Development of Predictive Model for Vertical Burn Flame Retardant Test from Microscale Combustion Calorimeter Data

ChBE 6745/4745 - Data Analytics for Chemical Engineers
Semester Project

11/25/2020

Po Hsien Hsu
Liyun Ren
Ziheng Shen
Jason Yao
Christian Herridge

1 Introduction

1.1 Project background

Microscale combustion calorimeter (MCC) is an efficient instrument to evaluate flammability of materials from only milligram sized samples[1-5]. MCC uses milligram-sized samples to study the pyrolysis of materials under controlled environment (atmosphere/heating rate/temp. range). Through the decomposition of the products, the heat release rate Q(t)(W/g) is calculated based on the oxygen consumption theory as a function of time. Due to the high cost of many standardized flame retardant (FR) tests, MCC can provide significant design flexibility for flame retardant material formations and rapid prototyping options.

However, the correlations of flame retardant testing results are often inconsistent with specific parameters extracted from MCC data. Although the correlation of MCC data and pure polymer materials has been demonstrated, the use of MCC for flame retardant reinforced polymer materials has not been broadly established to reliably predict FR test results. Despite its accuracy for measuring heat release characteristics of a pure polymer, the use of MCC for polymers containing flame retardants has not been widely demonstrated. Many studies have been dedicated to exploring the effectiveness and limitations of MCC for polymers containing flame retardants[6-10].

Here, different machine learning methods are explored to fully exploit the information obtained by MCC. A set of FR formulated adhesives and fiber materials are investigated to establish the correlation of MCC parameters with conventional flammability tests.

1.2 Overall goal

In this project, MCC data from 41 different kinds of adhesive materials and fibers were analyzed. The material set contains 22 different kinds of adhesive materials and 19 different fibrous materials with various FR formulations. Along with the MCC data, the dataset provided also contains information on pass/fail in accordance with FAR25.853 (a) vertical burn flame retardant test standard. The goal of this project is to establish a correlation between MCC data with actual flame retardant testing results. There are five useful parameters that can be extracted from MCC data and used as features of the model, including total heat released, heat release capacity, ignition temperature, burning temperature and fire growth capacity.

2 Data description

The original dataset contains heating rate(°C/s), temperature range(°C), total heat flow (W), Oxygen(%), Oxygen flow (cc/min), Nitrogen flow (cc/min) and HRR (W/g). The first step for data cleaning was to extract temperature range, HRR and heating rate from the original dataset since these are the main components needed to obtain four features which include Heat Release Capacity (HRC), Ignition Temperature(T_i), Burning Temperature(T_b), and Fire Growth Capacity (FGC). The new datasets containing points for each of the 41 materials used was converted into a .csv file for further processing and feature extraction. The flame retardant testing results of all the materials from FAR25.853(a) were also converted into a pass/fail dataset.

After converting these datasets into .csv files, the baseline correction operation was performed to compensate for the shift of HRR data with respect to temperatures. Uneven baselines in the HRR signal can bring uneven peak shifts across different temperatures leading to analysis errors. After doing baseline correction, it was possible to extract the four features of all datasets. All the code for baseline correction and feature extractions are shown in the Jupyter Notebook.

2.1 Dataset details

2.1.1 Data sources

These materials consist of different non-halogenated FR additives. Several milligrams for samples are weighed and used in a Govmark Microscale Combustion Calorimeter. The MCC was operated at a heating rate of 1 °C/sec to 750 °C in the pyrolysis zone. Oxygen and nitrogen flow rates were set at 20 and 80 cc/min, respectively.

2.1.2 Raw data

Fig. 1[11] shows a typical heat release rate versus temperature curve as obtained from Govmark MCC. With increasing temperature, the samples in MCC went through thermal decomposition. During the decomposition process, combustible gas products are yielded at elevated temperatures. The heat from combustion can be evaluated from MCC curve. At the end of the pyrolysis, the amount of solid left in the samples are char residuals.

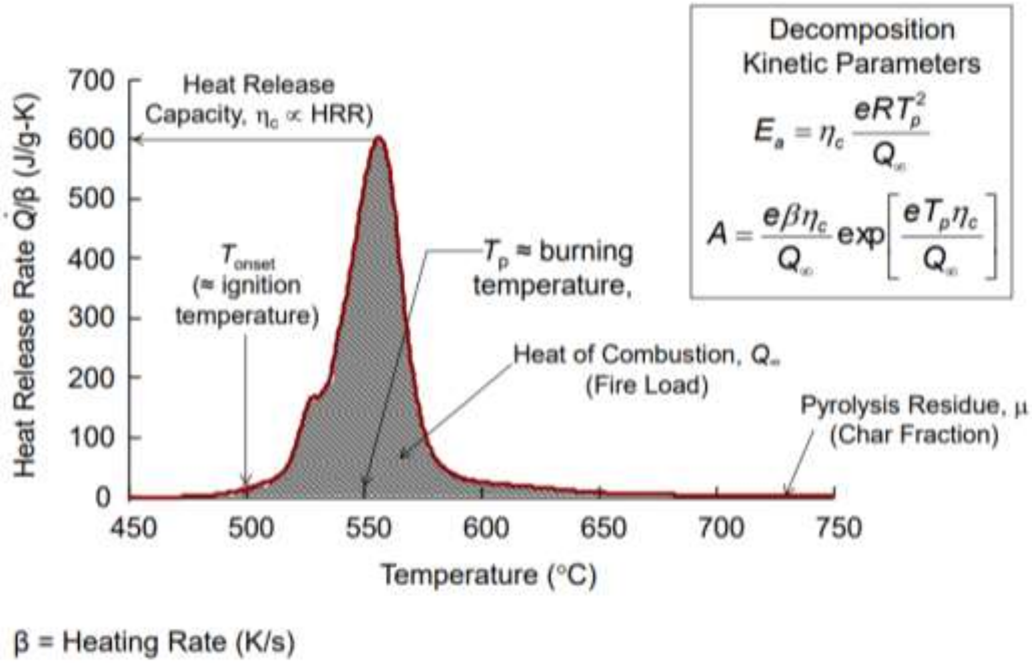

Figure 1 Heat release rate versus temperature curve as obtained from MCC **[11]**

