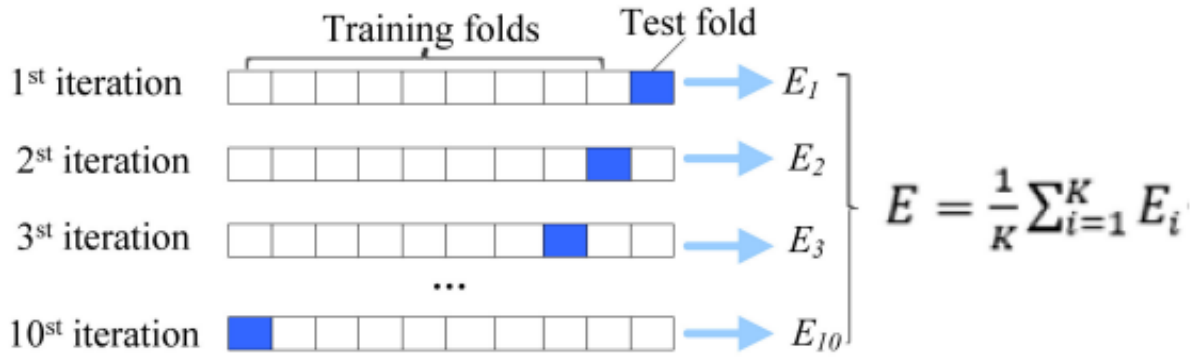


Figure 3.2: Representation of 10-Fold Cross-Validation

## 3.5 Smaller and Larger Data Sets

As the research progressed, the data set continued to grow. As a result, this entire process was repeated twice with another data set created in Excel which consisted of 2922 rows and 7 columns of the same variables. This data is significantly larger than the first, and the two data sets are compared to confirm that increasing the data set will decrease the error of the model.

# Chapter 4

# Results and Discussion

## 4.1 Data Correlation

Figures 4.1 and 4.2 show the correlation between the input parameters and output parameters of the entire data set. They also show the range of values of each parameter. The most varied parameters are laser power and scanning speed while spot size and porosity were the same value for most of the data entries. Power has a positive linear relationship to melt-pool width and melt-pool depth most exemplified in the bottom left corner of each scatter plot with power as the x-axis. This would make sense as the more power the laser has, the more energy exerted upon the substrate, increasing the melt-pool width and depth [2]. However, some higher power data points lie in the bottom right corners of the scatter plots with power as the x-axis. These power data points are associated with higher scanning speeds; consequently, these data points have lower depths of penetration and surface area spread. Even though the power parameter is set at a high threshold, the scanning speed can dictate the depth of penetration if set too high. The scanning speed has a negative exponential relationship to both melt-pool width and melt-pool depth. The widths and depths are greatest when the scanning speed is the slowest. This is expected; as the scanning speed increases, there is less time for the energy of the laser to penetrate the substrate. This is clearly illustrated in the scanning speed vs melt-pool width and the scanning speed vs melt-pool depth plots. The highest value of each output parameter is in the top left corner; as scanning speed increases, both output parameters slowly decrease. Figure 4.3 shows the heatmap correlation matrix where the correlation among the features and target variables are qualitatively portrayed on a scale of -0.5 to 1 [2].

