

# Machine Learning Predictions of Single Clad Geometry in Directed Energy Deposition

Michael Juhasz

Department of Mechanical and Industrial Engineering, Youngstown State University, Youngstown, USA

*Within the Additive Manufacturing area of Directed Energy Deposition (DED), single clad geometry prediction has been well covered within the literature. Currently, the two accepted methodologies of geometry prediction are physics numerical simulation, and semi-empirical regression. This work seeks to add a viable alternative through machine learning techniques. Machine learning has enjoyed many successes in the past few years due to the availability of large datasets for which these techniques scale beautifully. However, in small, high variance, tabular datasets, such as most results from physical experimentation, these techniques suffer. Presented here is a selection of machine learning methodologies which are used to extract models that perform and generalize well. Neural Networks (NNs), Gaussian Process (GP) modeling, Support-Vector Machines (SVMs), and Gradient Boosted Decision Trees (GBTs) for regression and classification are explored in this paper. These four methodologies will be applied to a small dataset containing some single clad data available in the literature and previously unpublished experimental results of this author. These techniques produce models not only with good agreement with experimental data, but also non-material specific generalizable results. Lastly, a discussion of data augmentation using Generative Adversarial Networks (GANs) with preliminary results put forth to illustrate unique, exploitable advantages capable within the machine learning paradigm.*

## Introduction

Directed Energy Deposition is in effect a combination of, and borrows heavily from, the robust technology of laser cladding/welding and computer numeric control (CNC). Toyserkani et al. provides a generous introduction to the subject of laser cladding [1]. The basis of DED technology is a coupling of the cladding technology and the motion control available in CNC machinery. [2–4]. Padding shafts in steel mills, which is the buildup/repair of parts by welding, has been an eleventh hour practice for many years. The technique is found in most literature teaching shielded metal arc welding (SMAW) for at least the last 70 years [5]. The buildup of components (non-repair purposes) using welding goes back to 1960s Germany, though what is thought of as DED required the advent of CNC and did not appear in an additive manufacturing sense until the 1990s [6]. Starting in the 1990s, DED has been the focus of significant research [7–11] including the manufacturing of metal and ceramic structures [12–19].

The physical phenomena occurring during directed energy deposition with powder injection can be described sequentially. Laser energy strikes the substrate and imparts most of its energy there. The remaining energy is split amongst reflection and heating of powder particles in flight as they

make their way to the substrate. The process is best described as the laser creating a melt pool into which metal powder is injected in order to create the clad [20]. Adjacent tracks and successive layers of solidified material are deposited to build up a three-dimensional object [21]. In blown powder DED, metal powder is conveyed through an inert carrier gas and supplied to the melt pool via the deposition tool head [15]. The deposition tool head also directs the laser energy to the substrate, and provides an inert shielding gas to protect the melt pool from the oxidizing atmosphere.

In an attempt at deeper understanding of directed energy deposition and cladding, models have been introduced to predict results and dynamics of the process. Pinkerton details the relatively current status of modeling. Those models fall into two general categories: Physics-based, and Empirical-statistical based [22]. To these two broad classes, this author seeks to introduce a subset to the Empirical-statistical models of a relatively new range of methods broadly termed as data science.

The basic physical phenomenon has been previously described. Any physics-based mathematical model applied the phenomenon would, at a minimum, need to address heat transfer, mass transfer, and continuity. Under certain sets of assumptions and boundary conditions, analytic results can be derived. These analytic results have varying levels of sophistication, spanning from overly simple models that lack detail but prove very useful in practical applications, up to complex analytic results that exhibit more accurate predictions, but can be unwieldy to apply [23,24]. Relaxation of assumptions and the introduction of complex phenomena, such as radiation pressure and surface tension effects, take physics-based models from *analytically* solvable into the *numerically* solvable regime [25–28]. Though other discretization techniques have been tried, finite element models furnish the most usable numerical method to solve these complex sets of equations. All these various models have their place and find their best applicability when the physical system most closely matches the model assumptions. All these models lose predictive power in the face of process variability.

Empirical-statistical models address process variability through statistics of direct measurements to construct process maps which predict behavior that otherwise is not directly tested. Empirical-statistical models are the more pragmatic of the two modeling categories. They rely only upon available user inputs to assess resultant behavior. The trade off is the loss of information of the physical phenomenon, lost as part of the effort to better understand and control the process itself. These models cannot indicate whether, for example, radiation pressure is a significant factor in deposition height, but rather can indicate what combination of process variables will produce a given height within margin. Regression-type models are, by and large, the simplest and most studied form. The regression surface of choice:

$$R \sim \eta(P^\alpha \widehat{m}^\beta V^\gamma) + \varepsilon \quad \text{Eqn. 1}$$

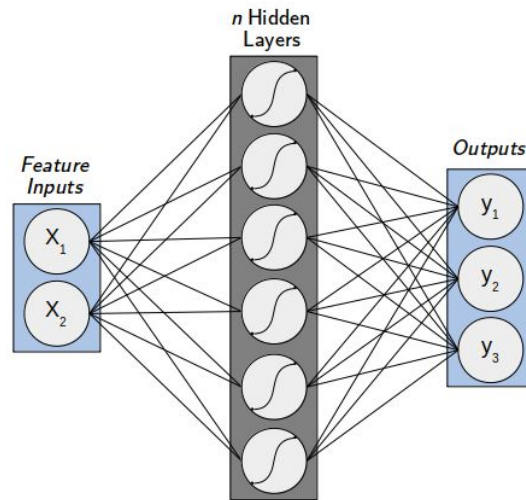
Where  $R$  is the response of interest.  $P$ ,  $m$ ,  $V$  are laser power, mass flow rate, and traverse velocity respectively. Lastly, the free parameters used in the fit are  $\eta$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\varepsilon$ . These models have been fitted using a variety of material combinations, such as cobalt alloys on carbon steel [29,30], nickel alloys on carbon steel [31,32], copper on aluminum [33], a cobalt alloy on cast iron [34], stainless steels on carbon or stainless steel [35,36], a nickel alloy on Inconel [37], WC12Co on stainless steel [38], and also Ti-6Al-4V on Ti-6Al-4V [39]. While all the studies achieved regression parameters that described that individual case well, the parameter values appear only applicable to a given material combination. This illustrates the apparent loss of phenomenological information which would be captured across all conditions. However, this should not be discouraging. Regression models are easily applied and have significant value in a practical sense.

In addition to regression-type empirical-statistical models, there are more exotic versions that have been examined in the literature. These include machine learning techniques and neural networks [40–43], NARMAX (Nonlinear Autoregressive Moving Average Model with eXogenous inputs) [44], and those based on block models [45]. Given the proper setup to generate the required amount of data, the models perform well and lend themselves readily to feedback control systems. The incorporation of sensors into the work envelope and the handling of large data sets generally detract from the reliability of these methods. This is an active field of research.

Data Science and Machine Learning (ML) is a burgeoning set of statistical methods that have been developed as a response to increasingly larger and larger datasets produced by advancing technology. It is not the aim of this author to provide a comprehensive review as this subject still lacks distinct borders. In the literature ML methods have been tried a few times in regards to laser-based DED. Mondal et al. [40,41] evaluated neural network effectiveness of predicting clad geometry in two separate papers, both with limited success. Their results come at the very beginning of what is now called the “Deep Learning” revolution, and therefore reflect the limited knowledge base of such networks at the time. More recently, Khanzadeh et al. [42] examined porosity prediction via in situ thermal imaging in DED by first preconditioning the data by principal component analysis (PCA), then applying K-nearest neighbor, decision trees, and support vector machine techniques. That group [43] also used an unsupervised self-organizing map (clustering) algorithm to parse thermal images of the molten pool temperature distribution during DED into classes. In both reports, melt pool temperature distributions were shown to correlate well with the location of porosity in the build.

Out of the set of statistical methods associated with Machine Learning, four methods were chosen for varying reasons which are discussed next. With the latest successes of deep learning neural networks, though not typically applied to tabular data, are enormously popular at present. It seemed necessary to evaluate their effectiveness on a tabular dataset. Gaussian processes were chosen for the Gaussian inductive prior which seemed a natural fit for experimental, physical science data. Support vector machines were selected, as they are well researched and enjoyed many successes in machine learning through the 1990s and the early 2000s. Lastly, anecdotally gradient boosted decision trees are known for their performance on tabular data.

Neural Networks (NNs) are a machine learning technique loosely based on the notion of synaptic neuron firing. **Figure 1** shows the general architecture of neural networks where (left to right in the **fig. 1**) the input layer is fully connected to  $n$  hidden layers, and ends with a fully connected output layer.



