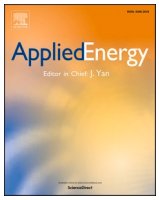
Applied Energy 390 (2025) 125781

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Full-scale dynamic anaerobic digestion process simulation with machine and deep learning algorithms at intra-day resolution

Alberto Meola [a,b], Sören Weinrich [a,c,*]

[a] DBFZ, Deutsches Biomasseforschungszentrum Gemeinnützige GmbH, Torgauer Straße 116, Leipzig 04347, Germany
[b] Leipzig University, Faculty of Mathematics and Computer Science, Augustusplatz 10, Leipzig 04109, Germany
[c] Münster University of Applied Sciences, Faculty of Energy · Building Services Environmental Engineering, Stegerwaldstraße 39, Steinfurt 48565, Germany

HIGHLIGHTS

- Random Forest and LSTM Neural Networks are recommended for AD process prediction.
- Linear models show high performance difference between validation and test datasets.
- Only few measurements are highly influential for prediction of methane production.
- Data preparation parameters are highly influential in model performances.

ARTICLE INFO

Keywords:
Artificial intelligence
Biogas technology
Dynamic process prediction
Feature importance
Bioprocess modelling

ABSTRACT

Machine learning algorithms have been proven to be effective in predicting characteristic process variables of the anaerobic digestion process. However, industrial application has rarely been investigated, and the most effective algorithms for typical operating conditions have not been defined. Thus, 13 machine learning, deep learning and statistical algorithms were applied to three full-scale datasets at intra-day resolution. A systematic procedure was applied for reliable data preparation and hyperparameter optimization. Methane yield was predicted one step, 12 h and 24 h in advance. Results indicate that random forest and long short-term memory neural networks are the most robust algorithms, while further linear models can be advantageous in specific situations. Previous step methane yield and fed volatile solids are, in general, the most relevant parameters, while further laboratory measurements can be advantageous at high feed quantities. Data preparation is crucial to allow less complex models (such as linear models) to perform well. This study defines appropriate machine learning algorithms and essential measurements for characteristic process conditions at different data resolutions, when predicting dynamic intra-day methane production of industrial-scale anaerobic digestion processes, as a reliable basis for model-based process monitoring and control.

1. Introduction

Anaerobic Digestion (AD) is a biochemical process that transforms organic matter, including municipal waste, livestock manure, and energy crops, into biogas and nutrient-rich digestate. In the context of wastewater and agricultural engineering, there is a necessity to develop and assess future-oriented operational concepts for AD plants [1]. The application of AD processes to offer demand-driven power presents a potential solution to address the intermittent nature of renewable energy conversion [2], but robust monitoring and control systems are required to ensure stable and efficient operating conditions at all times. Due to the non-linearity of the AD process, model-based monitoring and control systems are required for optimal plant performance [3]. Among the available models, the semi-mechanistic Anaerobic Digestion Model No. 1 (ADM1), presented by Batstone et al. [4] is frequently applied for AD process modelling. However, owing to the restricted availability of available measurements for model application, various simplifications of the original ADM1 have been suggested [5,6,7]. Independently from model complexity, several phenomena observed in the AD process (such as mixing and micro-oxygenation) have not been modelled yet due to

* Corresponding author at: Münster University of Applied Sciences, Faculty of Energy · Building Services Environmental Engineering, Stegerwaldstraße 39, Steinfurt 48565, Germany.
E-mail address: weinrich@fh-muenster.de (S. Weinrich).

https://doi.org/10.1016/j.apenergy.2025.125781
Received 16 November 2024; Received in revised form 5 March 2025; Accepted 18 March 2025
Available online 7 April 2025

their microbiological complexity and highly non-linear nature. Thus, there is a growing interest in employing phenomenological or empirical modelling techniques, such as Machine Learning (ML), for non-linear prediction of characteristic variables in AD processes [8,9]. An overview of selected studies is presented in Table 1. To utilize various measurements available at industrial biogas plants and ensure comparability of the applied procedures, mainly full-scale investigations were chosen within the literature review. Thus, individual publications differ in the applied algorithms, selected input and output features, time resolution or individual data preparation techniques, among other aspects.

Generally, most investigations in Table 1 depict biogas production at daily resolution, while only a few studies include methane production as output feature [10,15,12,23] or use process data at intra-day resolution [13,17]. Thus, typically daily measurements were used to describe continuous (and often steady-state) operation of industrial AD plants. Furthermore, most investigations simulate AD plants treating agricultural or municipal wastes (including sewage sludge) while only a few studies use lignocellulosic biomass [20] or energy crops [13] as substrates for the AD process. Sun et al. [22], for instance, predicted steady-state biogas production from full-scale AD of food waste with a Multilayer Perceptron (MLP) Neural Network (NN), whereas Dittmer et al. [13] used Linear Regression (LR) to predict dynamic operation with irregular feeding patters of agricultural biomass and dung at hourly resolution.

Several publications in Table 1 apply ML and Deep Learning (DL) algorithms to predict methane or biogas production. Among the most commonly used ML methods, LR, Random Forest (RF), Support Vector Regression (SVR), and Extreme Gradient Boosting (XGBoost) have been successfully implemented, as demonstrated by Li et al. [17]. The authors applied these algorithms to model biogas production from an industrial-scale AD plant digesting food waste. More complex approaches, such as Long Short-Term Memory (LSTM) NNs and Convolutional Neural Networks (CNNs) have been investigated less frequently [14,19]. Additionally, some studies employ advanced architectures, such as the dual-stage attention-based LSTM NN with a Variable Selection Network (VSN) used by Jeong et al. [14]. The authors tested several architectures to predict biogas production from AD of wastewater treatment sludge and food waste leachate at daily resolution.

In general, the selection of an appropriate ML or DL algorithm should be based on the specific characteristics of the available data and experimental conditions. Thus, suitable data preparation procedures are crucial for effective process depiction with ML or DL models [24]. However, seven out of fourteen publications analyzed in Table 1 either do not make use of data preparation or perform only mandatory operations such as data cleaning or replacing zeroes with close to zero values [10,11]. Thus, only the remaining seven authors applied individual data preparation procedures. Among the seven publications, Schroer and Just [18] used LR, the Tree-based Pipeline Optimization Tool (TPOT), MLP and a hybrid linear-MLP model to successfully predict biogas production one day in advance during full-scale AD of municipal wastewater sludge and high-strength waste such as fats, oil and grease. The data preparation process involved linear regression-based feature selection, addition of time variables and distribution-based train-test split. Results show that the MLP overperformed all the applied models, while TPOT and the hybrid model did not outperform the LR model. This study represents one of the rare occasions where authors used a systematic concept for the optimization of model hyperparameters and data preparation techniques, including data engineering and feature selection.

