Comparative Analysis of Various Machine Learning Algorithms for Predicting Plastic Degradation in Oceans

Andrew Albano, Kalkidan Gebru,
{aalbano, kgebru}@umich.edu

***Abstract*—The study of polymer degradation is gaining major attraction from scientists to policy-makers. Although extensive research has been done using different approaches to link properties to degradation types using multi-physics modeling, experimental testing, there have been limited machine learning approaches taken to understand the features that strongly affect degradation. Particularly there is a gap in classifying the degradation rates based on polymer properties. Here we use data from experimental testing to train different machine learning models to benchmark their performance on the dataset.**

I. Motivation

Optimistic promises of the benefits of plastics have largely come to fruition with their set of diverse properties and versatility, however, the challenge of waste management, particularly the accumulation of plastic in the ocean, has gained traction as a growing global challenge [1] with many important questions surrounding this global dilemma. Large or clustered floating debris colonized by marine organisms serves as transportation for non-native species to new environments [2]. Another area of particular concern is the abundance of small plastic fragments and microplastics that can be ingested by marine that has the potential to transfer toxic substances to the food chain [3]. Given this issue, we consider how the collection, sorting, and identification of ocean plastics at a molecular level can be used to form a predictive model that answers questions about the degradation rate of these plastics in the ocean. Previous work has predicted a hierarchy of features [4] that regulate degradation by linking abiotic and biotic degradation behavior in seawater with physical properties. Using the same data set as [4], our goal was to compare the appropriateness of various algorithms to classify ocean plastics into fast, medium, and slow degradation categories using polymer properties as features.

II. Related Work

We present an overview of three mechanisms that influence marine plastic degradation at the molecular level. Next, we discuss some of the quantifiable metrics that have been previously linked to degradation. Lastly, we provide a brief review of the original machine learning model trained for the purpose of classifying degradation rate.

A. Degradation mechanisms

Plastic marine debris degradation is influenced by three mechanisms at a molecular level. First, depending on surface energy, bacteria colonize surfaces in the ocean and have a propensity for biofilm formation [5–7] resulting in mass loss due to surface erosion. Second, abiotic hydrolysis of functional groups severs the large macromolecules comprising a piece of plastic, thereby reducing molecular weight [8]. Lastly, exposure to UV radiation and oxygen results in photodegradation [9, 10].

B. Features previously linked to degradation

Daglen et al. [11] measured the yield for various polymers with different glass transition temperatures while at 26 °C. The yields were significantly higher when $T_g < 26°C$. The study suggests that when the temperature is greater than T_g the degradation rate is higher. This observation is reinforced for most polymers in the dataset by [4].

The presence of particular bonds, impacts the photodegradation rate. Previous work reveals that reactivity decreases as follows: $PVC > PS > PP > PE$ [8].

Delre et. al. [12] notes that photodegradation is mostly confined to the particle surface where UV photons are absorbed. Consequently, small particles with a high surface area to volume ratio degrade faster than larger particles with a lower surface area to volume ratio.

Hydrophocity, quantified by $LogP(SA)^{-1}$, predicts water solubility [13]. Min et.al. [4] notes that $LogP(SA)^{-1}$ values have sensitivity to hybridization (i.e., $\%sp^3$ and sp^2 carbons), density, large numbers of atoms, and how these atoms are connected. They found that this descriptor applied to more polymers than single features, like % nitrogen atoms, which works well for PA, or % sp^3 carbons, which was informative but better suited for a single class of polymers, like polyesters. Lastly, Min et.al. [4] observed that abiotic hydrolysis is more sensitive to increases in hydrophobicity, enthalpy of melting, and % crystallinity than biotic processes.

C. Prior machine learning implementations

Aside from the initial study by Min et.al.[4], the use of machine learning to classify degradation rates is novel research. Min et.al. [4] trained decision trees with a depth of 2 - 3 levels, using M_n , T_g, enthalpy of melting, and $LogP(SA)^{-1}$ for the features. Their accuracy increased from 72.2% with two-levels containing two features to 87.1% with three levels consisting of four features. While applying ten-fold cross validation, there accuracy was was 57.8% and 63.2% for the two level and three level models respectively.

