🌐 ***United International Journal of Engineering and Sciences*** 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📘 *Volume 5 | Special Issue 1 | 2025 Edition*
National Level Conference on "Advanced Trends in Engineering Science & Technology" – Organized by RKCE

Mechanical Property Prediction of 3D-Printed Specimens using Machine Learning

V Srirama murthy[1], Daggubati Sunil[2], D Komali[3], Dr. Kotteda Tarun Kumar[4]
[1]Associate Professor, Dept. of Mechanical Engineering, R K college of Engineering, Vijayawada
Line [2]Assistant Professor, Dept of Mechanical Engineering, Narayana college of Engineering, Gudur,
[3]Assistant Professor, Dept. of computer science Engineering, R K college of Engineering, Vijayawada
[4]Assistant Professor, Dept of Mechanical Engineering, Sagi Rama Krishnam Raju Engineering College, Bhimavaram
vsr.308@gmail.com

***Abstract* – This project explores the prediction of mechanical properties of 3D-printed specimens using Python-based machine learning algorithms. Additive manufacturing (AM) offers flexibility in creating customized and complex parts, but predicting mechanical properties such as tensile strength, hardness, and elasticity remains challenging due to the variability in printing parameters, material types, and environmental conditions. Leveraging Python's powerful libraries such as Scikit-learn, Keras, Tensor Flow, and Pandas, machine learning models are built to predict mechanical behaviour based on process data. Key algorithms such as support vector machines, random forests, and deep learning neural networks are employed to analyze the relationships between input parameters (e.g., layer height, print speed, material type) and the mechanical performance of printed specimens. The models are optimized and validated using real-world datasets, enabling accurate and efficient predictions of mechanical properties. This Python-based approach reduces trial-and-error experimentation, leading to faster optimization of 3D printing processes for enhanced material performance.**

Keywords: Python, machine learning, 3D printing, mechanical properties, Sickest-learn, Keras, Tensor Flow, additive manufacturing, deep learning, random forest, prediction model.

I. INTRODUCTION

3D printing, or additive manufacturing, has gained significant attention for its ability to produce complex geometries with high precision. Polylactic Acid (PLA), a biodegradable thermoplastic made from renewable resources, is widely used due to its environmental benefits, ease of printing, and cost-effectiveness. Understanding the mechanical properties of 3D-printed PLA, such as tensile strength, Young's modulus, and elongation at break, is essential for optimizing design and performance. These properties depend on printing parameters like layer height, printing speed, infill density, nozzle temperature, and bed temperature.

Traditional mechanical testing methods are often time-consuming and resource-intensive, limiting the efficiency of optimizing printing parameters.

This project aims to employ machine learning (ML) to predict the mechanical properties of 3D-printed PLA specimens based on printing parameters. The goals are to reduce experimental testing, enable customization of printing settings for specific mechanical behaviors, and optimize material usage.

Machine learning offers a data-driven approach to model the relationship between printing parameters and mechanical properties. By training ML models on experimental data, the project seeks to accurately predict properties such as tensile strength and stiffness, allowing for efficient design optimization and material performance enhancement.

The approach involves collecting experimental data on PLA specimens printed with varied settings, extracting key features, and training models including regression algorithms and advanced techniques like random forests and neural networks. Model accuracy will be evaluated using error metrics and coefficient of determination (R^2). The trained model will guide parameter optimization for improved performance in practical applications.

Integrating machine learning with 3D printing technologies can significantly enhance the design and manufacturing process by providing reliable predictions of material behavior. This approach holds potential to improve efficiency and innovation across diverse fields such as aerospace, medical devices, and consumer products..

FIG 1: PLA Material

II LITERATURE REVIEW

Machine learning in predicting mechanical behaviour of additively manufactured parts [1] **:**Although applications of additive manufacturing (AM) have been significantly increased in recent years, its broad application in several industries is still under progress. AM also known as three-dimensional (3D) printing is layer by layer manufacturing process which can be used for fabrication of geometrically complex customized functional end-use products. Since AM processing parameters have significant effects on the performance of the printed parts, it is necessary to tune these parameters which is a difficult task. Today, different artificial intelligence techniques have been utilized to optimize AM parameters and predict mechanical behaviour of 3D-printed components. In the present study, applications of machine learning (ML) in prediction of structural performance and fracture of additively manufactured components has been presented. This study first outlines an overview of ML and then summarizes its applications in AM. The main part of this review, focuses on applications of ML in prediction of mechanical behaviour and fracture of 3Dprintedparts. To this aim, previous research works which investigated application of ML in characterization of polymeric and metallic 3D-printed parts have been reviewed and discussed. Moreover, the review and analysis indicate limitations, challenges, and perspectives for industrial applications of ML in the field of AM. Considering advantages of ML increase in applications of ML in optimization of 3D printing parameters, prediction of mechanical performance, and evaluation of 3D-printed products is expected.