**Figure 1:** Typical Neural Network architecture

Neural networks have been the subject of considerable attention over the last decade among the computer science community [46–48]. The successes have made the science community at large take notice. Scientists have been grappling to incorporate neural networks into their traditional paradigms [49,50]. An advantage of neural networks is the capability to learn non-linear models and develop relationships between features which would remain otherwise unnoticed using more traditional techniques [51,52]. Some of the disadvantages of neural networks include that, in general, they have non-convex loss surfaces where there can exist multiple local minima. A fair amount of effort has been leveled at ensuring convergence to global minima, i.e., stochastic gradient descent and its various forms [53,54]. Also, different weight initializations can lead to different model outcomes. Neural networks require tuning a number of hyperparameters, such as

the number of neurons, hidden layers, and iterations. Also, they are not only sensitive to architecture, but also to feature scaling [55]. Even given these disadvantages, the power of neural networks has been shown in multiple landmark achievements [46–48] to be worth the initial energy spent in architecture and in hyperparameter tuning.

Gradient Boosted Decision Trees (GBTs) is a machine learning technique for regression and classification problems. GBTs are generally well suited to tabular data, such as the dataset in this investigation [56]. The technique produces a prediction model from an ensemble of weaker prediction models. The weaker models are optimized from a loss function using gradient descent [57]. It is in the optimization technique that both NNs and GBTs have commonality.

The remaining models chosen for this investigation share a commonality: both can be considered as kernel methods. These two models are Gaussian Processes and Support Vector Machines [58]. While there are certainly differences between them, they do share the common mechanism of Mercer’s Theorem and the so-called “kernel trick.” The models in general project the data into a higher dimensional space, where the prior and optimization can be applied, after which the models project the solution back into the original space [59,60].

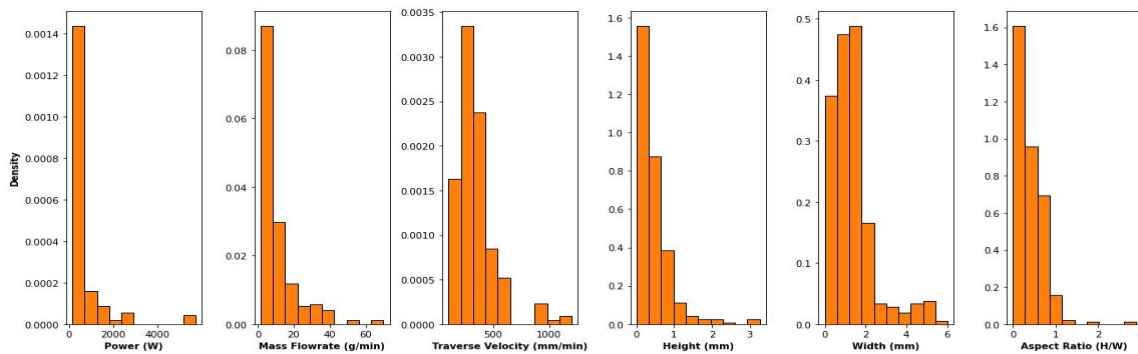
Gaussian process (GP) models take as its inductive prior that in the high dimensional space the data of interest is normally distributed [61]. This is a useful prior, as measurements made of natural processes under controlled experimental conditions generally are well approximated by a normal distribution. Of course, this previous description is a notional explanation of the model. Much effort was expended to generate kernels which at the outset have the highest probability to properly describe the data. Algorithmically, once the base kernel is constructed, the GP model iteratively defines the optimal covariance of the kernel, which also best matches the input data [62]. This covariance is simply the dependence of one data point upon another. The positive features of Gaussian processes are that predictions are probabilistic. Therefore, confidence intervals and other statistical measures can be provided. GP is versatile, as different kernels can be specified. However, it should be noted that leveraging the prior yields fair results with only commonly specified kernels. The main disadvantage of GP is that the method is not sparse, and therefore, at high data dimensionality, computation can be costly.

Support Vector Machine (SVM) models also project data into a high dimensional space. SVM, however, differs from Gaussian process models. As a classifier, SVMs seek a hyperplane in high dimensional space, which creates the maximum margin between classes [63,64]. As a regressor, SVM’s seek the minimum  $\pm\epsilon$  (epsilon envelope) which describes the data [65]. Though the same general method is involved, the high dimensional spaces are very different in character between a classifier and regressor. The advantages of support vector machines are that they use only a subset of data points in the decision function (called support vectors, hence the name), so

the method is relatively memory efficient. Just as in Gaussian Processes, SVMs are versatile. Different kernels can be specified and fair results are achievable with only commonly specified kernels. SVMs differ from GPs in that SVMs do not directly provide probability estimates. These estimates must be calculated by other means, which, in certain situations, can be a disadvantage.

## Methods and Materials

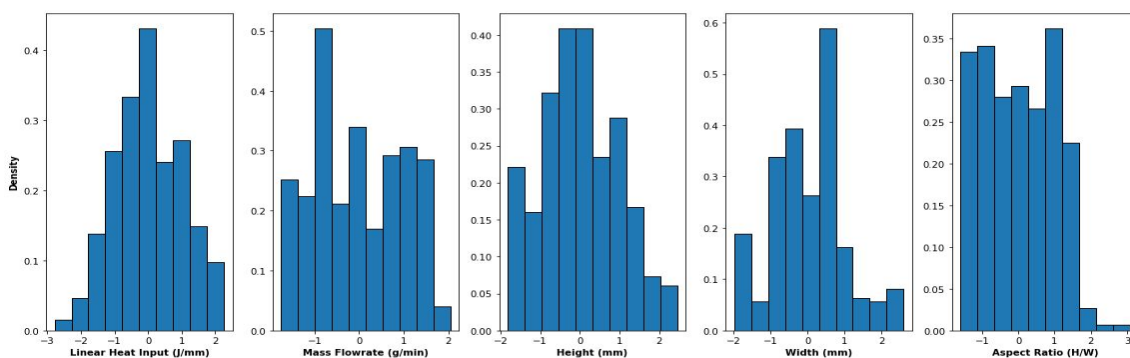
The dataset of interest is a collection of process parameter study data points from previously unpublished work of this author and results found in the literature [29–33,35,37–39]. This is by no means a comprehensive list, and this author invites researchers in the field to add to the dataset (available on GitHub @ [github.com/mjjuhasz/Canvasback.git](https://github.com/mjjuhasz/Canvasback.git) filename is DED\_Database.csv). The dataset is comprised of input information (Powder Material, Substrate Material, Spot Size (mm), Power (W), Mass Flow Rate (g/min), and Traverse Velocity (mm/min)) and the deposition resultants (Height (mm), Width (mm), and Contact Angle (deg)). This dataset also, in places, has some missing information. A combination of imputation and data exclusion were employed on the dataset to preserve as much information as possible and only exclude data when necessary. The methodology that was employed is to develop dataset-wide models based on the aforementioned ML techniques to assess the predictive power of those models. Comparisons are drawn between these four different models to assess if a preferential technique exists. Also, although minimal, the results of these ML-based models will be compared to empirical-statistical correlations that already exist in the literature. To aid in these comparisons and plotting, the combined parameter of “linear heat input”, Power (W)/Traverse Velocity (mm/min), is used in combination with Mass Flow Rate (g/min).



