https://doi.org/10.48047/AFJBS.6.12.2024.5106-5119

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Prediction Of Surface Roughness In 3D Printing Based Additive Manufacturing with Machine Learning

Manikyam Sandeep*, Dr Ganapathi Raju Nadimpalli, Dr Swetha Perumalla, Dr G Krishna Mohana Rao, Gosula Suresh

Department of ME, ACE Engineering College, Hyderabad,&https://orcid.org/0000-0002-0110-5539
Department of IT, GRIET, Hyderabad, Telangana, India&https://orcid.org/0000-0003-2076-000X
Department of CSE, JNTUH, Hyderabad, Telangana, India&https://orcid.org/0009-0001-0057-3592
Department of ME, JNTUH, Hyderabad, Telangana, India&https://orcid.org/0000-0002-0506-3444
Department of ME, ACE Engineering College, Hyderabad, &https://orcid.org/0009-0004-8694-7402
Corresponding Author:sandeep.mech109@gmail.com

Article History
Volume 6, Issue 12, 2024
Received: June 10, 2024
Accepted: July 5, 2024
doi:
10.48047/AFJBS.6.12.2024.5106-5119

Abstract
In today's industry, machine learning (ML) and additive manufacturing (AM) are revolutionary technologies. By modelling surface roughness based on thermal analysis and predicting its surface roughness value using machine learning, this work attempts to improve the surface quality of 3D printed objects. The optimization of key parameters such as layer height (LH), printing speed (PS), nozzle temperature (NT), and infill density (ID) will take place.ML algorithms such random forest regressor, XG-Boost, support vector machines, and linear regression can be used to make the prediction. The PLA+ material characterization will also be looked at.To analyze parameter effects, experiments employ Taguchi's Design of Experiment with orthogonal array, and machine learning methods will be used to determine which model is the most correct.The work focusses on LH, ID, PS, NT, and platform temperature as the five input parameters that affect layer geometries. By optimizing AM processes, advanced machine learning algorithms seek to improve the surface quality of 3D printed items.
Key Words: 3D Printing, Taguchi Method, Regression, Artificial intelligence, Machine Learning- Google Colab, Regression

1.Introduction

Additive Manufacturing (AM), also known as 3D printing, represents a revolutionary approach to fabricating objects by layering materials in a step-by-step fashion. Originating in the mid-1980s with the advent of advanced stereolithography (SL) techniques, A Mhas since evolved to encompass various methodologies such as laminated object manufacturing, fused deposition modeling, and 3D printing. Despite its transformative potential, the high initial costs of AM machines have limited accessibility for medium and small enterprises.

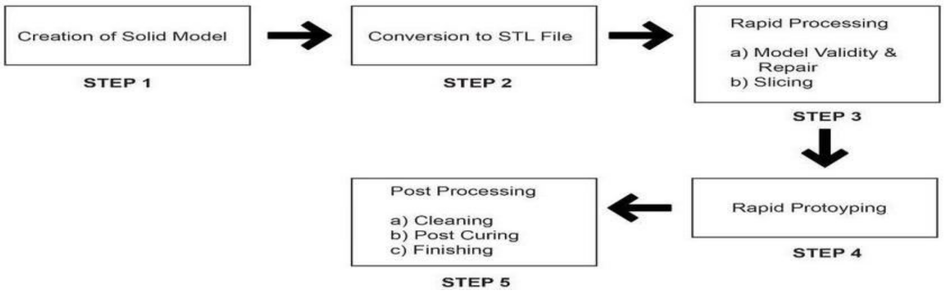


Fig1.1**:**Step by step procedure in Rapid Prototyping process

2.Literature Review

Sufyan Ghani et al. [1] A notable development in the realm of civil engineering is the use of machine learning techniques (MLTs) to forecast the compressive strength (C) of self-compacting concrete (SCC). Using well-known artificial intelligence methods such artificial neural networks (ANN), adaptive neuro-fuzzy inference systems (ANFIS), and extreme learning machines (ELM), the study methodically assessed six MLTs. **ItaiLishneretal.**[2]A class of machine learning methods called artificial neural networks is used to model nonlinear relations in datasets. The architecture of the ANN is made up of several layers, including activation functions and hidden layers. The artificial neurone is the fundamental unit of an ANN. A mathematical function called an artificial neurone has individual weighted inputs, and the sum is routed through a transfer function to an output link. **Ajanwachuku NwaguChimaetal.**[3]Artificial Neural Networks (ANN) have proven to be highly valuable in a wide range of prediction situations, including several domains including medical, banking, meteorology, stock markets, engineering, and cybersecurity. The current increase in research provides evidence of the adaptability and efficiency of Artificial Neural Networks (ANN) in enhancing the accuracy of predictions. This distinguishing feature sets ANN apart from other methods used for forecasting..**SiddardhaKoramatietal.**[4]This study represents a notable advancement in the utilisation of machine learning, particularly artificial neural networks (ANN), for forecasting urban accidents within the context of Hyderabad, India. The study employs an extensive crash record obtained from Hyderabad police data covering the period from 2015 to 2019.**Terpenny et al.** [5] This study provides an extensive analysis of machine learning data processing and management in the context of additive manufacturing (AM) research and applications. The evaluated publications over the past four years provide a summary of the data handling methods used for four key types of data: tabular data, graphic data, 3D data, and spectrum data.The primary techniques for handling data include feature extraction, discretisation, data processing, feature selection, and feature learning. The utilisation of machine learning techniques in a wide range of additive manufacturing applications has been observed.