Fused deposition modelling: process, materials, parameters, properties, and applications [2]**:4** In recent years, 3D printing technology has played an essential role in fabricating customized products at a low cost and faster in numerous industrial sectors. Fused deposition modelling (FDM) is one of the most efficient and economical 3D printing techniques. Various materials have been developed and studied, and their properties, such as mechanical, thermal, and electrical, have been reported. Numerous attempts to improve FDM products' properties for applications in various sectors have also been reported. This paper reviews and explains various techniques used in 3D printing and the various polymers and polymer composites used in the FDM process.

III DRAFTING

Drafting of specimen as per ASTM standards in AUTOCAD:

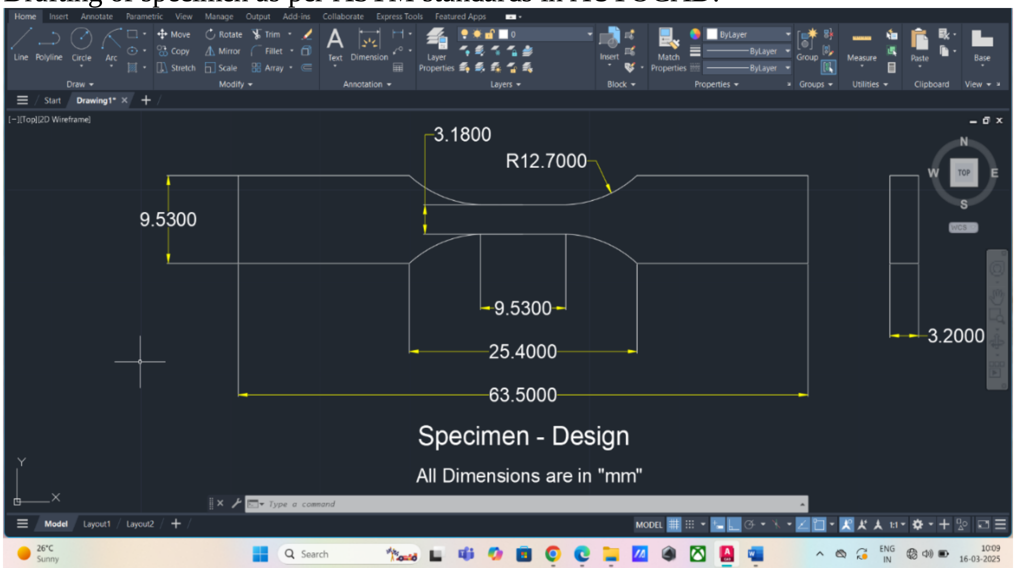


FIG-1, 2D MODEL DESIGN BY USING AUTOCAD

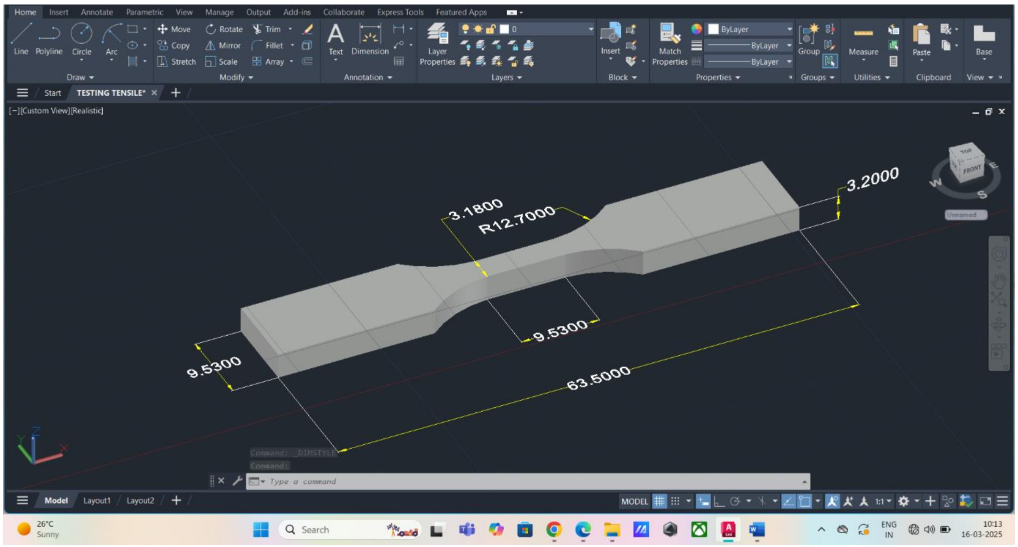


FIG-2, 3D MODEL DESIGN BY USING AUTOCAD

Converting an AutoCAD diagram into STL format by Using AutoCAD:

1. Open your AutoCAD file: Launch AutoCAD and open the diagram which is to be converted.
2. Select the objects: Choose the objects that we want to convert to STL. Then we can select individual objects or entire layers.
3. Use the "Export" command: Go to the "File" menu, select "Export," and choose "STL" as the file type.
4. Configure export settings: In the "STL Export" dialog box, adjust settings like resolution, tolerance, and units as needed.
5. Export the STL file: Click "OK" to export the selected objects as an STL file.

Converting an STL file into slicing:

Converting an STL file into slicing instructions is a crucial step for 3D printing.

Slicing Software:

Cura: A widely used slicing software developed by Ultimaker.

Slicing Process:

1. Import the STL file: Open the slicing software and import the STL file that we want to slice.
2. Configure slicing settings: Adjust settings like:
   - ➢ Layer height
   - ➢ Infill density
   - ➢ Support material
   - ➢ Adhesion sheet
   - ➢ Print speed
   - ➢ Temperature
3. Slice the model: The software will slice the STL file into layers and generate G-code instructions for your 3D printer.
4. Review and adjust: Review the sliced layers and adjust settings as needed to achieve optimal print quality.

Output Files:

1.G-code file: The sliced instructions are saved as a G-code file(.gcode or .g)

2. Print file: slicing software generates a print file (.print or .3d) that contains the G-code and other print settings.

Considerations:

