# MSE 1065 Lab 8 – Prediction of total crack length using various ML models

In this lab, we will be fitting different machine learning models on the dataset downloaded from the paper titled "Using deep neural network with small dataset to predict material defects". The data consists of 22 columns. The input features are the first 21 columns and constitute different elemental compositions and strain percentage values. We will be predicting total crack length ("TCL"), values of which are detailed in the last column of the dataset. Your submission will be a single jupyter notebook containing all your code and analysis.

Part 1 (5 points): Load the dataset and perform normalization by computing mean and standard deviation for each feature. Create train and test split with test sample size of 20%.

Part 2 (10 points). Fit a *Linear regression model* and compute Mean absolute error (MAE), r-squared value and root mean squared error, corresponding to both training and test set. Perform *regularization or fit ridge regression model* for 5 different values of alpha. The alpha can be chosen as [0.01, 0.1, 1.5, 2.0, 5.0]. Compare the test set error of Linear model with ridge regression model and also how MAE or RMSE varies with the increase in alpha. Plot the predictions versus true values for the best model that you obtained in this section.

Part 3 (5 points). Fit a *neural network* using Keras/SKLearn library on the training set created in Part 1. The neural network has two hidden layers with 6 hidden units each. Compute Mean absolute error (MAE), r-squared value and root mean squared error, corresponding to both training and test set. Has neural networks improved the predictions as compared to linear models?

Optional:- Once finished with Part 3, you can add another layer to your neural network model with 3 hidden units and see if it improves the prediction. You can also implement cross-validation to select from 4 different neural network architecture. Feel free to choose the architecture detail of individual networks in this case.

Part 4 (10 points). Fit your *favourite model*, other than linear regression and neural network on this dataset and compute the performance. As an example, we covered Polynomial regression, Kernel methods, Gaussian Processes and ensemble methods in the class. You can use any of these.

Please feel free to use any model even that is not covered in the class, as long as your implementation use libraries such as SKLearn, GLMNet and Keras.