**Figure 2:** Histograms of relevant variables in the dataset.

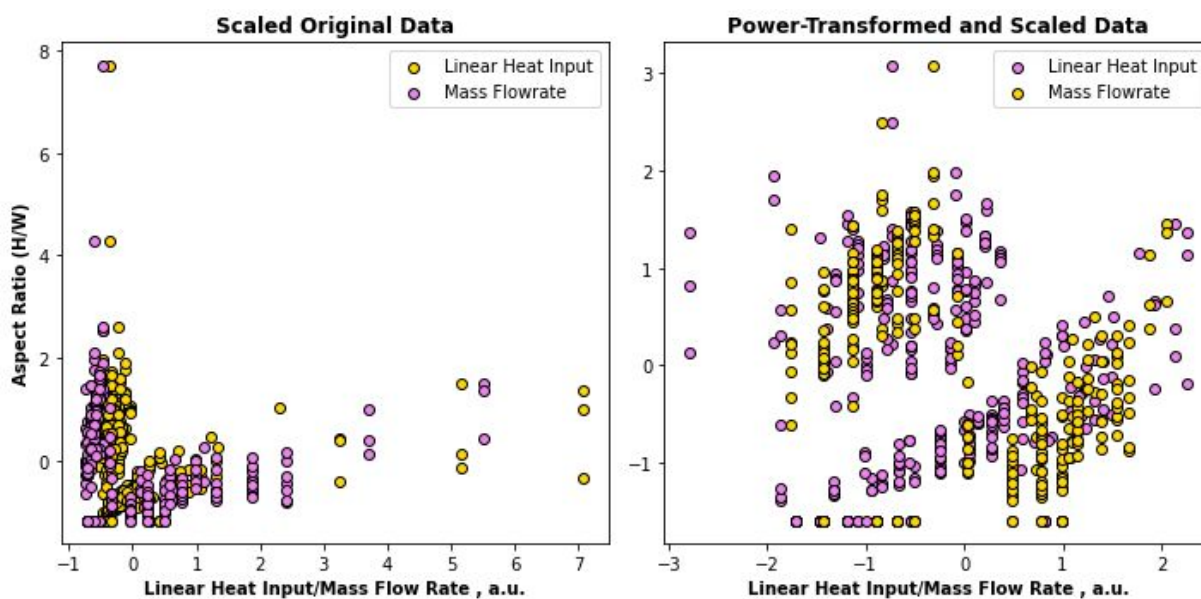
**Figure 2** shows the distributions of relevant data in the dataset. The data distributions are mostly of an exponential form. This exponential form of the distributions can be countered by appropriate measures, but in order to achieve better results across all methods tested, the data was power-transformed and scaled [66]. Power-transforming the data results in distributions

which tend to be more Gaussian-like, or at least more balanced, as shown in **fig. 3**. In general, machine learning techniques prefer feature distributions which are isotropic since these techniques are minimizing across a complex cost function in order to find a solution.



**Figure 3:** Histograms of preprocessed variables in the dataset.

**Figure 4** further illustrates the effect of power-transforming and scaling the original dataset. The original data exhibits two distinct manifolds within the space, whereas the power-transformed data is more homogeneous through the space. The manifolds do not completely disappear with preprocessing but are definitely reduced.

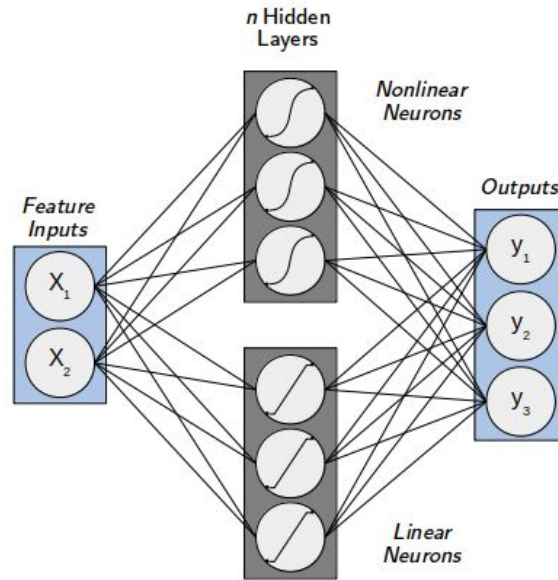


**Figure 4:** Scatterplot comparison of original, and preprocessed data.

For classification to be performed, the output of interest, aspect ratio (H/W), must be split into classes before application of the ML techniques. Since the original data is of a continuous nature, a target value of .5 was chosen and intervals of increasing distance were formed around that target value. The final intervals or classes for aspect ratio are as follows:

$$\{(min, .375), (.375, .425), (.425, .475), (.475, .525), (.525, .575), (.575, .625), (.625, max)\}$$

With the dataset preprocessed, application of the previously mentioned machine learning techniques was performed. Of note, the neural network architecture (**fig. 5**) used in this study borrows from McGhee [67] and uses what is termed a linear ‘skip connection.’ The generalized notion is that this set of neurons with only a linear activation function will capture linear artifacts in the data whilst the nonlinear neurons will then capture the nonlinear artifacts. McGhee suggests this architecture as a means to increase the effectiveness of NNs on tabular data [67].



**Figure 5:** Neural network architecture used in this study for both the regressor and classifier networks.

Summarized in **Table 1** are all the pertinent parameters for each method used in this investigation. All methods employed used 80% of the dataset for training, and were subsequently validated on the remaining 20% of the dataset. The results which are presented later are based upon this withheld 20% validation data. Any extra details and programmatic method construction are available in the code (available on GitHub @ [github.com/mjjuhasz/Canvasback.git](https://github.com/mjjuhasz/Canvasback.git) filename is DED\_Process\_Parameter.ipynb)

**Table 1:** Method parameters for the ML techniques used in this study.

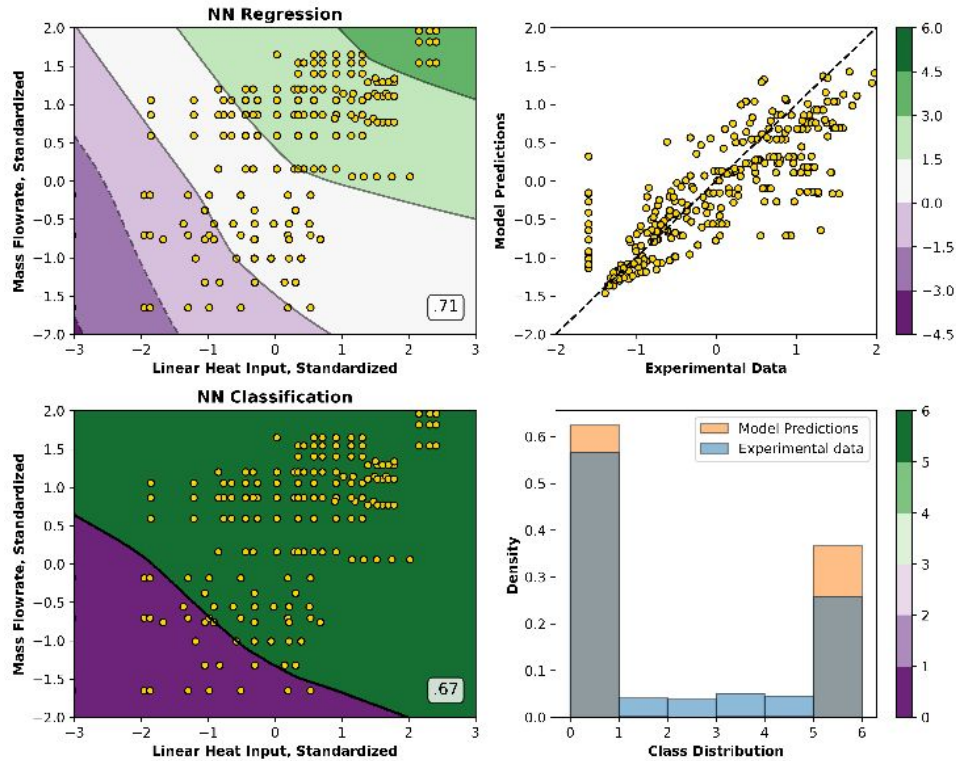
Method Parameters					
Neural Network	Linear Neurons	Nonlinear Neurons	Nonlinear Hidden Layers	Activation	Learning Rate



Regression	283	210	1	ELU	.000871
Classification	358	744	1	ELU	.000465
<b>Gaussian Process Model</b>	Kernel				
Regression/ Classification	['Constant' x 'Radial Basis Function' + 'White noise'] + ['Constant' x 'Radial Basis Function' + 'White noise']				
<b>Support Vector Machines</b>	Kernel	Gamma		C	
Regression	Radial Basis Function	$1/(\text{'\# of features'} \times \text{'feature variance'})$		5	
Classification				12	
<b>Gradient Boosted Decision Trees</b>	Estimators	Learning Rate	Max Depth	Loss	
Regression	1000	.01	20	Least Squares	
Classification	1000	.01	20	Deviance	

