# A Novel Approach to Classifying Plastic Degrading Enzymes using Convolutional Neural Networks

Soharth Hasnat[1#], Fariha Anjum Shifa[2#], Shabab Murshed[2], Sarker Tanveer Ahmed Rumee[2*], MST Murshida Mahbub[1*]

[1]Department of Genetic Engineering and Biotechnology, East West University, Dhaka, Bangladesh

[2]Department of Computer Science and Engineering, University of Dhaka, Bangladesh

#Authors equally contributed to this work

**Correspondence to:** Sarker Tanveer Ahmed Rumee, E-mail: rumee@cse.du.ac.bd and MST Murshida Mahbub, E-mail: murshida@ewubd.edu

**Abstract**

The growing accumulation of plastic waste presents a significant environmental challenge, necessitating innovative approaches to mitigate its impact. Enzymatic degradation has emerged as a promising solution for addressing plastic pollution. However, the isolation and characterization of plastic-degrading enzymes (PDEs) through laboratory experiments are costly, time-consuming, and often complicated by nonculturable microorganisms. Consequently, accurate in silico identification of PDEs is desirable to explore the diversity of natural enzymes and harness their potential for combating plastic pollution. This study introduces a novel feature extraction strategy for identifying plastic-degrading enzymes, incorporating Autocorrelation (AAutoCor), Composition of k-spaced Amino Acid Pairs (KSAP), Dipeptide Deviation from Expected Mean (DDE), Composition/Transition/Distribution (C/T/D), Conjoint Triad, and Secondary Structure. A combination of ANOVA and XGBoost, feature selection methods, was applied to optimize the feature dimensions for improved performance. Seven supervised machine learning models were employed to evaluate the dataset: Convolutional Neural Network(CNN), Random Forest Classifier, Feedforward Neural Network, Logistic Regression, Naive Bayes Classifier, K-nearest Neighbor, and XGBoost Classifier. Among these models, the CNN model demonstrated the best performance, achieving an accuracy of 0.96, an F1 score of 0.80, and an ROC-AUC score of 0.96. These findings underscore the potential of the proposed system as an accurate predictor of plastic-degrading enzymes from environmental sequences. This approach significantly enhances efforts to develop sustainable solutions to plastic waste by accelerating the discovery of novel PDEs.

Keywords: Plastic-degradation; Machine-learning, Pollution, Enzymmes and sequence

# 1 Introduction

Plastic waste continuously poses threats to the soil, marine, and freshwater environments.[1,2] When broken down, plastics may convert into micro- and nano-plastics, which may enter the human body through ingestion or inhalation with significant health risks. Despite their harmful effects, the use of plastic is increasing daily along with growing populations.[3] Annually, over 0.3 billion tons of plastics are produced worldwide, and only 21% have been recycled; the rest is released to the environment.[4] To date, over 200 enzymes have been reported with the plastic degradation capacity.[5,6] The already discovered plastic degrading enzymes offer promise in identifying more such enzymes through computational approaches. Mining the huge sequence databases that may harbor undiscovered degrader sequences is timely and compelling as they may include superior features. Although laboratory-based experiments are continuously finding plastic degrading enzymes, they are time-consuming and costly.[7–9] While wet lab-based findings are the solid verification of the biological activities, computational screening can save time and costs which made the later approaches as routine procedures to predict a particular function before conducting lab-based experiments. Among the computational approaches, machine learning-based approaches are gaining popularity in finding protein functions from the sequence data. This study applied several machine learning methods using wet lab-verified plastic degrading enzymes as positive control compared with proven non-plastic degrading enzymes as negative control to get a highly accurate plastic degrader predictor. The organization of the paper is as follows. Section 2 describes the background information necessary for the study. In section 3 we discuss the methods and parameters of model used for the implementation. The experiment results and future works are presented in section 4.

# 2 Background

## 2.1 Performance metrics

### 2.1.1 F1-score

F1-score is utilized to quantify binary classification systems, classifying examples as 'positive' or 'negative.' It is the harmonic mean of the model's precision and recall.[10] The metric is calculated using the following formula:

$$F1 = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{1}$$

### 2.1.2 Matthews Correlation coefficient

The Matthews Correlation Coefficient (MCC), called the phi coefficient, is an evaluative metric for binary classification models. It offers a balanced evaluation of a model's effectiveness by considering true positives, true negatives, false positives, and false negatives. The MCC is determined by utilizing the following formula:

$$MCC = \frac{TP \times TN - FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{2}$$

Here: TP, TN, FP, and FN refer to True Positives, True Negatives, False Positives, and False Negatives respectively. MCC is useful for dealing with imbalanced datasets or evaluating the overall quality of a binary classification model. So, it is considered in our evaluation for balancing the trade-off between precision and recall.

#### 2.1.3 ROC curve

A Receiver Operator Characteristic (ROC) curve is a graphical plot that shows the diagnostic ability of binary classifiers.[11] It thoroughly analyzes the model's behavior when threshold values are considered, depicting the compromise between specificity and sensitivity. The Area Under the ROC Curve (AUC) encapsulates the interplay between these rates, concisely summarizing the model's overall performance. We utilized this metric to inform our decision-making process regarding the model's acceptability.

# 3 Method and materials

## 3.1 Training Dataset

### 3.1.1 Positive dataset

Two hundred and eight PD enzyme sequences were collected from the Plastic Degradation Database (Plastic DB) to create a positive dataset.[5] We further cleaned up the dataset by discarding 26 proteins that contained non-standard amino acids. Finally, the positive dataset contained 182 protein sequences (S1 File). The gathered PD proteins varied in length from 108 to 914.