3.Materials And Methods

3.1. Material Selection:PLA+ (Enhanced PLA): For this study, PLA+ is the material of choice due to its superior properties over standard PLA. PLA+ offers increased strength, durability, and heat resistance, making it ideal for experimental work in 3D printing.

3.2. Taguchi Design of Experiments (DOE) Method:

The Taguchi DOE method is a statistical approach used to optimize processes and improve quality by systematically varying parameters and analyzing their effects. This method reduces the number of experiments needed by using orthogonal arrays, making it both efficient and cost-effective.

SI. No	PS	NT	ID	LH	SR
1	50	200	20	0.1	2.833
2	50	210	25	0.12	3.922
3	50	215	30	0.14	4.825
4	50	220	35	0.16	5.885
5	50	225	40	0.18	5.1
6	60	200	25	0.14	6.408
7	60	210	30	0.16	5.606
8	60	215	35	0.18	4.536
9	60	220	40	0.1	5.421
10	60	225	20	0.12	4.202
11	70	200	30	0.18	4.788
12	70	210	35	0.1	5.263
13	70	215	40	0.12	4.717
14	70	220	20	0.14	5.321
15	70	225	25	0.16	5.347
16	80	200	35	0.12	4.523
17	80	210	40	0.14	4.732
18	80	215	20	0.16	4.98

					5
19	80	220	25	0.18	4.673
0	80	225	30	0.1	5.051
21	90	200	40	0.16	5.692
22	90	210	20	0.18	4.897
23	90	215	25	0.1	4.606
24	90	220	30	0.12	5.427
25	90	225	35	0.14	5.351

Table3.1: Taguchi values from Minitab software

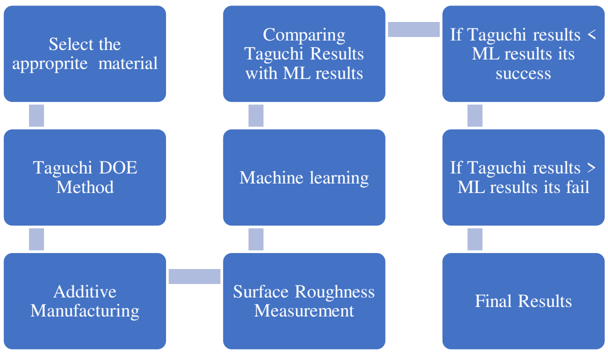


Fig3.1:Experimentation Process

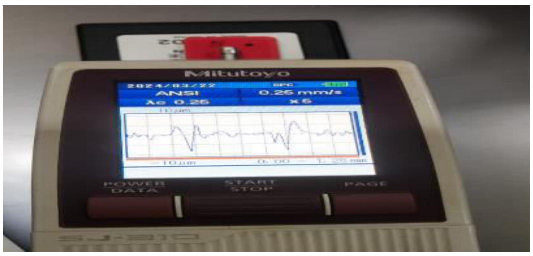


Fig4.1Mitutoyo Surface roughness test SJ-210

4. Surface Roughness Measurement:
Surface roughness, a key quality parameter of 3D-printed objectsis quantitatively measured using profilometers, with the Mitutoyo machine employed for this purpose.

5.**Machine Learning:** Machine learning involves the use of algorithms & statistical models to analyze and predict. Here is the Correlation Matrix for Which parameter mainly affects the output surface roughness values, which is shown in Figure 2. The Figure 3 showing the Metric Representation which mean that what algorithm has a lower error than others.