## Results

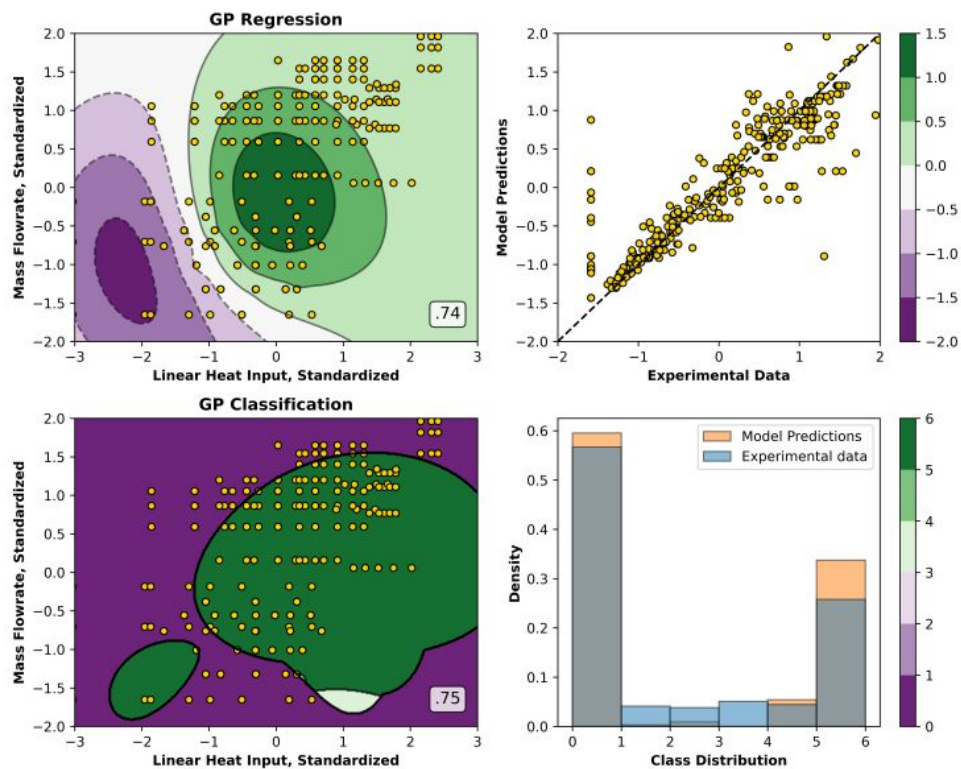
The purpose of this study was to determine whether ML-based prediction methods can be used effectively to predict single clad geometry in DED. Figures 6, 7, 8, and 9 demonstrate, respectively, the four ML techniques applied to the dataset.



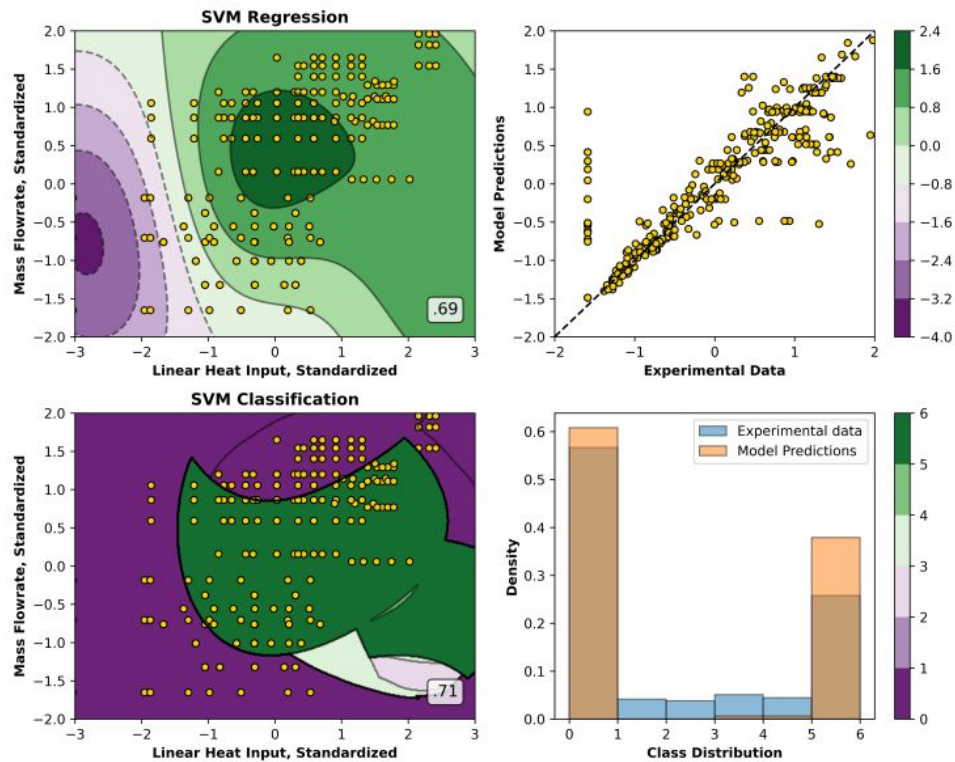
**Figure 6:** Neural network results for regression and classification. Response surface plot (left) and prediction accuracy (right) of the model.

Neural network regression (**fig. 6**) exhibits a response plot that is strikingly similar to the logarithmic curves that are inherent in the empirical-statistical models discussed earlier. That plot is also representative of what is found in the literature. There is fair amount variance still present in the results of the regression model. This indicates a shortcoming in the model where there is an already high variance dataset. However, the success of the model is that it produces a generalization between material and processing conditions. This non-reduction of variance is clearly seen when examining the classification results of the neural network model. The network is only a binary classifier, even though it was subject to multiclass input. This is not entirely surprising given the problem at hand. It's much harder to create a single clad of ideal geometry than not. There is therefore an inherent instability in the process and the model picks up on this, hence the variance.

The Gaussian process model (**fig. 7**) by the metrics is more successful, though only marginally, at regression and classification than the previously described neural network models. The response surface indicates a departure from the logarithmic curves of the NN and empirical-statistical models. This departure results in less variance than those models. This is further supported by the classification results where the GP model is more than simply a binary classifier. , The inclusion of more classes yields a better predictive model.



**Figure 7:** Gaussian process results for regression and classification. Response surface plot (left) and prediction accuracy (right) of the model.

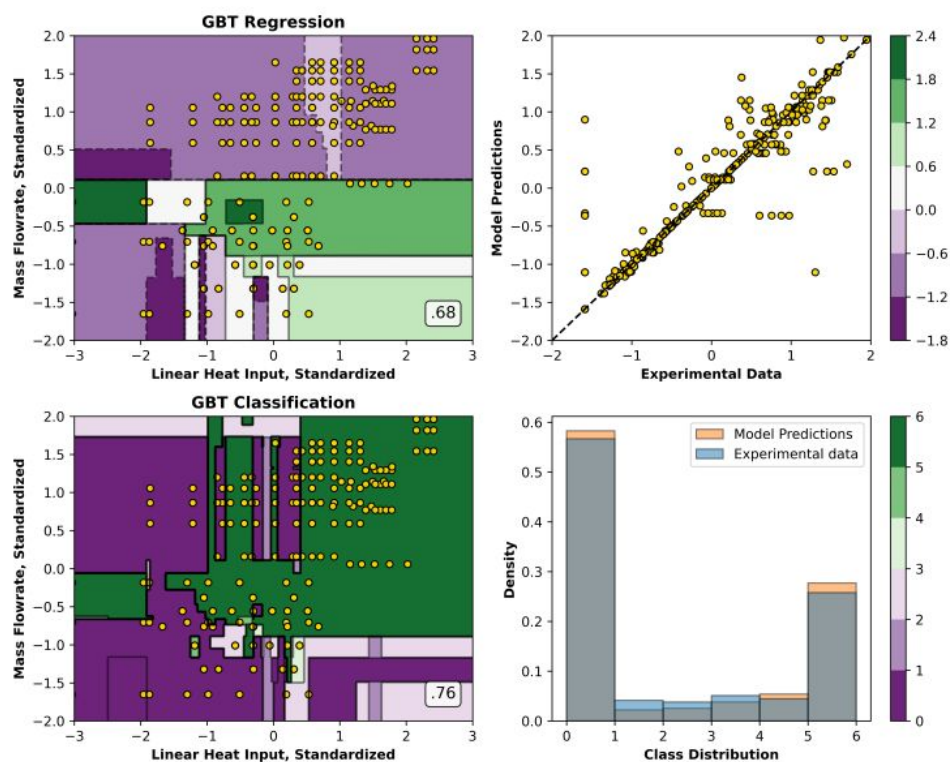


**Figure 8:** Support vector machine results for regression and classification. Response surface plot (left) and prediction accuracy (right) of the model.

Support vector machines (SVMs), the other kernel method chosen, underperformed when compared against the GP model. The deduction is that the Gaussian inductive prior of the GP model is more informative at high dimension versus the radial basis kernel in the case of this dataset. SVMs when compared to the NNs, have lower variance and better classification accuracy, even though the SVMs and NNs both entirely overestimate non-optimal clads as classifiers.