III. Methodology

A. Dataset

Sourced from various studies by [4] with priority given to ocean studies or seawater in a lab setting, the dataset consists of over 110 polymers categorized by type of polymer, specimen shape, physical attributes, experimental parameters (i.e. time in seawater, temperature), weight loss during exposure, and abiotic and biotic conditions. A sample of the bulk property descriptors includes density, molecular weight, glass transition temperature T_g, enthalpy of melting, and % crystallinity. At a molecular level, the dataset considers the hybridization of various atoms, and considered architecture using the number and percentage of hydrogen atoms, cyclic rings, and other substituents per monomer. Currently, no other databases have been compiled for degradation classification.

There are several notable issues with the database. Since the dataset was compiled from multiple sources, there are three different reporting methods for weight loss making a ranking system difficult. An approach was developed, and [4] should be consulted for additional details.

B. Prescreening and feature selection considerations

: We chose to eliminate the features related to the physical attributes of the plastic because we are concerned with how degradation relates to polymer properties rather than the physical attributes of the plastic debris. For the same reasons, we eliminated features related to the architecture of the polymer, with the exception of $\%sp^3$ carbon and total sp^3 carbon due to its link to $LogP(SA)^{-1}$. We also disregard the weight loss reported in units of $(mg/(cm^2 day)$, opting for percentage loss per day metrics. We assume that the % weight loss per day and BOD $\%/day$ were linear, consistent with [4], allowing us to disregard the length of time in the ocean or simulated ocean environment.

The complex nature of polymers and their associated properties requires an understanding of the behavior at a molecular level to accurately estimate the remaining missing data points in the dataset. We chose not to use imputation methods to estimate the missing data since there was a significant variation of polymer types in the dataset and properties are often polymer-specific requiring models of the polymer to estimate. To determine the features, we plotted each of the features against each other and labeled each data point with its degradation classification rank to see if the data was linearly separable. We also split the dataset into subsets based on the weight loss reporting method. For each subset we employed listwise deletion (complete case analysis) for our initial feature selection screening. Any polymer that had one or more missing values in any of the remaining features was completely removed from the dataset. We used correlation matrices for both subsets to identify the Pearson correlation coefficient of each feature with the weight loss reporting method. We conducted a standard two-tailed t-test to check the significance of each correlation coefficient. To finalize our feature selection, we considered related studies that identify features that do not generalize to other polymers well and may be biased in the dataset due to the listwise deletion. We returned to the initial dataset and eliminated all features that were not selected, then used listwise deletion to drop data with missing values. We used a label encoder to convert the degradation rank into numerical form, and standardized the features. The dataset was then split into a training, validation, and test set.

C. SVM

Support Vector Machine (SVM) (Cortes & Vapnik, 1995) is a method for the classification of linear and nonlinear data, and uses nonlinear mapping to transform the original training data into a higher dimension. Support Vector Machines then search for the linear optimal separating hyperplane, which essentially is a boundary which separates the records into classes. The Support Vector Machine finds the optimal hyperplane using support vectors (which can be characterized as the most significant training records) and margins (which are defined by the support vectors) (Han et al., 2011). The Support Vector Machine can be trained using several functions: the Linear Kernel Function, Quadratic, Gaussian Radial Basis (GRB), Multilayer Perceptron Kernel (MP) functions.[14]

D. Linear SVM

Linear SVM is used for data that can be divided into two classes using a single straight line. This type of data is called linearly separable data, and the classifier employed is known as a Linear SVM classifier.[14]

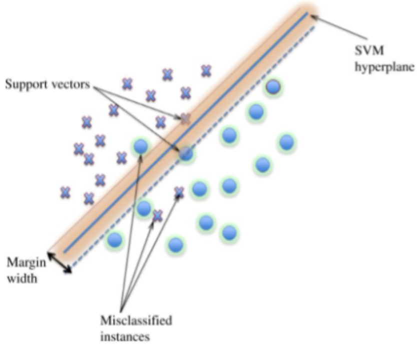

Fig. 1. Dual Soft Margin SVM Example

A linear SVM learns a linear decision boundary (through the origin) using:

$$f(\mathbf{x};\boldsymbol{\theta}) = \operatorname{sign}\big(g(\mathbf{x})\big) = \operatorname{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