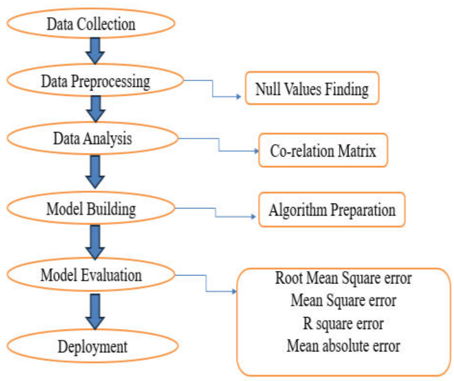


Fig5.1: Process of machine learning

STEP 1: Data collection

SI. No	PS	NT	ID	LH	SR
1	50	200	20	0.1	2.833
2	50	210	25	0.12	3.922
3	50	215	30	0.14	4.825
4	50	220	35	0.16	5.885
5	50	225	40	0.18	5.1
6	60	200	25	0.14	6.408
7	60	210	30	0.16	5.606
8	60	215	35	0.18	4.536
9	60	220	40	0.1	5.421
10	60	225	20	0.12	4.202
11	70	200	30	0.18	4.788

12	70	210	35	0.1	5.26 3
13	70	215	40	0.1 2	4.71 7
14	70	220	20	0.1 4	5.32 1
15	70	225	25	0.1 6	5.34 7
16	80	200	35	0.1 2	4.52 3
17	80	210	40	0.1 4	4.73 2
18	80	215	20	0.1 6	4.98 5
19	80	220	25	0.1 8	4.67 3
20	80	225	30	0.1	5.05 1
21	90	200	40	0.1 6	5.69 2
22	90	210	20	0.1 8	4.89 7
23	90	215	25	0.1	4.60 6
24	90	220	30	0.1 2	5.42 7
25	90	225	35	0.1 4	5.35 1

Table 5.1 : Input and output of the Taguchi

Fig 5.2 ML data

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[45] df = pd.read_excel("ML DATA.xlsx")
```

```
[46] df.head()
```

	PS	NT	ID	LH	SR
0	50	200	20	0.10	2.833
1	50	210	25	0.12	3.922
2	50	215	30	0.14	4.825
3	50	220	35	0.16	5.885
4	50	225	40	0.18	5.100

Next steps: Generate code with df View recommended plots

PS –Printing Speed; NT – Nozzle Temperature ;ID – Infill Density; LH – Layer Height

```
[152] df.head()
```

	PS	NT	ID	LH	SR
0	50	200	20	0.10	2.833
1	50	210	25	0.12	3.922
2	50	215	30	0.14	4.825
3	50	220	35	0.16	5.885
4	50	225	40	0.18	5.100

```
[153] df.describe()
```

	PS	NT	ID	LH	SR
count	25.000000	25.000000	25.000000	25.000000	25.000000
mean	70.000000	214.000000	30.000000	0.140000	4.964440
std	14.433757	8.779711	7.216878	0.028868	0.701375
min	50.000000	200.000000	20.000000	0.100000	2.833000
25%	60.000000	210.000000	25.000000	0.120000	4.673000
50%	70.000000	215.000000	30.000000	0.140000	4.985000
75%	80.000000	220.000000	35.000000	0.160000	5.351000
max	90.000000	225.000000	40.000000	0.180000	6.408000

STEP 2: Data Preprocessing:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   PS      25 non-null     int64
 1   NT      25 non-null     int64
 2   ID      25 non-null     int64
 3   LH      25 non-null     float64
 4   SR      25 non-null     float64
dtypes: float64(2), int64(3)
memory usage: 1.1 KB
```

```
[155] df.isnull().sum() # hence no data cleaning is required

PS    0
NT    0
ID    0
LH    0
SR    0
dtype: int64
```

Fig 5.3 ML data

STEP 3: Data Analysis

```
# Compute the correlation matrix
correlation_matrix = df.corr()

# Visualize the correlation matrix using a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".4f", linewidth = 0.2)
plt.title('Correlation Matrix')
plt.show()
# There is no strong relationship between any input para and output para
```