The results (**fig. 9**) from the gradient boosted decision tree (GBT) model lends credence to its effectiveness on tabular data. As a classifier, the GBT model outperformed the other models. The resulting classification distribution closely follows that of the dataset. As a regressor, GBT showed the lowest variance, however, the coefficient of determination,  $r^2$  value, is the lowest of all models examined. This suggests that outliers might be overpenalized by the model, lowering the  $r^2$  value. Further investigation is necessary. This effect may be a blessing in disguise, and what may be being observed is that the other models are incorporating outliers at

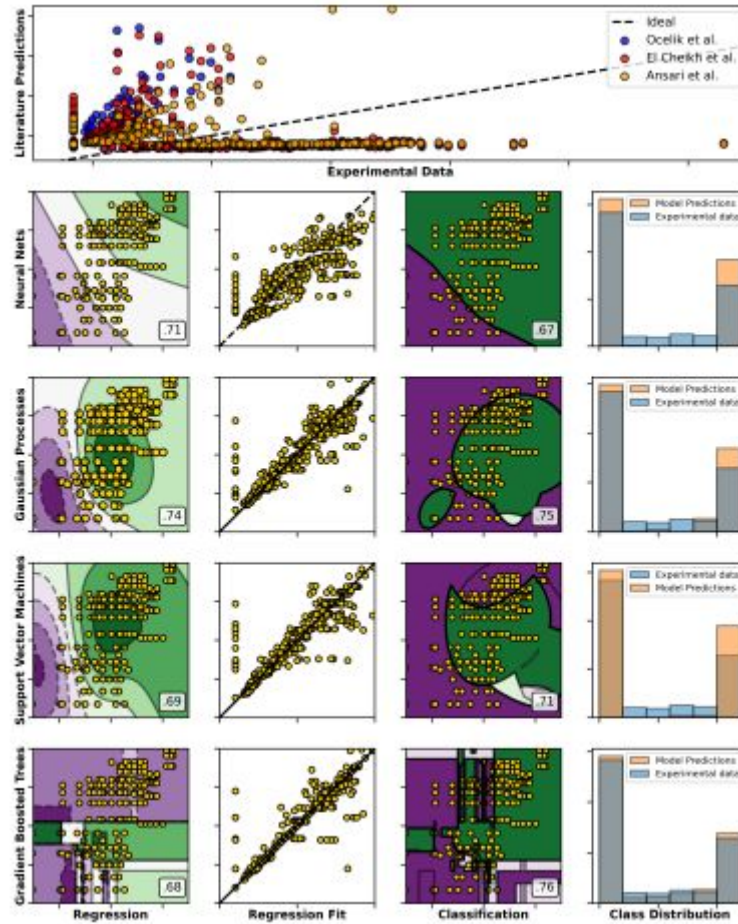
the cost of variance for an increase in coefficient of determination, whereas the overpenalization in GBT models may suggest that outlier removal is appropriate.



**Figure 9:** Gradient boosted tree results for regression and classification. Response surface plot (left) and prediction accuracy (right) of the model.

**Figure 10** summarizes the results of all the ML methods tested with this dataset. The best and worst regression methods were GP and GBT, respectively, and the best and worst classification models were GBT and the NN, respectively. More on the results and their generalizations in the following discussion.



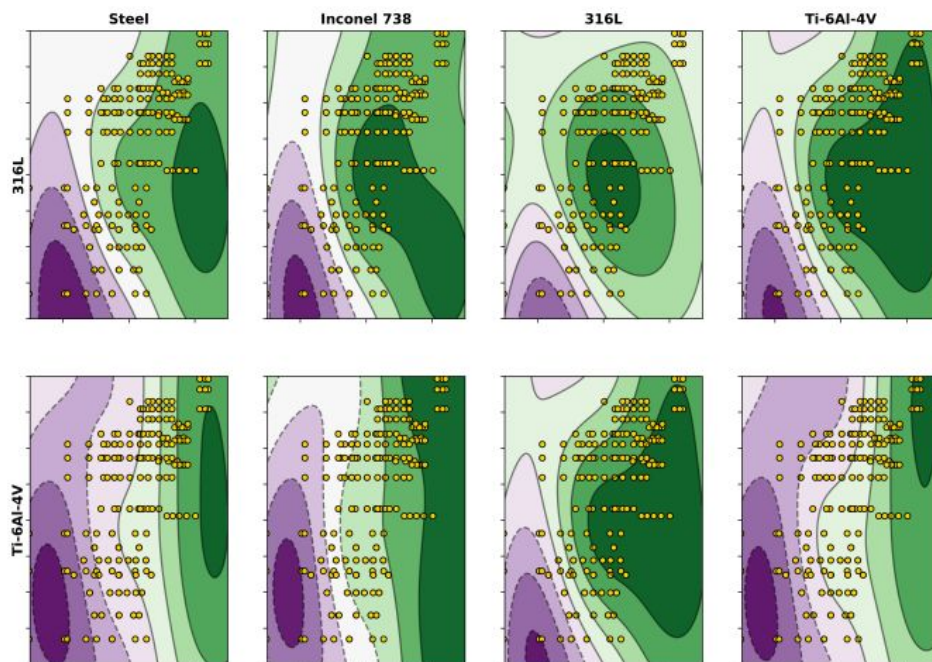


**Figure 10:** Summary of results for regression and classification models from this study with a comparison to literature results (top).

## Discussion

The purpose of this study was to determine whether ML-based prediction methods can be used effectively to predict single clad geometry in DED. Considering that these ML models were able to process a dataset containing data from different sources, with varying materials, and including as many input features as available, all while achieving a  $>70\%$  average accuracy, should indicate that these methods do have a place when examining experimental, tabular data. Which ML method is preferred for analysis of tabular data appears, at least at this juncture, to be dataset dependent. This is evidenced by the results summary shown in fig. X, where there are pros and cons for each method. As a holistic approach to all DED single clad data, ML techniques significantly outperform the empirical-statistical models from literature. This in no way should be read as a criticism of the work and results of those studies. Those studies achieved good results with a single substrate and single clad material. The point here is that the lack of generalization capability in an empirical-statistical model is something that can be leveraged in a

ML type framework. An interpretation of what this study has shown is that roughly 70% of clad dynamics is rooted in the process itself and is material invariant. To further illustrate the preceding point, also available within the ML framework is the ability to interpolate between materials (**fig. 11**).

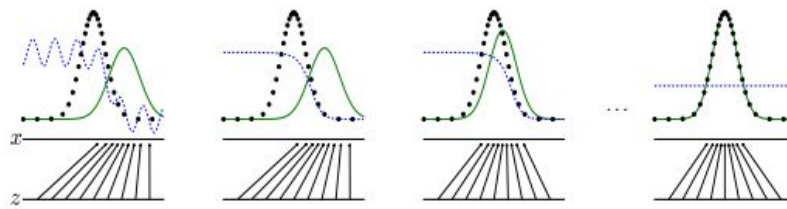


**Figure 11:** Result of material interpolation within an ML-model. The rows of the figure represent substrate materials while the columns of the figure represent cladding materials.

The ability to interpolate between materials allows researchers to reduce the time and cost of developing process maps for new material combinations. Further analysis of these models may yield stable manifolds of process variables that result in good quality clads independent of material.

Pragmatically, there is no suggestion here that the information contained within this study be followed to the letter. The choice of linear heat input as the independent variable, and aspect ratio as the dependent variable, is purely based on the portrayal of results which can be easily compared. There are artifacts of linear heat input which obfuscate the intent of a process parameter map. It is suggested that these ML techniques be used in conjunction with raw experimental parameters rather than combined parameters for analysis and development of individualized process parameter maps. The code to perform exact analysis of this study is provided in the github repository (available on GitHub @ [github.com/mjjuhasz/Canvasback.git](https://github.com/mjjuhasz/Canvasback.git) filename is `DED_Process_Parameter.ipynb`), and that code can be easily amended for individual needs.

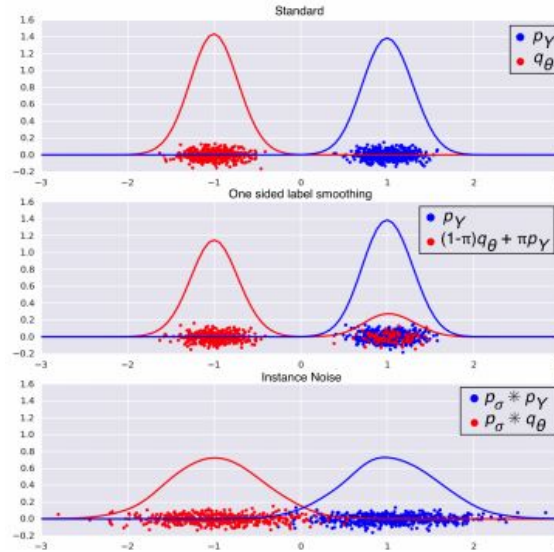
Introduced back in 2014 by Goodfellow et al. [68], generative adversarial networks (GANs) consist of two neural networks stuck in a minimax game against each other. The Generator network takes latent space noise ( $z$ ) as input and outputs into the native variable space ( $X$ ). The Discriminator network then takes the Generator data as well as traditionally sampled data in  $X$  as an input and is tasked with discriminating between the two data sources. Notionally the Discriminator network decides which data is “real” and which is “fake”. As these networks train, the Generator becomes better and better at fooling the Discriminator. In the case of the Generator, this training amounts to forming a Bayesian estimator on the sample distribution and is shown graphically in **fig. 12**. There are, of course, limits to success of either network in this minimax game as outlined by Goodfellow et al [68].



