Predicting Porosity and Microstructure of 3D Printed Part Using Machine Learning

by

Priya Dhage

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Industrial and Systems Engineering)
in the University of Michigan-Dearborn
2020

Master's Thesis Committee:

Professor Pravansu S. Mohanty, Chair
Associate Professor German Reyes-Villanueva
Assistant Professor Tanjore V. Jayaraman

Acknowledgements

I want to express my sincere gratitude to my Professor Dr. Pravansu Mohanty for giving me this invaluable opportunity to explore the area of Additive Manufacturing and Machine learning and for his patience, motivation, and guidance throughout the master's program. I also would like to thank my parents Mr. Prakash N Dhage and Mrs. Pushpa P Dhage, for their love and support that motivated me throughout my education.

Besides my advisor, I would like to extend my sincere thanks to my teammates Ramcharan Palacode Visveswaran, Aniket C Jadhav, and Neeraj Karmarkar, without whom this study wouldn't be as constructive. I would also like to thank my friend Aishwary for his constant support.

Table of Contents

List of Tables

List of Figures

Abstract

Additive Manufacturing (AM) is characterized as building a 3-D object one layer at a time. Due to flexibility in design and functionality, additive manufacturing (AM) is an attractive technology for the manufacturing industry. Still, the lack of consistency in quality is one of the main limitations preventing the use of this process to produce end-use products. Current techniques in additive manufacturing face a significant challenge concerning various processing parameters, including scan speed/velocity, laser power, layer thickness, etc. which leads to the inconsistency of the quality of the printed products.

Therefore, this research focuses on change, especially on the monitoring and regulation of processes, and helps us predict the level of porosity in a 3D printed part and classify grain growth structure as equiaxed or columnar given the simulation data using state-of-the-art machine learning algorithms. The input parameters considered in this study that affects porosity and grain growth structure are energy density, gas atmosphere, powder particle size and shape, and overlap rate. The data for training machine learning models are collected using ANSYS Additive Manufacturing simulations. The total data collected for porosity prediction is 482 data points, and for the grain growth structure is 12,333 data points.

In order to predict the porosity and grain growth structure, a technique based on Artificial Intelligence (Machine learning) is suggested to make the necessary compensations to process monitoring and control, which will subsequently improve the quality of the final product.

A feed-forward ANN model is trained in this methodology using an error back-propagation algorithm to predict the porosity level. Also, different classification models such as Support Vector Machines, Meta-classifier classify the microstructure as columnar or equiaxed grains, resulting in part quality improvement. The Backpropagation Neural Network model for porosity prediction gave an accuracy of 100% while outperforming other models. The best results for microstructure prediction are achieved by Meta-classifier, K-Nearest Neighbor, and Random Forest classifier with 100% accuracy.

The findings in this study provide evidence and insight that Artificial intelligence and machine learning techniques can be used in the field of Additive Manufacturing for real-time process control and monitoring with the scope of implementation on a larger scale.

Chapter 1: Introduction

1.1 Motivation

Since the beginning of the year 1980, overcoming barriers in additive manufacturing has been a topic of interest. Devices and ideas have since been developed by many researchers to overcome the obstacles in additive manufacturing. Over the past decade, as companies become increasingly internationalized and globalized, real-time process control has been discussed by many researchers and continues to be a high-priority research goal. One such research includes UK industry research with organizations in the aerospace, defense, automotive, medical devices industry, and heavy machinery on barriers to progression in additive manufacturing. They identified eighteen barriers in additive manufacturing. Some of the obstacles included in-process monitoring, repeatability, quality, mechanical properties, cost, materials, etc. (Seale et al., 2018).

Sadly, metal parts' quality and repeatability still significantly hamper AM's widespread use as viable manufacturing processes, especially for industries with stringent quality standards for products, such as aerospace and healthcare sectors. Also, current additive manufacturing techniques face a significant challenge concerning numerous processing parameters, such as scanning speed, laser power, and layer thickness, which leads to the inconsistency of the quality of the printed products. Process control and monitoring must be implemented in real-time for overcoming the issue of quality and repeatability. To produce more reliable results, we need fast and efficient alternatives, such as the utilization of a machine learning platform (Tapia & Elwany, 2014).

Machine learning has increasingly become popular in additive manufacturing. It is the utilization of artificial intelligence (AI) framework, which extracts essential information from raw data and discover patterns to solve a complex problem. Machine learning centers around the advancement of computer programs intending to access and utilize data to learn independently and produce accurate results through experience. Currently, machine learning research has been used in various fields and businesses. The impact of machine learning has been felt extensively across a range of companies and industries, focusing on data-intensive issues, such as consumer services, control of logistics chains, etc. With a reliable training data set, the ML models detect a pattern in the training set and make inferences based on these patterns. On one side, the trained AI models can make predictions and select the best parameters for processing the model, and on the other hand, they can deal with real-time data to detect errors and defects in the process (Meng et al., 2020).

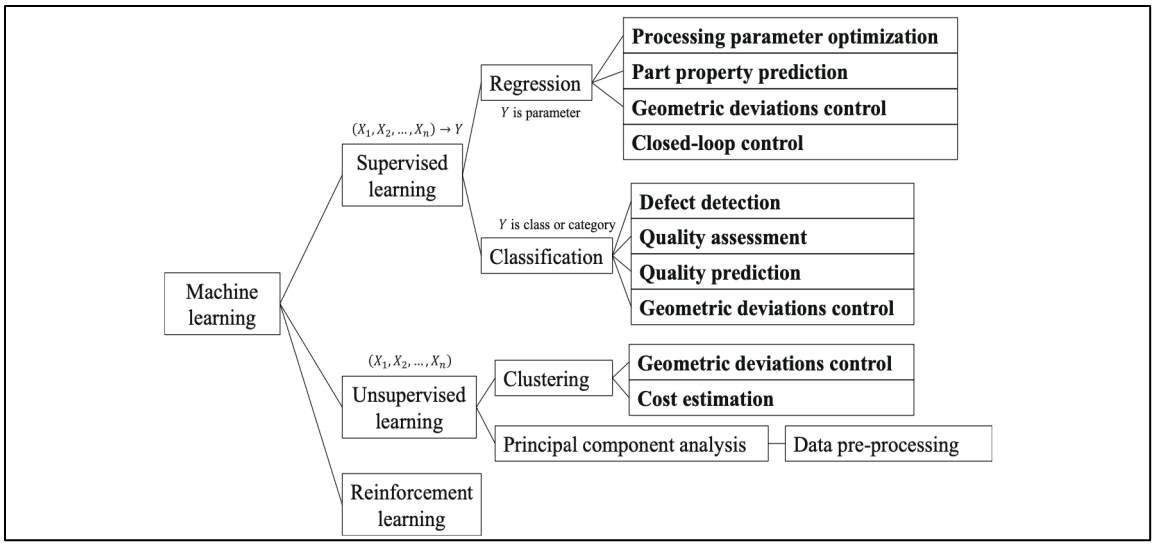

Figure 1: Distribution of ML Applications in AM field (Meng et al., 2020)

1.2 Additive Manufacturing

1.2.1 Overview

American Society for Testing and Materials (ASTM, 2013) defined Additive Manufacturing (AM) as “a process of joining mate- rials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies”. The object is built by removing material from a solid object until the final product is achieved. In the 1980s, Additive Manufacturing was mostly used to make prototypes, which were not functional, and the process was known as rapid prototyping. With improvements in Additive Manufacturing, its uses also expanded. In mid-2000s Additive Manufacturing was utilized to build functional objects. Recently companies like Nike, Ford, and General electric are using Additive manufacturing as a part of their business process.

American Society for Testing and Materials categorizes Additive Manufacturing into seven categories which are Binder Jetting, Directed Energy Deposition, Material Extrusion, Material Jetting, Powder Bed Fusion, Sheet Lamination and vat photopolymerization (ASTM, 2013).

The whole extent of this thesis is centered around these families, specifically, Powder Bed Fusion process. Powder Bed Fusion is a process in which heat source such as a laser or an electron beam is used to consolidate the powdered materials by melting them together.

1.2.2 Powder Bed Fusion

Additive Manufacturing Research Group mentioned in a study that Powder Bed Fusion is a process of building structures from powdered materials using lasers to selectively fuse or melt the particles in an enclosed chamber, layer by layer. The two different Powder Bed Fusion methods are Selective Laser Melting (SLM) and Selective Laser Sintering (SLS). These methods can be used with metals and alloys to create functional components (A. M. R. G.).

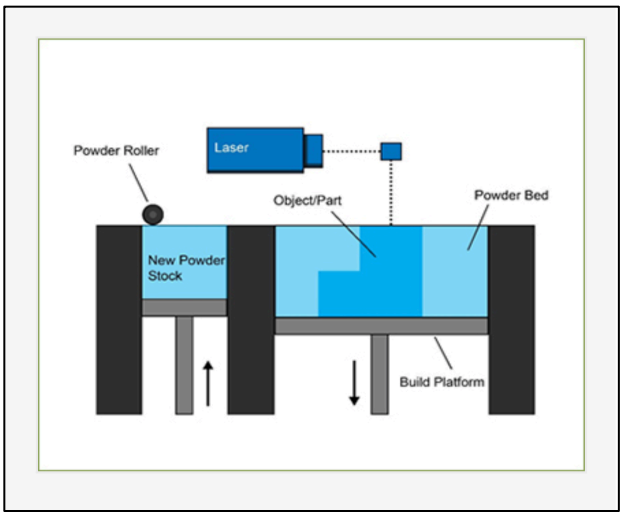

Figure 2: Process of Powder Bed Fusion (A. M. R. G.)

The spread of the powder material over the previous layers is generally involved in all PBF processes. To make this possible, there are different mechanisms, including a roller or a blade (Figure 2). The hopper or reservoir below the bed provides fresh supplies of material. Direct metal laser sintering (DMLS) is the same as SLS but with metals, not plastics. The cycle sinters, layer by layer, the material. Selective heat sintering differs from other methods by fusing powder content using a heated thermal print head. As before, layers are inserted in between fusion layers with a roller (A. M. R. G.).

Chapter 2: Purpose and Research Question

In order to control an additive manufactured part's quality, it is essential to have real-time process control and monitoring system. In this study, a technique of machine learning is used to characterize a knowledge discovery process to classify and predict grain growth structure/microstructure and the level of porosity in a 3D printed part. Also, an approach will be presented on how to validate the content of the results.

Two key research problems are being addressed in this study:

1. Predicting the level porosity using a machine learning technique in a 3D printed part, given the simulation data.
2. Using a machine learning technique, classifying the grain growth structure as equiaxed or columnar grains with respect to grain diameter.