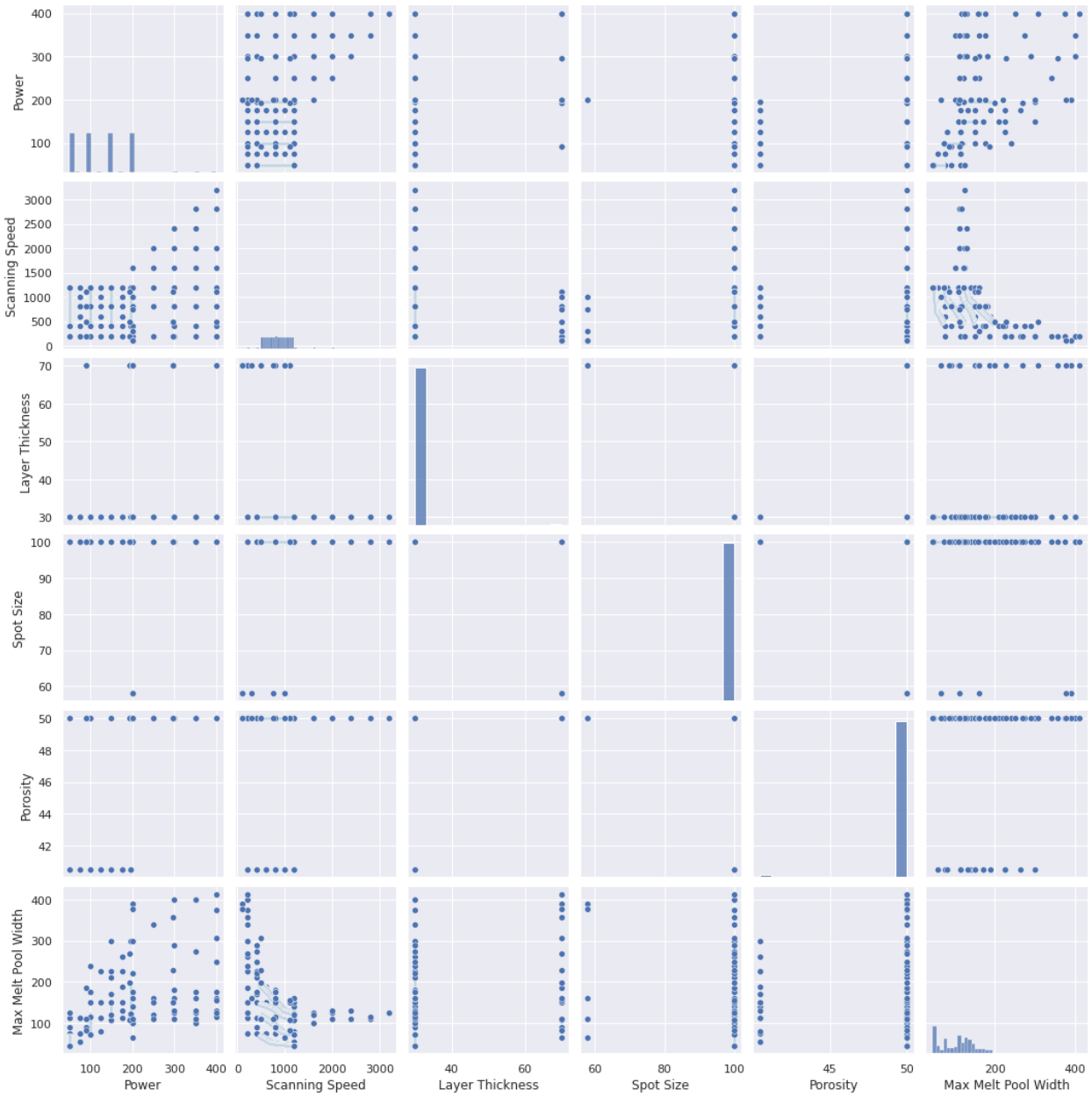


Figure 4.1: Scatter Plots Between Melt-Pool Widths and Correlated Variables

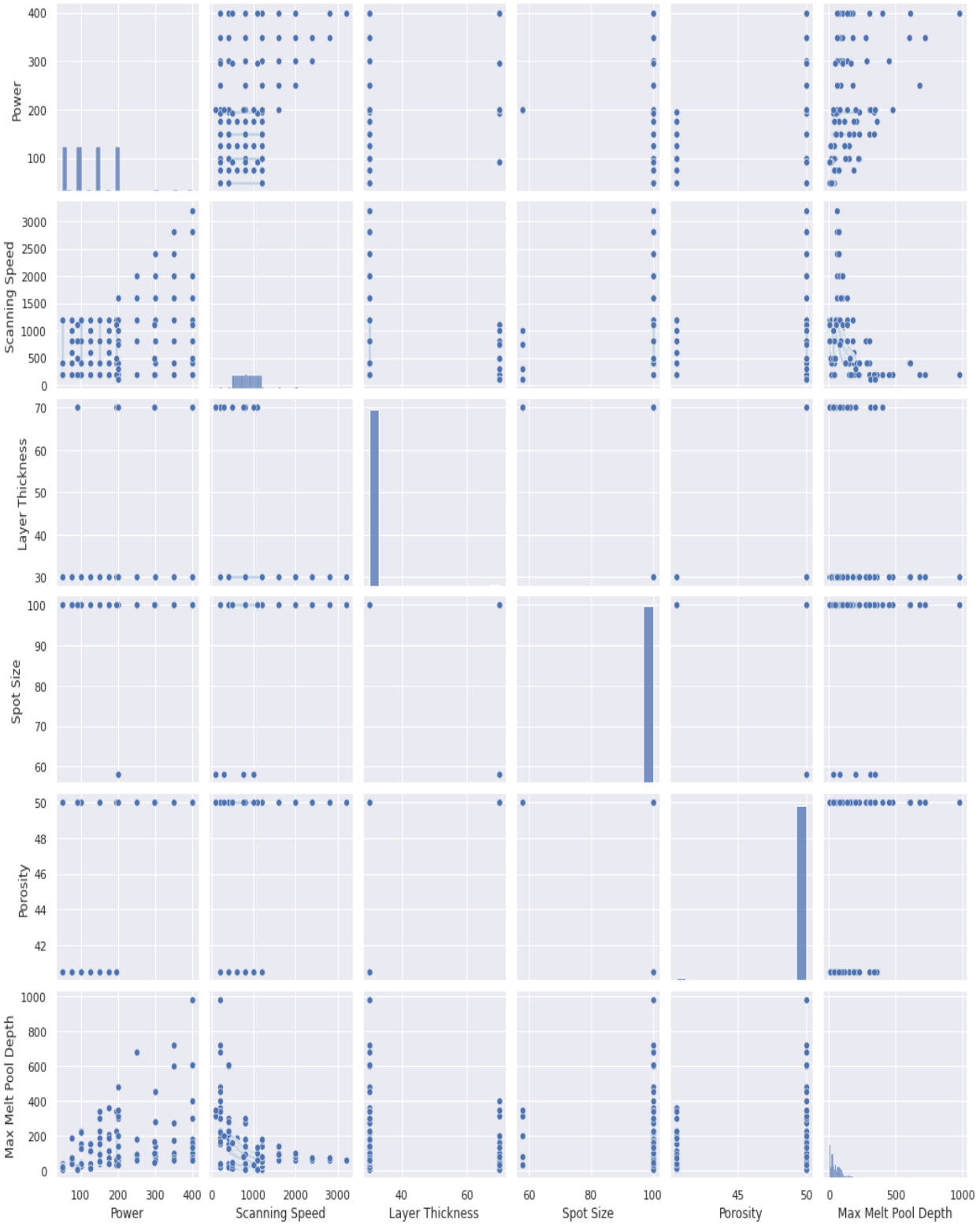


Figure 4.2: Scatter Plots Between Melt-Pool Depths and Correlated Variables

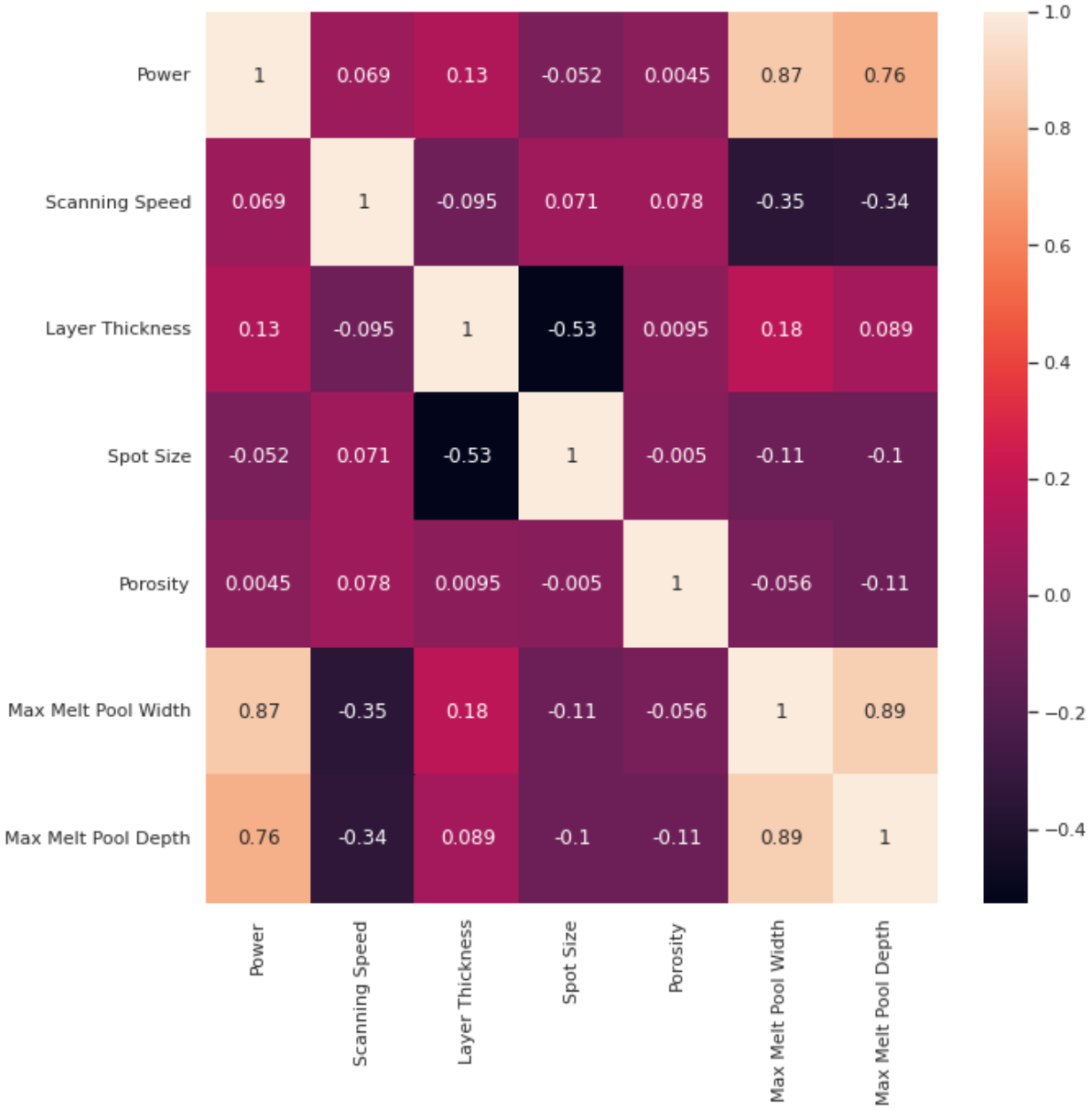


Figure 4.3: The Heat Map Correlation Matrix

## 4.2 Performance Analysis

First, the data set is tested using Python code and implementing the five-fold and ten-fold cross-validation techniques for each of the data sets [2]. The negative mean absolute error is

obtained from various regressor models and compared for evaluating the performance. These results are obtained by using open-source library codes from scikit-learn and modifying the codes based on the current analysis. For the smaller data set, the highest error in both the five-fold cross-validation and ten-fold cross-validation was calculated by the linear regression model. This error was significantly higher than the lowest calculated error, the extra trees regressor. As a whole, the