And the training error is minimized using:

$$\begin{aligned}\mathrm{Err}_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{y_i \neq f(\mathbf{x}_i,\boldsymbol{\theta})} \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{y_i\cdot(\mathbf{w}^\top\mathbf{x}+b)\leq 0}\end{aligned}$$

We split the data into a training and test set using the Scikit-learn library with built-in function train_test_split(X, y, test size=0.3, random state=42). We initialized a hyper-parameter grid of 10^i from $i = -1$ to $i = 2$. For each feature in the selection list, we trained an SVM-linear model using the built-in SVC class. We iterated through the hyper-parameters using GridSearchCV, with 3-fold cross-validation, with random_state equal to 42 for reproducibility.

E. Polynomial SVM

In a polynomial kernel for SVM, the data is mapped into a higher-dimensional space using a polynomial function. The dot product of the data points in the original space and the polynomial function in the new space is then taken. The polynomial kernel is often used in SVM classification problems where the data is not linearly separable. By mapping the data into a higher-dimensional space, the polynomial kernel can sometimes find a hyperplane that separates the classes. [15]

For degree d polynomial, a homogeneous polynomial kernel is defined as:

$$K(\mathbf{u},\mathbf{v}) = (\langle \mathbf{u},\mathbf{v}\rangle)^m\,, \quad m = 1,2,\ldots$$

A non-homogeneous polynomial kernel is defined as:

$$K(\mathbf{u},\mathbf{v}) = (\langle \mathbf{u},\mathbf{v}\rangle + r)^m\,, \quad \forall r > 0,\, m = 1,2,\ldots$$

We also tried to enhance the performance of the polynomial by changing the hyper-parameter C. We split the data into a training and test set using the Scikit-learn library with built in function train_test_split(X, y, test_size=0.3). We initialized hyper-parameter values of 0.001, 0.005, 0.01, 0.02, 0.05, 0.1.

F. RBF SVM

An RBF neural network is a single hidden layer feedforward neural network (SLFN) that uses radial basis functions as activation functions.[16]

The RBF kernel is given by the expression below:

$$K(\mathbf{u},\mathbf{v}) = \exp\left(-\gamma\,\|\mathbf{u}-\mathbf{v}\|_2^2\right)$$

An RBF kernel is used as one of our methods to classify the dataset. We changed the hyper-parameters gamma and C to improve performance of the model. We used 3-fold cross-validation to test the various hyper-parameters on validation sets.

G. Neural Network

The extension of conventional artificial neural networks is deep neural networks. Compared to conventional neural networks, there are two main differences that deep neural networks have. It is for conventional neural networks to have one or two hidden layers. On the other hand, there are many hidden layers in deep neural networks. [10]

Formally, a neural network can be expressed as:

$$\begin{aligned} z(m) &= \sigma\left(\mathbf{v}_m^\top \mathbf{x} + b_m\right), \quad m = 1,\ldots,M \\ y(k) &= h\left(\mathbf{w}_k^\top \mathbf{z} + c_k\right), \quad k = 1,\ldots,K \end{aligned}$$

$$\mathbf{V} = \begin{bmatrix}\mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_m^\top\end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix}\mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_K^\top\end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_M\end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_K\end{bmatrix}.$$

We used a non-linear activation function, the softmax function, for multi-class classification. Represented by the function below:

$$t_k = \mathbf{w}_k^\top \mathbf{z} + c_k$$

$$y(k) = h(t_k) = \frac{e^{t_k}}{\sum_{j=1}^{K} e^{t_j}}$$

We used a cross-entropy loss function instead of mean square error function. The cross-entropy loss is given by:

$$H(p, q) = -\sum_{x \in \text{classes}} p(x) \log q(x)$$

Like the SVM models we used a built-in function to train the neural network. PyTorch was used to do training on the neural network. The test size used for our model was 0.2 and a 2 hidden layer neural network was defined for training our model. We also converted our PyTorch tensor to reside in CPU because the dataset was small.