**Figure 12:** Underlying mechanism of GANs. Generator learns the distribution of the dataset in question by mapping the latent space ( $z$ ) into the experimental variable space ( $x$ ) [68].

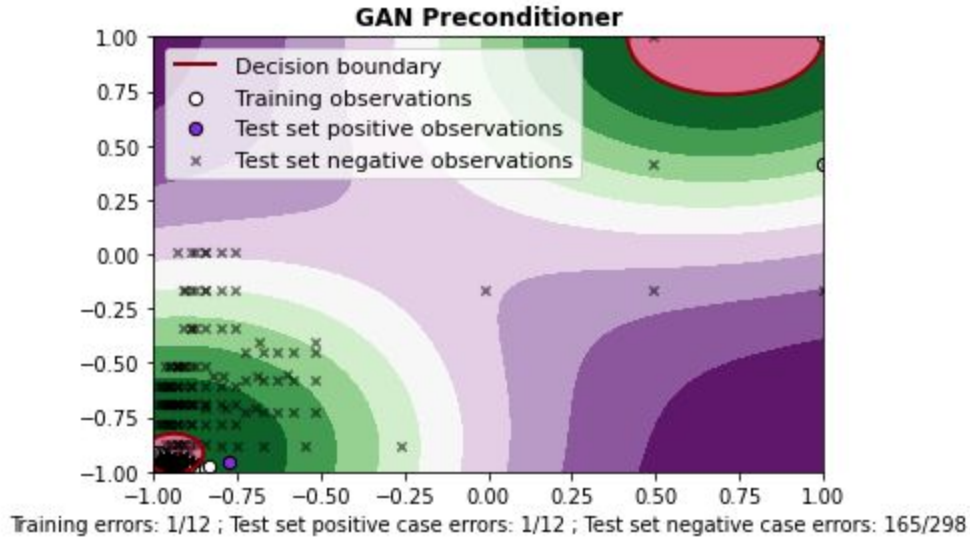
The theory related to GANs suggests that they should be readily convergent. However, personal experience and other authors [69,70] reveal that GANs are indeed highly unstable during the training process. When a mismatch between theory and practice exists, a thorough evaluation of assumptions made in the model can be enlightening. As examined in Sønderby et al [71], the violated assumption made by the GAN architecture is the supposition that there exists some overlap in the distributions at the start of the training process. Without this important overlap, the Kullback-Liebr divergence becomes uninformative as a loss function, leading to the instability. Sønderby et al [71] suggests the addition of noise, termed “instance noise”, to both distributions, which effectively creates the required overlap. This noise can then be annealed out for training of the GAN as originally intended.





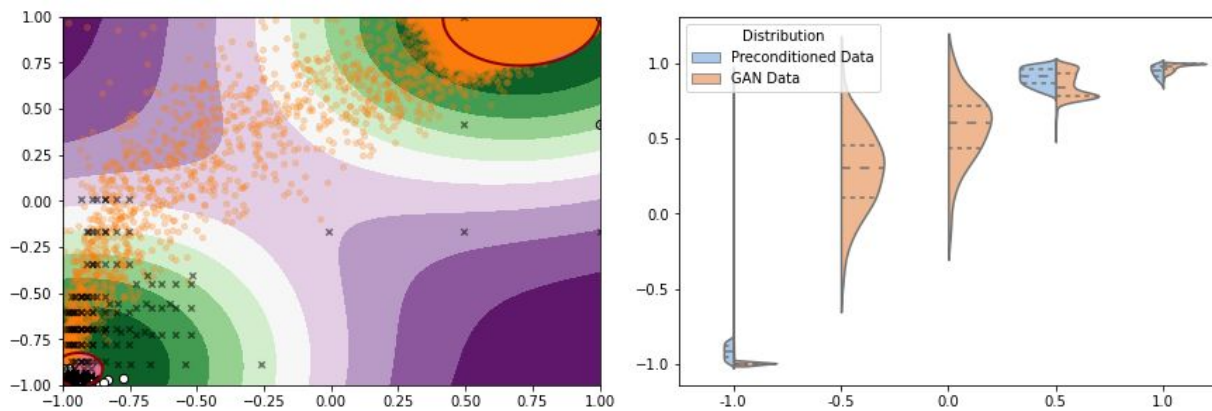
**Figure 13:** Instance noise and its effect on the Generator and Discriminator distributions as outlined by Sønderby et al [71].

Even in light of the addition of instance noise, experience during this investigation has also shown GANs to be difficult to achieve consistent results on tabular data. After removing all randomness associated with weight initialization in the model, the inconsistencies persisted. Typically, GANs are applied to image data, and it was eventually the power of the convolutional prior that is a common feature in image processing neural networks. The convolutional prior acts to highlight important features of an image, which in turn allows the Discriminator to discern between real and fake data at the outset. The inconsistencies seen in tabular data GANs originated from the condition that, at the beginning of training, the Discriminator has no ability to distinguish real from fake data. This muddled state of the Discriminator means that most, if not all, of the Generator data passes as real, and the training of the GAN as a whole fails. In order to approximate the inductive prior of the convolution, first the dataset is processed using a single class SVM or isolation forest method. The previous listed methods are used for detection of outliers/anomalies/novelties, depending on the literature. The results (**fig. 14**) of either of these two methods have two purposes: (1) to augment the data by increasing the size of the dataset, which has been shown to increase the performance of ML-based techniques, and (2) to provide the inductive prior, or preconditioner, needed to give the Discriminator network an initial representation to enable it to distinguish between real and fake data.



**Figure 14:** Results of one-class SVM data preconditioner. This data is then used to pre-train the Discriminator network before GAN training.

Once this pre-training of the Discriminator is complete, training of the GAN network, using instance noise and the augmented data, can commence. It remains an open question whether the augmented data is required during this phase of the training. The larger augmented dataset was used because of the known advantages of NNs on larger datasets. This phase of the training ends when after the instance noise is annealed out using Generator gradients as a metric to progress the annealing. The last phase is what would be considered traditional GAN training as outlined by Goodfellow [68], the results of which are shown in **fig. 15**.



**Figure 15:** (Left) Figure 14 with overlay of 50k points sampled from the Generator. (Right) Slices taken at (-1, -0.5, 0, 0.5, 1) from both the preconditioned data, and the resulting GAN distribution.

**Figure 15** shows the distributions of the pre-trained Generator and the final Generator model received from the GAN training. The incredible aspect of GANs in general is that the Generator arrives at these distributions without ever processing, or seeing, any experimental data. The results can be viewed in two ways: (1) as an alternative to traditional neural networks as a means at arriving at a process model, or (2) as a means to examine the variability of the distribution to be used as a feedback loop to guide further experimentation. This second perspective can be incorporated as part of an experimentation methodology, in essence directing further experiments to investigate regions of low support of the GAN distribution. This could result in fewer individual tests being required to understand a given phenomena, which decreases the cost of science. This is an exciting and yet open question in combining machine learning with the physical sciences, upon which hopefully the aforementioned GAN strategy begins to shed light.

## Conclusions

This work provides insights into implementation of machine learning techniques on low quantity, high variance datasets. This hopefully gives a framework for applying these techniques to other experimental datasets. The general goal is to reduce the cost per data point in the experimental sciences. Specific conclusions shown are:

- Regression performance that exceeds performance based on proposed combined parameters as reported in the literature.
- An ensemble regression accuracy of 70.5% and an ensemble classification accuracy of 72.3% across the entire dataset.
- Correlative sub-manifolds within the models tested which generalize successful deposition process parameters between materials.
- Generative adversarial networks may provide novel insights into experimental uncertainty and also provide a mechanism to guide future research.

## Acknowledgements

I'd like to acknowledge Jason Walker, Ph.D., C. Virgil Solomon, Ph.D., and Brett Conner, Ph.D. for the needed push and encouragement to publish this work. Pedro Cortes, Ph.D. for the initial discussions which led me "down the rabbit hole" and into machine learning methods. Lastly, I need to thank John B. Juhasz, Esq. for editorial help and making sense of my original drafts.