ten-fold cross-validation error in all models with the exception of the melt-pool depth in linear regression, was slightly smaller than the error of the five-fold cross-validation error. The 10-fold cross-validation extra trees regressor produced the least amount of error and performed the best of all the models.

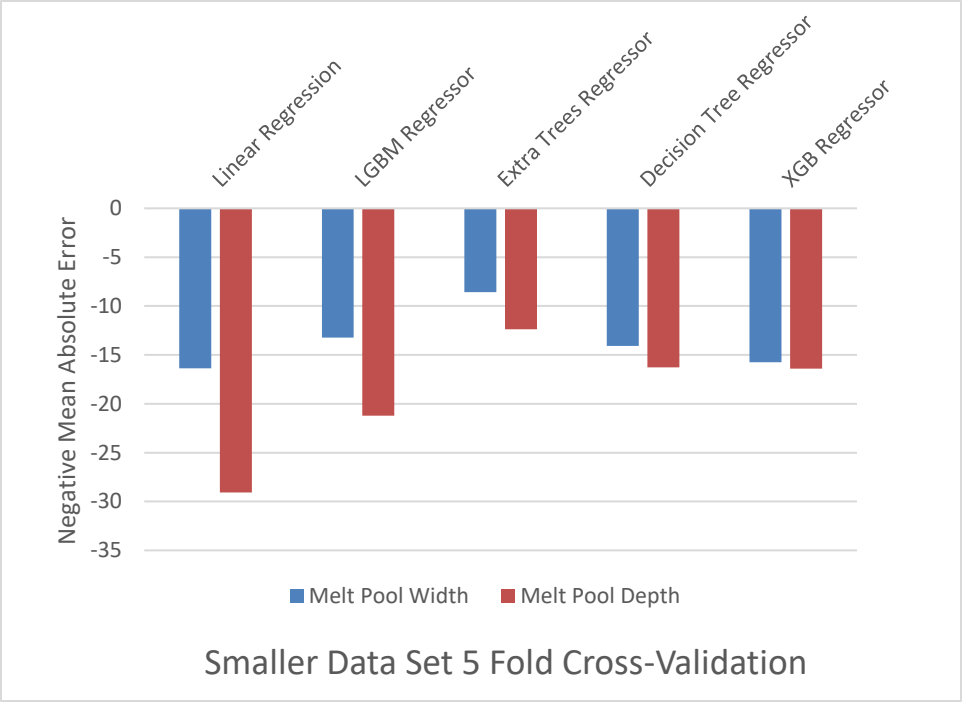


Figure 4.4: Smaller Data Set 5-Fold Cross-Validation

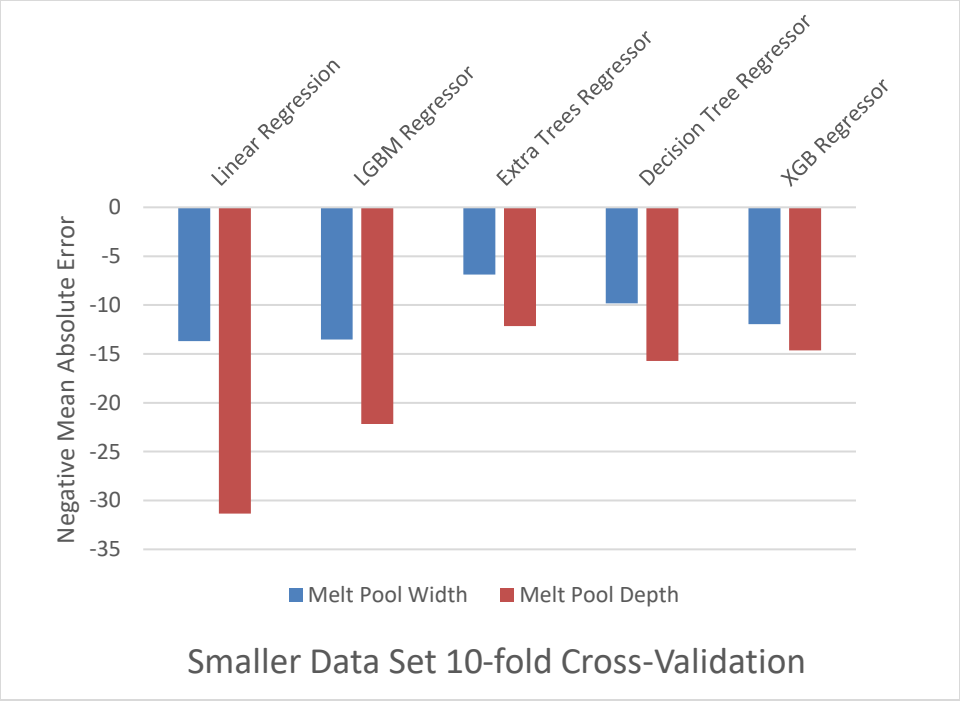


Figure 4.5: Smaller Data Set 10-Fold Cross-Validation

For the larger data set, the five-fold cross-validation error was again highest in the linear regression model. The extra trees regressor had the lowest error and was considerably lower than all the other models. The linear regression model produced the highest error in the ten-fold cross-validation model while the extra trees regressor performed the best, producing the least amount of error. All models performed better in the ten-fold cross-validation than in the five-fold cross-validation for the larger data set.