Chapter 3: Review of Literature

The idea of using Machine learning/AI for process control and monitoring in Additive manufacturing isn't new. A study by Qi et al. (2019) mentioned the popularity of Additive manufacturing (AM) process which is increasing in the scholarly world and industry because of the one of a kind points of interest it has in correlation with conventional subtractive manufacturing. Also, it should be considered that the processing parameters in AM are hard to tune as they apply a tremendous effect on the printed microstructure and the performance of the resulting item. According to this study, the task to construct a procedure-structure–property–execution i.e., PSPP for AM utilizing conventional numerical and analytical models, is difficult. Today, the AI (ML) strategy is a substantial method to perform regression analysis and complex pattern recognition without building and fathom the underlying physical models (Qi et al., 2019).

Quality assurance and control are characterized as the most crucial test to the vast selection of AM technologies for metals parts in aviation industries by many researchers. Another way to deal with this problem is to execute in-situ inspection systems and process monitoring to improve the printed parts and AM techniques' quality. It has been featured by various research efforts for as far back as a decade and keeps on being prioritized research. A research proposed a comprehensive closed-loop monitoring system and real-time inspection method to address the quality control metal-based AM techniques (Chua et al., 2017).

Chen et al. (2017) conducted a study to explore bead shape, melt pool profiles and embraced a finite-element model. The simulations carried out change from the powder scale to the part scale and focus on just a couple of aspects of the entire procedure because of the absence of an in-depth comprehension of the AM process.

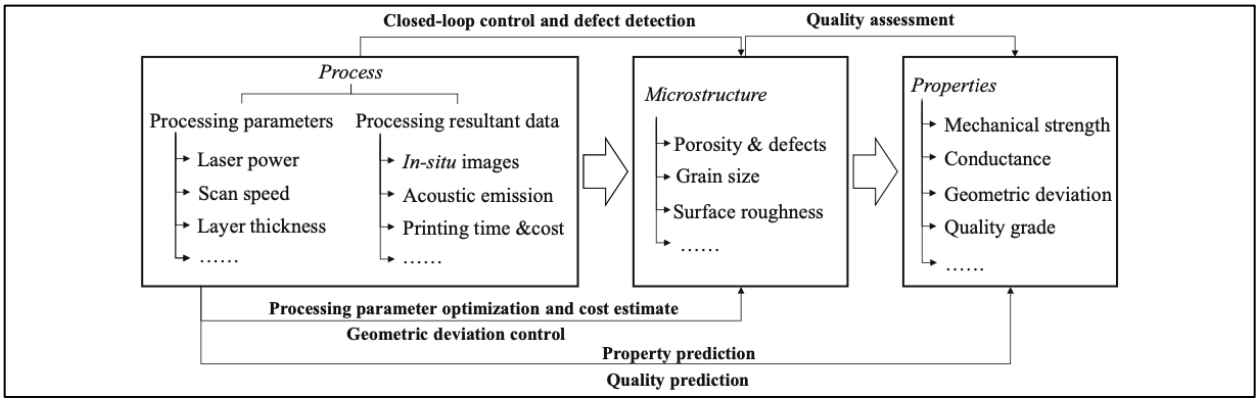

Figure 3: Process monitoring and control in Additive Manufacturing (Meng et al., 2020)

It is currently unreasonable to predict the entire AM process precisely and rapidly using these material science-driven techniques in a short time. Along with the previously mentioned material science-driven models, data-driven models have been broadly utilized in AM; these models have brought together the name of AI (Machine Learning).

Better quality control in AM techniques can be accomplished with the identification of principle correlation between the input i.e., process parameters and the output parameters, such as density and mechanical properties.

Categories	Process parameters
Material	Composition, Powder density, Morphology, Powder size, Distribution, Thermal properties, Flow properties, Melt pool viscosity, Material absorptivity
Laser	Mode, Wavelength, Power (Controllable), Frequency, Pulse width, Spot size (Controllable), Offset (Controllable), Intensity profile
Scan	Scan speed (Controllable), Scan pattern (Controllable), Layer thickness (Controllable), Scaling factors, Scan spacing (Controllable), Pulse distance
Environment	Inert gas type, Oxygen level (Controllable), Pressure (Controllable), Gas flow velocity (Controllable), Preheating temperature (Controllable)

Figure 4: Process parameters included in SLM process (Chua et al., 2017)

In an investigation, the effects of SLM parameters through Volumetric Energy Density (VED) on microstructure, mechanical properties, and porosity were presented. It concluded that for IN718 columnar dendritic microstructure was the dominating microstructure in SLM, parallel to the build direction. This microstructure may expand through a few layers and comprises columnar grains with a cellular structure and extremely fine dendrites. A few sorts of defects were also observed, including a lack of fusion pores and spherical pores. The range of degree of porosity differed somewhere between 0.48% and 1.35%, which mainly consisted of spherical pores. The increase in Volumetric Energy Density (VED) prompts a decrease in the degree of porosity (Moussaoui et al., 2018).

The impact of scan speed and laser power on the melt pool and the microstructure porosity was contemplated in a study. High-density IN 718 sections can be manufactured with L-PBF by picking the optimum scan speed, and laser power was the critical commitment of this investigation. A direct relationship of energy density with the melt pool dimensions was seen in the scope of 2 to 10 (J/mm^2). With the increase in laser power, a direct increment in porosity level was observed.

Thus, based on the thorough process parameters investigation, ideal laser power and scan speed to manufacture IN 718 effectively easily are presented in the study (Kumar et al., 2019).

Parimi et al. (2012) proposed in the research that an EBSD map was plotted on the lower layers of the bead to comprehend the distribution of grain size. There is a bimodal grain size populace over the bead. As per this research, in IN718 a single bead can be classified into three areas: Extremely fine equiaxed area in the bead limit with a mean size of ~ five μm diameter at the base layer, columnar zone from the base to the bead's center with a mean size of ~50-100 μm, with a ratio of 2-5 and coarse equiaxed to elongated grains only at the bead center with a mean size of ~10-20 μm, with a ratio of 1-2.

As for the grain size distribution, the largest grain size at the base layers was exceptionally huge ~ 400 μm size, contrasted with the other layers ~ 100-150 μm. The average grain size will, in general, increment from base to the highest point of the build.

Chapter 4: Material and Experimental Parameters

4.1 Material

There was a great need by designers for more corrosion-resistant materials and more durable materials for high-temperature applications. Stainless steel applied and developed in the 20th century filled in as a beginning stage to fulfill high-temperature engineering material necessities. They before long were seen as restricted in their strength abilities. Stainless varieties termed as ‘‘super-alloys’’ were made by the metallurgical community, which had increased needs.

A widely used nickel-based superalloy, IN718 has highly desirable properties, which are also consistent with a particular interest in aerospace and automotive. Nickel, iron-nickel, and cobalt-base alloys are superalloys that are generally used above temperatures 540 °F or 1000 °F, IN718 a popular iron-nickel-base superalloy is an expansion of stainless steel and for the most part, is wrought (Donachie & Donachie, 2002). Out of the different materials available commercially, Inconel 718 was chosen as the material due to its availability for porosity and microstructure simulation.

Laser Powder Bed Fusion (L-PBF) process uses a focused and restricted laser beam for melting the alloy particles and rapid solidification of the melt pool area layer by layer with an optimized pattern to achieve higher density 3D part. The 3D part made from L-PBF results from the making of micron-sized melt pools because of the rapid solidification of these melt pools and high energy localized laser irradiation.

4.2 Energy Density (E.D)

The melt pool attributes impact the quality of build for various materials, which has been generally reported in the literature. There is a decrease in overall processing efficiency by an increase in time due to the melt pool's small size (and depth). Interestingly, in large melt pool size, the overall processing efficiency is increased. Still, on the other hand, there is pores formation due to the vaporization of powder/substrate, which leads to an overall porosity increase in the material. Subsequently, the build's quality, which includes the overall surface roughness and the final density of part, is highly dependent upon the melt pool characteristics, mainly size and shape. On the other hand, the melt pool characteristics are controlled by the laser beam's energy density.

In the L-PBF process for a given material, the optimized and controlled energy density can be accomplished by controlling the predefined controllable boundaries. The parameters that are most important and related to energy density are scan speed (v), laser power (P), the layer thickness (t) and hatch distance (d). The most common formula for calculating the energy density is given as follows:

$$E.D = \frac{P}{V \times t \times d}$$

Generally, the diameter of the laser beam is fixed with energy distribution. Still, on the other hand, the parameters mentioned above can be adjusted all at once/individually to accomplish the ideal energy density. The final energy density achieved affects the melt pool characteristics, which affects the optimum density and microstructure of the finished part (Moussaoui et al., 2018).

4.3 Gas Atmosphere

Generally, another critical parameter in L-PBF that may add to the variety in material properties is identified with the particular kind of shielding gas utilized. When exposed to air at high temperatures, most of the metals tend to oxidize, due to which it becomes essential to use inert gas to prevent oxidization of the melt pool. Similar to the case in welding processes, which share some similarities in liquefying and hardening phenomenon in contrast with power regimes, the atmosphere under which a metal is handled through SLM can influence the heat transfer, science, and the melt pool characteristics of the material.

The kind of gas utilized might be a critical supporter of the expense of operations. For instance, pure nitrogen gas for nickel-based alloys may give a noteworthy cost sparing over argon gas SLM preparing, in light of the gas alone. On the contrary, nitrogen has increased solubility in the melt pool, which affects the alloy's composition and leads to an increase in porosity compared to argon gas. The utilization of argon or nitrogen gas can reduce the porosity level on solidification due to metal reactivity or interactions based on the alloy used. Also, it has been found that there is a direct correlation between the part density and the speed of inert gas flow (Bean et al., 2019).

There is an increase in the overall part density with a higher rate of flow of inert gas and also less variation in the local density. Based on the above findings, we have used argon gas as the inert gas atmosphere for our study; this gas atmosphere parameter is kept constant for the ML models.

4.4 Powder Particle Shape and Size

Another critical parameter taken into consideration in this study is material i.e., IN718 powder shape and size. The shape and size of inconel powder have a significant effect on printing and the parts final quality. A literature search uncovers that a lot of work has been done so far on the

process optimization to print Inconel powders effectively; but, a limited measure of exploration has been completed to consider the powder attributes, and especially those of Inconel powder, for AM. It is essential to know the powder particle size distribution in AM, as different strategies have their prerequisites. For instance, SLM as of now utilizes powders with powder sizes running from 15 μm to 63 μm whereas for electron beam melting the size varies from 45 μm to 105 μm, because of challenges such as effects of smoking and sparking (Nguyen et al., 2017).

According to ISO 13320-1 standard (International Organization for Standardization [ISO], 1999), the average powder particle size for SLM applications is around 31 μm. The powder particle size distribution directly affects the build quality, especially on the printed part's surface finish. A better surface finish of the 3D printed part can be achieved using a smaller average particle size. The factors that affect the quality of built parts are the range of particle size and the mean particle size. A wide particle size prompts an uneven layer spread, and high surface roughness is created of the printed part (Nguyen et al., 2017).