IV. Results

A. Linear Separability

Prior to removing features we plotted each feature against each other to see if the data was linearly separable and determined that the data could not be linearly separated with a hard margin. In Figure 2 we show a sample of this analysis using $LogP(SA)^{-1}$ against four other features. 196 plots were generated and can be found on the git link provided in the Appendix.

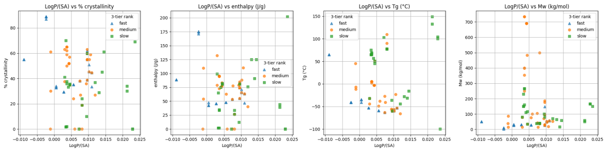

Fig. 2. A small sample of the 196 plots generated to check linearly separability between features

B. Feature Selection

Splitting the screened dataset by the weight loss reporting methods of $\%wt.loss/day$ and $BOD\%/day$, yielded subsets of sizes (65,13) and (15,13) respectively. Using the pandas library, we computed the correlation matrices using DataFrame.corr(method='pearson'). We considered the Pearson correlation coefficient of each feature with the weight loss and BOD, which are shown in Table I. A correlation heatmap for each reporting method is included in Appendix Figure 11 and Figure 12.

Performing the t-test using the Pearson correlation coefficient, yielded significant relationships for total sp^3 Carbon, M_w, M_n, T_g, T_m, and % crystallinity. Previous studies have identified that total sp^3 Carbon is better suited for a single class of polymers such as polyesters [4], and $LogP(SA)^{-1}$ generalizes to more polymers. This indicated that the data set may be biased towards polyesters. We replaced the feature with $LogP(SA)^{-1}$ for better generalization. M_w and M_n have strong correlation with each other (0.993754), so it is redundant to include both. Literature points out that the relationship between T_m values and degradation show opposite trends for polyesters and polyamides [4], and that using % crystallinity and enthalpy of melting allow an easier comparison of more polymers. Thus, we removed T_m and replaced it with enthalpy of melting. The final feature selections were M_w, T_g, $LogP(SA)^{-1}$, % crystallinity, and enthalpy of melting.

TABLE I
Pearson correlation coefficients for (% / day) and wt. loss (% / day) with each feature

Property	BOD (% day^{-1})	wt. loss (% day^{-1})
% sp3 C	0.386571	0.096774
(Tw-Tg)/(LogP/SA)	0.339553	0.124088
LogP(SA)-1 (Å^{-2})	0.318224	0.119470
Total sp3 C	0.259687	0.329406
Enthalpy (J g^{-1})	-0.267946	-0.140959
Mw (kg mol^{-1})	-0.297775	-0.210163
Mn (kg mol^{-1})	-0.311399	-0.218463
Den (g mL^{-1})	-0.381368	-0.087104
Mw/Mn	-0.431366	0.125908
Tg (°C)	-0.553801	-0.295374
% cryst	-0.595591	-0.107286
Tm (°C)	-0.646002	-0.358992

C. Linear SVM

The model with features $LogP(SA)^{-1}$ and M_w and hyper-parameter equal to 100, shown in Figure 3 obtained the best results on the test set. The accuracy of the training set was 60% and the accuracy of the test set was 71%. We highlight the results of this model in particular to show the impact of the dataset size and split on the test accuracy. The result of this model is significantly higher than the training set because of bias towards medium degradation polymers in the test set compared to fast and slow polymers which were misclassified in the training set. This points to the need to employ better prescreening techniques to ensure an appropriate distribution of degradation rates in both training and test sets. Other feature combinations ranged in accuracy from 35% to 65%, with the combination of $LogP(SA)^{-1}$ and T_g having the next highest accuracy. We include a complete compilation of all the generated plots in our git. The link can be found in the Appendix. Overall, we conclude that a linear model is not sufficient to classify the degradation rate of polymers, which prompts the exploration of non-linear models.

