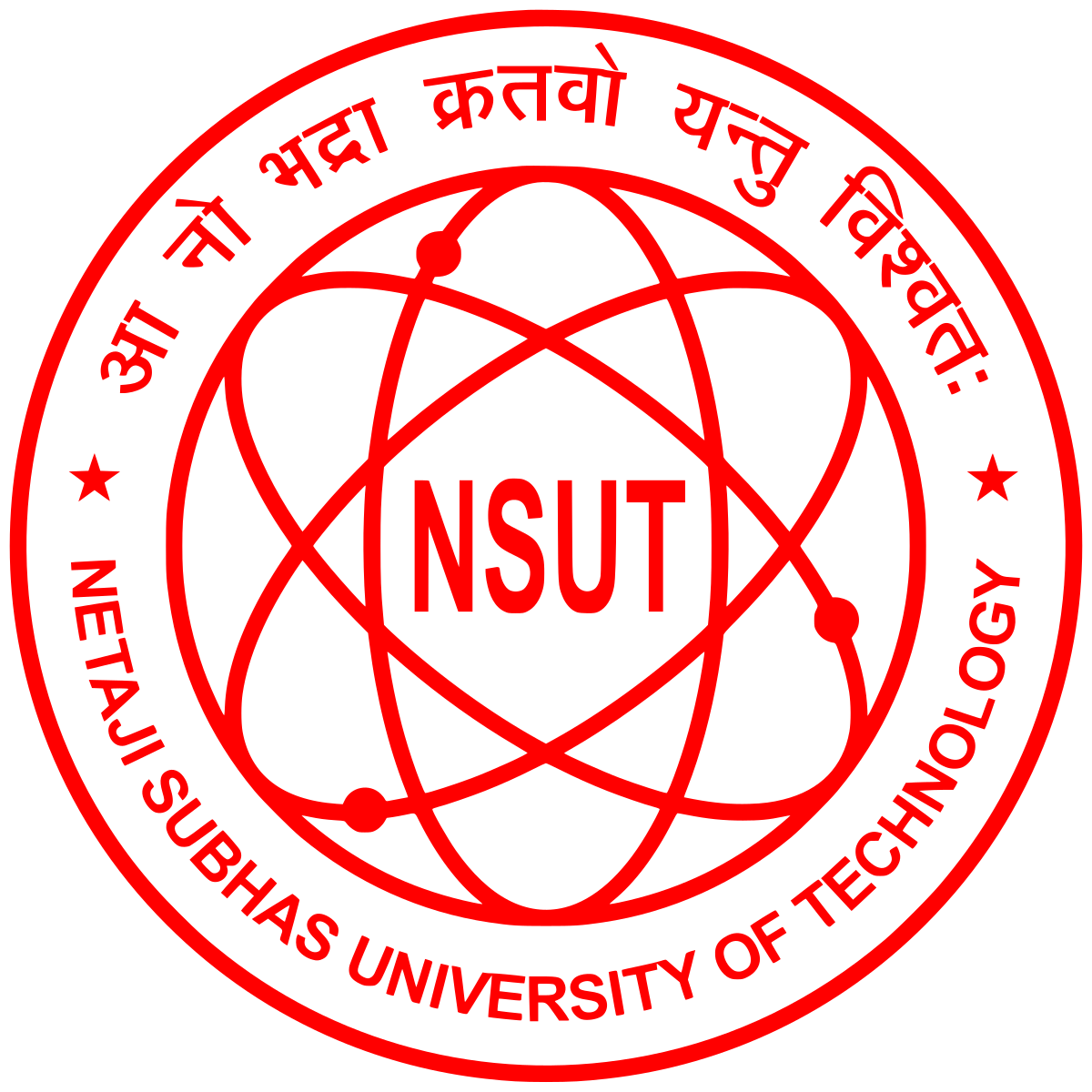
****

**Machine Learning Project Report**

**Predicting Residual Strength of FRP Composites Under Fatigue**

Uday Singh - 2023UCM2333

Tanishka Khandelwal – 2023UCM2339

**Contents**

1. Introduction
2. Dataset
3. Methodology
4. ML models used
5. Graphs and confusion matrix
6. References
7. **Introduction**

Fiber-Reinforced Polymer (**FRP**) composites experience progressive damage when subjected to cyclic (fatigue) loading, leading to a reduction in their load-carrying capacity, known as residual strength. Accurate prediction of residual strength is crucial for ensuring the safety and reliability of structures made from FRP composites.

Several methods are used to predict residual strength:

1. **Empirical Models**: Based on experimental data, these models often use power-law relationships to describe strength degradation over cycles.
2. **Damage Mechanics Models**: Use damage variables to represent material degradation, linking them to residual strength through mathematical formulations.
3. **Fracture Mechanics Approaches**: Focus on crack initiation and propagation, particularly useful for modelling delamination and other localized failures.
4. **Micromechanical and Multiscale Models**: Simulate the behaviour of fibres, matrix, and interfaces to predict damage evolution and residual strength.
5. **Machine Learning Techniques**: Modern data-driven approaches that predict residual strength using input features like stress levels, cycle counts, and material properties.