It was observed that there were upgrades in both mechanical properties and surface finish for the sample with the most reduced particle size. Also, it was seen that during the printing procedure that if the particle size is excessively large, then the laser power can't dissolve the biggest particle totally or will over-melt the little particles, prompting to pores formation and balling effects. Hence it is proposed that the optimum range of powder particle size should be between 15 μm to 63 μm or fall inside a much smaller range.

During powder raking, flowability is considered an essential parameter for powder as it helps in stimulating the process. Among the different shapes of particles, namely spherical, irregular and angular, just spherical shaped particles can give the highest flowability. Particles of such powders

have negligible contact regions with neighboring powder particles, thus making less internal friction and requiring the insignificant raking force to flow.

According to the American Society for Testing and Materials (ASTM) B213 standard, the virgin IN718 powder had a slightly higher flow rate than the recycled powder's flow rate. The reason behind this is that powder particles turned out to be somewhat distorted after some time and were not, at this point, spherical; also, a portion of the recycled powders likewise would, in general, stay together. Also, recycled powders may contact dampness during the powder recovery process, for instance, because of impacting, sieving, stockpiling, and reloading forms. High dampness, which can ascend to 90% in a tropical domain, is an extraordinary worry in a powder's qualities. It unequivocally influences a powder's flowability conduct during the powder-raking procedure, and along these lines brings about negative impacts on the printing of parts. This study used the powder shape as spherical, which is constant and varying powder size (Nguyen et al., 2017).

4.5 Overlap Rate

Melt pool abnormalities and balling happen at high output speeds; they couldn't be the main reason for the end of the set of process parameters because of the additional impact of the absence of overlap rate on the densification and consolidation the built part. It was also discovered that there is no single process parameter or combination of process parameters that will result in an optimized unique procedure boundary mix that will result in superior qualities (Balbaa et al., 2020).

4.6 Grain Size Diameter

A research found that dominating microstructure in L-PBF for IN718 is columnar grains in the build direction. An average large grain diameter of 44 μm grain growth is limited by rapid solidification, bringing about irregular small grains on the outside of columnar grains. A

microstructure overwhelmed by columnar grains results from rapid solidification and directional growth in the X–Z plane, which is in the build direction with an average small grain diameter of 10.3 μm and unpredictable grains in the X–Y plane. It was noted in the research that the for wrought IN718 in the X–Y plane exhibited equiaxed grains of an average diameter of 40.7 μm. According to ASTM grain size 11, the average grain diameter of 7.9 μm is found in L-PBF specimens for IN718, which started with much small and finer grain size. For L-PBF, the as-built specimen had a mix of equiaxed grains with an average grain diameter of fewer than 10.3 μm (Newell et al., 2019).

Chapter 5: Theoretical Background

This section presents an overview of the relevant theoretical bases that are answered in the sense of the experiment's framework:

• Data mining, and its benefits.

• Machine learning.

• Promising algorithms in the field of Machine Learning.

• Hyperparameter tuning, and different hyperparameters in Machine Learning.

5.1 Data Mining

Data mining is where the appropriate approach and algorithm are selected and applied to the data set. Data mining involves taking any form of data and applying analytical algorithms to reveal models or patterns within the data set and using those structures to classify the data into different classes (labels). Several research areas are included in data mining, mainly database systems, statistics, and pattern recognition. Data mining tasks are classified according to the data set information that the algorithm has about the existing classes. It is further divided into two categories of supervised and unsupervised learning.

5.1.1 Supervised Learning

Supervised learning has labeled dataset for training. The training dataset has input values, and corresponding output values used to train the ML model, which infers the functional relationship between them. Supervised learning can be utilized for regression, classification, and ensemble learning.

5.1.2 Unsupervised Learning

Unsupervised learning doesn't have a labeled dataset. The ML algorithm itself tries to find the relation between the training dataset. Grouping parameters cluster the dataset, and the output/target class is identified based on it. Unsupervised learning can be utilized clustering, dimensionality reduction, and association. For detecting abnormal conditions, unsupervised learning is useful. The use of supervised and unsupervised learning will rely upon apparent advantages for a given situation.

As the proposed task is a question of binary classification concerning the two recognized classes of skilled translation and predictive model for predicting continuous data, this work will concentrate on supervised learning methods.

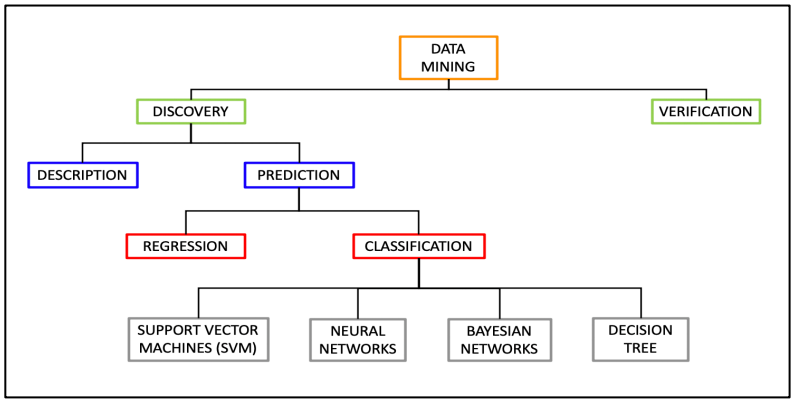

Figure 5: Classification of Data Mining Techniques

5.2 Machine Learning

Machine learning concerns the development and investigation of frameworks that can learn patterns automatically from the data information. Machine learning models can be used for defect detection, prediction, classification, regression, or forecasting. The most crucial factor that helps decide the effectiveness of an ML model is the data used, as ML models are just on a par with the data for training that has set them up to be.

There's something in machine learning called the "No Free Lunch" hypothesis. In short, it claims that almost no algorithm fits better for any problem and is particularly important in case of supervised learning i.e., predictive modeling.

You cannot say, for instance, that neural networks are often better than decision trees, or vice versa. There are a lot of variables at stake, including the dataset size and structure. As a consequence, several different algorithms should be tried for the problem, when using a hold-out "testing set" of data to evaluate output and pick winner.

The ML algorithms included in this research are (1) Support Vector Machines (SVM), (2) Naïve Bayes, (3) K-Nearest Neighbors algorithm (KNN), (4) Neural Networks- Backpropagation Neural Networks, (5) Stacking Classifiers and (6) Random Forest Classifier.

5.2.1 Support Vector Machines (SVM)

The goal of the Support vector machine is to discover a hyperplane in N-dimensional space (N - the number of features) that mainly classifies the given data points. To distinguish two classes in the data, many ways/hyperplanes could be used, but the main objective is to find a plane with a maximum margin, which is the distance between both classes' data points.

Thus, under the following two assumptions, SVMs can be described as linear classifiers:

1. The margin must be as large as possible.
2. Support vectors are the most important data points, since they are the ones that tend to be labeled incorrectly.

Figure 6 shows the maximum margin and also the support vector data points which are important assumptions in Support Vector Machines.

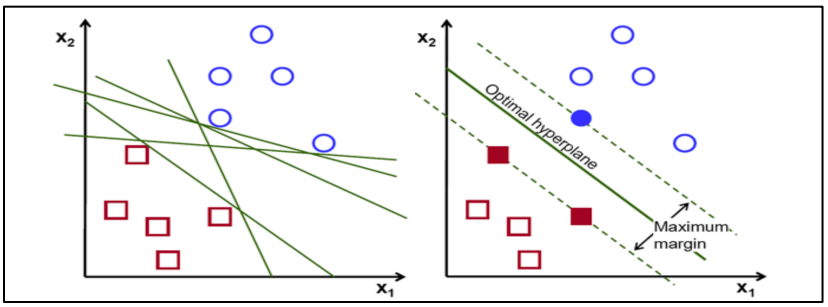

Figure 6: Support Vector Machine with different hyperplanes (Gandhi, 2018)

Hyperplane can be defined as the decision boundary between the two classes that help distinguish the data. Data points on both sides of the hyperplane indicate different classes. The number/dimensions of hyperplanes depend on the number of features in the data points. There will be only one-dimensional hyperplane for two input features and two-dimensional hyperplane for three input features (Gandhi, 2018).

5.2.2 Naïve Bayes

Naïve Bayes Theorem is based on calculating the probability of an event will occur, given that already another event occurred. It gives a way to calculate the probability of a hypothesis given the prior knowledge we have. This algorithm is usually used in two-class or multi-class classification of given data. It is called 'naive' because it is based on the assumption that all the variables are independent of each other; this is a naïve assumption in real-world examples.

Bayes' Theorem is given as:

$P(h|d) = (P(d|h) * P(h)) / P(d)$

Where,

P(h|d) is the probability of hypothesis h given the data d, known as the posterior probability.

P(d|h) is the probability of data d given that the hypothesis h was true.

P(h) is the probability of hypothesis h being true (regardless of the data), known as the prior probability of h.

P(d) is the probability of the data (regardless of the hypothesis).

Here, we are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h).

Naïve Bayes algorithm is used for calculating (1) Class probabilities and (2) Conditional probabilities. In this study, we have used only class probabilities to classify two classes. Class probability is defined as the probabilities of each class in the training dataset. In simple terms, it the frequency of instances belonging to each category divided by total instances (Yiu, 2019).

5.2.3 K-Nearest Neighbors Algorithm (KNN)

K-Nearest Neighbors (K-NN) is a supervised machine learning algorithm used for regression and classification problems. K-Nearest Neighbors assumes that similar things are close to each other. K-NN is often termed as a non-parametric algorithm which implies that in order to implement K-NN, there are no assumptions that must be met. On the other hand, parametric models such as linear regression have many assumptions that data must satisfy before it could be implemented which isn't always the case with K-NN.

Since it's an instance-based teaching and learning method; K-NN is a memory based systematic technique. The classification model adapts instantly as we gather new data for the training. It enables the algorithm to rapidly adapt to input changes during real-time use. One amongst K-NN's biggest benefits is that K-NN could be used for either regression or classification problems. One of the major drawbacks of using K-NN is that it doesn’t perform well on imbalanced data. Figure 7 shows that similar data points are close to each other. K-NN Algorithm hinges on similarity and

some mathematics to find distance between the points in the dataset. The popular method to calculate the distance between the two points is Euclidean distance or the straight-line distance.