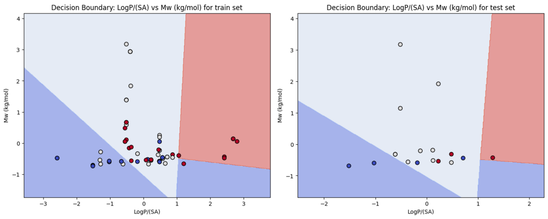

Fig. 3. Training (Left) and Test (Right) Set of the linear SVM model with hyperparameter equal to 100 using the features $LogP(SA)^{-1}$ and M_w. Blue, white, and red represent fast, medium, and slow degradation rates

For parameter optimization on the remaining models, we limited the features to $LogP(SA)^{-1}$ and T_g. Literature review points to $LogP(SA)^{-1}$ as an indicator of abiotic hydrolysis, and T_g as a factor in photo-degradation. Combining these studies with the fact that it was the second highest performing feature combination in the linear-SVM model, gives reason to believe that the next focus should be tailored to this combination.

D. Polynomial SVM

Following the same procedure as the linear SVM, we trained the model with a degree 5 polynomial mapping. Due to the number of plots of each feature against each other, additional plots can be found on our git. Focusing on $LogP(SA)^{-1}$ and T_g we show the initial model results in Figure 4.

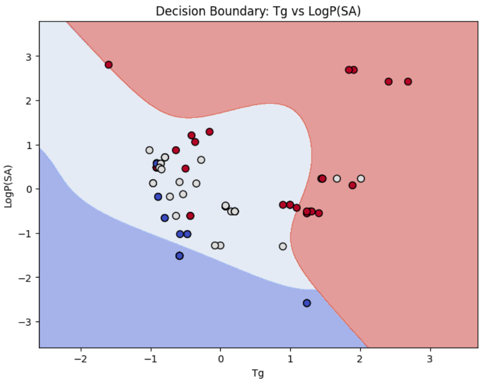

Fig. 4. $LogP(SA)^{-1}$ vs T_g decision boundary using degree 5 polynomial mapping without optimizing parameters and plotted with all data points

We then show how the accuracy changed as we tried different polynomial mappings while using hyperparameter C=100 in Figure 5. From Fig [5] it can be concluded that the test accuracy peaks at degree 5 with a testing accuracy of 61% and training accuracy of 63%. A high-degree polynomial kernel requires more data to accurately estimate the decision boundary without overfitting. Overall, as the degree increases, the number of features in these transformed spaces grows exponentially, enabling the model to fit increasingly intricate decision boundaries. With a simple dataset, the data over-fits quickly at higher degrees. A lower degree under-fits the dataset.

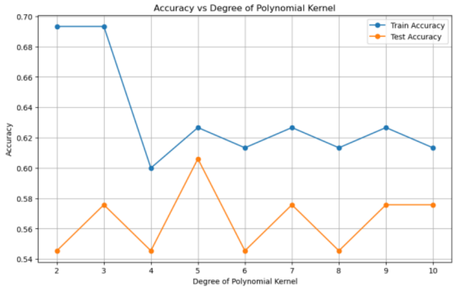

Fig. 5. Accuracy vs polynomial degree

We then show the change in accuracy as we adjusted the regularization parameter while using degree 5 polynomial mapping in Figure 6.

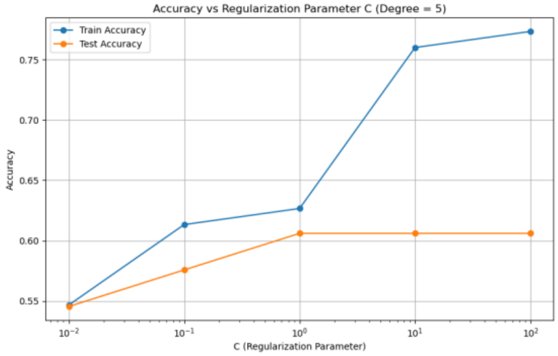

Fig. 6. Accuracy vs Regularization Parameter for a Polynomial Kernel.