1. Optimize slicing settings: Experiment with different settings to achieve optimal print quality and speed.

2. Use support material wisely: Support material can help with complex prints, but it can also increase print time and material usage.

3. Monitor print quality: Keep an eye on the print quality and adjust slicing settings as needed.

G-code Conversion:

Introduction to G-code Conversion:

Converting an STL file into G-code is an essential step in the 3D printing workflow. G-code, short for "Geometric Code" or "Gantry Code," contains instructions that control the movement and behavior of 3D printers. This guide provides a detailed overview of the process, tools, and considerations involved in converting STL files to G-code.

Understanding G-code:

G-code is a standardized language used in 3D printing to communicate instructions to the printer's firmware[10]. Each G-code command specifies a particular action, such as moving the print head, extruding filament, or adjusting printer settings. Understanding G-code syntax and structure is crucial for successful 3D printing.

Tools for G-code Conversion:

Several software tools are available for converting STL files into G-code. Each tool offers unique features and capabilities to streamline the conversion process. Some popular options include:

1. Slicer Software: Slicer software such as Ultimaker Cura, PrusaSlicer, and Simplify3D are commonly used for slicing STL files and generating G-code for 3D printing.

2. Printer Firmware: Some 3D printers come with built-in firmware that can directly process STL files and generate G-code internally.

3. Third-party Converters: Various third-party software tools and utilities specialize in converting STL files to G-code, offering additional features such as advanced slicing algorithms, customizable settings, and simulation capabilities.

Step-by-Step Guide to G-code Conversion:

Here's a detailed walkthrough of the conversion process using Ultimaker Cura as an example:

1. Install and Launch the Slicer Software:

• Download and install the latest version of Ultimaker Cura or your preferred slicer software.

• Launch the software and ensure that it is properly configured for your 3Dprinter.

2. Import the STL File:

• Open the STL file you wish to convert into G-code by selecting "File" > "Open File" or dragging the file into the slicer interface.

3. Configure Print Settings:

• Adjust print settings such as layer height, infill density, print speed, and support structures based on your preferences and requirements.

