SLATT UNDERGRADUATE RESEARCH FELLOWSHIP
2020 FINAL REPORT

SCHOLAR NAME:	Christina Tan
FACULTY ADVISOR:	Yamil Colón
PROJECT PERIOD:	Summer 2020 – Spring 2021
PROJECT TITLE:	Automated Detection of Defects in Porous Materials with Machine Learning
CONNECTION TO ONE OR MORE ENERGY-RELATED RESEARCH AREAS (CHECK ALL THAT APPLY):	**(X) Energy Conversion and Efficiency** **( ) Sustainable and Secure Nuclear** **( ) Smart Storage and Distribution** **( ) Transformation Solar** **( ) Sustainable Bio/Fossil Fuels** **( ) Transformative Wind**

MAJOR GOALS AND ACCOMPLISHMENTS

Summarize your research goals and provide a brief statement of your accomplishments (no more than 1-2 sentences). Indicate whether you were able to accomplish your goals by estimating the percentage completed for each one. Use the next page for your written report.

RESEARCH GOALS	ACTUAL PERFORMANCE AND ACCOMPLISHMENTS	% OF GOAL COMPLETED
Test different methods of anomaly detection	Five anomaly detection methods, supervised and unsupervised, were evaluated on four datasets made for outlier detection. The overall best methods tested were logistic regression and local outlier factor novelty detection.	100%
Run sensitivity analysis for varying percentages of outliers	A sensitivity analysis was run on the unsupervised learning programs using the cardio dataset with varying percentages of outliers, ranging from 1% to 9%.	100%
Create MOF structures computationally and characterize them	8418 Zirconium-based metal-organic frameworks were generated using ToBaCCo. These were characterized using zeo++ by pore diameters, accessible volume, pore size, and surface area.	100%
Add and quantify defects in MOF structures	Defects were added into structures by removing Zirconium-based nodes. This was done by modifying the cif file, and the defected cif was then characterized by zeo++. More defected structures need to be made to further explore anomaly detection for this dataset.	90%
Write program to detect defected structures	We have found some success in detecting structures with higher quantities of defects using Isolation Forest but are continuing to explore how structures with less defects can be found.	50%

RESEARCH OUTPUT

Please provide any output that may have resulted from your research project. You may leave any and all categories blank or check with your faculty advisor if you are unsure how to respond.

CATEGORY	INFORMATION
EXTERNAL PROPOSALS SUBMITTED	(Sponsor, Project Title, PIs, Submission Date, Proposal Amount)
EXTERNAL AWARDS RECEIVED	(Sponsor, Project Title, PIs, Award Date, Award Amount)
JOURNAL ARTICLES IN PROCESS OR PUBLISHED	(Journal Name, Title, Authors, Submission Date, Publication Date, Volume #, Page #s)
BOOKS AND CHAPTERS RELATED TO YOUR RESEARCH	(Book Title, Chapter Title, Authors, Submission Date, Publication Date, Volume #, Page #s)
PUBLIC PRESENTATIONS YOU MADE ABOUT YOUR RESEARCH	(Event, Presentation Title, Presentation Date, Location)
AWARDS OR RECOGNITIONS YOU RECEIVED FOR YOUR RESEARCH PROJECT	(Purpose, Title, Date Received)
INTERNAL COLLABORATIONS FOSTERED	(Name, Organization, Purpose of Affiliation, and Frequency of Interactions)
EXTERNAL COLLABORATIONS FOSTERED	(Name, Organization, Purpose of Affiliation, and Frequency of Interactions)
WEBSITE(S) FEATURING RESEARCH PROJECT	(URL)

OTHER PRODUCTS AND SERVICES (e.g., media reports, databases, software, models, curricula, instruments, education programs, outreach for ND Energy and other groups)	(Please describe each item in detail)

RESEARCH EXPERIENCE

Please let us know what you thought of your research experience: Did this experience meet your expectations? Were lab personnel helpful and responsive to your needs? What else could have been done to improve your experience or achieve additional results?

This experience has met and exceeded my expectations. Being able to do this computational research project before and continuing through the pandemic has been an incredible learning experience that was very flexible with working from home and on my own time, and the Slatt fellowship provided me the resources necessary to be able to do so. Prof. Yamil Colón has been a fantastic mentor; he has been very helpful and responsive to my needs with regards to academic guidance and one-on-one meetings. I have enjoyed this experience and am deeply grateful for this fellowship and all it has allowed me to do.

FINAL WRITTEN REPORT

(Please use the space below to describe your research project and objectives, any findings and results you can share, and graphs, charts, and other visuals to help us understand what you achieved as a result of this research experience.)

Introduction

Metal-organic frameworks (MOFs) are a class of materials synthesized using organic and inorganic reactants to make a porous crystalline structure. The pores give MOFs their unique properties that make them an advantageous material because they can be adapted for different use cases by modification of the reactants. The applications of these structures can be dependent on the quantity of defects present. One such application is catalysis where specific defects can be used as catalytic sites. In this case, the number of defects significantly changes the activity of the structure (1). This can also be used to improve performance in energy conversion and storage applications (2). Determining how many defects are present in a material can be difficult without advanced characterization techniques. This research project aims to detect and quantify defects in porous materials using simple characterization techniques and machine learning algorithms.

Outlier Detection Datasets

To validate outlier detection algorithms, they were first tested on datasets designed for this use case. Four datasets with different percentages of outliers and distributions were used from the Outlier Detection Data Sets (ODDS) library of Stony Brook University to test various methods of outlier detection. All of these datasets were originally from the UCI machine learning repository and preprocessed.