2.1.3 Physical features

The physical meanings of the 5 features we manually extracted from the MCC & FR datasets are shown below:

- **Total heat release (Q)**
  The total amount of heat released through burning of the specific material. It can be calculated as the area under the MCC curve.
- **Heat release capacity (HRC)**
  Peak specific heat release rate (PHRR) divided by heating rate.
- **Ignition temperature (T_i)**
  Ignition temperature of a polymer is the temperature of the peak specific heat release rate (PSHRR).
- **Burning temperature (T_b)**
  Temperature when 95% of the total heat release(Q) from burning of the material.
- **Fire growth capacity (FGC)**
  FGC is defined by the following equation to include heat release capacity and ignition capacity of the material.

$$FGC = \left(\frac{\mathrm{Q}}{T_b - T_i}\right)\left(\frac{T_b - T_0}{T_i - T_0}\right), where\ T_0 = 25\ ℃$$

2.2 Methodology

2.2.1 Prediction models

Our project goal is to effectively predict the results of an FR test by passing in the features we acquired. By analyzing the results from our baseline model with Support Vector Classifier, we extended our predictive classifier models to K-Nearest Neighbors (KNN), Decision Tree, and the Random Forest. Our models are implemented using packages from sklearn.

- **SVC**: sklearn.svm.SVC
- **KNN**: sklearn.tree.KNeighborsClassifier
- **Decision Tree**: sklearn.tree.DecisionTreeClassifier
- **Random Forest**: sklearn.ensemble.RandomForestClassifier

2.2.4 Cross validation

A hold-out validation is obtained from our data using sklearn's train_test_split function to partition the data into training and testing sets and k-fold cross validation was used by the GridSearchCV method to optimize each model. Initially, our data ran into performance issues with cross validation due to the limited number of measurements and was too small for model training and validation. While the intention of CV is to reduce overfitting, the lack of data in each subset used by CV caused model performance to vary significantly depending on the chosen points. The sparse amount of data used for training did not generalize well enough—each model was underfit and made wild guesses rather than learning and fitting the patterns in the data. This was remedied with a generative method to create enough synthetic data points for cross validation to be applied effectively.

2.2.5 Hyperparameter tuning

The GridSearchCV algorithm, which takes as an input a range of values for critical hyperparameters, was implemented to our models. We then extracted the optimal hyperparameters as the model input. **Appendix C** describes the details of hyperparameters included for each model.

2.2.6 General strategy

The overall workflow for our prediction models is shown in the block pipeline diagram (**Fig. 2**). As shown with the black dashed lines, several physical features are extracted and rescaled as the input to our baseline model, which is a support vector classifier. An improved model, shown in the green box, was developed due of the poor performance and the lack of robustness of the baseline model. Two extra feature matrices are created to compare with the original one. Additionally, considering the small size of our dataset, a generative method is implemented to expand our feature matrices to almost 7 times of the original size. After that, the expanded feature matrices are transferred to KRR, Decision tree and random forest classifiers in addition to the baseline SVC model. The performance of the improved model is evaluated based on standard metrics and fitting speed.