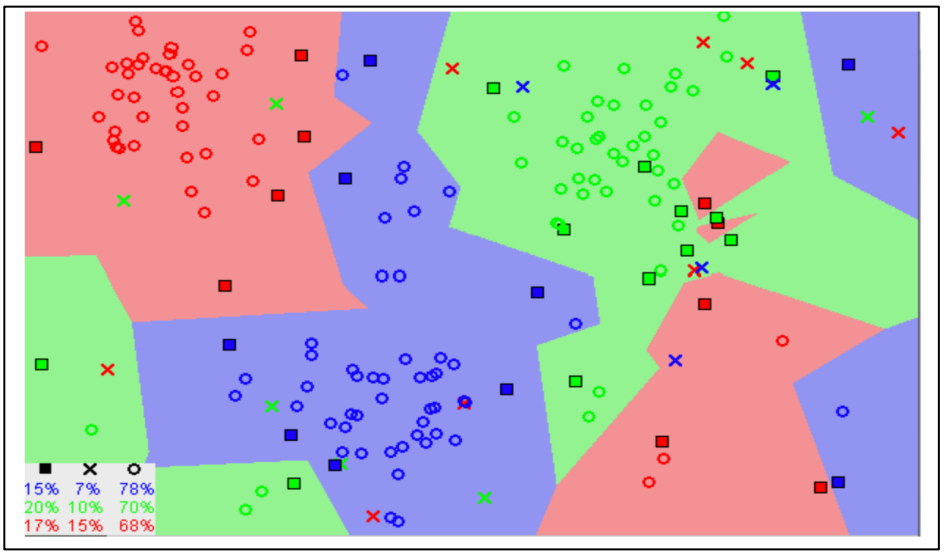

Figure 7: K-NN different groups/clusters formed using nearest neighbor (Yiu, 2019)

K-NN algorithm uses all the data for training, rather than splitting the data into training and validation dataset. When an output is needed for a new data point, the algorithm calculates the nearest neighbor for that new data point by going through all the data instances and then outputs the instances' mean. The selection of K is user specified in the algorithm (Yiu, 2019).

5.2.4 Neural Networks- Backpropagation Neural Networks

Neural networks can be made from three layers of neurons: the input layer, the hidden layers, and the last layer- the output layer. The hidden layer or layers in the middle comprises numerous neurons, with associations between the layers. As the neural network "learns" from the data, the weights of these neurons' associations are fine-tuned, permitting the system to think of exact predictions.

Neural networks usually depict the way our brain works, more specifically the way it represents information. Neural networks that have many hidden layers in it is known as deep neural networks.

The most fundamental building block in a neural network is the Backpropagation algorithm. The chain rule is a method used to train this model effectively. In the Backpropagation algorithm, after each forward pass through the layers, a backward pass is performed by backpropagation, which adjusts the model's weights and biases (Simpson, 2018).

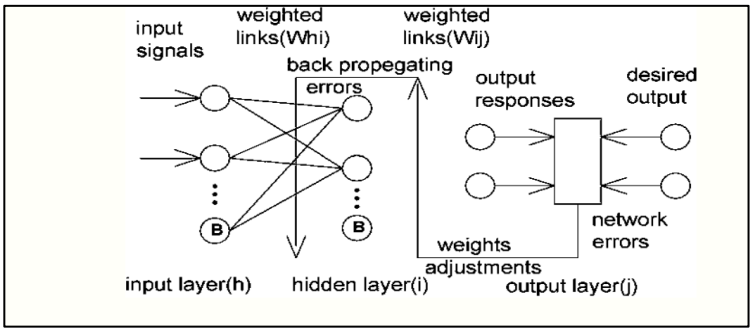

Figure 8: Working of a Backpropagation Neural Network (Simpson, 2018)

The forward pass proliferates the input vector through the system to give output at the last/output layer. The backward pass is like the forward pass; then again, an error is backpropagated through the system to decide how the weights are to be changed during training the system. During the backward pass, the value goes along the weighted association in the reverse course to that which was taken during the forward pass. Figure 8 shows a backpropagation system. A unit in the hidden layer will send the initiation/activation to each unit in the final layer during the forward pass. Thus, during the backward pass, a unit in the hidden layer will get error signals from each unit in the final layer.

The working of a backpropagation neural network is as follows:

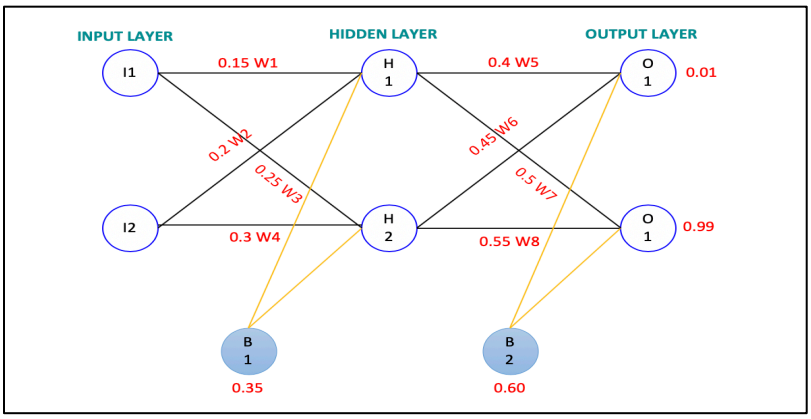

Figure 9: Architecture of Backpropagation Neural Network

I_1: Input layer neuron 1,

I_2: Input layer neuron 2,

H_1: Hidden layer neuron 1,

H_2: Hidden layer neuron 2,

O_1: Output layer neuron 1,

O_2: Output layer neuron 2,

B_1: Bias 1,

B_2: Bias 2.

Figure 9 represents a backpropagation neural network. The first step in this includes choosing the right activation function, which determines each neuron/node's activation value for the whole network. The activation function used in this example is sigmoid, which is commonly used and represented as:

$$\text{Activation function Sigmoid} = \frac{1}{1 + e^{-x}}$$

The next step is to forward propagate information from one layer of the network to order for which we have to calculate the weighted sum of inputs for H_1 function.

$$H_1 = I_1.W_1 + I_2.W_2 + B_1,$$

The value for H_1 is calculated and then passed through the activation function where we calculate the function value of net H_1 represented as follows:

$$net\ H_1 = \frac{1}{1 + e^{-H_1}}$$

The same process is used to calculate all other units represented as:

$$H_2 = I_1.W_3 + I_2.W_4 + B_1,$$

$$net\ H_2 = \frac{1}{1 + e^{-H_2}}$$

$$O_1 = net\ H_1.W_5 + net\ H_2.W_6 + B_2,$$

$$net\ O_1 = \frac{1}{1 + e^{-O_1}}$$

$$O_2 = net\ H_1.W_7 + net\ H_2.W_8 + B_2,$$

$$net\ O_2 = \frac{1}{1 + e^{-O_2}}$$

And then passed through activation function. After calculating the $net\ O_1, net\ O_2$ the total error for the network is the sum of E_{O_1} and E_{O_2} is calculated using the formula below:

$E_{O_1} = \frac{1}{2}\ (0.01 - net\ O_1)^2$,

$E_{O_2} = \frac{1}{2}\ (0.01 - net\ O_2)^2$

Total error, $E_{Total} = E_{O_1} + E_{O_2}$

When the model does not predict accurate results, we consider that the weights are not tuned properly, and we backpropagate the total error backward in such a way that the weights are fine-tuned. Now for backward propagation of this error, we calculate the partial derivative concerning weight using chain rule represented as:

$$\frac{\partial E_{Total}}{\partial W_5} = \frac{\partial E_{Total}}{\partial net\ O_1} + \frac{\partial net\ O_1}{\partial O_1} + \frac{\partial O_1}{\partial W_5}$$

To calculate and update the final weight W_5 we need to subtract the value obtained from $\frac{\partial E_{Total}}{\partial W_5}$ with the current weight and multiply the learning rate η represented as:

$$W_5^* = W_5 - \eta \times \frac{\partial E_{Total}}{\partial W_5}$$

Similarly, the whole network's weights are updated/backpropagated one by one and then again forward propagated to see the network's predictions. This process needs a lot of such iterations before making results closer to the expected/target outputs (Goodfellow, 2016).

5.2.5 Random Forest Classifier

Like its name infers, random forest comprises countless individual decision trees that work as an ensemble. Every particular tree in the random forest lets out a class prediction, and the class with the most votes turns into our model's prediction (see Figure 10).

The original idea of driving random forest is a straightforward yet amazing one - the wisdom of crowds.

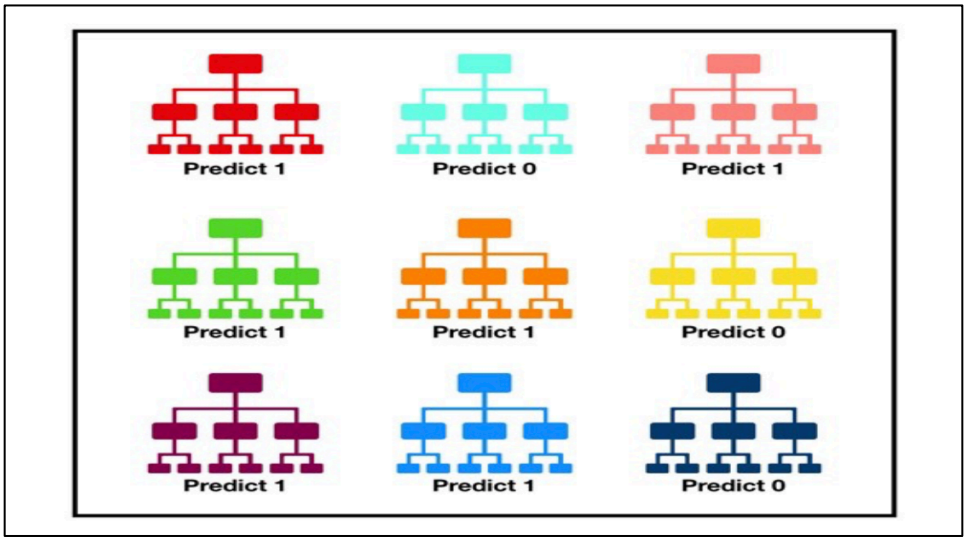

Figure 10: Predictions made by different Decision Trees in Random Forest Classifier (Yiu, 2019)

The explanation that the random forest model works so well in data science is:

Many uncorrelated models (trees) working as a committee will outperform any of the individual models. The low relationship between models is the key. The same as how ventures with little associations (like stocks and securities) meet up to frame a more prominent portfolio than the sum of its parts, uncorrelated models can deliver ensemble predictions that are more precise in comparison to any of the individual predictions. This impact is that the trees shield each other from their mistakes as long as they don't continue all fail a similar way (Yiu, 2019).

5.2.6 Stacking Classifiers

The least complicated type of stacking can be depicted as an ensemble learning method where the predictions of multiple different classifiers (alluded as level-one classifiers) are utilized as new features to train a meta-classifier on a single dataset. The selection of meta-classifier can be any classifier of your decision. Figure 11 shows three unique classifiers, C1, C2, and C3, that gets trained. The predictions made by them get stacked and are utilized as features to train the meta-classifier, making the last prediction (Ceballos, 2019).