The cardiotocography (cardio) dataset contained 21 features including fetal heart rate and uterine contraction measurements. Three of these features are graphed below in Figure 1 with points classified as normal or pathologic:

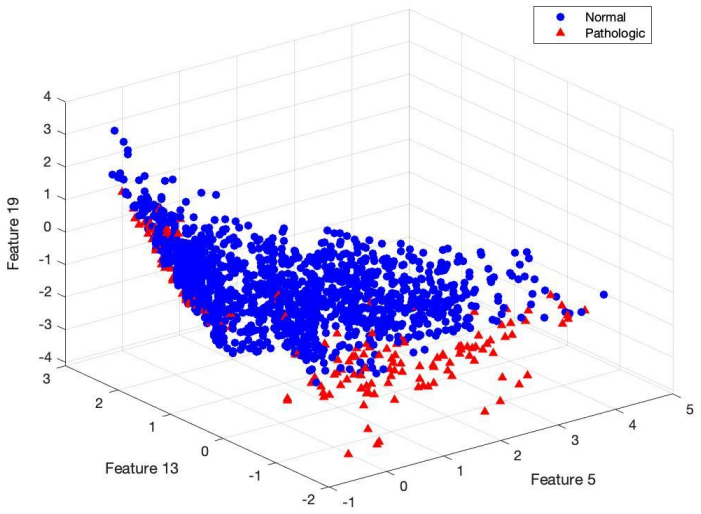

Figure 1. Scatter Plot of Cardio Data

The pathologic class was used as outliers with 176 samples out of 1831 (9.6%). As seen in Figure 1, this dataset had two clusters of outliers close to the edges of the normal data. One cluster, shown on the left, was more tightly packed with similar values for feature 5. The other

cluster was further away from the normal data and more spread out in the negative region of feature 19.

A third dataset used was the satimage-2 dataset, whose features were values of pixels in a satellite image. Each sample was classified by the soil in the image and class 2 was labeled as outliers. This dataset contained 36 features with 71 outliers out of 5803 samples (1.2%). We modified this dataset for the unsupervised methods to only have 4 features: those of the central pixel. Three of these four features are plotted below in Figure 2:

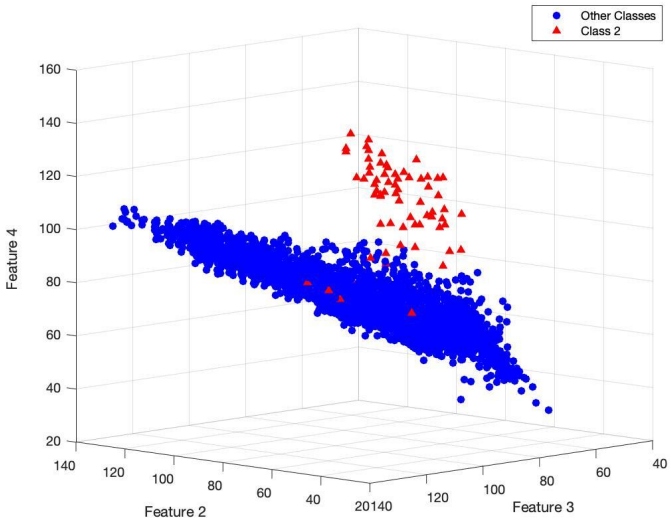

Figure 2. Scatter Plot of Satellite Image Data

The normal data generally follows a linear relationship between features 3 and 4. As shown on the plot, the majority of the outliers do not follow this trend.

Another dataset used was the shuttle dataset. Three features from this dataset are graphed below in Figure 3 with points classified as outliers in red:

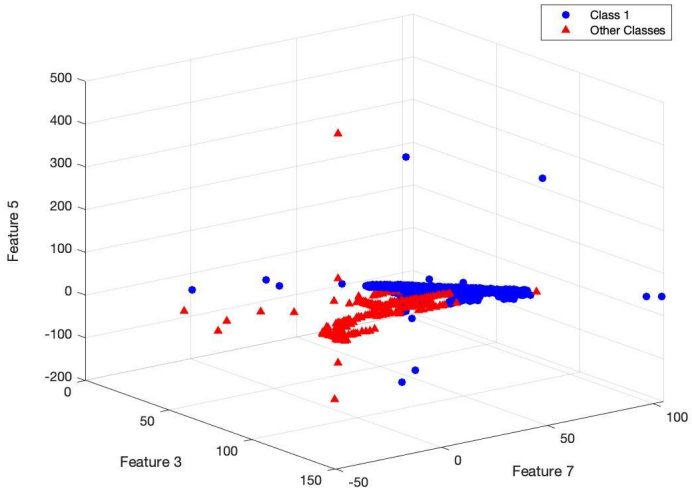

Figure 3. Scatter Plot of Shuttle Data

Class 1, containing the majority of the points, was classified as the inlier class, and the rest of the classes were labeled outliers. This dataset had 9 features with 3511 outliers out of 49097 points (7.2%). The normal points fall in two intersecting planes. The outliers in this dataset had lower values for feature 7 and did not fall in the aforementioned planes of the normal cluster.