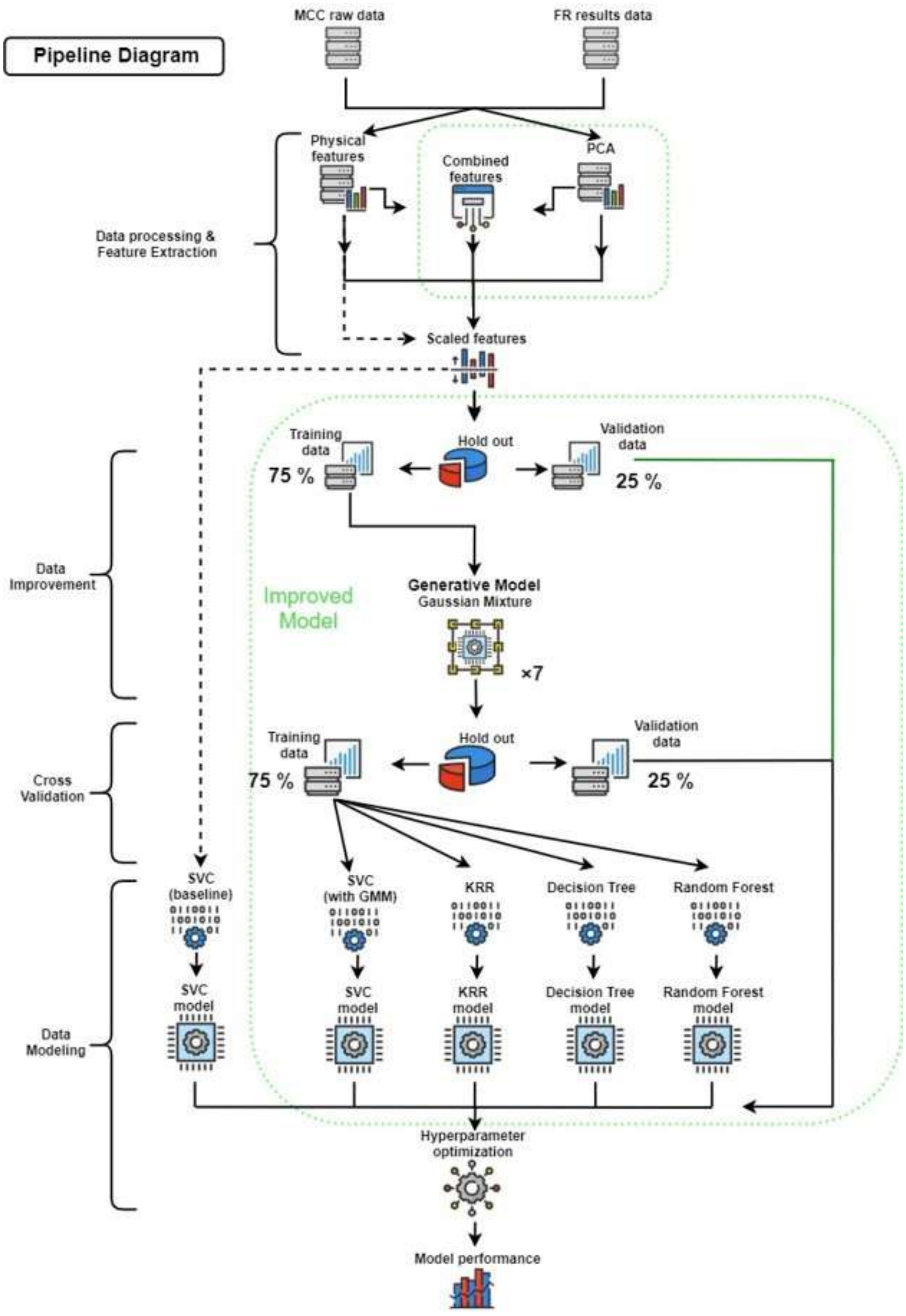

Figure 2 Pipeline Diagram

3 Results

3.1 Feature extraction

3.1.1 Physical features

Five features with physical meaning, as mentioned in section 2, are extracted using different functions. To verify the correctness and robustness of the feature extraction algorithms, THR and HRC (two of the five features derived from temperature and HRR data) for 21 adhesive materials were manually calculated using commercial software package Govmark Deatak. Results from software are benchmarked with the ones calculated by our codes, as shown in **Appendix B**. The THR and HRC values for only 4 adhesives demonstrate errors that are larger than 10%. Results for other samples show good consistency. The highest error found is 19.8% in **Appendix B**, which showed minimal impact on model accuracy. The discrepancy of the THR and HRC value is most likely originated from the different fitting algorithm of the Deatak software, which uses multiple gaussians to find the area under the peak.

3.1.2 Principal component analysis

Although a feature matrix consisting of parameters with physical meaning was created, it is worth exploring the prediction power between those manually extracted features and data-driven features, especially for our spectra-esque data. Therefore, principal component analysis (PCA) was performed, and several principal components were extracted from the raw MCC data (i.e., temperature and HRR data) with and without baseline correction. The extracted principal components, as well as the manually extracted features, were all be considered in the prediction.

The HRR values (y values) of all datapoints in each MCC curve were treated as input features of PCA as it is HRR values that determine the shape of curves. It is also worth noting that the temperature values (x values) of each observation are not aligned to each other. To improve the accuracy of PCA, a new grid of temperature values was created. This temperature grid is an array consisting of values uniformly distributed from 75 °C to 740 °C, which includes 1399 points in total. The HRR values of all observations were predicted using a 1-D linear interpolation based on the new temperature grid. As a result, 10 PCs were extracted out of the 1399 features. As shown in **Fig. 3**, the first 6 PCs contributed almost 90% of the total variance. The 10 PCs were reconstructed to the original coordinate, demonstrated in **Fig. 4**. Based on the figures, there was no appreciable difference between PCs from raw data and baseline corrected data. Consequently, the subsequent analyses are all performed based on the PCs extracted from baseline corrected data.

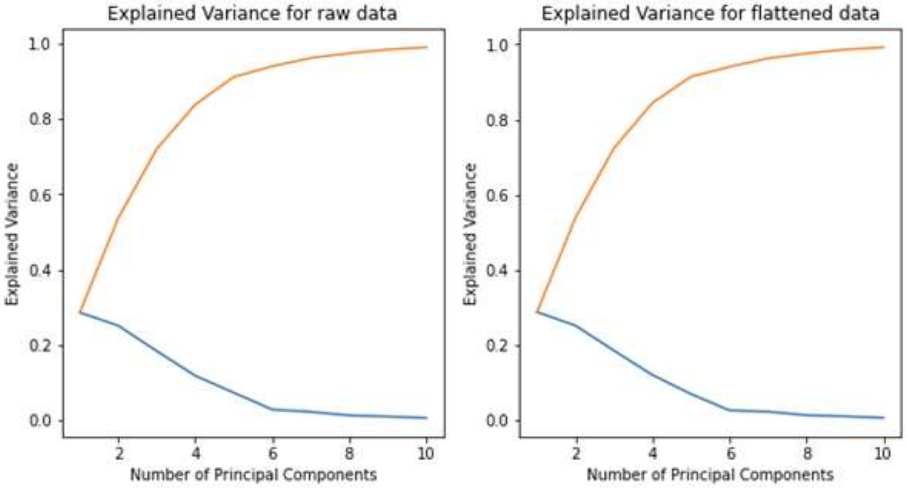

Figure 3 Explain variance and cumulative variance of the 10 principal components calculated from (left) original MCC data without any correction (right) baseline corrected MCC data