Considering the impact of C, with small C, the optimization function tolerates more misclassified points to achieve a wide margin (uses a simple boundary). At higher C, most mis-classifications in the training data have already been corrected. However, increasing C further shows a plateau in performance as the model starts over fitting beyond a certain point. Nevertheless, the optimal accuracy for the given model is still limited to about 61%.

E. RBF SVM

Similarly, we considered each feature against each other with the RBF mapping. We include the initial decision boundary of $LogP(SA)^{-1}$ vs T_g without optimization efforts in Figure 7. This achieved 54% on the test set.

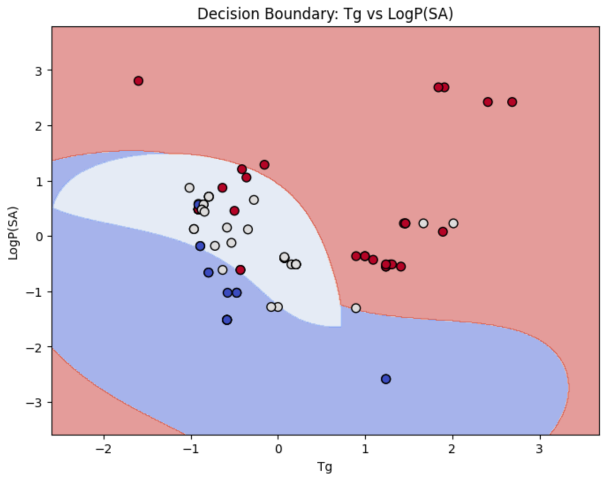

Fig. 7. Decision boundary of $LogP(SA)^{-1}$ and T_g model with RBF kernel before optimization efforts

We attempted to use smote, a resampling technique to handle unequal class distribution. We improved the overall accuracy to 65%. The decision boundary is shown in Figure 8.

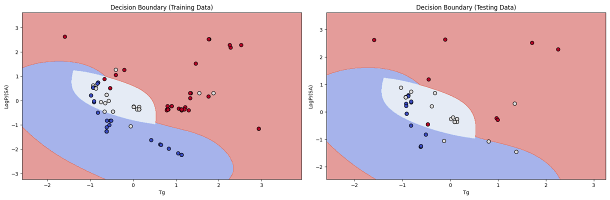

Fig. 8. $LogP(SA)^{-1}$ and T_g model with RBF kernel and class imbalance handled using smote resample function

We then attempted to improve the performance by varying the C and Gamma hyper-parameters as shown in Figure 9.

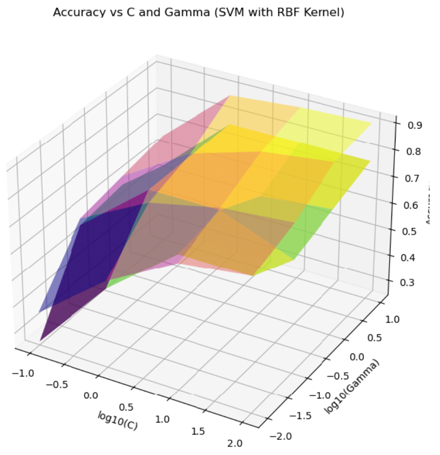

Fig. 9. Visualization of Accuracy vs. C and γ.

Similarly, for the RBF kernel it can be see that the model parameters perform better on a higher values of hyper-parameter values such as γ and C while its performance is poor in the lower values of the hyper-parameter. However, it should be noted that the model has limitations due to the limited size of the dataset.

F. Neural Network

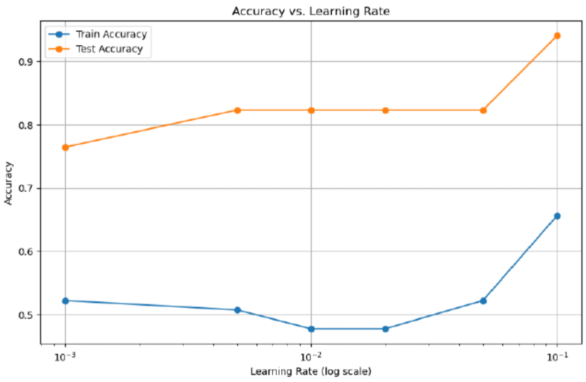

Fig. 10. Accuracy vs Learning Rate for a Shallow Neural Network.

