Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Enhancing Machine Learning models of biogas production in wastewater treatment plants using hybrid Bayesian optimization based on Gaussian process

Hadjer SADOUNE

hadjersadoune5@gmail.com

Draria Nuclear Reasearch Centre

Rachida RIHANI

Laboratoire Phénomènes de Transfert, Université des Sciences et de la Technologie Houari Boumediene

Francesco Saverio MARRA

Istituto di Scienze e Tecnologie per l'Energia e la Mobilità Sostenibili - CNR

Research Article

Keywords: Industrial wastewater discharge, Large-scale anaerobic digester, DNN, XGBoost, Gaussian Processes Hyperparameter

Posted Date: June 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9917351/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Enhancing Machine Learning models of biogas production in wastewater treatment plants using hybrid Bayesian optimization based on Gaussian process

Hadjer SADOUNE [a,b] ; Rachida RIHANI [b] ; Francesco Saverio MARRA [c]

[a] *Draria Nuclear Reasearch Centre, Draria,, Algeria.*

[b] *Laboratoire Phénomènes de Transfert, Université des Sciences et de la Technologie Houari Boumediene, U.S.T.H.B, BP 32 El-Alia, Bab-Ezzouar, Algeria.*

[c] *Istituto di Scienze e Tecnologie per l'Energia e la Mobilità Sostenibili - CNR, viale G. Marconi 4, Naples, 80125, Italy*

H I G H L I G H T S

- Comparative evaluation of DNN and XGBoost models for predicting biogas production in industrial wastewater treatment systems.
- Implementation of BO-GP for robust and efficient hyperparameter tuning.
- Identification of the learning rate as the most critical hyperparameter influencing the predictive performance of the DNN.
- Efficient exploration of hyperparameter space demonstrated through convergence analysis.
- The DNN model outperformed XGBoost with superior predictive accuracy ($R^2 = 0.7809$).

Abstract

This study aims to compare machine learning (ML) models, particularly Deep Neural Networks (DNN) and Extreme Gradient Boosting (XGBoost), to predict biogas production in an industrial wastewater treatment plant. A Bayesian Optimization approach using Gaussian Processes (BO-GP) was selected to optimize the hyperparameters of these models. The daily operational data from July 2019 to January 2022 were used to train and evaluate the models.

The DNN model outperformed XGBoost, achieving R-squared, MAE and RMSE values of 0.7809, 148.0927 and 198.7302, respectively, compared with the values of 0.73, 163.2680 and 216.2166 obtained with XGBoost. The BO-GP approach effectively identified the optimal hyperparameters for both models, with the DNN utilizing a simple architecture of two hidden layers with 91 neurons each. Convergence analysis demonstrated that the optimization process efficiently explored the hyperparameter space, with the DNN showing more significant improvements. For the DNN, learning rate was found to be the most influential hyperparameter. This study demonstrates the potential of machine learning approaches, particularly DNNs optimized using BO-GP, for modeling biogas formation amid anaerobic digestion processes. The findings contribute to applying advanced data-driven techniques in wastewater treatment and renewable energy production.

Key-words: Industrial wastewater discharge, Large-scale anaerobic digester, DNN, XGBoost, Gaussian Processes Hyperparameter.

1. Introduction

Wastewater treatment facilities (WWTPs) have a crucial role in maintaining public health and environmental sustainability. However, the energy-intensive nature of these processes poses significant challenges for plant managers seeking to optimize operations while minimizing costs. The increasing complexity of influent wastewater elements and the treatment steps inside WWTPs further complicates this task.

In recent years, Machine Learning (ML) has appeared as a promising technique for predicting and optimizing various aspects of WWTP operation. By leveraging the vast amount of data generated by real-time sensors in full-scale plants, ML algorithms can identify patterns and relationships that inform more effective control strategies. This approach offers significant potential for energy saving, improved treatment efficiency, and enhanced overall plant performance.

To date, most studies have focused on developing specific ML algorithms or evaluating their performances. However, there is a growing need for more comprehensive assessments that integrate ML with real-world WWTP operations and that consider the complexities inherent in these systems.

A primary consideration pertains to the appropriate preprocessing of the data collected from the plant, which involves a number of essential steps to maintain a good quality of the data and its relevance to modelling. Oliveira et al. [1], have clearly shown the efficacy of implementing data preprocessing using Principal Components Analysis (PCA) prior to its input into a backpropagated neural network. In a previous study Sadoune et al. [2], the authors of the present study addressed the challenges associated with utilizing a dataset derived from a real-world scenario.

Advancements in the ML field are occurring at a rapid pace, and new algorithms and tools emerge frequently, presenting the challenge of evaluating the most reliable approaches for the problem at hand. Indeed, the adoption and comparison of numerous ML approaches to simulate water treatment plants is documented in the literature.

In a study by Mjalli et al. [3], researchers created an Artificial Neural Network (ANN), specifically a Feed-Forward Neural Network (FFNN), to simulate the operations of the Doha West wastewater treatment facility in Qatar. The researchers utilized raw sewage quality parameters (BOD, COD, and TSS) as inputs to forecast the BOD, COD, and TSS levels in the effluent stream.

Rustum [4] conducted a comparative analysis of three techniques: the Kohonen Self Organising Map (KSOM), backpropagation artificial neural networks (BPANN), and adaptive network based fuzzy inference system (ANFIS). The study also examined and tested the feasibility of creating hybrid models by combining these approaches.

Pai et al. [5], investigated the efficacy of three distinct adaptive neuro fuzzy inference system (ANFIS) models in predicting effluent characteristics from an industrial park's wastewater treatment facility. The researchers focused on forecasting suspended solids, chemical oxygen demand, and pH levels. To provide a benchmark for comparison, the study also utilized an artificial neural network model. The research findings demonstrated that ANFIS exhibited statistically significant superiority over ANN of effluent modelling preciseness.

Guo et al. [6], evaluated the performance of Artificial Neural Networks (ANNs) and Support Vector Machine (SVM) in modeling Wastewater Treatment facilities to forecast the removal efficiencies of BOD5, COD, TSS, NH4-N, and TP. The SVM model exhibited greater predictive accuracy during the training phase and comparable results in the validation phase. Nevertheless,

sensitivity analysis revealed that the ANN technique was more effective for predicting T-N concentration at 1-day intervals, particularly in the case of establishing cause-effect relationships between T-N concentration and input variables for combined food waste and wastewater treatment.

De Clercq et al. [7], considered the application of three machine learning algorithms - elasticnet, random forest, and extreme gradient boosting - to model the evolution of biomethane in an industrial-scale anaerobic codigestion (ACoD) plant. The researchers utilized four years of operational data to train these models, which were then used to forecast biomethane output for various time horizons ranging from 1 to 40 days. The findings demonstrated that the random forest and extreme gradient boosting (**XGBoost**) techniques, which do not assume linear relationships, substantially outperformed the elasticnet model, which operates under linear assumptions.

Some researchers [8,9], compared four distinct approaches to predict the effluent biological oxygen demand, chemical oxygen demand and total nitrogen. These included three artificial intelligence-based non-linear models, i.e. feed forward neural network (FFNN), adaptive neuro fuzzy inference system (ANFIS), and support vector machine (SVM). Additionally, a traditional multi-linear regression (MLR) method was utilized. To enhance the effectiveness of the AI-based models, Neural Network Ensemble (NNE) techniques were implemented.

To predict BOD5 and COD, Aghdam et al. [10], have developed several machine learning models, including gene expression programming (GEP), multilayer perception neural networks, multi-linear regression, k-nearest neighbors, gradient boosting, and regression trees. These models were based on monthly data collected from the incoming wastewater of 7 treatment facilities in Hong Kong over a three-year period. Among these models, GEP demonstrated superior performance in terms of prediction accuracy, as evidenced by multiple statistical metrics.

The analysis of the literature underscores the ongoing necessity to revise previously developed approaches in order to assess the potential advantages of newer methodologies.

An additional critical consideration is the optimization of specific architectures and hyperparameters for Artificial Neural Networks (ANNs) applied to wastewater treatment plants [11]. Indeed, the implementation of complex models on smaller datasets increases the likelihood of overfitting. Simple neural network models better balance between bias and variance, hence avoiding from memorizing the training data and failing to generalize to new, unseen data. It has been noted that the efficacy of complex systems was not enhanced with the addition of concealed layers in deep networks [12].

Previous studies have shown good performance achieved by various types of artificial neural networks with a simple architecture to predict anaerobic digestion processes [13, 14, 15].

Sakiewicz et al. [16], developed a predictive model using artificial neural networks that integrate the combined effects of seven main adjustable operational parameters and five wastewater quality characteristics to optimize biogas production in an anaerobic fermentation process integrated with wastewater treatment. Their results indicated that an ANN architecture with two hidden layers effectively captured the complex nonlinear dependencies present in an AD data system. Ghiat et al. [17], examined the utilization of two machine learning models, DNN and **XGBoost**, to forecast the overall energy usage in a closed agricultural greenhouse to improve microclimate control and energy efficiency. They implemented GridSearchCV as a hyperparameter tuning tool for model development, resulting in a 2-layer DNN model with 20 neurons in each layer, a batch size of 5 and 100 epochs. The optimal hyperparameters **for XGBoost** were found to be a learning rate of 0.1, 600 turns, a subsample size of 0.5, a highest depth of 12, and a lowest child weight of 3.

In a study by Alruqi et al. [18], a ML approach employing Gaussian process regression was applied to forecast biomethane yield from a combination of organic waste and sugarcane vinasse. The training hyperparameters were optimized using a Bayesian approach to develop a robust autoregressive model. The R^2 of the model increased by 0.72% during the test phase, thereby preventing overtraining. Clercq et al. [19], compared ML models from industrial production data provided by two AD plants in southern China, Hainan & Shenzhen, using different residue inputs. They trained models, including logistic regression, k-nearest neighbors, random forest, perceptron, and random forest, to predict biogas production. The kNN model showed the best performance with an out-of-sample R^2 value of 0.87 [19].

From the analysis of the literature, the XGBoost method appears to outperform many other ML approaches. Nevertheless, a systematic comparison, made by consistently optimizing the hyperparameters of different approaches, and confronting the performances on the same dataset, has never been presented. The present investigation aims to contribute to the endeavor of enhancing wastewater treatment plants (WWTPs) by examining the potential of the more recent extreme gradient boosting (XGBoost) method with respect to the equally promising deep neural network (DNN). Both models involve a large number of hyperparameters, which poses a real challenge in identifying the optimal combination to improve and optimise their performance. To make consistent the comparison, the study incorporates Bayesian optimisation based on Gaussian process (BO-GP) as a method for selecting the best hyperparameters of both models. Unlike manual or grid search methods, BO offers a more efficient and less computationally demanding approach, enabling faster convergence towards optimal solutions. This approach can lead to models with superior efficiency and reliability, thus contributing to the overall improvement of treatment plant performance. The outline of the article is as follows. The utilized dataset and a description of its

originating plant are first presented. Subsequently, the employed pre-processing methodology is delineated. The investigated machine learning (ML) models are introduced, with emphasis on their respective hyperparameters, followed by an explanation of the quality indices used for performance assessment. The subsequent section addresses the optimization process. The results of implementing the entire process for constructing, optimizing, and applying the two selected ML approaches are then presented and analyzed. Finally, the principal findings of the investigation are reported.

2. Materials and Methods

2.1. Data description and digester design

This study relies on the modelling of biogas produced from an industrial wastewater treatment plant (WWTP) for soft drinks and breweries located in Algiers, Algeria. Daily data were collected, covering the period from July 10, 2019, to January 15, 2022. To produce biogas and efficiently remove organic pollutants in a space-saving and cost-effective manner, this plant performs anaerobic treatment of industrial wastewater using an Upflow Anaerobic Sludge Blanket (UASB) digester. It has a working volume of 1,574 m^3 and a total volume of 1,914 m^3. The internal diameter of the digester is 19 m, and its height is 7.14 m. This digester is designed to treat an average of 9620 kg/d of Chemical Oxygen Demand (COD) and approximately 6.1 kg COD/m^3 per day in terms of organic load rate. The UASB digester is an anaerobic wastewater treatment facility commonly used to treat high-strength organic wastewater [20, 21]. It operates in an upflow mode; in this case, the influent is supplied at the bottom of the digester, where there is a dense mixed population of bacteria, and then flows upwards through a sludge bed. The diameter of the granulated sludge ranges from 0.5 to 2 mm making the sludge self-sustaining because it remains suspended inside the reactor. COD removal efficiency is highly related to the availability of

granulated sludge [22]. A blanket zone is created above the sludge bed, where certain biomasses are suspended. When the wastewater plant operates in upflow mode, dissolved organic pollutants are degraded by microbial activity. The sludge traps and retains suspended solids and biomass, thus leading to efficient digestion. Notably, UASB digesters are more attractive in tropical regions because of their high digestion rate in the mesophilic mode. The detailed scheme of the full-scale digester and data preparation procedure adopted in this study have been described previously [2]. Given the significant amount of missing data in the raw datasets, it was crucial to handle them with care to avoid the loss of relevant information. After data preparation, the dataset included 17 columns and 388 rows. One column represents the target feature, daily biogas production, whereas 16 columns represent the input features, such as flow influent (m^3/d), Total Chemical Oxygen Demand, CODt influent (mg/L), Total Suspended Solids, TSS influent (mg/L), pH inline influent, temperature, T inline influent, pH lab influent, T lab influent, Sludge Volume 30, SV30 influent (ml/L), COD load (kg/d), TSS load (kgTSS/d), pH lab effluent (mg/L), T lab effluent, TSS effluent (mg/l), CODt effluent (mg/L), SV30 effluent, and COD removal. These data were selected to predict daily biogas production using machine learning models.

