Journal of Power and Energy Engineering, 2025, 13(7), 40-59
https://www.scirp.org/journal/jpee
ISSN Online: 2327-5901
ISSN Print: 2327-588X

An Empirical Analysis on Renewable Energy: Biogas Production Prediction Using Machine Learning

Md. Mahedi Hassan[1], Arif Hossen[2], Md. Nurunnabi Sarker[3], Yeasin Arafat[4], Aslam Khan[1], Shafiqul Islam Talukder[5], Bikash Kumar Saha Roy[1]

[1]Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh
[2]Business Analytics, International American University, California, USA
[3]Data Analysis, Westcliff University, California, USA
[4]IT Management, Westcliff University, California, USA
[5]Computer Science, Westcliff University, California, USA
Email: mahedi7171@gmail.com, arifhossen4295@gmail.com, nurunnabisarker17@gmail.com, y.arafat.570@westcliff.edu, mail2aslkh@gmail.com, shafiqul.cse2017@gmail.com, talk2rajucis@gmail.com

How to cite this paper: Hassan, Md.M., Hossen, A., Sarker, Md.N., Arafat, Y., Khan, A., Talukder, S.I. and Roy, B.K.S. (2025) An Empirical Analysis on Renewable Energy: Biogas Production Prediction Using Machine Learning. *Journal of Power and Energy Engineering*, **13**, 40-59.
https://doi.org/10.4236/jpee.2025.137002

Received: June 20, 2025
Accepted: July 26, 2025
Published: July 29, 2025

Abstract

Biogas is gaining prominence as a renewable energy source with significant potential to reduce greenhouse gas emissions and mitigate environmental impacts associated with fossil fuels. This study presents an improved biogas production estimation method using machine learning (Ridge Regression, Lasso Regression, Random Forest, XGBoost, LightGBM, and GBM) combined with explainable AI (XAI) techniques to enhance model interpretability. Our rigorous evaluation using Wilcoxon Signed-Rank Tests demonstrated that LightGBM and XGBoost consistently outperformed other algorithms—LightGBM achieved superior performance in the 70:30 train-test split (RMSE = 0.075, R^2 = 0.895), while XGBoost excelled in both 80:20 (RMSE = 0.091, R^2 = 0.847) and 50:50 splits. These models proved significantly better than traditional methods ($p < 0.05$) in all comparisons, with particularly strong performance against linear regression approaches (p-values as low as 0.001). The analysis identified waste efficiency and total daily waste (kg/day) as the most critical predictive features. Despite dataset limitations (U.S.-only livestock data, n = 344), these findings offer valuable guidance for biogas professionals and investors to optimize production forecasting and operational management strategies for improved renewable energy outputs. The research demonstrates how machine learning can enhance both prediction accuracy and interpretability in renewable energy applications.

Keywords

Biogas, Renewable Energy, Energy Production Management, Machine Learning, XAI, Analysis, Shapash

1. Introduction

Due to the refuse that domestic and industrial activities have produced, more and more well-developed and emerging nations have turned to identifying other energy sources without these sources. Not too long ago, fossil fuels were the world's chief energy source for most of the global primary energy supply. Yet, the environmental havoc generated by fossil fuels and the reduction of natural resources have brought renewable energy sources into the spotlight as a prospective contender to ensure an environmentally sustainable future for generating energy. Interest in biogas as an alternative energy source has grown significantly over the past few years, primarily because of its ability to reduce greenhouse gas emissions. The minutes to days are the retention time, pH and compositions of medium (organic carbon and nitrogen), the temperature in the digester tank, the operating pressure of digested systems, and volatile fatty acids as essential parameters controlling biogas production yield from anaerobic digestion processes [1]. Moreover, machine learning (ML) for testing this model is a promising method to approximate complex nonlinear associations with potential outcomes. This classifies it as having the highest potential for predicting and controlling anaerobic digester performance [2]. All this is done using computer algorithms that learn from data and find insights in the absence of being explicitly programmed where to look. Numerous researchers have proposed innovative and effective strategies for modeling the biogas process using ML techniques. These methodologies encompass support vector machines, adaptive neuro-fuzzy inference systems, k-nearest neighbors, random forests, and artificial neural networks [3]. In controlled laboratory-scale experiments, three-layer artificial neural networks and nonlinear regression models have been used to predict biogas production performance [4]. Besides, adaptive neuro-fuzzy inference systems were used to simulate and optimize biogas generation from cow manure combined with maize straw in a study at pilot scale [5]. On the other hand, random forest and extreme gradient boosting (XGBoost) were successfully utilized in an industrial-scale co-digestion plant [6]. More literature should be written regarding artificial intelligence-based models for estimating biogas production and identifying essential factors influencing production from full-scale sludge digestion processes in biological treatment plants. Most researchers are focused on biogas output prediction and model building with laboratory- or pilot-scale reactors. The current study uses algorithms such as Ridge Regression, Lasso Regression, k-nearest Neighbor (KNN), ElasticNet Regression, Classification and Regression Trees, Random Forest, and finally Extreme Gradient Boosting, Light Gradient Boosting Machine, Gradient Boosting Ma-

chine, and CatBoost on U.S. biogas data. The biogas output rates were predicted using the data, and it was transferred to a fully operational anaerobic sewage digester system. The main aim of the current study is to check the efficacy of these machine learning models and validate the important variables that predict biogas production.

The technical contributions of this paper are as follows:

- To look at how well different machine learning methods work at predicting daily biogas production and compare them.
- To optimize the algorithm's effectiveness through hyperparameter tuning and various preprocessing methods.
- To evaluate the significant differences in performance between two machine learning models based on their RMSE scores, using the Paired Wilcoxon Signed-Rank Test across multiple train-test splits.
- To deliver information and suggestions based on the test results to help the right organizations and funders with both global and local explanations using the XAI tool SHAPASH.

This paper part is structured in the rest of the section. As part of the related works, this reads Section 2. Section 3 presents the methodological approach we used in our experiments. This subsection will describe the methodology used to address the research problem/objective and elaborate on all methods, techniques, and tools. The results of our investigations are detailed in Section 4. The paper concludes with a summary of our findings and their implications in Section 5. In conclusion, we address the potential for future research in this discipline to be further investigated and refined.

