JOURNAL OF
EMERGING INVESTIGATORS

Article

Minimizing distortion with additive manufacturing parts using Machine Learning

Shreya Yarlagadda[1], Emily Broadhurst[2]
[1] Valley Christian High School, San Jose, California
[2] Stanford University, Stanford, California

SUMMARY

Additive manufacturing technologies are increasingly being used in both academic and corporate settings as an efficient and precise method of manufacturing. However, the accuracy of affordable 3D printing methods is compromised by distortion of the plastic filament. These inaccurate 3D prints may not be usable; incorrect sizing of specific features, such as holes, can mean the part does not fit in the larger assembly and would need to be discarded. This can contribute to plastic pollution due to mismanagement, compounded by difficult access to the recycling process of the most popular filament, Polylactic Acid (PLA). In this study, we investigated to what extent the distortion introduced in the additive manufacturing process can be predicted and minimized. We hypothesized that predicting the computer-aided design (CAD) dimensions based on historical print misalignments would be effective in minimizing this distortion. We used a histogram-based gradient boosting regressor and trained it on 3D printing data of hole dimensions printed in PLA filament, obtained from Mendeley Data. Our model predicts the CAD geometry that results in the desired final print dimension, accounting for distortion. The model had a root mean squared error of 0.0614 mm, a mean absolute error of 0.0452 mm, and an R2 score of 99.617%. These results can be used to minimize inaccuracies and prevent waste production.

INTRODUCTION

3D printing is evolving in its use by many big companies to produce safe and reliable products and is developing in the fields of medicine, defense, engineering, and others (1). However, since accurate technologies are being monopolized or involve complex processes, those available for academic or prototyping purposes are susceptible to distortion and failure (2). Especially with widely used 3D printers, the plastic filament is prone to sizing mistakes which can lead to warping of the parts, making them unusable. For example, the printer may continue its job on top of a distorted piece, creating an unviable product. Although manual tolerancing — printing larger dimensions to account for distortion during the printing process — occurs, it is estimated and thus not a reliable way to achieve the desired geometry. Additionally, proper recycling of this waste is hard to manage and difficult to access, often resulting in mishandling and consequently, pollution.

Polylactic acid (PLA), a biobased polymer, is the most used 3D printing material (3). However, when it isn't recycled properly, it degrades slowly and contributes to the large amount of plastic pollution present (4,5). Plastic consumption has quadrupled over the past 30 years and 22% of this plastic is mismanaged (6). Prior studies have investigated using recycled PLA as the material in 3D printing for a more sustainable alternative to the directly produced virgin PLA (7). The use of recycled PLA mixed with unused PLA and additives, such as copper microparticles and organic silanes used to improve the PLA's performance and processability, would be more cost-effective and sustainable compared to pure PLA, however, further study is needed to determine the most sustainable and strong mixture of recycled PLA, unused PLA, and these aforementioned additives. Additionally, Teruel et al. investigated using predictive models to reduce the variability of recycled polypropylene plastic and Barreto et al. explored the potential of rice husk as a filler for PLA filaments (8,9). These findings are promising for opening new avenues for study, but still need to be investigated for longevity over multiple bouts of recycling and their effectiveness in implementation for various applications in 3D printing. Rather than making the material more sustainable, in this study, we wanted to provide a direct solution to the plastic waste issue by reducing the distortion to parts during the printing process, leading to less waste produced overall.

Several studies have been conducted concerning print distortion and failure reduction by using AI models to monitor print quality in-situ (during the manufacturing process) (10-14). Neural networks have been developed using audio emissions and vision systems to classify the print quality for each layer printed, achieving high accuracies (80-90%), however they do not provide solutions for real-time changes that should be made to stabilize the print quality (10-12). In response, other research has explored using state-of-the-art object detection algorithms and vision systems with neural networks to estimate adjustments to the process parameters and input values, such as altering the drive voltage of Liquid Metal Jet Printing to control ink volume, reaching high accuracies (13,14). This research can affect the print quality, but the complex technology involved is not accessible or applicable to the general public and most academic settings.