Generally, from the result it can be seen that the shallow neural network has a better accuracy of classifying the data. At a small learning rate of Lr=0.001 the training accuracy is about 52.24 % while the testing accuracy is 76.47 % while at a bigger learning rate Lr=0.1 both the training accuracy and testing accuracy became 0.656 and 0.9412, respectively. This is due to the tiny gradient update at very small learning rates that causes the optimization process to move very slowly towards the loss minimum while at high learning rates the weight updates during the gradient descent are larger, enabling the model to quickly navigate the loss. Additionally, from the results it can also be seen that for intermediate values of learning rate it can be seen that the training accuracy decreases with increasing testing rate due to small training dataset which causes the optimizer fail to converge perfectly.

V. Conclusion

Overall, our training and test accuracy of our deep neural network has better learning rate has compared to SVM. However, our model is challenged due to the limited sample size of the dataset with a significant number of empty row that makes dataset too simple, which generally limits the training accuracy of each model used to train the dataset. We have also shown that the linear SVM model could not classify the degradation rates with sufficient accuracy. We have shown that despite the limited size of the dataset, k-fold cross-validation could be used to optimize the hyperparameters on both SVM models and neural networks. To improve the dataset further, a larger variety of PA, PS, and PU would be helpful.

The approach used in this paper was limited by several factors in the dataset. First, the 3-tier degradation rank assigned to each polymer was not directly based on the weight loss reporting method. In some instances, a polymer assigned a fast degradation rate (assigned via previous studies) had a smaller weight loss per day reported than that of polymers with a medium or slow degradation rate. This could point to non-linear degradation rates and various lengths of time in the ocean environment. Furthermore, by classifying the degradation rate using polymer properties alone, we neglect to consider the various biotic and abiotic factors that we know impact degradation, particularly due to biomass causing surface erosion, and temperature and C=O bonds contributing to photo-degradation.

During prescreening, we erroneously split the data set into training and test sets after conducting the feature selection. We also should have employed techniques to ensure less biased samples of the degradation rates in the training and testing sets.

References

[1] J. R. Jambeck *et al.*, "Plastic waste inputs from land into the ocean," *Science*, vol. 347, pp. 768–771, 2015.

[2] D. K. A. Barnes, "Invasions by marine life on plastic debris," *Nature*, vol. 416, pp. 808–809, 2002.

[3] E. L. Teuten, J. M. Saquing, D. R. U. Knappe, M. A. Barlaz, S. Jonsson, A. Björn *et al.*, "Transport and release of chemicals from plastics to the environment and to wildlife," *Royal Society*, vol. 364, 2009.

[4] K. Min *et al.*, "Ranking environmental degradation trends of plastic marine debris based on physical properties and molecular structure," *Nature Communications*, vol. 11, 2020.

[5] K. Lakshmi *et al.*, "Influence of surface characteristics on biofouling formed on polymers exposed to coastal sea waters of india," *Colloids and Surfaces B: Biointerfaces*, vol. 91, pp. 205–211, 2012.

[6] H.-C. Flemming and S. Wuertz, "Bacteria and archaea on earth and their abundance in biofilms," *Nature Reviews Microbiology*, vol. 17, pp. 247–260, 2019.

[7] T. Muthukumar *et al.*, "Fouling and stability of polymers and composites in marine environment," *International Biodeterioration & Biodegradation*, vol. 65, pp. 276–284, 2011.

[8] B. Gewert *et al.*, "Pathways for degradation of plastic polymers floating in the marine environment," *Environmental Science: Processes & Impacts*, vol. 17, pp. 1513–1521, 2015.

[9] ——, "Identification of chain scission products released to water by plastic exposed to ultraviolet light," *Environmental Science & Technology Letters*, vol. 5, pp. 272–276, 2018.

[10] B. Rånby, "Photodegradation and photo-oxidation of synthetic polymers," *Journal of Analytical and Applied Pyrolysis*, vol. 15, pp. 237–247, 1989.