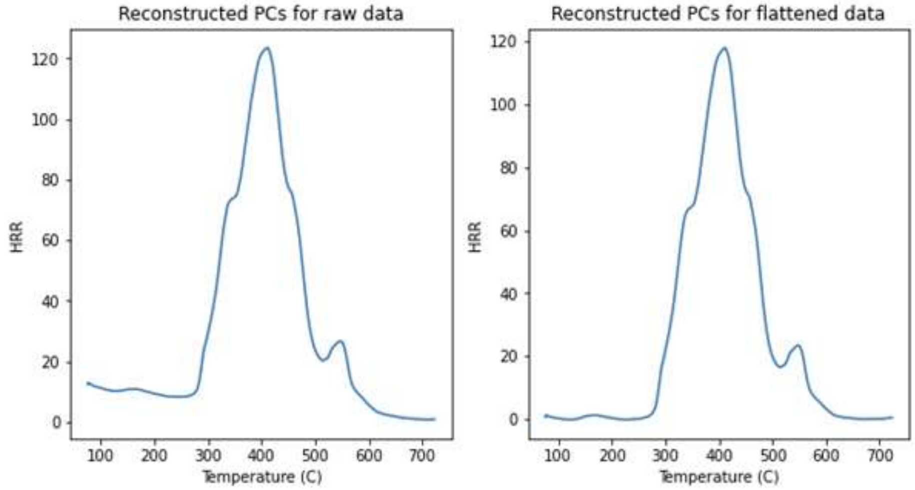

Figure 4 Reconstructed curve based on PCs for (left) original MCC data without any correction (right) baseline corrected MCC data

3.2 Generative methods

Gaussian Mixture Models (GMMs) are generative clustering method used to help address the issue of sparse data that applies to our data set. By using sklearn's GaussianMixture methods to fit a number of gaussians to the pass and fail classes of our data set, we were able to create a number of synthetic data points that closely followed the patterns present in the original data set. The number of gaussians was optimized using a Bayesian Information Criteria (BIC) and a final data set was created from the synthetic samples obtained from our GMMs. Because they are already a rare case, materials with borderline flame retardant properties are left as is and no synthetic data is generated for this class.

Initially, we train GMMs on all of our initial data points to assess their performance in fitting and synthetic sample generation. Since our data is sparse to begin with, the issue of variation in model fitting is still present, and the results for our BIC optimization vary between 5-8 component gaussians to use for the model. This means that it is difficult to pinpoint the exact best number of gaussians to use, due to the random nature of the model but can be remedied by setting a random state for the model. Furthermore, our GMMs tend to overfit the original data even after optimization, causing some strange artifacts in the generated data in the form of small linear clusters shown in **Fig. 5**. However, the model does a good overall job of preserving the trends in the original data while augmenting our data set with new synthetic points.

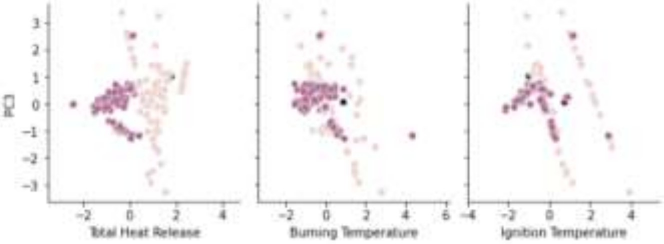

Figure 5 Linear Artifacts

To account for data leakage we reduced the number of data points used in GMM fitting to 75% and held-out the remaining 25% for model validation. GMMs were trained on the training set and the resulting features plotted against each other in **Fig. 6** shows the results of training on 75% of the dataset. Even with the smaller training set, GMMs perform quite well in generating points that follow the patterns in the original data. In this case, the points are noticeably less scattered and more concentrated into specific clusters, which seems to mimic the original data better than using all of the data. However, there is still variation present that slightly changes the results with different random_seed values used. We can use this new data to significantly improve model performance and successfully predict the classes of the materials in our hold-out 25% of the original data with high accuracy. While the generative method does help us improve model stability and predictive power, it also introduces bias. Even though this our hold-out method allowed for validation of these models, the synthetic points may not generalize well and perhaps may only be useful for allowing our model to predict FR test results for materials with similar properties to the adhesives and fibrous materials in our dataset. Extending this to use our models to predict the class of new materials remains to be tested.

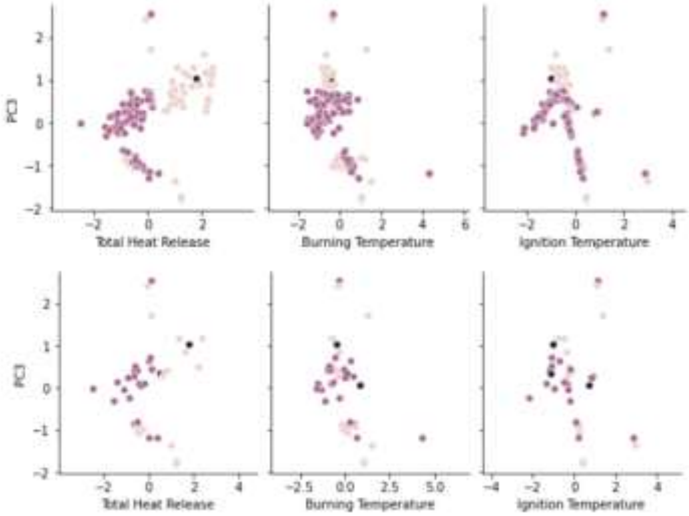

Figure 6 (Top row) GMM synthetic data trained with 75% of original data
(Bottom row) Original data

A Kernel Density Estimation (KDE) method was also tested for generation of synthetic data, but performed significantly worse than GMMs. At all bandwidths, significant artifacts are present that reduce the reliability of the data. For lower bandwidths, we see in **Fig. 7.** that small clusters are generated around the initial points rather than attempting to generalize for the entire data set—the model is extremely overfit. At higher bandwidths, unlike the GMMs, our KDE methods do not distinguish between areas of more concentrated points and areas were points are more scattered, causing the generated data to tend towards a normal gaussian distribution. These irregularities may be caused by the sparse data set, where our KDE models had trouble correctly fitting the data. Although the optimization of a KDE method could be further explored, we chose to utilize GMMs for their relative ease of use and good results.