The final dataset used to validate outlier detection methods was the vowels dataset, which contained 12 features that comprise a speaker's utterance of two vowels. The utterances (samples) were classified by the speaker. Three of these features are plotted below in Figure 4:

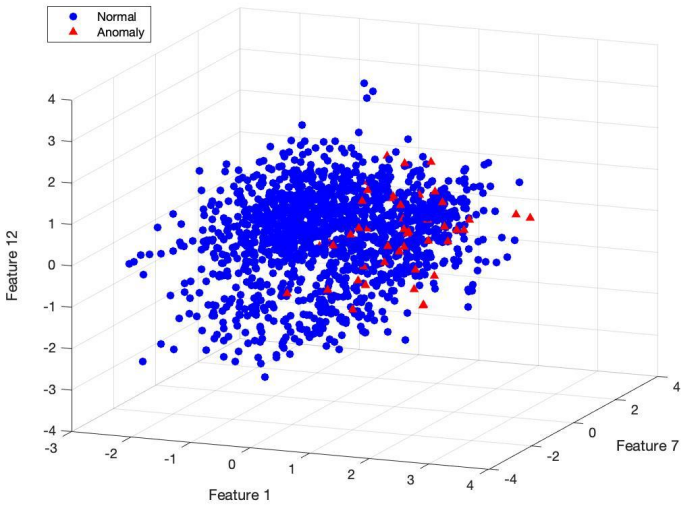

Figure 4. Scatter Plot of Vowels Data

The outlier class shown in red were the utterances of speaker 1. The samples of the other speakers were classified as inliers. This dataset contained 50 outliers out of 1456 samples (3.4%). The outliers do not deviate far from the normal data and were in the positive region of feature 1.

Anomaly Detection Algorithms

Using these datasets, different methods for anomaly detection were tested to determine the best algorithms for each case. Both supervised and unsupervised learning were evaluated using their F1 score, which is the harmonic mean of the model's precision and recall. F1 scores range from 0 to 1 with 1 being a perfect score.

The first supervised learning method tested was Gaussian distribution. For this method, the training set was assumed to contain all (or mostly) normal samples. We calculated the mean and standard deviation of each feature using the training set and used the following equation, the Gaussian probability density function, to determine the probability that x would be the value of a certain feature:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{(Eqn. 1)}$$

where μ is the mean and σ^2 is the variance. By taking the product of the features' individual probabilities, we calculated the probability that a sample was an inlier. Using this model based on the training set, ε (the maximum probability at which a sample would be classified as an outlier) was determined based on the value which gave the highest F1 score using the cross validation set. This completed model was then tested on new samples, where samples with $p < \varepsilon$ were classified as outliers.

For some datasets, we also tried training the model using the bootstrap method to calculate the mean and standard deviation for the features. Rather than calculating the metrics directly from points in the training set, we randomly sampled the training set with replacement and calculated the mean and standard deviation of each feature. This process was repeated a thousand times and the mean of each metric was determined to train the model.

For this model and logistic regression, we used three ways of dividing the data into training, cross-validation, and test sets. The first split which we refer to as Split 0 divided the anomalies proportionally through the three sets. The training set was sixty percent of the dataset including sixty percent of the outliers, and the cross-validation and testing sets were each twenty percent of the data with twenty percent of the outliers. The second way we split the data (Split 1) did not include any outliers in the training set and split the anomalous data evenly between the cross-validation and testing sets. The last split, Split 2, was random, again with sixty percent of the data in the training set and twenty percent in both the cross-validation and testing set. For each dataset, Gaussian Distribution was run one hundred times for each data split.

For the cardio dataset, Split 1 using the bootstrap method gave the highest average F1 score with the least variance. The F1 scores for each model tested is shown below in Table 1:

Table 1. F1 Scores for Cardio Dataset Using Gaussian Distribution

Data Split	F1 Score Using All Features	F1 Score Using Limited Features	F1 Score Using All Features (Bootstrap)
0	0.551 ± 0.059	0.633 ± 0.074	————
1	0.538 ± 0.188	0.536 ± 0.176	0.744 ± 0.021
2	0.604 ± 0.047	0.677 ± 0.081	————

Split 1 produced the highest F1 scores using all of the features. However, this model also generated the lowest F1 scores because for half of the trials, it predicted the majority of the

samples to be outliers. This caused the standard deviation for the results of this model to be high. To remedy this, we employed the bootstrap method to train the model. This worked very well and consistently gave higher F1 scores with an average of 0.744 ± 0.021. Then, trying to improve the performance of the model, we chose seven features that had more normal distribution. While the model with less features did perform better than the entire dataset a few times, it did not perform consistently and had lower recall, meaning the outliers the model identified were likely identified correctly, but the model did not find many of the actual outliers. This might have been due to the further cluster. The model might have a larger ε, which included the cluster farther from the normal data but missed the other because it was closer to the normal data. The bootstrap method using all the features was more consistent and identified most of the outliers but more false positives. This could have been due to the cluster closer to the normal data. Because it was closer, ε may have been chosen to include this anomalous data which caused the model to also identify normal data that is the same distance from the mean as the anomalous cluster as outliers; therefore the model identified more false positives. Because the distance from the mean varies for the outliers and the normal data, Gaussian distribution is more likely to misidentify some of the points for this dataset.