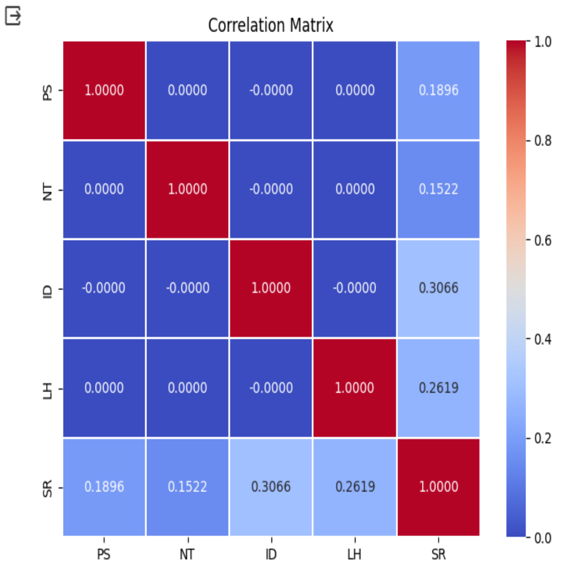
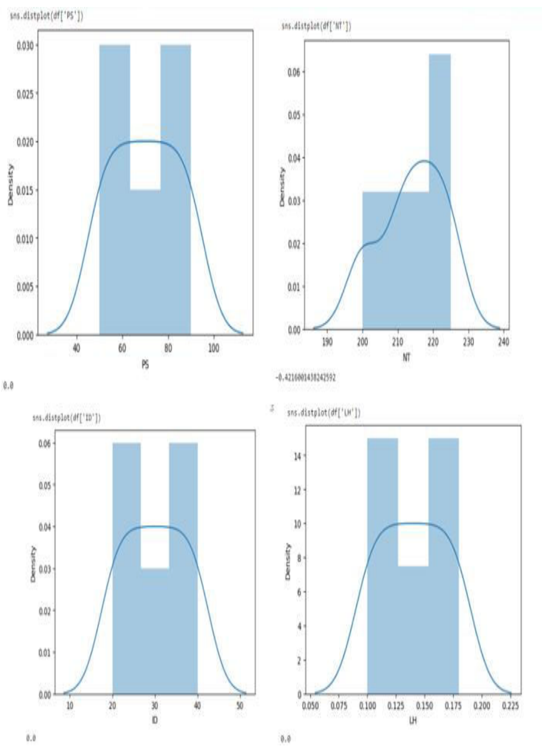


Fig no 5.4: Corelation Matrics chart

```
[211] sns.distplot(df['PS'])
      plt.show()
      print(df['PS'].skew())

      #Distribution plot
      # skews near to 0--- it follows the normal  Distribution

      sns.distplot(df['NT'])
      plt.show()
      print(df['NT'].skew())

      #Distribution plot
      # skews near to 0--- it follows the normal  Distribution

      sns.distplot(df['ID'])
      plt.show()
      print(df['ID'].skew())

      #Distribution plot
      # skews near to 0--- it follows the normal  Distribution

      sns.distplot(df['LH'])
      plt.show()
      print(df['LH'].skew())

      #Distribution plot
      # skews near to 0--- it follows the normal  Distribution
```

Fig5.5: Distribution Plot




```
[213] sns.boxplot(df['PS'],orient = 'h')
      plt.show() # input parameters only
      # Hence there are no outliers in this PS

      sns.boxplot(df['NT'],orient = 'h')
      plt.show() # input parameters only
      # Hence there are no outliers in this NT

      sns.boxplot(df['ID'],orient = 'h')
      plt.show() # input parameters only
      # Hence there are no outliers in this ID

      sns.boxplot(df['LH'],orient = 'h')
      plt.show() # input parameters only
      # Hence there are no outliers in this LH
```

Fig 5.6 Visualization ofDistribution Plot

Fig 5.7: ML data

Boxploting It is a visualization used to represent whether the outliers are present or not in this case there are no outliers for this dataset.

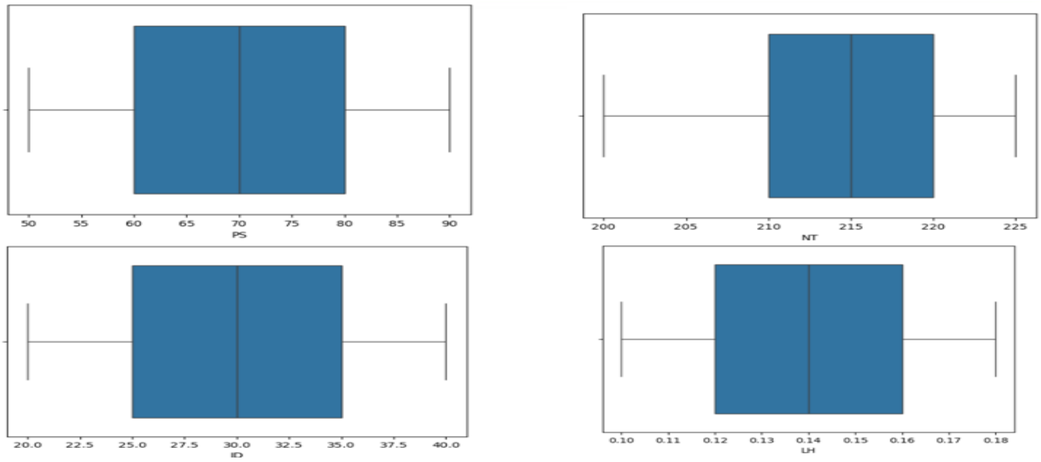


Fig 5.8 : Boxplot Representation

STEP4: Model BuildingModel building in machine learning involves creating a mathematical representation by generalizing and learning from training data. Let's break down the steps to build a machine-learning model.