**2. Dataset Overview**

* **Source**: GitHub Dataset from published research.
* **Filename**: Residual\_Fatigue.csv
* **Features**:
  + Material (encoded)
  + Nature (encoded)
  + Stacking (encoded)
  + UTS (Ultimate Tensile Strength)
  + Stress Amplitude
  + Stress Ratio
  + Frequency
  + Number of Plies
  + Fatigue Cycles
  + Normalized Fatigue Life
* **Target Variable**: Target (Residual Strength)

**3. Methodology**

**1. Material and Specimen Preparation**

* Select the FRP composite (e.g., Carbon/Epoxy).
* Define layup configuration (e.g., [0/90], quasi-isotropic).
* Fabricate standard test specimens (e.g., ASTM D3479 for fatigue).

**2. Fatigue Testing**

* Perform cyclic loading at various stress levels and stress ratios (e.g., R = 0.1).
* Use a servo-hydraulic testing machine.
* Record:
  + Number of cycles to failure
  + Intermediate residual strength at specific cycle intervals

**3. Damage Monitoring (Optional)**

* **Monitor internal damage progression using:**
  + Stiffness degradation
  + Acoustic Emission (AE)
  + Ultrasonic C-scan or Thermography

**4. Data Collection**

* **Collect:**
  + Initial strength R0R\_0R0​
  + Residual strength R(N)R(N)R(N) after selected number of cycles
  + Cycle count NNN
  + Failure modes

**5. Model Selection**

**Choose an appropriate prediction model:**

* Empirical: e.g., power-law decay
* Damage Mechanics-Based
* Machine Learning Models (if dataset is large):
  + Inputs: stress, cycle count, damage indicators
  + Outputs: predicted residual strength

**6. Model Fitting and Validation**

* Use regression or optimization tools (e.g., Python, MATLAB) to fit model parameters.
* Validate the model using:
  + Root Mean Square Error (RMSE)
  + R-squared (R²)
  + Cross-validation (if ML is used)

**7. Prediction and Analysis**

* Use the trained model to predict residual strength at unseen cycle counts.
* Plot residual strength vs. cycle count.
* Analyse trends across different stress levels or layups.

**8. Conclusion**

* Summarize accuracy, limitations, and potential improvements.
* Discuss applicability to real-world loading scenarios.

**4. Machine Learning Models Used**

**4.1 Baseline Regressors**

* **Linear Regression :** It minimizes the sum of squared errors (SSE) to find the best-fit line. It's simple but can overfit with many features or multicollinearity.
* **Ridge Regression :** It reduces overfitting by shrinking coefficients, especially when features are highly correlated. It doesn’t eliminate any features.
* **Lasso Regression :** It helps with feature selection by forcing some coefficients to zero, effectively removing less important features.
* **Elastic Net :** It balances between Ridge and Lasso, working well when there are many features, especially with correlations or when only a few are significant.

**4.2 Tree-Based Models**

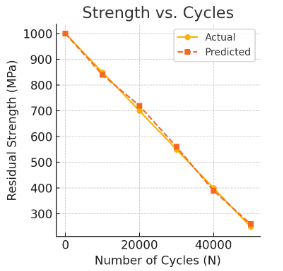
* **Random Forest Regressor** (GridSearchCV for hyperparameter tuning)
* **Gradient Boosting Regressor** (GridSearchCV for hyperparameter tuning)

**4.3 Neural Network**

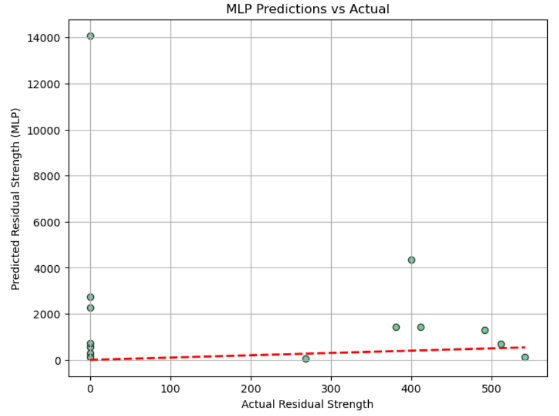
* **Multilayer Perceptron (MLP)** using scikit-learn's MLP Regressor

**5. Graphs and Confusion matrix**

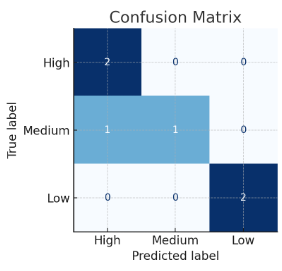
**1. Residual Strength vs. Number of Cycles –** shows how strength degrades over time.

****

**2. Actual vs. Predicted Strength –** checks regression accuracy.

****

**3. Confusion Matrix –** classifies strength into High, Medium, Low and compares predictions.

****

1. **References**

**Dataset :**

[**https://github.com/Dewa1989/Residual-Fatigue-Strength/tree/main**](https://github.com/Dewa1989/Residual-Fatigue-Strength/tree/main)