2.2 Machine learning types

Machine learning algorithms can be divided into three main types (Figure 1): supervised, unsupervised, and reinforcement learning [23].

Supervised learning algorithms use labelled training data to assist models in making accurate predictions. The essential goal of this process is to create a correlation between the input values (x) and the corresponding output values (y) by using labelled input-output pairs during the training phase to predict the correct output for never-seen input values. Supervised learning is widely used in machine-learning applications. Typically, this task entails both classification and

regression processes [24]. When the output y, also known as the response variable or target, is categorical or nominal nature, the process falls in theis class of pattern recognition or classification [25]. When y assumes a continuous nature or a real value, the process is often referred to as a regression one. Input x can range in complexity from simple vectors of numbers to more complex and organized entities such as images, sentences, texts, time-series data, etc. [25]. Input data points x ($x \in X$) can have different scale levels. The latter's include nominal (lacking order); ordinal (processing order but no distance), and cardinal (featuring both order and distances). Nominal and ordinal data are mostly discrete, and cardinal data are continuous [11].

The data provided to unsupervised learning models lacks any accompanying labels that can be used to rectify the algorithm. The goal of this approach is to identify 'interesting patterns' in data [25]. Consequently, it is designed to efficiently acquire knowledge of the patterns present in the data to find analogies between the inputs in such a way that inputs that possess common features are classified together [26]. Unsupervised learning problems may include discovering groups of similar examples within the data, known as clustering, determining data distribution within the input space, known as density estimation, or projecting data from an n-dimensional space for visualization [27]. Reinforcement learning is a learning paradigm in which an agent operates within an environment and attempts to learn to operate effectively when receiving feedback. It involves the process of learning ways to effectively map different situations into appropriate actions to optimise a numerical reward signal. When the learner is unsure which actions to take, actions yielding the most reward via a series of trials must be found [28]. Reinforcement learning algorithms interact with an environment; therefore, there is a feedback loop between the learning system and its trials [29].

The purpose of this study is to evaluate the potential of using an ML approach to predict biomethane daily volume in a real wastewater treatment plant. The regression phase has been considered, and the dataset contains a subset of training data that consists of a set of input features x_i to which target values y_i are assigned. These target values reflect the expected output that the ML model should generate. The i index indicates the data record indexed from 1 to the upper limit N of experimental observations. The attainment of an accurate model is determined by its performance on previously unseen samples (test data) during the testing phase.

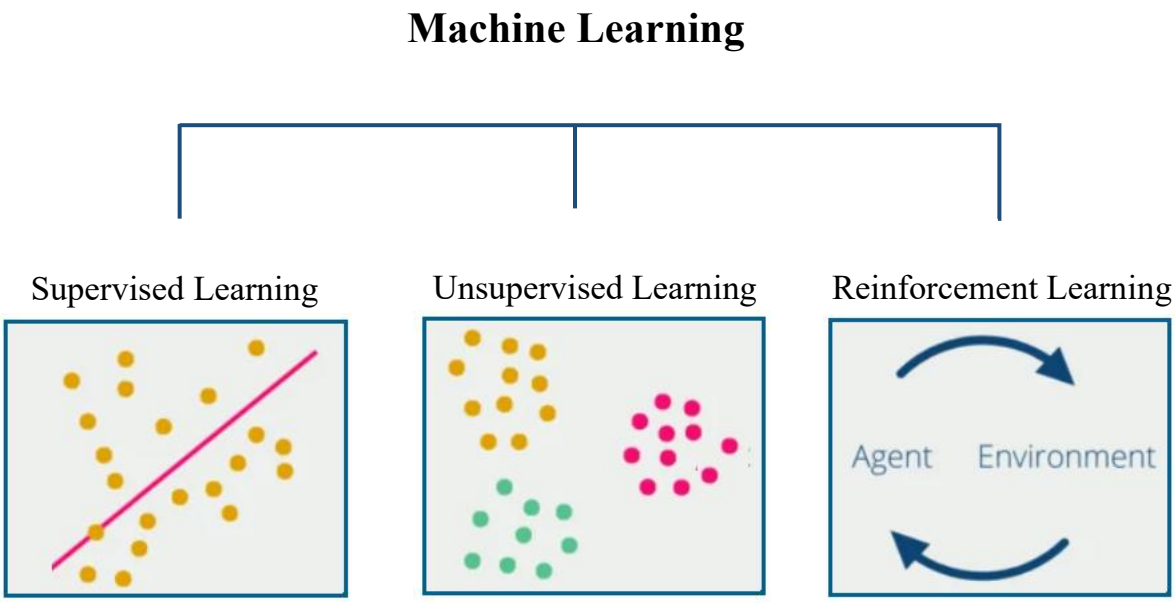


Fig. 1. Three primary categories of Machine Learning: Unsupervised, Supervised, and Reinforcement Learning

2.3 Models evaluation

Performance metrics are essential for evaluating the effectiveness of machine learning models. They evaluate the quality and performance of the model and assess whether the model fits well with the provided data. The choice of metrics depends on the specific task (classification,

regression, clustering, etc.) and desired output. Each category has its own specific set of metrics that are used to assess the efficacy of the model. The regression models generate a continuous output. Therefore, a metric based on a distance between the predicted and actual values is required.

In this study, three model metrics are used to assess the quality and performance of the selected models. These metrics include the coefficient of determination, R^2 (Eq. 2), the root-mean-square error RMSE (Eq. 3), and the mean absolute error MAE (Eq. 4).

$$R^2 = 1 - \frac{\sum(a_i - p_i)^2}{(a_i - \mu_a)^2} \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2} \quad (3)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a_i - p_i| \quad (4)$$

where

a_i : Actual value

p_i : Predicted value

μ_a : The mean value

2.4 Extreme Gradient Boosting (XGBoost) algorithm

The Extreme Gradient Boosting (XGBoost) algorithm is an ensemble learning approach used to train multiple models and generate a unique optimal output [30]. It is a decision-tree-based machine learning algorithm that can be considered as a modified version of the Gradient Boosting (GB); both algorithms share similar principles [31]. In GB, a gradient descent technique is employed to minimize the loss function while gradually incorporating additional models sequentially until further improvements cannot be achieved. In XGBoost, the decision trees are

formed in parallel, which allows XGBoost to effectively address the challenges of both overfitting and underfitting compared to GB [31]. XGBoost algorithm exhibits strong performance when applied to structured or tabular datasets for classification and regression prediction tasks [32]. The use of different optimization techniques enables the effective handling of large and complex datasets. It demonstrates its effectiveness in various applications [33], owing to its several advantages, including its ability to handle missing values, data scaling requirement, and a computationally effective implementation [34, 35]. To fit the training dataset, the XGBoost algorithm first generates an initial prediction. A decision tree is built using the residuals and a similarity score of the residuals is computed. Then, the assessment of the gain in similarity resulting from consecutive splits is performed. The gains are compared to establish both a feature and a threshold for a certain node. The output value calculation for each leaf is likewise based on the residuals. The resulting output of the tree is subsequently used as the residual update for the dataset, thereby building another tree. The above procedure is iteratively performed until the residuals cease to decrease or until a predetermined number of iterations has been reached. In contrast to the uniform weighting approach employed by Random Forest approaches [36], each successive tree in the given framework acquires knowledge from its predecessors, while being subjected to varying degrees of importance. The XGBoost algorithm was thoroughly described by [37].

2.4.1 XGBoost hyperparameters

The XGBoost algorithm offers a wide range of hyperparameters. Each hyperparameter affects the architecture and training process of the model by regulating a particular component of the model. This influence affects the complexity, generalizability, and susceptibility of the model to overfitting. The identification of the most suitable hyperparameter configuration is crucial to

attain the optimal performance of the XGBoost model when applied to a specific dataset. Therefore, these hyperparameters must be carefully adjusted.

This algorithm builds a series of boosting cycles or trees sequentially, whose number is determined by the "n_estimators" hyperparameter. Each subsequent tree attempts to correct the errors made by previous trees. Although a larger number of trees may enable the detection of complex patterns in the data, it also introduces the possibility of overfitting, if not adequately managed. The boosting algorithm used is determined by the booster parameter. This decision influences the complexity of the model and its ability to detect nonlinear relationships in data. During the boosting phase, max_depth hyperparameter controls the maximum depth of the trees in the model. This parameter can also be used to identify deeper patterns in the training data. The learning rate hyperparameter determines the magnitude of weight adjustments performed by the optimizer. A lower value may provide a more accurate result but slower updates, whereas a higher value leads to faster updates but a less accurate result. The parameter gamma serves to regulate the minimum loss reduction required to make additional divisions on the leaf nodes of the trees. It functions as a regularization mechanism to prevent overfitting. The alpha and lambda hyperparameters are used to apply the L1 and L2 regularization terms to the weights, respectively. These terms penalize high-weight values to mitigate overfitting and enhance the smoothness of the models. Subsample, colsample_bytree, colsample_bylevel, and colsample_bynode hyperparameters introduce randomness into the training process by controlling the fraction of samples or features used for building each tree. Prudent fine-tuning of these hyperparameters enables a model to achieve sufficient generalization, while avoiding both overfitting and underfitting.

Regarding the XGBoost implementation, a variety of hyperparameters have been tuned to enhance the performance of the model and fully exploit its potential. Seven hyperparameters determine the search space explored: n_estimators, with values ranging from 50 to 500; max_depth, varying from 1 to 10; the booster hyperparameter, with two options considered, namely gbtree and dart; the learning rate, spanning from 0.01 to 0.5; gamma, ranging from 1 to 20; lambda, with values from 1 to 50; and subsample, with a range between 0.5 and 0.9.

2.4 DNN algorithm

The neural network hyperparameter configuration is described as follows. The same number of features is taken to construct the input layer (16 input neurons) because of their substantial impact on the anaerobic digestion process. The activation function and number of layers and neurons are the hyperparameters that are adjusted in the dense layer. The dense layer has been defined as having one to five hidden layers. Three hyperparameters, the number of neurons (ranging from 10 to 100), dropout rate (between 0.1 and 0.9), and activation function (elu and selu), have been examined for each hidden layer. One neuron in the output layer is responsible for predicting the biogas production. The RMsprop and Adam are studied as optimizers' hyperparameters, with the learning rate option varying between 10^{-2} and 10^{-6}. DNN models were trained for 200 epochs. To address the potential issue of overfitting, an early stopping technique was employed utilizing a patience value of 10 to stop the training process.

2.5. Gaussian Process Approach for Hyperparameter Optimization

Gaussian Process (GP) is a sophisticated nonparametric approach for approximating complex regression models [38], which can be used for functional spaces containing continuous,

discrete, or mixed domains. It is formally defined as a collection of random variables x, where any finite subset $(x_1, x_2, \ldots, x_n) \in x$, follows a joint Gaussian distribution [39, 40]. However, GP is a Gaussian distribution over function that is defined by the mean function $\mu: x \to \mathbb{R}$ and the kernel or covariance function $k: x * x \to \mathbb{R}$.

The mean function represents the expected value of the function at any point x in the input space $\mathbb{R}$. It is typically assumed to be zero ($\mu(x) = 0$) for the sake of simplicity, although it can also be selected based on prior knowledge or assumptions about the data. The covariance function (also referred to as the kernel function) $k(x, x')$ determines the covariance between the function values at two given points x and x' in the input space. It conveys the concept of resemblance among the input points. Popular options for covariance functions include the Gaussian kernel, squared exponential kernel, Matérn kernel, etc.

The Gaussian Process is often defined as follows:

$$f(x) \sim GP(\mu(x), k(x, x'))$$

Where

The mean function $\mu(x)$ is,

$$\mu(x) = E[f(x)]$$

The covariance function k is defined as:

$$k(x, x') = E[f(x) - \mu(x))\ (f(x') - \mu(x'))]$$

E being the statistical expectation. GP is used as a surrogate model, with the Matérn kernel serving as the covariance function, which is given by

$$k_{Matérn}(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|x - x'\|}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}\|x - x'\|}{l}\right)$$

where $\|x - x'\|$ is the Euclidean distance between the input vectors x and x'. K_{ν} is the modified Bessel function of the second kind, and Γ represents the gamma function.

The parameters σ^2, ν, and l are strictly positive scalars. σ^2 represents the covariance magnitude, while ν is the smoothness parameter that governs the mean-square differentiability of the resulting functions. Consequently, ν affects the speed at which the derivatives of the functions change in the input space. l is the length scale parameter. It controls the measurement of covariance between the points in the input space.

