7-20-2023

Evaluation of Machine Learning Methods for Multivariate Classification with Application to Environmental Datasets

Xianqiang Fu

EVALUATION OF MACHINE LEARNING METHODS FOR MULTIVARIATE CLASSIFICATION WITH APPLICATION TO ENVIRONMENTAL DATASETS

by

Xianqiang Fu

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Biostatistics

The University of Memphis

August 2023

Acknowledgments

I am sincerely grateful to my advisory committee, whose combined expertise, guidance, and unwavering support have made this work possible.

To Dr. Joyce Jiang, my advisor, whose insightful perspectives and steady guidance have been pivotal to my journey. Our bi-weekly meetings have not only served as waypoints on the path of this thesis but have also helped shape my broader view of our topic. Your ability to demystify complex concepts and provide practical programming advice is something I deeply appreciate.

To Dr. Hongmei Zhang, who was always ready with support and advice when I needed it. Your professional expertise and encouragement have been critical to my master's journey. Your guidance has lit the path for me in moments of uncertainty, for which I am truly grateful.

Lastly, to Dr. Chunrong Jia, whose grant has allowed me to embark on this educational journey in Biostatistics. Your support and guidance from an environmental perspective have been invaluable in broadening my understanding and providing me with a holistic view of our work.

I want to acknowledge each of them for their mentorship, which has been a guiding light throughout this master's program, leading this academic work and shaping my path as a potential biostatistician. Thank you all.

Abstract

As environmental data grows in complexity, machine learning presents an avenue to extract meaningful insights from such data. This study aimed to investigate the applicability and performance of various machine learning methods for multi-class classification problems, with a specific focus on complex environmental data, including Polycyclic Aromatic Hydrocarbons (PAHs). In the current study, we evaluated ten machine learning models to assess their performance in multivariate classification problems using simulation studies. The results showed that Regularized Multinomial Logistic Regression (RMLR) has higher classification accuracy when the independent variables are independent, while the Gradient Boosting Machine (GBM) outperformed others when the independent variables are highly correlated. Furthermore, the feature selection accuracy of three different methods was also evaluated. GBM and Random Forest (RF) showed a higher sensitivity compared to other methods across different data settings. Based on these findings, it appears that linear models such as RMLR and MLR may not achieve optimal performance when confronted with highly correlated independent variables. Instead, tree-based methods, such as GBM and RF, prove to be a better choice. Overall, it is crucial to choose the appropriate machine learning methods based on the complexity of environmental data and the specific requirements of the task.

Table of Contents

List of Tables

List of Figures

1. Introduction

Exposure to multiple environmental chemicals is an increasingly concerning issue in today's world, as various sources, including industrial processes and human activities, contribute to the widespread contamination of breathing air, drinking water, and food with an array of pollutants (Ramlogan 1997, Kim, Jahan et al. 2013, Schweitzer and Noblet 2018, Manisalidis, Stavropoulou et al. 2020). Addressing this concern requires understanding the combined impact of these pollutants on human health, which has become crucial considering the growing availability of environmental data. However, the state of the science on simultaneous exposure to multiple pollutants remains in its early stages, with gaps and challenges in environmental data analyses (Oskar and Stingone 2020, Zhong, Zhang et al. 2021, Maitre, Guimbaud et al. 2022).

Addressing these challenges has increased the importance of multi-class classification problems in both machine learning and environmental communities (Hao, Fan et al. 2022, Liu, Lu et al. 2022). A range of machine learning methods, including supervised and unsupervised learning techniques, have been developed and applied to analyze environmental datasets (Tahmasebi, Kamrava et al. 2020, Zhong, Zhang et al. 2021). However, while the number of machine-learning algorithms continues to grow, relatively few studies offer a comprehensive comparison of different learners, as model comparison studies are often limited to only a few models. Applying machine learning techniques to environmental data presents unique challenges, such as dealing with large volumes, high dimensionality, noise, and missing data. These challenges often arise from the complex nature of environmental data and the various sources of data collection (Hino, Benami et al. 2018, Liu, Lu et al. 2022).

In this study, our objective is to address the research gaps and challenges associated with analyzing exposure profiles by applying innovative data science methodologies to environmental

datasets. Through the integration of relationships among environmental chemicals within a machine learning framework, we will investigate the potential of various machine learning methods for environmental data analysis. To assess the performance of these machine learning methods, we will apply them to real-world environmental datasets and utilize simulated data that closely resemble actual environmental data.

This research will contribute to the growing knowledge in applying machine learning methods to environmental data analysis, addressing the unique challenges inherent in these complex datasets, and providing valuable insights into the sources, distribution, and impacts of environmental pollutants.

2. Literature Review

The growing importance of environmental data analysis in fields such as climate science, meteorology, hydrology, and ecology has led to an increased interest in machine learning methods to gain insights, predict future trends, and facilitate decision-making. Supervised learning techniques, in particular, have demonstrated considerable potential in handling complex environmental datasets with known input features and desired outputs.

The literature on supervised learning methods, which include linear regression, support vector machines, decision trees, and random forests, has shown that these techniques can achieve high accuracy in prediction tasks when sample labeled data is available. However, the model's quality is dependent on the quality of the labeled data, and labeling errors can result in poor performance (Sathya and Abraham. 2013, Sunori, Bhakuni Negi et al. 2021). In environmental data analysis, supervised learning techniques have been employed for both regression tasks involving

continuous outcomes and classification tasks involving categorical outcomes (Reichstein, Camps-Valls et al. 2019, Liu, Lu et al. 2022).

Multi-class classification tasks, which involve assigning one of more than two class labels to an instance, have been tackled using methods such as One-vs-One and One-vs-Rest strategies, decision trees, random forests, neural networks, or ensemble methods like boosting and bagging (Freund and Schapire 1997, Bishop and Nasrabadi 2006, Ştefan 2012, LeCun, Bengio et al. 2015). These methods have been applied to various environmental studies, with some focusing specifically on PAHs analysis, demonstrating the effectiveness of machine learning methods in identifying and quantifying specific pollutants in environmental samples (Zhao, Wang et al. 2019, Bajomo, Ju et al. 2022).

This literature review highlights the increasing adoption of supervised learning methods in environmental data analysis, and their ability to identify patterns and better understand the sources and impacts of these pollutants. By employing classification technology and feature selection algorithms, researchers can capture a snapshot of ambient PAHs in both temporal and spatial dimensions, reduce data dimensionality, and identify the most crucial features for characterizing the environment. However, future studies should continue to evaluate the performance of supervised learning methods using simulated data and real-world environmental datasets to ensure the selection of the most suitable method for analyzing environmental data.

3. Machine Learning Methods

In this study, we will employ a range of machine learning algorithms that have been widely used in the field, including Linear Discriminant Analysis (LDA), Multinomial Logistic Regression (MLR), Regularized Multinomial Logistic Regression (RMLR), Support Vector Machine (SVM), Naïve Bayes Classifier (NBC), Decision Tree (DT), Random Forest (RF), K Nearest Neighbor (KNN), Gradient Boosting Machine (GBM), and Convolutional Neural Networks (CNN). These algorithms were chosen due to their maturity and effectiveness in various applications.

Linear Discriminant Analysis (LDA)

LDA, a popular dimensionality reduction technique, is commonly used for supervised learning in pattern recognition and machine learning. It aims to project data onto a lower-dimensional subspace while maximizing the separation between different classes. However, LDA has limitations, such as assuming linear separability, sensitivity to outliers, limited applicability to high-dimensional data, class imbalance, and non-normality.