Previous studies suggest that using in-situ monitoring is either not effective in real-time scenarios for improving the quality of prints, or the technology necessary to make it effective is inaccessible for those without a large budget (10-14). Therefore, in-situ monitoring would not cut down on waste produced from failed or unusable prints in most cases. In response, recent studies have focused on predicting the print parameters of a 3D print before the manufacturing process that optimize a certain factor of the part (15-17). Neural networks have been used to optimize the print parameters

https://doi.org/10.59720/24-086

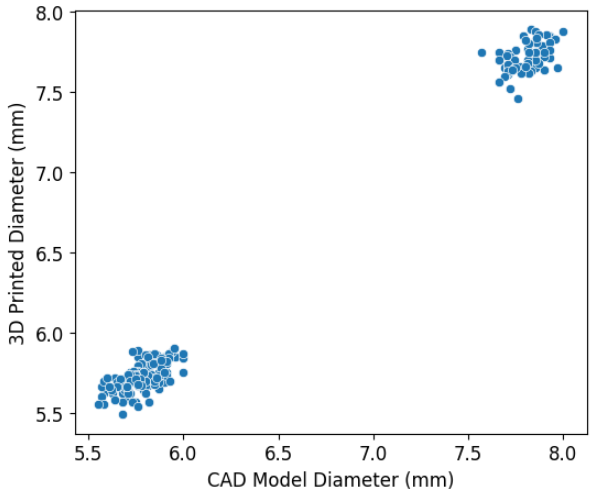

Figure 1: Visualization of dataset comparing the CAD model diameter to 3D printed diameter of holes. Scatterplot of the hole samples (n=180) with CAD diameters (mm) before printing on the x-axis and PLA 3D printed diameters (mm) on the y-axis. The values were obtained from the original processed dataset.

for greater strength, hardness, surface finish, and tensile strength of both recycled and virgin PLA parts (15, 16). Additionally, Nelaturi et al. used a printability map that simulated the printing process to evaluate the expected structural deviations before manufacturing for distortion correction, however, did not maintain the original dimensions (17).

We aimed to add to the previous research by using an effective and cost-efficient machine learning model to predict dimensions instead of parameters in the 3D print and optimize the precision of the print. We hypothesized that predicting the CAD dimensions that should be printed, accounting for historical print misalignments, would be effective in minimizing the distortion. To test this, we utilized a supervised learning regression model, the histogram-based gradient boosting regressor, which outputs the predicted CAD geometry of holes based on the desired final print dimension and numerical parameters (perimeter speeds, vertical shells, and layer height), accounting for any distortion during the print process. Our model trained well on the data and had near negligible error, showing the potential of AI to minimize distortion in 3D printed PLA parts. In academic and prototyping settings, this model can be used as a more precise form of estimated tolerancing, reducing distortion and printing waste.

RESULTS

To determine if machine learning could be applied to predict the optimal CAD geometry, we utilized a dataset from Mendeley Data, which included information about printed surfaces with holes in PLA using a Prusa MK4 3D printer. The recorded data were CAD dimensions of holes (mm), the final printed dimensions in PLA (mm), and the print parameters: vertical shells, layer height (mm), and external, small, and overall perimeter speeds (mm/s). We picked the histogram-based gradient boosting regressor after testing it on our dataset in comparison with other models. We then tuned several of the hyperparameters of this model (max_iter, max_depth, learning_rate, and max_leaf_nodes), optimizing the mean squared error of the model with 80% training, 10% validation, and 10% testing. We dropped the small and external perimeter speeds from the input matrix since they did not affect the results of the model, then created our final model.

When visualizing our dataset, we observed a large gap in the data, as holes with CAD dimensions between around 6.2 and 7.6 mm were not used in the samples (**Figure 1**). We observed some differences in the range and centers of distortion when we graphed the distortion against the values of the vertical shell, layer height, and perimeter speed values. This is due to the differences in the printing process associated with the specific aforementioned parameters that could affect how much the melted plastic spread into the hole. For example, if the perimeter speed is higher, the plastic may not have as much time to cool before another layer is extruded on top of it. Additionally, by comparing histograms of the hole dimensions before and after 3D printing, we saw a noticeable difference in the distributions of observations in the higher range of values, indicating that larger dimensions experience more variability between the CAD dimensions and final printed dimensions (**Figure 2**).