To investigate explainability of applied ML algorithms different procedures were applied in individual investigations summarized in Table 1. Thus, established procedures such as Permutation Feature Importance (PFI), Mean Decrease of Impurity (MDI) for RF or Shapley Additive Explanations (SHAP) were frequently used to investigate model behaviour and dependencies. Zhang et al. [23], for example, applied the SHAP algorithm to the prediction of biogas yield and methane content from municipal solid waste and straw through a dry fermentation process. Results show that the weight of the percolate tank content and level of liquid in the percolate tank are the most important input parameters for both biogas production and composition.

Depending on individual process conditions a clear understanding of model complexity is required for efficient application of ML and DL algorithms in industrial operation. While several studies investigated the application of multiple model types to a single data set, no direct comparison of different model types and experimental conditions (e. g., steady-state or dynamic operation) has yet been performed. As shown in Table 1, many investigations utilize daily measurements for model application (training, test and simulation), which however are not sufficient for online application of model-based monitoring and control procedures. Furthermore, industrial datasets most often show a high imbalance of online and offline measurements (with considerable measurement error and high amount of missing data). Detailed investigation for systematic application of suitable procedures for data preparation and hyperparameter optimization are required for direct comparison and comprehensive evaluation of different model types and process conditions.

This work aims to bridge the gap between experimental application of ML algorithms to AD processes and industry usage of such algorithms, providing guidance for future development of ML-based monitoring and control procedures. Thus, three different full-scale datasets were analyzed and modelled with 12 ML and DL algorithms and one purely statistical model. All algorithms and data preparation processes were optimized based on a meta-heuristic optimization pipeline [24]. The three datasets differ within the type and number of substrates used, the stability of the process and quality of available measurements. Thus, the experimental setup represents typical operating conditions of industrial AD plants (including e. g., pulse feeding or limited amount of laboratory analyses). Moreover, the impact of the prediction horizon and data resolution on model performances was analyzed. Importance of each feature was calculated for the best performing models and the most relevant hyperparameters in the model optimization process were highlighted and discussed within a Sensitivity Analysis (SA).

2. Methods

In the current investigation 13 prediction models were tested on three datasets. All the datasets were resampled at 15 min, 1 h and 6 h resolution. Moreover, datasets were evaluated also on three different Observation Distances (ODs). As described in Meola et al. [24],

"[…] the Observation Distance (OD) is defined to characterize the time interval between historical data and prediction of methane production. For prediction at time t, all measurements of available features t to $t - OD$ are ignored, considering the historical input from $t - OD$ backwards. Only feeding quantities are provided to the model until the time t at each prediction since they are considered as operational control variables."

Thus, the tested ODs were one timestep (15 min, 1 h and 6 h), 12 h and 24 h. One combination of an individual dataset, data resolution and OD is defined as an optimization instance (Table 2).

Each dataset was split with a 70:20:10 ratio in train, validation and test data. The models were trained on the training set, model hyperparameters were optimized on the validation dataset (based on [24]), and model performance was evaluated on the test dataset.

Time and model performance on specific datasets, data resolution and OD were analyzed separately, as well as differences between validation and test performance. Finally, SA was performed on all datasets at OD 24 h and FI was performed on the best performing models.

2.1. Prediction models and Hyperparameters

The AD process is a non-linear process with high measurement uncertainty and unknown process behaviour [25]. Nevertheless, linear and generally less robust models were tested, since individual algorithms for

Table 1
Summary of recent publications on the application of ML and DL algorithms for AD process prediction.

Authors and year	Applied algorithms	Input features	Output feature (s)	Reactor scale	Substrates used	Time resolution	Data preparation technique	Hyperparameters optimization	Explainability
De Clercq et al. [10]	EN, RF, XGBoost	Feed type and quantities, day number	Methane production	Full-scale	15 substrates including animal manure, food waste and percolate	Daily	Data cleaning, generation of additional features	Manual optimization	PFI and PDP
Hansen, Bolette D. et al. [11]	Ensemble of various ML algorithms	Gas production of previous days	Biogas production	Full-scale	Seaweed, manure, eulat, pectin and other 14 substrates	Daily	Replacement of zeros with close-to-zeros values	Unknown	–
Wang et al. [12]	RF, EN, SVR, k-NN	Carbon, Nitrogen, cellulose and xylan content, C/N Ratio	Methane production	Lab-scale	Different substrates – such as manure, food waste and corn stover – depending on considered reactor	Daily	–	Unknown	MDI for RF
Dittmer et al. [13]	LR	Previous gas production, feed type and quantities	Biogas production	Full-scale	Corn, grass, whole crop silage, sugar beet and dung	Hourly	–	Lags determined via cross-correlation	LR coefficients
Jeong et al. [14]	Dual-stage attention LSTM	17 process variables, including input and output sludge, mass and organic loading rates	Biogas production	Full-scale	Wastewater	Daily	VSN	Bayesian optimization network	Unknown feature importance technique
Long et al. [15]	EN, RF, MLP, k-NN, SVM, XGBoost	Genomic data, VFAs concentration, temperature, OLR, HRT, substrate type	Methane production	Lab-scale	Several substrates, such as waste activated sludge or seaweed	Daily	–	Unknown	MDI for RF
Wang et al. [16]	Ensemble of ML algorithms	Feed type and quantities, TS, VS, VFA content, alkalinity and VFAs/ alkalinity ratio	Biogas production	Full-scale	WWTP sludge and 29 additional substrates such as brine, dairy, fats and oils	Daily	TPOT	TPOT	PFI and PDP
Li et al. [17]	RF, SVM, XGBoost, logistic regression, MLP	Feed type, quantities, and qualities, 14 digestate properties such as TS, COD, VFAs among others	Biogas production	Full-scale	Restaurant and household food waste	ca. 3 h	Local outlier algorithm detection, k-NN as imputation method	Grid search	–
Schroer and Just [18]	Ensemble ML models, MLP	Previous gas prediction, VS loading, hour of the day and several operational parameters such as aeration basin air flow and recycled activated sludge	Biogas production	Full-scale	WWTP sludge	Daily	Feature selection with LR and TPOT	Grid search	Ridge regression coefficients, SHAP
Sappl et al. [19]	TFT, ARIMA, k-NN	Operational parameters such as temperature, total carbon loading, raw sludge dry matter load, sludge loss on ignition, and pH values	Biogas production	Full-scale	WWTP sludge	Daily	Outlier removal, VSN	Grid search	Attention-based FI
Wang et al. [20]	LR, EN, MLP, SVM, GPR, ELM, decision tree, k-NN, ensemble	Substrate composition (TS, VS and cellulose among others)	Biogas production	Lab-scale	Lignocellulosic biomass	Daily	Data cleaning	Unknown	–
Yildirim and Ozkaya [21]	k-NN, RF, SVM, MLP, XGBoost	Digestate VS, TS and VFAs concentration, alkalinity, pH, reactor temperature, gas composition	Biogas production	Full-Scale	Biowastes such as manure and slaughterhouse waste	Ca. 4 days	Data cleaning	10-fold cross-validation	PCA and F-score
Sun et al. [22]	MLP	Biomass type and fed mass, digestate VS concentration and pH, reactor temperature and volume, OLR and HRT	Biogas production	Lab-scale	Food waste of several origins, such as kitchen waste or food distributors waste	Unknown	–	Several meta-heuristic methods such as COA and MVO	SA
Zhang et al. [23]	Ensemble of ML algorithms	Substrate and digestate TS and VS, VFAs concentration, electrical conductivity	Biogas production and methane content	Full-scale	Municipal solid waste and straw	Unknown	Autogluon and H_2O – both AutoML frameworks	Autogluon and H_2O – both AutoML frameworks	SHAP

