Team 2

ChE 4230

Group Project Part 3

8 April 2022

In this section, we created fault identifiers and fault classifiers. The two that we used were artificial neural networks (ANN) and support-vector machines (SVM).

Before utilizing these fault detection methods, the data must be preprocessed as the raw data is noisy and hard to understand. This was done by reading the raw data from a csv using pandas into a dataframe. That dataframe was then scaled using the standard scaler tool in sklearn. The data was then split into training and testing data using the train\_test\_split function in sklearn with 20% of the data being test data. From that point the data was put through 5 different forms dimensionality reduction: PCA, TSNE, SVD, isomap, and LLE. The data was then graphed and we found that the two best methods were TSNE and SVD. The TSNE and SVD data was taken and we performed clustering on it with 3 different methods: Kmeans, DBScan, and Agglomerative clustering. We found that agglomerative clustering using the data from TSNE dimensionality reduction and 5 clusters showed the best splitting of the data. For this reason, we used this data for our model.

We then built a model builder script that could take our data and create a neural network using the MLPClassifier in sklearn. The MLPClassifier takes the training data and attempts to fit it to a model and then tests the model by trying to predict what y values will come from the test x values. We can then quantify the error by comparing the predicted value to the actual test value from the train\_test\_split. Another form of model used was a decision tree which overall does the same as the MLPClassifier but uses a decision tree based model instead of equation based. Both of these can be visualized by plotting them.

We can take this model from the MLPClassifier and build a neural network and use the ANN to classify the errors that the model produces and decide how many internal nodes is best for the neural network. We found that 200 internal nodes produced the lowest amount of error. The ANN quantifies error based on information gain and entropy loss and adjusts the internal nodes in order to maximize information gain and keep the running entropy loss low.

We created a support vector machine (SVM) as a potential alternative to ANN for classifying faults. The SVM pulls the preprocessed data and fits the data to a model such that the error from misclassification is minimized. The SVM accomplishes this by finding the most optimum decision boundary between data points of each class (finds the maximum distance between nearest data points of all classes). We used the skelarn SVC (support vector classifier) to fit the data and make a prediction. We then used sklearn metrics to quantify the weighted and unweighted averages of accuracy and precision for each set of predicted data. From the SVM classifier results, we found that (using the RBF kernel for SVC) our model was 97% precise (weighted) and 97% accurate (F-1 score).