I'd like to also acknowledge the Python [72,73] community for the tools and libraries which made this work possible. This includes Scikit-Learn [74], Tensorflow [75], Keras [76], NumPy

[77,78], Pandas [79], Matplotlib [80], Seaborn [81], and Jupyter [82]. A huge thank you goes out to the creators and contributors in this community.

## References

- [1] E. Toyserkani, A. Khajepour, S.F. Corbin, Laser Cladding, CRC Press, 2004.
- [2] J.E. Ward, Automatic Programming of Numerically Controlled Machine Tools, MASSACHUSETTS INST OF TECH CAMBRIDGE ELECTRONIC SYSTEMS LAB, 1959.  
<https://apps.dtic.mil/dtic/tr/fulltext/u2/241754.pdf>.
- [3] J. Francis Reintjes, Numerical Control: Making a New Technology, Oxford University Press, 1991.
- [4] P. Smid, CNC Programming Handbook: A Comprehensive Guide to Practical CNC Programming, Industrial Press Inc., 2003.
- [5] J.W. Giachino, W.R. Weeks, Welding skills, American Technical Publishers, 1985.
- [6] P.M. Dickens, M.S. Pridham, R.C. Cobb, I. Gibson, G. Dixon, Rapid prototyping using 3-D welding, in: 1992 International Solid Freeform Fabrication Symposium, repositories.lib.utexas.edu, 1992.  
<https://repositories.lib.utexas.edu/handle/2152/64409>.
- [7] C. Atwood, M. Ensiz, D. Greene, M. Griffith, L. Harwell, Laser engineered net shaping (LENS (TM)): A tool for direct fabrication of metal parts, (1998). <https://www.osti.gov/biblio/1549>.
- [8] G.K. Lewis, E. Schlienger, Practical considerations and capabilities for laser assisted direct metal deposition, Mater. Des. 21 (2000) 417–423.
- [9] M.L. Griffith, L.D. Harwell, J.T. Romero, Multi-material processing by LENS, 1997 International. (1997). <https://repositories.lib.utexas.edu/handle/2152/71406>.
- [10] N. Shamsaei, A. Yadollahi, L. Bian, S.M. Thompson, An overview of Direct Laser Deposition for additive manufacturing; Part II: Mechanical behavior, process parameter optimization and control, Additive Manufacturing. 8 (2015) 12–35.
- [11] S.M. Thompson, L. Bian, N. Shamsaei, A. Yadollahi, An overview of Direct Laser Deposition for additive manufacturing; Part I: Transport phenomena, modeling and diagnostics, Additive Manufacturing. 8 (2015) 36–62.
- [12] A.R. Nassar, T.J. Spurgeon, E.W. Reutzel, Sensing defects during directed-energy additive manufacturing of metal parts using optical emissions spectroscopy, in: Solid Freeform Fabrication Symposium Proceedings, University of Texas Austin, TX, 2014.  
<http://sffsymposium.engr.utexas.edu/sites/default/files/2014-024-Nassar.pdf>.
- [13] M. Qian, F.H. Froes, Titanium Powder Metallurgy: Science, Technology and Applications, Elsevier Science, 2015.
- [14] J.M. Flynn, A. Shokrani, S.T. Newman, V. Dhokia, Hybrid additive and subtractive machine tools--Research and industrial developments, Int. J. Mach. Tools Manuf. 101 (2016) 79–101.
- [15] K.A. Lorenz, J.B. Jones, D.I. Wimpenny, M.R. Jackson, A review of hybrid manufacturing, in: Solid Freeform Fabrication Conference Proceedings, sffsymposium.engr.utexas.edu, 2015.  
<http://sffsymposium.engr.utexas.edu/sites/default/files/2015/2015-8-Lorenz.pdf>.
- [16] J. Keist, K. Taminger, T.A. Palmer, Structure-Property Correlations for Additively Manufactured Ti-6Al-4V Components Produced Using Directed Energy Deposition Processes, in: V. Venkatesh, A.L. Pilchak, J.E. Allison, S. Ankem, R. Boyer, J. Christodoulou, H.L. Fraser, M.A. Imam, Y. Kosaka, H.J. Rack, A. Chatterjee, A. Woodfield (Eds.), Proceedings of the 13th World Conference on Titanium, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2016: pp. 1395–1400.
- [17] V.K. Balla, S. Bose, A. Bandyopadhyay, Processing of Bulk Alumina Ceramics Using Laser Engineered Net Shaping, Int. J. Appl. Ceram. Technol. 5 (2008) 234–242.
- [18] V.K. Balla, P.D. DeVasConCellos, W. Xue, S. Bose, A. Bandyopadhyay, Fabrication of

- compositionally and structurally graded Ti--TiO<sub>2</sub> structures using laser engineered net shaping (LENS), *Acta Biomater.* 5 (2009) 1831–1837.
- [19] T. DebRoy, H.L. Wei, J.S. Zuback, T. Mukherjee, J.W. Elmer, J.O. Milewski, A.M. Beese, A. Wilson-Heid, A. De, W. Zhang, Additive manufacturing of metallic components--process, structure and properties, *Prog. Mater. Sci.* (2017).  
<https://www.sciencedirect.com/science/article/pii/S0079642517301172>.
- [20] J.J. Lewandowski, M. Seifi, Metal Additive Manufacturing: A Review of Mechanical Properties, *Annu. Rev. Mater. Res.* 46 (2016) 151–186.
- [21] I. Gibson, D. Rosen, B. Stucker, Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing, Springer New York, 2016.
- [22] A.J. Pinkerton, Advances in the modeling of laser direct metal deposition, *J. Laser Appl.* 27 (2015) S15001.
- [23] J.M. Pelletier, M.C. Sahour, M. Pilloz, A.B. Vannes, Influence of processing conditions on geometrical features of laser claddings obtained by powder injection, *J. Mater. Sci.* 28 (1993) 5184–5188.
- [24] E. Toyserkani, A. Khajepour, S. Corbin, Three-dimensional finite element modeling of laser cladding by powder injection: Effects of powder feedrate and travel speed on the process, *J. Laser Appl.* 15 (2003) 153–160.
- [25] T. Amine, J.W. Newkirk, F. Liou, An investigation of the effect of direct metal deposition parameters on the characteristics of the deposited layers, *Case Studies in Thermal Engineering.* 3 (2014) 21–34.
- [26] J.C. Heigel, P. Michaleris, E.W. Reutzel, Thermo-mechanical model development and validation of directed energy deposition additive manufacturing of Ti–6Al–4V, *Additive Manufacturing.* 5 (2015) 9–19.
- [27] W. Ya, B. Pathiraj, S. Liu, 2D modelling of clad geometry and resulting thermal cycles during laser cladding, *J. Mater. Process. Technol.* 230 (2016) 217–232.
- [28] F. Wirth, K. Wegener, A physical modeling and predictive simulation of the laser cladding process, *Additive Manufacturing.* 22 (2018) 307–319.
- [29] R. Colaco, T. Carvalho, R. Vilar, Laser cladding of stellite 6 on steel substrates, *High Temperature Chemical.* 3 (1994) 21–29.
- [30] X. Wu, B. Zhu, X. Zeng, X. Hu, K. Cui, Critical state of laser cladding with powder auto-feeding, *Surf. Coat. Technol.* 79 (1996) 200–204.
- [31] M. Qian, L.C. Lim, Z.D. Chen, W.I. Chen, Parametric studies of laser cladding processes, *J. Mater. Process. Technol.* 63 (1997) 590–593.
- [32] U. de Oliveira, V. Ocelík, J.T.M. De Hosson, Analysis of coaxial laser cladding processing conditions, *Surf. Coat. Technol.* 197 (2005) 127–136.
- [33] G. Chryssolouris, S. Zannis, K. Tsirbas, C. Lalas, An Experimental Investigation of Laser Cladding, *CIRP Ann.* 51 (2002) 145–148.
- [34] V. Ocelík, U. de Oliveira, M. de Boer, J.T.M. de Hosson, Thick Co-based coating on cast iron by side laser cladding: Analysis of processing conditions and coating properties, *Surf. Coat. Technol.* 201 (2007) 5875–5883.
- [35] H. El Cheikh, B. Courant, S. Branchu, J.-Y. Hascoët, R. Guillén, Analysis and prediction of single laser tracks geometrical characteristics in coaxial laser cladding process, *Opt. Lasers Eng.* 50 (2012) 413–422.
- [36] Z. Wang, T.A. Palmer, A.M. Beese, Effect of processing parameters on microstructure and tensile properties of austenitic stainless steel 304L made by directed energy deposition additive manufacturing, *Acta Mater.* 110 (2016) 226–235.
- [37] M. Ansari, R. Shoja Razavi, M. Barekat, An empirical-statistical model for coaxial laser cladding of