ARIMA: Auto Regressive Integrated Moving Average, COA: Cuckoo Optimization Algorithm, COD: Chemical Oxygen Demand, ELM: Extreme Learning Machine, EN: Elastic Net, FI: Feature Importance, GPR: Gaussian Process Regressor, HRT: Hydraulic Retention Time, k-NN: k-Nearest Neighbors, LR: Linear Regression, LSTM: Long Short-Term Memory, MDI: Mean Decrease of Impurity, ML: Machine Learning, MLP: Multi-Layer Perceptron, MVO: Multi-Verse Optimization, OLR: Organic Loading Rate, PCA: Principal Component Analysis, PDP: Partial Dependence Plots, PFI: Permutation Feature Importance, RF: Random Forest, SA: Sensitivity Analysis, SHAP: Shapley Additive Explanations, SVM: Support Vector Machine, SVR: Support Vector Regression, TFT: Temporal Fusion Transformer, TPOT: Tree-Based Pipeline Optimization Tool, TS: Total Solids, VFAs: Volatile Fatty Acids, VS: Volatile Solids, VSN: Variable Selection Network, WWTP: Wastewater Treatment Plant, XGBoost: Extreme Gradient Boosting Regressor.

data preparation can provide nonlinear generated features that can help linear models to model nonlinear process behaviour [24].

While several selected models are commonly used in relevant literature (RF, EN, k-NN, LR, GPR), recently developed new methods (1D-CNNs, LSTM, GRU, ELM) were also selected and modified to meet the requirements of the specific prediction task. GBR was selected instead of XGBoost due to the simpler implementation process while a combined LSTM/GRU was chosen instead of standard LSTM due to the fast and variable performance of GRU. ELM was upgraded to a multi-layer ELM and SARIMAX was applied instead of ARIMA due to the influence of the feed on the biomethane production. CNNs, in general, have been selected since they are proven to be effective in time series forecasting [26], while PLS and BR were applied for evaluating further linear models that might outperform LR and EN. ABR was chosen to evaluate one further gradient boosting technique apart from GBR. Ordinary multi-layer perceptrons were excluded due to their performances on time series being inferior than RNNs – with similar simulation time on optimized GPU architectures [27]. Moreover, transformers were excluded due to extremely long training time and to possible instability due to the number of hyperparameters to be optimized [28]. Due to high computational time, several other models were also excluded, such as support vector regression and genetic programming.

2.1.1. ML models

Several classical ML models were applied in this study to evaluate which model could provide the best results (Table 2).

As basic benchmark an ordinary least squares optimization of an LR shown in Eq. 1 was applied.

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{x}$ represents the input data, $\boldsymbol{\beta}$ represents the coefficient matrix and $\boldsymbol{y}$ represents the output value. The equation is solved as in Eq. 2.

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{argmin} S(\boldsymbol{\beta}) \tag{2}$$

with $\widehat{\boldsymbol{\beta}}$ being the estimated optimal coefficient matrix and S being the cost function.

Based on LR, Elastic Net involves the usage of the L1 (lasso method) and L2 (ridge method) to regularize the output of the linear model [29]. The linear coefficients are optimized by the addition of two regularization parameters λ_1 and λ_2, and model coefficients are calculated according to Eq. 3.

Table 2
Individual choices for the definition of different optimization instances and prediction models.

Prediction Model[1]	Dataset	Data Resolution	Observation Distance
LR, EN, BR, PLS, RF, ABR, GBR, GPR, k-NN, ELM, RNNs	Dataset B and C	15 min	1 timestep
		1 h	12 h
		6 h	24 h
	Dataset A	15 min	1 timestep
		1 h	12 h
			24 h
1D-CNN[2]	Dataset A, B and C	1 h	1 timestep
		6 h	12 h
			24 h
SARIMAX[2]	Dataset B and C	6 h	1 timestep
			12 h
			24 h

[1] ABR: AdaBoost Regressor, BR: Bayesian Ridge, ELM: Extreme Learning Machine, EN: Elastic Net, GBR: Gradient Boosting Regressor, GPR: Gaussian Process Regressor, k-NN: k-Nearest Neighbors, LR: Linear Regression, PLS: Partial Least Squares, RF: Random Forest, RNN: Recurrent Neural Networks.

[2] Due to the excessive time consuming training process at higher sample numbers, 1D-CNN was only applied at 1 h and 6 h resolution, while SARIMAX was only tested at 6 h resolution.

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\text{argmin}}\left(\| y - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda_2 \| \boldsymbol{\beta}\|^2 + \lambda_1 \| \boldsymbol{\beta}\|_1 \right) \tag{3}$$

This procedure (also called shrinkage) is used to prevent overfitting on the data, considering that new data samples might not follow the distribution of the training data [30].

Bayesian Ridge (BR) was applied as well, which consists of a Bayesian LR with an L2 regularization [31]. Bayesian LR is based on prior assumptions on the distribution of the dependent variable y. The LR model is then first expressed in probabilistic form as in Eq. 4.

$$\begin{aligned} \mu &= \alpha + \boldsymbol{\beta}\boldsymbol{x} \\ \boldsymbol{y} &\sim \mathcal{N}(\mu, \sigma) \end{aligned} \tag{4}$$

The dependent variable follows a normal distribution parametrized by mean μ – directly proportional to $\boldsymbol{X}$ parametrized by α and $\boldsymbol{\beta}$ – and the standard deviation σ. A prior distribution is then assumed for $\alpha, \boldsymbol{\beta}$ and σ, and the parameters of such distribution are then optimized according to Eq. 5 [32].

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{argmin}\left(\frac{1}{\sigma^2} \| \boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|_2^2 + \frac{1}{\tau^2} \| \boldsymbol{\beta}\|_2^2 \right) \tag{5}$$

with τ being the precision parameter for the prior distribution [33].

Additionally, Partial Least Squares (PLS) was applied in his canonical variation [34]. In PLS, a defined number of latent variables is defined as linear combination of the input features. Those latent variables are then used as input features for linear prediction of the output feature and are defined as those variables that maximize their covariance with the output feature. The canonical variation of PLS uses a canonical correlation analysis for the calculation of the covariance.