2.5.1 Optimizing EI via the GP algorithm

The expected improvement (EI) in the GP represents a trade-off between exploration and exploitation. It efficiently explores the search space by balancing exploration, which involves selecting points with high uncertainty, and exploitation, which involves selecting points with high potential for improvement. Consider $D=\{(x^{(i)},y^{(i)}), i=1,2,\ldots,k\}$ as the observation set, where $y^{(i)}$ is the observed value of the target variable at location $x^{(i)}$ in the parameter space [39]. For a Gaussian process with mean $\mu(x)$ and variance $\sigma^2(x)$, the probability of improvement follows from the normal density function [41]:

$$\mathrm{P}(\mathrm{f}(x^*)|D) = \frac{1}{\sigma(x^*)\sqrt{2\pi}} exp\left(-\frac{(f(x^*) - \mu(x^*))^2}{2\sigma^2(x^*)}\right)$$

Let:

$$\mathrm{I}(x^*) = \max(0, f(x^*) - f_{best})]$$

The EI function is defined as:

$$\mathrm{EI}(x^*) = E[\mathrm{I(x)}]$$

$$\mathrm{EI}(x^*) = E[\max(0, f(x^*) - f_{best})]$$

Mathematically, $\mathrm{EI}(x^*)$ can be expressed as follows:

$$\mathrm{EI}(x^*) = \int_{-\infty}^{\infty} \max(0, f(x^*) - f_{best}) \cdot P(f(x^* | D)\, df(x^*)$$

where D is the observed input-output data used for Gaussian process training. Then, it results:

$$\mathrm{EI}(x^*) = \int_{f_{best}}^{+\infty} (f(x^*) - f_{best}) \cdot P(f(x^*)|D)\, df(x^*)$$

This integral admits a closed-form expression:

$$EI(x^*) = (\mu(x^*) - f_{best}) \cdot \phi\left(\frac{\mu(x^*) - f_{best}}{\sigma(x^*)}\right) + \sigma(x^*) \cdot \phi\left(\frac{\mu(x^*) - f_{best}}{\sigma(x^*)}\right)$$

which, let:

$$Z = \frac{\mu(x^*) - f_{best}}{\sigma(x^*)}$$

is written as,

$$\mathrm{EI}(x^*) = \begin{cases} (\mu(x^*) - f_{best}) \,.\, \phi(Z) + \sigma(x^*) \,.\, \phi(Z) & \text{if } \sigma(x^*) > 0 \\ 0 & \text{if } \sigma(x^*) = 0 \end{cases}$$

In this expression:

$\mathrm{P}(x^* | D)$: the predictive distribution of the GP at point x^* given the observed data.

ϕ : the cumulative distribution function (CDF) of the standard normal distribution.

ϕ : the probability density function (PDF) of the standard normal distribution.

f_{best} : the highest function value that has been observed in the data

$\sigma(x^*)$: the standard deviation (uncertainty) of the GP prediction at point x.

$\mu(x^*)$: the mean prediction of the GP at point x.

The role of function EI can be explained in the following way.

If $\sigma(x^*) > 0$, meaning there is uncertainty in the prediction, EI measures the expected improvement in the objective function value at point x. It combines the potential improvement $(\mu(x^*) - f_{best})$ with the uncertainty $\sigma(x^*)$ of the prediction.

If $\sigma(x^*) = 0$, indicating that the uncertainty is zero since there is no uncertainty in the prediction and no potential for improvement.

2.5.2 Implementation of Bayesian optimisation based Gaussian process approach

This study focuses on selecting the most suitable hyperparameters for modelling anaerobic digestion processes at a real-scale plant using ML models (DNN and XGBoost). To achieve this result, the Bayesian Optimization-based Gaussian Process approach (BO-GP) was implemented using the Scikit-optimize package [42]. Scikit-optimize-0.9.0, often abbreviated as skopt, is a Python library designed to provide efficient and scalable tools for hyperparameter optimization and Bayesian optimization. The library offers robust implementations of BO methods, which are particularly useful for optimizing functions with expensive evaluations or non-smooth functions. Furthermore, it provides a simple and intuitive interface for defining optimization problems and specifying search spaces for hyperparameters using a range of optimization methods. Machine learning models were developed using Python programming language, incorporating the high-level deep learning libraries TensorFlow [43] and Keras [44], which are responsible for managing the neural network architecture. Keras is an Application Programming Interface (API) created by Google to develop high-level neural networks. Keras is seamlessly integrated with TensorFlow,

allowing for the smooth and efficient implementation, design, training, and evaluation of neural network models. The XGBoost package provides a comprehensive set of tools for training, optimizing, and deploying XGBoost models within the Python ecosystem.

The idea of the framework study may be elucidated by means of a flowchart, as shown in Figure 2. The flowchart provides a comprehensive overview of the sequential processes required to execute the program, with a particular focus on critical phases. The dataset has been divided into a training set (80%) and test set (20%). The objective function was subsequently created by including three sequential steps in the construction of the BO-GP for the ML model. The specification of the hyperparameter space to be optimized includes six hyperparameters for the neural network architecture and seven hyperparameters for the XGboost configuration models, together with their respective range values. Furthermore, properly configuring the ML models necessitates defining the hyperparameter space. The effectiveness and performances of machine learning-generated models and the optimal configuration model with the best hyperparameters combination during the BO-GP were assessed using evaluation metrics. Metrics were evaluated by calculating the Mean Absolute Error (MAE).

2.5.6 DNN and XGBoost Models comparison

DNN and XGBoost models have been trained on a biogas dataset containing 388 events with 16 features.

The search for the optimal value of the objective function was executed using the proposed algorithm, on the hyperparameters detailed in Table 1. The purpose was to determine the search approach for a surrogate GP model, which would be guided by the EI function to construct a model based on the hyperparameter values. The hyperparameter combinations (N_t) were configured to perform 100 trials. Following the training and evaluation of the generated ML models, the model

with the lowest MAE value and optimized hyperparameters is reported. Ultimately, the most accurate model configuration was preserved.

Table 1. Hyperparameter settings and the associated range values for ML models (DNN and XGboost) for the model generation

Model	Hyperparameters	Range
Neural network	Number of dense layers	$\mathbb{Z}$[1,5]
	Number of neurons	$\mathbb{Z}$[10, 100]
	Dropout rate	$\mathbb{R}$[0.1, 0.9]
	Activation function	[elu, selu]
	Optimizer	[RMsprop, Adam]
	learning rate	$\mathbb{R}[10^{-1}, 10^{-6}]$
XGBoost	n_estimators	$\mathbb{Z}$[50, 500]
	Max_depht	$\mathbb{Z}$[1,10]
	booster	[gbtree, dart]
	gamma	$\mathbb{R}$[1, 20]
	Reg_lambda	$\mathbb{Z}$[1,50]
	subsample	$\mathbb{R}$[0.5, 0.9]
	learning rate	$\mathbb{R}$[0.01, 0.5]

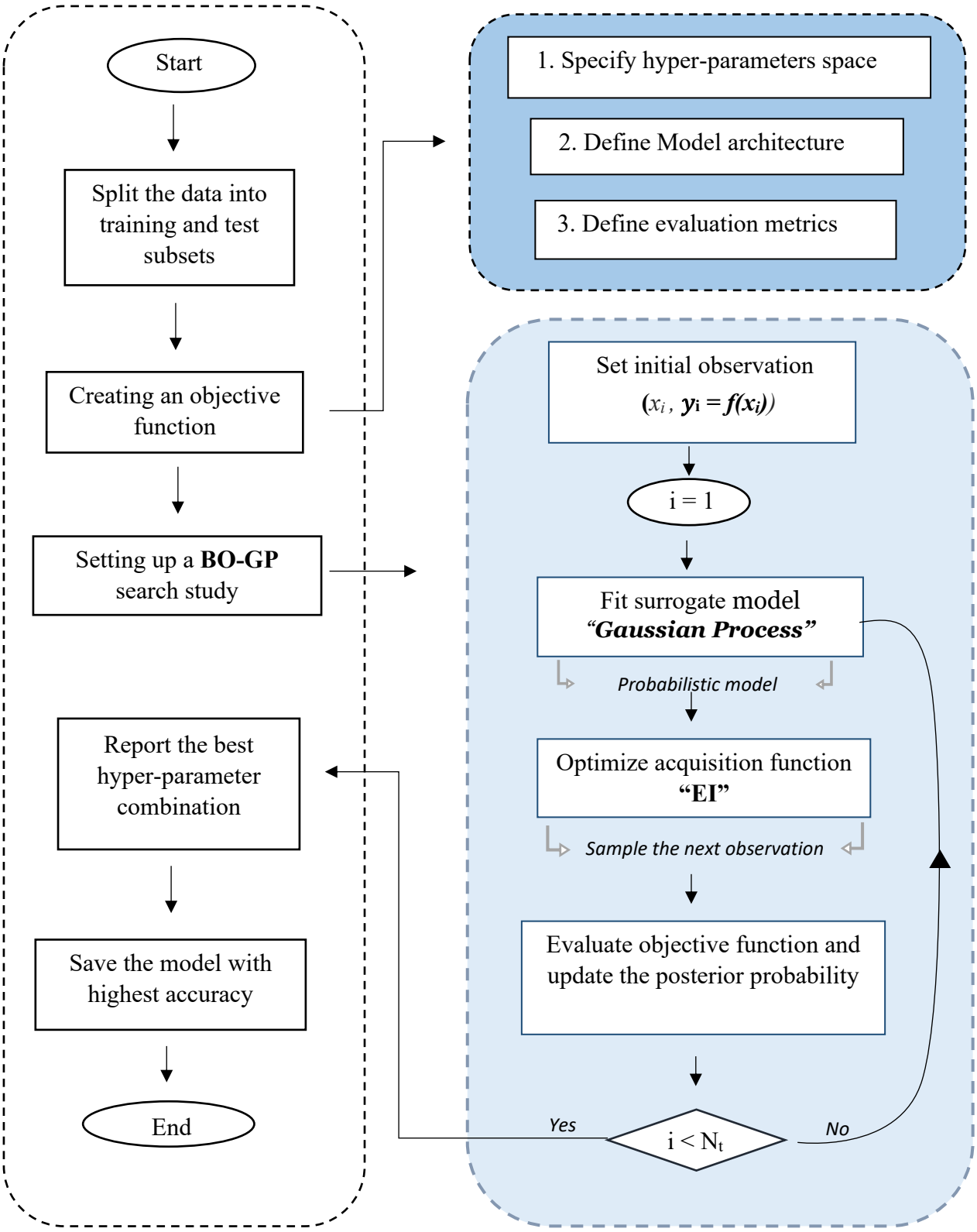


Fig. 2. Flowchart of the ML Models implementation using the hybrid BO-GP approach.

3. Results and Discussion

This section presents and discusses the results in two parts. The first part focuses on the experimental investigation of critical factors affecting the anaerobic digestion process, which are evaluated at the reactor input and outflow. The second part presents modelling results obtained using deep neural networks (DNNs) and XGBoost model and highlights their capacity to predict and optimize biogas production based on experimental data.

3.1 Temperature evolution during anaerobic digestion

Temperature is an essential parameter for the effectiveness and stability of the anaerobic digestion process. Optimal management of this parameter promotes microbial activity, ensures the efficient degradation of organic matter, and thereby maximises biogas production. Controlling the inlet and outlet temperatures is particularly important at a large-scale digester [45]. The inlet temperature guarantees that the effluent entering the digester reaches optimal thermal conditions to successfully initiate the biological process and thereby good overall performance. The output temperature indicates the internal thermal equilibrium of the system and the digester's effectiveness.

Figure 3 illustrates the temporal change in the inlet and output temperatures of the UASB digester, as averaged daily over a 15-week period selected from the dataset. It can be seen that the temperature at the inlet of the USAB digester ranges from 20°C to 30°C over this period, showing relatively stable temperature variation with some fluctuation. These fluctuations may be due to variations in ambient temperature, flow rate, or effluent composition. Such fluctuations affect the digester's internal temperature. Fluctuations at the outlet can also be caused by a sudden influx of overheated or underheated effluent.

Although the outlet temperature of the digester is higher, it fluctuates more than at the inlet, varying between 25 °C and 40 °C. The temperature at the digester's outlet is typically higher than the temperature at the inlet for a number of reasons [46]. These include the heat-generated by the biological and biochemical activities of microorganisms as well as the residence time of different materials in the digester; generally, a longer residence time leads to the higher temperature. In addition, the digester's insulation system reduces heat loss, resulting in a high outlet temperature. It should be noted that the facility is equipped with an automatic boiler that ensures accurate temperature control during digester operation. As the effluent progresses through the digester, the temperature increases. The biochemical reactions that occur in the digester, particularly the degradation of organic matter, are exothermic. This means that they generate heat, which can also contribute to an increase in temperature inside the digester. The increase in temperature at the outlet of the UASB is then explained by the combined heat input from the boiler and the biochemical processes inside the digester. In this system, digestion occurs under mesophilic conditions to reach optimal biogas yield. This mode promotes a stable microbial balance, ensures efficient conversion of organic matter and offers a promising energy yield, making the process advantageous.