2. Literature Review

Traditional statistical methods have been implemented in numerous investigations. For example, De Clercq *et al.* [7] employed operations research methods, such as data envelopment analysis, and statistical techniques, such as principal component analysis and multiple linear regression, to examine the factors influencing efficient biogas projects. Their research emphasized a variety of inefficiencies, such as diminishing returns to scale. Similarly, Terradas-Ill *et al.* [8] created a thermal model to predict biogas production in underground, unheated fixed-dome digesters. Nevertheless, their model needed to be more suitable for large-scale facilities and needed more validation against actual data. De Clercq *et al.* [9] evaluated food waste and biowaste initiatives using multi-criteria decision analysis, which considered technical, economic, and environmental factors. In light of their findings, they suggested six significant policy recommendations; however, they could have offered generalized modeling tools that project administrators could employ to enhance production efficiency by utilizing waste inputs. Nevertheless, the models developed in these studies are significantly limited by their inability to incorporate the most recent developments in machine learning to predict biogas output. Instead, they depend on conventional statistical performance metrics, in-

cluding root-mean-square error (RMSE) and R^2. Conversely, contemporary machine learning models are assessed according to their capacity to forecast data that has not yet been observed effectively [10]. To accomplish this, datasets are partitioned into training and testing partitions, with a preference for out-of-sample evaluation metrics. These metrics are essential because they assist in the identification of potential overfitting of the model to the training data. Moreover, these conventional models are limited by a balance between simplicity and precision; as such, they cannot accurately model the complex relationships between different biochemical entities. On the other hand, ML models are inherently universal approximators of reason [11]. Since ML models have numerous tunable parameters, they can identify subtle patterns in anaerobic digestion data sets without expert guidance. Several ML-based schemes that have been used for the prediction of biogas are as follows: Wang *et al.* [12] built a Tree-Based Automated ML AutoML to predict biogas production from anaerobic co-digestion of degradable organic waste. Sonwai *et al.* [13] compared the performance of RF, XGBoost random forest, and KRR applied on three models to predict the performances of SMY. The RF model was the best, with RS = 0.85 and an RMSE = 0.06. used RF to predict a computer-generated ADM1 dataset and then compared the results with three ML systems and a random forest model. Cheon *et al.* [14] predicted the methane yield in an anaerobic dissolving bio-electrochemical reactor by five ML models and demonstrated the ability of the methodology to interpret complex non-linear relationships throughout several input and output significant variables in a sophisticated, complex scheme. The authors argued that an effective prediction model can aid in process stability and prevent operational hazards by using the model in real near time. De Clercq *et al.* [15] utilized logistic regression, SVM, and k-NN regression models on an industrial biogas facility's data heap in China to improve the operational daily decision in ML models. Noticeably, this work did not focus on digester parameters such as temperature. A graphical user interface was used to convey the suggestions to various wastewater treatment plant engineers daily. In another study, researchers [16] compared five different ML algorithms, namely, ANN, RF, KNN, SVR, and XGBoost, on the ML to predict reliability percentile. Yildirim & Ozkaya provided a benchmark of RS = 0.9242 for prediction. Most researchers have used ML because of multilayer artificial neural networks, which are highly accepted and widely commented by the engineering community, for example, optimized ANN and FCM-clustered ANFIS methodologies used to predict biogas and methane performance. FCM-ANFIS with ten clusters obtained an RS = 0.9850, MAD = 1.2463, MAPE = 5.2343, and RMSE = 1.2343, which indicates an accurate model compared to the ANN approach.

A literature review shows that many researchers have used traditional statistical methods in predicting biogas production, while others have employed ANN and ML techniques [8]. However, it is necessary to utilize more suitable ML models and XAI methods, such as feature importance, to investigate the crucial levels influencing biogas yield. In this research, we intend to address these drawbacks by using

more appropriate ML models and XAI methods for feature identification.

3. Methodology

In this study, we propose a new top-down approach that uses machine learning techniques for preprocessing the data to predict daily biogas production using the secondary dataset. This is something new in our research because such an approach includes numerous explainable artificial intelligence (XAI) models to find the most substantial factors impacting biogas production.

3.1. Overview of the Methodology

This study develops a machine learning methodology for biogas production prediction using operational data from U.S. livestock facilities, implementing a rigorous analytical pipeline that begins with comprehensive data preprocessing, including ordinal and one-hot encoding, alongside data normalization demonstrated in Figure 1. We systematically evaluate model performance across three train-test splits (80:20, 70:30, and 50:50) incorporating both traditional regression techniques (Ridge, Lasso) and advanced ensemble methods (Random Forest, GBM, XGBoost, LightGBM), with hyperparameter optimization and cross-validation to ensure robustness. The evaluation employs RMSE and R^2 metrics complemented by Wilcoxon signed-rank tests to statistically verify performance differences, while SHAP analysis provides interpretability by identifying key predictive features like waste inputs and their directional impacts. This integrated approach combines predictive modeling with explainable AI techniques and offers both accurate forecasts and operational insights while addressing potential overfitting through regularization and extensive validation across multiple data partitions despite the limitations of a U.S.-specific dataset with a moderate sample size.

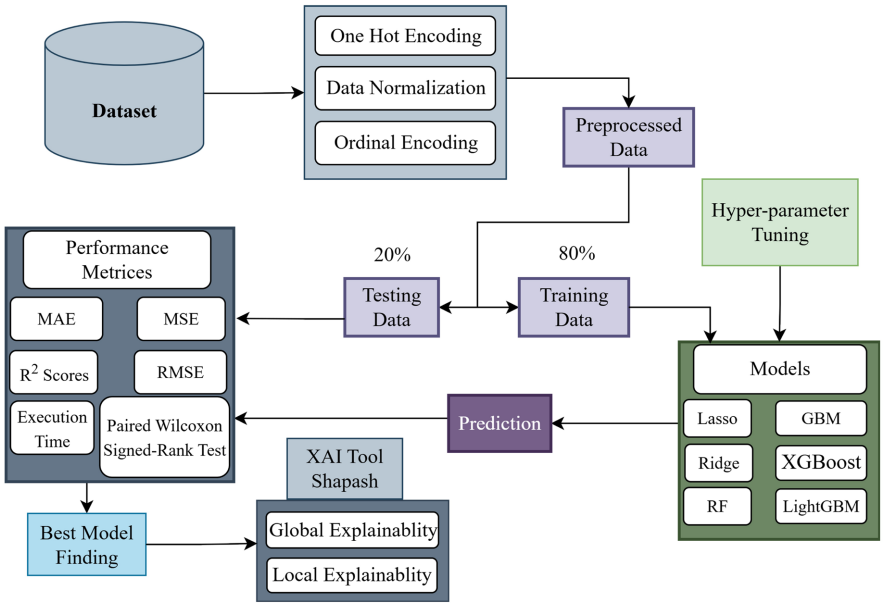

Figure 1. Overview of the proposed methodology.

3.2. Description of Dataset and Variables

This study uses data from the U.S. Biogas dataset on Kaggle to forecast biogas production at regular intervals [17]. Covering cattle farms from around the U.S., this large dataset paints a broad picture of how biogas is produced. A vital tool for understanding when and how much renewable energy may be usable. It has data from cattle, dairy cows, and pig biogas projects with chickens. It can prove handy for people from different professions, including agriculture, green energy, environmental policy, etc., displayed in Table 1. The dataset consists of 491 observations from various places in the United States and about 29 attributes. Data preparation is one of the vital parts of machine learning, but this step consumes too much time work (around 60% of the budget for a data science project) [2]. Missing values were handled through median imputation for numerical data and mode imputation for categorical data. Outliers were removed using the IQR method. Categorical variables were encoded using one-hot encoding.

Table 1. A brief description of the dataset.