Visualization:

Data visualization in machine learning is a crucial aspect that enables analysts to understand and make sense of data patterns, relationships, and trends. Let's dive into the significance of data visualization in machine learning and explore various types of visualization approaches.

```
# Assuming df is our DataFrame containing input (PS, NT, ID, LH) and output (SR) values

# Creating a pairplot to visualize the relationship between input and output variables
sns.pairplot(df, x_vars=['PS', 'NT', 'ID', 'LH'], y_vars=['SR'], kind='scatter')
plt.show()
```

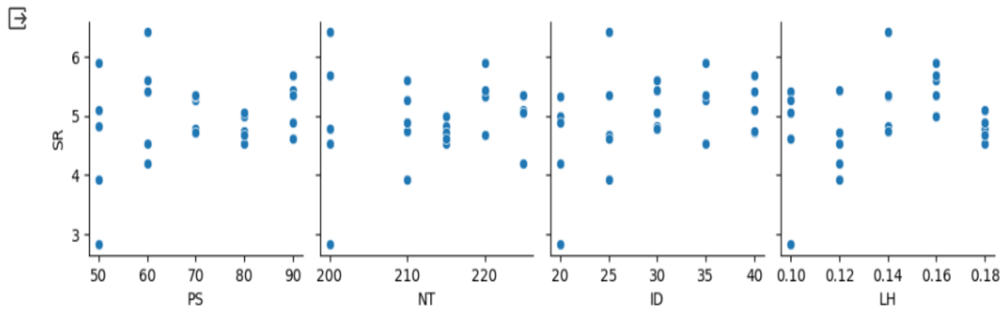


Fig 5.9: Paring plot visualization

STEP 5: MODEL EVALUATION

Model evaluation is a process of evaluating the considered algorithm which consists of how much error.

```
[182] # Initialize the regression model
      from sklearn.linear_model import LinearRegression
      linear_reg = LinearRegression()

      # Train the regression model
      linear_reg.fit(X_train, y_train)

      ▾ LinearRegression
      LinearRegression()

[185] # Make predictions
      y_pred_lin = linear_reg.predict(X_test)

[186] mse = mean_squared_error(y_test, y_pred_lin)

      # Root Mean Squared Error (RMSE)
      rmse = np.sqrt(mse)

      # Mean Absolute Error (MAE)
      mae = mean_absolute_error(y_test, y_pred_lin)


      print("Mean Squared Error:", mse)
      print("Root Mean Squared Error:", rmse)
      print("Mean Absolute Error:", mae)

      Mean Squared Error: 0.903854146986632
      Root Mean Squared Error: 0.9507124417964835
      Mean Absolute Error: 0.7570417319538109
```

```
[187] # Decision Tree Regression
      from sklearn.tree import DecisionTreeRegressor

      dt_reg = DecisionTreeRegressor(random_state = 42)

      dt_reg.fit(X_train, y_train)
      y_pred_dt = dt_reg.predict(X_test)

[188] mse = mean_squared_error(y_test, y_pred_dt)

      # Root Mean Squared Error (RMSE)
      rmse = np.sqrt(mse)

      # Mean Absolute Error (MAE)
      mae = mean_absolute_error(y_test, y_pred_dt)


      print("Decision Error Testing")

      print("Mean Squared Error:", mse)
      print("Root Mean Squared Error:", rmse)
      print("Mean Absolute Error:", mae)

      Decision Error Testing
      Mean Squared Error: 1.123931
      Root Mean Squared Error: 1.060156120578474
      Mean Absolute Error: 0.9874
```

Fig 5.10: Model EvaluationFig 5.11: Decision Tree Regression Algorithm

Decision Tree Regression:

Certainly! Here's a concise summary of Decision Tree Regression in machine learning:Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

From the algorithm, we can observe that the errors

MSE – 1.123

RMS – 1.060

MAE – 0.98

```
from sklearn.ensemble import RandomForestRegressor
# Ensemble ---> group of models working together

# Define and initialize the Random Forest model
rf_model = RandomForestRegressor(random_state=42)

# Train and evaluate each regression algorithm
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred_rf)
rmse = mean_squared_error(y_test, y_pred_rf)
mae = mean_absolute_error(y_test, y_pred_rf)

print("Random Forest Error Testing")

print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("Mean Absolute Error:", mae)

Random Forest Error Testing
Mean Squared Error: 1.0488527781200017
Root Mean Squared Error: 1.0488527781200017
Mean Absolute Error: 0.9224600000000006
```

```
[192] from sklearn.svm import SVR
      print("Support Vector Regression (SVR):")
      svr_reg = SVR()
      svr_reg.fit(X_train, y_train)
      y_pred_svr = svr_reg.predict(X_test)

      mse = mean_squared_error(y_test, y_pred_svr)
      rmse = mean_squared_error(y_test, y_pred_svr)
      mae = mean_absolute_error(y_test, y_pred_svr)

      print("Support Vector Regressor Testing")

      print("Mean Squared Error:", mse)
      print("Root Mean Squared Error:", rmse)
      print("Mean Absolute Error:", mae)

      print()

      Support Vector Regression (SVR):
      Support Vector Regressor Testing
      Mean Squared Error: 0.9623855707607373
      Root Mean Squared Error: 0.9623855707607373
      Mean Absolute Error: 0.7091014340498963
```