We used the input parameters (layer height, vertical shells, and perimeter speeds) and the CAD dimensions of the holes to predict the necessary dimensions to be 3D printed accounting for distortion. When choosing the type of model used, we compared the metric scores of the models on the three validation sets (5%, 10%, and 20% splits of the data) to observe which had the best results with the dataset. For instance, the histogram-based gradient boosting regressor's best results using 80% training, 10% validation, and 10% testing were a 0.0499 mm MAE, 0.0550 mm RMSE, and 99.3925% R^2 score. In comparison, the linear regression's metric scores

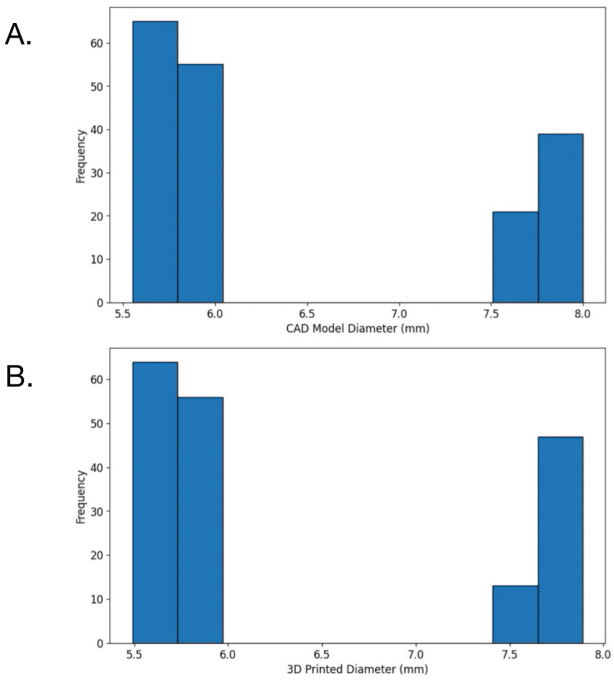

Figure 2: Histogram visualizations of CAD model diameter and 3D printed diameter data. Histograms of the frequency of (A) CAD diameter (mm) and (B) 3D printed diameter (mm) ranges of the 180 samples from the original dataset.

https://doi.org/10.59720/24-086

	Dropped Parameters								
	None	Vertical Shells	Layer Height (mm)	Perimeter Speed (mm/s)	Small Perimeter Speed (mm/s)	External Perimeter Speed (mm/s)	Final Printed Diameter (mm)	All Perimeter Speeds (mm/s)	Small & External Perimeter Speeds (mm/s)
MAE (mm)	0.04992	0.05726	0.05980	0.04992	0.04992	0.04992	0.82435	0.04692	0.04992
MSE (mm2)	0.00303	0.00399	0.00468	0.00303	0.00303	0.00303	0.72124	0.00302	0.00303
R^2 Score	0.99392	0.99199	0.99062	0.99392	0.99392	0.99392	-0.44671	0.99393	0.99392

Table 1: Metric scores for each dropped input feature. The mean absolute error values (mm), mean squared error (mm2), and R2 when dropping each input parameter from the input matrix for the model using 80% training, 10% validation, and 10% testing before hyperparameter tuning. After the input matrix was re-defined to exclude the indicated parameter(s), the model was trained using the training set and the error values in the table were from testing on the validation set.

with the same data split were a 0.0744 mm MAE, 0.0833 mm RMSE, and 98.6070% R^2 score. As another example, the random forest regressor's metric scores with the same data split were a 0.0529 mm MAE, 0.0612 mm RMSE, and 99.2481% R^2 score.

We also created a table of the model's metric values resulting from individually dropping each feature from the input matrix of the model (perimeter speeds, layer height, vertical shells, and printed diameter), which showed us that dropping the external and small perimeter speeds (mm/s) from the input did not affect the performance of the model, so those parameters were unnecessary and were dropped from the input matrix (**Table 1**). Then, when hyperparameter tuning, we created a graph of the error corresponding to each hyperparameter value for each validation set and were able to observe the curves the MSE went through and the optimal value for each hyperparameter, for example with the max depth hyperparameter (**Figure 3**).