Performing LDA is a three-step process:

Step 1: Calculate the between-class variance.

$$S_b = \sum_{i=1}^{g} N_i(\bar{x}_i - \bar{x})\,(\bar{x}_i - \bar{x})^T \qquad (1)$$

S_b: $between - class\ scatter\ matrix$;

N_i: $sample\ size\ of\ class\ i$;

$\bar{x}_i$: $sample\ mean\ of\ class\ i$;

$\bar{x}$: $overall\ mean$;

Step 2: Compute the within-class variance.

$$S_w = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (\bar{x}_{i,j} - \bar{x})\,(\bar{x}_{i,j} - \bar{x})^T$$

(2)

S_w: $within - class\ scatter\ matrix$;

N_i: $sample\ size\ of\ class\ i$;

$\bar{x}_{i,j}$: $sample\ mean\ of\ class\ i\ and\ j$;

$\bar{x}$: $overall\ mean$;

Step 3: Construct a lower-dimensional space that maximized S_b and minimized S_w.

$$SS = S_w^{-1}\, S_b$$

(3)

SS: $discriminant\ matrix$

Multinomial Logistic Regression (MLR)

MLR is a classification method that extends binary logistic regression to solve multi-class problems. Initially called a discrete choice model in finance and economics, MLR is now widely used in machine learning and predictive modeling to handle multi-class dependent variables. MLR faces several challenges when applied to environmental data analysis. These include the need for large sample sizes, which can be challenging to obtain due to logistical constraints, and the presence of multicollinearity, which makes it challenging to identify the unique contribution of each predictor variable. Additionally, spatial autocorrelation violates the independence assumption of MLR, and non-linear relationships between predictors and outcomes can limit the model's accuracy. Finally, imbalanced classes can lead to biased predictions, as the model may prioritize more common outcomes over rare ones.

In MLR, the dependent variable follows a multinomial distribution, which is a generalization of the binomial distribution for variables with more than two categories.

$$\mathcal{P}(\mathcal{X}_1, \dots, \mathcal{X}_\mathrm{k}) = \binom{n}{\mathcal{X}_1 \dots \mathcal{X}_\mathrm{k}} \mathcal{P}_1{}^{\mathcal{X}_1} \dots \mathcal{P}_\mathrm{k}{}^{\mathcal{X}_\mathrm{k}}$$

(4)

if $\mathcal{X}_j$=0, 1, ..., n; $\sum_{j=1}^{\mathrm{k}} \mathcal{X}_j = n$, $\sum_{j=1}^{\mathrm{k}} \mathcal{P}_j = 1$,

Denoted by X ~ M (n, P)

Suppose we have a dependent variable Y with K categories and a set of p independent variables, the logistic regression for multi-class, which is given by,

$$P(\mathcal{Y}_j = K | \mathcal{X}_i) = 1 - \sum_{k=1}^{\mathrm{K}-1} \mathrm{P}(\mathcal{Y}_i = k | \mathcal{X}_i)$$

(5)

The general form of the probability is,

$$P(\mathcal{Y}_j = k | \mathcal{X}_i) = \frac{\exp\,(\theta_i^T \mathcal{X}_i)}{\sum_{i=1}^{K} \exp\,(\theta_i^T \mathcal{X}_i)}$$

(6)

As the K-th class is your reference $\theta_K = (0, \dots, 0)^T$ and therefore

$$\sum_{i=1}^{K} \exp\,(\theta_i^T \mathcal{X}_i) = \exp(0) + \sum_{i=1}^{K-1} \exp\,(\theta_i^T \mathcal{X}_i) = 1 + \sum_{i=1}^{K-1} \exp\,(\theta_i^T \mathcal{X}_i)$$

(7)

Finally, for all $k < K$,

$$P(\mathcal{Y}_j = k | \mathcal{X}_i) = \frac{\exp\,(\theta_i^T \mathcal{X}_i)}{1 + \sum_{i=1}^{K-1} \exp\,(\theta_i^T \mathcal{X}_i)}$$

(8)

Where:

exp( ) : *is the exponential function*

y_j: *the dependent variable for the j th observation*

x_i: *the feature vector of the i th independent variable for the j th observation*

θ_i^T: *the tranpose of the column vector* θ_i, *resulting in a row vector*

This is the definition of the cost function J of theta for logistic regression.

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} log h_\theta\left(x^{(i)} + \left(1 - y^{(i)}\right) log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right)\right] \quad (9)$$

Where:

$J(\theta)$: *the cost function of MLR for a given set of parameters* θ

m: *the number of data points in the dataset*

$x^{(i)}$: *the feature vector for the i th data point*

$y^{(i)}$: *the actual value of the traget variable for the i th data point*

$h_\theta()$: *the logistic regression hypothesis function*

Regularized Multinomial Logistic Regression (RMLR)

RMLR is an extension of the multinomial logistic regression (MLR) that incorporates a regularization term to prevent overfitting and improve model generalization. RMLR minimizes a loss function incorporating a penalty term, either Lasso regularization (L1) or Ridge regularization (L2), to control model complexity and prevent overfitting. The L1 or L2 norm penalty encourages the model to select relevant predictor variables and reduce the magnitude of the remaining coefficients. The algorithm's performance depends on the choice of the regularization parameter, with incorrect values potentially leading to overfitting or underfitting. Finding the optimal regularization parameter can be time-consuming and requires extensive experimentation and validation.

The cost function J(θ) for RMLR can be defined as follows:

$$J\left(\theta\right) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} log h_\theta\left(x^{(i)} + \left(1 - y^{(i)}\right) log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2 \qquad (10)$$

Where λ is the regularization parameter, which controls the trade-off between minimizing the cost function and minimizing the magnitude of the coefficients. A large λ leads to higher regularization and smaller coefficients. To find the connection between the MLR and RMLR, note that when λ = 0, the regularization term vanishes, and the cost function J(θ) for RMLR reduces to the cost function for the MLR model. Thus, the MLR can be considered a particular case of the RMLR with no regularization (λ = 0).

Support Vector Machine (SVM)

SVMs are maximum-margin linear models of binary classification in linear models. SVMs can also extend to non-linear classification by projecting the original input space into a high-dimensional space (kernel trick). The basic idea behind SVMs is to find the hyperplane that maximally separates the data into different classes. In the two-class case, the hyperplane is defined as the decision boundary between the two classes.

SVMs were first introduced by Vladimir Vapnik in 1963 (Boser, Guyon et al. 1992), and since then, it has been widely used in various fields, such as computer vision, natural language processing, and bioinformatics. SVMs with the kernel function, which has been suggested by most of the researchers, has been proven to be the best classifier for most of the prediction systems with high accuracy. However, SVMs have limitations such as sensitivity to parameter tuning, limited interpretability, inability to handle imbalanced datasets, being computationally intensive, and susceptibility to noise. Proper parameter selection and preprocessing can help address some of these challenges, but it is essential to consider these limitations when applying SVMs to environmental data analysis.

In the context of classification, SVM aims to find the optimal hyperplane that separates the data points belonging to different classes with the maximum margin. The margin is the distance between the hyperplane and the closest data points from each class, called support vectors. The primary objective of SVM is to identify the optimal weight vector and bias term that maximize the margin between classes while accurately classifying the data points.

Naïve Bayes Classifier (NBC)

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, which is used to predict the probability of an observation belonging to a specific class, given a set of

predictor variables. Naive Bayes assumes that the predictor variables are independent of each other, given the class label, hence the name "naive".

One of the advantages of Naive Bayes is that it is computationally efficient and can handle high-dimensional data. It also has a low risk of overfitting, as it relies on strong assumptions about the independence of the predictor variables. However, its performance may be suboptimal when the independence assumption is violated, when classes are imbalanced, or when predictor variables are highly correlated. Despite these limitations, Naive Bayes is a simple and effective algorithm for various classification problems, particularly when predictor variables are mostly independent.

The Naive Bayes classifier works as follows:

1. Calculate the prior probability of each class, which is the proportion of the training data that belongs to each class.
2. For each predictor variable, calculate the probability distribution of the variable given each class. This is often done by assuming a probability distribution, such as Gaussian or Bernoulli, and estimating the distribution parameters from the training data.
3. For a new observation, calculate the posterior probability of each class given the observed predictor variables using Bayes' theorem. The class with the highest posterior probability is then assigned to the new observation.

Decision Tree (DT)

DTs are a widely utilized machine learning algorithm applicable to classification and regression tasks. They employ a tree-like structure that depicts a sequence of decisions and their potential outcomes. The simplicity and ease of understanding of decision trees make them popular in various fields, including finance, medicine, and marketing. Rather than being

represented by a specific mathematical formula, the DT algorithm is primarily algorithmic and heuristic. The algorithm starts with a root node that represents the entire dataset and then proceeds to split the data into smaller subsets based on different feature values. This process is executed recursively until a stopping criterion, such as a minimum number of samples per leaf node or a maximum tree depth, is reached.