### 3.1.2 Negative dataset

1523 non-PD protein sequences were extracted from UniProt to assemble the negative dataset.[12] We constructed a search model using an enhanced search option to find proteins that are not involved in the degradation activities. To do that, the names of twenty known PD enzymes were removed from the search model. Later, manual verification was performed to ensure that proteins only involved in the synthesis processes were included in the negative dataset. The supplementary file (S2 File) contains the constructed search model.

## 3.2 Feature extraction

Six descriptors have been chosen to extract features from the PD and non-PD enzymes (Table 1), and they are used to extract features from biological sequences.[13–17] This process used Biopython and ftrCOOL package (an RStudio package).[13,18]

| Descriptors | No. of features for each protein | Package | Extracted features | |
|---|---|---|---|---|
| | | | Positive Dataset | Negative Dataset |
| Autocorrelation (AAutoCor) | 240 | ftrCOOL | 43,680 | 365,520 |
| Composition of K-spaced amino Acids pairs (kSAP) | 3600 | ftrCOOL | 655,200 | 5,482,800 |
| Composition/Transition/ Distribution (C/T/D) | 343 | ftrCOOL | 28,126 | 522,389 |
| Conjoint triad | 343 | ftrCOOL | 28,126 | 522,389 |
| Dipeptide deviation from the expected mean (DDE) | 400 | ftrCOOL | 32800 | 609,200 |
| Helix, turn, and sheet | 03 | Biopython | 246 | 4,569 |
| Total | | | 788,178 | 7,506,867 |

Table 1: Descriptors of feature extraction methods

### 3.2.1 Autocorrelation

Autocorrelation is used to extract features based on the distribution of amino acid properties in the protein sequence.[19] To get the feature, we added three autocorrelation descriptors, Geary, Moran, and Normalized Moreau-Broto (NMBroto), in the R script. Subsequently, eight Aaidx (Amino acid index) IDs were incorporated into the script. Finally, the script was run with package ftrCOOL to acquire the feature matrix. The script is available in the supplementary file (S3 File).

### 3.2.2 Composition of k-Spaced Amino Acids pairs (KSAP)

This descriptor calculates the frequency of all amino acid pairs with k spaces.[13] In the R script, we added the range (rng) vector as a number, where each vector element shows the number of spaces between amino acid pairs. For each k in the rng vector, a new vector (whose size is 400) was created containing the frequency of pairs with k gaps. In our analysis, we set the rng value to ten (S3 File), which provided 3600 features for each protein in the feature matrix.

### 3.2.3 Conjoint Triad (CT)

We employed this descriptor to explore each amino acid's triad properties in protein sequences. For the calculation, this function turns 20 amino acids into seven classes according to their dipoles and volumes of the side chains. CT descriptor counts any three continuous amino acids as a unit resulting in 343 features extracted from each sequence.[13,20] This function returns a feature matrix corresponding to the R script (S3 file).

### 3.2.4 Composition/Transition/Distribution (C/T/D)

C/T/D is a group of descriptors representing the amino acid distribution pattern based on the protein's precise structural and physicochemical properties. Seven types of physical properties have been taken to calculate these features: hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, secondary structures, and solvent accessibility.[13] This group of descriptors can calculate each sequence's composition, transition, and distribution, and we use these parameters as vectors. Each element of the vector shows the number of spaces between the first and the second amino acids and the second and third amino acids of the tripeptide.[13] For each k in the rng vector, a new vector (size: $7^3$) is created, which contains the frequency of tri-amino acid with k gaps (S3 file).

### 3.2.5 Dipeptide Deviation from Expected mean (DDE)

DDE is considered one of the most critical descriptors for developing a sequence-based predictor for accurately identifying proteins.[17,21] Dipeptide composition has been widely adopted in various protein function prediction methods, such as B-cell linear epitope prediction.[22,23] For biological research purposes, the dipeptide combination of a protein sequence is primarily calculated as:

$$DC(m,n) = \frac{H_{mn}}{H-1}, m, n \in 2, ..., Y \tag{3}$$

Where $H_{mn}$ is the quantity of paired mn amino acids, and $H$ expresses the size of the protein sequence. Next, the theoretical mean (TM) and theoretical variance (TV) of a protein sequence are computed as:

$$TM(m,n) = \frac{C_m}{C_H} \times \frac{C_m}{C_H} \tag{4}$$

$$TV(m,n) = \frac{(TM(m,n)(1 - TM(m,n)))}{H - 1} \tag{5}$$

where $Cm$ is the number of codons, encrypting for the first amino acid, and $Cs$ is the number of codons, encrypting for the second amino acid in the given dipeptide. Finally, DDE is intended based on TV, TM, and DC as:

$$DDE(m,n) = \frac{(DC(m,n)(1 - TM(m,n)))}{\sqrt{TV(n,n)}} \tag{6}$$

#### 3.2.6 Feature of secondary structure (helix, turn, and sheet)

Helix, turn, and sheet are the principal structural features of proteins.[24] The secondary structures of plastic-degrading enzymes are expected to possess unique features. Hence, we extract quantitative features from the protein sequences by targeting the three crucial features (helix, turn, and sheet) using Biopython.[18]

### 3.3 Feature elimination methods

#### 3.3.1 Analysis of variance (ANOVA)

Our analysis determined the statistically significant difference between the means of three or more independent groups using ANOVA. It selects features by comparing the average of the dependent variable across different categories of each independent variable. This indicates that the independent variable significantly influences the dependent variable when there are substantial differences in the averages.[1] Additionally, ANOVA aids in ranking the identified features according to the computed F-statistic.[25,26]

#### 3.3.2 ANOVA and Permutation Feature Importance

An ensemble of feature selection strategies was employed, integrating ANOVA with Permutation Feature Importance (PFI). This approach involves the union of features selected by ANOVA and those identified through PFI, capitalizing on the strengths of both methodologies. ANOVA provides insight into the statistical significance of features, while PFI[27] evaluates the impact of each feature on model performance by measuring the degradation in performance when feature values are permuted. This combination of methods ensures that features are statistically significant and critical to maintaining high model performance, resulting in a more robust and reliable feature set.