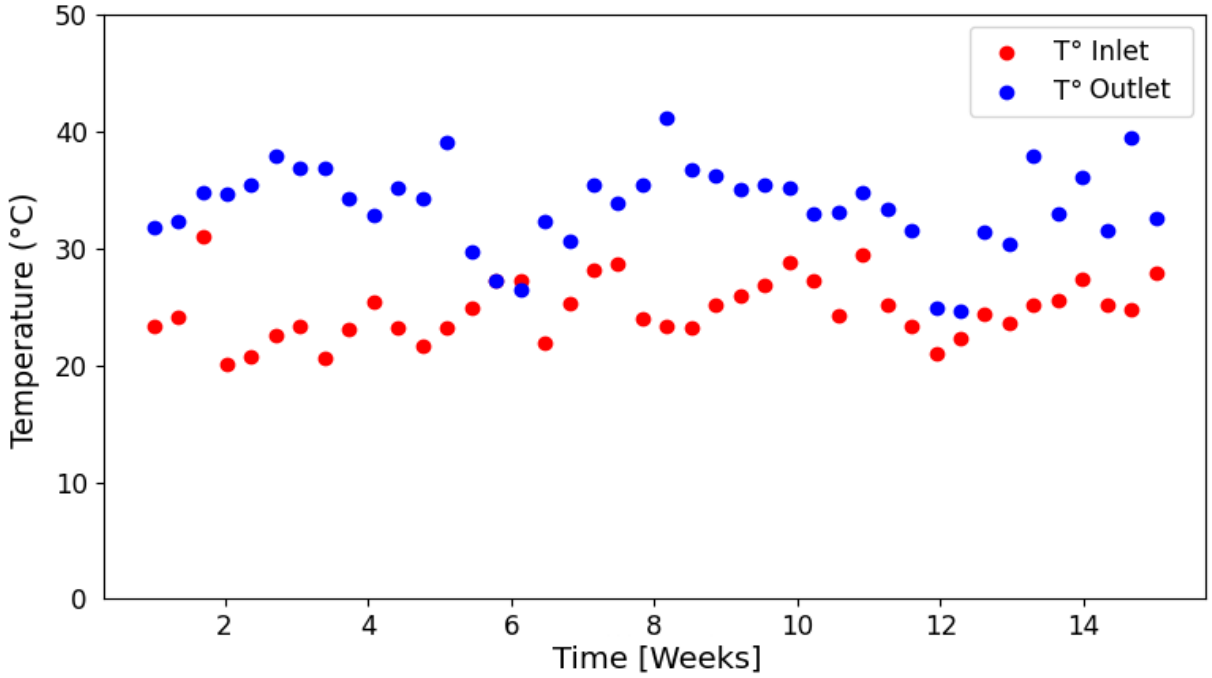


Fig.3. Temperature changes at the inlet and outlet of the UASB digester during wastewater anaerobic digestion.

3.2 pH evolution

The methanogenic microbial community is highly sensitive to pH, a factor that regulates its key biochemical reactions. This community is sensitive to variations in pH and to the acidification of the fermentation medium. It also plays an important role in the gas-liquid equilibrium of the reaction medium, and in particular in the buffering capacity of the medium [47].

Figure 4 illustrates the temporal evolution of pH at the inlet and outlet of the UASB digester. It can be seen that the pH values at the inlet of the UASB digester vary between 5 and 7.5, generally oscillating between an acidic and neutral environment. This is probably due to the presence of acidic compounds in the inflow. The main causes are fluctuations in the inlet flow rate and organic load.

However, the pH at the outlet is more stable, oscillating between 7.0 and 8.5 during the 12 weeks of anaerobic digestion, which corresponds to a neutral to slightly alkaline environment. These findings are similar to those reported by [45]. They confirm the UASB digester stability under anaerobic conditions for treating soft drink factories' and breweries' wastewater effluent. pH stability supported the methanogenic bacteria growth, which is essential for optimal anaerobic digestion. It has also been noted that, after 12 weeks of anaerobic digestion, the pH level at the digester outlet exhibits a similar trend to that at the inlet. However, the difference is smaller than during the fermentation process, which is probably due to a reduction in the organic load entering the digester during the last period.

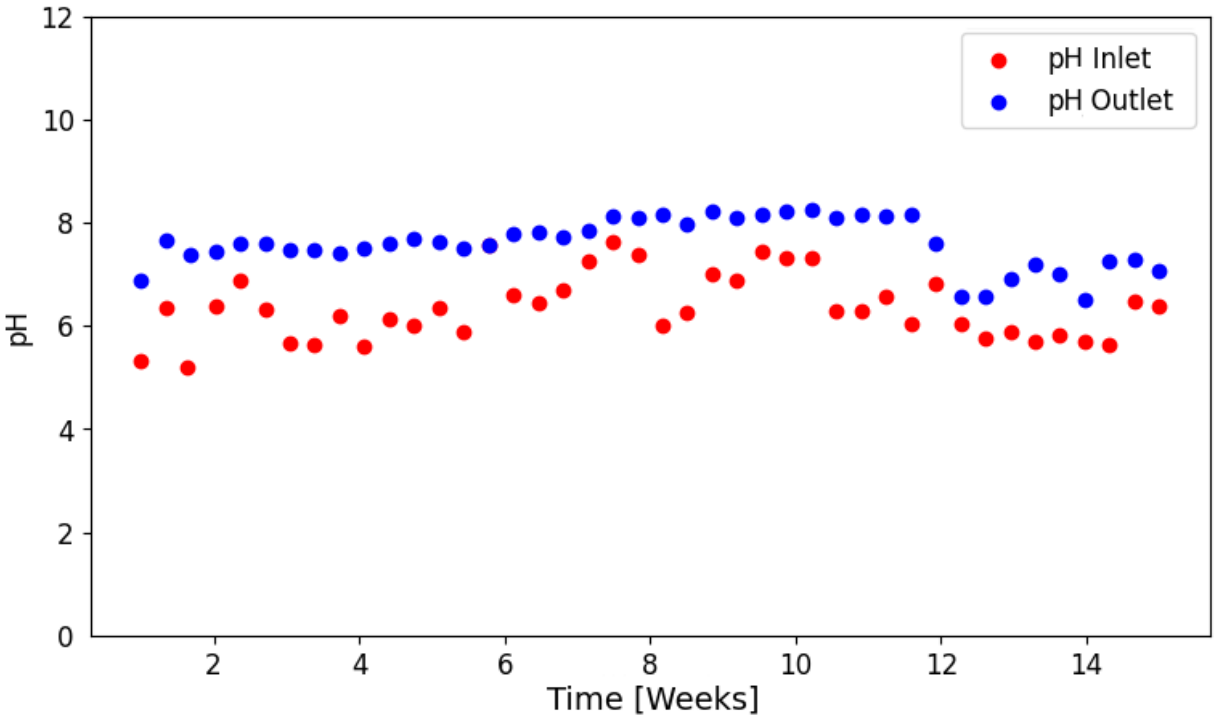


Fig.4. Temporal evolution of pH at the inlet and outlet of the UASB digester.

3.3 Chemical oxygen demand

Figure 5 displays the variation in chemical oxygen demand (COD) at the inlet and outlet of the digester over time. Clearly, the COD inlet values vary considerably over time, ranging from 2,000 mg/L to 6,000 mg/L. The COD values are consistent with the findings of Driessen et al. [48]. These values, however, are higher than the range of 2510 mg/L to 2617 mg/L reported in other studies [49]. These variations may be due to changes in the nature of the incoming wastewater discharged by the beverage industry. Production volume and chemical-based or detergents cleaning cycles are potential sources of these variations. The COD of the inflow is around 3000 mg/L during the first few weeks of anaerobic digestion of beverage waste. It then decreases slightly before increasing to a range of 4500–5000 mg/L between weeks 5 and 7. After the 7th week, the COD varies considerably, reaching a peak concentration of 6000 mg/L. This increase is likely attributed to a rise in the organic load during periods of intensive production.

This clearly shows that the organic load is variable, particularly when using waste from the soft drink and brewing industries.

This indicates that the organic load is variable, particularly when processing waste from the soft drink and brewing industries. Despite variations in the flow rate to the UASB digester, the COD values of the effluents remain relatively stable, ranging between 200 mg/L and 800 mg/L. This demonstrates the effectiveness of the system in reducing the organic load. These values are similar to those obtained by some authors [50, 49], for a UASB digester used as a treatment unit for brewery effluents. The average COD values of their effluents were 318 mg/L and 222 mg/L, respectively. The stability of the COD effluents values, even in the presence of occasional peaks, highlights the resilience of the digester, showing that it well adapts to feed disturbances and has sufficient buffering capacity to avoid major process disturbances.

The UASB digester achieved a stable COD removal, with efficiencies ranging from 87 to 95%. These values are higher than those obtained by Cecconet et al. [51], for a combined UASB/anaerobic sludge digestion (ASD) system treating municipal wastewater at psychrophilic temperatures. These results are consistent with those of Bakraoui et al. [45] ; Chelliapan et al. [51], who employed recycled paper mill wastewater.

A slight decrease in efficiency was observed during the 7th week, with a removal rate of 83%. This can be explained by the variability in the inflow's organic composition, a common occurrence in industrial wastewater treatment scenarios. It may take the microbial community in the UASB reactor longer to adapt to changes in the effluent, ensuring efficient degradation of organic matter. Furthermore, a reduction in hydraulic retention time may have contributed to the efficiency decrease during this period.

Nevertheless, the system recovered quickly, with efficiency improving after the 10th week and stabilizing at around 95% at the end of the fermentation period. The UASB digester showed promising capacity to treat high organic loads, achieving a high COD removal rate. This reflects the system's good performance despite effluent fluctuations and its ability to produce effluent that complies with current standards, thereby reducing environmental impact.

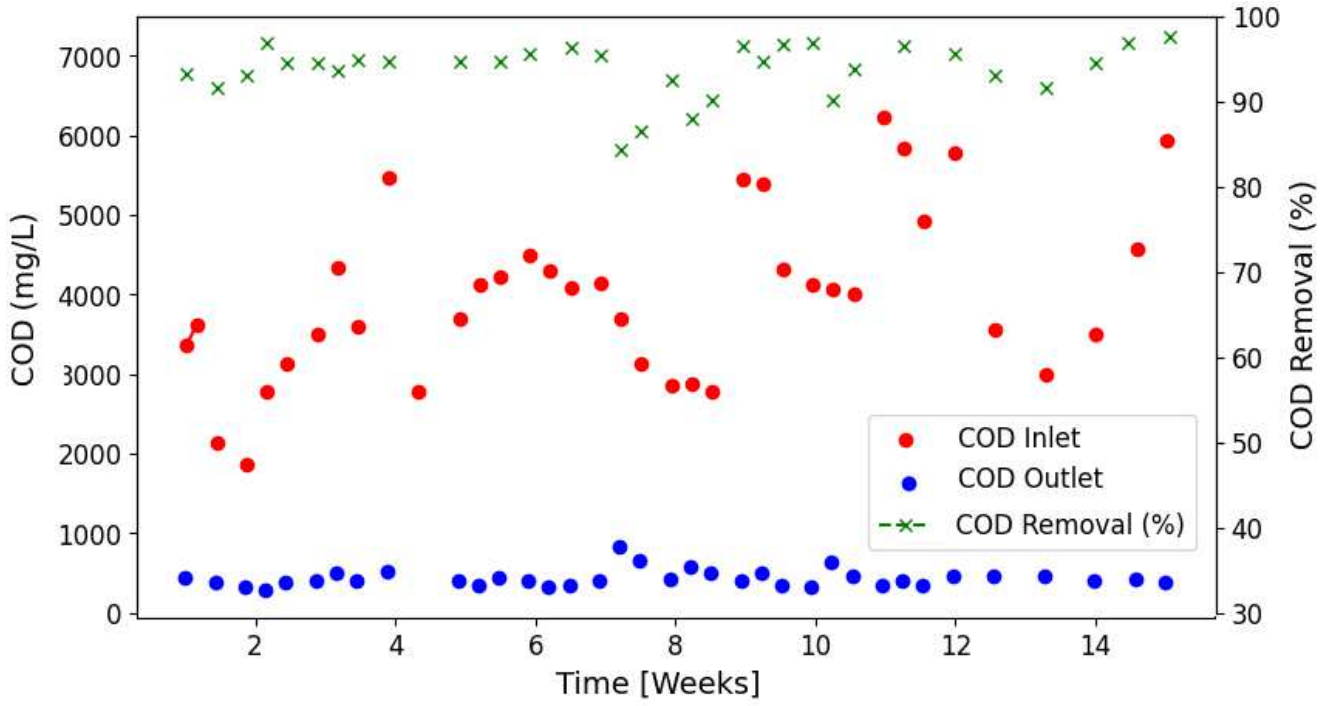


Fig.5. Temporal changes in COD levels at the inlet and outlet of the digester, along with the removal rate during AD.

3.4 Volatile fatty acids

Figure 6 depicts the variation in volatile fatty acids (VFAs) at the inlet and outlet of the UASB digester during the anaerobic digestion of wastewater from the soft drink and alcohol industry. VFAs are intermediate compounds formed during the decomposition of organic matter during anaerobic digestion. Their level typically indicates the digestion's efficiency and the balance between acid production and consumption. A significant variation is observed in the inlet concentration of volatile fatty acids (200 - 1000 mg/L) throughout the anaerobic digestion process. This variation is consistent with the variability in the organic load of the effluent feeding the digester. Similarly, the observed COD variations are due to periodic changes in the production cycles of the beverage industry. Nevertheless, the VFA concentration at the digester outlet

remained constant below 200 mg/L, mirroring the COD values achieved at the digester outlet (Figure 5). The bacteria involved in the anaerobic digestion of beverage waste consume VFA, then assimilate soluble sugars and significantly reduce the COD of the effluent [53]. This suggests efficient digestion, as the VFAs produced during acidogenesis and complex organic matter are broken down into simpler molecules, such as acetic acid, propionic acid and butyric acid, which are rapidly consumed by methanogenic bacteria, which convert them into methane and carbon dioxide during methanogenesis, the final stage of anaerobic digestion. It should be noted that VFA levels in the effluent are low, even during the weeks when VFA inputs are highest, indicating that the digestion process of beverage wastewater is well balanced.

The VFA values obtained are significantly lower than those reported by Sivagurunathan et al. [54], who obtained VFA concentrations between 15,800 and 17,080 mg COD/L in a CSTR reactor with immobilised cells used for the biological production of hydrogen from beverage production wastewater [54].