For the satimage dataset, all models performed similarly as seen in Table 2 below:

Table 2. F1 Scores for Satimage Dataset Using Gaussian Distribution

Data Split	F1 Score Using All Features	F1 Score Using Central Pixel	F1 Score Using All Features (Bootstrap)
0	0.642 ± 0.103	0.638 ± 0.094	0.629 ± 0.094
1	0.679 ± 0.045	0.689 ± 0.046	0.693 ± 0.047
2	0.633 ± 0.097	0.647 ± 0.088	———

All models regardless of the way the data was split or how many features were used performed similarly with average F1 score in the mid 0.6 range. Split 1 for all methods produced the highest averages with the smallest deviation but was not significantly different from the other data splits. The F1 scores for this dataset might be lower due to the outliers within the normal cluster and closer to the mean. This method would not detect these points because they are within the normal range, and with so few outliers in the dataset, these incorrectly labeled points might have a larger effect on the F1 score.

For the shuttle dataset, the Gaussian distribution model worked very well. All three data splits generated similar results with average F1 scores of 0.97 as shown in Table 3 below:

Table 3. F1 Scores for Shuttle Dataset Using Gaussian Distribution

Data Split	F1 Score
0	0.968 ± 0.005

1	0.972 ± 0.002
2	0.968 ± 0.004

Similar to the satimage dataset, Split 1 had the highest average with the smallest standard deviation but the F1 score was not significantly higher than either of the other models. The majority of anomalous samples were far from the normal data, similar to the satimage dataset. However, because there is a higher percentage of outliers, the model can be more accurately trained, and the F1 score is not as significantly lowered by a single incorrect prediction. This dataset had the highest average F1 score using the Gaussian distribution model.

The Gaussian distribution model did not work well for the vowels dataset as shown in Table 4 below:

Table 4. F1 Scores for Vowels Dataset Using Gaussian Distribution

Data Split	F1 Score
0	0.154 ± 0.061
1	0.211 ± 0.035
2	0.157 ± 0.060

These scores were likely low due to the proximity of the outliers to the normal data. The outliers were almost indistinguishable from the normal data in terms of distance from the mean, making it difficult to identify them using this method.

For all the datasets using Gaussian Distribution, Split 1 generally produced the highest and most precise results. For the cardio dataset, the bootstrap method is necessary to ensure consistency. The method worked adequately for the satimage dataset but produced lower F1 scores due to the low percentage of outliers. Gaussian Distribution worked best for the shuttle dataset containing a higher percentage of outliers that were further from the mean and did not perform well for the vowels dataset which was to be expected because the outliers are close to the mean.

The other supervised learning method tested was logistic regression. For this method the theta parameters (weights for each feature) were optimized using fmincg to minimize the logistic regression cost function for the training set. The cost function was given by the following equation:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

(Eqn. 2)

where m is the number of samples, y is the actual output, $h_\theta(x)$ is the predicted output, n is the number of features, λ is the regularization parameter to prevent overfitting, and θ is the weights

for the features. With the calculated theta parameters, the sigmoid function was used to classify the test cases as either normal or outlier using the following equation:

$$h_\theta(x) = (1 + e^{-(\theta^T x)})^{-1}$$

(Eqn. 3)

where x is the values of the features. Split 1 cannot be used for logistic regression because the training set must include outliers. In testing Splits 0 and 2, the findings were the same, so we only look at Split 2 with randomly assigned train, validation, and test sets. The average F1 scores for each dataset are shown below with varying λ to best fit the data:

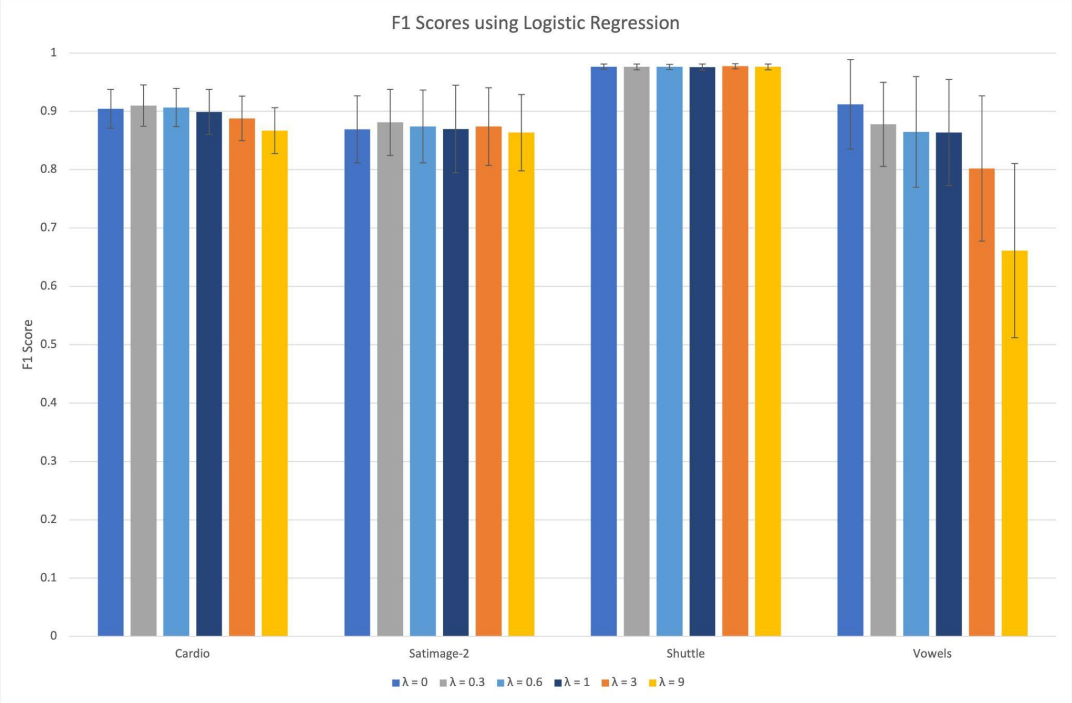

Figure 5. F1 Scores Using Logistic Regression

As shown in the figure above, the regularization parameter generally did not affect the accuracy of the model. The only dataset that showed a significant difference in performance due to the regularization parameter was the vowels dataset. For this dataset the average F1 score decreased and standard deviation increased as λ increased. This could be due to underfitting with a higher regularization parameter.