DTs offer several key advantages, including handling numerical and categorical features and accommodating non-linear relationships between features and target variables. They are also robust against outliers and capable of managing missing data. However, DTs can be susceptible to overfitting, which occurs when the tree becomes overly complex, modeling the data's noise rather than the underlying relationships. This can result in poor performance on new, unseen data.

The algorithm is shown in the following diagram,

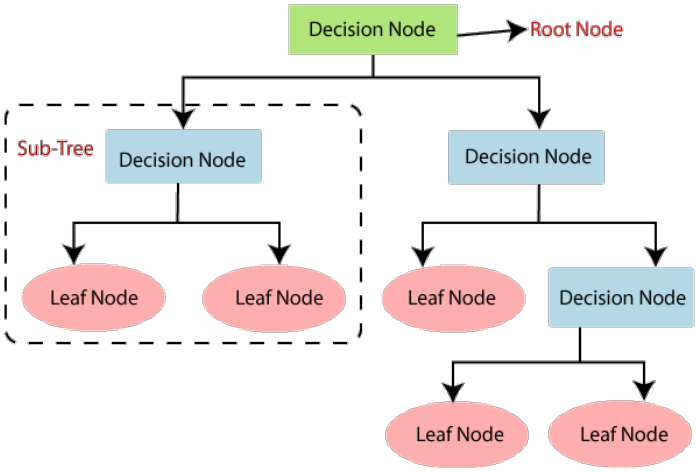

Figure 1. Flow diagram of the decision tree model.

Random Forest (RF)

RF is a widely utilized ensemble machine learning algorithm for classification and regression tasks. Aggregating the predictions of numerous decision trees generates more accurate and stable outcomes. A key advantage of the random forest algorithm is its ability to manage many features while remaining less susceptible to overfitting than individual decision trees. Additionally, it offers feature importance metrics, which can help identify the most significant features within the dataset. One drawback of random forests is that they can be computationally demanding and may require longer durations to generate predictions. To strike a balance between accuracy and computational cost, it is crucial to carefully choose the number of trees in the forest and the size of the random feature subsets utilized in each tree.

The random forest algorithm constructs multiple decision trees by randomly selecting subsets of features for each tree. These subsets are then used to split the data at each node. The final prediction is derived by averaging the predictions of all the trees within the forest. (See below)

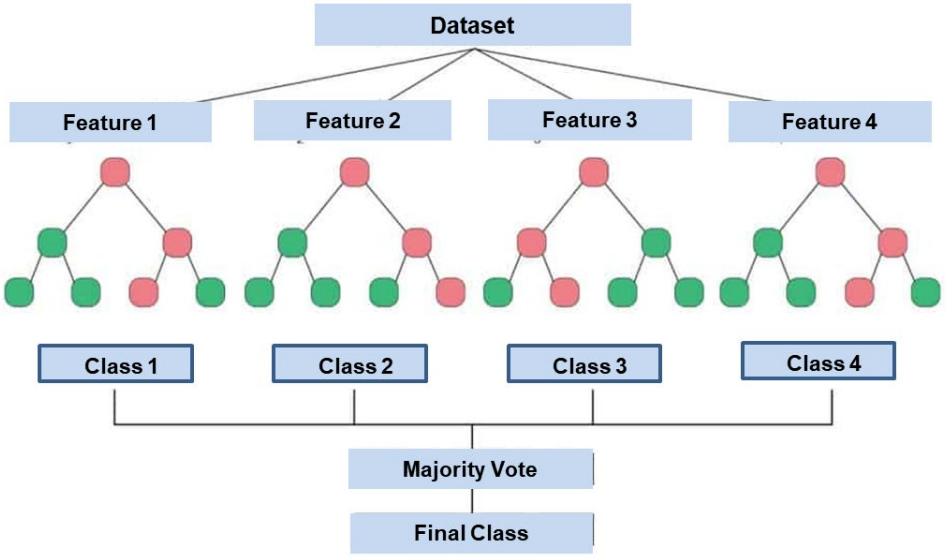

Figure 2. Simplified random forest model.

Gradient Boosting Machine (GBM)

Gradient Boosting is a powerful ensemble method for classification and regression problems that combines multiple weak learners to produce a strong prediction. It models complex, non-linear relationships between features and the target variable and is less prone to overfitting compared to individual decision trees. The algorithm iteratively focuses on areas with poor predictions, leading to improved performance. It also provides feature importance, which can be useful for understanding relationships and reducing dimensionality. However, gradient boosting can be computationally expensive and slow to make predictions. Careful choice of the number of weak learners and the learning rate is necessary to balance accuracy and computational cost.

The Gradient Boosting algorithm is a powerful ensemble learning method that builds on weak learners, typically decision trees, to iteratively improve the model's performance. The algorithm consists of the following four main steps:

1. Initialize the model with a constant prediction value or a simple model: The Gradient Boosting algorithm starts with an initial model, often a constant value (e.g., the mean of the target variable) or a simple model like a single decision tree. This initial model will be refined in the subsequent steps.
2. Compute the residuals: For each observation in the dataset, calculate the residual, which is the difference between the true value and the prediction made by the current model. These residuals represent the errors that the model needs to minimize in the next iterations.

3. Fit a weak learner on the residuals: Train a weak learner (usually a shallow decision tree) using the residuals as the target variable. This weak learner aims to model the errors made by the current model, thus helping to correct its predictions.
4. Update the model: Combine the predictions of the weak learner with the current model by assigning a weight to the weak learner's predictions and updating the model's predictions accordingly. The weight is typically determined using a learning rate parameter, which controls the contribution of the weak learner to the overall model.

These steps are repeated for a predetermined number of iterations or until a stopping criterion is met (e.g. when the improvement in model performance falls below a threshold). The final model is an ensemble of the initial model and all the weak learners, weighted by their respective learning rates.

K Nearest Neighbor (KNN)

KNN classification is a non-parametric, memory-based machine learning algorithm used for classification and regression problems. It assigns an observation to the majority class among its K nearest neighbors, with K being a user-defined parameter. KNN is simple, interpretable, and robust to noisy data, making it suitable for situations where interpretability is important.

However, KNN classification has some limitations. It can be computationally expensive when working with large datasets due to the need to calculate distances between all observations for each prediction. Additionally, KNN classification can be sensitive to the choice of K and the distance metric used, potentially resulting in a suboptimal performance with poor choices.

The equation is,

$$\varphi(x) = \underset{c \in \mathcal{Y}}{\arg max} \sum_{(\mathcal{X}i,\mathcal{Y}i)\in NN(x,\mathcal{L},k)} 1(\mathcal{Y}i = c)$$

(11)

NN$(x, \mathcal{L}, k)$ denotes the k nearest neighbors of x in $\mathcal{L}$.

Convolutional Neural Networks (CNN)

CNNs are a type of deep learning architecture designed explicitly for processing grid-like data, such as images. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers, which help extract hierarchical features from the input data. The convolutional layers use filters to detect local patterns, while pooling layers reduce the spatial dimensions, and the fully connected layers perform the final classification. CNNs have gained popularity due to their remarkable performance in image classification, object detection, and other computer vision tasks. Although primarily used for image processing, CNNs can be adapted for other data types, such as time series or structured environmental data, with appropriate modifications to the architecture.

The main components of a CNN include convolutional layers, pooling layers, and fully connected layers. The typical algorithm for a CNN is as follows:

1. Input: Begin with an input image or grid-like data, which is represented as a matrix of pixel values.
2. Convolutional Layer: Apply one or more filters (also called kernels) to the input data. Each filter is a small matrix that slides over the input data, performing element-wise multiplication and summing the results to create a new feature map. This process detects local patterns, such as edges and textures, in the input data.