- NiCrAlY powder on Inconel 738 superalloy, *Opt. Laser Technol.* 86 (2016) 136–144.
- [38] M. Erfanmanesh, H. Abdollah-Pour, H. Mohammadian-Semnani, R. Shoja-Razavi, An empirical-statistical model for laser cladding of WC-12Co powder on AISI 321 stainless steel, *Opt. Laser Technol.* 97 (2017) 180–186.
- [39] M. Nabhani, R.S. Razavi, M. Barekat, An empirical-statistical model for laser cladding of Ti-6Al-4V powder on Ti-6Al-4V substrate, *Opt. Laser Technol.* 100 (2018) 265–271.
- [40] S. Mondal, B. Tudu, A. Bandyopadhyay, P.K. Pal, Process optimization for laser cladding operation of alloy steel using genetic algorithm and artificial neural network, *International Journal of Computational Engineering Research.* 2 (2012) 18–24.
- [41] S. Mondal, A. Bandyopadhyay, P.K. Pal, Application of artificial neural network for the prediction of laser cladding process characteristics at Taguchi-based optimized condition, *Int. J. Adv. Manuf. Technol.* 70 (2014) 2151–2158.
- [42] M. Khanzadeh, S. Chowdhury, M. Marufuzzaman, Porosity prediction: Supervised-learning of thermal history for direct laser deposition, *Journal of Manufacturing.* (2018). <https://www.sciencedirect.com/science/article/pii/S0278612518300402>.
- [43] M. Khanzadeh, S. Chowdhury, M.A. Tschopp, In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes, *IJSE.* (2019). <https://www.tandfonline.com/doi/abs/10.1080/24725854.2017.1417656>.
- [44] E. Toyserkani, A. Khajepour, A mechatronics approach to laser powder deposition process, *Mechatronics* . 16 (2006) 631–641.
- [45] E.-W. Bai, An optimal two-stage identification algorithm for Hammerstein--Wiener nonlinear systems, *Automatica* . 34 (1998) 333–338.
- [46] H.S. Chang, M.C. Fu, J. Hu, S.I. Marcus, Google Deep Mind's AlphaGo, *OR/MS Today.* 43 (2016) 24–29.
- [47] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W.M. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, Others, Alphastar: Mastering the real-time strategy game starcraft ii, *DeepMind Blog.* (2019) 2.
- [48] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, *arXiv [cs.CL].* (2020). <http://arxiv.org/abs/2005.14165>.
- [49] B. Meredig, Five High-Impact Research Areas in Machine Learning for Materials Science, *Chem. Mater.* 31 (2019) 9579–9581.
- [50] R.K. Vasudevan, K. Choudhary, A. Mehta, R. Smith, G. Kusne, F. Tavazza, L. Vlcek, M. Ziatdinov, S.V. Kalinin, J. Hattrick-Simpers, Materials Science in the AI age: high-throughput library generation, machine learning and a pathway from correlations to the underpinning physics, *MRS Commun.* 9 (2019). <https://doi.org/10.1557/mrc.2019.95>.
- [51] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature.* 323 (1986) 533–536.
- [52] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- [53] L. Bottou, Stochastic gradient descent tricks, *Neural Networks: Tricks of the Trade.* (2012). [https://link.springer.com/chapter/10.1007/978-3-642-35289-8\\_25](https://link.springer.com/chapter/10.1007/978-3-642-35289-8_25).
- [54] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, *arXiv [cs.LG].* (2014). <http://arxiv.org/abs/1412.6980>.
- [55] Y.A. LeCun, L. Bottou, G.B. Orr, K.R. Müller, Efficient backprop, *Neural Networks: Tricks of the.* (2012). [https://link.springer.com/chapter/10.1007/978-3-642-35289-8\\_3](https://link.springer.com/chapter/10.1007/978-3-642-35289-8_3).
- [56] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, *Classification and regression trees*, (1984).

- [https://scholar.google.ca/scholar?cluster=3017989415557051150&hl=en&as\\_sdt=0,5&scioldt=0,5](https://scholar.google.ca/scholar?cluster=3017989415557051150&hl=en&as_sdt=0,5&scioldt=0,5).
- [57] Y. Freund, L. Mason, The alternating decision tree learning algorithm, Academia.edu. (n.d.).  
<https://www.academia.edu/download/11015335/atrees.pdf>.
  - [58] M. Seeger, Relationships between Gaussian processes, support vector machines and smoothing splines, Mach. Learn. (2000).  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.5095&rep=rep1&type=pdf>.
  - [59] B. Schölkopf, A.J. Smola, F. Bach, Learning with kernels: support vector machines, regularization, optimization, and beyond, (2002).  
<https://books.google.ca/books?hl=en&lr=&id=y8ORL3DWt4sC&oi=fnd&pg=PR13&ots=bLBW5wN5EG&sig=JGjTtMBM9m8p2H99ZwVMBkqaQm8>.
  - [60] T. Hofmann, B. Schölkopf, A.J. Smola, Kernel Methods in Machine Learning, Ann. Stat. 36 (2008) 1171–1220.
  - [61] C.K.I. Williams, C.E. Rasmussen, Gaussian Processes for Regression, in: D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, MIT Press, 1996: pp. 514–520.
  - [62] C.E. Rasmussen, Gaussian Processes in Machine Learning, in: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004: pp. 63–71.
  - [63] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. (1995).  
<https://link.springer.com/article/10.1007%252FBF00994018>.
  - [64] V. Vapnik, The nature of statistical learning theory, (2013).  
<https://books.google.ca/books?hl=en&lr=&id=EgqACAAQBAJ&oi=fnd&pg=PR7&ots=g3M1fvbW16&sig=3l866LDWoSvAHUR6QmBM5agrYIM>.
  - [65] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222.
  - [66] L. Massaron, Deep Learning for Tabular Data - Integrating Tensorflow/Keras and Scikit-Learn Worlds, (2019). [https://drive.google.com/file/d/1hoetupySNgf4yh-UBtWN\\_26GCMYcQj0P/view](https://drive.google.com/file/d/1hoetupySNgf4yh-UBtWN_26GCMYcQj0P/view) (accessed March 15, 2020).
  - [67] J. McGhee, Deep Learning for Tabular Data: A Bag of Tricks, (2020).  
<https://community.datarobot.com/t5/general-blog/deep-learning-for-tabular-data-a-bag-of-tricks/ba-p/4593>.
  - [68] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014: pp. 2672–2680.
  - [69] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv [cs.LG]. (2015). <http://arxiv.org/abs/1511.06434>.
  - [70] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in Neural Information Processing Systems, papers.nips.cc, 2016: pp. 2234–2242.
  - [71] C.K. Sønderby, J. Caballero, L. Theis, W. Shi, F. Huszár, Amortised MAP Inference for Image Super-resolution, arXiv [cs.CV]. (2016). <https://arxiv.org/abs/1610.04490>.
  - [72] G. Van Rossum, F.L. Drake Jr, Python tutorial, Centrum voor Wiskunde en Informatica Amsterdam, 1995.
  - [73] G. Van Rossum, F.L. Drake Jr, Python reference manual, Centrum voor Wiskunde en Informatica Amsterdam, 1995.
  - [74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Others, Scikit-learn: Machine learning in Python, The Journal

- of Machine Learning Research. 12 (2011) 2825–2830.
- [75] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, (2015). <http://tensorflow.org/>.
  - [76] F. Chollet, Others, Keras, (2015). <https://keras.io>.
  - [77] T.E. Oliphant, A guide to NumPy, Trelgol Publishing USA, 2006.
  - [78] S. van der Walt, S.C. Colbert, G. Varoquaux, The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science Engineering. 13 (2011) 22–30.
  - [79] W. McKinney, Others, Data structures for statistical computing in python, in: Proceedings of the 9th Python in Science Conference, Austin, TX, 2010: pp. 51–56.
  - [80] J.D. Hunter, Matplotlib: A 2D Graphics Environment, Comput. Sci. Eng. 9 (2007) 90–95.
  - [81] M. Waskom, O. Botvinnik, D. O’Kane, P. Hobson, mwaskom/seaborn: v0. 8.1 (September 2017), Zenodo. (2017).
  - [82] T. Kluyver, B. Ragan-Kelley, F. Pérez, B.E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J.B. Hamrick, J. Grout, S. Corlay, Others, Jupyter Notebooks-a publishing format for reproducible computational workflows, in: ELPUB, books.google.com, 2016: pp. 87–90.