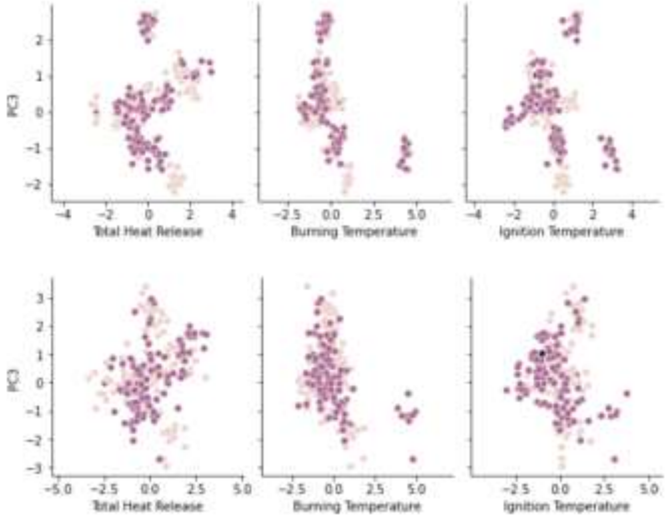

Figure 7 (Top row) KDE synthetic data at low bandwidths
(Bottom row) KDE synthetic data at high bandwidths

3.3 Standard Classification Performance Metrics

Standard classification performance metrics, including accuracy, precision and recall, are important parameters we used for evaluating the model performance. The full metrics of baseline model and improved models are listed in **Table E1** in Appendix E. In **Fig. 8** below, the accuracy metrics for our models when predicting the validation and original data sets are shown.

