Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Machine Learning Models for Predicting the Amount of Nutrients Required in a Microalgae Cultivation System

Geovani R. Freitas[a,b,c,d,e], Sara M. Badenes[c], Rui Oliveira[d,e], and Fernando G. Martins[a,b]*

[a] LEPABE, Department of Chemical Engineering, Faculty of Engineering, University of Porto, Porto, Portugal
[b] ALiCE, Associate Laboratory in Chemical Engineering, Faculty of Engineering, University of Porto, Porto, Portugal
[c] A4F – Algae for future, Campus do Lumiar, Estrada do Paço do Lumiar, Lisbon, Portugal
[d] UCIBIO, Department of Chemistry, NOVA School of Science and Technology, NOVA University Lisbon, Caparica, Portugal
[e] Associate Laboratory i4HB - Institute for Health and Bioeconomy, NOVA School of Science and Technology, Universidade NOVA de Lisboa, Portugal
* Corresponding Author: fgm@fe.up.pt.

ABSTRACT

Effective prediction of nutrient demands is crucial for optimising microalgae growth, maximising productivity and minimising the waste of resources. With the increasing amount of data related to microalgae cultivation systems, data mining and machine learning models to extract additional knowledge have gained popularity. In the development of such models, a data preprocessing stage is necessary due to the poor data quality. At this stage, cleaning and outlier removal techniques are employed to eliminate missing data and outliers, respectively. Afterwards, data splitting and cross-validation strategies are employed to ensure that the models are trained and evaluated with representative subsets of the data. Principal component analysis is also applied to simplify complex environmental datasets by reducing the number of features while retaining as much information as possible. To further improve prediction capabilities, ensemble methods are incorporated, leveraging multiple models to achieve a higher overall performance. Stacking, a popular ensemble technique, is used to combine the outputs of individual models into a single meta-model. The application of these machine learning methods has been demonstrated using a dataset acquired from the cultivation of the microalgae *Dunaliella* in a flat-panel photobioreactor. The results have shown that the data mining workflow, in combination with different machine learning models, was able to describe the nutrients' requirements enforcing good performance of microalgae *Dunaliella* β-carotene production in the carotenogenic phase.

Keywords: Machine Learning, Data Mining, Microalgae Cultivation, *Dunaliella* carotenogenesis.

INTRODUCTION

The use of data mining techniques and machine learning methods is rapidly increasing due to the need of extracting meaningful insights from large datasets. Data mining (DM) focuses on discovering patterns, trends, and valuable knowledge from the data, while machine learning (ML) develops algorithms that enable machines to improve at a task by learning from data without explicit programming [1]. In the DM process, various techniques are typically employed to uncover the underlying structure of data. However, when prediction of new instances is also crucial, ML algorithms are embedded within DM workflows to develop different models. This integration allows for more robust and accurate analysis, enhancing the overall capability of the DM process [2].

Various studies are found in the literature applying DM and ML methods to uncover valuable insights from large datasets on microalgae systems [3–5]. In this work, the application of these methods is demonstrated using a dataset obtained from the cultivation of the microalgae *Dunaliella* in a flat-panel photobioreactor (FP-PBR). This microalga is known for its ability to accumulate large amounts of carotenoids, with β-carotene being the most predominant. The high concentrations of this carotenoid change the colour of microalgae cells from green to orange. For the induction of β-carotene accumulation, *Dunaliella* cells should be exposed to certain growth

conditions, often also referred to as carotenogenic conditions, which are characterized by high salinity and light intensities, and limited nitrogen [6,7].

The main advantages of using FP-PBR systems to cultivate *Dunaliella* microalgae are their high productivity and uniform light distribution, employing narrow panels to achieve high area-to-volume ratios [8]. Therefore, the objective of this study is to develop ML models capable of predicting the nutrient requirements necessary to achieve good performance in the production of *Dunaliella* microalgae during the carotenogenic phase, for β-carotene production, in a FP-PBR system. This paper delves into the methods, where it is described the data preprocessing stage followed by the approach used for model development. The results section presents a comparative analysis of the different ML models, describing their evaluation metrics. Finally, the paper concludes by summarizing the key findings and highlighting their significance for process modeling.

METHODOLOGY

Dataset description and data preprocessing

Dataset was acquired from cultivating the microalga *Dunaliella* in a FP-PBR system. Data was selected based on two criteria: increase of the biomass concentration and production of carotenoids. Different features were considered to determine the target variable, i.e., the nutrient requirements of this microalga in carotenogenic phase. These features are described in the next section.