• Specify printer parameters such as build volume, nozzle diameter, and filament type to ensure compatibility.

4. Slice the Model:

• Click the "Slice" button to initiate the slicing process, during which the slicer software converts the 3D model into individual layers and generates corresponding G-code instructions.

• Review the sliced model preview to ensure accuracy and identify any potential issues such as overhangs, gaps, or printing artifacts.

5. Preview G-code:

• Review the generated G-code preview to visualize the printing path, layer-by layer structure, and estimated print time.

• Use simulation tools if available to identify and address potential printing issues before starting the print.

6. Export G-code:

• Once satisfied with the slicing settings and G-code preview, export the generated G-code file by selecting "File" > "Save G-code" or a similar option.

• Choose an appropriate location and file name for the exported G-code file and save it to your computer or directly to an SD card for printing.

7. Transfer G-code to Printer:

• If saving the G-code file to a computer, transfer it to the 3D printer via USB cable or SD card.

• Insert the SD card into the printer or establish a connection between the computer and printer, ensuring proper communication.

8. Start Printing:

• Use the printer's interface or controls to select and initiate the printing process, following any prompts or instructions displayed on the screen.

• Monitor the print progress and address any issues that may arise during the printing process.

Considerations for G-code Conversion:

• Print Settings Optimization: Adjust slicing parameters to achieve the desired balance between print quality, speed, and material usage.

• Support Structures and Rafts: Enable support structures or rafts as needed to enhance print stability and mitigate issues such as warping or sagging.

• Layer Adhesion and Cooling: Optimize cooling settings to improve layer adhesion and reduce the risk of overheating or deformations.

• Bed Leveling and Calibration: Ensure proper bed leveling and printer calibration to optimize print quality and accuracy.

Advanced Techniques and Optimization:

• Custom G-code Scripts: Implement custom G-code scripts to automate specific tasks or enhance printer functionality.

• Multi-material Printing: Explore advanced slicing features for multi-material printing, including dual extrusion and color blending.

• Experimental Features: Experiment with experimental features and plugins to unlock additional capabilities and optimize print performance.

Troubleshooting and Debugging:

• Print Failures Analysis: Analyze G-code and print settings to diagnose and trouble shoot common print failures such as stringing, layer shifting, or adhesion issues.

• Debugging Tools: Utilize debugging tools and visualization features within slicer software to identify and resolve slicing errors or inconsistencies.

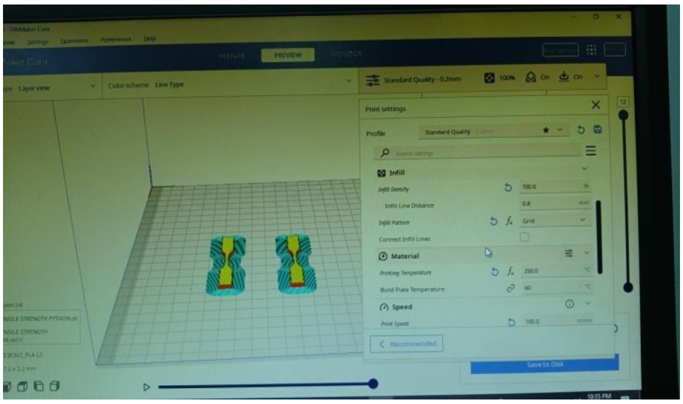


FIG-3, ULTIMAKER CURA

IV 3D PRINT (SPECIMEN) PROCESSING

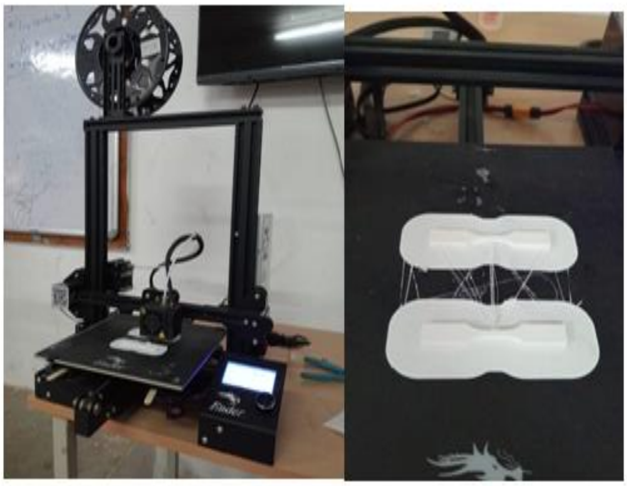


FIG-4, Fused Deposition Modelling (FDM) 3D printer (ender 3 v2)