#### 3.3.3 ANOVA and XGBoost Feature Selection

Another feature selection method employed was the integration of ANOVA with eXtreme Gradient Boosting (XGBoost).[28] XGBoost, a gradient boosting technique, assigns scores to features based on their contribution to improving model predictions across multiple

decision trees.[29,30] Combining ANOVA's statistical significance with XGBoost's predictive capability ensures that the features selected are statistically significant and highly influential in enhancing model accuracy. This hybrid approach provides a comprehensive and optimized set of features, thereby improving the overall performance and reliability of the predictive model.

| **Feature Name** | **Acc** | **F1** | **PR_AUC** | **ROC_AUC** | **MCC** |
|---|---|---|---|---|---|
| AutoCorrelation | 0.94 | 0.71 | 0.78 | 0.78 | 0.7 |
| Secondary Structure | 0.90 | 0.58 | 0.63 | 0.73 | 0.54 |
| C/T/D | 0.94 | 0.72 | 0.89 | 0.78 | 0.72 |
| DDE | 0.95 | 0.78 | 0.82 | 0.83 | 0.76 |
| KSAP | 0.94 | 0.74 | 0.80 | 0.81 | 0.73 |
| CJTriad | 0.94 | 0.72 | 0.75 | 0.81 | 0.69 |
| Feature Set(All features) | **0.95** | **0.78** | **0.84** | **0.90** | **0.75** |

Table 2: Comparison of individual features and feature set performance

## 3.4 Feature set/ descriptor set creation

Creating diverse feature combinations in machine learning can significantly enhance both model performance and interpretability. Each feature captures distinct aspects of the data, and their fusion can influence key performance metrics such as accuracy, precision, and recall. Through experiments comparing individual features and their combinations (Table 2), we observed that combining features uncovers deeper patterns in the data, leading to improved performance. To identify the optimal feature set, we constructed and evaluated five distinct feature sets using various machine-learning algorithms. Feature Set 1 includes Composition, Transition, and Distribution (CTD), Autocorrelation (Corr), Dipeptide Deviation from Expected Mean (DDE), K-spaced amino acid pair (KSAP), and Secondary Structure (SS). Feature Set 2 comprises CTD, Conjoint Triad (CT), DDE, KSAP, and SS. Feature Set 3 consists of CTD, CT, DDE, Corr, and SS. Feature Set 4 contains KSAP, CT, DDE, Corr, and SS. Finally, Feature Set 5 includes KSAP, Conjoint Triad (CJT), CTD, Corr, and SS. By evaluating the performance of these feature sets, we aim to determine which combination best captures the underlying data patterns and leads to the highest model accuracy.

## 3.5 Application of machine learning algorithms

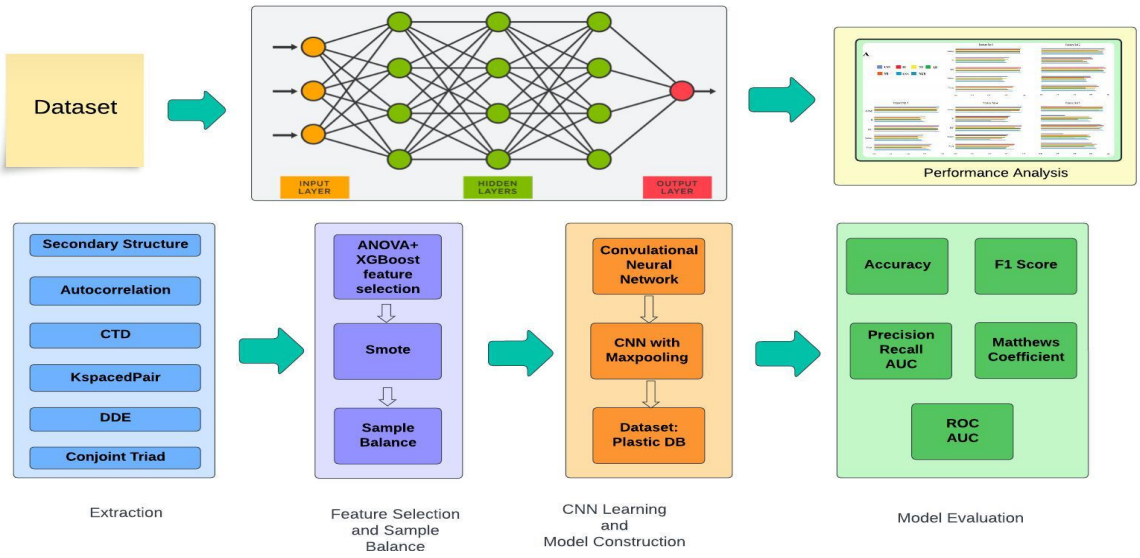


Figure 1: Work Flow Diagram

A total of seven different models, namely Convolutional Neural Network, Random Forest Classifier, Naive Bayes Classifier, XGBoost Classifier, K-Nearest Neighbor, Logistic Regression, and Basic Neural Network, were incorporated into this study. Each model technique was applied to the filtered dataset, but the Convolutional Neural Network (CNN) performed better considering the performance.