To determine the best model for each dataset, λ was varied, ranging from 0 to 9, and optimized to minimize the cost in the cross-validation set. The optimal models for each case are shown below in Table 5:

Table 5. F1 Scores for Regularized Logistic Regression

Dataset	λ	F1 Score
cardio	0.01	0.903 ± 0.037
satimage	0.01	0.873 ± 0.068
shuttle	1	0.976 ± 0.005
vowels	0	0.912 ± 0.077

This method worked well for all of the datasets, producing F1 scores greater than 0.80. Similar to the Gaussian distribution model, the shuttle dataset generated the best results using this method with the highest F1 score of 0.976 and the lowest standard deviation of 0.005, while the satimage dataset had a significantly lower score of 0.873 with a much higher standard deviation of 0.068. This is again likely due to the low percentage of outliers in the satimage dataset. With only 1.3% anomalous data, it is more difficult to model and correctly predict outliers. On the other hand, unlike Gaussian distribution, logistic regression worked well for the vowels dataset as seen in the table above with an average score of 0.912 ± 0.077. This is significantly better than the previous method for this dataset. However, it worked best without regularization; adding a regularization parameter decreased the average F1 score, increased the standard deviation, and increased the cross-validation cost. This was the only dataset used that minimized the cross-validation cost without any regularization. The cardio dataset was optimized at $\lambda = 0.01$, and all datasets except shuttle generated significantly better results using logistic regression than the Gaussian distribution model.

The other methods tested were unsupervised learning methods from the scikit learn package in Python. We tested Local Outlier Factor (LOF), Isolation Forest, and Elliptic Envelope. These were trained with unlabeled data. The training set for novelty detection with LOF was assumed to be all inliers while LOF Outlier Detection and Isolation Forest performed outlier detection meaning that the training set assumed some points to be outliers. For Isolation Forest and Elliptic Envelope, we compared models that used training sets that did not include any outliers with those that did. For all unsupervised learning methods, the satimage-2 dataset was reduced to 4 features to save time.

One method tested was Isolation Forest. Isolation Forest assumes anomalies will have values different from normal, making it a good option even if not much is known about the dataset. Isolation Forest selects a feature and randomly picks a value in between the minimum and maximum of that feature to split the data. Repeating this process, it makes a tree with anomalies more likely to have shorter paths. When a sample has a shorter path on multiple random trees, or a forest, it is classified as an outlier. The parameters optimized for this model were the number of base estimators, the number of samples used to train each base estimator, and the number of features used to train each base estimator. The performance for each dataset using

Isolation Forest is shown below where Isolation Forest models do not include outliers in the training set while Isolation Forest (Outlier) models do:

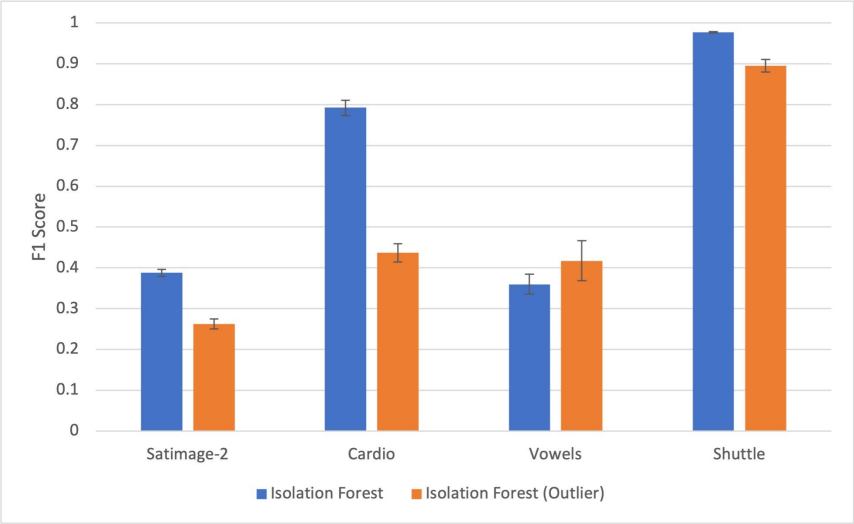

Figure 6. F1 Scores Using Isolation Forest

As shown on the graph, Isolation Forest worked well for the shuttle and cardio datasets, and Isolation Forest performed significantly better than Isolation Forest (Outlier) for all datasets except vowels. The cardio dataset saw an 80% increase in F1 score from outlier to non-outlier, and the better model found an average score of 0.792 ± 0.019. The results of the vowels dataset did not significantly differ between the two models, and neither model worked well. Both the satimage and vowels datasets performed poorly with F1 scores less than half of ideal. This is likely because of their low proportions of outliers. With less anomalous data to model, it is more difficult to predict outliers.