PROCESS PARAMETERS:

The Taguchi method was employed to systematically vary the printing parameters and prepare six specimens for tensile testing. The selected parameters and their respective levels are as shown in table1

Table-1

Printing Parameter	**Level 1**	**Level 2**	**Level 3**
Layer Thickness	0.16 mm	0.2 mm	0.28mm
Infill Density	90%	90%	100%
Print Speed	90mm/s	100mm/s	100 mm/s
Nozzle Temperature	240°C	245°C	250°C

The Following Table 2 shows each combination of printing parameters was assigned a unique code to facilitate identification and tracking during the printing and testing phases.

Table-2

Code	**Layer Thickness mm**	**Infill Density %**	**Print Speed mm/s**	**Nozzle Temperature °C**
TS-1	0.16	90	90	240
TS-2	0.16	100	100	250
TS-3	0.2	90	90	250
TS-4	0.2	100	100	240
TS-5	0.28	90	100	250
TS-6	0.28	100	90	240

FIG -5, 3D PRINTED SPECIMENS

V-TESTING

Tensile strength testing is a critical method for evaluating the mechanical properties of 3D-printed materials, such as Polylactic Acid(PLA).

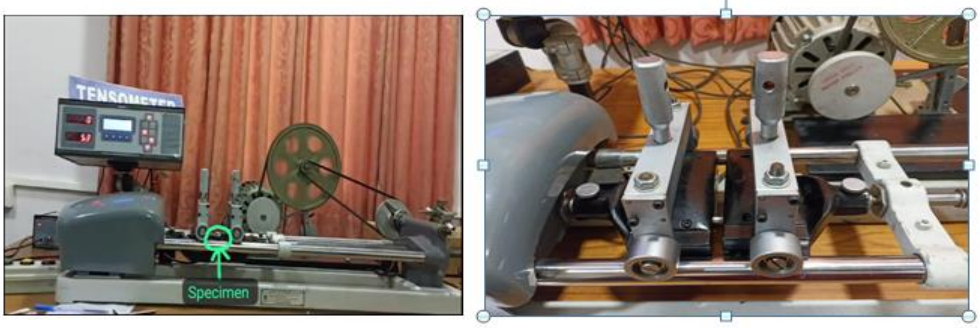
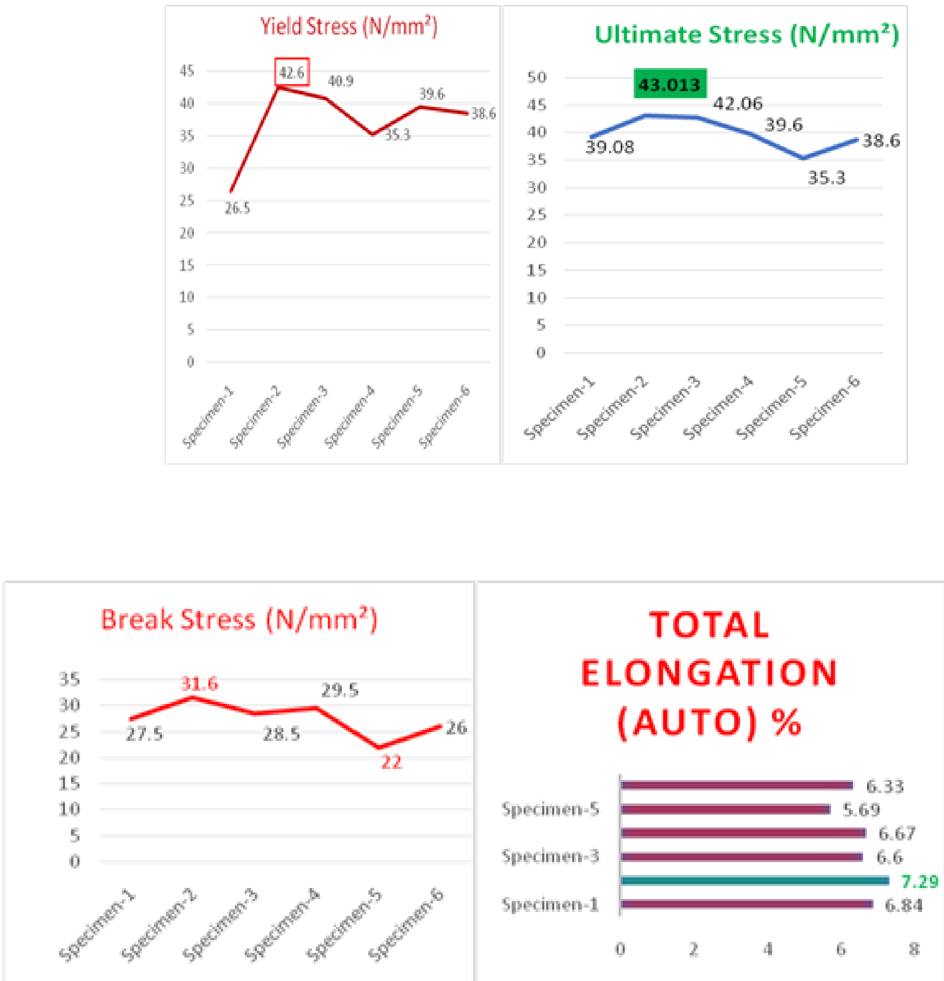