3. Activation Function: Apply a non-linear activation function, such as the Rectified Linear Unit (ReLU), to the feature maps from the convolutional layer. This introduces non-linearity into the model, allowing it to learn more complex patterns.
4. Pooling Layer (Optional): Perform down-sampling by applying a pooling operation, such as max-pooling or average pooling, to the feature maps. This reduces the spatial dimensions of the feature maps, which helps to control overfitting and reduce computation.
5. Repeat Steps 2-4: Stack multiple convolutional and pooling layers in sequence, each detecting increasingly complex patterns in the input data.
6. Fully Connected Layer: Flatten the output of the final convolutional or pooling layer into a one-dimensional vector and pass it through one or more fully connected layers, which perform classification or regression tasks based on the features extracted by the convolutional layers.
7. Output Layer: Apply an activation function, such as softmax for multi-class classification, to produce the final output probabilities or values.
8. Loss Function and Optimization: Define a loss function, such as cross-entropy for classification or mean squared error for regression, to quantify the difference between the predicted outputs and the true labels. Use an optimization algorithm, such as stochastic gradient descent (SGD) or Adam, to minimize the loss function and update the model's weights during training.
9. Train the CNN: Iterate through the training data, forward propagating the input through the network, computing the loss, and updating the weights using backpropagation and the chosen optimization algorithm.

10. Evaluate and Fine-tune: Assess the performance of the CNN on a validation or test dataset, and fine-tune the model's hyperparameters, such as learning rate, batch size, and the number of layers or filters, to achieve the best performance.

Once trained and optimized, the CNN can be used for tasks such as image classification, object detection, or semantic segmentation.

4. Environmental Polycyclic Aromatic Hydrocarbons (PAHs) Data

This dataset contains concentrations (in ng/m^3) of 30 PAHs in the Memphis Tri-state Area (MTA) ambient air. Memphis is the central city of Mid-South USA and a transportation hub. Nineteen sites were selected in three neighboring counties in Tennessee, Mississippi, and Arkansas (Figure 1). These sites represented industrial, urban, suburban, and remote land-use types in MTA. At each site, total suspended particle (TSP) samples were collected using a high-volume sampler every 12 days from March 13th, 2018, to May 25th, 2019. The collection media consisted of a quartz fiber filter (QFF) and a glass thimble containing polyurethane foam (PUF), and styrene-divinylbenzene polymer resin sorbent (XAD-2) to collect PAHs from ambient air. Approximately 200 to 350 m^3 of ambient air was drawn over 24 hours. The dataset also has site descriptions, sampling information, and analytical performance. More methodological details and the original data are available in the data article.

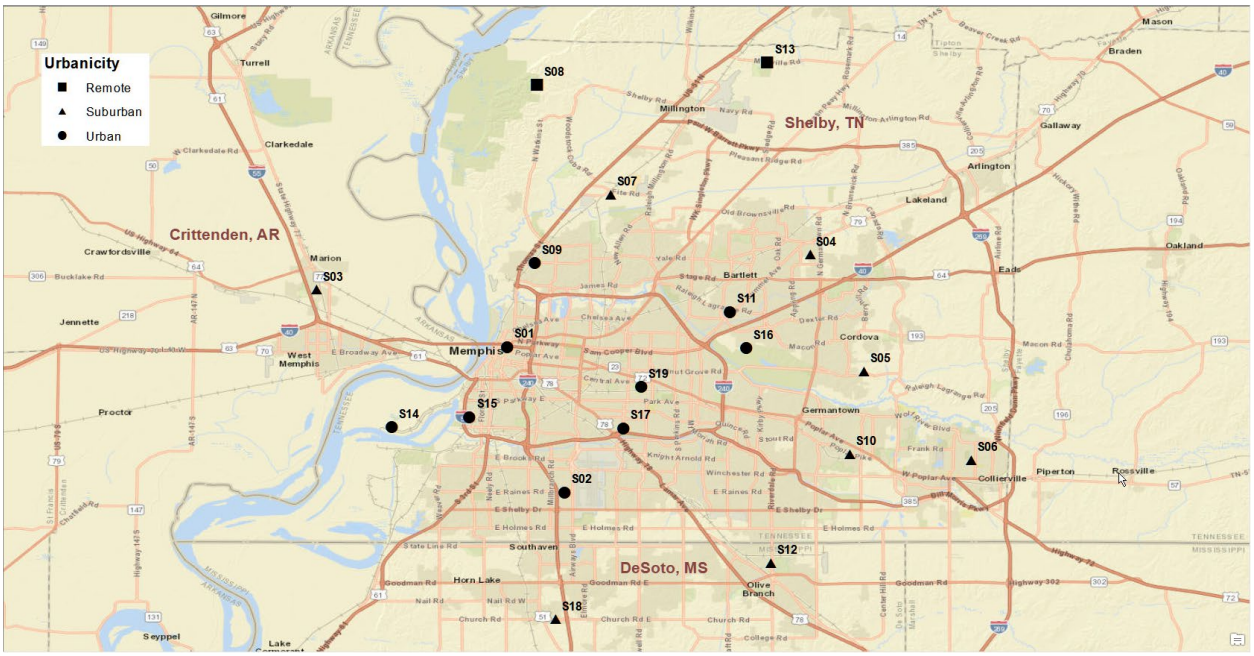

Figure 3. Monitoring sites of the PAH monitoring program in the MTA area.

This PAH dataset can be used to explore atmospheric chemistry and sources of PAHs, estimate population exposures to airborne PAHs and the associated health risks, and address environmental health disparities. Table 1 provides a descriptive summary of target PAHs, represented by their abbreviations, from 663 measurements. The data show that all analytes were somewhat detected, with the detection frequency (DF) varying between 4.7% to 100%. Naphthalene (NAP) had the highest mean value (26.9), the highest median (25.6), and the maximum value (992), suggesting it is the most prevalent analyte in this dataset. The standard deviation (SD) values, which indicate the spread of the measurements, also varied widely across the analytes, with NAP again having the largest SD (45.8), indicating considerable variability in its measurements. On the other end of the scale, analytes such as Dibenzo(a,l)pyrene (DIP), Coronene (Bajomo, Ju et al.), and Dibenzo(a,e)pyrene (DeP) were detected less frequently and had lower mean, median, and maximum values.

Table 1. Descriptive statistics of target PAHs measured in the Memphis PAHs Study.

Analytes	Abbr.	MTA (N=663)				
		DF (%)	Mean	SD	Median	Max
Naphthalene	NAP	100	26.9	45.8	25.6	992
Acenaphthylene	ACY	91	0.40	0.80	0.43	12.9
Acenaphthene	ACP	100	5.73	8.24	4.76	114
Fluorene	FLR	100	7.80	17.0	5.85	363
9-Fluorenone	FL9	100	2.18	2.63	1.64	22.4
Dibenzothiophene	DBT	100	1.64	2.36	1.34	15.4
Phenanthrene	PHE	98	12.9	17.2	13.5	100
Anthracene	ANT	100	4.15	6.74	1.17	53.9
Fluoranthene	FLT	100	3.44	5.03	3.40	29.3
Retene	RET	100	0.62	0.59	0.39	4.49
Pyrene	PYR	100	2.67	3.34	2.38	20.3
Benzo(c)phenanthrene	BcP	91	0.11	0.22	0.06	3.43
Cyclopenta(c,d)pyrene	CPP	78	0.40	0.65	0.03	2.46
Benz(a)anthracene	BaA	99	0.47	1.18	0.10	14.9
Chrysene	CHR	87	0.18	0.23	0.28	3.04
Benzo(b,j,k)fluoranthene	BbjkF	50	0.07	0.12	0.11	1.79
7,12-Dimethylbenz(a)anthracene	DMBA	20	0.02	0.02	0.01	0.28
Benzo(e)pyrene	BeP	73	0.13	0.12	0.16	1.40
Benzo(a)pyrene	BaP	29	0.04	0.07	0.05	1.01
Perylene	PER	12	0.01	0.01	0.01	0.19
3-Methylcholanthrene	MC3	18	0.02	0.04	0.01	0.45
Dibenz(a,h)acridine	DhACR	26	0.47	0.80	0.02	3.29
Dibenz(a,j)acridine	DjACR	4.7	0.02	0.02	0.02	0.24
Indeno[1,2,3-c,d]pyrene	IcP	16	0.04	0.05	0.04	0.52
Dibenz[a,h]anthracene	DhANT	6.9	0.02	0.02	0.01	0.22
Benzo(g,h,i)perylene	BgP	31	0.03	0.06	0.04	0.88
7H-Dienzo(c,g)carbazole	DBC	4.5	0.02	0.03	0.02	0.51
Dibenzo(a,l)pyrene	DIP	11	0.01	0.01	0.01	0.14
Coronene	COR	18	0.01	0.01	0.00	0.17
Dibenzo(a,e)pyrene	DeP	5.7	0.01	0.01	0.01	0.13
Σ30PAHs		**-**	**70.4**	**77.8**	**47.2**	**1,123**

Note: DF-detection frequency; SD - Standard deviation; Σ30PAHs – Sum concentration of the 30 target PAHs.