Column name	Description
Year Operational	The year when the project became operational.
Cattle	Number of cattle involved.
Dairy	Number of dairy cows involved.
Poultry	Number of poultry involved.
Swine	Number of swine involved.
Biogas Generation Estimate (cu-ft/day)	Estimated daily biogas production.
Electricity Generated (kWh/yr)	Estimated annual electricity generation.
Total Emission Reductions (MTCO2e/yr)	Estimated total emission reduction.
Operational Years	Number of years the project has been operational.
Total_Animals	Total number of animals involved in the project.
Biogas_per_Animal (cu-ft/day)	Estimated biogas production per animal.
Emission_Reduction_per_Year	Estimated annual emission reduction per animal.
Electricity_to_Biogas_Ratio	The ratio between electricity generation and biogas production.
Total_Waste_kg/day	Estimated daily waste production.
Waste_Efficiency	Efficiency of waste conversion to biogas.
Electricity_Efficiency	Efficiency of biogas conversion to electricity.

3.3. Machine Learning Algorithms

3.3.1. Lasso Regression

Lasso regression, or Least Absolute Shrinkage and Selection Operator, is a reg-

ularization technique used to enhance the prediction accuracy and interpretability of regression models by enforcing sparsity. Unlike ridge regression, which applies an L_2 penalty, lasso regression adds an L_1 penalty to the loss function, where Y_i are the observed values, $\widehat{Yi}$ that predicted values, βj the coefficients, and λ the regularization parameter [18]. The L_1 penalty tends to shrink some coefficients exactly to zero, effectively performing variable selection and yielding a simpler model that retains only the most significant predictors. This sparsity property makes lasso regression particularly useful when dealing with high-dimensional data where the number of predictors exceeds the number of observations [19]. By appropriately tuning the λ parameter, one can control the complexity of the model, balancing bias and variance to improve predictive performance and interpretability [20]. Lasso regression is widely utilized in fields like bioinformatics and economics where model simplicity and feature selection are crucial [21].

3.3.2. Ridge Regression

Ridge regression is an extension of linear regression that addresses multicollinearity among predictor variables by incorporating a regularization term. This technique adds a penalty equal to the square of the magnitude of the coefficients to the loss function, thus shrinking the coefficients and reducing their variance [22]. The ridge regression equation modifies the ordinary least squares (OLS) regression by adding a regularization parameter λ, which minimizes the following cost function where Y_i represents the observed values, Y_i the predicted values, βj the coefficients, and λ the regularization parameter. By tuning λ, one can control the trade-off between fitting the data well and keeping the model coefficients small, which helps mitigate overfitting. Ridge regression is instrumental in situations with many correlated predictors, as it improves the model's generalization performance [20]. This method retains all predictors in the final model, unlike other techniques such as Lasso regression, which can shrink some coefficients to zero [19].

3.3.3. Gradient Boosting Machine (GBM)

Gradient Boosting Machine (GBM) is an ensemble learning technique that builds a strong predictive model by sequentially combining multiple weak learners, typically decision trees, to minimize a predefined loss function. GBM constructs each tree iteratively, focusing on reducing the errors made by the previous trees. At each iteration, GBM fits a new tree to the residuals or negative gradients of the loss function from the ensemble of previously built trees. The new tree is then added to the ensemble, with its predictions weighted based on a learning rate parameter. This process continues iteratively until a predefined number of trees is reached or until further iterations cease to improve performance [23]. GBM is highly flexible and capable of capturing complex nonlinear relationships in the data. However, it is sensitive to hyperparameters such as the learning rate, tree depth, and the number of trees in the ensemble. Careful tuning of these hyperpa-

rameters is essential to prevent overfitting and achieve optimal performance. GBM has achieved remarkable success in various ML competitions and is widely used in practice for tasks such as classification, regression, and ranking [24].

3.3.4. Random Forest (RF)

Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy and generalization performance. Each decision tree in the forest is trained on a bootstrap sample of the training data and selects the best feature from a random subset of features at each split. The final prediction in Random Forest is typically made by averaging or taking a majority vote of the predictions from individual trees [25]. Random Forest offers several advantages over a single decision tree. Firstly, it reduces overfitting by averaging predictions from multiple trees, thereby improving the model's ability to generalize to unseen data. Secondly, it provides estimates of feature importance, which can help identify the most informative features in the dataset. The key hyperparameters in Random Forest include the number of trees in the forest Ntrees and the size of the random feature subset considered at each split m. These hyperparameters influence the model's bias-variance trade-off: increasing Ntrees typically reduces variance but may increase computational cost, while increasing m can decrease correlation between trees but may increase bias. Random Forest is robust to noisy data and can handle high-dimensional feature spaces, making it a popular choice for classification and regression tasks in various domains [19]. However, its interpretability is lower compared to single decision trees, as it is more challenging to interpret the combined predictions of multiple trees [26].

3.3.5. LightGBM

LightGBM is a gradient boosting framework developed by Microsoft that focuses on efficiency, speed, and accuracy. It is designed to handle large-scale datasets and can be significantly faster than other gradient boosting implementations, making it well-suited for both industry-scale applications and research [27]. LightGBM adopts a novel approach to tree construction known as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which enable it to achieve faster training times and lower memory usage without sacrificing predictive performance. Additionally, LightGBM supports parallel and distributed computing, allowing it to scale efficiently to multi-core CPUs and distributed computing environments [28]. The key hyperparameters in LightGBM include the learning rate η, tree depth maxdepth, number of leaves num_leaves, and regularization parameters λ and a. These hyperparameters influence the model's capacity, complexity, and generalization ability and must be carefully tuned to optimize performance. LightGBM offers native support for categorical features and missing values, which simplifies data preprocessing and feature engineering tasks. It supports both classification and regression tasks and can handle various loss functions, including binary classification, multi-class classification, and regression. LightGBM has become a popular choice among ML practitioners and researchers due to its

excellent performance, scalability, and ease of use.

3.3.6. XGBoost

XGBoost, short for eXtreme Gradient Boosting, is an optimized distributed gradient boosting library designed for efficiency, scalability, and flexibility [29]. It is an extension of the gradient boosting framework that focuses on computational speed and model performance. XGBoost has gained widespread popularity in ML competitions and real-world applications due to its state-of-the-art performance and robustness. XGBoost employs a boosted ensemble of decision trees, where each tree is trained sequentially to correct the errors made by the previous ones. It incorporates several innovative techniques, such as parallelized tree construction, approximate tree learning, and regularization, to improve training speed and accuracy [24]. Additionally, XGBoost supports both classification and regression tasks and can handle large-scale datasets with millions of samples and features. The key hyperparameters in XGBoost include the learning rate η, tree depth d, regularization parameters γ for minimum loss reduction and λ for regularization term on weights, and the number of trees in the ensemble T. These hyperparameters influence the model's bias-variance trade-off and must be carefully tuned to optimize performance and prevent overfitting. Where l is the loss function, Y_i is the true label of sample i, $\widehat{Yi}$ is the predicted value, T is the number of trees, $\bar{O}(f_k)$ is the regularization term for tree k, and f_k represents the predictions of the k-th tree. XGBoost has become a go-to choose for many ML practitioners and researchers due to its outstanding performance, ease of use, and versatility across various domains and datasets [23].