The CNN model was selected for the classification task in our investigation due to its superior accuracy compared to other machine-learning techniques. It used the feature sets as the input sequence feature vector. Five convolutional layers, three fully connected hidden layers, and a fully connected output layer with two output neurons that contain sigmoid activations for binary classifications make up the architecture of our deep neural network. Each convolutional layer has a kernel size of 1, a leakyReLU activation function, and batch normalization. The 1D filter performs correlation operations on the matrix. Since Sigmoid and Tanh can lead to the vanishing gradient problem, ReLu was utilized as the activation function. Batch normalization stabilizes the training by normalizing the internal covariate shift in each layer and reducing it.[31]

Additionally, it introduces some noise to lessen the overfitting of the data. A variety of filters are included in the convolution layer. Zero-padding is used in all convolutional layers to produce an output length equal to the input length. Unless otherwise stated, max-pooling with an identical window size and stride follows each convolutional layer. The fully connected hidden layer uses one hundred twenty-eight units with ReLU activation functions. Dropout, through random neuron deactivation during training, encourages the exploration of diverse paths and feature combinations. This enhances the model's capacity to generalize to unseen data, a critical aspect in developing an accurate system.[32]

In the training phase, a dropout probability of 0.2 was employed for the hidden layers as this value seemed to perform the best among other values. Regularizers are techniques used to prevent overfitting and make models more interpretable.[33] Common kernel regularizers such as L1 and L2 regularization add penalty terms to the loss function

to promote smaller weights in the model. After experimenting with various values, we used the L1 kernel regularizer with a strength of 0.001. All the parameters were initialized using He initialization,[34] and we trained the model for 74 epochs.

After each epoch, the parameters were updated using TensorFlow's implementation of Adam optimizer with a loss function of binary cross entropy. Three fully connected layers are then flattened, followed by a max-pooling layer. In CNN architectures, max pooling is frequently employed to minimize the spatial dimensions of feature maps while keeping crucial information. We used a sigmoid activation function in the output layer to determine the likelihood of each class label for classifying plastic-degrading enzymes.

### 3.6 Synthetic minority oversampling technique (SMOTE)

Our datasets have 182 samples of positive class and 1492 samples of negative class, which shows the necessity of using an oversampling technique to balance the data. Classification in an imbalanced dataset is challenging because of highly skewed data, with one class (the minority class) having significantly fewer examples than the other class (the majority class). This scenario can pose a bias towards the majority class, leading to a suboptimal result. We employed the SMOTE (Synthetic Minority Oversampling Technique) technique for synthesizing new samples to balance the data, thereby creating a more balanced distribution and improving the model's ability to learn from the minority class. We used the sampling strategy 'minority' to create new instances of minority class and balance the class distribution.[35]

## 4 Result and discussion

### 4.1 Feature Selection

When we observed the six individual features with enormous dimensions, it became evident that dimension reduction was necessary to filter out the most impactful ones. Retaining unnecessary features can increase dataset noise, adversely affecting model performance. It is important to note that there is no universal feature selection method; the choice of technique depends on the specific characteristics of the data. Consequently, three feature elimination methods were evaluated, as described in Section 3.3.

Before selecting an appropriate feature selection method for the dataset, we created five feature sets by combining the six individual features, as detailed in Section 3.4. Through rigorous analysis of the performance of these feature sets across the three distinct feature selection methods( table 3, it was observed that the ANOVA+XGBoost method consistently outperformed the other two methods across all five feature sets. A previous machine learning approach on plastic degrading enzymes only applied the XGB technique to select essential features.[36]

The integration of XGBoost[37] with ANOVA effectively leveraged the strengths of both techniques, resulting in more discriminative feature selection compared to the other two methods, thereby improving model performance. Moreover, the application of the ANOVA+XGBoost feature selection method led to significant dimension reduction in the features Autocorrelation, DDE, CTD, KSAP, SS, and Conjoint Triad, as shown in Table 4. The dimension of the Autocorrelation feature was reduced from 240 to 136, DDE from 400 to 108, CTD from 148 to 38, KSAP from 3600 to 109, and ConjointTriad from 343 to 137. This dimension reduction was crucial in mitigating potential overfitting in

| Feature Set | Method | Accuracy | F1 | ROC | Matthews | Pr_auc |
|---|---|---|---|---|---|---|
| Feature set 1 | ANOVA | 0.962 | 0.78 | 0.83 | 0.72 | 0.82 |
| | ANOVA + Permutation | 0.960 | 0.79 | 0.84 | 0.731 | 0.83 |
| | ANOVA + XGB_feature_selection | **0.961** | **0..795** | **0.85** | **0.735** | **0.84** |
| Feature set 2 | ANOVA | 0.95 | 0.76 | 0.81 | 0.75 | 0.81 |
| | ANOVA + Permutation | 0.960 | 0.78 | 0.84 | 0.77 | 0.82 |
| | ANOVA + XGB_feature_selection | **0.96** | **0.8** | **0.85** | **0.77** | **0.83** |
| Feature set 3 | ANOVA | 0.96 | 0.79 | 0.83 | 0.731 | 0.83 |
| | ANOVA + Permutation | 0.961 | 0.8 | 0.84 | 0.734 | 0.84 |
| | ANOVA + XGB_feature_selection | **0.96** | **0.8** | **0.85** | **0.74** | **0.84** |
| Feature set 4 | ANOVA | 0.95 | 0.78 | 0.84 | 0.75 | 0.82 |
| | ANOVA + Permutation | 0.96 | 0.79 | 0.84 | 0.74 | 0.82 |
| | ANOVA + XGB_feature_selection | **0.96** | **0.79** | **0.87** | **0.76** | **0.84** |
| Feature set 5 | ANOVA | 0.96 | 0.77 | 0.82 | 0.73 | 0.83 |
| | ANOVA + Permutation | 0.96 | 0.79 | 0.85 | 0.74 | 0.82 |
| | ANOVA + XGB_feature_selection | **0.96** | **0.79** | **0.85** | **0.77** | **0.85** |