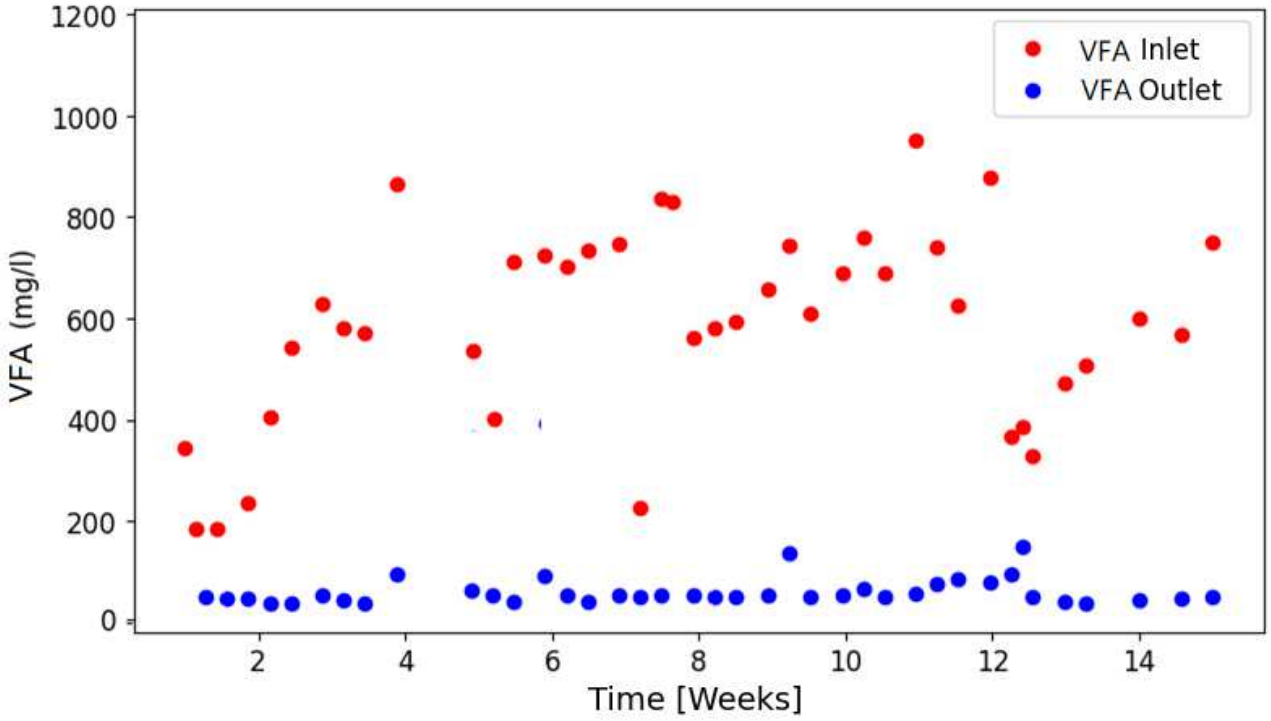


Fig.6. Temporal evolution of VFAs at the inlet and outlet of the UASB digester.

3.5 Biogas Production Modeling via Machine Learning

This section shows the impact of using the BO-GP hyperparameters approach on modelling biogas production from wastewater plants and then discusses the achieved results. Table 2 displays the optimal hyperparameters configuration for the neural network and XGBoost models created by BO-GP through 100 trials, and the corresponding MAE to assess the accuracy of the predicted biogas values. To enhance the comprehensive evaluation of the model's performance, two additional measures have been used to assess its quality and effectiveness: the coefficient of determination (R2) and the Root Mean Square Error (RMSE). Both models performed well in predicting biogas production, as indicated by their optimal configurations. However, the artificial neural network model showed satisfactory results and superior performance compared with the XGBoost model, as shown by the R2, MAE, and RMSE values of 0.7809, 148.0927, and 198.7302,

respectively, corresponding to respective improvements of 6.99%, 9.29%, and 8.09%. These performance metrics demonstrate the ability of the model to predict biogas production with good accuracy, making it a valuable tool for decision-making in wastewater treatment plants.

The optimal neural network configuration implements a simple architecture with two hidden layers, each containing 91 neurons. An insufficient number of neurons limits the capacity of an ANN to model a complex nonlinear mechanism, which may lead to excessive generalization [55, 16]. On the other hand, too many neurons can lead to overfitting, a case in which the error is minimal on the training data but significant on the test data, because the fitting functions become too complex as the number of neurons increases [55, 56]. To improve model generalization and robustness, a low dropout rate of approximately 0.143 was chosen. This means that only a small percentage of neurons is deactivated, as the model already incorporates a simple architecture, ensuring that relevant information is not lost during the training process. RMSprop was selected as the optimizer with a very low learning rate value of $5.68\ 10^{-4}$. The ELU activation function has been selected for the hidden and output layers. The BO-GP method established a straightforward neural network model architecture owing to the limited dimensions of the dataset (16 features and 388 rows). Three-layer ANN and nonlinear regression models were developed by some researchers Tufaner et al. [14], to predict the performance of biogas production using a hybrid reactor (AHR) for treating synthetic wastewater. The architecture of the ANN consists of 10:22:1 neurons arranged in the input, hidden, and output layers. These authors achieved good accuracy with an R2 value of 0.9852. To predict biogas generation from chemically treated co-digested agricultural waste, Almomani [15], built an ANN with one hidden layer, a few neurons (15), and producing more epochs (260) than in this study. The predictions of the AD process were highly accurate under various operating conditions and successfully predicted cumulative methane production with

99.1% of the data within ±10% of the experimental mean. In our previous investigation [2] (Sadoune et al., 2023), the optimized approach (BO-TPE), using the same dataset, employed a simple neural network architecture that consisted of a single hidden layer with 72 neurons. SELU and ReLu were selected as the hidden and output activation functions, respectively, with a dropout value of 0.14478, a learning rate of 0.00727, and a momentum of 0.7122. RMSprop was selected as the optimizer. However, the performance of a neural network model employing a GP approach was improved by an increase in performance accuracy R-squared from 0.712 to 0.7809, representing an approximate improvement of 6.89%. In contrast, some studies have achieved excellent performance using more complex neural network architectures. Shin et al. [57], developed a DNN model to predict engine-out NOx emissions during the WLTP cycle of a diesel engine. The BO approach improved the accuracy of the model and allowed the optimal configuration to be determined more efficiently than grid search and random sampling. The accuracy of the model was further improved by incorporating the hidden neuron determination logic, resulting in an R2 value of 0.9675. The optimal hyperparameters for the DNN model included 9 hidden layers with varying numbers of neurons in each layer, ELU activation function, and Adam optimization function.

Besides, the XGBoost model provided the following optimal hyperparameters: the boosting process was performed using 500 rounds (n_estimators) with a maximum tree depth of 7 (max_depth). DART was selected as the booster with a gamma value of 20, regularization lambda term of 1 (reg_lambda), subsample ratio of 0.9, and learning rate of 0.246. A large number of estimators allow the model to capture more complex patterns within the data, thus enhancing predictive accuracy. The selected maximum depth value enables the model to capture the interactions among features at a higher order. This degree of depth achieves a balance between

capturing and learning from intricate data relationships and sustaining effective computations. The DART booster employs dropout techniques to mitigate overfitting by avoiding dependence on individual trees, thereby improving the robustness and generalization capabilities. Note that a high gamma value can make the model more conservative, limiting the frequency of splits in the decision trees. Only splits that resulted in a substantial improvement in the performance of the model were considered. This parameter aids in tree pruning by minimizing unwanted splits. A small value of the regularization parameter lambda imposes a penalty on large coefficients, hence regulating model complexity. The subsample ratio of 0.9 introduces randomness by sampling 90% of the training data for each tree. All of these hyperparameters selected values contribute to a more generalized model that performs better on test data. Furthermore, the learning rate of 0.246 is quite high, which accelerates the convergence of the model by making larger updates to the model weights. This helps the model achieve optimal performance more quickly. However, to mitigate overfitting caused by the elevated learning rate, the model depends on regularization methods and conservative gamma settings. The selected hyperparameters reflect a thoughtful evaluation of the depth and breadth of the model. This combination allows fast learning while maintaining both accuracy and robustness. The resulting configuration yielded a model with an R-squared value of approximately 0.73, MAE value of 148.093, and RMSE value of 198.730. These results indicate good performance prediction and effective generalization of unseen data. Putatunda et al. [58], performed hyperparameter optimization of the XGBoost algorithm on six real-world datasets using Bayesian optimization with the Hyperopt, Grid search, and random search methods. They found that Hyperopt outperforms Grid search and Random search due to its higher prediction accuracy, making it the best approach for hyperparameter optimization considering accuracy and time computations. Some researchers [17] investigated two machine learning models, DNN and XGBoost, to predict the total energy consumption of an enclosed agricultural greenhouse with the

aim of improving microclimate control and energy efficiency. They implemented GridSearchCV as a hyperparameter tuning tool for model development, resulting in a 2-layer DNN model with 20 neurons in each layer, a batch size of 5 and a number of epochs of 100. The optimal hyperparameters for the XGBoost were found to be a learning rate of 0.1, 600 turns, a subsample size of 0.5, a maximum depth of 12 and a minimum child weight of 3. The results reveal that the DNN model outperforms the XGBoost model, showing superior predictive performance with an R^2 of 80.9%, an RMSE of 171.1, and an MAE of 130.3.

Table 2

Optimal DNN and XGboost models based on BO-GP hyperparameters selection.

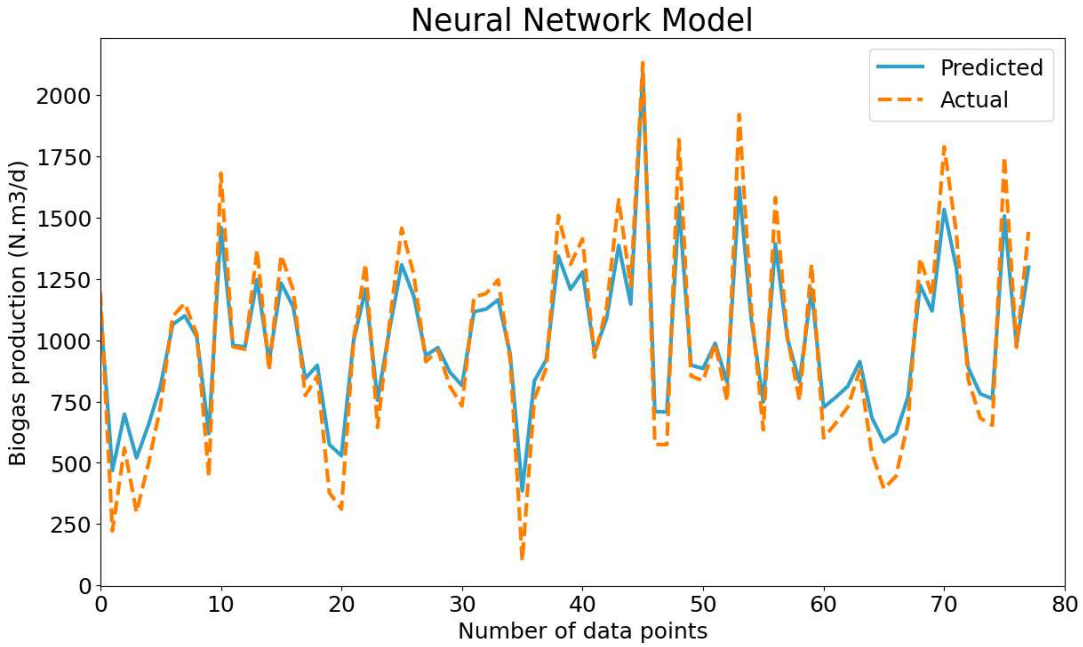
Machine learning Models	Hyperparameters	Optimal Hyperparameters Selection	Model evaluation		
			R^2	MAE	RMSE
Neural Network	Number of dense layers	2	0.7809	148.0927	198.7302
	Number of neurons	91			
	Dropout rate	0.143			
	Activation function	Elu			
	Optimizer	RMSprop			
	learning rate	$5.68\ 10^{-4}$			
XGBoost	n_estimators	500	0.7299	163.2680	216.2166
	Max_depht	7			
	booster	dart			
	gamma	20			
	Reg_lambda	1			
	subsample	0.9			
	learning rate	0.246			

3.5.1. Modelling biogas production using the DNN and XGboost models

Figure 7 displays the results of tests that compare the predicted and actual values of biogas generated from wastewater using the DNN and XGBoost models. The DNN model has shown

superior fitting to real biogas data compared to the XGBoost model. The highest biogas volume predicted by the DNN model was approximately 2090.051 Nm^3/d, which is nearly close to the actual amount of biogas produced at the plant corresponding to 2131 Nm^3/d. In this case, an R-squared value of 0.7809 was achieved, which indicates a good fit between the predicted and actual values. This can be explained by the 78.09% variation in the real biogas values. This is corroborated by the small disparity between the maximum predicted and real values, indicating the high accuracy of the model. The good correlation between the predicted and real values clearly shows that the DNN model effectively captures the underlying patterns and interactions involved during the wastewater conversion process. It must be underlined that the training and evaluation of models were performed using a limited dataset involving only 388 data points. According to the available literature, such a small amount of data is insufficient for training DNN models. These indicators suggest that the use of the GP approach helps develop a DNN model that successfully achieves generalization and avoids data overfitting. Hence, it serves as a significant tool for decision-making in wastewater treatment plants. In addition, the XGBoost model showed a good fit, as it only captures 73% of the variance in the case of real biogas production values with an R2 value of 0.73. The high biogas volume reached was approximately 1891.294 Nm^3/d, which is lower than the real output value and that reached by the DNN model. This suggests that the XGBoost model may not fully capture the peak production scenarios as well as the relationship between organic matter degradation and gas production. It shows also a lesser ability to stay in phase with the actual signal. High biogas production scenarios can be more variable and noisier, increasing the difficulty for the model to learn consistent patterns. This larger variability can dilute the predictive power of a model for extreme outcomes. This underperformance at higher values indicates that there are constraints in capturing extreme production scenarios, which may be the result of model bias, insufficient feature representation, or unbalanced data distribution with fewer high-production instances.