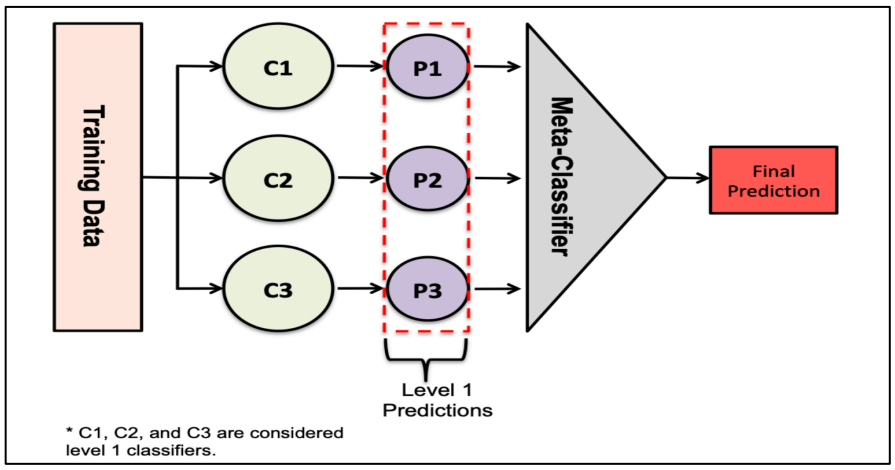

Figure 11: Architecture of Stacking Classifier (Ceballos, 2019)

As opposed to stacking, no learning happens at the meta-level when combining classifiers by a voting plan (for example, majority, probabilistic or weighted voting). The voting plan continues as before for all unique training sets and learning calculations (or base-level classifiers). The least complicated voting plan is the majority vote. As indicated by this voting plan, each base-level classifier chooses its predictions. The example is ordered in the class that gathers the most votes (Dzeroski & Bernard, 2004).

The classifiers used in this study for stacking are the K-Nearest Neighbors algorithm (KNN), Naïve Bayes, and Random Forest Classifier.

5.3 SMOTE Analysis

Class prediction or classification aims to build a rule based on a set of samples with known class label from training set that can be used to classify appropriate samples to the class label. There are several different classification algorithms/classifiers and they are based on the values determined for each sample by the variables (features). The training data are most frequently class-imbalanced: the volume of findings relating to each class is not equal. The problem of training ML Models from class-imbalanced data has become increasingly attentive in many fields.

The emergence of a class imbalance has major learning implications on model performance, generally generating classifiers which have low predictive accuracy for the minority class and fail to classify the majority of new samples. Classification with class imbalanced data is skewed toward the dominant class. The imbalance is much greater toward high-dimensional data, where the number of parameters is greater than the number of samples considerably. Under sampling or over-sampling, which produces class-balanced results, may mitigate the problem. Under-sampling is usually beneficial while random oversampling is not. The most common form of oversampling

method proposed to enhance random oversampling is Synthetic Minority Oversampling Technique (SMOTE). SMOTE is a technique that focuses on the nearest neighbors measured by the Euclidean Distance in feature space between the data points (Blagus & Lusa, 2013).

5.4 Hyperparameter Tuning

The ideal approach to consider hyperparameters is the resemblance of settings of an algorithm/model that can be changed by performance optimization, just like turning the handles of an AM radio to get an unmistakable signal. For the creation of an AI/ML model, choices are given to characterize the model's architecture. Generally, one doesn't promptly know what the ideal parameters might be for a given model. Along these lines, one has to prefer to have the option to investigate a scope of conceivable outcomes.

Models can have numerous hyperparameters and finding the best parameters that can be treated as a problem. A model hyperparameter is an arrangement outside to the model and whose worth can't be assessed from data. They are utilized in procedures to assist in estimating model's parameters, usually specified by the user, and regularly tuned for a problem in the predictive model.

The best optimal parameters for a model cannot be known beforehand. Rules of thumb can be used to duplicate hyperparameters that were used for other issues, or the best values for parameters can be searched using trial and error method. When an AI/ML model is tuned for a particular issue, you are basically tuning the hyperparameters of the model to find the parameters of the model that bring about the best parameters. The hyperparameters that are tuned in this study are as follows:

5.4.1 Learning Rate and Gradient Descent

A stochastic gradient descent algorithm is used to train neural networks. It is an optimization technique that evaluates the gradient error for a model's current state utilizing data points from the training dataset and then backpropagates the algorithm's error by updating the weights. It is known as backpropagation in simple terms.

The weights amount, which is updated in the training period, is known as the 'learning rate or step size.' In particular, the learning rate is a configurable hyperparameter utilized in the neural network for training, which has a positive value (small), ranging between 0.0 and 1.0.

The learning rate controls how rapidly the model is adjusted to the problem. Small values of learning rates require all the more epochs for training given the little changes made to the weights with each update, though high learning rates bring about fast changes and require fewer epochs for training. A learning rate that is too huge can make the model converge excessively fast to a solution, which is suboptimal, though a learning rate that is too little can cause the procedure to stall out (Brownlee, 2019).

5.4.2 Optimizer

Optimizers are strategies or algorithms used to update/change the neural network parameters, such as learning rates and weights, to lessen the error function. The choice of optimizer determines how the weights and learning should be updated in a neural network. Optimizer algorithms or methods are liable for reducing the error function and give the most precise outcome possible.

Stochastic Gradient Descent is the optimizer used in this study, which is a variation of Gradient Descent. It attempts to update the parameters of the model more frequently. In this, the model

parameters are changed after the calculation of the error function on each training step. If the dataset contains 100 data points, SGD will refresh the model parameters multiple times (100) in a single cycle of dataset rather than just one-time update as in Gradient Descent (Doshi, 2019).

5.4.3 Number of Epochs and Batch Size

An Epoch is when an entire dataset is passed forward and in reverse through the neural network just a single time. One epoch is too enormous to take care of to the computer once at a time, so it is partitioned into a few small batches. An increase in the number of epochs leads to an increase in the update of weights in a neural network, which helps the curve go from underfitting to ideal and further towards overfitting.

The number of training data points present in a single batch is known as batch size. The entire dataset cannot be passed at once through the neural network; therefore, it is divided into several batches (Sharma, 2017b).

5.4.4 Activation Function

Activation function (also known as transfer function) is mainly used to get output from a node. It determines the neural network's output in terms of yes or no and maps the final values in a range of 0 to 1 or -1 to 1. The activation function used in this study is ReLU, also known as the rectified linear unit. It is mostly used in deep neural networks and convolutional neural networks. ReLU considers linearity for positive values and zeroes for negative values. In simple terms, it returns an amount ranging between 0 to input value for an input value.

ReLU function has replaced many other activation functions because it accelerates the speed of training. It reduces the overall computation time of training a neural network (Sharma, 2017a).

Chapter 6: Experiments and Simulations

As discussed in the literature and the idea of this study, the next step is to generate data from simulations for ML models. Meaningful data gathering is an essential step for ML models as we want it to learn useful information and make predictions based on that data. The simulations were run concerning the PBF process on ANSYS Additive Manufacturing Suite. To start the simulation geometry of a simple 3*3*1 (mm) wall was used. Out of the different materials available commercially, Inconel 718 was chosen as the material due to its availability for porosity and microstructure simulation.

ANSYS Additive suite software is dedicated to additive manufacturing, which consists of (1) Additive Print and (2) Additive Science. Additive Print is a tool to perform quick simulations to know that a part will print successfully. On the other hand, Additive Science is used to understand the machine's optimum parameters and different materials.

In this study, we used Additive Science, which simulates the Laser Powder Bed Fusion (L-PBF) process, i.e., the layer-by-layer building of metal parts. There are four types of simulations available in Additive Science, namely Thermal History Simulation, Microstructure (Thermal History and Microstructure simulation types are Beta at Release 2019 R2), Single Bead Simulation, and Porosity Simulation. This study used microstructure simulations, and porosity simulations result for training ML models.

The porosity simulations help to get information on the level of porosity in an additively manufactured part. As there are many layers simulated in porosity simulation, we can choose to change the input machine configuration parameters.

Figure 12 below shows the list of machine parameters used in porosity simulation. Along with the machine configuration, we need to specify the geometry of the part. The valid input values for length, height, and width of the geometry is between 1and 10 (mm). This set of input parameters gives us a lack of fusion porosity in terms of solid ratio as the output (ANSYS, Inc. Additive User's Guide, 2019)

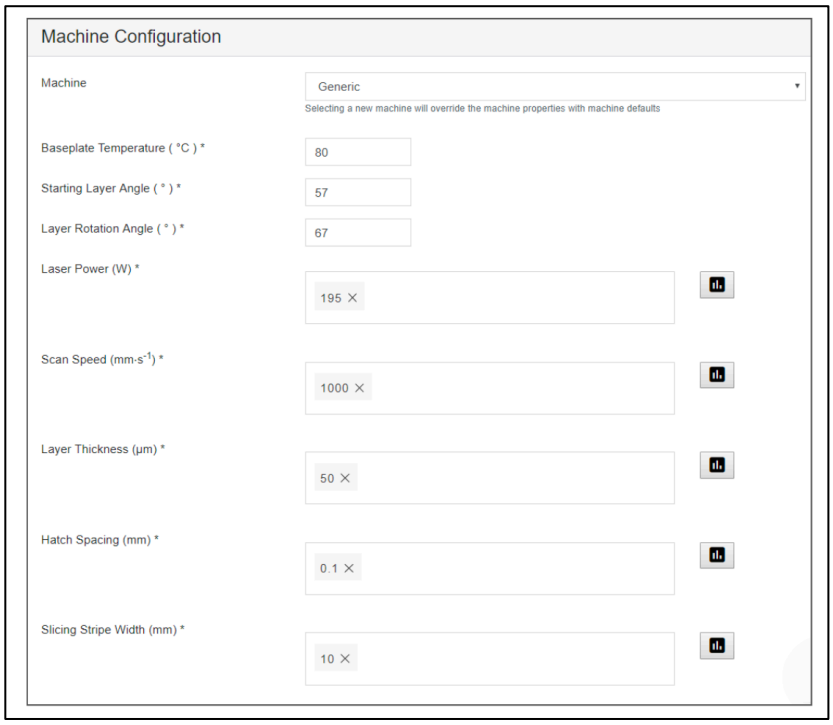

Figure 12: Machine configuration inputs in ANSYS Additive Manufacturing Suite (ANSYS, Inc. Additive User's Guide, 2019)

For performing microstructure simulations, the same inputs for machine configuration and geometry are used along with sensor dimensions. The results are simulated from a coaxial average sensor, which is the only available ANSYS option. The output from microstructure simulations includes the grain size distribution, grain orientation angle, and melt pool dimension in XY, XZ,

and YZ directions. The figure 13 below indicates the output file and graph of grain size distribution as outputs from microstructure simulations. Also, the distribution of grains along different planes is shown.