For direct description of non-linear process behaviour as well, further non-linear ML methods are implemented. RF is an ensemble learning algorithm that combines multiple decision trees [35]. Each tree is trained on a random subset of available data, and the final prediction is an average of the predictions from individual trees, as shown in Eq. 6.

$$\boldsymbol{y} = \frac{1}{N_T} \sum_{i=1}^{N_T} f_i(\boldsymbol{x}) \tag{6}$$

with N_T indicating the number of decision trees used by the model to perform the prediction. Since the trees are not correlated, RF helps reduce overfitting. In this study, the RF algorithm was utilized in its regressor version.

A similar method is the AdaBoost Regressor (ABR). This algorithm works by iteratively training a sequence of weak learners and assigning weights to each data point based on the errors made by the previous models [36]. The final prediction is a weighted sum of the individual weak learner predictions as shown in Eq. 7.

$$\boldsymbol{y} = \sum_{i=1}^{N_L} \alpha_i f_i(\boldsymbol{x}) \tag{7}$$

with N_L indicating the number of weak learners used by the model to perform the prediction. The coefficient α is assigned to each regressor, which allows specific weak learners to have a higher weight compared to others. This is achieved by updating α at each learning iteration depending on the training error. The higher the training error, the lower the weight assigned to each regressor. This approach allows to adaptively adjust the weights of misclassified samples, focusing on the harder-to-predict instances in each iteration. The adaptability of ABR contributes to improved overall prediction performances.

Unlike ABR, which sequentially adjusts the weights of misclassified samples, Gradient Boosting Regressor (GBR) focuses on fitting subsequent models to the residuals of the previous ones [37]. This iterative process minimizes a loss function by optimizing each new model to the residuals of the ensemble, gradually reducing the overall error. The

functional dependencies to predict model output of GBR are summarized in Eq. 8.

$$\boldsymbol{y} = F_{m-1}(\boldsymbol{x}) + \alpha_m h_m(\boldsymbol{x}) \tag{8}$$

with $F_{m-1}(x)$ representing the ensemble of previous models, α_m the learning rate for model m, and $h_m(x)$ represents the weak learner – typically a regression tree – trained to forecast residual errors.

For testing Bayesian inference also on non-linear models, gaussian processes in their regression type were applied. The Gaussian Process Regressor (GPR) is a non-parametric Bayesian approach for regression [38]. It models the underlying function as a probabilistic distribution of possible functions that could explain the provided data, assigning a probability to each function based on its wellness of fit. Mathematically, the predicted value for a given input is a Gaussian distribution, as shown in Eq. 9.

$$\boldsymbol{y} \sim \mathcal{N}\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\right) \tag{9}$$

In general, GPRs are features-efficient, producing valid process predictions with a low amount of input features.

Another approach represents the usage of neighbouring values to determine the best fit [39]. The k-NN in its regressor form predicts the target value by averaging the target values of its k nearest neighbors in the feature space, as presented in Eq. 10.

$$\boldsymbol{y} = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{y}_i \tag{10}$$

where $\boldsymbol{y}_i$ is the target value of the nearest neighbour i to $\boldsymbol{x}$, and k is the number of neighbors considered.

2.1.2. DL models

Considering the complexity of the AD process, DL models were applied and tested as well. While such models do not guarantee a global optimum [40,41], they often outperform simpler ML algorithms in time series prediction tasks [42] and might be faster when a Graphic Processing Unit (GPU) is used [43].

2.1.2.1. Multi-layer ELM. Extreme Learning Machine (ELM) is a type of feedforward NN capable of achieving high performances with low time and resource consumption [44]. ELM NNs do not use backpropagation as weight optimization method, but apply a Moore-Penrose generalized inverse matrix instead to compute the optimal solution for individual weight updates [45]. It has been proven that ELMs perform better when well-extracted hidden features are provided to the model [46], and therefore a Multi-Layer ELM (ML-ELM) was tested on the available datasets. ML-ELM uses a multi-layer structure where the hidden neurons assume the weights of the hidden layer in an ELM autoencoder, following Eq. 11.

$$\boldsymbol{a} = \boldsymbol{\beta}^T \text{ with } \boldsymbol{\beta} = \boldsymbol{K}\boldsymbol{T} \text{ and } \boldsymbol{K} = \left(\boldsymbol{H}^T\boldsymbol{H} + \frac{\boldsymbol{I}}{C}\right)^{-1}\boldsymbol{H}^T \tag{11}$$

where $\boldsymbol{K}$ is the pseudo-inverse of the ELM-based autoencoder's hidden layer output matrix $\boldsymbol{H}$, $\boldsymbol{T}$ is the output of the previous layer and C is the regularization constant. The input and hidden weights $\boldsymbol{W}$ and $\boldsymbol{V}$ are then updated based on $\boldsymbol{\beta}$. The output $\boldsymbol{y}$ is then calculated as in Eq. 12.

$$\boldsymbol{y} = g(norm(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{H}))\boldsymbol{\beta} \tag{12}$$

with g being the activation function and norm being the layer normalization.

2.1.2.2. LSTM and GRU. Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) that are designed to remember information for extended periods of time. They were introduced by Hochreiter and Schmidhuber [47] to address the vanishing gradient problem, which is a difficulty encountered by traditional RNNs where the contribution of information decays geometrically over time.

As any other NN, LSTMs have a predefined number of neurons distributed into layers, that apply sequential operations to the input value, multiplying weight of each neuron to predict the output values. Each LSTM neuron has an input gate, a forget gate, and an output gate [48], as illustrated in Fig. 1. These gates collectively decide how to update the current cell state and what information to output. The forget gate decides which information should be discarded and the input gate updates the cell state with new information. The cell state is the component of the cell which is responsible for the propagation of the information from the previous steps. The output gate defines the next hidden state.

Gated Recurrent Unit (GRU) NNs are a variation of LSTM NNs where the gating mechanism is simplified by merging the forget and input gates into a single update gate, and combining the cell state and hidden state into a reset gate [49], as shown in Fig. 1. This simplification reduces the computational complexity of the model, while still allowing it to capture long-range dependencies in sequential data.

A hyperparameter belonging to the present model application is the choice of LSTM or GRU cell for the RNNs model category.

2.1.2.3. 1D-CNN. One-Dimensional Convolutional NNs (1D-CNN) are a type of NN that is able to recognize patterns in the provided data. Unlike their 2D counterparts, which are commonly used for image processing, 1D-CNNs operate over a temporal sequence, looking for hidden patterns and relationships within the available data.

The key component of 1D-CNNs is the convolutional layer that uses a 1D convolution operation. In this operation, a filter (or kernel) slides across the input time series to extract local features. The size of the filter, often referred to as kernel size, is a sensitive hyperparameter [50]. This operation involves element-wise multiplication of the filter with a window of the input time series, and then summing the results to produce the output. Convolutional layers are normally accompanied by pooling layers – especially MaxPooling layers, that return the highest value out of a specific set of values – that reduce the size of the input value, conserving the most important information.

In the current application, the CNN is structured with two alternated layers of convolution and MaxPooling, followed by either a dense layer or a LSTM layer, depending on the choice of the optimizer for further data processing.