However, the XGBoost result highlights both the strengths and limitations of the model, demonstrating that this model is best suited to general operational contexts, while indicating opportunities for improving predictive accuracy. The results of this study demonstrate the effectiveness of using the GP approach to enhance the development of ML models, particularly DNN models. They also illustrate the challenge in discovering the optimal hyperparameter combinations for limited real-world wastewater data that may include noisy data.



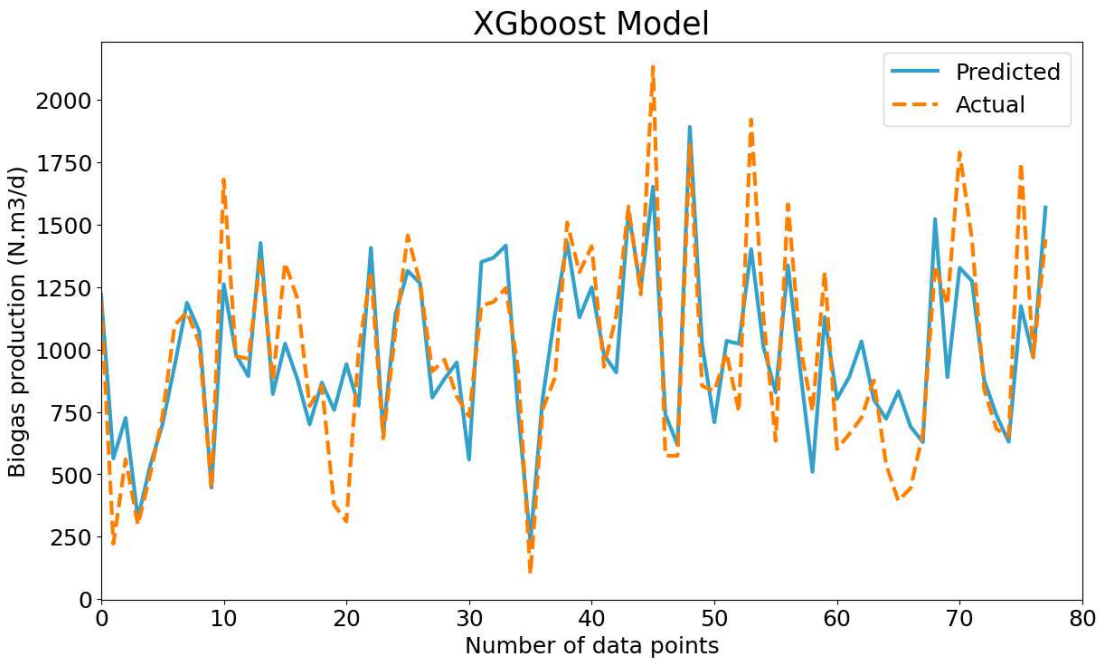


Fig. 7. Comparison between the actual and predicted biogas values derived from wastewater using the DNN and XGBoost models.

3.5.2. Convergence of the BO-GP optimization process for the DNN and XGBoost models

The convergence plot from scikit-optimize is a powerful tool for visualizing the performance of the optimization process. It specifically monitors the best observed value at each iteration, providing useful information on the rate and effectiveness of the BO-GP optimization process. This plot depicts the convergence of the models tested during the optimization trials based on the objective function "MSE". Then, it allows us to assess whether combinations of hyperparameters improve the model's performance after a specific number of trials. Simultaneously, it evaluates the ability of the model to accurately predict biogas production during anaerobic digestion by examining its convergence behaviour and the rate at which it converges towards an optimal solution. Figure 8 illustrates the convergence of the BO-GP optimization process approach during iterations for the DNN and XGBoost models. The plot depicts the MAE, which is the objective function being minimized versus the iteration number. Analyzing DNN convergence, the MAE

objective function starts at 1000 at the first iteration but quickly drops to a value less than 180 already at the second iteration. The significant decrease in the MAE objective function during the iterations suggests that the optimization process efficiently identified inherent improved hyperparameters compared to the initial estimation. As the iteration proceeds, the MAE continues to decrease but at a more gradual pace. This clear reduction indicates that the optimization process improves steadily but not as radically as in the initial cycles. It can be observed that at the 100th iteration, the MAE achieves its optimum value of approximately 148. The optimization process of BO-GP demonstrates its effectiveness in identifying superior hyperparameters, as seen by the significant decrease in MAE at the beginning and subsequent continuous enhancements towards progressively better sets of hyperparameters as the trials progress. The suitability of the model for DNNs has significantly improved after 100 iterations and has not yet reached a plateau, suggesting the possibility of achieving even better results via ongoing tuning.

Besides, the XGBoost model exhibits convergence of all MAE values over 100 iterations, resulting in a range between 250 and 163. The XGBoost model quickly reached a stable performance range (performance plateau). The corresponding convergence plot, yellow line, display a regular reduction of the MAE with the iterations. With respect to blue line relevant to DNN optimization, which cyclically improves substantially every 50 iterations, the best value appears almost attained after 100 iterations, indicating that the optimization process has thoroughly explored the hyperparameters space within the specified range. However, this behaviour also suggests that within the selected hyperparameters ranges, further improvements cannot be attained, or that the model's performance is less sensitive to the optimized hyperparameters. The specified range of hyperparameters may not include values that might result in significantly reduced MAEs, indicating a potential need to redefine or broaden this range.

This plot confirms earlier findings that indicate that the DNN model exhibits a high level of responsiveness to the BO-GP approach, achieving better improvements in MSE compared to the XGBoost model.

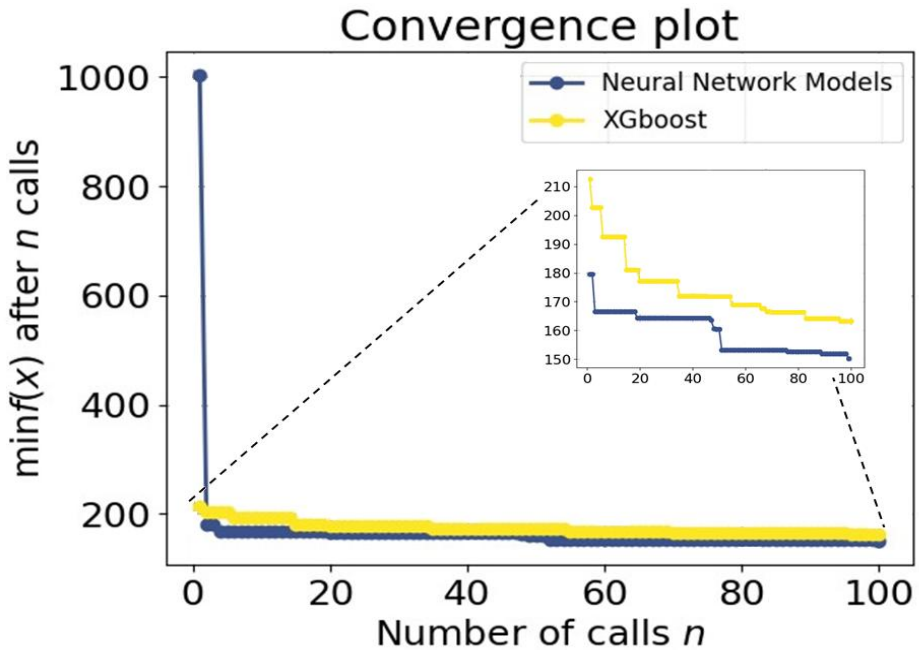


Fig. 8. Convergence plot of the BO-GP optimization process for the DNN and XGboost models.

3.5.3. Effect of hyperparameter on Neural Network models

The skopt package offers a range of visualizations that demonstrate the influence of hyperparameter tuning on objective function and trial numbers, providing information about hyperparameter performance and the accuracy of the "neural network configuration" scenario. A Partial Dependence Plot (PDP) is a popular approach for visualizing the average marginal influence of one or two features in the prediction of a black-box model [59]. They provide information on how certain hyperparameters affect the predicted model performance when applied to surrogate models [59]. This visualization describes the Bayesian optimization tuning procedure and highlights the promising regions of the hyperparameter space that exhibit the potential for optimal or near-optimal values of the objective function. The PDP matrix generated by the neural network

model is shown in Figure 9. The diagonal of the matrix indicates the impact of a single hyperparameter on the objective function as measured by the mean absolute error (MAE) metric. Conversely, the off-diagonal elements of the matrix represent the influence between two different hyperparameters. The dashed red line in the diagonal plots indicates the optimal selection of each hyperparameter. The yellow and blue regions in the off-diagonal plots indicate the lower values (better combinations) and higher values (poor combinations) of the objective function, respectively. The points that were sampled during the optimization process are represented by black dots, where the number of trials is fixed at 100. Each plot displays a red star corresponding to the optimal value of the two hyperparameters. Based on the diagonal plots, it is clear that the first plot, which corresponds to the evaluation of the average learning rate at hyperparameter variations while keeping the other hyperparameters constant, has the most significant impact on the objective function. Some researchers [60, 2] have asserted that the learning rate is the most crucial hyperparameter and should always be adjusted. This is because of its direct impact on training dynamics and model performance. The speed of the updates during optimization is determined by this parameter, which directly affects the degree to which the model converges to an optimal solution. Moreover, achieving efficient training, steady convergence, and superior model performance in deep-learning tasks requires careful adjustment of the learning rate. The number of neurons and hidden layers are also important hyperparameters that significantly influence the selection of the optimal model architecture. The activation function, dropout rate, and optimizer have almost the same influence, as shown in the partial dependence plots, which are flat over these three hyperparameters. Sadoune et al. [2], suggested the use of TPE as a technique to optimize hyperparameters in DNNs configurations, including the activation function, dropout rate, learning rate, number of neurons, number of hidden layers, and the optimizer. These authors noticed that the learning rate hyperparameter is the most pivotal aspect in the configuration of a DNNs

compared with other optimized hyperparameters. This constitutes 52% of the overall configuration, followed by the number of hidden layers. The number of neurons has a negligible influence, whereas the optimizer, dropout rate, hidden activation, and output function have minimal impact. The learning rate converges to the optimal value of the objective function, which is approximately 10^{-2}, then subsequently increases. However, the best neural network model identified $5.6\ 10^{-4}$ as the optimal learning rate. This value is located in the near-optimal region but not precisely in the optimal region. This is because the objective function is affected by other hyperparameters that represent the optimal combination of the neural network model (i.e., hyperparameter configurations). Furthermore, the PDP may be less reliable if the generated samples originate from a noisy validation set (poorly learned regions). A high value of the learning rate has the potential to induce instability or failure in convergence during the optimization process, thereby increasing the likelihood of overshooting the optimal parameters. On the other hand, if the learning rate is too low, the training progress can be slow, resulting in extended periods of convergence and lower accuracy. A fine balance suitable to a particular neural network architecture, dataset properties, and optimization technique is required to determine the optimal learning rate. The partial dependence plot for the number of neurons indicates that, as the number of neurons increases, the objective function "MAE" decreases, implying that the model works optimally with 91 neurons. It should be noted that neural network models with two or three hidden layers perform better. The number of neurons in each layer directly influences the capacity and expressiveness of a neural network, whereas the number of hidden layers determines the depth of the network. An increased number of neurons inside the hidden layer facilitates the network's ability to learn more complex patterns from data, thereby potentially capturing complex relationships. Nevertheless, an extremely high number of neurons and layers might result in overfitting, a phenomenon in which the neural network acquires the ability to memorize training data rather than generalizing to unseen data.

However, to avoid increasing the complexity of the model, the BO-GP strategy based on the expected improvement acquisition function opted for just two hidden layers because the increased number of neurons compensates for this choice. The dropout rate hyperparameter does a small favour for neural network configurations with low values; therefore, the model performs better with a value for this hyperparameter of 0.15. The dropout rate in DNNs architecture is crucial for addressing overfitting, enhancing generalization, and enhancing the robustness of the model. The use of dropout mitigates this issue by impeding the co-adaptation of neurons during the training process, ensuring that no individual neuron substantially depends on other neurons for making predictions. Elu and RMSprop have been selected as the activation function and optimizer, respectively, for the best neural network configuration model. As seen in the off-diagonal plots, the distribution of points picked by the BO-GP is more concentrated in or around the yellow region, which corresponds to the most promising regions of the search space (i.e., the region with the lowest mean absolute error rate). Only a few instances are located in the blue region, which represents unfavourable regions where the scoring function has a high value. This confirms the effectiveness of the proposed approach. It can be noted that during the initial phase of the GP algorithm execution, all hyperparameters exert a similar influence on the performance of the neural network models developed, leading to an expanded search space. As the trials advanced, the surrogate model utilized the acquisition function to refine the selection process by identifying the most favourable hyperparameters and reducing their ranges. Consequently, this leads to an enhancement in the objective function score. In summary, the results indicate that the objective function score exhibits greater sensitivity to variations on three hyperparameters, namely the learning rates, the number of neurons, and the number of hidden layers. These hyperparameters play a significant role in the selection of the best neural network configuration.