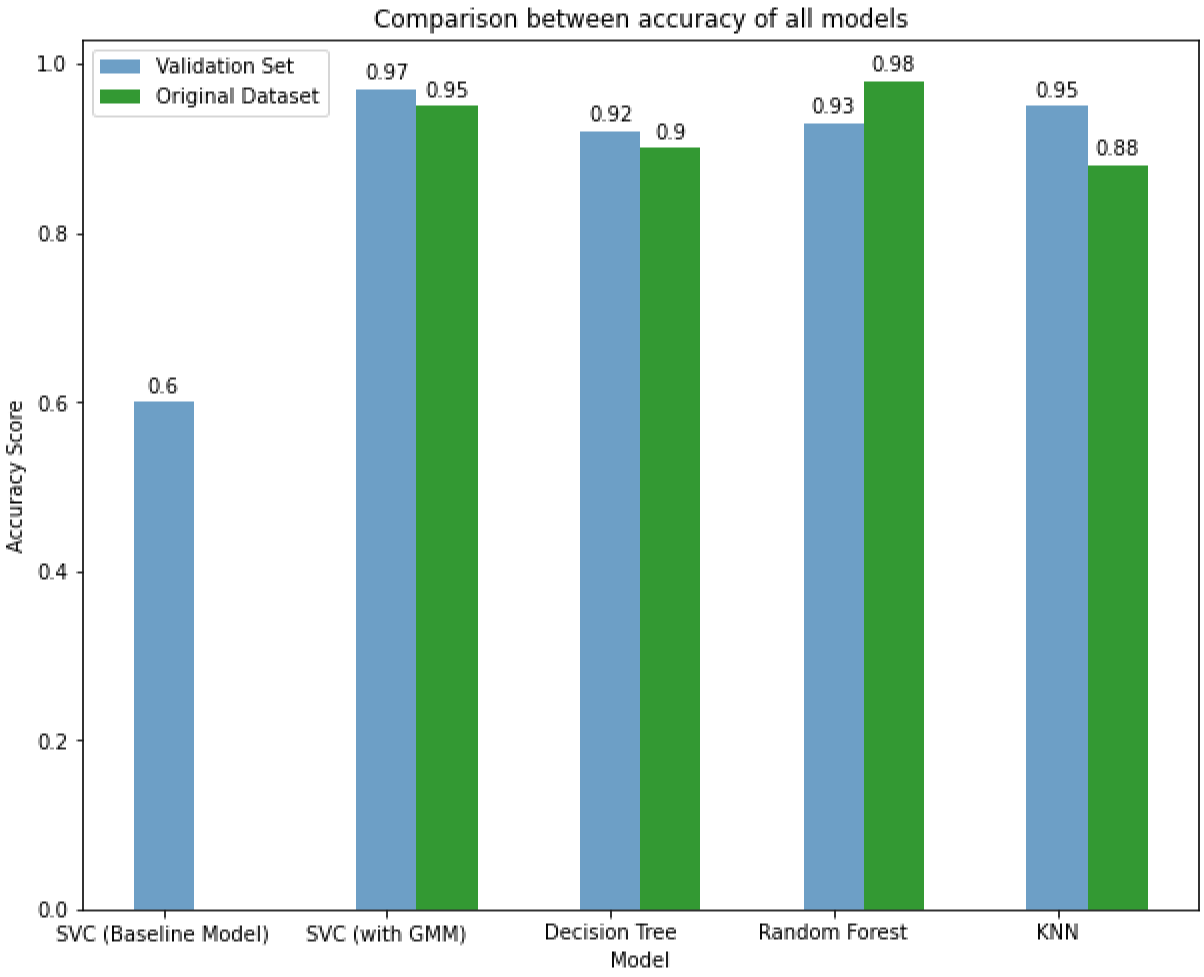

Figure 8 Comparison between accuracy of all models

3.3.1 Results of baseline model

After running several iterations of our baseline model, we noticed that the accuracy, precision, and recall metrics changed drastically based on the random training sets chosen. This is caused by the small size of our data set and is addressed in improved models by generating additional data through the discussed generative methods. Additionally, the model occasionally fails to predict the borderline class (label 2), which is composed of materials that do not have fixed flame retardant test results (sometimes pass and sometimes fail). With regards to scores of the validation set and the full dataset, it seems that the data is not overfitting much and seems to generalize well when applied to the larger data set.

3.3.2 Results of improved models

The generative model was successful in improving the power of our models to predict the results of a flame retardant test. The generated data estimated from our implementation of the gaussian mixture model gave us more flexibility and a larger data set on which to train and test our classifiers. We saw marked improvements across all performance metrics when predicting both the full set including our generated data as well as the original data set of empirical MCC data. As seen in **Table E1** in the appendix as well as **Fig 8**. above, the accuracy, precision, and recall classifier metrics for our baseline model scored poorly (accuracy of 0.60, and precision and recall values well below 0.5) before we applied the method to generate new data. When the SVC model was run using the new GMM data, we found good performance both for predicting the full data set as well as the original MCC data. The decision tree classifier was the only model that stands above the improved SVC with values of 0.98 for precision, recall, and accuracy when predicting the original MCC data.

3.4 Training Cost

In addition to the predictive power and classification performance metrics discussed above, model speed and efficiency is an important aspect in the model selection process and can be a tradeoff when discussing overall performance of one model over another. As the amount of data and the range of hyperparameters increases, model efficiency can become more important when choosing a model. Seen below in **Fig 9.**, wall-clock times were obtained for the model fitting process with the hyperparameters gained from the grid search. These values are also listed in Table E2 in Appendix E. Given these wall clock times, the random forest is not necessarily the best model to use. Due to the slow speed of the model, it may be pragmatic for real world applications to use the SVC model, where performance metrics are similar but run times are significantly lower. This would especially help in the case of much larger datasets potentially used in practice.

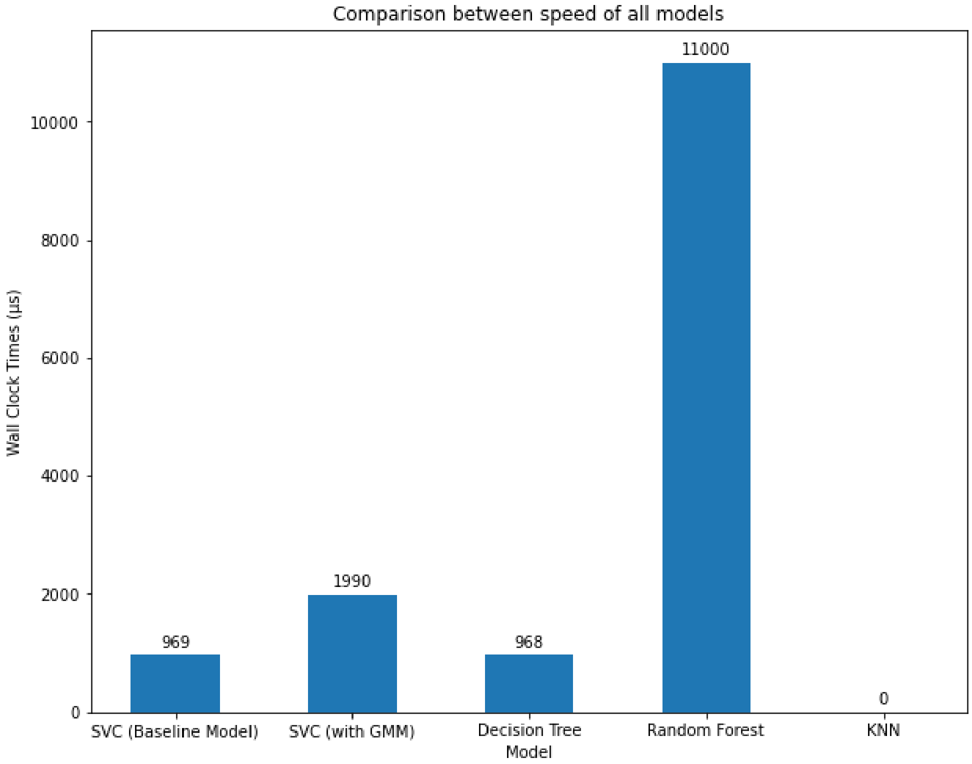

Figure 9 Comparison between speed of all models

4 Discussion

4.1 Effect of scaling

Both the unscaled data and scaled data were used for each model to see if the data scaling had significant impact on model performance. **Table 1** listed the comparison of performance metrics with and without feature scaling. For models trained by the unscaled data, SVC (baseline model) showed better accuracy comparing to scaled data. However, considering the poor robustness of the baseline model, this trend is not reliable. The decision tree model showed the most significant accuracy improvement after feature scaling.