2.1.3. SARIMAX

The Seasonal ARIMA with eXogenous inputs (SARIMAX) is a time series forecasting method that extends ARIMA by incorporating seasonal components and exogenous variables [51]. It models the relationship between the observed series and its lagged values, seasonal differences, and external predictors. In the present study, the selected external predictors are set with the Minimum Redundancy Maximum Relevance (mRMR) algorithm [52], and the number of external predictors is defined within the applied optimization pipeline presented in Section 2.3. The fundamental equation for SARIMAX can be expressed as shown in Eq. 12.

$$\boldsymbol{y} = c + \varphi + \vartheta + \Phi + \Theta + \boldsymbol{X_t}\beta + \varepsilon_t \tag{12}$$

with c being a constant, $\varphi, \vartheta, \Phi, \Theta$ being autoregressive, moving average, seasonal autoregressive and seasonal moving average components, respectively. Furthermore, $\boldsymbol{X_t}$ represents the exogenous variable(s) and β their coefficient, and ε_t the white noise. Seasonal parameters were implemented, since the biogas production follows a seasonal-like pattern, represented by regular sections in the dataset, due to the regularity of the feed times in the tested datasets. Due to the impossibility for such a model to predict the methane yield without information on variable amount of substrate fed, exogenous variables were implemented as well.

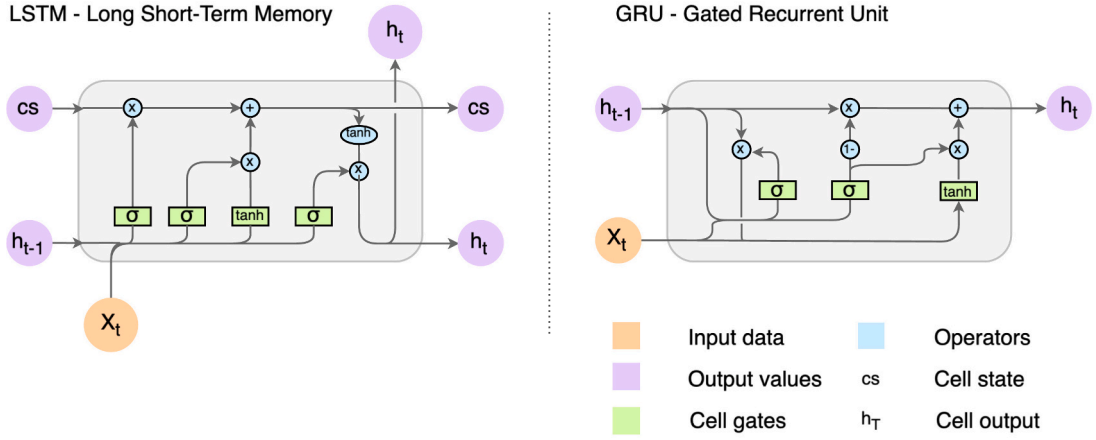


Fig. 1. Cell structure of LSTM and GRU NNs.

2.2. Feature importance

The SHAP method was applied to evaluate the importance of model features within the prediction algorithm [53]. The SHAP algorithm assigns each feature an importance value in predicting the output of a model by considering its contribution to the difference between the actual prediction and the expected prediction. It calculates these contributions using cooperative game theory principles [54], specifically the Shapley value (φ), which evaluates the marginal contribution of each feature in all possible combinations, ensuring fair allocation of credit among features.

2.3. Data preparation and Hyperparameter optimization

Effective preparation of training and test data is essential for the successful application of ML algorithms in scientific research [55]. Based on the optimization procedure proposed in Meola et al. [24], a Genetic Algorithm (GA) is designed to identify optimal conditions for data preparation and prediction model hyperparameter estimation, independently from the applied model. A simplified scheme of the applied pipeline is summarized in Fig. 2. The GA was set with 50 generations, 10 parents mating and a parents retention rate of 80 %. The gene mutation was set as adaptive, with a gene mutation probability of 20 % for gene values providing worse results, and a gene mutation probability of 8 % for good performing gene values (Marsili-Libelli and Alba, 2000). This optimization pipeline is applied to all the tested models and datasets.

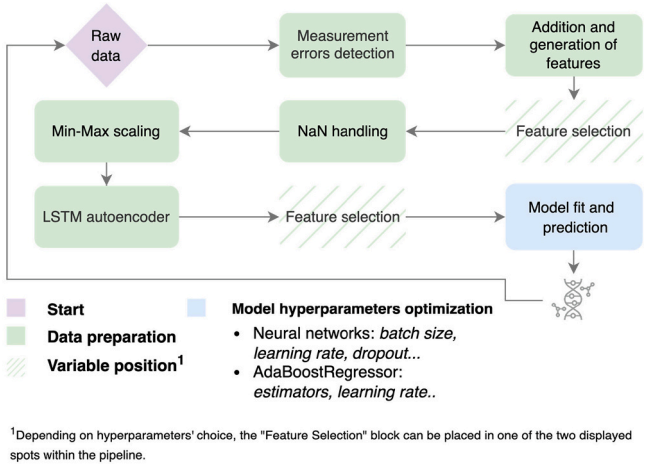


Fig. 2. Data preparation and model hyperparameter optimization pipeline. Adapted from Meola et al. [24].

The applied optimization pipeline includes a heuristic procedure for detection of impossible values, feeding correction, isolation forest algorithm for measurement error detection, an autoencoder for creation of additional input features and a mRMR algorithm for feature selection, among others. The model hyperparameters available to the optimizer are summarized in Table A1 and A2, while the data preparation parameters are presented in Fig. 1 in Meola et al. [24].

For process improving of missing data handling, the used data optimization procedure was extended with a soft sensor to include an additional option for NaN handling. The soft sensor is based on BR, ABR or GBR (see Section 2.1.1), and it switches to the next model in case an individual algorithms outputs invalid results. The structure is summarized in Algorithm 1.

Algorithm 1. Simplified structure of soft sensor for imputation of unknown values

Algorithm 1: Simplified structure of soft sensor for imputation of unknown values

Data: Table with missing elements (df)
Result: Filled up table, timesteps with no possible imputation are discarded
df_valid ← df with removed columns with more than 20% NaNs
df_no_nans ← df_valid (only train) with removed NaN rows
for feature in df_valid:
    df_fit ← df_no_nan without selected **feature**
    df_fit ← removal of redundant/irrelevant features with mRMR
    temp_df_valid ← df_valid
    input_df ← df_valid only with features selected in step 5
    target_idx ← timesteps where the **feature** is NaN
    model ← BayesianRidge/AdaBoostRegressor/GradientBoostingRegressor is fitted using df_fit as input and the **feature** from df_no_nans
    df_valid ← model generated missing values are added
df ← generated model missing values are added
nans_features ← df columns with more than 20% NaNs
for feature in nans_features:
    df_fit ← df_valid (only_train) in the timesteps where **feature** is valid
    df_target ← **feature** (only train) where valid
    df_fit ← removal of redundant/irrelevant features with mRMR
    model ← BayesianRidge/AdaBoostRegressor/GradientBoostingRegressor is fitted using df_fit as input and df_target as output
    df ← model generated missing values are added
end

The soft sensor is designed to impute missing values using a model trained on the existing training values at the same timestep.