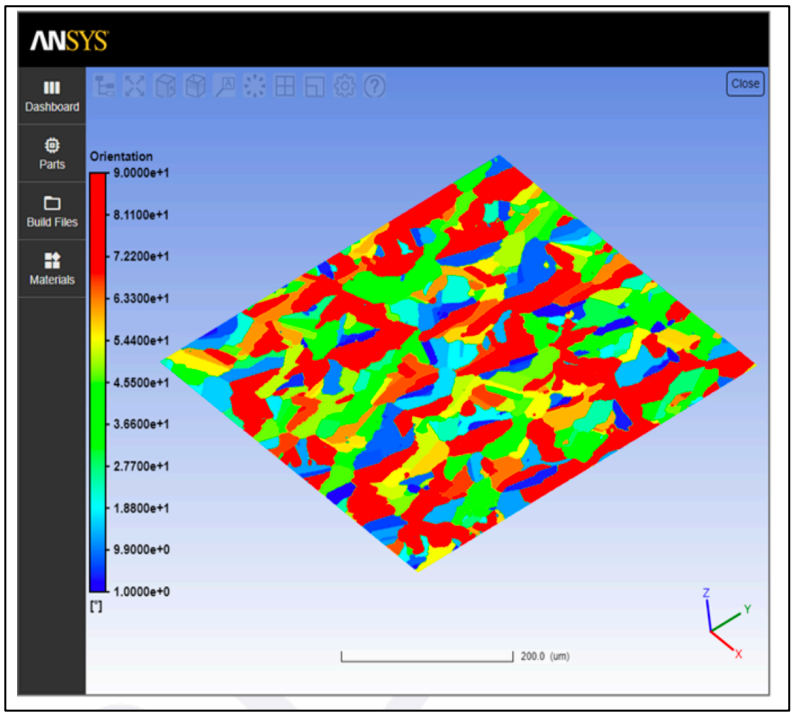

Figure 13: Grain distribution along XY plane in ANSYS (ANSYS, Inc. Additive User's Guide, 2019)

Chapter 7: Data Collection

The data collected from the simulations were used for the development and evaluation of the proposed ML Models. Figure 14 below shows the overall architecture of the ML Models for porosity and microstructure predictions.

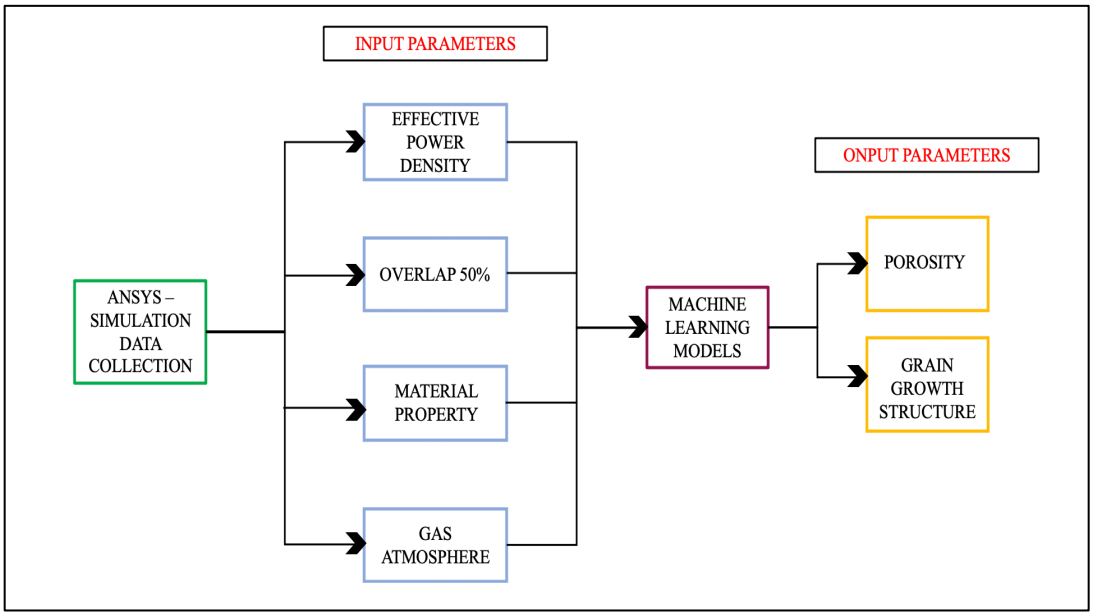

Figure 14: Architecture of the proposed Machine Learning System

The data collected from the simulations were used as input parameters to the ML Models. These parameters were then fed to the ML Model, and predictions were made whether the grain evolution structure was equiaxed/columnar and what is the percentage of porosity present in an additively manufactured part.

Table 2 shows the input and output parameters for porosity prediction. The total data included for porosity prediction is 482 data points, further divided into 80% training and 20% testing dataset. The aggregate data is gathered using ANSYS simulation for porosity prediction.

The input parameters considered in this study include energy density, laser power, powder size, gas atmosphere, overlap rate, and grain size. The outputs included in relation to the input are porosity percent and the type of grain structure i.e., equiaxed or columnar grains.

Sr. No	**Laser Power(W)**	**Scan Speed (mm/s)**	**Hatch Spacing**	**Layer Thickness (mm)**	**Energy Density (J/mm^3) $= \frac{P}{V \times t \times d}$**
1	175	500	0.1	0.05	70
2	175	650	0.1	0.05	53.84
3	500	1200	0.1	0.05	83.33
4	375	1000	0.1	0.05	75
5	375	2000	0.1	0.05	37.5
6	425	1200	0.1	0.05	70.83
7	370	711	0.1	0.05	105.71
8	180	800	0.1	0.05	45
9	175	1275	0.1	0.05	27.45
10	285	640	0.1	0.05	89

Table 1: Parameters to calculate Energy Density

Table 2 below indicates a few randomly selected input parameters for the porosity model with the corresponding value of porosity. The data shown in Table 2 is the training dataset, which has parameter gas atmosphere and overlap rate constant as 1ppm and 50% while varying other

parameters. The higher overlap rate increases the time taken to fabricate the printed part which results in an increase in the primary time. So, overlap rate is expected to have an optimum value. Shielding gas in PBF affects material porosity, surface strength and process stability. The gas flow rate therefore needs to be consistent and constant to minimize variability in material properties.

Sr. No	Energy Density (J/mm^3)	Laser power(W)	Powder Size (um)	Gas Atmosphere (PPM)	Overlap Rate (%)	Porosity
1	70	175	49	1	50	0
2	53.84	175	47	1	50	0.0002
3	83.33	500	31	1	50	0
4	75	375	51	1	50	0
5	37.5	375	25	1	50	0.0007
6	70.83	425	31	1	50	0
7	105.71	370	25	1	50	0
8	45	180	27	1	50	0.0019
9	27.45	175	22	1	50	0.0584
10	89	285	54	1	50	0.01

Table 2: Input data for training Porosity Prediction Model

The data used for testing/validating the model's precision is taken from a study conducted by Kumar (2019).

Table 3 below indicates a randomly selected set of training data for the microstructure prediction model. The dataset was categorized based on Newell et al., (2019) experiments where they discovered that equiaxed grains had an average diameter below 10.3 um and columnar grains had an average diameter of 44 um for as built L-BPF samples. The total dataset had 12,333 data points used in training and testing the model with six input parameters listed below. The data was split into 75% training and 25% testing set and used to see different ML model performances. The use of different ML model indicated which model predicts/outperform other models.

Sr. No	Energy Density (J/mm^3)	Laser power (W)	Powder Size (um)	Gas Atmosphere (PPM)	Overlap Rate (%)	Grain Diameter	Columnar=1 Equiaxed=0
1	70	175	33	1	50	2.76	0
2	53.84	175	56	1	50	44.72	1
3	83.33	500	55	1	50	51.7	1
4	75	375	44	1	50	7.23	0
5	37.5	375	57	1	50	27.17	1
6	70.83	425	20	1	50	9.3	0
7	105.71	370	25	1	50	156.82	1
8	45	180	37	1	50	4.37	0

Table 3: Input data for training Microstructure Prediction Models

Chapter 8: Results

This chapter will summarize the results obtained, the data used, and the experimental method to solve the research questions discussed in chapter 3. Also, the setting up of the dataset and the experiments will be discussed.

The section below provides the numbers for the method described in this study. Table 4 provides an overview of the number of algorithms, the dataset, the total number of attributes used, etc. used for the prediction and classification.

Number of algorithms used in the system	6
Total number of datasets	2
Total number of input attributes	6
Total number of datapoints in porosity model	482
Total number of datapoints in microstructure model	12,333

Table 4: Statistics of used data.

Evaluation of the machine learning model's accuracy for porosity and microstructure prediction is discussed as follows:

8.1 Backpropagation Neural Network for Porosity and Density Prediction

Here, a Backpropagation Neural Network model was used for predicting porosity and density in a 3D printed part. Due to a lack of experimental data, simulation data was collected from ANSYS

additive suite. The total data gathered from the simulation was 482 data points, which were fed into the backpropagation neural network where the input and output parameters were defined. The data was then split into training and testing with a ratio of 0.8 and 0.2. and was finally trained on a backpropagation neural network. The training and testing accuracy are nearly 99%. In order to achieve this accuracy for the model, hyperparameter tuning was done, which helps reduce the overall cost/error, which subsequently results in better prediction by the model.

Sr. No	Parameter	Initial Value	Final Value
1	Learning Rate	0.01	**0.0001**
2	Optimizer	Adam	**Adam**
3	Number of layers	5	**2**
4	Number of neurons in layer 1	16	**3**
5	Number of neurons in layer 2	32	**6**
6	Activation function	ReLU	**ReLU**
7	Number of Epoch	100	**150**
8	Batch Size	50	**10**

Table 5: Backpropagation Neural Network hyperparameters with final changes to suit the scope of this study.

To evaluate how accurate the backpropagation model is 7 data points for cross-validation from a research paper were considered. Table 6 showed the results for the cross-validation using data points from research papers with corresponding input parameters. Cross-validation or evaluation of ML Models on new data is important to detect if the model is overfitting or failing to generalize the pattern. The predicted/estimated results are close to the actual porosity level (Figure 15).

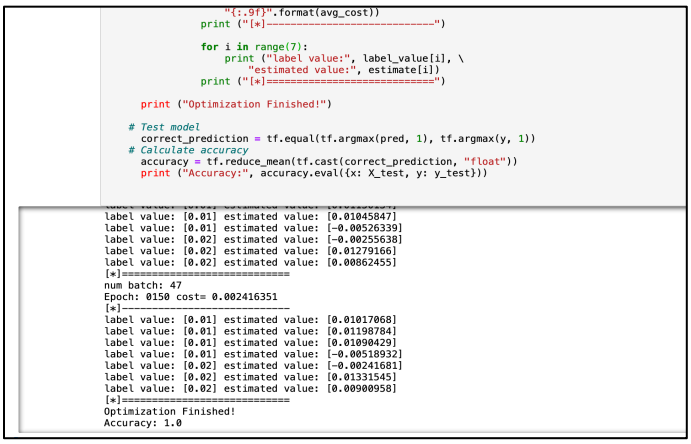

Figure 15: Overview of results obtained for porosity from Backpropagation Model in jupyter notebook

The table 6 shows the input parameter values, resultant output porosity level from research paper and the output predicted by ML model (Kumar, 2019). The negative estimated porosity value signifies that actual porosity values are close to zero. The model has thus predicted negative values. The negative outputs can be considered zero in this case, since the entire dataset consists of a positive value.