Our final model was the histogram-based gradient boosting regressor, which after hyper parameter tuning had a

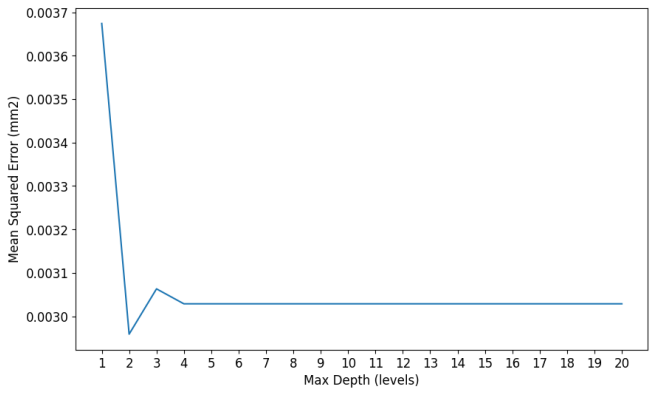

Figure 3: Graph of mean squared error for each value of the hyperparameter max depth. Graph of the MSE values (mm^2) for each value of the max depth parameter in terms of the levels (which limits the amount of iterations of the decision trees) used in the model, from 1 to 20 levels, for the model using 80% training, 10% validation, and 10% testing. The model was tested using the validation set to optimize the max depth for the best mean squared error.

0.0452 mm mean absolute error (MAE), 0.0614 mm root mean squared error (RMSE), and 99.617% R2 score. The small error values indicated that our model accurately predicts the CAD diameters needed to print a particular dimension with specified parameters. Typical tolerance values for PLA range between 0.125 and 0.5 mm, depending on the type of fit desired (loose, standard, or tight) (18). Therefore, error values of 0.0452 and 0.0614 mm are not likely to affect the usefulness of the part.

From the training loss curves, the mean squared error (MSE) decreased with more iterations of training (epochs), showing that the model effectively identifies patterns in the data. Towards the later epochs, the training loss curves all converge and stay constant, indicating the model has learned all it can from the data at that point. We also observed that the validation loss curves are very similar in shape and distribution to the training loss curves, which indicates our models can learn from the training data and accurately generalize the learned patterns to the validation dataset, which it has previously not been trained on (**Figure 4**). We were assured that our models were not over or underfitted to the data through our observations of these loss curves; the data trains the models well to make accurate predictions on the training datasets, and the models are able to generalize those established relationships to make accurate predictions on the validation sets as well.

We visualized predictions of the model on the testing set, noting that they were not linear and were affected not only by the printed diameter but also the other input features. For example, several data points with a printed dimension of 5.7 mm had differing predicted CAD dimensions due to the other parameters (vertical shells, layer height, and perimeter speeds) (**Figure 5**). We also visually examined the error of our final model by plotting the testing values of the final printed dimensions with the corresponding residuals of the CAD geometry. The residuals appeared random, apart from the gap from around 6.0 to 7.5 mm introduced because our original dataset did not include printed dimensions between those values (**Figure 6**). The absence of a pattern in the residual plot signals that the relationship established by our histogram-based gradient boosting regressor is adequate in predicting the CAD dimensions, and we do not have evidence of systematic error. We observed a few outliers above and below most residuals

https://doi.org/10.59720/24-086

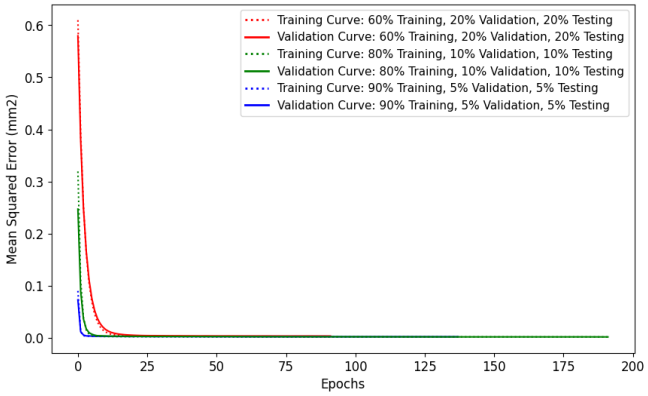

Figure 4: Loss curves of the histogram-based gradient boosting regressor. Line graph of the mean squared error values for each iteration of the model through the dataset, for the training and validation sets (92, 138, and 192 epochs). Dotted curves represent the training curves and solid curves represent the validation curves for each training, validation, and testing split, separated by color. Training and validation sets were randomly selected from the original dataset and were used to train and tune the models. Mean squared error is is the error resultant from the models' predictions on the training and validation sets.

in this plot (higher than 0.1 or lower than -0.1); however, the majority of the residuals are between -0.05 to 0.05 mm, which is not concerning given that PLA can also flex to a certain extent and fit into the larger part if it is slightly distorted (**Figure 6**).