Another modification from the original pipeline consists of the possibility of the feature selection to be either after the addition of features or directly after the LSTM autoencoder (shown as Variable position in Fig. 2). Feature selection can be performed using two different

approaches. The first approach applies feature selection to the original dataset without NaN handling and autoencoder features, thereby automatically adding all autoencoder-generated features by default. The second approach involves performing feature selection after the NaN handling and the addition of the autoencoder features. This method may potentially improve the feature selection process by excluding less relevant autoencoder-generated features. This leads to an additional data preparation parameter (0 = feature selection is positioned after LSTM autoencoder; 1 = feature selection is positioned after the addition of features) to be optimized by the GA. The feature selection is performed through a mRMR algorithm [56], as in the original description of the implemented pipeline.

2.4. Sensitivity analysis on pipeline hyperparameters

A variance-based SA was performed on data preparation parameters and model hyperparameters of the optimization pipeline (see previous Section). The applied SA technique – also referred to as Sobol' method – quantifies the contribution of each input variable to the overall output variance of a model [57]. In this case, the model input are the data preparation parameters and the model hyperparameters, while the output is the prediction error on the validation dataset. The validation dataset was chosen over the test error because the optimization pipeline is optimized on the validation error. Thus, using test error as output values could lead to invalid results. The Sobol' method involves a quasi-random sampling from the original input data distribution and an estimator based on Saltelli et al. [58] to calculate the Sobol Total-Effect indices. Since the standard number of samples (1024) originates 28,672 input parameter combinations, it was impossible to run such a high number of simulations. Therefore, an RF Regressor – described in this paper as surrogate RF – was fitted on the pipeline data and was used for the calculation of Sobol indices. The surrogate RF model was optimized first with a random search, and the results were further improved with a grid search. Each surrogate RF model was trained on one optimization instance at a time (e. g., BR predicting dataset A at a resolution of 15 min and an OD of 24 h, according to Table 2).

2.5. Data availability

To evaluate the tested algorithms on several, realistic scenarios, three full-scale datasets for AD of agricultural substrates and residues were chosen. Each dataset includes different operating and process conditions to allow for a comprehensive analysis of the model performances. Database A represents a steady state AD process with constant substrate characteristics and many available features ($n = 51$). Dataset B includes less sensor features, but more in general ($n = 57$) with highly variable OLR and high measurement uncertainty. Dataset C uses several substrates at various OLR, with 93 available features. All the experiments were carried on in a single primary continuous stirred tank reactor (CSTR) with a total volume of 188 m^3 at the Deutsches Biomasseforschungszentrum (DBFZ, German Biomass Research Center) research biogas plant. The data availability and the progression of the OLR for each full-scale experiment is illustrated in Fig. 3 and Fig. 4, respectively.

2.5.1. Dataset A

AD of rye whole crop silage (solid substrate) and cattle manure (liquid substrate) were conducted at steady-state conditions with a constant OLR of approximately 4 kg VS m^{-3} d^{-1} and an average HRT of 30 days for 189 days. The reactor was fed on average with 1.4 t of solid substrate and 2.8 m^3 of liquid substrate per day, with an average liquid to solid feed ratio of 2:1 m^3 t^{-1}.

2.5.2. Dataset B

In Dataset B dynamic AD of corn silage (solid substrate) and cattle manure (liquid substrate) was investigated with variable OLR between 2 and 12 kg VS m^{-3} d^{-1}, with peaks of 16 kg VS m^{-3} d^{-1}, for 165 days. The one-week mean HRT varied between 22 and 97, with a mean of 40 days. The high OLR towards the end of the experiment led to process inhibition.

2.5.3. Dataset C

In dataset C, the conditions were dynamic with a mean organic loading rate of 3.6, and a maximum of 8.9 kg VS m^{-3} d^{-1}. HRT varied between 23 and 66 days with peaks of 90, 120 and 140 days. The reactor was fed on average with 4 t of a mixture of solid and liquid substrate per day, with a highly variable liquid to solid feed ratio between 81.3 and 0.3 m^3 t^{-1}. Liquid substrate was uniquely cow manure, while solid substrate variated among apple pomace, sugar beet, grass silage and corn silage.

The measurements available in each dataset (including data resolution) are summarized in Table 3.

2.6. Data manipulation

All three datasets used for simulations were resampled (up sampled when feature resolution was lower than the desired one, down sampled when the resolution was higher than the desired one) to 15 min, 1 h and 6 h resolution. Depending on the feature, the down sampling was carried out by averaging the values over time (such as methane yield or VFAs concentration in digestate) or summing the values over time (such as feeding amounts).

While RNNs, 1D-CNN and SARIMAX natively support a third time dimension for consideration of previous time steps, all the other applied models support only one timestep at a time. Thus, all the other applied models support a sliding window as long as the sequence length. For such models, an additional data preparation parameter has to be added and optimized by the GA in the optimization pipeline (see Section 2.3), which is the sequence length for the output feature, which can assume the same values as the sequence length.

2.7. Implementation

The simulations were carried out on three different High-Performance Clusters (HPCs). All the simulations involving neural networks were carried out on an HPC with one AMD EPYC 7551P CPU with 64 threads and a maximum clock of 3 GHz, and with one Nvidia GeForce RTX 2080 Ti encompassing 11 GB of RAM as GPU. 15 min resolution simulations were carried on with 40 GB RAM, 1-h resolution with 30 GB and 6-h resolutions with 20 GB RAM. All the other simulations were either executed on a HPC with 2 Intel Xeon E5–2698 V3 CPUs with 64 threads, a maximum clock of 3.6 GHz and 110 GB of RAM in total; or they were executed on a HPC with two AMD EPYC 7713P CPU with 128 threads and a maximum clock of 3.7 GHz each, and a RAM distribution following the first HPC. To ensure a comparability while analyzing the execution time, the optimization pipeline was programmed in Python 3.9, but is compatible with any python version from 3.7 to 3.11. The pipeline makes use of several packages. Pandas (McKinney, 2010) and Numpy (Harris et al., 2020) were used for the data preparation processes, Scikit-learn (Pedregosa et al., 2011) for both data preparation process and prediction model. Tensorflow (TensorFlow Developers, 2022) was applied for the autoencoder and for the prediction model, while Pygad (Gad, 2021) was used for the development of the GA optimizer. Each optimizer iteration (data preparation, model train and model test) returns a $7 \bullet 10^7$ error when the process takes more than 30 min. SHAP [53] was used for the calculation of feature importance and SALib [59] was applied for the calculation of Sobol' indices.