Table 3: Performance Comparison of Different Feature Selection Methods

the models and simplifying computational processes. The decrease in overfitting risk, facilitated by dimension reduction, ensured the model's accuracy.

| Features | Dimension | Reduced dimension |
|---|---|---|
| Secondary structure | 3 | 3 |
| Autocorrelation | 240 | 136 |
| CTD | 148 | 38 |
| KSAP | 3600 | 109 |
| DDE | 400 | 108 |
| Conjoint Triad | 343 | 137 |

Table 4: Dimension reduction of features

## 4.2 Oversampling

Previous studies stated that imbalanced datasets could lead to biased models since the algorithms favor the class with more instances.[38] In this work, imbalanced data can lead to suboptimal model performance by favoring the majority class while struggling to identify the minority class. Our oversampling techniques increased the number of instances in the minority class to match the majority class, ensuring balance among the different classes.[39] There are various methods to perform oversampling. After careful evaluation, the SMOTE (Synthetic Minority Oversampling Technique) method was considered best for this investigation.[35]

A comparative study was conducted between models trained on oversampled data and those trained without sampling. The aim was to investigate the influence of oversampling on the model's generalization ability and its effectiveness in predicting both classes. The analysis output (supplementary file 5) demonstrated the superiority of oversampled datasets. Models trained on oversampled data consistently outperformed those trained without sampling for all five feature sets considered. The discriminating performance underlines the benefits of employing oversampling techniques, particularly in the presence of imbalanced classes, as it leads to improving our model accuracy.