The heatmap (Figure 2) visualizes our dataset's correlation between different variables. In this map, red represents a positive correlation, and blue represents a negative. We notice that some

variables are highly correlated, as the intense red colors indicate. For example, the two variables circled with red frame might positively correlate because of the high color. In previous studies, those two variables, fluoranthene, and pyrene, are commonly used in diagnostic ratio techniques to infer the likely sources of PAHs.

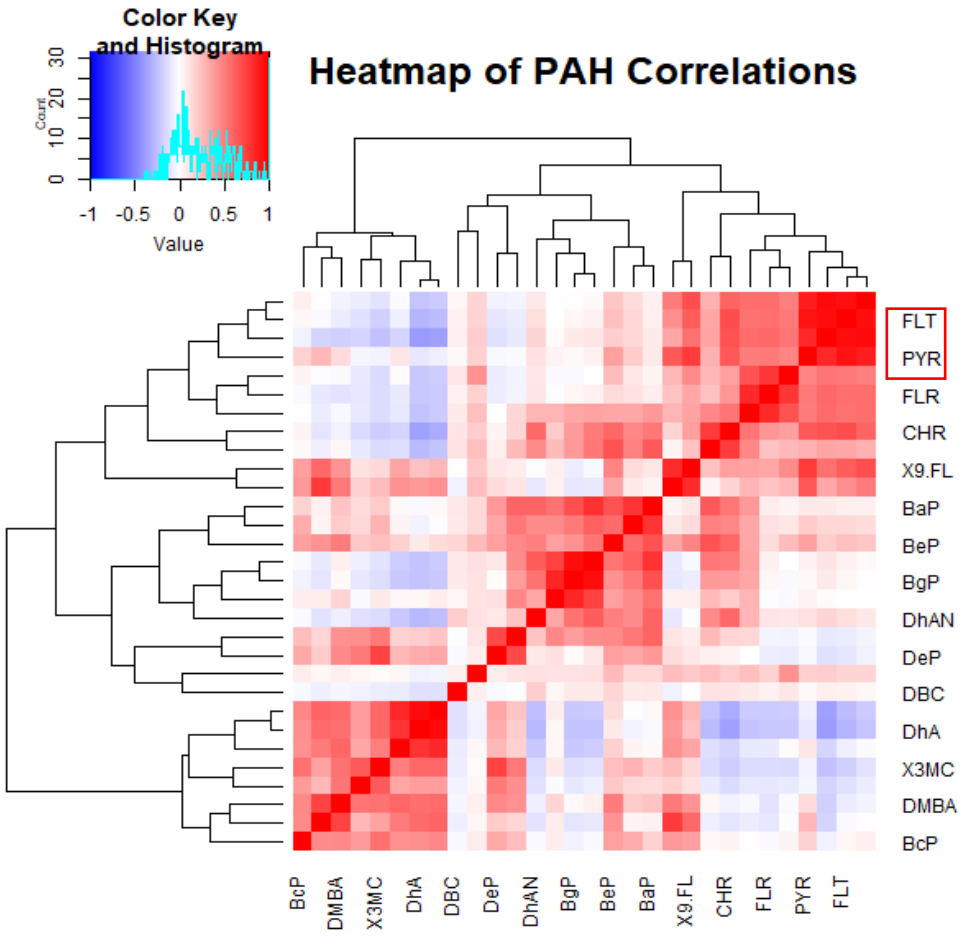

Figure 4. Heatmap visualization correlations among PAHs.

In summary, this dataset illustrates a broad range of PAHs in the environment, with varying degrees of prevalence and variability. Given the diverse prevalence and considerable variability across these compounds, this data set forms a robust basis for the study. It allows for examining different machine learning techniques' efficacy in multivariate classification, thereby contributing a valuable resource for our simulation studies.

5. Objectives

When studying the association between a multi-category outcome and an environmental data profile, it is essential to note that there are no established guidelines or recommendations regarding the choice of appropriate methods. To address this gap, the current study aims to evaluate the performance of different machine learning methods in multi-class classification using simulated data under different scenarios. We will focus on the evaluation of model classification accuracy and the sensitivity for feature selection. Furthermore, we plan to apply the selected machine learning methods to an environmental dataset to identify key Polycyclic Aromatic Hydrocarbon (PAH) features that represent the environmental characteristic. Specifically, we will focus on two aspects: the spatial aspect, represented by the site or geographical location, and the temporal aspect, represented by the season.

6. Simulation Study

A simulation study is a practical approach for evaluating and comparing the performance of various machine learning algorithms. In this study, two simulation strategies have been devised to assess the effectiveness of 10 machine learning techniques in handling a multi-class classification problem, utilizing five-fold cross-validation and examining the feature selection capabilities of 3 selected methods.

The simulation settings encompass several key components. Firstly, the sample size is set at 600 observations with 30 independent variables in the dataset. Secondly, we create independent variables using two strategies: in Strategy 1, each of the 30 independent variables is randomly generated from a normal distribution N (0,1); in Strategy 2, We standardized the PAH dataset and randomly selected 600 observations. We applied the following four functions to generate g(y):

a. Linear function: $g(y) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \varepsilon$

b. Linear function with significant interactions: $g(y) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 (x_3 * x_4) + \varepsilon$

c. Non-linear function with a quadratic term: $g(y) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_3 - x_4)^2 + \varepsilon$

d. Non-linear function with a cosine term: $g(y) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 \cos(x_3 * x_4) + \varepsilon$

Here, x_1 through x_5 represents the independent variables, and β_1 through β_5 are the regression coefficients. The regression coefficients are set to 5, 4, 3, 4, and 5, respectively. ε is the random error and is distributed as N(0,1).

Thirdly, the dependent variable is obtained by categorizing g(y) into four groups based on quartiles.

Finally, ten machine learning methods are applied to the simulated data for multi-class classification. Five-fold cross-validation is utilized to evaluate the performance of the machine learning methods. The data set is split into five parts. Each time, four parts of the data are selected as training data and the last part as test data. The training data is used to build up the

machine learning algorithm. The classification accuracy for the test data is then calculated, denoted as θ, for each particular machine learning method.

We will repeat the above procedures for each scenario 100 times. The average classification accuracy is calculated as $\theta = \sum_{i=1}^{100} \theta_i$.

In this part of the study, we're set to evaluate feature selection accuracy across multiple settings thoroughly. Guided by classification accuracy results from different simulation studies performed above, we choose RMLR, RF, and CNN for their superior classification accuracy compared to other methods. In each simulation, we'll identify the top contributing features to the outcome, compare these with the actual features, and calculate the proportion of correct feature selection, denoted as p. The mean selection accuracy will be computed by $P = \sum_{i=1}^{100} p_i$, representing the average accuracy across all simulations.

7. Simulation Results

Classification Accuracy of the 10 Machine Learning Methods

Tables 2 and 3 summarize the classification accuracy of ten machine learning models across four scenarios under Strategies 1 and 2. The comparison metrics include the mean prediction accuracy and standard deviation (SD) for each model on each test data set.

Table 2. Comparison of classification accuracies for simulated data in Strategy 1.

Strategy 1	**Scenario 1 (Linear)**		**Scenario 2 (Linear with interaction)**		**Scenario 3 (Non-linear with quadratic)**		**Scenario 4 (Non-linear with Cosine)**	
Model	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA	0.83	0.020	0.85	0.016	0.85	0.018	0.85	0.017
MLR	0.84	0.015	0.91	0.011	0.91	0.014	0.90	0.014
RMLR	**0.86**	0.015	**0.91**	0.012	**0.91**	0.012	**0.91**	0.014
SVM	0.67	0.025	0.68	0.022	0.68	0.023	0.68	0.021
NBC	0.56	0.020	0.63	0.027	0.63	0.023	0.63	0.023
DT	0.49	0.022	0.80	0.018	0.80	0.019	0.80	0.019
RF	0.62	0.020	0.78	0.019	0.78	0.018	0.78	0.019
KNN	0.39	0.024	0.39	0.025	0.39	0.022	0.39	0.025
GBM	0.65	0.023	0.82	0.018	0.83	0.016	0.83	0.016
CNN	0.74	0.021	0.76	0.016	0.75	0.017	0.75	0.019

Based on the simulation results for Strategy 1, we can observe the following key trends:

Linear (Scenario 1): RMLR emerged as the superior model with a mean accuracy of 0.86, while MLR closely followed with a mean accuracy of 0.84. LDA also posted a notable mean accuracy of 0.83.