3.4. XAI Tools

Shapash

Shapash is a Python library designed for model interpretation and explanation. It extends the SHAP (Shapley Additive exPlanations) framework by providing automated, customizable, and interactive explanations for ML models [30]. Shapash simplifies the process of understanding model predictions and feature impacts, making it accessible to users with varying levels of expertise. One of the key features of Shapash is its ability to generate intuitive and interactive visualizations of SHAP values, allowing users to explore how individual features contribute to model predictions [31]. These visualizations include summary plots, force plots, and dependence plots, which provide insights into feature importance, interactions, and effects on predictions. Shapash also offers functionality for model comparison, sensitivity analysis, and global feature importance assessment, enabling users to gain a comprehensive understanding of their models' behavior. It supports various ML models, including tree-based models, linear models, and ensemble methods, making it versatile across different domains and applications [32]. With its user-friendly interface and powerful visualization capabilities, Shapash has become a valuable tool for model interpretation, debugging, and validation in data science projects.

3.5. Performance Measure Metrics

Several performance evaluation metrics can be used in the measurement of the accuracy of a model. Here is a description of some common ones:

RMSE (Root Mean Square Error): RMSE measures the square root of the average squared differences between the predicted and actual values. It provides a way to measure the magnitude of prediction errors, with lower values indicating better model performance.

R^2 Scores (Coefficient of Determination): The R^2 score quantifies how well the predicted values approximate the actual values. It ranges from 0 to 1, with higher values indicating a better fit. An R^2 score of 1 means the model explains all the variability of the target data around its mean.

MAE (Mean Absolute Error): MAE calculates the average absolute differences between predicted and actual values. It is a straightforward measure of prediction accuracy, with lower values indicating fewer errors and better model performance.

MSE (Mean Squared Error): MSE measures the average of the squared differences between predicted and actual values. It emphasizes larger errors due to the squaring process, with lower values indicating better performance.

Execution Times: Execution time refers to the amount of time a model takes to train and make predictions. Shorter execution times are generally preferable, especially in applications requiring real-time or near-real-time predictions.

4. Result Analysis

After implementation of the proposed methodology several outputs have been obtained from the analysis.

4.1. Hyperpramaeter Tuning on the Models

We conduct hyperparameter tuning on the selected machine learning models, which include Ridge, Lasso, Random Forest (RF), Gradient Boosting Machine (GBM), XGBoost, and LightGBM, to enhance their performance. This process involves using grid search, a technique that systematically works through multiple combinations of hyperparameters, and cross-validation to assess the models' effectiveness and ensure they generalize well to unseen data. The goal is to identify the optimal set of parameters for each model, improving their accuracy and overall predictive capabilities. The best hyperparameters recommended for Ridge, Lasso, and other regression methods (including RF—Random Forest, GBM—Gradient Boosting Machine, XGBoost, and LightGBM) are highlighted in Table 2. The best hyperparameter for Ridge with value 1.0 and Lasso for the poly dataset is "alpha". In Random Forest, the important hyperparameters are "n_estimators and max_depth" assigned as 10 and 100. The optimal hyperparameters for GBM and XGBoost are thus: "learning_rate" = 0.05 and "n_estimators" = 100. The LightGBM function also uses learning_rate as 0.1 and n_estimators as 100, which are the same as the Random Forest Classifier for these values governed above.

These are necessary hyperparameter settings which help to improve the performance of each algorithm in regression tasks.

Table 2. Hyperparameters value of the regressors.

Algorithms	Best Hyperparameters	Hyperparameter Value
Lasso	"alpha"	0.01
Ridge	"alpha"	1.0
GBM	"learning_rate", "n_estimators"	0.05, 100
RF	"max_depth", "n_estimators"	10, 100
LightGBM	"learning_rate", "n_estimators"	0.1, 100
XGBoost	"learning_rate", "n_estimators"	0.05, 100

4.2. Result of the ML Regressor in the Different Ratio of Training and Testing

The performance of various regression algorithms using an 80:20 training-to-testing ratio is summarised in Table 3. Based on several measures, including execution times, R^2 Scores, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), Random Forest (RF), LightGBM, and XGBoost, the table compares the performance of several regression techniques. Despite having a moderate execution time of 11.089 seconds, XGBoost stands out as the top performance with the lowest error metrics (RMSE: 0.091, MAE: 0.051, MSE: 0.008) and the most excellent R^2 score of 0.847. Then comes LightGBM, which has a slightly greater error rate but runs faster. While GBM and RF have the same RMSE and MSE, RF outperforms GBM in MAE but takes far longer to execute. Lasso has the most significant error metrics but makes up for it with the quickest execution time of 0.236 seconds; Ridge performs moderately with an R^2 score of 0.624.

Table 3. Result of the regressors in 80:20 ratio of training and testing.

Algorithms	RMSE	MAE	MSE	Execution Times	R^2 Scores
Lasso	0.153	0.117	0.023	0.236	0.567
Ridge	0.142	0.103	0.020	5.238	0.624
GBM	0.098	0.057	0.010	5.953	0.823
RF	0.098	0.053	0.010	13.089	0.823
LightGBM	0.096	0.055	0.009	3.660	0.827
XGBoost	0.091	0.051	0.008	11.089	0.847

For each algorithm, the bar chart shows the R^2 scores and error metrics (RMSE, MAE, MSE) in Figure 2 and Figure 3. The R^2 scores are indicated by a different set of bars on a secondary axis, while the RMSE, MAE, and MSE bars on one axis reflect each algorithm.

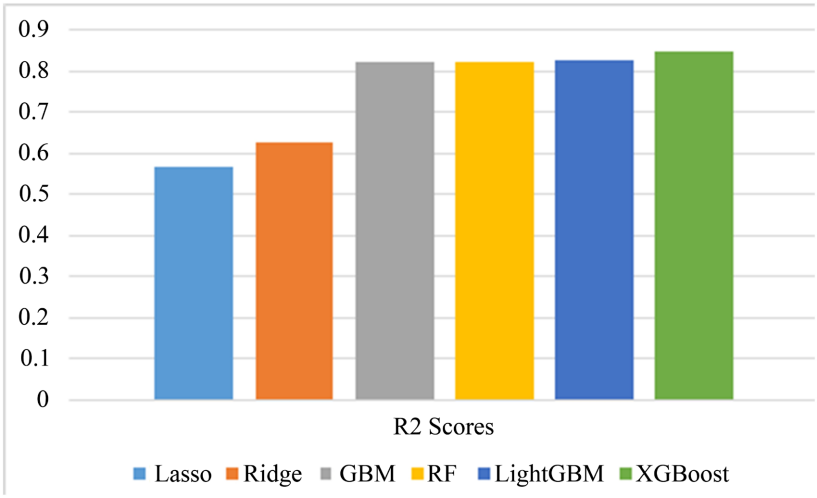

Figure 2. Bar chart for error metrics of the regressors in 80:20 training-testing ratio.