Data preprocessing is a critical step in the DM and ML pipeline, where raw data is transformed into a clean and usable format to improve data quality, ultimately enhancing the performance and accuracy of ML models. Typical preprocessing techniques include data cleaning, outlier detection and removal, feature scaling, feature selection, and dimensionality reduction.

Data cleaning

Data cleaning involves removing missing data, noise and correcting inconsistencies in the data [9]. In this work, it was decided to remove the observations with the missing values. Although this method is not very effective, its application can avoid the potential biases introduced by imputation methods like mean imputation.

Outlier detection and removal

Outliers are observations that significantly deviate from the overall distribution, which can skew results and lead to inaccurate model predictions. In datasets with multiple dimensions, one of the methods for detecting outliers is the calculation of the Mahalanobis distance [8], represented by the following equation:

$$D_M = \sqrt{(X-\mu)^T S^{-1} (X-\mu)} \quad (1)$$

In this formulation, X is the observation, μ is the mean of all observations, and S is the covariance matrix. The Mahalanobis distance follows a chi-squared distribution with degrees of freedom equal to the number of variables in the dataset. By selecting a significance level, a threshold can be determined from the chi-squared distribution. Observations with a squared Mahalanobis distance exceeding this threshold are detected as potential outliers and, therefore, should be removed [10].

Feature scaling

Feature scaling ensures that all numerical features are on a comparable scale. Without feature scaling, features with larger magnitudes can dominate the learning process, leading to biased or suboptimal model performance [9]. In this work, the z-score normalization was performed, also known as standardization. This approach transforms features to have a mean of 0 and a standard deviation of 1, being represented by the equation:

$$Z = \frac{X-\mu}{\sigma}, \quad (2)$$

where X is the feature value, μ is the mean, σ is the standard deviation.

Feature selection

Feature selection involves selecting a subset of the most relevant features from a dataset to improve model performance, reduce complexity, and prevent overfitting. One of the feature selection techniques is called minimum redundancy maximum relevance (mRMR). The criterion of maximum relevance ensures that the selected features are highly correlated with the target variable, therefore contributing significantly to the prediction task, while the criterion of minimum redundancy aims to reduce overlap between the selected features, ensuring that each one contributes unique information to the model. By selecting the most relevant features, mRMR helps in the development of models that are more accurate and interpretable [11].

Dimensionality reduction

When dealing with high-dimensional data, it is essential to perform dimensionality reduction techniques to decrease the number of variables in a dataset while retaining as much information as possible. Among various dimensionality reduction techniques, Principal Component Analysis (PCA) stands out as one of the most widely used and effective methods. PCA transforms the original features into a smaller set of uncorrelated components, called principal components (PC), which capture the maximum variance in the data [10]. The number of PC chosen is an important decision that is often guided by examining the plot of cumulative explained variance ratio versus number of PC, selecting the smallest number of PC that explain most of the data variance [9,12].

Model development

After data preprocessing stage, this work employed a systematic approach to model development, focusing on the design of different ML models. This approach consists of various stages, including data splitting, cross-validation, hyperparameter tuning, and ensemble learning.

In this work, the DM process and ML methods were programmed in Python 3.12 using the library Scikit-learn 1.4 and constructed using Jupyter Notebook 7.0 from Anaconda Navigator.

Data splitting

Data splitting consists of dividing the observations into at least training and test sets, where the training set is used for model construction and the test set for model evaluation. In this work, the Kennard-Stone (KS) algorithm was implemented for data splitting with a train-test split ratio of 80:20. The KS algorithm operates by maximizing the Euclidean distance between selected samples, ensuring that the chosen observations are evenly distributed across the feature space [13].

Cross-validation

Cross-validation (CV) is a method used to evaluate model performance and ensure generalizability to unseen data. One of the most popular techniques is k-fold CV, where the data is divided into k folds, and the model is trained and tested k times, each time using a different fold as the test set. Then, the results from each fold are averaged to produce a more robust estimate of the model performance [9,10]. In this work, it was applied k-fold CV with the training set being divided into 5 folds.

Hyperparameter tuning

Hyperparameters are settings that control the learning process of a ML model, such as learning rate, number of layers, or regularization strength. Unlike model parameters, which are learned from the data during training, hyperparameters must be predefined and set before the training process. Hyperparameter tuning is the process of selecting the optimal values for a ML model's hyperparameters. One effective method for hyperparameter tuning is GridSearch, which systematically evaluates all possible combinations of predefined hyperparameter values. This method ensures thorough exploration but can be computationally expensive [1,2].