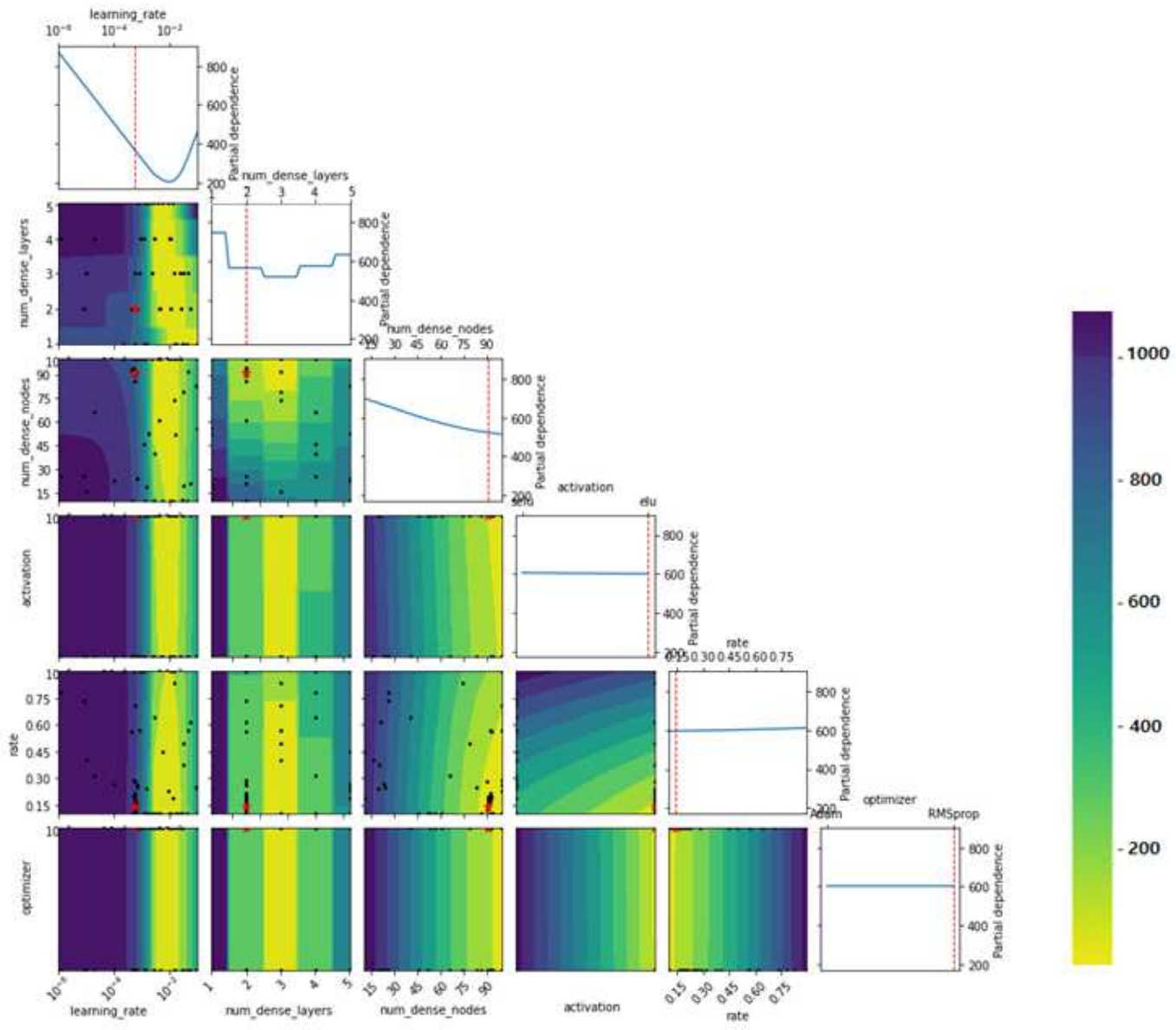


Fig. 9. Scikit-Optimize partial dependence plots for neural network configuration models with Bayesian optimization based on the Gaussian process. Contour plots of MAE in the sub diagonal graphs; blue lines in the diagonal graphs report the effect of the hyperparameters, with the optimal values indicated with a red star symbol and red dashed lines.

The objective plot 2D function in Scikit-Optimizer is used to better visualize the interactions between the two hyperparameters that constitute the optimal neural network model configuration.

Figure 10 shows the 2-D contour plots landscapes of the optimized hyperparameters. They reveal the presence of several sampled points around the red star. Plot A demonstrates that an increase in the learning rate value from 10^{-3} to 10^{-1} leads to an improved model performance. This improvement occurs for all neural network configurations (trials) with 1 to 5 hidden layers, constituting an area with the optimal score, where the majority of the samples lie. Plot B shows that the model configuration can achieve optimal performance with any number of neurons as the learning rate approaches the promising region. In fact, many samples are positioned next to the promising region around the values of 89 – 93 and 3.10^{-4}-10^{-3}, which correspond to the number of neurons and learning rate, respectively. These values should be an estimate of the optimal combination of the two hyperparameters, marked by a red star. As seen in contour plot C, the models show good accuracy when the number of hidden layers fell between 2 and 3, with a high number of neurons exceeding 80. However, when the number of hidden layers is between 4 and 5, the models tend to have fewer than 50 neurons. Considering the limited dataset size in this investigation, the GP approach demonstrates that it is more effective to build a neural network model with a smaller architecture, balancing between number of neurons and number of hidden layers, rather than an increased complexity of the model.

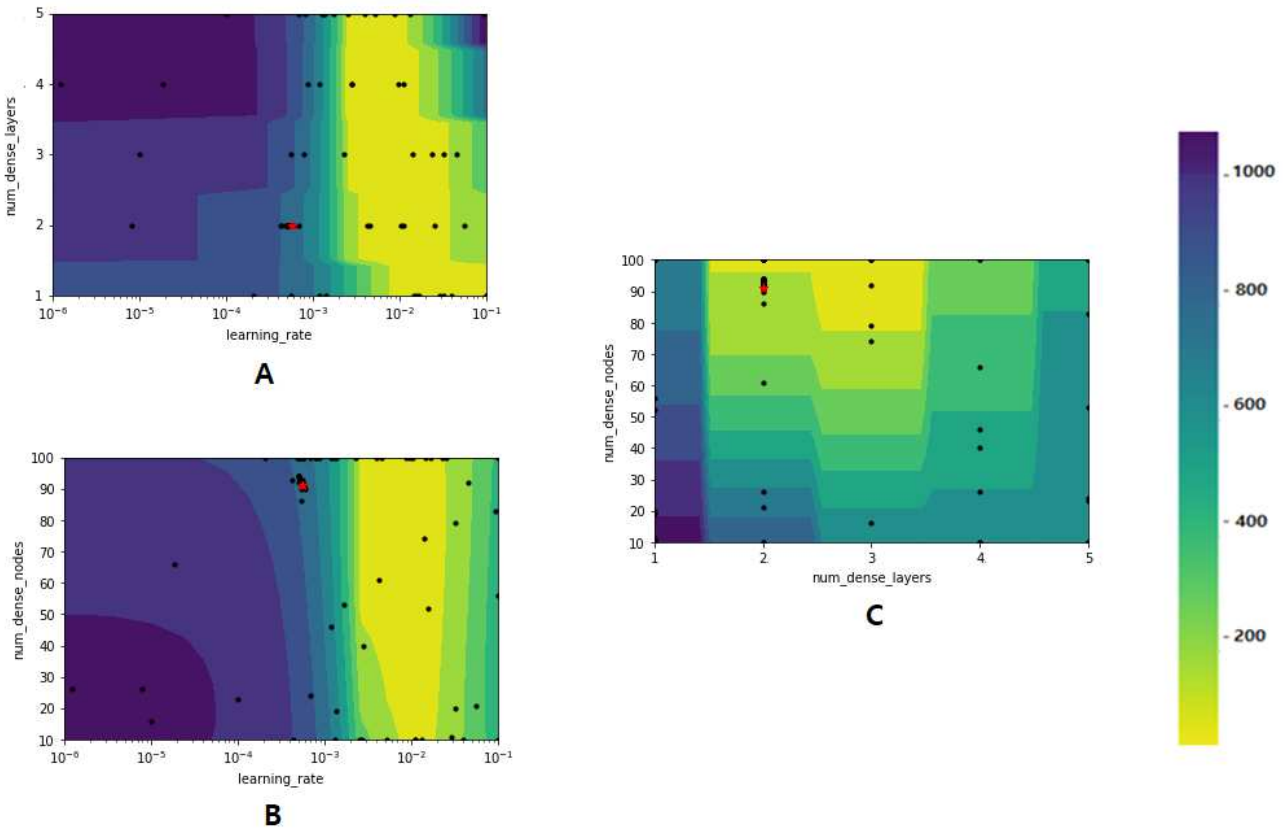


Fig. 10. 2-D contour plots landscapes of MAE values as function of the optimized hyperparameters that contribute to the best configuration of the neural network model. Plot A shows a landscaped plot for the learning rate and the number of hidden layers hyperparameters. Plot B shows a landscaped plot for the learning rate and the number of neurons hyperparameters. Plot C shows a landscape plot of the number of hidden layers and neurons hyperparameters.

3.5.4. Evaluation of sample order by tuning points optimization

Figure 11 depicts histogram plots that show the sample distributions for each hyperparameter across the selected interval range. In the 2-D search space, scatter plots display samples of every possible hyperparameter combination. Note that the point colours correspond to the order of sample rankings. The optimal configurations are indicated by red stars. These plots are helpful in visualizing how the optimization algorithm explores and exploits the hyperparameter space over

successive trials. They represent the trajectory of the optimization process, showing how the algorithm samples and refines the hyperparameter configurations over successive steps. Typically, the x-axis represents the trial number or optimization process step and the y-axis represents the best observed objective value (maximized MAE score). The histograms show that in the majority of trials the sampled hyperparameters are close to those selected for the best configuration. This is consistent with the result obtained from the partial dependence plot. The learning rate and number of hidden layers hyperparameters have sampled more than 55 points out of a total of 100 in the trial, ranging between $3\ 10^{-4}$ and $5\ 10^{-4}$ with two neurons, respectively, in the region where the red star is positioned. The histogram of the number of neurons hyperparameters shows that the majority of points had been sampled between 90 and 100 neurons, with over 60 points falling within this range. Several neural network configurations have shown relatively low values for the dropout rate hyperparameter ranging from 0.15 to 0.20. More than 40 points have been sampled in these configurations. More than 60 trials selected ELU as an activation function for the neural network model configuration. For the optimizer, the Adam and RMSprop hyperparameters had little impact on the neural network model scores. Adam and RMSprop optimizers have almost the same sample distribution percentage. Off-diagonal plots are represented by scatter plots of different colours. The darkest points (blue) reflect the earliest trials, and the lightest points (yellow) represent the latest trials. It is clear that there are brighter points around the red star, which indicates that the objective function exhibited improvement as the trials progressed. Initially, in early trials, the BO-GP approach focuses on exploring a variety of hyperparameters combinations over a wide range of search spaces rather than focusing on a specific region. This exploration aims to gather diverse information regarding the behaviour of the objective function and identify regions that might contain the global optimum. However, the trial samples are chosen in regions where the surrogate model exhibits significant uncertainty, indicating limited overall knowledge of the behaviour of

the objective function. As more trials progress and the surrogate model is updated with additional data points, BO-GP begins a transition towards a more balanced exploration-exploitation strategy. The algorithm leverages the information gained from previous trials to refine its understanding of promising regions in the search space. BO-GP dynamically adjusts its exploration-exploitation trade-off based on the surrogate model's predictions and the guidance of the acquisition function. Points are selected to achieve a balance between exploring areas with uncertain or potentially high-value objective function outcomes (exploration) and exploiting regions of the search space that are likely to yield the best performance based on the current knowledge (exploitation).