Additionally, we performed a k-fold cross validation to evaluate the performance of the model when faced with different training and testing data. After performing the 10-fold cross validation, the mean RMSE was 0.0632 mm with a standard deviation of 0.0082 mm. The mean MAE was 0.0507 mm with a standard deviation of 0.0095 mm. The mean R2 score was 99.558% with a standard deviation of 0.1017%. The low standard deviations compared with the means told us that our model was consistent with new training and testing datasets. Additionally, the original results are within 1 standard deviation of the mean scores, which is not unusual, so we can be further assured that our model did not overfit to the specific patterns of the validation dataset when tuning hyperparameters.

DISCUSSION

We believe the model's good performance was due to the clustering of data points in a small range, as well as the strong relationship between CAD geometry and parameters and the final print dimensions. We suggest that the histogram-based gradient boosting regressor performed the best due to the "ensemble" learning method it employs, similar to random forest regression. Because both methods combine several decision trees, the analysis of the patterns in the dataset is more complex, but accurate, which led to both methods performing relatively well when their metric scores were observed before hyperparameter tuning. However, the random forest regressor uses decision trees in parallel, which do not interact with each other, while the histogram-based gradient boosting regressor uses decision trees that iterate and improve on each other, leading to more precise results for certain datasets like ours. One limitation, however, is that contrary to a linear regressor or neural network, the histogram-based gradient boosting regressor only considers the patterns observed within the range of the dataset, so if extrapolated, it will not be able to predict accurately and will assume the same prediction as the nearest input value inside the range.

Therefore, one aspect to take into consideration is the gap in data between about 6.2 and 7.5 millimeters, where there are no holes tested within this range of diameters. Due to the danger of extrapolation, we cannot assume that the observed patterns exist within that range, and we are limited by our data. Additionally, the data only contained 180 samples, and the range of values (5.5-8 mm) was not very large. In general, data within the field of additive manufacturing is sparse, so the collection of more data to build better models would be helpful.

Additionally, the model using 90% training, 5% validation, and 5% testing data did not yield the highest accuracy despite having a large training dataset, which we believe is due to the resulting small validation set. Small validation sets could skew the hyperparameters to represent a minority of the dataset and may not result in the most effective tuning for the whole dataset. On the other hand, we think the model using 60% training, 20% validation, and 20% testing data did not have as much training data as the model with 80% training, 10% validation, and 10% testing to identify patterns of the dataset. This is why we believe the final model (80% training, 10% validation, 10% testing) had a balance between training and tuning success through the training and validation datasets, leading to good performance on the testing dataset.

A natural expansion to this research on the relationship between CAD dimensions with printing parameters of holes and the final printed dimensions is an investigation of similar relationships with other CAD geometries or optimization of different filaments as well. For example, ABS (Acrylonitrile Butadiene Styrene) or TPU (Thermoplastic Polyurethane) are other popular filaments that could be optimized for distortion. Other CAD geometries to be investigated include different hole shapes, slots, grooves, and overhangs, as these geometries are also frequently distorted. Additionally, developing a web application or add-on to CAD technologies that would

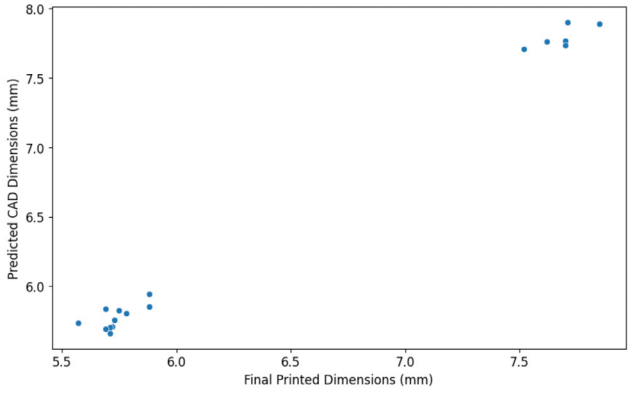

Figure 5: Scatterplot of model predictions for the printed dimensions. Scatterplot showing the predicted CAD dimensions, in mm, of the holes in testing set (n=18, 10% of the total dataset) corresponding to the printed dimensions (mm). The model was trained on the training data (80% of total dataset) and the hyperparameters were tuned on the validation set (10% of the total dataset) before predicting on the testing data shown.