Ensemble learning

Ensemble learning methods combine multiple models to improve prediction capabilities, robustness and generalization compared to individual models. Stacking, a popular ensemble technique, integrates predictions from several models using a single meta-model, which learns to optimize the final predictions. This approach leverages the strengths of different models, making stacking highly effective for complex problems [1,2].

Other ensemble technique is bagging, also known as bootstrap aggregating. This technique reduces variance by training multiple models on random subsets with replacement. Each model learns independently, and their outputs are aggregated. Bagging enhances stability and reduces overfitting, with random forest being the most common example that applies bagging to decision trees for improved performance [1,9].

Model evaluation

Evaluation metrics are essential for assessing the performance of ML models. The coefficient of determination (R^2) is used to measure the proportion of variance in the dependent variable that is explained by the model [10], being mathematically expressed by the equation below:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (3)$$

where n is the number of observations, y is the actual output, $\hat{y}$ is the predicted output, and $\bar{y}$ is the mean of the actual output.

Root mean squared error (RMSE) is another evaluation metric, whose value quantifies the average magnitude of errors between predicted and actual values, with lower values indicating higher accuracy [1]. RMSE is calculated according to the equation below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (4)$$

While R^2 provides a relative measure of fit, RMSE gives an absolute measure of prediction errors, making them complementary metrics for model evaluation.

RESULTS

Data preprocessing and constructing dataset

In the analysed dataset, the percentage of missing values was around 7 %. After removing such values, the Mahalanobis distance was calculated for each observation to detect possible outliers, as shown in Figure 1.

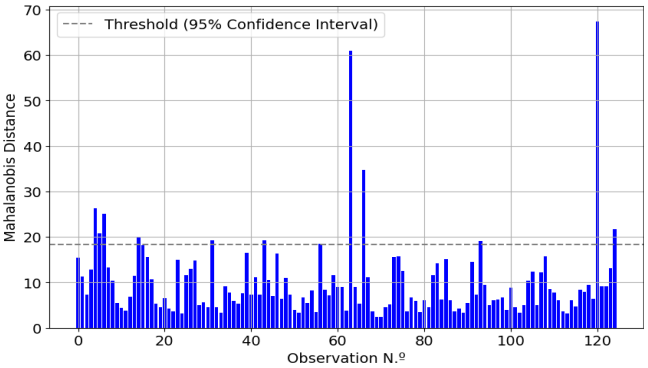

Figure 1: Mahalanobis distance for each observation in the analysed dataset.

By selecting a significance level of 95%, the threshold was determined based on the chi-squared distribution, represented by the dashed line. Observations with a Mahalanobis distance exceeding this value were identified as outliers and subsequently removed. After that, it was performed standardization to ensure that all features have a consistent scale. In the final stage of the preprocessing data workflow, PCA was applied to reduce complexity while preserving as much variability as possible. Figure 2 shows the percentage of the dataset variance that is cumulatively explained by each PC.

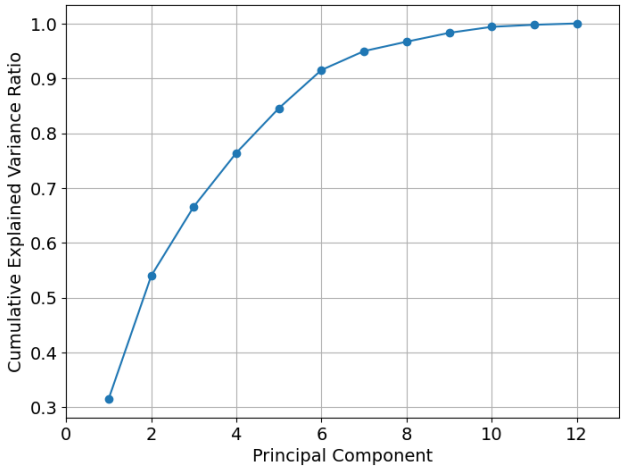

Figure 2: Cumulative variance ratio explained by the number of principal components.

From Figure 2, it was observed that the first PC explains around 30 % of the dataset variance. To explain 90 % of this variance, at least 6 PC are required, making 6 the chosen number of PC in this work.