Table 1 Performance metrics of all prediction models with and without feature scaling

Model	Original dataset	Accuracy	Precision	Recall
SVC (Baseline model)	unscaled	0.97	0.98	0.98
	scaled	0.70	0.48	0.50
SVC (with GMM)	unscaled	0.85	0.57	0.62
	scaled	0.82	0.55	0.6
Decision Tree	unscaled	0.82	0.55	0.60
	scaled	0.92	0.95	0.95
Random Forest	unscaled	0.92	0.94	0.94
	scaled	0.95	0.97	0.78
KNN	unscaled	0.83	0.55	0.6
	scaled	0.80	0.54	0.58

4.2 PCA features and physical features

Table 2 Performance metrics of all prediction models

Model	Feature matrix	Accuracy	Precision	Recall
SVC (Baseline model)	Physical	0.40	0.44	0.28
	PCs	0.60	0.39	0.56
	Combined	0.60	0.33	0.20
SVC (with GMM)	Physical	0.95	0.94	0.96
	PCs	0.95	0.63	0.65
	Combined	0.97	0.97	0.97
Decision Tree	Physical	0.97	0.64	0.67
	PCs	0.93	0.63	0.63
	Combined	0.92	0.94	0.94
Random Forest	Physical	0.97	0.64	0.66
	PCs	0.95	0.64	0.65
	Combined	0.93	0.93	0.93
KNN	Physical	0.98	0.98	0.98
	PCs	0.97	0.64	0.66
	Combined	0.95	0.64	0.64

Three feature matrices composed of: (1) 5 features with physical meanings (2) the first 5 principal components (PCs) (3) first 3 PCs in addition to 5 physical features, respectively, were used to compare the performance between manually extracted features and data-driven features. **Table 2** shows the performance metrics of all prediction models using different feature matrices. For conciseness, only results for validation sets are demonstrated. Surprisingly, no significant difference of the results can be observed among three feature matrices. All three feature matrices had equally poor for the baseline model. In fact, the baseline model has a severe robustness issue due to the small size. Therefore, results from baseline models are not reliable. For the other four prediction models, the three metrics all showed good accuracy for the validation sets, all of which are all greater than 0.9. The robustness also improved with the generated dataset. Therefore, we might preliminarily conclude that the manually extracted features perform almost equally with the PC features. It is most likely the generative method that improve the performance of all prediction models.

5 Future paths

The overall performance of our classification models showed promising results. However, there is an issue regarding the small datasets we now train our models with. This lack of data points leads to robustness issue as we mentioned in our discussions. We would implement the model with a larger dataset (wait for the data access from 3M company) for future work—the team is planning to apply the model with more MCC dataset from different FR additive reinforced polymer materials to evaluate the accuracy of current models. With a larger dataset, it is also possible to introduce char residual as an additional feature to the current predictive model and potentially fix the robustness issue of the models. To extend the current study, it is intended to categorize FR additive reinforced polymer materials based on FR mechanism (e.g. intumescent/char former/

endothermic cooling aids) to extract the most relevant features and accurate predictive model to direct FR additive formulation utilizing the machine learning approaches.

6 References

[1] R. E. Lyon and R. N. Walters, “Pyrolysis Combustion Flow Calorimetry,” J. Anal. Appl. Pyrolysis, 71, pg 27-46 (2004).

[2] R. E. Lyon, R. N. Walters and S. I. Stoliarov, “Thermal Analysis of Flammability,” Flame Retardant 2006, pp. 111-122 (2006).

[3] R. E. Lyon and R. N. Walters, “Microscale Combustion Calorimeter,” U.S. Patent 5981290 (Nov. 9, 1999).

[4] R. E. Lyon, “Flammability Tester,” U.S. Patent Application, U.S. 20060133445 A1 (June 22, 2006).

[5] R. E. Lyon, “Heat Release Rate Calorimeter for Milligram Samples,” U.S. Patent 6464391 B2 (Oct. 15, 2002).

[6] B. Schartel, K. H. Pawlowski, R. E. Lyon, “Pyrolysis Combustion Flow Calorimeter: A tool to assess flame retardated PC/ABS materials,” Thermochimica Acta, 462, pp. 1-14 (2007)

[7] R. E. Lyon, R. N. Walters, M. Beach, F.P.. Schall, “Flammability Screening of Plastics Containing Flame Retardant Additives,” Proceedings of the 16th International Conference ADDITIVES 2007

[8] R. E. Lyon, L. Speitel, R. N. Walters, S. Crowley, “Fireresistant Elastomers,” Fire Mater., 27, pp. 195-208 (2003)

[9] P. M. Hergenrother, C. M. Thompson, J. G. Smith, J. W. Connell, J. A. Hinkley, R. E. Lyon, R. Moulton, “ Flame Retardant Aircraft Epoxy Resins Containing Phosphorus,” Polymer, 46, pp. 5012-5024 (2005)

[10] R. E. Lyon, R. N. Walters and S.I. Stoliarov, “Screening Flame Retardants for Plastics Using Microscale Combustion Calorimetry,” Polymer Engineering & Science, 47, pp. 1501-1510 (2007)

[11] R. E. Lyon and M. L. Janssens, “Polymer Flammability,” Final Report DOT/FAA/AR-05/14 (May, 2005).