Another method tested was Local Outlier Factor (LOF). For this method we tested for both outlier and novelty detection. LOF classifies outliers based on the local density of a sample using k-nearest neighbors. Samples with lower density are determined outliers. Outlier detection takes the whole dataset and predicts which samples are anomalies based on local density. Novelty detection takes a training set of normal samples to determine where there are higher densities and can then predict anomalies when presented with new samples. For LOF, we optimized the number of neighbors to compare the sample to and the leaf size. For the chosen data, leaf size did not affect the performance of the model at all.

The performance of Local Outlier Factor for each dataset is shown below:

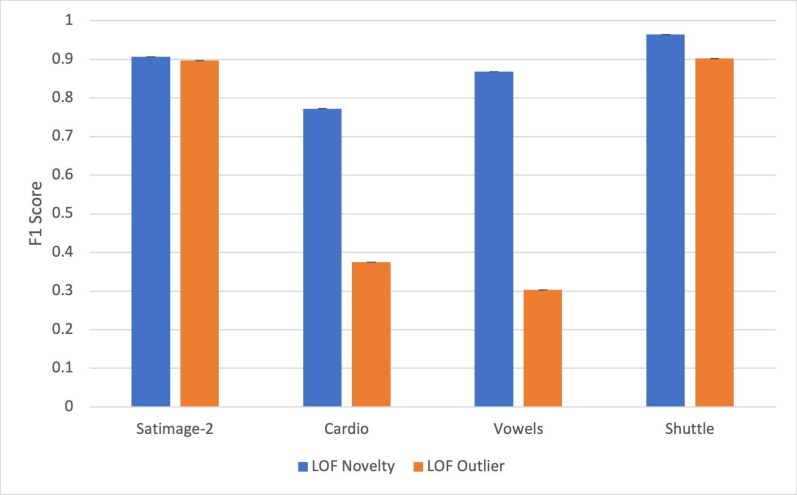

Figure 7. F1 Scores Using Local Outlier Factor

As shown in the figure above, LOF Outlier Detection performed very well for the satimage and shuttle dataset but not for the vowels and cardio datasets. This is likely because the outliers in the satimage and shuttle datasets are further away from the majority of data while many outliers in the vowels and cardio datasets are within the majority of the data points. LOF Novelty Detection worked well for all the datasets and was the best method for the vowels dataset. This model is trained on inlier data, so the higher density areas are determined by normal data. When the model predicts whether a point is an outlier or not, the space that the outliers occupy does not have high density, so it is more likely predicted correctly. While outliers in the vowels dataset are very near the inliers, they occupy the majority of one region and can be detected using this method. All of these models had almost no variance with -16 orders of magnitude. Generally LOF Novelty Detection performed best out of the unsupervised learning methods.

The last method tested was Elliptic Envelope which uses Gaussian distribution to fit an ellipse with outliers further from the center. It generally did not work well; even the shuttle dataset, which consistently produces F1 scores within 10% of 1, generated scores less than 0.80.

After evaluating many anomaly and outlier detection programs, the most consistent supervised learning algorithm was logistic regression and unsupervised learning algorithm, Local Outlier Factor Novelty Detection. These two models generally resulted in high F1 scores for all datasets evaluated, regardless of percentage of outliers and distance of outliers from the mean.

Sensitivity Analysis

We used the cardio dataset to perform sensitivity analysis on the unsupervised machine learning algorithms tested by varying the percentage of outliers from 1.2% to 9.6%. Like the models above, the parameters for these algorithms were optimized for the highest F1 scores. The results of this analysis are shown below in Figure 8:

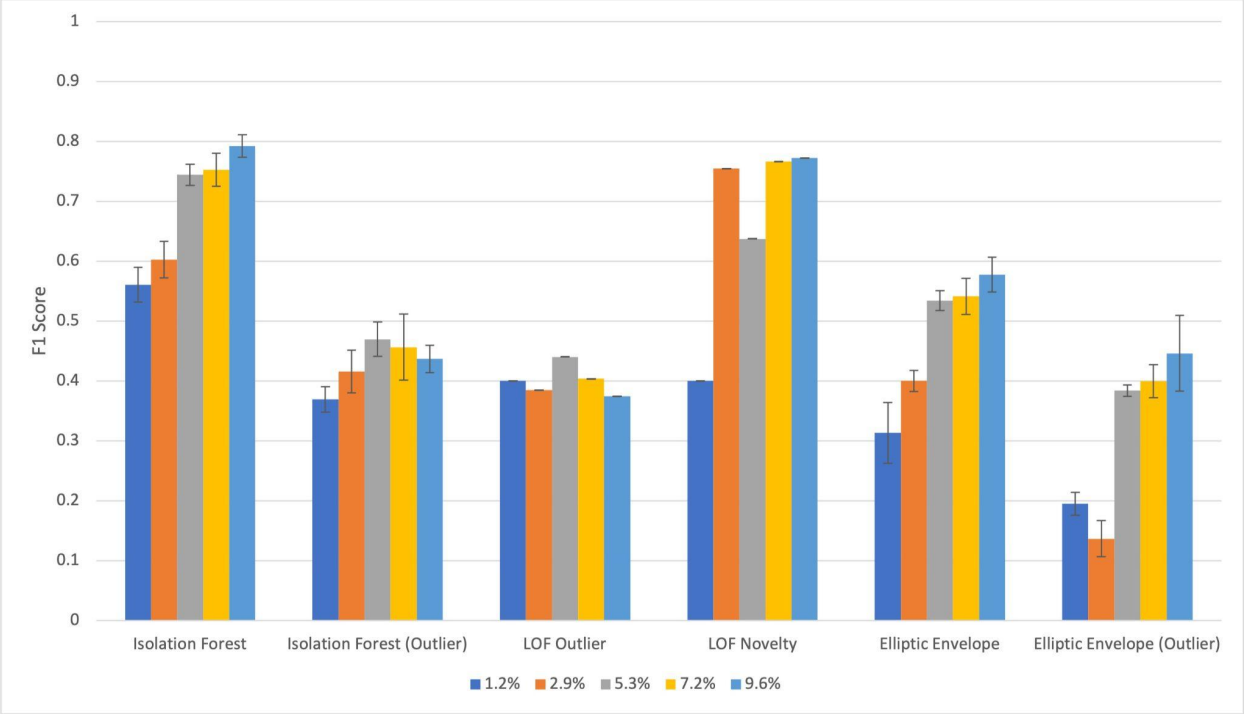

Figure 8. Performance of Sklearn Algorithms Based on Ratio of Outliers

As shown on the figure above, Isolation Forest and LOF follow the same trends as before with Isolation Forest without outliers in the training set outperforming Isolation Forest with outliers in the training set and LOF Novelty Detection performing significantly better than LOF Outlier Detection. Additionally the average F1 score for Isolation Forest increases as the ratio of outliers increases; the best performance is at 9.6% outliers, giving an F1 score significantly better than that of the model for 5.3% outliers. Elliptic Envelope follows a similar trend. Generally the model for 5.3% outliers performs significantly better than lower proportions, indicating that it is best to have at least 5.3% outliers in a dataset for more accurate predictions. For LOF Novelty Detection, there is an unexpected dip in F1 scores at 5.3% outliers. This could be due to the inlier points included in that sample set being very close to the outliers. Even with the dip, LOF Novelty Detection is still the best algorithm for all ratios but 1.2%. This suggests that regardless of the ratio of outliers in the dataset, LOF Novelty Detection will perform relatively well. Based on this analysis, Isolation Forest and LOF Novelty Detection are the best algorithms for a larger range of outlier ratios, and models generally perform better with outlier ratios greater than 5.3%.

MOF Dataset

Once the machine learning algorithms have been validated, we created the dataset of porous materials with known quantities of defects. For this project, we focused on Zirconium-based metal-organic frameworks generated by ToBaCCo, the Topologically Based Crystal Constructor. Using this program, we generated 8418 Zr-based MOFs with varying

topologies including bele, atne, and acs. One structure with an acs topology and 6c-Zr nodes is shown below in Figure 9:

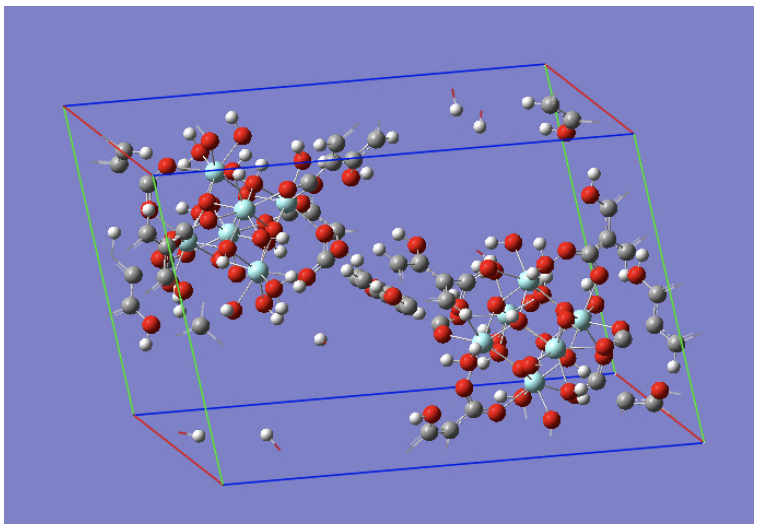

Figure 9. Unit Cell of a MOF with acs Topology and Zr Nodes

The structures were created in cif files, which can be viewed like the figure above. To add defects, we edited the cif files to delete Zr nodes and create new structures. All of these frameworks were then characterized using zeo++ by pore diameters, accessible volume, and surface area to be the features of the dataset. Our dataset includes 17 features and 8032 defected structures in addition to the structures without defects. These can be combined in many ways to optimize anomaly detection. However, many of the features are not affected by adding defects. When comparing the distribution for individual features of the normal and defective structures, they are almost indistinguishable as seen in the figures below:

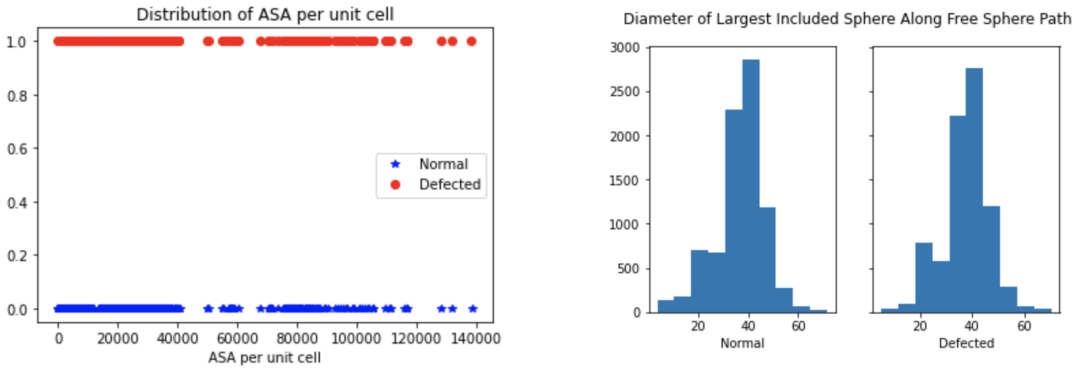

Figure 10. Distributions of Features by Classification

Anomaly Detection on MOF Dataset

Initially, we tried running the sklearn algorithms on our dataset with all of our undefective structures and 10% defective structures that were randomly selected. These models did not perform well with F1 scores less than 0.35 due to the lack of differentiation between the normal and defective structures shown above. Next we tried to vary the features used to see if