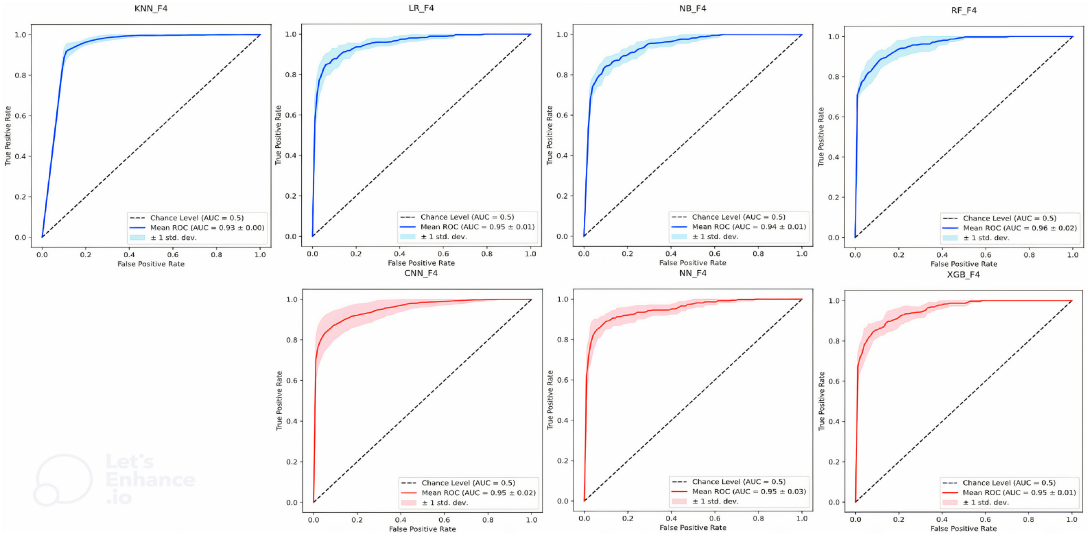


Figure 2: Performance analysis of ROC curves for each model on feature set 4

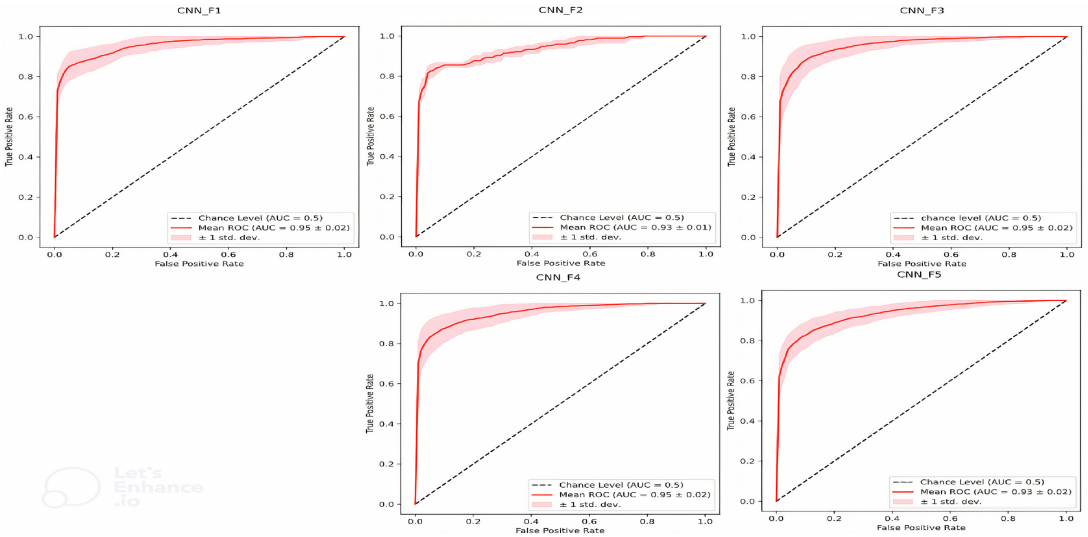


Figure 3: CNN model performance comparison on five feature sets

| Feature set | Method | Acc | F1 | ROC | MCC | PR |
|---|---|---|---|---|---|---|
| Feature set 1 | CNN | 0.95 | 0.77 | 0.95 | 0.75 | 0.85 |
| | RF | 0.93 | 0.7 | 0.95 | 0.67 | 0.83 |
| | NN | 0.94 | 0.76 | 0.9 | 0.74 | 0.82 |
| | LR | 0.93 | 0.72 | 0.93 | 0.68 | 0.74 |
| | NB | 0.93 | 0.7 | 0.93 | 0.66 | 0.76 |
| | KNN | 0.93 | 0.72 | 0.94 | 0.69 | 0.82 |
| | XGB | 0.94 | 0.74 | 0.95 | 0.71 | 0.83 |
| Feature set 2 | CNN | 0.94 | 0.75 | 0.93 | 0.73 | 0.83 |
| | RF | 0.95 | 0.77 | 0.96 | 0.74 | 0.86 |
| | NN | 0.85 | 0.68 | 0.86 | 0.64 | 0.82 |
| | LR | 0.92 | 0.62 | 0.92 | 0.67 | 0.66 |
| | NB | 0.92 | 0.68 | 0.93 | 0.65 | 0.8 |
| | KNN | 0.92 | 0.7 | 0.93 | 0.67 | 0.79 |
| | XGB | 0.93 | 0.71 | 0.96 | 0.68 | 0.82 |
| Feature set 3 | CNN | 0.95 | 0.75 | 0.95 | 0.72 | 0.84 |
| | RF | 0.91 | 0.67 | 0.94 | 0.64 | 0.81 |
| | NN | 0.94 | 0.75 | 0.88 | 0.72 | 0.83 |
| | LR | 0.92 | 0.68 | 0.93 | 0.64 | 0.7 |
| | NB | 0.92 | 0.69 | 0.94 | 0.65 | 0.76 |
| | KNN | 0.92 | 0.71 | 0.92 | 0.69 | 0.8 |
| | XGB | 0.94 | 0.74 | 0.96 | 0.71 | 0.83 |
| Feature set 4 | CNN | **0.96** | **0.80** | **0.96** | **0.77** | **0.86** |
| | RF | 0.95 | 0.77 | 0.96 | 0.74 | 0.86 |
| | NN | 0.94 | 0.76 | 0.87 | 0.74 | 0.84 |
| | LR | 0.94 | 0.75 | 0.95 | 0.72 | 0.78 |
| | NB | 0.93 | 0.71 | 0.94 | 0.68 | 0.75 |
| | KNN | 0.81 | 0.52 | 0.93 | 0.51 | 0.73 |
| | XGB | 0.95 | 0.76 | 0.95 | 0.73 | 0.86 |
| Feature set 5 | CNN | 0.62 | 0.37 | 0.92 | 0.34 | 0.78 |
| | RF | 0.95 | 0.74 | 0.96 | 0.74 | 0.86 |
| | NN | 0.94 | 0.74 | 0.86 | 0.71 | 0.81 |
| | LR | 0.95 | 0.78 | 0.94 | 0.76 | 0.82 |
| | NB | 0.89 | 0.57 | 0.9 | 0.52 | 0.67 |
| | KNN | 0.95 | 0.72 | 0.93 | 0.7 | 0.83 |
| | XGB | 0.95 | 0.76 | 0.96 | 0.73 | 0.85 |

Table 5: Performance analysis of five feature set combinations using five methods

## 4.3 Model performance analysis

In this study, we evaluated the performance of seven models-Convolutional Neural Network (CNN), Random Forest Classifier, Naive Bayes Classifier, K-Nearest Neighbor, Logistic Regression, XGBoost Classifier, and a Basic Neural Network—on five feature sets, designated as F1, F2, F3, F4, and F5, to determine which model produces the best results.

Through a detailed analysis of performance metrics across all model and feature combinations, we concluded that feature set 4, which includes Secondary Structure(SS), Conjoint Triad(CT/CJT), DDE, Autocorrelation(AutoCorr), and K-spaced Amino Acid

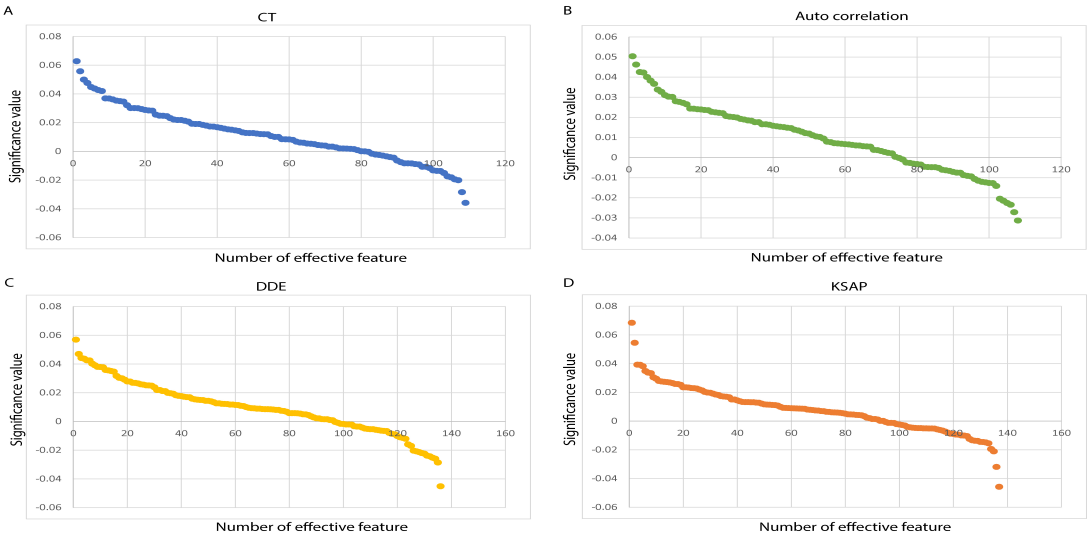


Figure 4: A. Importance of top 137 features of CT on the model performance B. Importance of the 136 features of Auto correlations and their impact on model performance C. Assessment of the 108 features of DDE and their influence on model performance D. Impact of the 109 features of K-spaced amino acid pairs on model performance.