Linear with Significant Interactions (Scenario 2): RMLR and MLR showcased an exceptional performance with the highest mean accuracy of 0.91. GBM followed with a good mean accuracy of 0.82.

Non-linear with Quadratic Term (Scenario 3): RMLR and MLR again dominated with an equally high mean accuracy of 0.91. GBM was not far behind, with a mean accuracy of 0.83.

Non-linear with Cosine Term (Scenario 4): RMLR exhibited the highest mean accuracy of 0.91, closely followed by MLR and GBM, with mean accuracies of 0.90 and 0.83, respectively.

To sum up, while MLR and RMLR consistently demonstrated superior performance, it's worth noting that models such as SVM, NBC, DT, RF, GBM, and CNN always performed less efficiently across all function types. KNN, in particular, was consistently the least accurate model in all the scenarios. As for the SD, it was relatively low across all models and scenarios, indicating the stability of model performances without significant variances. The largest SD observed was for KNN in Scenario 4 (0.025), while the smallest SD was for MLR in Scenario 2 (0.011). This suggests that the models' results, overall were tightly concentrated around the mean.

Table 3. Comparison of classification accuracies for simulated data in Strategy 2.

Strategy 2	**Scenario 5 (Linear)**		**Scenario 6 (Linear with interaction)**		**Scenario 7 (Non-linear with quadratic)**		**Scenario 8 (Non-linear with Cosine)**	
Model	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA	0.70	0.060	0.72	0.069	0.74	0.062	0.71	0.061
MLR	0.80	0.039	0.87	0.020	0.88	0.021	0.88	0.015
RMLR	**0.82**	0.043	0.89	0.026	0.89	0.027	0.89	0.024
SVM	0.76	0.055	0.76	0.070	0.77	0.054	0.76	0.058
NBC	0.67	0.057	0.70	0.080	0.70	0.078	0.70	0.076
DT	0.72	0.046	0.92	0.042	0.92	0.042	0.92	0.040
RF	0.78	0.047	0.92	0.032	0.92	0.037	0.91	0.036
KNN	0.70	0.053	0.72	0.065	0.74	0.058	0.72	0.055
GBM	0.78	0.044	**0.95**	0.027	**0.95**	0.029	**0.94**	0.025
CNN	0.80	0.044	0.84	0.011	0.84	0.011	0.84	0.010

Based on the simulation results for Strategy 2, we can observe the following key trends:

Linear (Scenario 5): RMLR achieved the highest mean accuracy at 0.82. Other high-performing models included MLR and CNN, with mean accuracies of 0.80.

Linear with Significant Interactions (Scenario 6): GBM outperformed all other models, boasting the highest mean accuracy of 0.95. This was closely followed by DT and RF, each with a mean accuracy of 0.92.

Non-linear with Quadratic Term (Scenario 7): GBM came out on top with the highest mean accuracy of 0.95. Similar to Scenario 6, DT and RF were the next best performing models, each with a mean accuracy of 0.92.

Non-linear with Cosine Term (Scenario 8): GBM continued its trend of high performance, achieving the highest mean accuracy of 0.94. RMLR and DT followed closely, with mean accuracies of 0.89 and 0.92, respectively. NBC, again, showed the lowest mean accuracy of 0.70.

Across all scenarios in Strategy 2, RMLR, DT, RF, and GBM have demonstrated superior performance across all function types, while SVM, LDA, KNN, and NBC consistently yielded lower mean accuracies compared to the other models. Mainly, NBC always produced the most insufficient mean accuracy in all scenarios. Regarding standard deviation (SD), the values generally suggest a moderate to high level of variability around the mean accuracy for each model. The largest SD was 0.080 (NBC, Scenario 6), while the smallest was 0.010 (CNN, Scenario 8), indicating a relatively wide range of results.

Feature Selection

Based on the results from classification accuracy, RMLR, RF, GBM and CNN were chosen to evaluate their performance for their feature selection further. Table 4 and 5 provides the mean and standard deviation of the feature selection accuracy of four methods used to build eight scenarios in Strategy 1 and 2.

Table 4. Sensitivity of feature selection in linear and linear with interaction scenarios.

Strategy 1	Scenario 1 (Linear)				Scenario 2 (Linear with interaction)			
	RMLR	**RF**	**GBM**	**CNN**	**RMLR**	**RF**	**GBM**	**CNN**
Mean	1.00	1.00	1.00	0.00	0.54	0.54	0.54	0.07
Std	0.00	0.00	0.00	0.00	0.06	0.05	0.06	0.05
Strategy 2	**Scenario 5 (Linear)**				**Scenario 6 (Linear with interaction)**			
	RMLR	**RF**	**GBM**	**CNN**	**RMLR**	**RF**	**GBM**	**CNN**
Mean	0.00	0.48	0.58	0.02	0.00	0.48	0.54	0.07
Std	0.00	0.17	0.17	0.03	0.00	0.13	0.10	0.06

Table 5. Sensitivity of feature selection in non-linear (quadratic and cosine) scenarios.

Strategy 1	Scenario 3 (Non-linear with quadratic)				Scenario 4 (Non-linear with Cosine)			
	RMLR	**RF**	**GBM**	**CNN**	**RMLR**	**RF**	**GBM**	**CNN**
Mean	0.53	0.54	0.54	0.07	0.53	0.53	0.54	0.07
Std	0.05	0.06	0.06	0.05	0.05	0.05	0.06	0.05
Strategy 2	**Scenario 7 (Non-linear with quadratic)**				**Scenario 8 (Non-linear with Cosine)**			
	RMLR	**RF**	**GBM**	**CNN**	**RMLR**	**RF**	**GBM**	**CNN**
Mean	0.00	0.51	0.53	0.07	0.00	0.47	0.53	0.08
Std	0.00	0.13	0.08	0.06	0.00	0.11	0.10	0.07

Based on the simulation results for Strategies 1 and 2, we can observe the following key trends:

Linear (Scenario 1 and 5): In Strategy 1, RMLR, RF, and GBM ideally identified all essential features (mean accuracy 1.00) with no variation (std deviation 0.00). However, CNN failed to identify basic features (mean accuracy 0.00). In Scenario 5, RMLR couldn't identify any important features (mean accuracy 0.00). RF and GBM had moderate performance (mean accuracy of 0.48 for RF and 0.58 for GBM) with moderate variability (std deviation of 0.17 for RF and GBM), and CNN barely recognized any features (mean accuracy 0.02).

Linear with Significant Interactions (Scenario 2 and 6): Under Strategy 1, RMLR, RF, and GBM exhibited moderate yet comparable accuracy around 0.54 with minimal variability (standard deviation of 0.06 for RMLR and GBM, and 0.05 for RF). CNN showed subpar accuracy (0.07). In Strategy 2, RMLR could not identify any significant features (mean accuracy of 0.00). GBM and RF's accuracy was marginally reduced compared to Scenario 1 (mean accuracy of 0.48), while CNN's performance remained comparable to Scenario 2 (mean accuracy of 0.07).

Non-linear with Quadratic Term (Scenario 3 and 7): Under Strategy 1, both RMLR, RF, and GBM displayed similar, moderate performance (mean accuracy of 0.53-0.54), with CNN showing inferior performance (mean accuracy of 0.07). For Strategy 2, RMLR was again unsuccessful in identifying any critical features (mean accuracy of 0.00), both GBM and RF's performance was slightly decreased compared to Strategy 1 (mean accuracy of 0.51), and CNN's performance was marginally inferior to Strategy 1 (mean accuracy of 0.07).