https://doi.org/10.59720/24-086

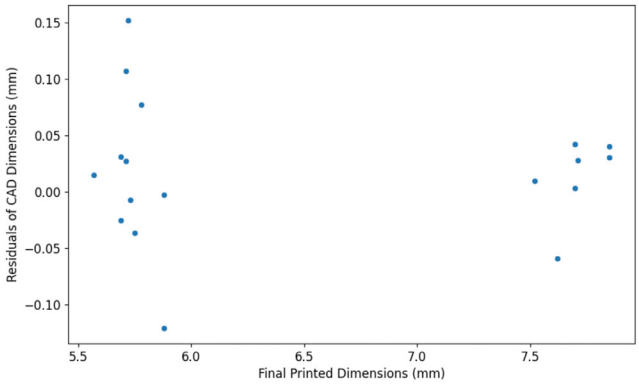

Figure 6: Randomly distributed residuals from the final model for each printed dimension in the testing set. Residual scatterplot showing the final dimensions (mm) from the testing set (n=18) in relation to the CAD dimension residuals (mm), which are the true CAD dimensions from the dataset minus the predicted CAD dimensions from the output of the model. The model was trained on the training data (80% of total dataset) and the hyperparameters were tuned on the validation set (10% of the total dataset) before predicting on the testing data shown.

output the dimension to print based on an input of the desired dimension and layer height, vertical shells, and perimeter speeds could provide ease of use and a practical application of the findings, instead of the manual estimated tolerancing that occurs today. Particularly with different materials and complex designs, an automated process accounting for distortion would increase efficiency in prototyping and product creation. If such a method were implemented and available to the larger public, the manufacturing process would be much more accurate and minimize these distorted parts, which could lead to a reduced amount of plastic waste from this field.

The primary advantage of our work compared to previous research lies in the usability of similar models as an effective, inexpensive solution to counter PLA distortion in 3D printed models. Our model's training process and performance demonstrates the effectiveness of a histogram-based gradient boosting regressor in predicting distortion from additive manufacturing of PLA parts. If applied to a real-world situation, we have evidence to suggest machine learning models like the one developed in this study can be used to account for distortion and decrease the plastic waste produced from distorted 3D printed products.

MATERIALS AND METHODS

Dataset

The dataset used came from Mendeley Data and was contributed by Diana Popescu from the Universitatea Politehnica din București (19). It includes 180 samples of holes with 7 parameters; perimeter speed (mm/s), external perimeter speed (mm/s), small perimeter speed (mm/s), vertical shells, layer height (mm), and the final printed diameter (mm) were used as input for predicting the CAD geometry (diameter in mm) in PLA parts.

Data Visualization

To visualize the dataset, we plotted the CAD diameter vs. the final printed diameter. We calculated distortion by finding the difference between the CAD dimension and the actual printed dimension. This distortion calculation method was chosen to account for the spread of melted plastic upon 3D printing holes, which usually causes the hole, and thereby the print dimension, to be smaller than it was supposed to be.
For the vertical shells, perimeter speed, and layer height, there are a few fixed values, and the data is spread evenly between them. However, since the data of the CAD dimensions and final printed dimensions are continuous with many different values included in the dataset, we also utilized histograms to observe patterns in the distribution of the hole dimension data.

Model Selection

The objective of our model was to output the CAD dimensions needed from an input of the desired final print dimension and the settings used (perimeter speeds, layer height, vertical shells) with the most precise and accurate results. We obtained 3 different training, validation, and testing datasets by splitting the original dataset into 60% training, 20% validation, and 20% testing; 80% training, 10% validation, and 10% testing; and 90% training, 5% validation, and 5% testing. We chose these percentages because they are in the range of widely accepted ratios, and present different distributions to see which distribution has the greatest balance between training the data well and not overfitting to the data.

To compare the metrics with this dataset and find the best fit for our situation, we generated multiple sklearn models using the sklearn package in python version 3.10.12 (random forest regression, linear regression, lasso regression, ridge regression, decision tree regression, support vector regression, gradient boosting regression, and histogram-based gradient boosting regression) and tested them on our three validation sets. The histogram-based gradient boosting regressor resulted in the least mean squared error and mean absolute error, and highest R^2 score when using the 80% training, 10% validation, and 10% testing ratio, so it was chosen for our model.