Furthermore, the mRMR algorithm was applied to select the features relevant to the model: cells concentration, culture volume, irradiance, time duration for new addition of nutrients, and carotenoids concentration.

Model development and evaluation

After preprocessing the data, the KS algorithm was applied to split the observations into training (80 %) and test (20 %) sets. Then, five-fold CV was applied in combination with GridSearch to find the optimal values of hyperparameter for each ML model. In this work, four ML algorithms were employed to predict the target variable: ridge regressor (RR), random forest (RF), artificial neural network (ANN) and support vector regressor (SVR).

Although RR and RF are easier to interpret, they are better suited for linear or moderately complex relationships and struggle with highly nonlinear patterns. On the other hand, ANN and SVR excel at capturing nonlinear relationships, however they are more challenging to interpret and require careful tuning and computational resources, particularly for large datasets [1,10].

Table 1 shows the hyperparameter values optimized in each one of the ML models in this work.

Table 1: Optimal hyperparameter values obtained for each ML model.

Model	Hyperparameter	Optimal value
RR	Alpha	25
	Maximum number of iterations	1000
RF	Number of estimators	100
	Maximum depth	5
ANN	Number of hidden neurons	1st layer: 4 2nd layer: 2
	Activation function for the hidden layer	Hyperbolic tangent
	Alpha	0.1
SVR	C	1
	Kernel function	Radial basis
	Gamma	0.5

In Table 1, it was observed that ANN and SVR have a larger number of hyperparameters to optimize compared to RR and RF. In this case, a more detailed tuning process is necessary for achieving optimal performance.

After optimizing the hyperparameter values, scatter plots were created for each ML model. These plots visually represent the relationship between predicted and actual values, providing a clear evaluation of model performance. In addition, the coefficient of determination (R^2) was calculated for each model, quantifying how well the predictions aligned with the observed data. The results, summarized in Figure 3, show the effectiveness of each ML model in predicting the nutrient requirements, in arbitrary units (A. U.), for each observation in the analysed dataset.

According to Figure 3, RF and SVR models displayed signs of overfitting, highlighted by the difference between the R^2 values for the training and test sets. Overfitting occurs when the ML model learns not only the underlying patterns but also the noise in the training data, leading to an overly complex model that cannot predict new data accurately. In such cases, the R^2 for the training set is typically very high, indicating a well-fitted model, while the R^2 for the test set is much lower, revealing poor generalization to unseen data. Overfitting arises due to several causes, such as insufficient regularization, model complexity due to hyperparameter values, or excessive training iterations that allow the ML model to memorize noise instead of only learning meaningful patterns [1,9].

In the scatter plot of RR model (see Figure 3 (a)), some observations are relatively dispersed around the