Pair(KSAP), performed as the most critical set of features for classifying plastic-degrading enzymes(Table 5). The Convolutional Neural Network (CNN) model trained on feature set 4 demonstrated superior performance compared to other model-feature combinations. Figure 3 shows the performance of CNN models trained on five feature sets consecutively.

While accuracy only considers correctly categorized tuples unaffected by their classes (plastic-degradable/non-plastic-degradable), we place greater weight on F1 due to its balance of precision and recall. The results, as shown in Table 5, indicate that CNN trained on feature set 4 consistently outperforms the other variations of the model and feature sets. While some models achieved F1 scores ranging from 57% to 77%, CNN surpassed them with a score of 80%. In conclusion, our investigation demonstrated that CNN outperforms XGBoost and other models in classifying plastic-degrading enzymes, showcasing the power of deep learning techniques in handling complex biological data.

The primary objective of this research was to accurately identify plastic-degrading enzymes (PDEs) from protein sequences using an in silico approach. A key finding of our study is identifying critical features that significantly contribute to plastic degradation. By employing a novel combination of ANOVA and XGBoost (XGB) feature selection, we pinpointed four main feature categories that impact model performance: K-spaced amino acid pairs (KSP), Dipeptide Deviation from Expected (DDE), Autocorrelation, and Composition, Transition, Distribution (CTD). Through our analysis, Autocorrelation emerged as the most influential feature set. Model performance dropped significantly when Autocorrelation was excluded from the training process, as indicated in Supplementary File S4. Following this, we found KSP, CTD, and DDE features necessary. To further assess feature importance, we categorized them into two groups based on their contribution to the F1 score: positive values, which caused a decrease in the F1 score when removed, and negative values, where feature removal resulted in a slight increase in the F1 score. The graph in Supplementary File S4 illustrates these positive and negative values. While features with positive values were deemed crucial for distinguishing plastic-degrading enzymes, those with negative values—despite slightly enhancing the score—were minimal

in effect and thus excluded from further analysis. Future work will focus on several key areas to improve model performance further. These include expanding the dataset with more diverse protein sequences, exploring deep learning models such as Transformers for better feature extraction, and addressing class imbalance with more advanced oversampling techniques. Additionally, incorporating domain knowledge on protein structure and function may enhance the biological interpretability of the model's predictions.

# 5 Conclusion

In conclusion, the growing threat of plastic waste to the environment calls for innovative solutions, and enzymatic degradation offers a promising avenue for mitigating this challenge. While traditional methods for identifying plastic-degrading enzymes (PDEs) are resource-intensive and limited by non-culturable microorganisms, this study introduces a powerful in-silico approach to accelerate the discovery of PDEs. By employing a novel feature extraction strategy combined with advanced machine learning techniques, particularly a high-performing Convolutional Neural Network, the proposed system demonstrated exceptional accuracy in predicting PDEs. These findings highlight the system's potential to facilitate the identification of novel enzymes, contributing to the development of sustainable strategies for managing plastic pollution.

# Author's Contributions

**Soharth Hasnat**: Formal analysis, Conceptualization, Methodology, Software, Visualization, Writing – review & editing.
**Fariha Anjum Shifa**: Formal analysis, Methodology, Software, Visualization, Writing – review & editing.
**Shabab Murshed**: Methodology, Software, Visualization, Writing – review & editing.
**Sarker Tanveer Ahmed Rumee**: Conceptualization, Supervision, Validation, Writing – review & editing.
**MST Murshida Mahbub**: Conceptualization, Supervision, Writing – review & editing.

# Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

# Author's approval

All authors read and give consent for publication.

# Competing interests

The authors have declared that no competing interests exist.

# Supplementary Information

All supplementary are available at `https://doi.org/10.6084/m9.figshare.27263676`

# References


[1] AJPAS A, A feature selection based on one-way-anova for microarray data classification, *AJPAS JOURNAL* **3**:1–6, 2016.

[2] Azevedo-Santos VM, Brito MF, Manoel PS, Perroca JF, Rodrigues-Filho JL, Paschoal LR, Gonçalves GR, Wolf MR, Blettler MC, Andrade MC, *et al.*, Plastic pollution: A focus on freshwater biodiversity, *Ambio* **50**(7):1313–1324, 2021.

[3] Ghatge S, Yang Y, Ahn JH, Hur HG, Biodegradation of polyethylene: a brief review, *Applied Biological Chemistry* **63**:1–14, 2020.

[4] Jambeck JR, Geyer R, Wilcox C, Siegler TR, Perryman M, Andrady A, Narayan R, Law KL, Plastic waste inputs from land into the ocean, *Science* **347**(6223):768–771, 2015.

[5] Gambarini V, Pantos O, Kingsbury JM, Weaver L, Handley KM, Lear G, Plasticdb: a database of microorganisms and proteins linked to plastic biodegradation, *Database* **2022**:baac008, 2022.

[6] Gan Z, Zhang H, Pmbd: a comprehensive plastics microbial biodegradation database, *Database* **2019**:baz119, 2019.

[7] Ho Thanh Lam L, Le NH, Van Tuan L, Tran Ban H, Nguyen Khanh Hung T, Nguyen NTK, Huu Dang L, Le NQK, Machine learning model for identifying antioxidant proteins using features calculated from primary sequences, *Biology* **9**(10):325, 2020.