FIG-6, Tensile Testing on Tensometer

FIG-7, Specimens after Tensile Test



Graph-1

Python script that reads the table data, analyzes it, and generates visualizations using Matplotlib and Pandas. The script includes:

- Data entry as a Pandas DataFrame
- Basic statistical analysis
- Visualization of Yield Force, Yield Stress, and Yield Strain vs. different parameters

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

file_path = "C:/Users/komal/Downloads/TENSILE RESULTS.csv"
df = pd.read_csv(file_path)

# Display column names for reference
print("Dataset Columns:", df.columns.tolist())

# Identify the columns related to Stress and Strain for all specimens
stress_cols = [col for col in df.columns if "Stress" in col]
strain_cols = [col for col in df.columns if "Strain" in col]

# Extract computed mechanical properties
mechanical_properties = []
for stress_col, strain_col in zip(stress_cols, strain_cols):
    stress = df[stress_col]
    strain = df[strain_col]

yield_index = stress.idxmax() // 4  # Approximate yield point (change if needed)
ultimate_index = stress.idxmax()  # Ultimate stress point
break_index = len(stress) - 1  # Last recorded point

mechanical_properties.append({
        "Yield Stress": stress.iloc[yield_index],
        "Ultimate Stress": stress.iloc[ultimate_index],
        "Break Stress": stress.iloc[break_index],
        "Yield Strain": strain.iloc[yield_index],
        "Ultimate Strain": strain.iloc[ultimate_index],
        "Break Strain": strain.iloc[break_index]
    })

# Convert to DataFrame
mechanical_df = pd.DataFrame(mechanical_properties)

# Define target variable based on Ultimate Stress
mechanical_df["Strength_Class"] = np.where(
mechanical_df["Ultimate Stress"] >mechanical_df["Ultimate Stress"].mean(), "Strong", "Weak"
)

# Convert categorical labels to numeric
```

```
mechanical_df["Strength_Class"] = mechanical_df["Strength_Class"].map({"Weak": 0, "Strong": 1})

# Features and target
X = mechanical_df.drop(columns=["Strength_Class"])
y = mechanical_df["Strength_Class"]

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Model Accuracy
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"✅Model Accuracy: {accuracy:.2f}")

# Take user input for prediction
print("\n□ Enter properties for prediction:")
user_data = []
for feature in X.columns:
   value = float(input(f"Enter {feature}: "))
user_data.append(value)

# Predict Strength
user_df = pd.DataFrame([user_data], columns=X.columns)
prediction = model.predict(user_df)[0]
strength_label = "Strong" if prediction == 1 else "Weak"

print(f"\n□ Prediction: The material is predicted to be **{strength_label}**.")
```

This will take inputs of yield stress, ultimate stress, break stress, yield strain, ultimate stress, and break stress from the user. After giving these inputs, it will generate output by saying whether the material is "strong or weak".