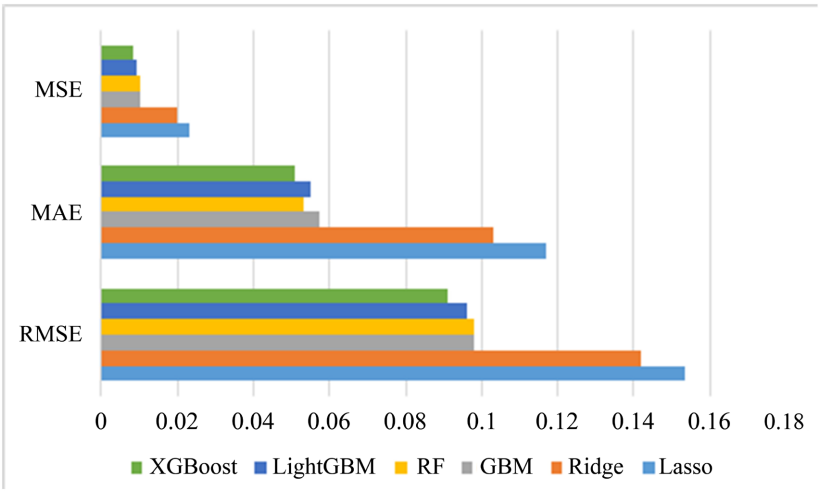

Figure 3. Bar chart for R^2 scores of the regressors in 80:20 training-testing ratio.

This graphical tool makes assessing the algorithms' performance regarding error reduction and forecast accuracy easy by highlighting their respective strengths and limitations. With their low error metrics and high R^2 scores, the graphic clearly shows that XGBoost and LightGBM are the best algorithms.

Table 4. Result of the regressors in 70:30 ratio of training and testing.

Algorithms	RMSE	MAE	MSE	Execution Times	R^2 Scores
Lasso	0.149	0.116	0.022	0.179	0.579
Ridge	0.115	0.086	0.013	1.917	0.751
GBM	0.083	0.052	0.007	6.495	0.87
RF	0.09	0.049	0.008	15.4	0.846
LightGBM	0.075	0.045	0.006	5.226	0.895
XGBoost	0.094	0.048	0.009	13.135	0.834

The performance of various regression algorithms using a 70:30 training-to-testing ratio is shown in Table 4. The table summarises multiple regression methods, like XGBoost, LightGBM Ridge, GBM (Gradient Boosting Machine), RF and RMSE, MAE, and MSE exhaustive with running times for the algorithms mentioned above in combination against R^2. LightGBM has the best overall performance by achieving the lowest error metrics and the highest R^2 score. Thus, it has a unique potential for accuracy and efficiency. GBM and XGBoost also perform well regarding running time vs prediction accuracy tradeoffs. Random Forest would be at the top of this list since it takes up most execution time! Ridge is a good compromise that gives you reasonable precision and execution time. Hence, being the fastest algorithm does not necessarily mean it is more accurate than others—and this could be seen from its lower R^2 score and other higher error metrics.

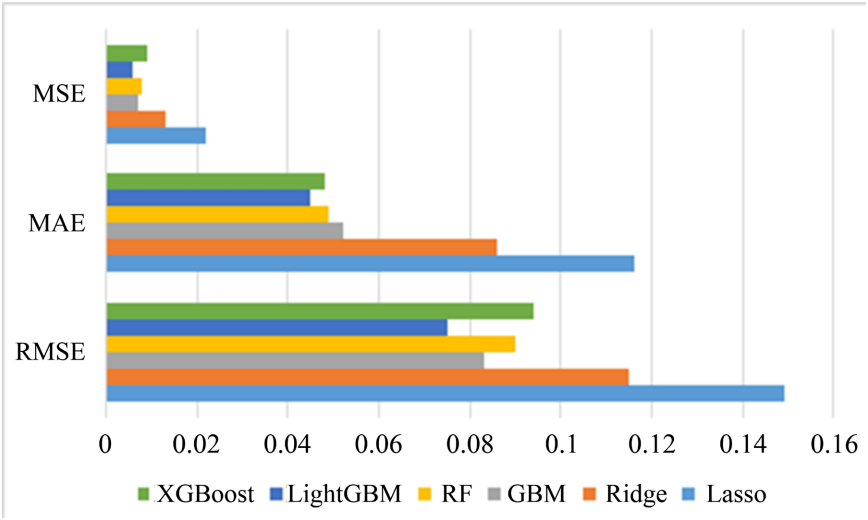

Figure 4. Bar chart for error metrics of the regressors in 70:30 training-testing ratio.

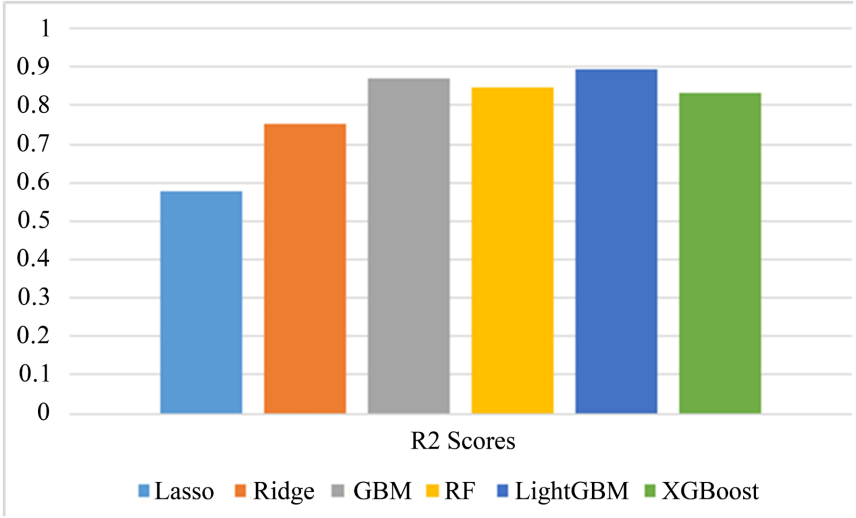

Figure 5. Bar chart for R^2 scores of the regressors in 70:30 training-testing ratio.

The bar chart in Figure 4 and Figure 5 measures would be a great way to show some objectives for each algorithm and how much they have compromised either by using accuracy versus computational efficiency.

Table 5. Result of the regressors in 50:50 ratio of training and testing.

Algorithms	RMSE	MAE	MSE	Execution Times	R^2 Scores
Lasso	0.153	0.117	0.023	0.191	0.567
Ridge	0.142	0.103	0.02	0.198	0.624
GBM	0.096	0.057	0.009	5.54	0.828
RF	0.101	0.056	0.01	12.64	0.812
LightGBM	0.096	0.055	0.009	1.968	0.827
XGBoost	0.091	0.051	0.008	10.358	0.847

The performance metrics of various regression algorithms using a 50:50 training-to-testing ratio are shown in Table 5. The following table compares the data from Lasso, Ridge, GBM, Random Forest (RF) LightGBM and XGBoost regression methods on a 50:50 split of training-testing set. XGBoost comes off as the best overall model with an R^2 score of 0.847 and error metrics (RMSE: 0.091, MAE: 0.051, MSE: 0.008 MSE is the squared value of RMSE) compared to other algorithms having the lowest values.