3. Results and discussion

In the following section, the simulations results as well as prediction performances, time performances, hyperparameters sensitivity and

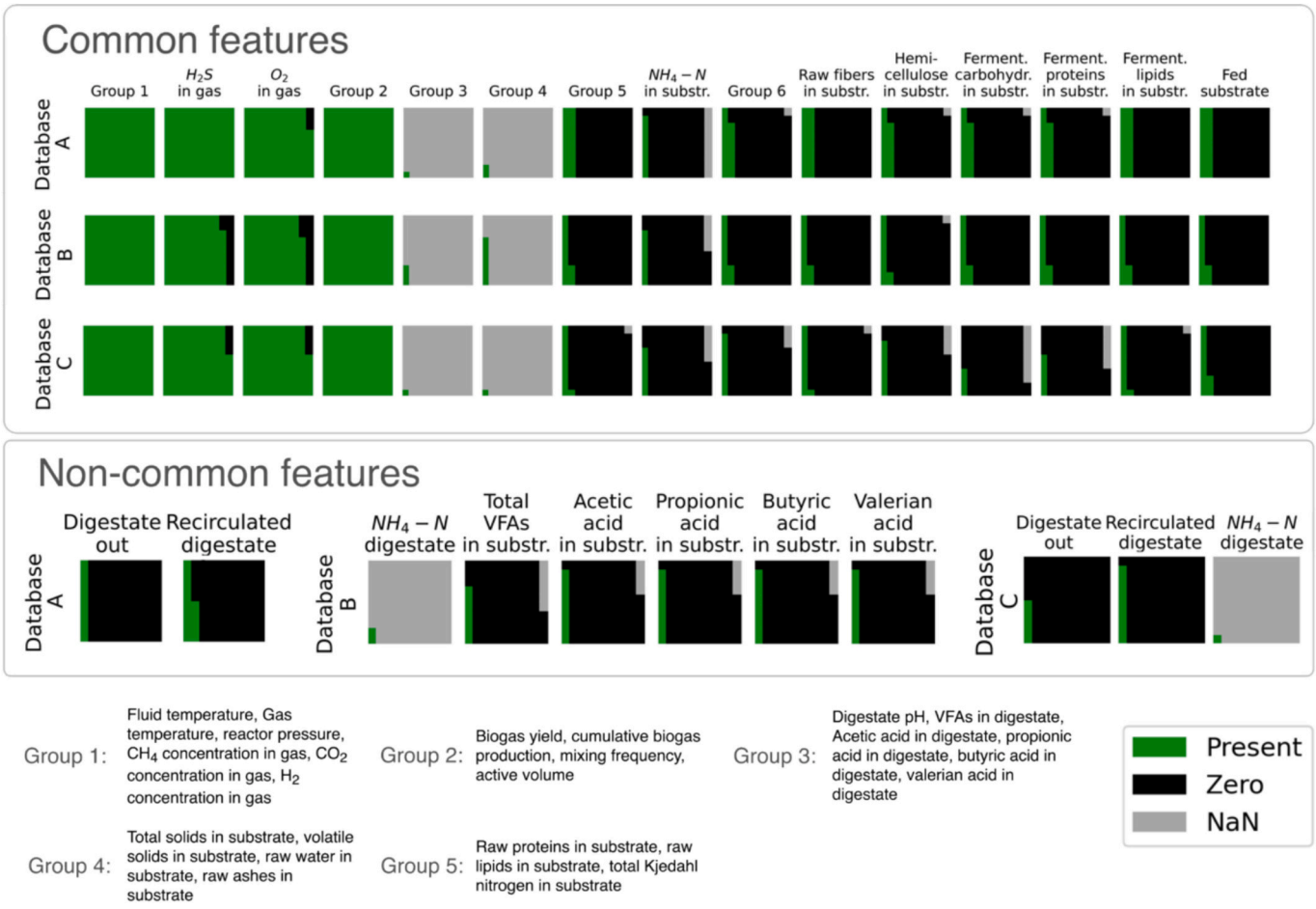


Fig. 3. Data availability in the three datasets.

feature importance will be evaluated. The prevailing error metric applied for measuring prediction performances is the Root Mean Squared Scaled Error (RMSSE), as described in Meola et al. [24]. Thus, the RMSSE is calculated as the ratio between the RMSE of the applied model, and the RMSE of a naïve forecast, that predicts the next output value as corresponding to the present output value.

3.1. Model performances

The 13 models were tested on all available datasets, resolutions and ODs. In this section, validation and test errors on all prediction scenarios will be presented and discussed. While prediction error alone is not a reliable parameter to measure how accurate, in general, ML algorithms can be when modelling physical-chemical scenarios, they can give an impression on how powerful such models are in comparison to each other.

The best result per optimization scenario is meant as the best hyperparameter combination for a specific model that produced best performances on the validation dataset. In general, models perform better on the test dataset in comparison with what is shown in Fig. 5, Fig. 6 and Fig. 7, but it would not be scientifically correct to show the best hyperparameter combination originating best performances on test dataset, since in a real scenario the test dataset is unknown. Fig. 5 shows individual results for dataset A.

Among all resolutions and ODs in dataset A, GPR and PLS perform regularly worse than the other models. Instead, ABR is able to perform comparably to the other models only in 50 % of the occasions. The ML-ELM performs better at a 15 min resolution when the OD is higher, while it performs comparatively worse especially at OD of 12 h.

Models perform in general better at 1 h resolution compared to 15 min resolution, except when the OD reaches 24 h. At that point, LSTM/GRU and LR perform better at 15 min resolution.

For the simpler task of predicting 1 h in advance, LR performs better on the test set than all the other models, with 61 % as RMSSE error. At an OD of 12 h and 24 h, EN performs better than LR, probably because the L1 and L2 regularization allow the model to better understand underlying relationships between input and output data. In general, linear models seem to perform better than any other more complex model, as also demonstrated by Dittmer et al., (2021b). BR, LR and EN always outperform any other model at 15 min resolution and BR and EN are in the top three performing model in all the OD scenarios at 1 h resolution. Another notable model in this context is RF, which is among the three best performing models in all OD scenarios at 1 h resolution. While more complex models, such as XGBoost, RNNs and 1D-CNN show acceptable performances, their complexity is unnecessary to reach optimal prediction performances in processes without inhibition or significant variations in the fed substrate amount. Since the performances of the model were already optimal at 1 h resolution, experiments on 6 h resolution were not performed.

While for the dataset A most of the models were able to predict the biogas production at any OD and resolution, model efficiency for dataset B is much lower, as illustrated in Fig. 6.