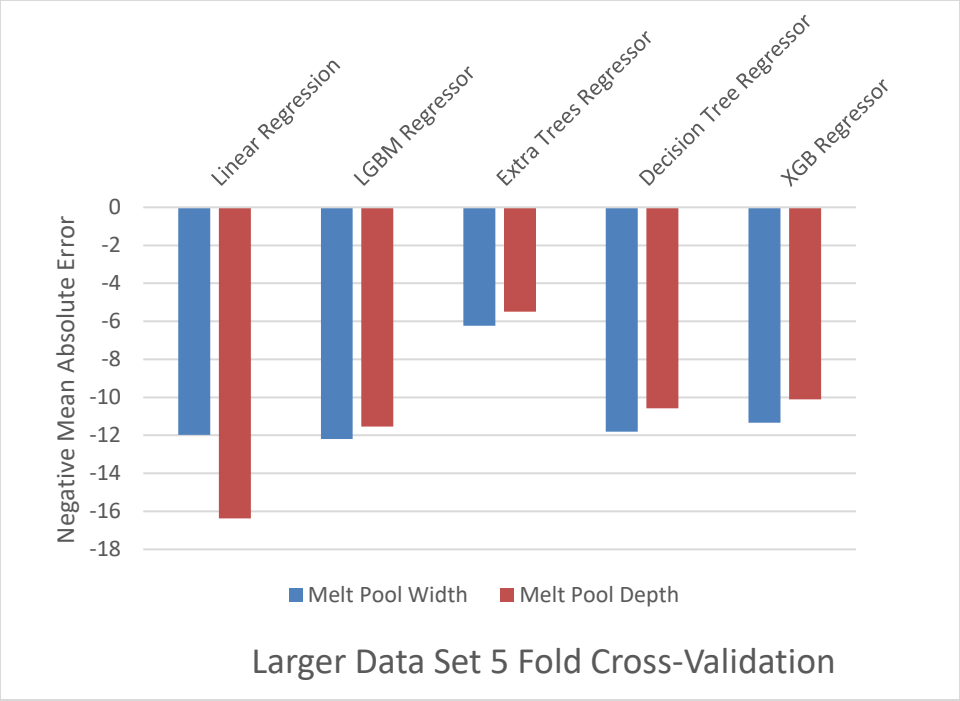


Figure 4.6: Larger Data Set 5-Fold Cross-Validation

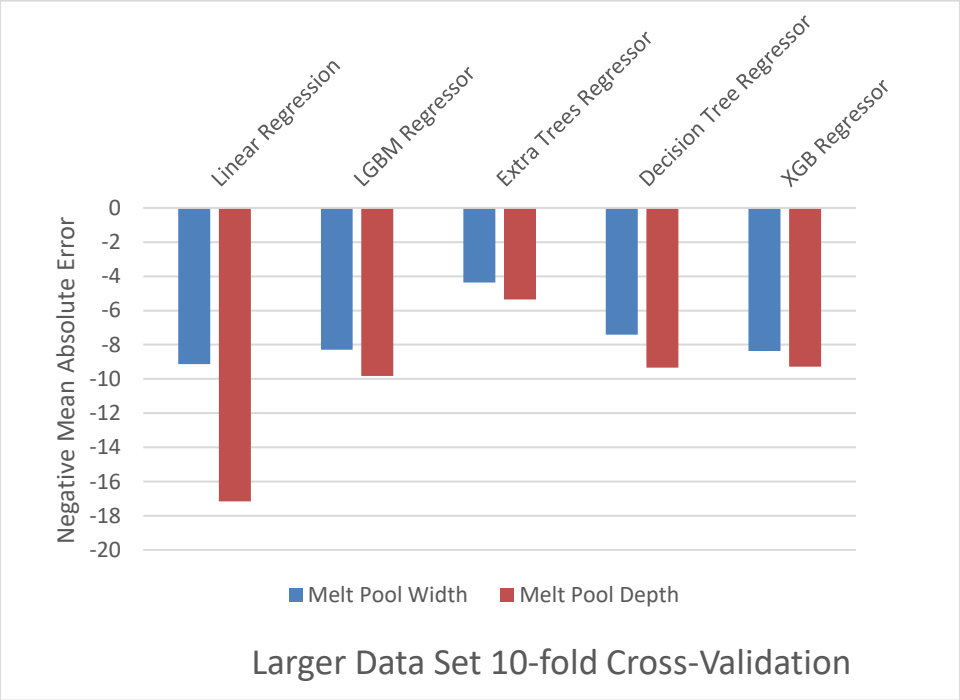


Figure 4.7: Larger Data Set 10-Fold Cross-

The larger data set outperforms the smaller data set in all regression models. This confirms the hypothesis that increasing the data set will increase the accuracy of the model. The best performing model, extra trees regressor, calculated the smallest error in the ten-fold cross-validation of the larger data set in comparison to the other k fold cross-validation models.

More results on the performance of the regression analysis are shown in Tables 4.1 to 4.8. The melt-pool width and depth are predicted separately by two regression models. These results are obtained by training and testing the data set using Weka 3.6.14 tool.

Tables 4.1 to 4.4 exhibit the error values of the width and depth melt-pool predictions in five-fold cross-validation and ten-fold cross-validation. The highest error values in all models are produced in the simple linear regression model, and the lowest error values in all models are calculated in the additive regression random forest models. The lowest error function within the additive regression random forest is the mean absolute error. In the depth prediction, the mean absolute error is slightly smaller in the ten-fold cross-validation with a value of 6.2466 than the five-fold cross-validation showing a value of 7.199. Regarding the mean absolute error width prediction, the values are nearly identical for the five-fold and 10-fold cross-validations producing values of 3.4759 and 3.4928 respectively.