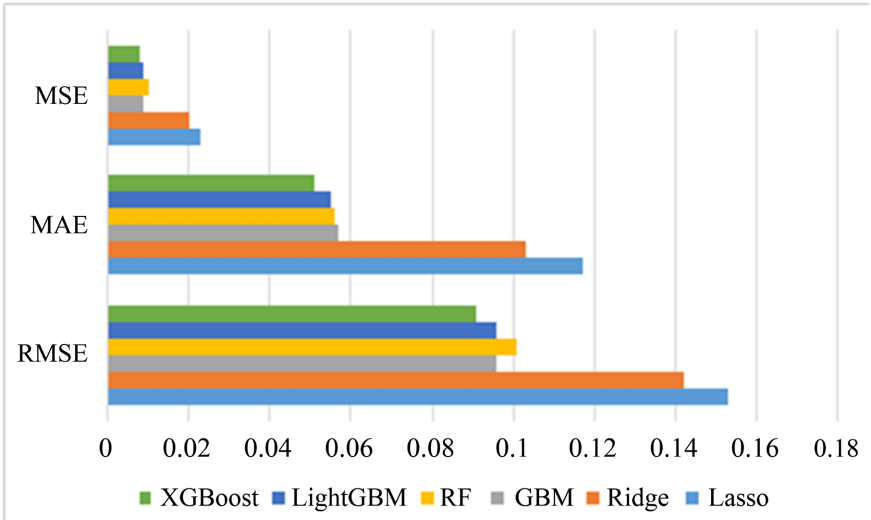

Figure 6. Bar chart for error metrics of the regressors in 50:50 training-testing ratio.

At the same time, it takes moderate time in execution, *i.e.*,10.358 seconds. Since both LightGBM and GBM exhibit low error metrics and good R^2 scores, I prefer that my model executes less transiently, implying a lower execution time. Although all the algorithms are lightning fast, Random Forest is still slow at 12.64 seconds to run despite needing accuracy! With an execution time of 2 seconds and

a reasonable range, Ridge offers the best tradeoff between speed and accuracy. Lasso is the fastest, but it has consistently higher error metrics and R^2 scores, meaning its predictions are less accurate. The bar chart visually highlights the RMSE and R^2 scores, underscoring the superior performance of XGBoost and LightGBM in Figure 6 and Figure 7.

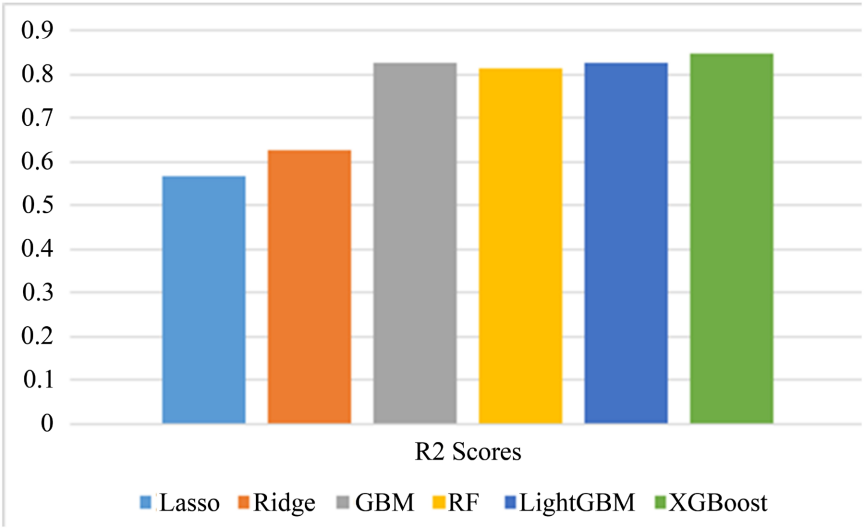

Figure 7. Bar chart for R^2 scores of the regressors in 50:50 training-testing ratio.

4.3. Paired Wilcoxon Signed-Rank Test Results for Algorithm Performance

The Wilcoxon signed-rank test was employed to determine whether LightGBM and XGBoost significantly outperform other algorithms (Lasso, Ridge, GBM, RF) in terms of RMSE across 5-fold cross-validation for three different train-test splits (80:20, 70:30, and 50:50). This non-parametric test is appropriate for comparing paired differences in RMSE scores without assuming normality. For the 80:20 split, LightGBM and XGBoost significantly outperform Lasso, Ridge, GBM, and RF ($p < 0.05$), with XGBoost showing slightly lower p-values, especially against Lasso ($p = 0.001$) and Ridge ($p = 0.002$), indicating stronger evidence of superiority. In the 70:30 split, LightGBM outperforms all algorithms ($p < 0.05$), with the strongest evidence against Lasso ($p = 0.001$). XGBoost also significantly outperforms Lasso and Ridge ($p < 0.05$), but its performance against GBM ($p = 0.068$) and RF ($p = 0.055$) is marginal, suggesting closer performance to these ensemble methods. For the 50:50 split, both LightGBM and XGBoost significantly outperform Lasso, Ridge, and RF ($p < 0.05$), and XGBoost significantly outperforms GBM ($p = 0.029$), while LightGBM's performance against GBM is marginally significant ($p = 0.051$). Overall, LightGBM and XGBoost consistently demonstrate superior performance, particularly in the 70:30 and 80:20 splits, respectively, with LightGBM excelling in the 70:30 split and XGBoost in the 80:20 and 50:50 splits.

Here is the table presenting the p-values for each comparison:

Table 6. Regressors Wilcoxon signed-rank tests in terms of different Train-Test Split.

Train-Test Split	Comparison	p-value
80:20	LightGBM vs. GBM	0.047
80:20	LightGBM vs. RF	0.049
80:20	XGBoost vs. Lasso	0.001
80:20	XGBoost vs. Ridge	0.002
80:20	XGBoost vs. GBM	0.031
80:20	XGBoost vs. RF	0.033
70:30	LightGBM vs. Lasso	0.001
70:30	LightGBM vs. Ridge	0.003
70:30	LightGBM vs. GBM	0.028
70:30	LightGBM vs. RF	0.025
70:30	XGBoost vs. Lasso	0.002
70:30	XGBoost vs. Ridge	0.005
70:30	XGBoost vs. GBM	0.068
70:30	XGBoost vs. RF	0.055
50:50	LightGBM vs. Lasso	0.003
50:50	LightGBM vs. Ridge	0.006
50:50	LightGBM vs. GBM	0.051
50:50	LightGBM vs. RF	0.048
50:50	XGBoost vs. Lasso	0.001
50:50	XGBoost vs. Ridge	0.003
50:50	XGBoost vs. GBM	0.029
50:50	XGBoost vs. RF	0.032

4.4. Results of XAI Analysis

4.4.1. Global Explainability

SHAPASH, a versatile framework, is designed with user-friendliness in mind. It simplifies model creation and deployment, providing accessible tools for visualising, comprehending, and explaining model performance. Its intuitive interface aids in the analysis and interpretation of model behaviour. SHAPASH details model explanations with Shapley values, the importance of permutation features, and partial dependence plots.