some were more affected by defects. This generated similar or worse results than the initial attempt, so we turned to principal component analysis (PCA). This allowed us to visualize the data and see how the data could be grouped. We also created features to quantify the defects: the ratio of the void fraction of the structure with defects to that of the normal structure and the ratio of the accessible volume (AV) per mass of the structure with defects to that without. Shown below is a PCA run with all the normal structures in blue and the defective structures with midrange AV per mass ratios shown in red:

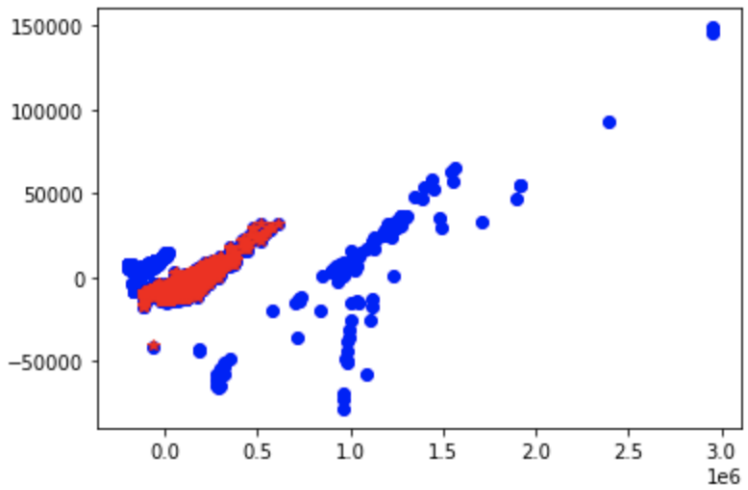

Figure 11. PCA on All Normal Data and Midrange AV per Mass Ratio

On the left, the blue cluster contains more than 5000 structures with the AV per mass ratios close to one, and to the right, the blue dots are the structures that have the lowest AV per mass ratio. Separating the data into these three clusters and running anomaly detection algorithms on each individually, Isolation Forest generated F1 scores up to 0.60.

We also tried to separate the data by topology. However, many of the topologies that comprised an adequate proportion of the dataset had ratios very close to one. While Isolation Forest produced F1 scores up to 0.50, the correctly detected defects did not seem to have any pattern and appeared random.

Finally we ran PCA on the structures with the lowest void fraction and AV per mass ratios. These structures have the most defects and should be the most distinguishable from normal data. Using only the structures with AV per mass ratios less than 0.91, Isolation Forest produced an F1 score of 0.762 and the following correctly predicted points:

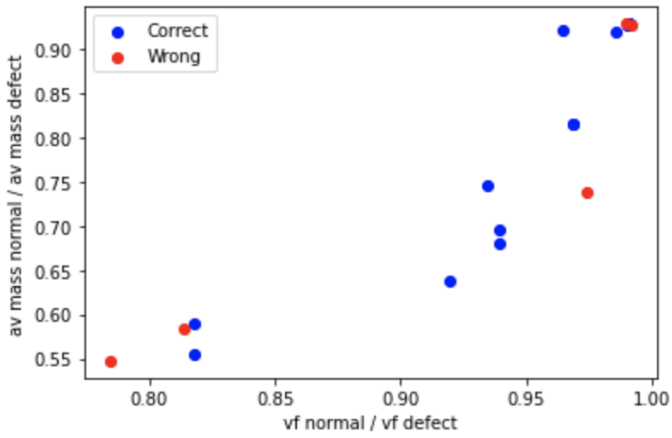

Figure 12. Prediction Accuracy of Isolation Forest on AV per Mass Ratio < 0.91

As shown in the figure, the model correctly predicted the middle void fraction ratios while incorrectly predicting the extremities. This might be fixed using a small slice of AV per mass ratio, but the higher F1 score suggests that we are moving in the right direction. More structures in this range of the ratios will be generated to execute these algorithms on a larger dataset.

References

(1) Vermoortele, Frederik, Bart Bueken, Gaëlle Le Bars, Ben Van de Voorde, Matthias Vandichel, Kristof Houthoofd, Alexandre Vimont, et al. 2013. “Synthesis Modulation as a Tool To Increase the Catalytic Activity of Metal–Organic Frameworks: The Unique Case of UiO-66(Zr).” *Journal of the American Chemical Society* 135 (31): 11465–68. https://doi.org/10.1021/ja405078u

(2) Qiu, T., Liang, Z., Guo, W., Tabassum, H., Gao, S., & Zou, R. (2020). Metal-Organic Framework-Based Materials for Energy Conversion and Storage. *Acs Energy Letters*, *5*(2), 520–532. https://doi.org/10.1021/acsenergylett.9b02625