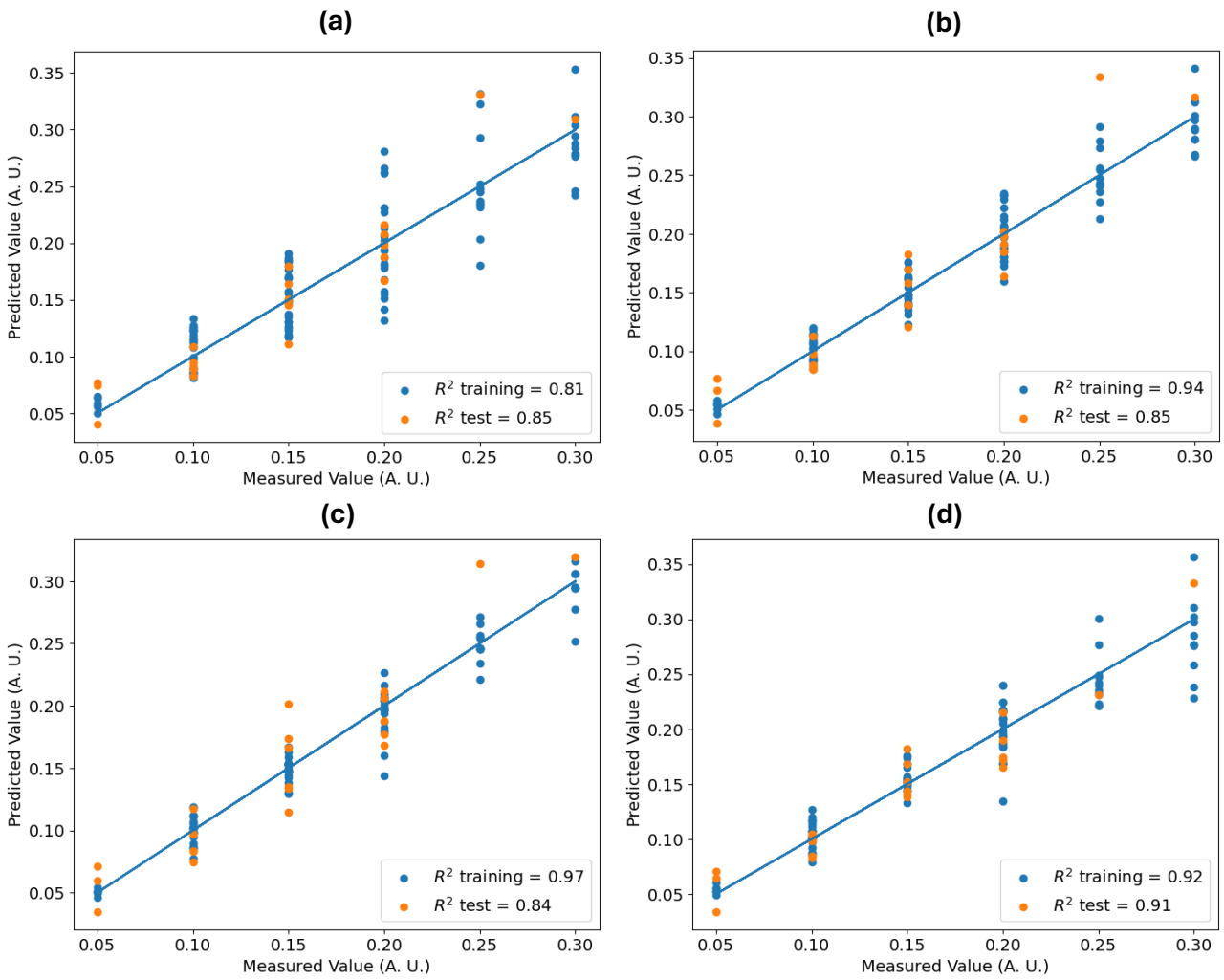

Figure 3: Scatter plots for each ML model considered in this work: (a) RR, (b) RF, (c) SVR, and (d) ANN. The coefficient of determination (R^2) is displayed for each plot, providing a quantitative measure of model performance.

diagonal line of perfect prediction, indicating moderate alignment between predictions and true values. In contrast, the scatter plot for ANN (see Figure 3 (d)) displays less observations dispersed along the diagonal, achieving superior R^2 values for both training and test sets. Superior RMSE values were also found for ANN. These results indicate that ANN generalizes better than RR, making it the preferred ML model for nutrient requirements.

Stacking process was also applied to the dataset by choosing as estimators the four ML models. In this case, the outputs of these ML models were considered in the stacking process, and ANN was chosen as the ML model that combines all the outputs, i.e., the meta-model. The scatter plot of the stacking model is shown in Figure 4.

In Figure 4, the scatter plot of the stacking model presents a good model fit, however when compared to the ANN model, its outputs exhibit more dispersion, reflecting a less accurate prediction. This is consistent with the fact that the R^2 values for both the training and test sets are higher in the ANN model compared to the stacking model, suggesting that the ANN model captures the underlying patterns in the data more effectively.

Several factors could contribute to the stacking model presenting a worse performance compared to the ANN model. One possibility is that the estimators used in the stacking process may not have provided complementary information, leading to limited improvement. In addition, the meta-model might not have been well-optimized, resulting in poor generalization on unseen data. Furthermore, the complexity added by the stacking method might have introduced more variance without offering significant predictive gains, especially if the estimators were not sufficiently diverse.

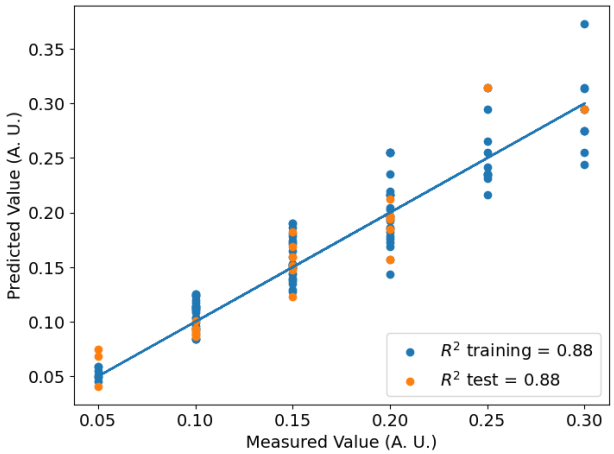

Figure 4: Scatter plot of the ML model generated through the stacking process.