[11] B. C. Daglen, "Factors affecting the photodegradation rates of polymers that contain (cyclopentadienyl-(carbon monoxide)-molybdenum) in the backbone," Ph.D. dissertation, University of Oregon, 2008.

[12] A. Delre, M. Goudriaan, V. H. Morales, A. Vaksmaa, R. T. Ndhlovu, M. Bass *et al.*, "Plastic photodegradation under simulated marine conditions," *Marine Pollution Bulletin*, vol. 187, 2023.

[13] E. Yildirim, D. Dakshinamoorthy, M. J. Peretic, M. A. Pasquinelli, and R. T. Mathers, "Synthetic design of polyester electrolytes guided by hydrophobicity calculations," *Macromolecules*, vol. 49, 2016.

[14] G. Cosma, D. Brown, M. Archer, M. Khan, and A. G. Pockley, "A survey on computational intelligence approaches for predictive modeling in prostate cancer," *Expert Systems with Applications*, vol. 70, 2017.

[15] PyCodeMates, "Svm kernels: Polynomial kernel," 2022, accessed: 2024-12-18. [Online]. Available: https://www.pycodemates.com/2022/10/svm-kernels-polynomial-kernel.html

[16] M. Kumar, R. Tiwari, R. Gupta, A. Kumar, M. Jain, and A. Thakur, "Prediction and application of solar

radiation with soft computing over traditional and conventional approach – a comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 136, p. 110432, 2021.

VI. Appendix

A. Supplementary Code

Supplementary code containing our models and additional plotted results can be accessed at our GitHub repository:
Ocean Plastic Degradation Prediction

B. Supplementary Figures

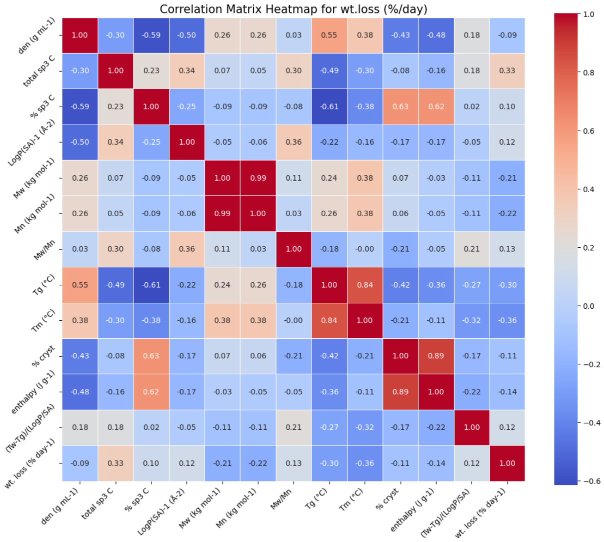

Fig. 11. A Pearson correlation coefficient of the features for polymers reported using wt.loss (% / day)

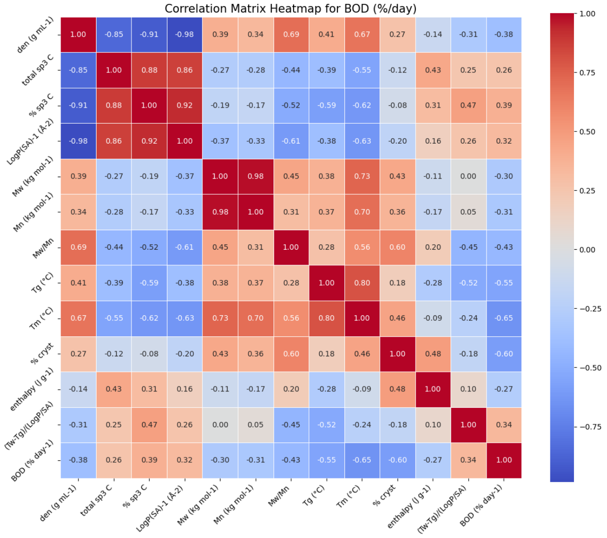

Fig. 12. A Pearson correlation coefficient of the features for polymers reported using BOD (% / day)