Table 4.1: Melt-Pool Width Prediction for 5-Fold Cross-Validation for Smaller Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.9124 | 11.8832 | 22.4233 | 30.4354 | 40.9033 |
| Simple Linear Regression | 0.7418 | 21.9754 | 36.725 | 56.2837 | 66.9915 |
| SVM Regression (normalized) kernel poly | 0.9074 | 11.0029 | 23.0219 | 28.1807 | 41.9952 |
| SVM Regression (standardized) kernel poly | 0.9088 | 10.9688 | 22.8557 | 28.0934 | 41.6919 |
| kNN (k=3) | 0.9169 | 5.9672 | 21.953 | 15.2833 | 40.0453 |
| kNN (k=5) | 0.911 | 6.2561 | 22.7518 | 16.0232 | 41.5024 |
| kNN (k=7) | 0.9071 | 6.4085 | 23.3343 | 16.4135 | 42.5649 |
| kNN (k=9) | 0.8983 | 6.6048 | 24.3083 | 16.9162 | 44.3417 |
| Additive Regression (Decision Stump) | 0.9232 | 13.0485 | 21.1553 | 33.42 | 38.5902 |
| Additive Regression (Random Forest) | 0.9821 | 3.4759 | 10.3384 | 8.9026 | 18.8587 |
| Additive Regression (Random Tree) | 0.9541 | 5.1415 | 16.95 | 13.1685 | 30.9192 |
| RepTree | 0.9616 | 6.735 | 15.0409 | 17.2499 | 27.4367 |
| Random Forest | 0.9775 | 4.0185 | 11.5742 | 10.2923 | 21.1129 |
| Random Tree | 0.9547 | 5.9081 | 16.6614 | 15.1318 | 30.3927 |
| Multi-Layer Perceptron | 0.9693 | 7.7993 | 13.667 | 19.9756 | 24.9304 |

Table 4.2: Melt-Pool Depth Prediction for 5-Fold Cross-Validation for Smaller Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.8091 | 27.5143 | 49.4172 | 56.9285 | 58.7073 |
| Simple Linear Regression | 0.6324 | 30.1196 | 65.1447 | 62.3192 | 77.3914 |
| SVM Regression (normalized) kernel poly | 0.8079 | 19.6716 | 54.8246 | 40.7017 | 65.1312 |
| SVM Regression (standardized) kernel poly | 0.8077 | 19.6611 | 54.8745 | 40.6798 | 65.1905 |
| kNN (k=3) | 0.8762 | 9.0828 | 41.305 | 18.7928 | 49.07 |
| kNN (k=5) | 0.8694 | 9.5005 | 43.1183 | 19.657 | 51.2241 |
| kNN (k=7) | 0.8426 | 10.3226 | 46.1571 | 21.3581 | 54.8343 |
| kNN (k=9) | 0.8277 | 11.004 | 48.0119 | 22.7678 | 57.0377 |
| Additive Regression (Decision Stump) | 0.82 | 24.8862 | 48.3859 | 51.4909 | 57.482 |
| Additive Regression (Random Forest) | 0.93 | 7.199 | 30.9371 | 14.8951 | 36.753 |
| Additive Regression (Random Tree) | 0.8681 | 10.6584 | 42.5885 | 22.0528 | 50.5948 |
| RepTree | 0.8405 | 14.4961 | 45.5895 | 29.9932 | 54.16 |
| Random Forest | 0.9058 | 9.1202 | 35.6688 | 18.8702 | 42.3743 |
| Random Tree | 0.8438 | 11.5544 | 47.2061 | 23.9066 | 56.0805 |
| Multi-Layer Perceptron | 0.9334 | 13.0646 | 30.5374 | 27.0315 | 36.2782 |

Table 4.3: Melt-Pool Width Prediction for 10-Fold Cross-Validation for Smaller Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.9028 | 12.0265 | 23.5775 | 30.7875 | 43.0162 |
| Simple Linear Regression | 0.7446 | 21.9747 | 36.5538 | 56.2543 | 66.6911 |
| SVM Regression (normalized) kernel poly | 0.879 | 11.5338 | 26.2092 | 29.5262 | 47.8176 |
| SVM Regression (standardized) kernel poly | 0.8796 | 11.515 | 26.1574 | 29.478 | 47.7232 |
| kNN (k=3) | 0.9063 | 6.0586 | 23.1808 | 15.5097 | 42.2925 |
| kNN (k=5) | 0.901 | 6.3954 | 23.8423 | 16.372 | 43.4994 |
| kNN (k=7) | 0.9039 | 6.4076 | 23.5888 | 16.4033 | 43.0369 |
| kNN (k=9) | 0.892 | 6.7965 | 24.9185 | 17.3988 | 45.4628 |
| Additive Regression (Decision Stump) | 0.9124 | 13.5996 | 22.8324 | 34.8144 | 41.6567 |
| Additive Regression (Random Forest) | 0.9805 | 3.4928 | 10.7886 | 8.9414 | 19.6834 |
| Additive Regression (Random Tree) | 0.9625 | 4.7474 | 15.0955 | 12.1532 | 27.5412 |
| RepTree | 0.9547 | 7.4082 | 16.4928 | 18.9647 | 30.0904 |
| Random Forest | 0.9767 | 3.9497 | 11.7548 | 10.111 | 21.4462 |
| Random Tree | 0.953 | 6.0636 | 16.8871 | 15.5226 | 30.8098 |
| Multi-Layer Perceptron | 0.9256 | 16.7674 | 22.9872 | 42.9238 | 41.9393 |

Table 4.4: Melt-Pool Depth Prediction for 10-Fold Cross-Validation for Smaller Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.8069 | 27.5294 | 49.6764 | 56.9235 | 58.9976 |
| Simple Linear Regression | 0.6348 | 30.1076 | 64.9649 | 62.2545 | 77.1549 |
| SVM Regression (normalized) kernel poly | 0.7927 | 19.8468 | 55.3832 | 41.0379 | 65.7753 |
| SVM Regression (standardized) kernel poly | 0.7916 | 19.9009 | 55.4788 | 41.1498 | 65.8888 |
| kNN (k=3) | 0.8773 | 9.0116 | 40.9779 | 18.6336 | 48.6669 |
| kNN (k=5) | 0.8721 | 9.4356 | 42.6792 | 19.5103 | 50.6875 |
| kNN (k=7) | 0.849 | 9.9648 | 45.2984 | 20.6045 | 53.7981 |
| kNN (k=9) | 0.8313 | 10.6759 | 47.4597 | 22.0749 | 56.365 |
| Additive Regression (Decision Stump) | 0.8326 | 23.5127 | 46.7722 | 48.6181 | 55.5485 |
| Additive Regression (Random Forest) | 0.9429 | 6.2466 | 28.0736 | 12.9162 | 33.3412 |
| Additive Regression (Random Tree) | 0.8158 | 12.1472 | 52.5371 | 25.1171 | 62.3951 |
| RepTree | 0.789 | 14.3253 | 51.6844 | 29.6208 | 61.3824 |
| Random Forest | 0.9194 | 8.1925 | 33.3494 | 16.94 | 39.607 |
| Random Tree | 0.8556 | 11.5718 | 46.2141 | 23.9274 | 54.8856 |
| Multi-Layer Perceptron | 0.9113 | 18.7831 | 34.7865 | 38.8385 | 41.3138 |

The results of the larger data set seen in Tables 4.5 to 4.8 are similar compared to the smaller data set. In all cross-validation models, simple linear regression once again produces the most error while additive regression random forest produces the least. The mean absolute error calculated the least amount of error in all models as well. However, the five-fold cross-validation models for both the width and depth predictions were more accurate than the ten-fold cross-validation models. The width prediction for the five-fold cross-validation model is 0.8579 while the ten-fold cross-validation model is 0.8833; the depth prediction for the five and ten cross fold models shows a value of 1.3664 and 1.4864, respectively.