Fig 5.12: Random Forest Algorithm

Fig 5.13: SVR Algorithm

XG Boost Algorithm:

XGBoost, short for **extreme Gradient Boosting**, is a powerful machine-learning algorithm known for its efficiency, speed, and accuracy. It belongs to the family of **boosting**

algorithms, which are ensemble learning techniques that combine

```
print("XGBoost Regression:")
from xgboost import XGBRegressor
xgb_reg = XGBRegressor()
xgb_reg.fit(X_train, y_train)
y_pred_xgb = xgb_reg.predict(X_test)


mse = mean_squared_error(y_test, y_pred_xgb)
rmse = mean_squared_error(y_test, y_pred_xgb)
mae = mean_absolute_error(y_test, y_pred_xgb)

print("XGBoost Regression Error Testing")

print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("Mean Absolute Error:", mae)
```

```
XGBoost Regression:
XGBoost Regression Error Testing
Mean Squared Error: 0.7063381139076628
Root Mean Squared Error: 0.7063381139076628
Mean Absolute Error: 0.7972017280578612
```

Fig5.14 : XGB Algorithm

STEP 6: DEPLOYMENT

- In this case, get a surface roughness value by running the cell in Colab.
- In this case to assign the values for the new dataset.
- For each input's given case the data will be allocated into a separate row (predicted_value[0]) that row will be used for the prediction.
- Here considered XGBoost as the best algorithm from Model Evaluation.

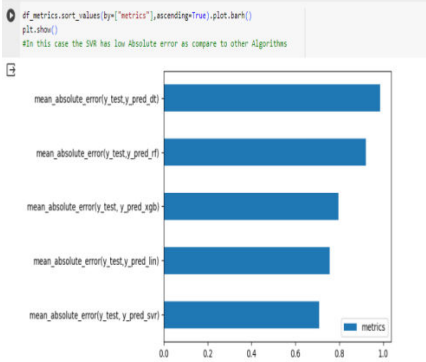
```
[216] value1=eval(input("Enter the value of Printing Speed:"))
      value2=eval(input("Enter the value of Nozel temperatue:"))
      value3=eval(input("Enter the value of Infilled Density:"))
      value4=eval(input("Enter the value of Layer Height:"))


      # Define a dictionary with input values
      new_data_dict = {'PS': value1, 'NT': value2, 'ID': value3, 'LH': value4}

      # Convert the dictionary to a DataFrame
      new_data = pd.DataFrame([new_data_dict])

      # Make predictions
      predicted_value = xgb_reg.predict(new_data)

      # Print the predicted value
      print("Predicted value:", predicted_value[0])
```

```
Enter the value of Printing Speed:50
Enter the value of Nozel temperatue:210
Enter the value of Infilled Density:25
Enter the value of Layer Height:0.12
Predicted value: 3.9237566
```

Fig 5.15: ML output Prediction Result

Mean absolute error:

Mean square error:



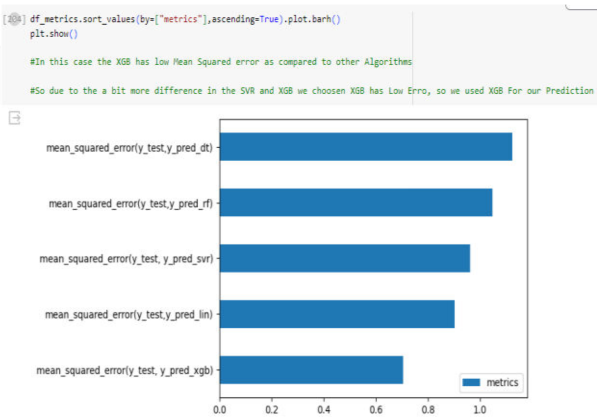


Fig 5.16: Metric Representation of MAE*Fig* 5.17: Metric Representation of MSE

5.Results And Discussion:

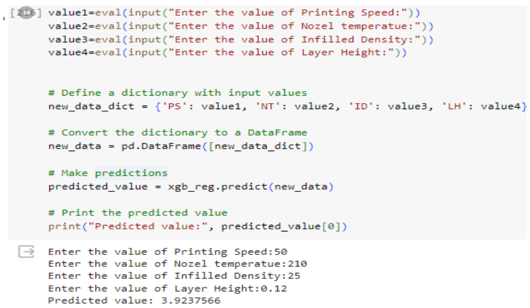
Comparison of Taguchi values and ML values

1.