Non-linear with Cosine Term (Scenario 4 and 8): In Strategy 1, both RMLR, RF, and GBM showed similar, moderate performance (mean accuracy of 0.53-0.54), while CNN exhibited poor performance (mean accuracy of 0.07). Under Strategy 2, RMLR failed to identify crucial features (mean accuracy of 0.00), both GBM and RF's performance decreased slightly compared to Strategy 1, and CNN's performance decreased relative to Scenarios 2 and 3 in Strategy 2 (mean accuracy of 0.08).

In summary, in Strategy 1, RMLR, RF and GBM had moderate to perfect feature selection accuracy across all models, while CNN performed poorly. In Strategy 2, RMLR failed to recognize important features across all models. RF and GBM had moderate accuracy, and CNN had very low but slightly improved accuracy compared to Strategy 1. This suggests that the

nature of the independent data in different strategies significantly affects the accuracy of the feature selection.

From these results, we can conclude that the choice of feature selection method can significantly impact the ability to correctly identify essential features, particularly as the complexity of the relationships in the data increases. Each method has its strengths and weaknesses: RMLR performs well on linear relationships but struggles with complexity; RF and GBM show moderate and robust performance; and CNN works with more superficial structures but might potentially handle complex relationships better.

Application of Machine Learning Methods in PAH Data

In this study, we collected a total of 663 samples. Descriptive analysis revealed that the distributions of PAH compounds were generally consistent across all 19 sites, suggesting a relatively uniform spatial distribution of PAHs in the region. We defined seasons based on local climate conditions and changes in weather patterns that could potentially influence air quality. Therefore, our data incorporated four distinct meteorological seasons: spring, summer, fall, and winter. On the other hand, the sites were categorized based on urbanicity and potential sources of pollution. This led to classifying three types: urban, suburban, and rural.

However, it is worth noting that our dataset was not evenly distributed across the different categories, particularly concerning the site. This imbalance could potentially impact the initial results of our analysis, as machine learning models could be biased towards overrepresented classes.

Table 6. Classification accuracy of machine learning methods by season and site.

Model	Accuracy	
	Season	**Site**
LDA	0.79	0.56
MLR	0.86	0.59
RMLR	0.85	0.58
SVM	0.77	0.59
NBC	0.61	0.30
DT	0.82	0.59
RF	0.87	0.63
KNN	0.77	0.54
GBM	**0.88**	**0.65**
CNN	0.85	0.62

For season classification, the GBM model outperformed the other models with an accuracy score of 0.88. The RF and MLR models also demonstrated impressive performance with accuracy scores of 0.87 and 0.86, respectively. However, the NBC model struggled in this task, recording the lowest accuracy score of 0.61.

For site classification, the GBM model showcased superior performance again, yielding the highest accuracy of 0.65. The RF and CNN models were closely followed, which registered accuracy scores of 0.63 and 0.62, respectively. Like the season prediction, the NBC model underperformed in site prediction, producing the lowest accuracy score of 0.30.

These results indicate the effectiveness of the GBM, RF, and MLR models in analyzing PAH data, specifically for predicting season and site. Conversely, the results suggest that the NBC model might not be as effective in this context, given its comparatively lower prediction accuracy for both tasks.

Figure 3 illustrates the feature importance rankings determined by the Random Forest (RF) model for season and site, respectively. The importance score for each feature in the RF model indicates how much the model's prediction accuracy decreases when the feature is permuted (i.e., randomly shuffled). In other words, it reflects how much the model relies on each feature for accurate prediction. A higher importance score means the feature contributes more to the model's prediction accuracy.

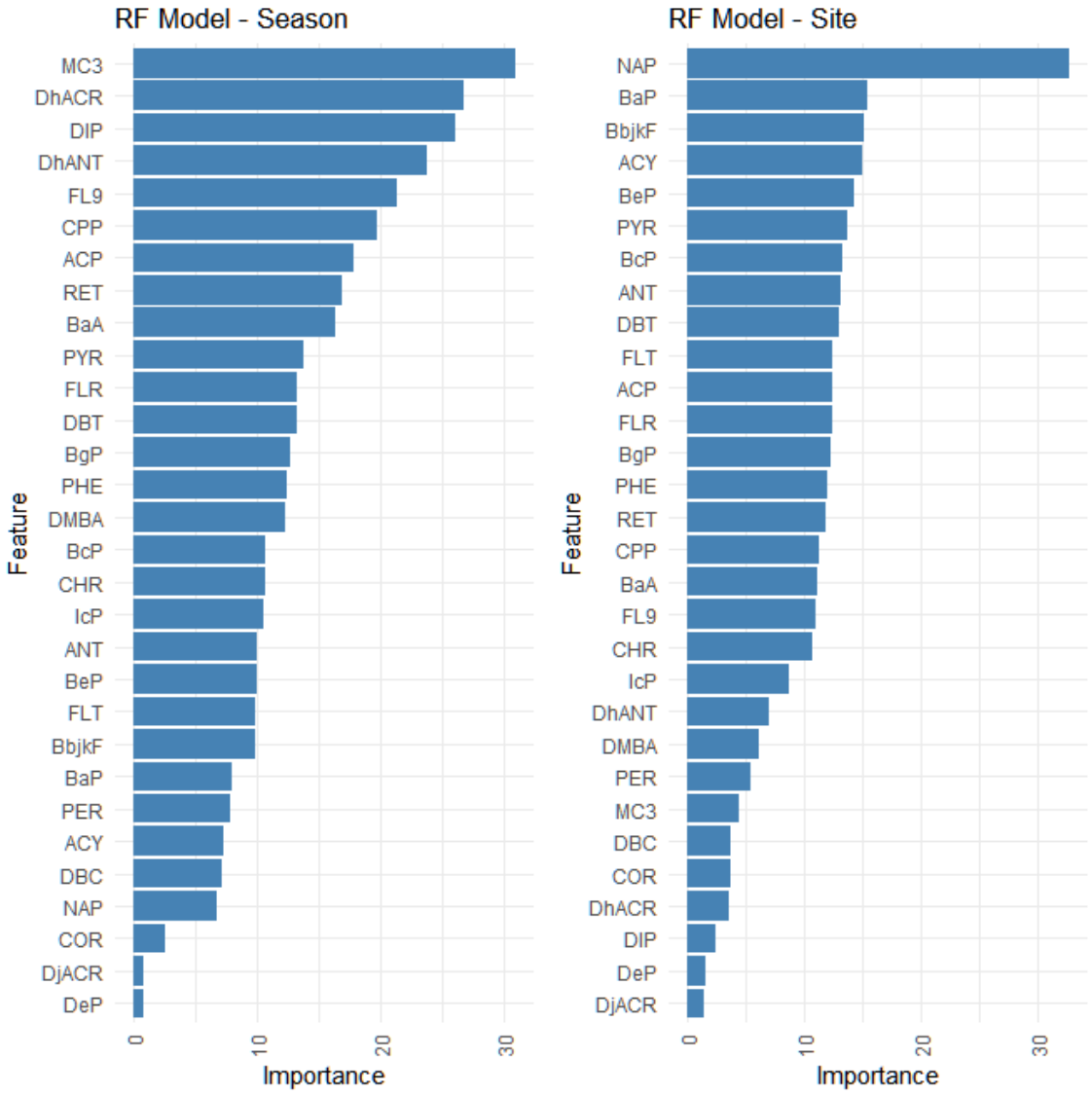

Figure 5. Seasonal and site-based analysis of feature importance in random forest.

For the season, MC3 has the highest importance score (30.94), indicating that it is the most influential factor in predicting the "Season" in the Random Forest model. Other features such as DhACR, DlP, and DhANT also show relatively high importance, suggesting these features also have substantial impacts on the model's classifications. Conversely, COR, DjACR, and DeP have much lower importance scores, implying they have less influence on the prediction of "Season" in this model.

NAP stands out as the most noteworthy feature of the site, with the highest importance score (32.74). Other significant features include BaP, BbjkF, and ACY, which also demonstrate high importance. On the other hand, features like DlP, DeP, and DjACR have the lowest importance scores, suggesting they contribute less to site classification.

Different sets of features are essential for predicting "Season" and "Site". This could suggest other environmental processes or sources of pollution are influencing these two different outcomes.