When evaluating the larger and smaller data sets, the larger data set is significantly more accurate. The lowest error value for the width prediction of the larger and smaller data sets were 0.8579 and 3.4759, respectively. And the depth prediction's lowest error values for the larger and smaller data sets were 1.3664 and 6.2466, respectively. This data confirms once again that increasing the data set will produce a more accurate model.

Table 4.5: Melt-Pool Width Prediction for 5-Fold Cross-Validation for Larger Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.955 | 7.0737 | 13.1638 | 19.8162 | 29.6433 |
| Simple Linear Regression | 0.8651 | 14.8741 | 22.2585 | 41.6682 | 50.1237 |
| SVM Regression (normalized) kernel poly | 0.9549 | 6.9062 | 13.3004 | 19.347 | 29.951 |
| SVM Regression (standardized) kernel poly | 0.9547 | 6.9093 | 13.3452 | 19.3556 | 30.0518 |
| kNN (k=3) | 0.9765 | 1.259 | 9.5792 | 3.5269 | 21.5712 |
| kNN (k=5) | 0.973 | 1.3538 | 10.2875 | 3.7925 | 23.1664 |
| kNN (k=7) | 0.9662 | 1.4538 | 11.4696 | 4.0727 | 25.8282 |
| kNN (k=9) | 0.9641 | 1.5096 | 11.8186 | 4.2289 | 26.6141 |
| Additive Regression (Decision Stump) | 0.9511 | 9.0502 | 13.8195 | 25.3531 | 31.12 |
| Additive Regression (Random Forest) | 0.9913 | 0.8579 | 5.8506 | 2.4032 | 13.1749 |
| Additive Regression (Random Tree) | 0.9738 | 1.3918 | 10.1122 | 3.8989 | 22.7714 |
| RepTree | 0.9738 | 2.6613 | 10.0963 | 7.4553 | 22.7357 |
| Random Forest | 0.9905 | 1.2214 | 6.121 | 3.415 | 13.7837 |
| Random Tree | 0.9842 | 1.9343 | 7.8958 | 5.4188 | 17.7804 |
| Multi-Layer Perceptron | 0.9789 | 5.039 | 9.4655 | 14.1161 | 21.3153 |

Table 4.6: Melt-Pool Depth Prediction for 5-Fold Cross-Validation for Larger Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.8613 | 14.0505 | 27.5084 | 38.023 | 50.7921 |
| Simple Linear Regression | 0.7607 | 17.7688 | 35.1421 | 48.0853 | 64.8871 |
| SVM Regression (normalized) kernel poly | 0.857 | 11.4893 | 29.487 | 31.092 | 54.4454 |
| SVM Regression (standardized) kernel poly | 0.8572 | 11.4812 | 29.5241 | 31.0701 | 54.5139 |
| kNN (k=3) | 0.9534 | 1.9049 | 16.4715 | 5.1549 | 30.4134 |
| kNN (k=5) | 0.9398 | 2.05484 | 18.7751 | 5.5704 | 34.6667 |
| kNN (k=7) | 0.9277 | 2.2442 | 20.522 | 6.0732 | 37.8922 |
| kNN (k=9) | 0.9213 | 2.4152 | 21.3363 | 6.536 | 39.3957 |
| Additive Regression (Decision Stump) | 0.8783 | 14.2288 | 26.1502 | 38.5055 | 48.2843 |
| Additive Regression (Random Forest) | 0.9758 | 1.3664 | 11.8903 | 3.6978 | 21.9545 |
| Additive Regression (Random Tree) | 0.9531 | 2.0225 | 16.3949 | 5.4732 | 30.2718 |
| RepTree | 0.9153 | 5.0917 | 22.0243 | 13.7791 | 40.666 |
| Random Forest | 0.9687 | 1.9305 | 13.482 | 5.2244 | 24.8935 |
| Random Tree | 0.9466 | 2.9903 | 17.7637 | 8.0922 | 32.7992 |
| Multi-Layer Perceptron | 0.972 | 7.0322 | 13.7756 | 19.0303 | 25.4355 |

Table 4.7: Melt-Pool Width Prediction for 10-Fold Cross-Validation for Larger Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.9523 | 7.1164 | 13.5423 | 19.9364 | 30.4991 |
| Simple Linear Regression | 0.865 | 14.882 | 22.2719 | 41.6916 | 50.1594 |
| SVM Regression (normalized) kernel poly | 0.9533 | 6.8939 | 13.5311 | 19.313 | 30.474 |
| SVM Regression (standardized) kernel poly | 0.9531 | 6.897 | 13.5746 | 19.3219 | 30.5718 |
| kNN (k=3) | 0.9655 | 1.3693 | 11.5633 | 3.8359 | 26.0421 |
| kNN (k=5) | 0.9713 | 1.3336 | 10.5926 | 3.7361 | 23.856 |
| kNN (k=7) | 0.9667 | 1.4421 | 11.3915 | 4.0399 | 25.6533 |
| kNN (k=9) | 0.9634 | 1.4803 | 11.9357 | 4.1469 | 26.8809 |
| Additive Regression (Decision Stump) | 0.9521 | 9.1662 | 13.6946 | 25.6788 | 30.842 |
| Additive Regression (Random Forest) | 0.9905 | 0.8833 | 6.101 | 2.4746 | 13.7404 |
| Additive Regression (Random Tree) | 0.9702 | 1.5059 | 10.8199 | 4.2187 | 24.3679 |
| RepTree | 0.9844 | 2.3686 | 7.8149 | 6.6355 | 17.6002 |
| Random Forest | 0.9905 | 1.1896 | 6.1105 | 3.3326 | 13.7617 |
| Random Tree | 0.9841 | 1.9006 | 7.8906 | 5.3244 | 17.7708 |
| Multi-Layer Perceptron | 0.9764 | 5.1238 | 9.6067 | 14.3541 | 21.6356 |