```
🔹 Enter properties for prediction:
Enter Yield Stress: 50
Enter Ultimate Stress: 90
Enter Break Stress: 55
Enter Yield Strain: 4
Enter Ultimate Strain: 6
Enter Break Strain: 8

🔍 Prediction: The material is predicted to be **Strong**.
```

.

VI. Conclusion

Based on the provided graphs, here are some conclusions regarding the mechanical properties of the specimens:

Yield Stress:

The highest yield stress is observed in Specimen-3 (42.6 N/mm²).

The lowest yield stress is in Specimen-1 (26.5 N/mm²).

Most specimens exhibit yield stress values in the range of 35–40 N/mm².

Ultimate Stress:

Specimen-3 has the highest ultimate stress (43.013 N/mm²).

Specimen-5 has the lowest ultimate stress (35.3 N/mm²).

The ultimate stress values show moderate variations among specimens, mostly between 35–43 N/mm².

Break Stress:

Specimen-3 has the highest break stress (31.6 N/mm²).

Specimen-6 has the lowest break stress (22 N/mm²).

Break stress values indicate some inconsistency across specimens.

Total Elongation:

The highest elongation percentage is in Specimen-1 (7.29%).

The lowest elongation is observed in Specimen-5 (5.69%).

Most specimens exhibit elongation between 6% and 7%.

General Observations:

Specimen-3 shows the best overall mechanical properties with high yield, ultimate, and break stresses.

Specimen-5 has the lowest combination of ultimate stress and elongation, suggesting lower ductility and strength.

Specimen-1 has the highest elongation but the lowest yield stress, indicating a more ductile nature but lower strength.

Specimen-6 exhibits the lowest break stress, indicating potential weakness in fracture resistance.

The trained model predicts whether the material is **"Strong" or "Weak"** based on the provided values by using Machine learning successfully completed.

References

.

[1] C. J. Culbreath, B. Gaerke, M. S. Taylor, S. D. McCullen, and O. T. Mefford, "Effect of infill on resulting mechanical properties of additive manufactured bioresorbable polymers for medical devices," Additive Manufacturing, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2589152920301496

[2] K. Rajan, M. Samykano, and K. Kadirgama, "Fused deposition modelling: Process, materials, parameters, properties, and applications," 2022. [Online]. Available: https://www.researchgate.net/publication/358884035_Fused_deposition_modeling_process_materials_parameters_properties_and_applications

[3] S. Ganeshkumar, S. Dharani Kumar, U. Magarajan, et al., "Investigation of tensile properties of different infill pattern structures of 3D-printed PLA polymers: Analysis and validation using finite element analysis in ANSYS," Materials, vol. 15, no. 5142, 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9331637/pdf/materials-15-05142.pdf

[4] L. G. Bracaglia, B. Smith, E. Watson, N. Arumugasami, G. Mikos, and J. Fisher, "3D printing for the design and fabrication of polymer-based gradient scaffolds," Acta Biomaterialia, vol. 74, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1742706117302015

[5] L. Sandanamsamy, J. Mogan, K. Rajan, W. S. W. Harun, I. Ishak, F. R. M. Romlay, M. Samykano, and K. Kadirgama, "Effect of process parameter on tensile properties of FDM printed PLA," Materials Today: Proceedings, 2021.

[6] Y. Song, Y. Li, W. Song, K. Yee, K.-Y. Lee, and V. L. Tagarielli, "Measurements of the mechanical response of unidirectional 3D-printed PLA," Polymer Testing, vol. 72, 2018.

[7] M. Bembenek, Ł. Kowalski, and A. Kosoń-Schab, "Research on the influence of processing parameters on the specific tensile strength of FDM additive manufactured PET-G and PLA materials," Polymers, vol. 14, no. 2446,

[8] K. Sood, R. K. Ohdar, and S. S. Mahapatra, "Parametric appraisal of mechanical property of fused deposition modelling processed parts," Materials & Design, vol. 31, no. 1, pp. 287–295, 2010.

[9] D. M. Pilipović, P. Rao, and M. Šercer, "Experimental analysis of properties of materials for rapid prototyping," The International Journal of Advanced Manufacturing Technology, 2008. [Online]. Available: https://link.springer.com/article/10.1007/s00170-007-1310-7

[10] M. Heidari-Rarani, M. Rafiee-Afarani, and A. M. Zahedi, "Mechanical characterization of FDM 3D printing of continuous carbon fiber reinforced PLA composites," Composites Part B: Engineering, vol. 160, pp. 214–220, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1359836818335686