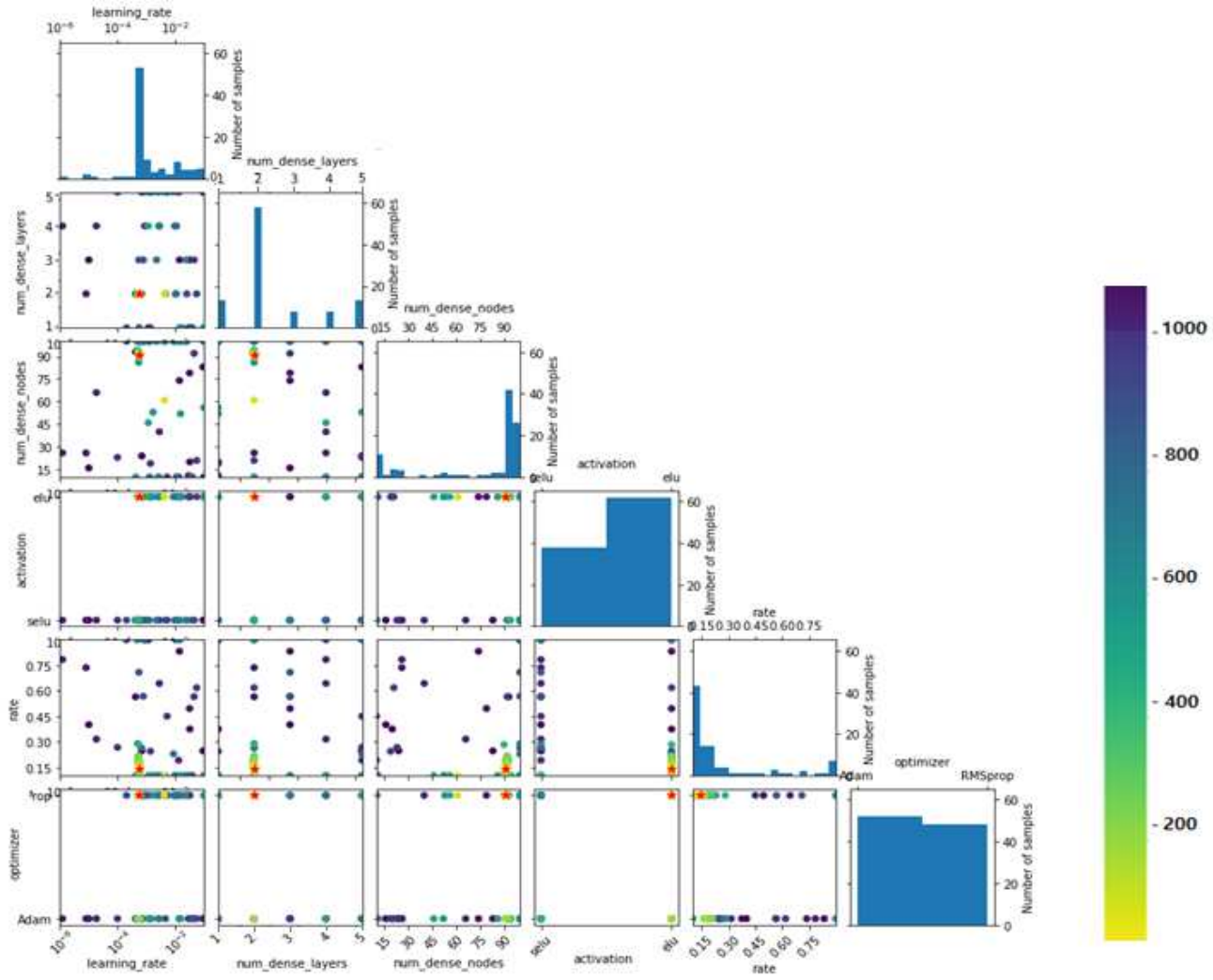


Fig. 11. Evaluation of sampling order via tuning points optimization.

4. Conclusions

This study investigated the application of machine learning models, specifically Deep Neural Networks (DNN) and Extreme Gradient Boosting (XGBoost), for predicting biogas production from an industrial wastewater treatment plant, starting from a dataset with a relatively low number of samples and with some noise, a condition well representing a real case scenario. A Bayesian optimization approach using Gaussian processes (BO-GP) was employed to optimize the hyperparameters of these models. The key findings and conclusions are as follows.

1. The DNN model outperformed the XGBoost model in predicting biogas production, achieving an R-squared value of 0.7809, compared to 0.73 for XGBoost. This indicates that the DNN model was able to capture the complex relationships in the anaerobic digestion process more effectively, at least when starting from a not very large dataset.

2. The BO-GP approach proved effective in identifying optimal hyperparameters for both models. For the DNN, a simple architecture with two hidden layers and 91 neurons per layer was found to be optimal, along with a low dropout rate of 0.143 and a learning rate of 5.68×10^{-4}. The XGBoost model's optimal configuration included 500 estimators, a maximum tree depth of 7, and a learning rate of 0.246.

3. The convergence plots demonstrated that the BO-GP optimization process efficiently explored the hyperparameter space, with the DNN model showing more significant improvements in Mean Absolute Error (MAE) compared to XGBoost.

4. For the DNN model, the learning rate was found to be the most influential hyperparameter, followed by the number of neurons and hidden layers. This highlights the importance of carefully tuning these parameters for optimal model performance.

5. The developed models, particularly the DNN, show promise for practical application in predicting biogas production in wastewater treatment plants. This can aid in optimizing plant operations and improving energy recovery from the anaerobic digestion process.

The study was conducted with a limited dataset of 388 data points. Future research could explore the performance of these models with larger datasets and investigate the incorporation of additional relevant features to further improve prediction accuracy.

In conclusion, this study demonstrates the potential of machine learning approaches, particularly DNNs optimized using BO-GP, for modelling biogas production in anaerobic digestion processes. The findings contribute to the growing body of knowledge on applying advanced data-driven techniques in wastewater treatment and renewable energy production. Further research in this direction could lead to more efficient and sustainable operation of wastewater treatment plants and enhanced biogas recovery.

References

1. Oliveira-Esquerre KP, Mori M, Bruns RE (2002) Simulation of an industrial wastewater treatment plant using artificial neural networks and principal components analysis. Brazilian Journal of Chemical Engineering 19(4):365–370. https://doi.org/10.1590/S0104-66322002000400002
2. Sadoune H, Rihani R, Marra FS (2023) DNN model development of biogas production from an anaerobic wastewater treatment plant using Bayesian hyperparameter optimization. Chemical Engineering Journal 471:144671. https://doi.org/10.1016/j.cej.2023.144671
3. Mjalli FS, Al-Asheh S, Alfadala HE (2007) Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. Journal of Environmental Management 83(3):329–338. https://doi.org/10.1016/j.jenvman.2006.03.004
4. Rustum R (2009) Modelling activated sludge wastewater treatment plants using artificial intelligence techniques. PhD thesis, Heriot-Watt University

5. Pai TY, Yang PY, Wang SC, Lo MH, Chiang CF, Kuo JL, Chu HH (2011) Predicting effluent from the wastewater treatment plant of industrial park based on fuzzy network and influent quality. Applied Mathematical Modelling 35(8):3674–3684. https://doi.org/10.1016/j.apm.2011.01.019
6. Guo H, Jeong K, Lim J, Jo J, Kim YM, Park JP, Kim JH, Cho KH (2015) Prediction of effluent concentration in a wastewater treatment plant using machine learning models. Journal of Environmental Sciences 32:90–101. https://doi.org/10.1016/j.jes.2015.01.007
7. De Clercq D, Wen Z, Fei F, Caicedo L, Yuan K, Shang R (2020) Interpretable machine learning for predicting biomethane production in industrial-scale anaerobic co-digestion. Science of the Total Environment 712:134574. https://doi.org/10.1016/j.scitotenv.2019.134574
8. Nourani V, Elkiran G, Abba SI (2018) Wastewater treatment plant performance analysis using artificial intelligence—an ensemble approach. Water Science and Technology 78(10):2064–2076. https://doi.org/10.2166/wst.2018.477
9. Nourani V, Asghari P, Sharghi E (2021) Artificial intelligence based ensemble modeling of wastewater treatment plant. Journal of Cleaner Production 291:125772. https://doi.org/10.1016/j.jclepro.2020.125772
10. Aghdam E, Mohandes SR, Manu P, Cheung C, Yunusa-Kaltungo A, Zayed T (2023) Predicting quality parameters of wastewater treatment plants using artificial intelligence techniques. Journal of Cleaner Production 405:137019. https://doi.org/10.1016/j.jclepro.2023.137019
11. Bartz-Beielstein T, Zaefferer M, Mersmann O (2023) Tuning: Methodology. In: Hyperparameter tuning for machine and deep learning. Springer, Singapore, pp 7–26. https://doi.org/10.1007/978-981-19-5170-1_2
12. Benkendorf DJ, Hawkins CP (2020) Effects of sample size and network depth on a deep learning approach to species distribution modeling. Ecological Informatics 60:101137. https://doi.org/10.1016/j.ecoinf.2020.101137
13. Beltramo T, Hitzmann B (2019) Evaluation of linear and non-linear prediction models optimized with metaheuristics: application to anaerobic digestion processes. Engineering in Agriculture, Environment and Food 12(4):397–403

14. Tufaner F, Demirci Y (2020) Prediction of biogas production rate from anaerobic hybrid reactor by artificial neural network and nonlinear regression models. Clean Technologies and Environmental Policy 22:713–724
15. Almomani F (2020) Prediction of biogas production from chemically treated co-digested agricultural waste using artificial neural network. Fuel 280:118573
16. Sakiewicz P, Piotrowski K, Ober J, Karwot J (2020) Innovative artificial neural network approach for integrated biogas–wastewater treatment system modelling. Renewable and Sustainable Energy Reviews 124:109784
17. Ghiat I, Al-Ansari T (2024) Enhancing efficiency in closed agricultural greenhouses: A data-driven predictive model for energy consumption. IOP Conference Series: Earth and Environmental Science 1372:012084
18. Alruqi M, Sharma P (2023) Biomethane production from waste using Bayesian optimization. Fermentation 9(2):120
19. De Clercq D, Jalota D, Shang R, Ni K, Zhang Z, Khan K, Yuan K (2019) Machine learning powered software for accurate prediction of biogas production. Journal of Cleaner Production 218:390–399
20. Tetteh EK, Rathilal S, Chetty M, Armah EK, Asante-Sackey D (2019) Treatment of water and wastewater for reuse and energy generation
21. Sevillano CA, Pesantes AA, Peña Carpio E, Martínez EJ, Gómez X (2021) Anaerobic digestion for producing renewable energy. Entropy 23:145
22. Ahmed Y, Yaakob Z, Akhtar P, Sopian K (2015) Production of biogas and performance evaluation of existing treatment processes. Renewable and Sustainable Energy Reviews 42:1260–1278
23. Bonaccorso G (2017) Machine learning algorithms. Packt Publishing
24. Jordan MI, Mitchell TM (2015) Machine learning: Trends, perspectives, and prospects. Science 349:255–260
25. Brei VA (2020) Machine learning in marketing: Overview and applications. Foundations and Trends in Marketing 14(3):173–236
26. Marsland S (2014) Machine learning. CRC Press
27. Bishop CM (2006) Pattern recognition and machine learning. Springer

28. Shakarami A, Ghobaei-Arani M, Shahidinejad A (2020) A survey on computation offloading approaches. Computer Networks 182:107496
29. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
30. Avanijaa J (2021) Prediction using XGBoost regression algorithm
31. Sridharan M (2023) Performance prediction using XGBoost
32. Weeraddana N, Premaratne S (2021) Prediction using XGBoost algorithms
33. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. Proceedings of KDD, pp 785–794
34. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Annals of Statistics
35. Torlay L, Perrone-Bertolotti M, Thomas E, Baciu M (2017) Machine learning classification. Brain Informatics 4:159–169
36. More AS, Rana DP (2017) Review of random forest classification techniques to resolve data imbalance. Proceedings of ICISIM, IEEE, pp 72–78
37. Lei Y, Jiang W, Jiang A, Zhu Y, Niu H, Zhang S (2019) Fault diagnosis method using XGBoost. Processes 7:589
38. Pensoneault A, Yang X, Zhu X (2020) Nonnegativity-enforced Gaussian process regression. Theoretical and Applied Mechanics Letters 10:182–187
39. McIntire M, Ratner D, Ermon S (2016) Sparse Gaussian processes for Bayesian optimization. UAI Proceedings
40. Murugan P (2017) Hyperparameter optimization using Bayesian approach. ArXiv
41. Brochu E, Cora VM, De Freitas N (2010) Tutorial on Bayesian optimization. ArXiv
42. Louppe G (2017) Bayesian optimisation with scikit-optimize
43. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mane D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viegas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2016) TensorFlow: Large-scale machine learning on distributed systems. arXiv:1603.04467
44. Chollet F (2020) Keras deep learning library. https://keras.io

45. Bakraoui M, Karouach F, Ouhammou B, Aggour M, Essamri A, Bari HE (2020) Biogas production from wastewater using UASB digester. Biotechnology Reports 25
46. Merlin G, Kohler F, Bouvier M, Lissolo T, Boileau H (2012) Importance of heat transfer in anaerobic digestion. Bioresource Technology 124:59–67
47. Borja R, Alba J, Banks CJ (1996) Anaerobic digestion of olive oil wastewater. Process Biochemistry 31:219–224
48. Driessen W, Vereijken T (2003) Recent developments in biological treatment of brewery effluent
49. Firew T, Daniel F, Solomon SS (2018) Wastewater treatment plant performance assessment. Journal of Applied Sciences and Environmental Management 22:1285–1292
50. Mahmoud N (2008) High strength sewage treatment in UASB reactor. Bioresource Technology 99:7531–7538
51. Cecconet D, Mainardis M, Callegari A, Capodaglio AG (2022) Psychrophilic wastewater treatment. Chemosphere 297:134228
52. Chelliapan S, Mahat SB, Din MDF, Yuzir A, Othman N (2012) Anaerobic digestion of paper mill wastewater. Iranian Journal of Energy and Environment
53. Owuor OB (2020) Use of biogas from brewery wastewater. PhD thesis
54. Sivagurunathan P, Sen B, Lin CY (2015) Hydrogen production from wastewater. Applied Energy 147:1–9
55. Andrade Cruz I, Chuenchart W, Long F, Surendra KC, Andrade LRS, Bilal M, Liu H, Tavares Figueiredo R, Khanal SK, Ferreira LFR (2022) Machine learning in anaerobic digestion. Bioresource Technology 345:126433
56. Alrashed AAAA, Karimipour A, Bagherzadeh SA, Safaei MR, Afrand M (2018) Thermophysical properties modeling using ANN. International Journal of Heat and Mass Transfer 127:925–935
57. Shin S, Lee Y, Kim M, Park J, Lee S, Min K (2020) Deep neural network with Bayesian optimization. Engineering Applications of Artificial Intelligence 94:103761
58. Putatunda S, Rama K (2018) Comparative analysis of hyperparameter optimization
59. Moosbauer J, Herbinger J, Casalicchio G, Lindauer M, Bischl B (2021) Explaining hyperparameter optimization. NeurIPS
60. Bengio Y (2012) Practical recommendations for training deep architectures. Springer