Table 4.8: Melt-Pool Depth Prediction for 10-Fold Cross-Validation for Larger Data Set

| Method | Correlation coefficient | Mean absolute error | Root mean squared error | Relative absolute error (%) | Root relative squared error (%) |
|---|---|---|---|---|---|
| Linear Regression | 0.8596 | 14.0663 | 27.6622 | 38.0711 | 51.0798 |
| Simple Linear Regression | 0.7601 | 17.7506 | 35.1793 | 48.0428 | 64.9605 |
| SVM Regression (normalized) kernel poly | 0.8558 | 11.5286 | 29.5535 | 31.2026 | 54.5723 |
| SVM Regression (standardized) kernel poly | 0.8557 | 11.5256 | 29.6124 | 31.1944 | 54.681 |
| kNN (k=3) | 0.9521 | 1.9341 | 16.6673 | 5.2347 | 30.7771 |
| kNN (k=5) | 0.9428 | 2.0005 | 18.3892 | 5.4143 | 33.9567 |
| kNN (k=7) | 0.9292 | 2.2402 | 20.3321 | 6.0631 | 37.5444 |
| kNN (k=9) | 0.918 | 2.4152 | 21.7399 | 6.5368 | 40.144 |
| Additive Regression (Decision Stump) | 0.8807 | 14.0449 | 25.8221 | 38.0132 | 47.6819 |
| Additive Regression (Random Forest) | 0.9707 | 1.4864 | 13.0059 | 4.023 | 24.0161 |
| Additive Regression (Random Tree) | 0.9265 | 2.3986 | 20.4576 | 6.4918 | 37.7762 |
| RepTree | 0.9024 | 4.7016 | 12.725 | 43.0811 | 34.7261 |
| Random Forest | 0.963 | 2.0568 | 14.6108 | 5.5667 | 26.9797 |
| Random Tree | 0.9401 | 3.0304 | 18.9886 | 8.2018 | 35.0636 |
| Multi-Layer Perceptron | 0.958 | 8.0572 | 15.5944 | 21.807 | 28.7959 |

## 4.3 Model Validation

The predicted results from the ML model are compared with the CFD modeling and experimental results for the melt-pool width and depth [2]. The CFD simulation of the L-PBF process with Ti-6Al-4V powder-bed material is conducted in ANSYS 2019 R3 [25]. The laser melting experiments are conducted in a custom-designed ytterbium fiber laser processing system (IPG model: YLR-200-AC-Y11) [35]. The cross-sections of the processed specimens are examined using scanning electron microscope (SEM) to obtain melt-pool width and depth results. The process parameters, i.e., the five features are kept same for both the experiment and the CFD model in order to facilitate the comparative study. The values for the simulation and experimental parameters and the five features considered for the ML model are shown in Table 4.9, where UDF represent user defined functions in terms of temperature [2, 25, 35].

Table 4.9: List of Simulation and Experimental Parameters [35]

| **Parameters** | **Values** |
|---|---|
| Solidus temperature, $T_S$ (K) | 1878 |
| Liquidus temperature, $T_L$ (K) | 1938 |
| Latent heat of fusion, $L_f$ (kJ/kg) | 440 |
| Spot size of laser beam, $\Phi$ (μm) | 100 |
| Scanning speed, $v_s$ (mm/s) | 300 |
| Laser power, $P$ (W) | 200 |
| Initial temperature, $T_{in}$ (K) | 298 |
| Laser absorption efficiency, $\eta_l$ | 0.865 |
| Powder porosity (%) | 50 |
| Powder layer thickness, $l_t$ (mm) | 0.07 |
| Beam penetration depth, S (μm) | 62 |
| Convection coefficient, $h$ (W/m$^2$-K) | 10 |
| Effective viscosity, $\mu$ (kg/m-s) | UDF |
| Specific heat, $c_p$ (J/kg-K) | UDF |
| Thermal conductivity, $k$ (W/m-K) | UDF |
| Emissivity, $\varepsilon$ | UDF |
| Density, $\rho$ (kg/m$^3$) | UDF |

The comparison of the results and percentage of deviation are depicted in Table 4.10 and Table 4.11. As a whole, the larger data set yielded a maximum error of 11.30% while the smaller data set yielded a maximum error of 11.57% confirming that a larger data set model is more accurate. However, because the experimental and CFD model values vary greatly, the smaller data set produces better results in some individual comparisons of the CFD and experimental

models. For example, the smaller data model predicted the exact depth of the CFD model but produced 11.57% error when compared to the experimental model. Overall, the machine learning model results for melt-pool width and depth show a good agreement with the CFD modeling and experimental results. The ML model can save a significant amount of time while predicting the melt-pool geometry as compared to the other two techniques.

Table 4.10: Results Comparison and Validation for Smaller Data Set

| Variable | ML | CFD Model | Exp.* | % of Deviation from CFD Model | % of Deviation From Exp.* |
|---|---|---|---|---|---|
| Melt-Pool Width | 201 | 186 | 205 | 8.06 | 1.95 |
| Melt-Pool Depth | 168 | 168 | 190 | 0 | 11.57 |

Table 4.11: Results Comparison and Validation for Larger Data Set

| Variable | ML | CFD Model | Exp.* | % of Deviation from CFD Model | % of Deviation From Exp.* |
|---|---|---|---|---|---|
| Melt-Pool Width | 192 | 186 | 205 | 3.22 | 6.34 |
| Melt-Pool Depth | 187 | 168 | 190 | 11.30 | 1.58 |

# Chapter 5

# Conclusion and Future Work

## 5.1 Concluding Comments

A supervised machine learning (ML) model is developed to predict the melt-pool width and depth of Ti-6Al-4V alloy in the laser powder-bed fusion (L-PBF) process. Five-fold and ten-fold cross-validation methods were compared in order to show which model produced more accurate results. A smaller data set consisting of 667 data points was tested against a larger data set consisting of 2922 data points to confirm that a larger data set will yield more accurate results. The Additive Regression Random Forest 10-fold cross-validation model generated with the larger data set produced the most accurate results with a maximum error of 11.30%. The overall study shows that the ML model can be a very good tool to generate faster yet reliable results for melt-pool geometry and similar parameters in the L-PBF process.

## 5.2 Future Work

Future work for this research includes improving the accuracy of the model. This can be achieved in numerous methods including: finding data that has more varied porosity and layer thickness values in order to create a more robust model, increasing the data set even further, focusing on more than five parameters, creating a neural network with images of melt-pool geometries that could be even more accurate than the regression models.

# References


[1] Rahman, M. S., Schilling, P. J., Herrington, P. D., and Chakravarty, U. K., 2021, "A Comparison of the Thermo-Fluid Properties of Ti-6Al-4V Melt Pools Formed by Laser and Electron-Beam Powder-Bed Fusion Processes," ASME J. Eng. Mater. Technol., 143(2), pp. 021003:1–13.

[2] Ciaccio, J., Rahman, M. S., Chakravarty, U., 2021, "A Machine Learning Approach for Predicting Melt-pool Dynamics of Ti-6Al-4V Alloy in the Laser Powder-Bed Fusion Process," 2021 ASME International Mechanical Engineering Congress and Exposition, 2021, pp. 1–9.

[3] Wu, D., Wei, Y., Terpenny, J., 2018, "Surface Roughness Prediction in Additive Manufacturing Using Machine Learning," 2018 ASME 13th International Manufacturing Science and Engineering Conference, 2018, pp. 1–6.

[4] http://www.lgcnsblog.com/inside-it/be-a-designer-yourself-with-a-3d-printer-2

[5] Demir, A., Mazzoleni, L., Caprio, L., Pacher, M., Previtali, B., 2019, "Complementary Use of Pulsed and Continuous Wave Emission Modes to Stabilize Melt-pool Geometry in Laser Powder Bed Fusion," Optics and Laser Technology, 113, pp. 15–26.