If Taguchi values < Machine Learning values
Project Success

2.

If Taguchi values > Machine Learning values **Project Fail**



```
value1=eval(input("Enter the value of Printing Speed:"))
value2=eval(input("Enter the value of Nozel temperatue:"))
value3=eval(input("Enter the value of Infilled Density:"))
value4=eval(input("Enter the value of Layer Height:"))

# Define a dictionary with input values
new_data_dict = {'PS': value1, 'NT': value2, 'ID': value3, 'LH': value4}

# Convert the dictionary to a DataFrame
new_data = pd.DataFrame([new_data_dict])

# Make predictions
predicted_value = xgb_reg.predict(new_data)

# Print the predicted value
print("Predicted value:", predicted_value[0])

Enter the value of Printing Speed:50
Enter the value of Nozel temperatue:210
Enter the value of Infilled Density:25
Enter the value of Layer Height:0.12
Predicted value: 3.9237566
```

Figure 6.1. Algorithm used for Prediction

Prediction in Machine Learning: The dataset provided encompassed various combinations of input parameters and corresponding surface roughness values obtained from both Taguchi and Machine Learning methods. Figure 4 which represent that used XG-Boost Algorithm for prediction of surface roughness in Machine Learning.

7.Conclusion

The objective of our project was to develop a predictive model for surface roughness in Polylactic acid (PLA+) polymer material within the context of Additive Manufacturing (AM), by leveraging Machine Learning (ML) techniques. We aimed to achieve this by scrutinizing critical printing parameters, namely layer height, infill density, printing speed, and nozzle temperature.

To accomplish this, we adopted a multifaceted approach involving the integration of various ML algorithms and statistical methodologies. Specifically, we employed linear regression, support vector machine (SVM), XG-Boost, and random forest regressor algorithms.

Additionally, we incorporated Mini Tab Taguchi's Design of Experiment (L25) Method to streamline our analysis and comparison process.
Upon thorough evaluation, our results revealed that the XG Boost algorithm consistently outperformed its counterparts in terms of predictive accuracy. Moreover, when comparing the Taguchi values with those predicted by our ML models, we observed that the Taguchi values were consistently lower. This implies that our project methodology was sound and effective in generating accurate predictions for surface roughness in PLA+ polymer material through the amalgamation of ML and AM techniques.

8.References

1.Huang,M.;Jin,S.;Tang,Z.;Chen,Y.;Qin,Y.AMethodforPredictingSurfaceFinishofPolylactic AcidPartsPrintedUsingFusedDepositionModeling.*Processes*2023,*11*,1820.https://doi.org/10.3390/pr11061820

2.Ahmed,B.A.;Nadeem,U.;Hakeem,A.S.;UlHamid,A.;Khan,M.Y.;Younas,M.;Saeed,H.A.PrintingParameterOptimizationofAdditiveManufacturedPLAUsingTaguchiDesignofExperiment.Polymers2023,15,4370.https://doi.org/10.3390/polym15224370

3.Ahmad,M.N.*etal.*(2020).OptimizationonSurfaceRoughnessofFusedDepositionModelling(FDM)3DPrintedPartsUsingTaguchiApproach.In:Jamaludin,Z.,AliMokhtar,M.N.(eds)IntelligentManufacturingandMechatronics.SympoSIMM2019.LectureNotesinMechanicalEngineering.Springer,Singapore.https://doi.org/10.1007/978-981-13-9539-0_24

4. Anuj Saxenaa, Qasim Murtazaa, Paras Kumar: Prediction Of Surface Roughness In Additively Manufactured Samples In Pla+ Polymer Material Through Machine Learning

5.A.E.Tontowi,L.Ramdani,R.V.Erdizon,D.K.Baroroh:Optimization of 3D Printer Process Parameters for Improving Quality of Polylactic Acid Printed Part. https://www.researchgate.net/publication/317079668.

6. VijaypalPoonia, Rishi Kumara, Rakhee Kulshrestha, Kuldip Singh Sangwan: This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)

7. Sufyan Ghani , Nishant Kumar, Megha Gupta & Sunil Saharan: Machine learning approaches for real- time prediction of compressive strength in selfcompacting concrete Research Published: 16 December 2023.

8. Itai Lishner * and Avraham Shtub: Citation: Lishner, I.; Shtub, A. Using an Artificial Neural Network for Improving the Prediction of Project Duration. Mathematics 2022, 10, 4189. https://doi.org/10.3390/

9. R. Anitha, S. Arunachalam, and P. Radhakrishnan, “Critical parameters in uencing the quality of prototypes in fused deposition modeling.”M. Domingo-Espin, J. M. Puigoriol-Forcada.