Model Tuning and Testing

After choosing the model, we tuned its hyperparameters for each data split by determining which values would minimize the MSE when tested on each validation set. We optimized the MSE, a metric that heavily punishes outliers, because large distortions to 3D printed parts could be undesirable and lead to harsher consequences than slightly distorted parts. We tuned the maximum depth, maximum iterations, learning rate, and maximum leaf nodes hyperparameters by conducting a grid search and writing functions for each hyperparameter that would test the mean squared error of each reasonable value (1-20 maximum depth, 50-200 maximum iterations, 0.01-0.5 learning rate, and 2-31 maximum leaf nodes) on the validation sets and show the value that minimized the mean squared error produced. Through this method, we identified the hyperparameter values that resulted in the lowest MSE.

The optimized parameters for the model using 60% training, 20% validation, and 20% testing were a max depth of 3, a learning rate of 0.2, a max iterations of 92, and a max leaf nodes of 5. The optimized parameters for the model using 80% training, 10% validation, and 10% testing were a max depth of 2, a learning rate of 0.7, a max iterations of 138, and a

https://doi.org/10.59720/24-086

max leaf nodes of 4. The optimized parameters for the model using 90% training, 5% validation, and 5% testing were a max depth of 3, a learning rate of 0.42, a max iterations of 192, and a max leaf nodes of 6.

We observed the error values and R^2 score for each of the three models to find which way of splitting the data was most effective for our dataset. We chose the model with the least error and highest R^2 score, which was the model using 80% training, 10% validation, and 10% testing from the dataset, although the model using 60% training, 20% validation, and 20% testing had similar scores of 0.0455 mm MAE, 0.0614 mm RMSE, and 99.578% R^2 score.

We also tested the effects of each input feature (perimeter speeds, layer height, vertical shells, and printed diameter) on the performance of the model by removing them one at a time from the input matrix and observing the change in the resulting model metric outputs (prior to hyperparameter tuning). We observed that excluding all three of the perimeter speed parameters from the X matrix slightly decreased the error of the un-tuned model and increased the R^2 score, therefore improving its performance when compared with the un-tuned model including all the input values (**Table 1**). However, tuning the hyperparameters for this new X matrix resulted in our model excluding perimeter speeds to have a higher error and a lower R^2 score (worse results) than the model including perimeter speed on the test set, with a 0.0503 mm MAE, 0.0639 mm RMSE, and 99.585% R^2 score. In other words, keeping the perimeter speed categories in the X matrix helped the model's tuned hyper parameters accurately reflect the patterns in the dataset. So, since dropping two perimeter speed parameters had no effect on the metric scores of the model, we dropped the external perimeter speed (mm/s) and small perimeter speed (mm/s) from the X matrix (**Table 1**). A final model was created with the dataset split, tuned hyperparameters, and dropped parameters to observe the final metrics and accuracy of the model, which gives the CAD geometry needed for the desired print dimension to be achieved with the print parameters.

To evaluate the performance of our model with different training and testing data, we conducted a k-fold cross validation using the sklearn package in python. This procedure splits the dataset into k (a number that is chosen) folds, and each fold can be used as the testing set while the others are used to train the model. Therefore, the model will be trained and tested with k different variations and the metric scores can be observed to determine how skilled the model is with new train and test sets. We chose a k value of 10 because it is a widely accepted value that results in scores with low bias.

ACKNOWLEDGMENTS

We would like to thank the Inspirit AI Research Fellowship program for guiding us through this process.