[6] Oliveira, J. P., LaLaonde, A., Ma, J., 2018, "Processing Parameters in Laser Powder Bed Fusion Metal Additive Manufacturing," Materials and Design, 193, pp. 1-12.

[7] Gong, H., Gu, H., Zeng, K., Dilip, J.J.S., Pal, D., and Stucker, B., 2017, "Melt-pool Characterization for Selective Laser Melting of Ti-6Al-4V Pre-alloyed Powder," Proceedings of the SFF, August, 2014, pp. 256−267.

[8] Dilip, J. J. S., Zhang, S., Teng, C., Zeng, K., Robinson, C., Pal, D., and Stucker, B., 2017, "Influence of Processing Parameters on the Evolution of Melt Pool, Porosity, and Microstructures in Ti-6Al-4V Alloy Parts Fabricated by Selective Laser Melting," Prog. Addit. Manuf., 2, pp. 157−167.

[9] Grierson, D., Rennie, A., Quayle, S., 2021, "Machine Learning for Additive Manufacturing," Encyclopedia MDPI, pp. 576–588.

[10] Padakandla, S., 2021, "A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments," ACM Computing Survey, 54(6), pp. 1–27.

[11] Gupta, A., Sharma, A., Goel, A., 2017, "Review of Regression Analysis Models," IJERT, 6(8), pp. 58−61.

[12] Maulud, D. H., and Abdulazeez, A. M., 2020, "A Review on Linear Regression Comprehensive in Machine Learning," JASTT, 1(4), pp. 140−147.

[13] Allamy, H.K. and Khan, R.Z., 2014, “Methods to Avoid Over-fitting and Under-fitting in Supervised Machine Learning (Comparative Study),” Proceedings of the CSCID, Jan 10, 2014, pp. 163−172.

[14] Jain, A. K., Murty, M N., Flynn, P. J., 1999, “Data Clustering: A Review,” ACM Computing Surveys, 31(3), pp. 265–323.

[15] Jaiswal, V., Agarwal, J., 2012, “The Evolution of the Association Rules,” International Journal of Modeling and Optimization, 2(6), pp. 726–729.

[16] Yoseph, F., Heikkila, M., 2020, “A New Approach for Association Rules Mining Using Computational and Artificial Intelligence,” Journal of Intelligent and Fuzzy Systems, pp. 1–14.

[17] Mohammed, M., Chung, K., Chyi, C., 2020, “Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations,” IEEE Access, 8, pp. 57–72.

[18] Staddon, J., 2020, “The Dynamics of Behavior: Review of Sutton and Barto: Reinforcement Learning: An Introduction (2nd ed.),” Journal of Experimental Analysis of Behavior, pp. 1–7.

[19] Mondal, S., Gwynn, D., Ray, A., and Basak, A., 2020, “Investigation of Melt-pool Geometry Control in Additive Manufacturing Using Hybrid Modeling,” Metals, 10(5), 683, pp. 1−23.

[20] Yang, Z., Lu, Y., Yeung, H., and Krishnamurty, S., 2019, “Investigation of Deep Learning for Real-time Melt-pool Classification in Additive Manufacturing,” 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), 2019, pp. 640−647.

[21] Lee, S., Peng, J., Shin, D., Choi, Y. S., 2019, “Data Analytics Approach for Melt-pool Geometries in Metal Additive Manufacturing,” Sci Technol Adv Mater, 20(1), pp. 972−978.

[22] Scime, L. and Beuth, J., 2018, “Using Machine Learning to Identify In-Situ Melt-pool Signatures Indicative of Flaw Formation in a Laser Powder Bed Fusion Additive Manufacturing Process,” Addit. Manuf., 25, pp. 155–165.

[23] Kamath, C., and Fan, Y. J., 2018, “Regression with Small Data Sets: A Case Study using Code Surrogates in Additive Manufacturing,” Knowl. Inf. Syst., 57, pp. 475−493.

[24] Akbari, M., Saedodin, S., Panjehpour, A., Hassani, M., Afrand, M., Torkamany, M. J., 2016, “Numerical Simulation and Designing Artificial Neural Network for Estimating Melt-pool Geometry and Temperature Distribution in Laser Welding of Ti6Al4V Alloy,” Optik, 127(23), pp. 11161−11172.

[25] Rahman, M. S., Schilling, P. J., Herrington, P. D., and Chakravarty, U. K., 2019, "Thermofluid Properties of Ti-6Al-4V Melt-pool in Powder-Bed Electron Beam Additive Manufacturing." ASME. J. Eng. Mater. Technol. October 2019; 141(4): 041006.

[26] https://www.upmet.com/sites/default/files/datasheets/316-316l.pdf

[27] https://parts-badger.com/properties-of-grade-5-titanium/

[28] https://www.onlinemetals.com/en/buy/stainless-steel-sheet-plate-316-annealed

[29] https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931

[30] Wills, S., Underwood, C. J., Barrett, P. M., Lautenschlager, S., 2021, "Learning to See the Wood for the Trees: Machine Learning, Decision Trees, and the Classification of Isolated Theropod Teeth," Palaeontology, 64(1), pp. 75–99.

[31] Soylemez, E., 2018, "Modeling the Melt-pool of the Laser Sintered TI6AL4V Layers with Goldak's Double-ellipsoidal Heat Source," Proceedings of the SFF, Austin, TX, August 2018, pp. 1721−1736.

[32] Kusuma, C., 2016, "The Effect of Laser Power and Scan Speed on Melt-pool Characteristics of Pure Titanium and Ti-6Al-4V Alloy for Selective Laser Melting," Master's Thesis, Wright State University, Dayton, Ohio.

[33] https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b

[34] Niu, M., Li, Y., Wang, C., Han, K., 2018, "RFAmyloid: A Web Server for Predicting Amyloid Proteins," Int. J. Mol. Sci., 19, pp. 1–13.

[35] Rahman, M. S., Schilling, P. J., Herrington, P. D., and Chakravarty, U. K., 2020, "Heat Transfer and Melt-pool Evolution During Powder-Bed Fusion of Ti-6Al-4V Parts Under Various Laser Irradiation Conditions," Proc. ASME 2020 International Mechanical Engineering Congress and Exposition, Vol. 4, Advances in Aerospace Technology, Portland, OR, Nov 16–19, 2020, IMECE2020-23838, pp. 1−10.

# Vita

Mr. Jonathan Ciaccio was born and raised in New Orleans, Louisiana. He completed his Bachelor of Science in Petroleum Engineering from the Louisiana State University in May of 2016. Beginning in January of 2018, he worked in the oil and gas industry in Midland, Texas for two and half years after graduation. He continued his education at the University of New Orleans, Louisiana starting in August of 2020 and completed his Master of Science in Mechanical Engineering in December of 2021.