[8] Huang YA, You ZH, Chen X, Chan K, Luo X, Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding, *BMC bioinformatics* **17**:1–11, 2016.

[9] Zhang L, Zhang C, Gao R, Yang R, Song Q, Sequence based prediction of antioxidant proteins using a classifier selection strategy, *Plos one* **11**(9):e0163274, 2016.

[10] Sasaki Y, *et al.*, The truth of the f-measure, *Teach tutor mater* **1**(5):1–5, 2007.

[11] Scikit-learn, *roc curve — scikit-learn 1.5.2 documentation.*

[12] Consortium U, Uniprot: a hub for protein information, *Nucleic acids research* **43**(D1):D204–D212, 2015.

[13] Zahiri Z, Mehrshad N, Mehrshad M, Df-phos: Prediction of protein phosphorylation sites by deep forest, *The Journal of Biochemistry* p. mvad116, 2023.

[14] Hou R, Wu J, Xu L, Zou Q, Wu YJ, Computational prediction of protein arginine methylation based on composition–transition–distribution features, *ACS omega* **5**(42):27470–27479, 2020.

[15] Tung CW, Prediction of pupylation sites using the composition of k-spaced amino acid pairs, *Journal of theoretical biology* **336**:11–17, 2013.

[16] Wang J, Zhang L, Jia L, Ren Y, Yu G, Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences, *International journal of molecular sciences* **18**(11):2373, 2017.

[17] Yuan SS, Gao D, Xie XQ, Ma CY, Su W, Zhang ZY, Zheng Y, Ding H, Ibpred: A sequence-based predictor for identifying ion binding protein in phage, *Computational and Structural Biotechnology Journal* **20**:4942–4951, 2022.

[18] Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, *et al.*, Biopython: freely available python tools for computational molecular biology and bioinformatics, *Bioinformatics* **25**(11):1422, 2009.

[19] Horne DS, Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities, *Biopolymers: Original Research on Biomolecules* **27**(3):451–477, 1988.

[20] Wang H, Hu X, Accurate prediction of nuclear receptors with conjoint triad feature, *BMC bioinformatics* **16**:1–13, 2015.

[21] Wang C, Wang Y, Wang Y, Wang Z, Zhang L, Liang Y, Chen L, Zou S, Dong H, The adp-ribosylation factor-like small gtpase fgarl1 participates in growth, pathogenicity and don production in fusarium graminearum, *Fungal biology* **124**(11):969–980, 2020.

[22] Dhanda SK, Vir P, Raghava GP, Designing of interferon-gamma inducing mhc class-ii binders, *Biology direct* **8**:1–15, 2013.

[23] Tyagi A, Kapoor P, Kumar R, Chaudhary K, Gautam A, Raghava G, In silico models for designing and discovering novel anticancer peptides, *Scientific reports* **3**(1):2984, 2013.

[24] Eisenberg D, The discovery of the $\alpha$-helix and $\beta$-sheet, the principal structural features of proteins, *Proceedings of the National Academy of Sciences* **100**(20):11207–11210, 2003.

[25] Dhanya R, Paul IR, Akula SS, Sivakumar M, Nair JJ, F-test feature selection in stacking ensemble model for breast cancer prediction, *Procedia Computer Science* **171**:1561–1570, 2020.

[26] Elssied NOF, Ibrahim O, Osman AH, A novel feature selection based on one-way anova f-test for e-mail spam classification, *Research Journal of Applied Sciences, Engineering and Technology* **7**(3):625–638, 2014.

[27] Hastie T, Tibshirani R, Friedman JH, Friedman JH, *The elements of statistical learning: data mining, inference, and prediction*, Springer, 2009.

[28] Demir S, Sahin EK, An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using adaboost, gradient boosting, and xgboost, *Neural Computing and Applications* **35**(4):3173–3190, 2023.

[29] Qu Y, Lin Z, Li H, Zhang X, Feature recognition of urban road traffic accidents based on ga-xgboost in the context of big data, *IEEE Access* **7**:170106–170115, 2019.

[30] Adler AI, Painsky A, Feature importance in gradient boosting trees with cross-validation feature selection, *Entropy* **24**(5):687, 2022.

[31] Bjorck N, Gomes CP, Selman B, Weinberger KQ, Understanding batch normalization, *Advances in neural information processing systems* **31**, 2018.

[32] De Rosa GH, Papa JP, Yang XS, Handling dropout probability estimation in convolution neural networks using meta-heuristics, *Soft Computing* **22**:6147–6156, 2018.

[33] Santos CFGD, Papa JP, Avoiding overfitting: A survey on regularization methods for convolutional neural networks, *ACM Computing Surveys (CSUR)* **54**(10s):1–25, 2022.

[34] Narkhede MV, Bartakke PP, Sutaone MS, A review on weight initialization strategies for neural networks, *Artificial intelligence review* **55**(1):291–322, 2022.

[35] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP, Smote: synthetic minority oversampling technique, *Journal of artificial intelligence research* **16**:321–357, 2002.

[36] Jiang R, Shang L, Wang R, Wang D, Wei N, Machine learning based prediction of enzymatic degradation of plastics using encoded protein sequence and effective feature representation, *Environmental Science & Technology Letters* **10**(7):557–564, 2023.

[37] Chen T, Guestrin C, Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[38] Spelmen VS, Porkodi R, A review on handling imbalanced data, *2018 international conference on current trends towards converging technologies (ICCTCT)*, IEEE, pp. 1–11, 2018.

[39] Chen C, Shi H, Jiang Z, Salhi A, Chen R, Cui X, Yu B, Dnn-dtis: Improved drug-target interactions prediction using xgboost feature selection and deep neural network, *Computers in Biology and Medicine* **136**:104676, 2021.