Received: March 04, 2024
Accepted: July 05, 2024
Published: January 15, 2025

REFERENCES

1. "The 3-D Printing Revolution." *Harvard Business Review*, 16 Nov. 2015, hbr.org/2015/05/the-3-d-printing-revolution. Accessed 06 Feb. 2024.
2. Lohr, Steve. "3-D Printing Grows beyond Its Novelty Roots." *The New York Times*, 3 July 2022, www.nytimes.com/2022/07/03/business/3d-printing-vulcan-forms.html. Accessed 06 Feb. 2024.
3. Alsop, Thomas. "Most Used 3D Printing Materials Worldwide 2018." *Statista*, 2 Mar. 2020, www.statista.com/statistics/800454/worldwide-most-used-3d-printing-materials/. Accessed 06 Feb. 2024.
4. McKeown, Paul, and Matthew D. Jones. "The Chemical Recycling of PLA: A Review." *Sustainable Chemistry*, vol. 1, no. 1, 2 May 2020, pp. 1–22, https://doi.org/10.3390/suschem1010001.
5. Royer, Sarah-Jeanne, et al. "Not so Biodegradable: Polylactic Acid and Cellulose/Plastic Blend Textiles Lack Fast Biodegradation in Marine Waters." *PLOS ONE*, vol. 18, no. 5, 24 May 2023, https://doi.org/10.1371/journal.pone.0284681.
6. "Plastic Pollution Is Growing Relentlessly as Waste Management and Recycling Fall Short, Says OECD" *Organisation for Economic Co-Operation and Development*, www.oecd.org/environment/plastic-pollution-is-growing-relentlessly-as-waste-management-and-recycling-fall-short.htm. Accessed 06 Feb. 2024.
7. Hasan, Mohammad Raquibul, et al. "Potential of recycled PLA in 3D printing: A Review." *Sustainable Manufacturing and Service Economics*, vol. 3, 2024, p. 100020, https://doi.org/10.1016/j.smse.2024.100020.
8. Teruel, Ricardo, et al. "Machine learning model and input batch management tool for the composition of new recycled polypropylene plastic material with reduced variability in target properties." *Journal of Cleaner Production*, vol. 435, Jan. 2024, p. 140390, https://doi.org/10.1016/j.jclepro.2023.140390.
9. Barreto, Gabriela, et al. "Rice husk with PLA: 3D filament making and additive manufacturing of samples for potential structural applications." *Polymers*, vol. 16, no. 2, 15 Jan. 2024, p. 245, https://doi.org/10.3390/polym16020245.
10. Shevchik, S.A., et al. "Acoustic Emission for in Situ Quality Monitoring in Additive Manufacturing Using Spectral Convolutional Neural Networks." *Additive Manufacturing*, vol. 21, May 2018, pp. 598–604, https://doi.org/10.1016/j.addma.2017.11.012.
11. Qi, Xinbo, et al. "Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives." *Engineering*, vol. 5, no. 4, Aug. 2019, pp. 721–729, https://doi.org/10.1016/j.eng.2019.04.012.
12. Zhang, Yingjie, et al. "Extraction and Evaluation of Melt Pool, Plume and Spatter Information for Powder-Bed Fusion AM Process Monitoring." *Materials & Design*, vol. 156, 20 July 2018, pp. 458–469, https://doi.org/10.1016/j.matdes.2018.07.002.
13. Zubayer, Md Hasib, et al. "Enhancing additive manufacturing precision: Intelligent Inspection and Optimization for defect-free continuous carbon fiber-reinforced polymer." *Composites Part C: Open Access*, vol. 14, July 2024, p. 100451, https://doi.org/10.1016/j.jcomc.2024.100451.
14. Wang, Tianjiao, et al. "In-Situ Droplet Inspection and Closed-Loop Control System Using Machine Learning for Liquid Metal Jet Printing." *Journal of Manufacturing Systems*, vol. 47, 15 May 2018, pp. 83–92, https://doi.

https://doi.org/10.59720/24-086

org/10.1016/j.jmsy.2018.04.003.
15. Wei, Hanjun, et al. “Optimizing FDM 3D printing parameters for improved tensile strength using the takagi–sugeno fuzzy neural network.” *Materials Today Communications*, vol. 38, Mar. 2024, p. 108268, https://doi.org/10.1016/j.mtcomm.2024.108268.
16. Kumar PVR, Girish, and Devaki Devi K. “Optimizing Mechanical Properties of virgin and recycled PLA components using ANOVA and neural networks.” *Mathematical Models in Engineering*, vol. 10, no. 1, 29 Feb. 2024, pp. 1–10, https://doi.org/10.21595/mme.2023.23630.
17. Nelaturi, Saigopal, et al. “Manufacturability feedback and model correction for additive manufacturing.” *Journal of Manufacturing Science and Engineering*, vol. 137, no. 2, 1 Apr. 2015, https://doi.org/10.1115/1.4029374.
18. “3D Printing Tolerances - Ankermake US.” *Ankermake,* 28 Apr. 2024, www.ankermake.com/blogs/guides/3d--printing-tolerances.
19. Popescu, Diana. “Datasets Comparison.” Dataset Comparison - *Mendeley Data*, 12 Dec. 2022, data.mendeley.com/datasets/compare/4h6ttzh9bf. Accessed 06 Feb. 2024.
20. Pedregosa et al. “Scikit-learn: Machine Learning in Python.» *JMLR 12*, pp. 2825-2830, 2011