Appendix A: Individual contributions

Table S1 Individual contributions from each group member

Group Member	***Contributions***
Christian Herridge	Feature Extraction, Improved Models, Hyperparameter tuning, Model speed and efficiency, Improved Model Results
Po Hsien Hsu	Feature Extraction, Improved Models, Hyperparameter tuning, Pipeline
Jason Yao	Feature extraction, Data exploration, Data visualization, Generative methods
Liyun Ren	Establish project scope/Provide MCC data and project description/Data cleaning/Perform model comparisons with/without feature scaling/Finalize draft
Ziheng Shen	Feature extraction, Improved models, PCA, Comparison between PC features and physical features

Appendix B: Benchmarks

Table S2 Comparison between THR/HRC values obtained from codes and software

Total HR - software (kJ/g)	***Total HR - code (kJ*C/s*g)***	***Error of THR (%)***	***HRC-software (J/g*k)***	***HRC - code (J/g*k)***	***error of HRC (code) (%)***
24.50	24.72	0.90	212.70	226.09	5.92
12.40	14.88	16.66	155.30	167.30	7.17
15.07	16.58	9.15	103.00	115.39	10.74
20.13	20.25	0.56	221.00	234.51	5.76
26.37	26.62	0.96	399.00	403.93	1.22
12.97	12.58	-3.06	230.67	287.52	19.77
28.37	27.36	-3.66	413.33	444.42	6.99
12.93	14.54	11.05	177.33	187.06	5.20
15.77	15.67	-0.65	426.00	424.94	-0.25
13.00	13.54	3.96	199.67	223.62	10.71
13.65	14.18	3.73	209.50	229.02	8.53
13.80	13.73	-0.48	218.50	235.37	7.17
19.20	18.80	-2.12	158.00	149.57	-5.63
10.90	10.11	-7.83	146.00	153.65	4.98
22.80	22.68	-0.55	290.33	301.38	3.67
13.30	13.51	1.55	118.33	128.40	7.84
24.40	23.94	-1.92	326.33	340.66	4.21
20.00	19.29	-3.66	277.00	289.13	4.20
17.23	17.13	-0.61	159.00	161.62	1.62
16.60	16.83	1.36	146.00	167.60	12.89
9.20	9.45	2.60	170.00	173.07	1.78
11.30	11.48	1.54	187.00	198.00	5.55

Appendix C: Hyperparameter Tuning

Hyperparameters are parameters which control the overall learning process implemented by a given model. These are values we choose before implementing any given classification method and are not found by model training. Generally, the input for the scikit learn models used here takes a single value for any given hyperparameter. Given that these values are extremely important in the function of our models, we implement the GridSearchCV algorithm which takes as an input a range of values for critical hyperparameters. GridSearchCV performs an exhaustive search over these values one by one to find optimized hyperparameters which maximize the performance of the model. We then extract the optimal hyperparameters to input to our models.

Chosen Hyperparameters to vary:

- SVC: Inverse regularization strength (Cs) as well as the kernel coefficient for our radial basis function kernel used in the algorithm (gamma)
- Decision Tree: Maximum depth of the tree (max_depth) as well as the function which measures the quality of each split (criterion). The maximum depth parameter was tailored specifically to the size of our dataset.
- Random Forest: Number of trees in the forest (n_estimators), the function which measures the quality of the splitting (criterion), maximum depth of each tree (max_depth), and the number of features to look for when determining the best split (max_features)
- K-Nearest Neighbors: How many of the closest neighbors are used in the algorithm (n_neighbors) to optimize our model.

Appendix D: Model Descriptions

SVC (baseline model):

The model that we chose to use as our initial baseline model for this project was a support vector classifier. SVC is a powerful supervised learning model commonly used in classification problems which attempts to generate a hyperplane which divides the classes which has the greatest margin distance between classes. This tool uses ‘support vectors’ which are points near the margins as guides to maximize this distance which results in a robust predictive ability to differentiate between classes.

KNN:

The k-nearest neighbors classifier implements learning based on the k nearest neighbors to each point and is a flexible tool for non-linear classification. This model operates on a principle that is very easy to understand: democracy. The class of a point is determined by letting its k-nearest neighbors "vote" on which class it should be in. The advantage of democracy is that it is "nonlinear" - we can distinguish classes with very complex structures.

Decision Tree:

The decision tree classifier attempts to learn decision rules in order to predict values for the variables targeted. It uses observations it gains from the features of the learning data to construct a branched structure of rules. At the end of the 'branches' are 'leaves' which decide which target class a given input belongs to.

Random Forest:

The random forest ensemble classifier is an ensemble learning tool which consists of a collection of randomized decision trees. It implements averaging algorithms to predict classes based on the statistical average of the 'forest' of these decision trees. This method has good predictive ability as well as control against a single decision tree's potential for high variance and overfitting the data.

Appendix E: Performance Tables

Table E1: Standard Classification Performance Metrics for all Models

Model	Dataset	Accuracy	Precision	Recall
SVC (Baseline model)	Validation	0.60	0.33	0.20
	Full	0.80	0.71	0.67
SVC (with GMM)	Validation	0.97	0.97	0.97
	Full	0.98	0.99	0.99
	Original	0.95	0.97	0.97
Decision Tree	Validation	0.92	0.94	0.94
	Full	0.96	0.98	0.87
	Original	0.90	0.93	0.84
Random Forest	Validation	0.93	0.93	0.93
	Full	0.98	0.99	0.99
	Original	0.98	0.98	0.98
KNN	Validation	0.95	0.64	0.64
	Full	0.95	0.63	0.64
	Original	0.88	0.59	0.63

Table E2: Wall-clock times for GridSearchCV and Model Fitting

Model	***Model Fitting Wall time w/ Best Estimator Hyperparameters***
SVC (Baseline Model)	969 µs
SVC (w/ GMM)	1.99 ms
Decision Tree	968 µs
Random Forest	11 ms
KNN	0 ns