8. Discussions and Conclusions

The current study evaluated the performance of ten machine learning models using simulated environmental data. The resulting data reveals that there are no "one-size-fits-all" methods. Instead, the efficacy of each model is inherently tied to the unique characteristics of the dataset it applies to.

When the explanatory variables are independent, and their association with the outcome is linear, RMLR consistently outperformed their counterparts across all function types. However, the tables turned to Strategy 2, where the complexity of data relationships escalated. Here, GBM,

DT, and RF emerged as frontrunners, proving adept at managing interactions and non-linear function types.

Contrary to expectations, KNN, often triumphant in various classification problems, demonstrated the least accuracy throughout our tests. This could be attributed to the high-dimensional nature of the dataset, which may lead to overfitting or heightened difficulty in identifying suitable neighbors - a phenomenon known as the "curse of dimensionality". Similarly, the NBC also underperformed, potentially due to the NBC's assumption of feature independence, an assumption that may not be applicable in complex datasets.

Beyond the performance of classification accuracy of the machine learning methods, our study also examined their performance for feature selection. In Strategy 1, when the exploratory variables are independent, RMLR and RF performed exceptionally well, particularly with superficial linear relationships, while the CNN lagged. In Strategy 2, however, with heightened complexity, RMLR struggled to recognize any significant features, indicating potential difficulty when faced with more intricate data structures.

These findings serve as a potent reminder of the importance of careful model and method selection tailored to the specifics of the dataset. Misidentification of crucial features may lead to suboptimal model performance and potentially misleading insights.

It's important to acknowledge the limitations of this study, notably the characteristic uncertainty, and noise that often accompany environmental data. Future research could address this by integrating these aspects into the modeling process. Moreover, further investigations could focus on evaluating the models' performance using actual environmental datasets for a more pragmatic assessment and exploring other emerging machine learning models.

9. Future Work

Our research in applying machine learning methods to environmental data sets has yielded promising results. Moving forward, we foresee several potential directions to extend this work:

Extending Machine Learning Models: Although we have achieved significant results with our chosen models, more machine learning techniques may offer enhanced performance or provide different perspectives. For instance, we aim to explore the Bayesian Kernel Regression model, known for its ability to handle complex non-linear relationships and uncertainties in data, which are common in environmental studies.

Exploring Unsupervised Learning Methods: While our current work mainly focused on supervised learning models, future work will include unsupervised learning, specifically clustering methods. Evaluating and applying these methods to our PAH data set could reveal hidden structures or patterns in the data that might benefit classification or anomaly detection.

Application to Varied Pollutants and Data Sets: We aim to apply our validated machine learning methods to different types of pollutants or environmental data sets. By doing so, we can potentially identify and understand a broader spectrum of pollution sources, patterns, and their effects on the environment.

Integration of Socioeconomic Factors: In future research, we also plan to explore the interaction between environmental profile and socioeconomic factors. When integrated with environmental data, socioeconomic data can provide a more holistic view of environmental issues, highlighting disparities and guiding interventions.

By continuing to explore and apply machine learning techniques to environmental data, we aim to deepen our understanding of environmental pollution sources and patterns. This could contribute to the creation of more effective strategies for pollution prevention and control, leading to a healthier and more sustainable future.

References

Bajomo, M. M., Y. Ju, J. Zhou, S. Elefterescu, C. Farr, Y. Zhao, O. Neumann, P. Nordlander, A. Patel and N. J. Halas (2022). "Computational chromatography: A machine learning strategy for demixing individual chemical components in complex mixtures." Proceedings of the National Academy of Sciences **119**(52).

Baldauf, R., N. Watkins, D. Heist, C. Bailey, P. Rowley and R. Shores (2009). "Near-road air quality monitoring: Factors affecting network design and interpretation of data." Air Quality, Atmosphere & Health **2**(1): 1-9.

Bishop, C. M. and N. M. Nasrabadi (2006). Pattern recognition and machine learning, Springer.

Boser, B. E., I. M. Guyon and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory. Pittsburgh, Pennsylvania, USA, Association for Computing Machinery**:** 144–152.

Freund, Y. and R. E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." Journal of Computer and System Sciences **55**(1): 119-139.

Hao, Y. X., T. J. Fan, G. H. Sun, F. F. Li, N. Zhang, L. J. Zhao and R. G. Zhong (2022). "Environmental toxicity risk evaluation of nitroaromatic compounds: Machine learning driven binary/multiple classification and design of safe alternatives." FOOD AND CHEMICAL TOXICOLOGY **170**.

Heung, B., H. C. Ho, J. Zhang, A. Knudby, C. E. Bulmer and M. G. Schmidt (2016). "An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping." Geoderma **265**: 62-77.

Hino, M., E. Benami and N. Brooks (2018). "Machine learning for environmental monitoring." Nature Sustainability **1**(10): 583-588.

Jia, C., X. Fu and L. Smith (2023). "Dataset of atmospheric concentrations of polycyclic aromatic hydrocarbons in the Memphis Tri-state Area." Data in Brief **47**: 108923.

Kim, K.-H., S. A. Jahan, E. Kabir and R. J. C. Brown (2013). "A review of airborne polycyclic aromatic hydrocarbons (PAHs) and their human health effects." Environment International **60**: 71-80.

LeCun, Y., Y. Bengio and G. Hinton (2015). "Deep learning." nature **521**(7553): 436-444.

Liu, X., D. Lu, A. Zhang, Q. Liu and G. Jiang (2022). "Data-Driven Machine Learning in Environmental Pollution: Gains and Problems." Environmental Science & Technology **56**(4): 2124-2133.

Maitre, L., J.-B. Guimbaud, C. Warembourg, N. Güil-Oumrait, P. M. Petrone, M. Chadeau-Hyam, M. Vrijheid, X. Basagaña and J. R. Gonzalez (2022). "State-of-the-art methods for exposure-health studies: Results from the exposome data challenge event." Environment International **168**: 107422.

Manisalidis, I., E. Stavropoulou, A. Stavropoulos and E. Bezirtzoglou (2020). "Environmental and Health Impacts of Air Pollution: A Review." Frontiers in Public Health **8**.

Oskar, S. and J. A. Stingone (2020). "Machine Learning Within Studies of Early-Life Environmental Exposures and Child Health: Review of the Current Literature and Discussion of Next Steps." Current Environmental Health Reports **7**(3): 170-184.

Ramlogan, R. (1997). "Environment and human health: a threat to all." Environmental Management and Health **8**(2): 51-66.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais and Prabhat (2019). "Deep learning and process understanding for data-driven Earth system science." Nature **566**(7743): 195-204.

Sathya, R. and A. Abraham. (2013). "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification." International Journal of Advanced Research in Artificial Intelligence **2**: 34-38.

Schweitzer, L. and J. Noblet (2018). Chapter 3.6 - Water Contamination and Pollution. Green Chemistry. B. Török and T. Dransfield, Elsevier**:** 261-290.

Ştefan, R.-M. (2012). "A Comparison of Data Classification Methods." Procedia Economics and Finance **3**: 420-425.

Sunori, S., P. Bhakuni Negi, P. Juneja, N. M, P. G. Prakash, A. Mittal and S. Maurya (2021). Unsupervised and Supervised Learning based Classification Models for Air Pollution Data.

Tahmasebi, P., S. Kamrava, T. Bai and M. Sahimi (2020). "Machine learning in geo- and environmental sciences: From small to large scale." Advances in Water Resources **142**: 103619.

Zhao, Y., L. Wang, J. Luo, T. Huang, S. Tao, J. Liu, Y. Yu, Y. Huang, X. Liu and J. Ma (2019). "Deep Learning Prediction of Polycyclic Aromatic Hydrocarbons in the High Arctic." Environmental Science & Technology **53**(22): 13238-13245.

Zhong, S., K. Zhang, M. Bagheri, J. G. Burken, A. Gu, B. Li, X. Ma, B. L. Marrone, Z. J. Ren, J. Schrier, W. Shi, H. Tan, T. Wang, X. Wang, B. M. Wong, X. Xiao, X. Yu, J.-J. Zhu and H. Zhang (2021). "Machine Learning: New Ideas and Tools in Environmental Science and Engineering." Environmental Science & Technology **55**(19): 12741-12754.