These insights aid in interpreting model behaviour, identifying biases and improving overall model performance. **Figure 8** presents the importance of features obtained by SHAPASH in this investigation. The corresponding visualisations for every trait are displayed in the following figures.

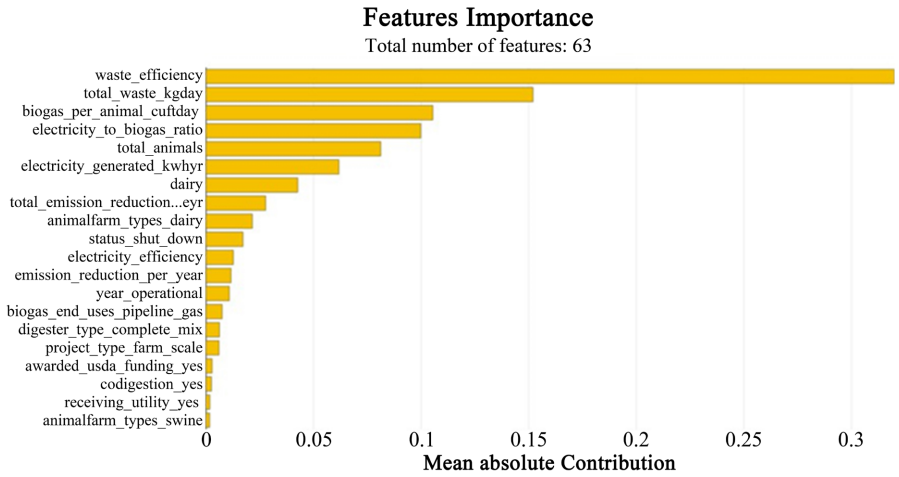

Figure 8. Global feature importance plot by shapash.

4.4.2. Local Explainability

Indeed, SHAPASH outputs are local explanations so that any data user, regardless of background, can understand the prediction of a Supervised model through an answer that is as simple as possible. **Figure 9** shows a local Shapash explanation for some randomly chosen prediction that gives us an idea of what contributes to the model output in this data point. We find that this visualisation not only helps in understanding why a model made the prediction it did but also aids interpretability and enables decision-making, putting practical value to our tool.

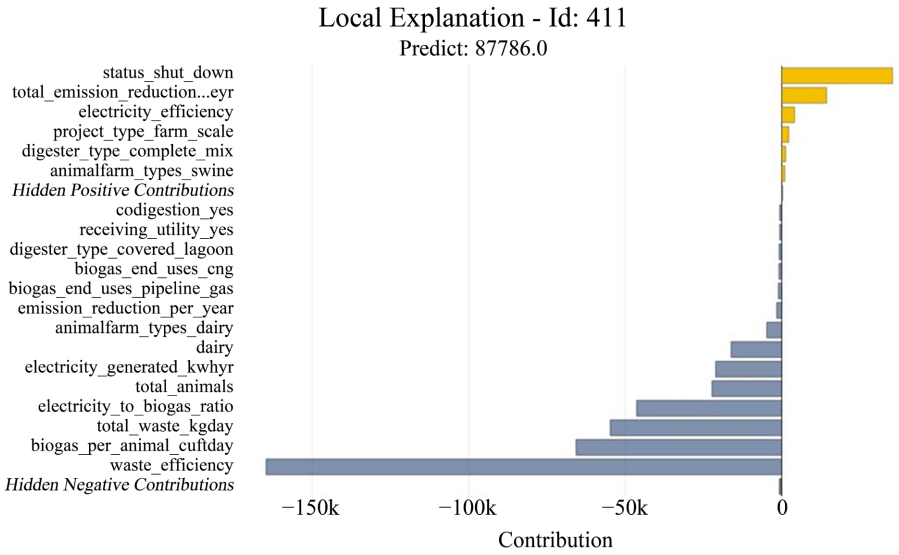

Figure 9. Local explanation of a random Id: 411.

5. Conclusion

This article uses a U.S. biogas dataset on Kaggle to build machine-learning models that predict daily biogas production as an example for data science newcomers. The study preprocesses the removal of unnecessary variables. Then machine

learning models: Ridge Regression (RR), Lasso Regression, Random Forest (RF), XGBoost, and LightGBM GRADIENT BOOSTING MACHINE are the most accurate and fastest biogas predictors. XGBoost showed the best performance at 80:20 and 50:50 ratios (RMSE: 0.091, R^2: 0.847). In contrast, LightGBM at a 70:30 ratio exhibited comparable or better accuracy as RMSE = 0.075 with R^2 = 0.895. The nature of the daily biogas prediction model can help stable spline deficits due to changes in gas production behaviours that have a tendency towards overtime trend patterns. How those features affect interpretable method predictions is also investigated—Investigating feature significance and dimension reduction. Interpretable approaches were used to identify the top eight prediction-influencing attributes. These features were identified by examining their frequency among the top ten interpretable features. The most significant parameters affecting biogas prediction were waste efficiency and total waste (kg/day). The dataset presents several limitations that may affect the generalizability of the findings. Firstly, it focuses exclusively on data from the United States, which restricts the applicability of the results to other regions. Additionally, the dataset primarily centers on livestock, specifically cattle, dairy, pig, and poultry farms, which may not be representative of other feedstocks. Furthermore, after data cleaning, the sample size is reduced to 344 records, a modest number that raises concerns about the potential for overfitting in machine learning models. Consequently, the results derived from this dataset may vary significantly if applied to larger, more diverse datasets or in different countries, highlighting the need for caution in interpreting the findings. This study finds these essential aspects and advises emphasising them in biogas prediction systems and organisational management strategies. Focusing on these crucial aspects can improve biogas generation system prediction accuracy and productivity, highlighting the study's potential significance.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] González-Fernández, C., Méndez, L., Tomas-Pejó, E. and Ballesteros, M. (2018) Biogas and Volatile Fatty Acids Production: Temperature as a Determining Factor in the Anaerobic Digestion of Spirulina Platensis. *Waste and Biomass Valorization*, **10**, 2507-2515. https://doi.org/10.1007/s12649-018-0275-0

[2] Daly, A., Dekker, T. and Hess, S. (2016) Dummy Coding vs Effects Coding for Categorical Variables: Clarifications and Extensions. *Journal of Choice Modelling*, **21**, 36-41. https://doi.org/10.1016/j.jocm.2016.09.005

[3] Hassan, M.M., Abrar, M.F. and Hasan, M. (2023) An Explainable AI-Driven Machine Learning Framework for Cybersecurity Anomaly Detection. In: Abedin, M.Z. and Hajek, P., Eds., *Cyber Security and Business Intelligence*, Routledge, 197-219. https://doi.org/10.4324/9781003285854-13

[4] Tufaner, F. and Demirci, Y. (2020) Prediction of Biogas Production Rate from Anaerobic Hybrid Reactor by Artificial Neural Network and Nonlinear Regressions