10. C. J. Luis Pérez, “Analysis of the surface roughness and dimensional accuracy capability of fused deposition modeling processes,” Int J Prod Res, vol. 40, no. 12, pp. 2865–2881, Aug. 2002, doi: 10.1080/00207540210146099

11. Dazhong Wu, Yupeng Wei, Janis Terpenny: https://pure.psu.edu/en/publications/surface-roughness

12. Terpenny, Yupeng Wei, Dazhong Wu: Surface Roughness Prediction in Additive Manufacturing using Machine Learning: https://hal.science/hal-04096906/document

13. Portoac ˘a, A.I.; Ripeanu, R.G.; Dinit,˘a, A.; T ˘anase, M. Optimization of 3D Printing Parameters.

14. Vamsi Krishna Pasam , Surendra Babu Battula, Madar Valli P. , Swapna M.: Optimizing Surface Finish in WEDM Using the Taguchi Parameter Design Method.

15. AnifatOlawoyin*, Yangjuin Chen: Optimisation of cutting parameters during turning of 16MnCr5 steel using Taguchi technique Published: 23 June 2022

16.Manikyam, S., & Kumar, P. J. (2021). Experimental Investigation And Surface Quality Parameters Optimization In WEDM Machining Of D-Series Tool Steels. *Advances in Materials and Processing Technologies*, *8*(sup2), 814–825. https://doi.org/10.1080/2374068X.2021.1946752
17.Manikyam Sandeep, P. Jamaleswarakumar, Enhancement and gray relational analysis of EN 31 steel using parameters of wire cut electric discharge machine, Materials Today: Proceedings, Volume 19, Part 2,2019,Pages 884-889,ISSN 22147853, https://doi.org/10.1016/j.matpr.2019.08.243.
18.Sandeep, M. (2019). Experimental investigation and optimization of mrr, surface roughness and overcut of AISI 316 stainless steel in EDM using Taguchi Method. *Int J Adv Sci Technol IJAST*, *28*, 16.
19. Manikyam Sandeep and M. Sandeep 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **912** 032047**DOI** 10.1088/1757-899X/912/3/032047
20.Manikyam Sandeep, P. Jamaleswara Kumar, Optimization of analytical modelling of aluminum-multiwall carbon nanotube composites ,Materials Today: Proceedings, Volume 19, Part 2,2019,Pages 837-842,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2019.08.141.
21.Sandeep, P.Kumar,and A.Abinay, *InternationalJournalofMechanicalandProductionEngineeringResearchandDevelopment*8, 905926(2018)https://doi.org/10.24247/ijmperddec201892 Google ScholarCrossref
22.Sandeep Manikyam and P. Jamaleswara Kumar Journal: Advances in Materials and Processing Technologies, 2022, Volume 8, Number sup2, Page 814 DOI: 10.1080/2374068X.2021.1946752
23.Manikyam Sandeep,Optimization of deep drawing process parameters for cylindrical cup,Materials Today: Proceedings,Volume 19, Part 2,2019,Pages 772-777,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2019.08.128.
24.Sandeep, M., Jamaleswara Kumar, P. (2020). Experimental Execution Analysis of Wire Electric Discharge Machining. In: Praveen Kumar, A., Dirgantara, T., Krishna, P.V. (eds) Advances in Lightweight Materials and Structures . Springer Proceedings in Materials, vol 8. Springer, Singapore. https://doi.org/10.1007/978-981-15-7827-4_45
25.Manikyam Sandeep, P. Jamaleswara Kumar, Optimization of analytical modelling of aluminum-multiwall carbon nanotube composites , Materials Today: Proceedings, Volume 19, Part 2,2019,Pages 837-842, ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2019.08.141.
26. Manikyam Sandeep /Afr.J.Bio.Sc. 6(10) (2024) .Prediction Of Surface Roughness Of Pla+ Material In Additive Manufacturing Using Machine Learning. Volume 6, Issue 10, 2024 Received: 19 March 2024 Accepted: 20 April 2024 Published: 10 May 2024. https://www.afjbs.com/uploads/paper/5880dd0e5bb5443243fb12e857a9b0f0.pdf
27. Manikyam Sandeep /Afr.J.Bio.Sc. 6(10) (2024).Prediction of Machining Parameters On DB6 Tool Steel Using ANN. Volume 6, Issue 10, 2024 Received: 19 March 2024 Accepted: 20 April 2024 Published: 10 May 2024. https://www.afjbs.com/uploads/paper/a28e10d99bc3c8c04d8df45e6aecab06.pdf