Sr. No	Energy Density (J/mm^3)	Laser power(W)	Powder Size (um)	Gas Atmosphere (PPM)	Overlap Rate (%)	Research paper porosity value	Estimated porosity value
1	103	330	60	1	50	0.01	0.0101
2	70.3	225	20	1	50	0.01	0.0119
3	89	285	37	1	50	0.01	0.0109
4	47	375	54	1	50	0.01	-0.0051
5	41	330	24	1	50	0.02	-0.0024
6	51	165	48	1	50	0.02	0.0133
7	117	375	31	1	50	0.02	0.0090

Table 6: Overview of results for actual vs predicted porosity level

8.2 Meta-Classifier and Support Vector Machine (SVM) for Microstructure Prediction

Here, Support Vector Machine (SVM) and Meta-classifier consisted of K-Nearest Neighbor (KNN), Naïve Bayes, Random Forest Classifier, and combined classification result as meta-classifier were used to predict the grain growth structure. The models used are all for addressing the classification problem. The dataset of 12,333 data points was used in training and testing of the model with six input parameters mentioned in Table 3. The steps for creating these classification models are similar to the backpropagation neural network except for the data balancing and hyperparameters in the models. The data is first loaded into the system. Then, input and output parameters are defined. The data is checked for imbalance (Figure 16 and 17) and then SMOTE analysis is done to balance the number of classes. The dataset is then split into a ratio of 0.8 and 0.2 for training and testing and finally, the models are trained.

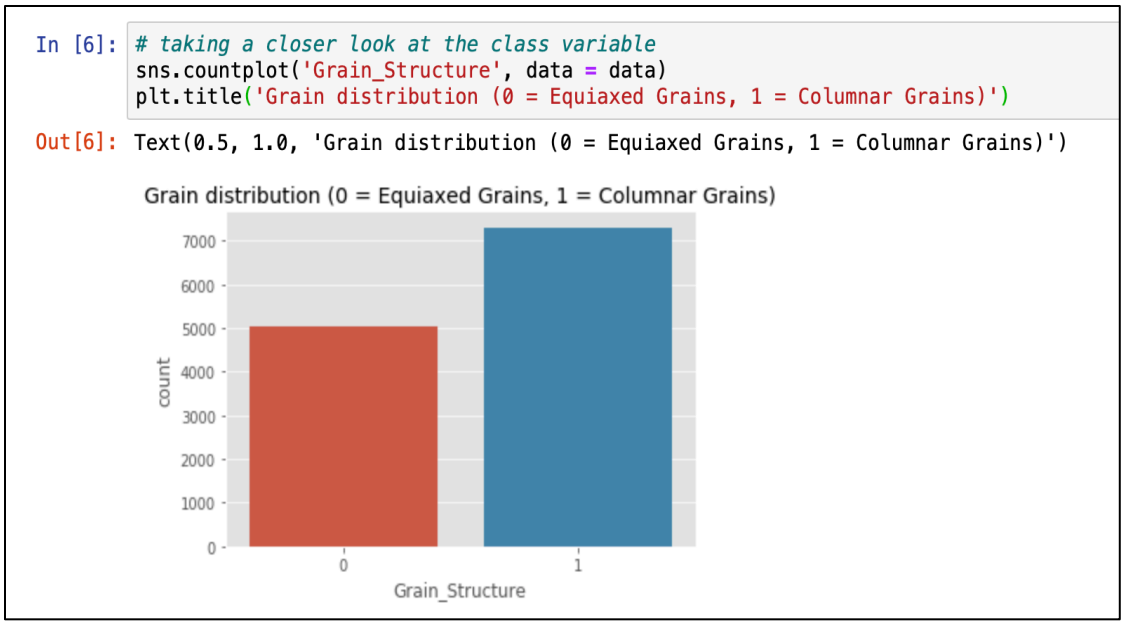

Figure 16: Class distribution of equiaxed and columnar grains in the dataset

```
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/Users/priyadhage/opt/anaconda3/lib/python3.7/site-packages/tensorboa
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprec
understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/Users/priyadhage/opt/anaconda3/lib/python3.7/site-packages/tensorboa
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprec
understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/Users/priyadhage/opt/anaconda3/lib/python3.7/site-packages/tensorboa
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprec
understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/Users/priyadhage/opt/anaconda3/lib/python3.7/site-packages/tensorboa
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprec
understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])

[(0, 7306), (1, 7306)]
```

Figure 17: Data balancing after applying SMOTE Analysis

Table 7 below indicates the microstructure prediction model for simulation data- Columnar grains= 1 and Equiaxed grains= 0 for input parameters (Newell et al., 2019):

Energy Density (J/mm^3)	**Laser power(W)**	**Powder Size (um)**	**Gas Atmosphere (PPM)**	**Overlap Rate (%)**	**Grain Diameter (um)**	**Output 0 = Equiaxed grain 1 = Columnar grain**
70	175	47	1	50	62.96	0 1
37.5	375	59	1	50	15.59	1
70.83	425	34	1	50	3.57	0

Table 7: Overview of results for columnar/equiaxed grains on training data

Energy Density (J/mm^3)	Machine Learning Model name	Accuracy %	Grain type	Actual output	Predicted output
70	Support Vector Machine (SVM)	99.8	1 = Columnar grain	1	1
	K-Nearest Neighbor (KNN)	100		1	1
	Naïve Bayes	99.0		1	1
	Random Forest Classifier	100		1	1
	Meta-Classifier	100		1	1
37.5	Support Vector Machine (SVM)	99.8	1 = Columnar grain	1	1
	K-Nearest Neighbor (KNN)	100		1	1
	Naïve Bayes	99.0		1	1
	Random Forest Classifier	100		1	1
	Meta-Classifier	100		1	1
70.83	Support Vector Machine (SVM)	99.8	0 = Equiaxed grain	0	0
	K-Nearest Neighbor (KNN)	100		0	0
	Naïve Bayes	99.0		0	0
	Random Forest Classifier	100		0	0
	Meta-Classifier	100		0	0

Table 8: Accuracy and actual vs predicted results by different ML models

For the cross-validation of the models for grain growth structure, data points used are listed in Table 9 (Newell et al., 2019). Only two data points from research paper matched the parameters used for running simulation. Therefore, only two data points were used for cross-validation. The results for the validation and accuracy of different algorithms are shown below in Table 10 & 11.

Parameter	Simulation Parameters	Research paper Parameters
Material	IN718	IN718
Process	L-PBF	L-PBF
Laser Power	175-500 W	370 W
Scanning Speed	500-2500 mm/s	700 mm/s
Layer thickness	0.05mm	0.14 mm
Energy Density	37.5-105.71 J/mm^3	40.72- 47.2 J/mm^3

Table 9: Comparison of input parameters of data points from research paper with simulation data points

```
In [6]: from sklearn import model_selection
        from sklearn.linear_model import LogisticRegression
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.naive_bayes import GaussianNB
        from sklearn.ensemble import RandomForestClassifier
        from mlxtend.classifier import StackingClassifier
        import numpy as np
        import warnings

        warnings.simplefilter('ignore')

        clf1 = KNeighborsClassifier(n_neighbors=1)
        clf2 = RandomForestClassifier(random_state=1)
        clf3 = GaussianNB()
        lr = LogisticRegression()
        sclf = StackingClassifier(classifiers=[clf1, clf2, clf3],
                                  meta_classifier=lr)

        print('3-fold cross validation:\n')

        for clf, label in zip([clf1, clf2, clf3, sclf],
                              ['KNN',
                               'Random Forest',
                               'Naive Bayes',
                               'StackingClassifier']):

            scores = model_selection.cross_val_score(clf, X_resampled, Y_resampled,
                                                     cv=3, scoring='accuracy')
            print("Accuracy: %0.2f (+/- %0.2f) [%s]"
                  % (scores.mean(), scores.std(), label))

        3-fold cross validation:

        Accuracy: 1.00 (+/- 0.00) [KNN]
        Accuracy: 1.00 (+/- 0.00) [Random Forest]
        Accuracy: 0.99 (+/- 0.00) [Naive Bayes]
        Accuracy: 1.00 (+/- 0.00) [StackingClassifier]
```

Figure 18: Overview of accuracy obtained from Meta-classifier in jupyter notebook

```
In [8]: # Now predict for a random example
        # Put testing data here and the maintain syntax as shown below

        example = np.array([[40.72, 370, 30, 1, 50, 9.75], [47.2, 370, 30, 1, 50, 12.20]])  #solution is (0,1)
        # Always declare this as lists of lists
        example = example.reshape(len(example), -1)
        # learn as idiom for all predictions

In [9]: prediction = clf1.predict(example)
        prediction1 = clf2.predict(example)
        prediction2 = clf3.predict(example)
        prediction3 = sclf.predict(example)

        print(prediction, prediction1, prediction2,prediction3)
        # 1= COLUMNAR GRAINS , 0 = EQUIAXED GRAINS

        [0 1] [0 1] [0 1] [0 1]
```

Figure 19: Overview of results obtained from Meta-classifier for data from research paper in jupyter notebook (Newell et al., 2019)

Energy Density (J/mm^3)	Laser power(W)	Powder Size (um)	Gas Atmosphere (PPM)	Overlap Rate (%)	Grain Diameter (um)	Output 0 = Equiaxed grain 1 = Columnar grain
40.72	370	30	1	50	9.75	0 0
47.2	370	30	1	50	12.20	1

Table 10: Results of columnar/equiaxed grains for data from research paper (Newell et al., 2019)

Energy Density (J/mm^3)	Machine Learning Model name	Accuracy %	Grain type	Actual output	Predicted output
40.72	Support Vector Machine (SVM)	99.8	0 = Equiaxed grain	0	0
	K-Nearest Neighbor (KNN)	100		0	0
	Naïve Bayes	99.0		0	0
	Random Forest Classifier	100		0	0
	Meta-Classifier	100		0	0
47.2	Support Vector Machine (SVM)	99.8	1 = Columnar grain	1	1
	K-Nearest Neighbor (KNN)	100		1	1
	Naïve Bayes	99.0		1	1
	Random Forest Classifier	100		1	1
	Meta-Classifier	100		1	1

Table 11: Actual vs predicted results by different ML models for data from research paper

Table 10 & 11 shows the results for cross-validation along with predicted and actual outputs from the models. All the models have predicted correct results/microstructure though there is a slight difference (less than or equal to 1 percent approximately) in the models' accuracy.

Chapter 9: Discussion

This chapter will address the assessment and discussion of the results achieved; the methodology is chosen, experiment validity, and reliability. It will also reflect on the approach taken to the research tasks, highlights its benefits, and consider whether the right method has been chosen to solve the problem in question.