CONCLUSION

This study demonstrates the potential of ML models to predict the nutrient requirements of *Dunaliella* microalgae during the carotenogenic phase in a FP-PBR system. The performance of four ML models was compared alongside a stacking model that combined the outputs of these models. Among all the ML models evaluated, the ANN model presented the best evaluation metrics, indicating its ability to capture complex relationships in the dataset and to provide accurate predictions. In addition, a key factor contributing to the good performance of these ML models was the implementation of a robust DM process, which included a detailed preprocessing stage. Finally, this study highlights the value of applying well-structured DM processes and systematically evaluating ML methods to identify the most suitable models for process modeling.

ACKNOWLEDGEMENTS

This work was supported by national funds through FCT/ MCTES (PIDDAC): LEPABE (Laboratory for Process Engineering, Environment, Biotechnology and Energy, Faculty of Engineering, University of Porto, UIDB/00511/2020, DOI: 10.54499/UIDB/00511/2020, UIDP/00511/2020, DOI: 10.54499/UIDP/00511/2020), ALiCE (Associate Laboratory in Chemical Engineering, Faculty of Engineering, University of Porto, LA/P/0045/2020, DOI: 10.54499/LA/P/0045/2020), UCIBIO (Research Unit on Applied Molecular Biosciences, UIDP/04378/2020, DOI: 10.54499/UIDP/04378/2020, UIDB/04378/2020, DOI: 10.54499/UIDB/04378/2020), and i4HB (Associate Laboratory Institute for Health and Bioeconomy, LA/P/0140/2020, DOI: 10.54499/LA/P/0140/2020). Geovani R. Freitas thanks the Portuguese Foundation for Science and Technology (FCT), Portugal, for the individual research grant PRT/BD/154543/2022.

REFERENCES

1. Géron A. HANDS-ON MACHINE LEARNING WITH SCIKIT-LEARN AND TENSORFLOW: CONCEPTS, TOOLS, AND TECHNIQUES TO BUILD INTELLIGENT SYSTEMS. O'Reilly (2018).
2. Witten I H, Frank E, Hall M A. DATA MINING: PRACTICAL MACHINE LEARNING TOOLS AND TECHNIQUES. Morgan Kaufmann Publishers (2011).
3. Reimann R, Zeng B, Jakopec M, Burdukiewicz M, Petrick I, Schierack P, Rödiger S. Classification of dead and living microalgae Chlorella vulgaris by bioimage informatics and machine learning. *Algal Res* 48 (2020).
4. Singh V, Verma M, Chivate M S, Mishra V. Machine learning-based optimisation of microalgae biomass production by using wastewater. *J Environ Chem Eng* 11 (2023).
5. Meenatchisundaram K, Gowd S C, Lee J, Barathi S, Rajendran K. Data-driven model development for prediction and optimization of biomass yield of microalgae-based wastewater treatment. *Sustain En Tech and Assmt* 63 (2024).
6. Barbosa M, Inácio L G, Afonso C, Maranhão P. The microalga Dunaliella and its applications: a review. *Applied Phycology* 4:99-120 (2023).
7. Ye Z W, Jiang J G, Wu G H. Biosynthesis and regulation of carotenoids in Dunaliella: Progresses and prospects. *Biotechnol Adv* 26:352-360 (2008).
8. Carvalho A P, Meireles L A, Malcata F X. Microalgal reactors: A review of enclosed system designs and performances. *Biotechnol Prog* 22:1490-1506 (2006).
9. Han J, Kamber M, Pei J. DATA MINING: CONCEPTS AND TECHNIQUES. 3rd Edition, Morgan Kaufmann Publishers. Elsevier (2012).
10. Alpaydin E. INTRODUCTION TO MACHINE LEARNING. The MIT Press (2010).
11. Ding C, Peng H. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. *Proceedings of the Computational Systems Bioinformatics* (2003).
12. Mazzelli A, Cicci A, Di Caprio F, Altimari P, Toro L, Iaquaniello G, Pagnanelli F. Multivariate modeling for microalgae growth in outdoor photobioreactors. *Algal Res* 45 (2020).
13. Kennard R W, Stone L A. Computer Aided Design of Experiments. *Technometrics* 11:137-148 (1969).