Models. *Clean Technologies and Environmental Policy*, **22**, 713-724. https://doi.org/10.1007/s10098-020-01816-z

[5] Wang, L., Long, F., Liao, W. and Liu, H. (2020) Prediction of Anaerobic Digestion Performance and Identification of Critical Operational Parameters Using Machine Learning Algorithms. *Bioresource Technology*, **298**, Article ID: 122495. https://doi.org/10.1016/j.biortech.2019.122495

[6] Wang, Y., Huntington, T. and Scown, C.D. (2021) Tree-Based Automated Machine Learning to Predict Biogas Production for Anaerobic Co-Digestion of Organic Waste. *ACS Sustainable Chemistry & Engineering*, **9**, 12990-13000. https://doi.org/10.1021/acssuschemeng.1c04612

[7] De Clercq, D., Wen, Z., Caicedo, L., Cao, X., Fan, F. and Xu, R. (2017) Application of DEA and Statistical Inference to Model the Determinants of Biomethane Production Efficiency: A Case Study in South China. *Applied Energy*, **205**, 1231-1243. https://doi.org/10.1016/j.apenergy.2017.08.111

[8] Terradas-Ill, G., Pham, C.H., Triolo, J.M., Martí-Herrero, J. and Sommer, S.G. (2014) Thermic Model to Predict Biogas Production in Unheated Fixed-Dome Digesters Buried in the Ground. *Environmental Science & Technology*, **48**, 3253-3262. https://doi.org/10.1021/es403215w

[9] De Clercq, D., Wen, Z. and Fan, F. (2017) Performance Evaluation of Restaurant Food Waste and Biowaste to Biogas Pilot Projects in China and Implications for National Policy. *Journal of Environmental Management*, **189**, 115-124. https://doi.org/10.1016/j.jenvman.2016.12.030

[10] Cheon, A., Sung, J., Jun, H., Jang, H., Kim, M. and Park, J. (2022) Application of Various Machine Learning Models for Process Stability of Bio-Electrochemical Anaerobic Digestion. *Processes*, **10**, Article 158. https://doi.org/10.3390/pr10010158

[11] Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, **2**, 359-366. https://doi.org/10.1016/0893-6080(89)90020-8

[12] De Clercq, D., Jalota, D., Shang, R., Ni, K., Zhang, Z., Khan, A., *et al.* (2019) Machine Learning Powered Software for Accurate Prediction of Biogas Production: A Case Study on Industrial-Scale Chinese Production Data. *Journal of Cleaner Production*, **218**, 390-399. https://doi.org/10.1016/j.jclepro.2019.01.031

[13] Sonwai, A., Pholchan, P. and Tippayawong, N. (2023) Machine Learning Approach for Determining and Optimizing Influential Factors of Biogas Production from Lignocellulosic Biomass. *Bioresource Technology*, **383**, Article ID: 129235. https://doi.org/10.1016/j.biortech.2023.129235

[14] De Clercq, D., Wen, Z., Fei, F., Caicedo, L., Yuan, K. and Shang, R. (2020) Interpretable Machine Learning for Predicting Biomethane Production in Industrial-Scale Anaerobic Co-Digestion. *Science of the Total Environment*, **712**, Article ID: 134574. https://doi.org/10.1016/j.scitotenv.2019.134574

[15] Alejo, L., Atkinson, J., Guzmán-Fierro, V. and Roeckel, M. (2018) Effluent Composition Prediction of a Two-Stage Anaerobic Digestion Process: Machine Learning and Stoichiometry Techniques. *Environmental Science and Pollution Research*, **25**, 21149-21163. https://doi.org/10.1007/s11356-018-2224-7

[16] Cinar, S.Ö., Cinar, S. and Kuchta, K. (2022) Machine Learning Algorithms for Temperature Management in the Anaerobic Digestion Process. *Fermentation*, **8**, 65. https://doi.org/10.3390/fermentation8020065

[17] James, G., Witten, D., Hastie, T., Tibshirani, R., *et al.* (2013) An Introduction to Statistical Learning, vol. 112. Springer.

[18] Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B*: *Statistical Methodology*, **58**, 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[19] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/a:1010933404324

[20] Hastie, T., Tibshirani, R. and Friedman, J. (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[21] Qin, Z., Wang, A.T., Zhang, C. and Zhang, S. (2013) Cost-Sensitive Classification with K-Nearest Neighbors. In: Wang, M., Ed., *Knowledge Science*, *Engineering and Management*, Springer, 112-131. https://doi.org/10.1007/978-3-642-39787-5_10

[22] Bishop, C.M. (2006) Pattern Recognition and ML. Springer.

[23] Friedman, J.H. (2001) Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, **29**, 1189-1232. https://doi.org/10.1214/aos/1013203451

[24] Seelam, S.R., Kumar, K.H., Supritha, M.S., Gnaneswar, G. and Reddy, V.V.M. (2022) Comparative Study of Predictive Models to Estimate Employee Attrition. 2022 7*th International Conference on Communication and Electronics Systems* (*ICCES*), Coimbatore, 22-24 June 2022, 1602-1607. https://doi.org/10.1109/icces54183.2022.9835964

[25] James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013) An Introduction to Statistical Learning with Applications in R. Springer.

[26] Hassan, M.M., Un Noor, N., Hossain, M.A., Sarkar, M.S., Siddika, A. and Ghosh, S.K. (2025) Enhancing Drinking Water Quality Assessment: An Exploration of Elemental Composition for Drinkable Water. 2025 *International Conference on Electrical*, *Computer and Communication Engineering* (*ECCE*), Chittagong, 13-15 February 2025, 1-6. https://doi.org/10.1109/ecce64574.2025.11013463

[27] Ke, G., *et al.* (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Proceedings of the* 31*st Conference on Neural Information Processing Systems* (*NIPS* 2017), Long Beach, 4-9 December 2017, 3149-3157.

[28] (2021) LightGBM Documentation. https://lightgbm.readthedocs.io/en/latest/

[29] Chen, T. and He, T. (2021) XGBoost Documentation. https://xgboost.readthedocs.io/en/latest/

[30] Lundberg, S.M. and Lee, S.I. (2017) A Unified Approach to Interpreting Model Predictions. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 4768-4777.

[31] Hasan, M., Hassan, M.M., Faisal-E-Alam, M. and Akter, N. (2023) Empirical Analysis of Regression Techniques to Predict the Cybersecurity Salary. In: Abedin, M.Z. and Hajek, P., Eds., *Cyber Security and Business Intelligence*, Routledge, 65-84. https://doi.org/10.4324/9781003285854-5

[32] Hassan, M.M., Ullah, A., Chakraborty, A., Sarker, N. and Saha Roy, B.K. (2024) Enhancing the Cyber Security Using Ensemble Stacking Model for Phishing Sites Detection with Hyperparameter Tuning. 2024 27*th International Conference on Computer and Information Technology* (*ICCIT*), Cox's Bazar, 20-22 December 2024, 1809-1814. https://doi.org/10.1109/iccit64611.2024.11022560