9.1 Results Interpretation

9.1.1 Porosity Prediction Model

For predicting the porosity on simulation data with a 3*3*1 (mm) geometry; regression analysis and random forest algorithm were tested - the output data type as continuous variable limits the use of many machine learning algorithms. Commonly used types of ML models included are simple regression analysis and its variants, Decision trees and Backpropagation Neural Network. Out of these model's Decision trees and Regression analysis models failed at predicting the value of porosity, and the Backpropagation Neural Network model outperformed these models.

The Backpropagation Neural Network model gave an accuracy of 100% while the cost function, also known as the error rate, was deficient as of 0.00241. The accuracy and error rate results were achieved after hyperparameter tuning of the neural network as it converges the cost function to minima which results in less deviation between the actual and predicted values of porosity for example 0.01 was predicted as 0.0119, and 0.02 was predicted as 0.013 (Table 6). It can be seen that the deviation in the values is very low (less than 0.01).

The main advantage of using the Backpropagation Neural Network model was not only that it predicted values close to the actual value with very less cost function but also that it requires very little computational power.

9.1.2 Microstructure Prediction Model

During the execution of classification models, it was found that the data was imbalanced (Figure 16 & 17), as a result of which the models could correctly predict only one class. The data was balanced for both classes-7306 data points for each class after implementation of the SMOTE methodology, and the model correctly predicted the results for columnar and equiaxed grains. Among all the algorithms used for microstructure prediction, the best results are achieved by Meta-classifier, K-Nearest Neighbor, and Random Forest classifier with 100% accuracy. However, on the other side, the accuracy achieved by SVM and Naïve Bayes classifier does not deviate much from the different three classifiers, which is 99.98 and 99.9%.

The validation of classifiers was done on both the simulation data and also for data from the research paper. The research paper data for validation indicated how accurate the classifiers are at predicting columnar or equiaxed grains. When we see the results of the classifiers' predictions in both cases, they predicted actual results, i.e., classifying equiaxed/columnar grains accurately. Also, it should be considered that all the five classifiers predicted accurate results. For the best classifier, Meta-classifier is considered as it gives results combined with other classifiers, and also its accuracy and prediction results are correct (Table 9).

The methods discussed effectively address all research questions, allowing prediction of porosity and microstructure present in a 3D printed part to be categorized and evaluated.

Chapter 10: Conclusion

This work addresses the issue of real-time process control and monitoring in the Powder Bed Fusion process. The processing time of the model used in this study was 10 minutes (approximately). It would help researchers and users to know the parameters such as porosity and microstructure beforehand. In this study, it is observed while running simulations for data collection that it took several hours for running one single simulation of geometry 3*3*1 (mm). Also, while doing actual experimentation, one can only observe the porosity and microstructure once the printing of part is finished. If there are any defects or any need for a particular grain structure, one cannot rectify it while the process is running. If the above approach of machine learning models is used, then the level of porosity and the grain growth structure will be pre-identified in minutes rather than waiting for hours to run a simulation or to let the whole experiment finish.

The whole process was achieved using an information discovery method consisting of the stages, collecting data, pre-processing data, choosing a suitable data mining approach to identify patterns among the data, and interpreting them. The findings were subsequently used for further analysis. Six different machine learning algorithms, namely Random Forest classifier, Backpropagation Neural Networks, k-Nearest Neighbor, Naive Bayes, Support Vector Machines, and Meta-classifier, used the pre-processed data in several iterations.

The algorithms were tuned according to their parameters and evaluated on data from research studies, which was isolated from the database before the models were equipped. The maximum results were achieved by Meta-Classifier, K-Nearest Neighbor, and Random Forest classifier with 0% of the misclassification rate for microstructure prediction and Backpropagation neural network with 0.00241 of the error rates for prediction of the value of porosity.

10.1 Further Research

The presented work discusses only porosity and microstructure prediction using a machine learning approach. Extending the research on the subject, an initiative to progress more and use other parameters such as mechanical properties, phases, thermal history, etc. can be done.

In machine learning models, especially neural networks, a large amount of training data is required to produce viable results. The data used was only 482 data points for porosity model due to which the architecture of the neural network had to be reduced. This study has the potential to be done on a larger scale and getting all the parameters related to a 3D printed part pre-identified.

The suggested machine learning modeling approach also needs some in-depth review of the relevant predictions and classifications to check and refine them. The preferred approach is only being verified using sample-based testing from research studies. Also, the simulations performed usually considers ideal scenarios and gives results based on those scenarios. On the other hand, while performing real experiments, many parameters affect the process, thus affecting the quality of the final part.

In summary, the presented work expands the related research on this subject by integrating performance metrics using domain-specific-focused machine learning methods. This machine

learning modeling technique can be extended to the algorithms and data for different materials with minimal tuning. Consequently, compatibility with numerous other materials can be achieved using this methodology.

References

ANSYS, Inc. (2019). Additive User's Guide (Print and Science). In. ANSYS, Inc. Southpointe 2600 ANSYS Drive Canonsburg, PA 15317.

ASTM. (2013). Standard Terminology for Additive Manufacturing Technologies. In. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959. United States

Balbaa, Mohamed Mekhiel, Sameh Elbestawi, Mohamed McIsaacb, Jeff. (2020). On selective laser melting of Inconel 718: Densification, surface roughness, and residual stresses *Materials and Design, 193*.

Blagus, Rok, & Lusa, Lara. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics, 14*(106), 1471-2105.

Brownlee, Jason. (2019). Understand the Impact of Learning Rate on Neural Network Performance. Retrieved from https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

Ceballos, Frank. (2019). Stacking Classifiers for Higher Predictive Performance. Retrieved from https://towardsdatascience.com/stacking-classifiers-for-higher-predictive-performance-566f963e4840

Chen, Qiang, Guillemot, Gildas, Gandin, Charles-Andre, & Bellet, Michel. (2017). Three-dimensional finite element thermomechanical modeling of additive manufacturing by selective laser melting for ceramic materials *Additive Manufacturing, 16*, 124-137.

Chua, Zhong YangAhn, Il Hyuk Moon, Seung Ki. (2017). Process Monitoring and Inspection Systems in Metal Additive Manufacturing: Status and Applications *International Journal of Precision Engineering and Manufacturing-Green Technology 4*, 235-245.

D.E.T., Seale L.E.J.Thomas Brown J.C.Kirkman Attallah M.M. Espino D.M. Shepherd. (2018). The barriers to the progression of additive manufacture: Perspectives from UK industry *International Journal of Production Economics, 198*, 104-118.

Donachie, Matthew, & Donachie, Stephen. (2002). *Superalloys: a technical guide*: ASM International.

Doshi, Sanket. (2019). Various Optimization Algorithms for Training Neural Network. Retrieved from https://medium.com/@sdoshi579/optimizers-for-training-neural-network-59450d71caf6

Dzeroski, Saso Zenko, Bernard. (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning, 54*, 255–273.

Gandhi, Rohith. (2018). Support Vector Machine - Introduction to Machine Learning Algorithms. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Ghetiya, Nilesh, & Patel, K. (2014). Prediction of tensile strength in friction stir welded aluminum alloy using artificial neural network. *Procedia Technology, 14*, 274 – 281

Glenn E. Bean, David B. Witkin, Tait D. McLouth Rafael J. Zaldivar. (2020). Process gas influence on microstructure and mechanical behavior of Inconel 718 fabricated via selective laser melting. *Prog Addit Manuf (2020)*.

Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaaron. (2016). *Deep Learning*: MIT Press.

Group, Additive Manufacturing Research. Powder Bed Fusion. Retrieved from https://www.lboro.ac.uk/research/amrg/about/the7categoriesofadditivemanufacturing/powderbedfusion/

Kumar, Pankaj, Farah, Jano, Akram, Javed, Teng, Chong, Ginn, Jon, & Misra, Mano. (2019). Influence of laser processing parameters on porosity in Inconel 718 during additive manufacturing. *The International Journal of Advanced Manufacturing Technology, 103*, 1497 – 1507.

Meng, Lingbin Mcwilliams, Brandon Jarosinski, William Park, Hye-Yeong Jung, Yeon-Gil Lee, Jehyun Zhang, Jing. (2020). Machine Learning in Additive Manufacturing: A Review *the Minerals, Metals & Materials Society, 72*(6).

Moussaoui, KamelRubio, WalterMousseigne, MichelSultan, Tarek Rezai, F. (2018). Effects of Selective Laser Melting additive manufacturing parameters of T Inconel 718 on porosity, microstructure and mechanical properties *Materials Science and Engineering, 735*, 182-190.

Newell, David J., O'Hara, Ryan P., Cobb, Gregory R., Palazotto, Anthony N., Kirka, Michael M., Burggraf, Larry W., & Hess, Joshuah A. (2019). Mitigation of scan strategy effects and material anisotropy through supersolvus annealing in LPBF IN718 *Materials Science and Engineering*.

Nguyen, Quy BauNai, Mui Ling SharonZhu, ZhiguangSun, Chen-NanWei, Jun Zhou, Wei. (2017). Characteristics of Inconel Powders for Powder-Bed Additive Manufacturing Engineering*, 3*(5), 695-700.

Parimi, Lakshmi Lavanya Attallah, Moataz M. Gebelin, J.C. Reed, Roger C. (2012). *Direct laser fabrication of inconel-718: effects on distortion and microstructure Paper* presented at the 12th international Symposium on Super Alloys. https://www.tms.org/Superalloys/10.7449/2012/Superalloys_2012_511_519.pdf

Qia, Xinbo Chen, Guofeng Li, Yong Cheng, Xuan Li, Chang peng. (2019). Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives. *Engineering, 5*(4), 721-729.

Raschka, Sebastian. (2015). Stacking Classifier. Retrieved from http://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

Razvi, Sayyeda Saadia Feng, Shaw Narayanan, Anantha Lee, Yung-Tsun Tina Witherell, Paul. (2019). *A review of machine learning applications in additive manufacturing.* Paper presented at the Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 1: 39th Computers and Information in Engineering Conference., Anaheim, California, USA. https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings/IDETC-CIE2019/59179/V001T02A040/1069728

Sadowski, Magda Ladani, LeilaBrindley, William Romanoa, John. (2016). Optimizing quality of additively manufactured Inconel 718 using powder bed laser melting process *Additive Manufacturing, 11*, 60-70.

Sharma, Sagar. (2017a). Activation Functions in Neural Networks. Retrieved from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Sharma, Sagar. (2017b). Epoch vs Batch Size vs Iterations. Retrieved from https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

Simpson, Michael. (2018). Machine Learning Algorithms: What is a Neural Network? Retrieved from https://www.verypossible.com/insights/machine-learning-algorithms-what-is-a-neural-network

Tapia, Gustavo Elwany, Alaa. (2014). A Review on Process Monitoring and Control in Metal-Based Additive Manufacturing *Journal of Manufacturing Science and Engineering, 136*(6), 10.

The 10 Best Machine Learning Algorithms for Data Science Beginners. (2019). Retrieved from https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/

Yiu, Tony. (2019). Understanding Random Forest. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2