The only resolutions and ODs at which the models are able to predict the biogas production with high precision are at 1 h resolution and 1 h OD and at all ODs at 6 h resolution, while no model can satisfactorily predict the biogas production at any OD at 15 min resolution. BR shows in every condition signs of overfitting on the validation data, with very high differences between the test and validation error. The multi-layer

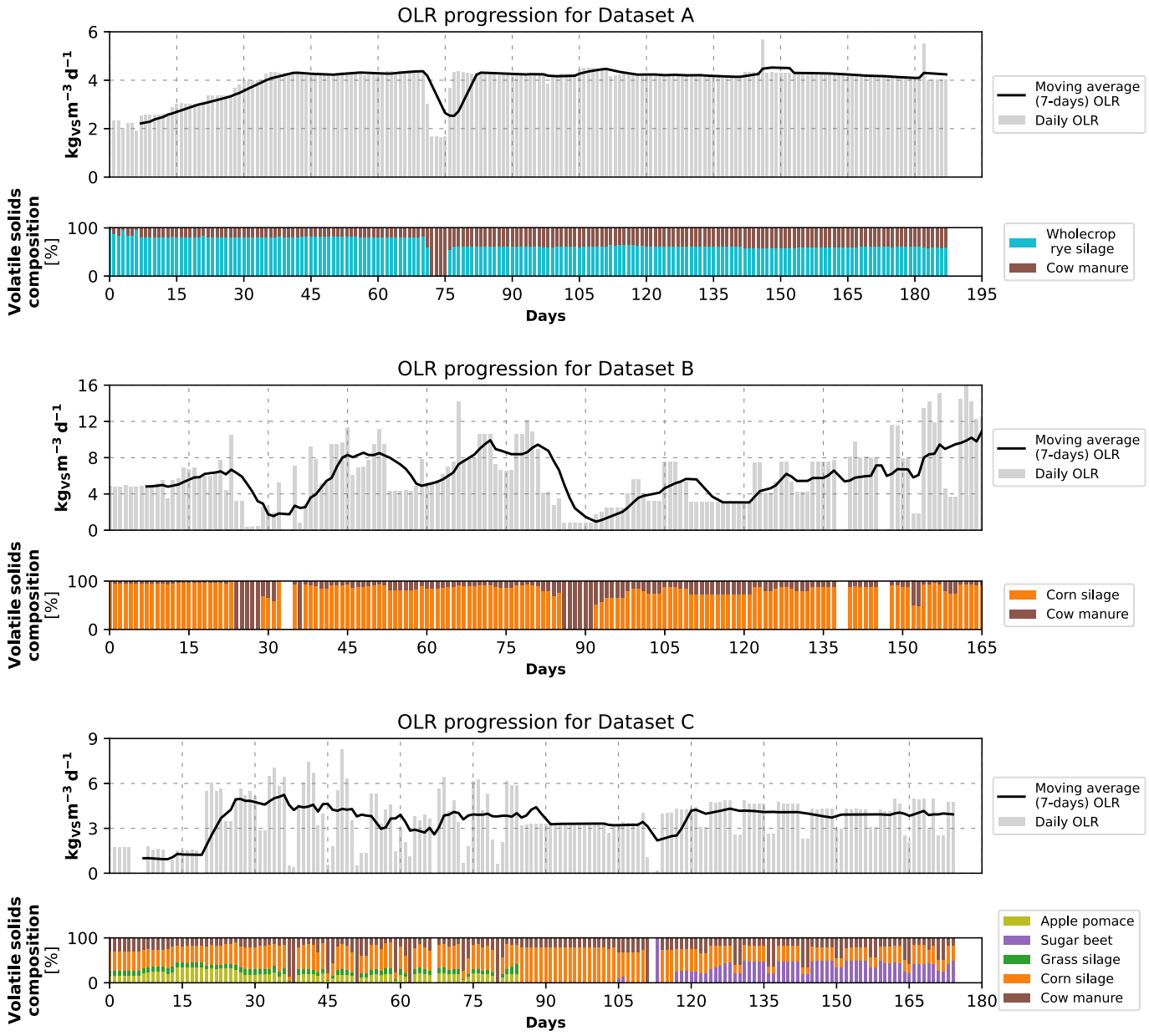


Fig. 4. OLR progression and VS composition for dataset A, B and C.

ELM, LR and EN also show clear signs of overfitting in 50 % of the cases. GBR is the best model when predicting biogas production 1 h in advance with 1 h resolution, with LSTM/GRU showing similar performances. At 6 h resolution, RF is the best model for ODs of 1 h and 12 h, while k-NN is the best model with 24 h OD. It is the only model able to predict biogas production at 24 h OD together with the SARIMAX algorithm. In general, more complex models – ABR, RNNs, RF Regressor – together with k-NN Regressor demonstrate higher performances than simpler models. However, 1D-CNN was not able to match the performances of other complex models, demonstrating to be unfit for the prediction of the dynamic methane yield at non-stationary conditions (for the specific substrate mixture and operating conditions). Therefore, complex models are required for modelling methane production during dynamic (unstable) conditions, even when the same substrate is fed. Sappl et al. [19] also demonstrate that more complex models are required for optimal process prediction, while the authors observed how k-NN performed comparably worse than the applied transformer architecture.

While the dataset C offers a challenge due to the frequent variation of substrates and the variability in OLR, at least one of the tested algorithms is always able to successfully predict the biogas production, as shown in Fig. 7.

In general, bias and variance alternate themselves in the prediction of this dataset. Test results are often more accurate than validation results. This phenomenon could occur because of the high number of substrates used and the substrate difference between train, validation and test dataset. In general, linear models (especially BR and LR) together with RF overperform all the other algorithms, in particular at lower OD, where those three models are always among the top three performers. At 12 h OD, RF does not perform as well as at one timestep OD, especially at 15 min resolution. Gradient Boosting Regressor can better predict methane yield at 12 h OD, as well as the RNNs, that performs better at 1 h resolution and 12 h OD, and is the best performing model at 6 h resolution, at same OD. At 24 h OD, the trend of RF losing prediction accuracy keeps being valid, especially at 15 min resolution, while the linear models keep providing optimal performances at any resolution. RNNs are among the best performing models along all time resolutions, being the absolute best performer at 1 h resolution. Thus, linear models and RNNs in particular are adequate models for predicting

Table 3
Available measurements for dataset A, B and C.

Type of feature	Resolution	Database	Amount	Examples
Operational (1 s)	1 s	A and C	8	Biogas production rate and temperature, reactor pressure and temperature, mixing frequency, recirculation, output digestate, active volume
		B	6	Same as in Dataset A, excluding recirculation and output digestate
Operational (2 h)	2 h	A, B, C	5	CH_4, CO_2, H_2S, O_2 and H_2 biogas content
Substrate	1 min[1]	A	28 (14 per substrate)	Laboratory analysis carried out on the two substrates (raw ashes and fat content among others)
		B	36 (18 per substrate)	Laboratory analysis carried out on the two substrates (raw ashes, fat content and VFAs content among others)
		C	70 (14 per substrate)	Same as Dataset A.
Digestate	1 to 7 days	A, B, C	10	Laboratory analysis carried out on the digestate (raw ashes, fat content, TS, VS and VFAs content among others)

[1] While the analyses on the substrates are performed only when the fed substrate is changed, the dataset is filled up with 0 s when there is no feed, and with the previous measurement at feed time.

dynamic methane production with changing substrate composition and high data availability. While the simplicity of linear models could allow them to appropriately model the methane yield through appropriate input values provided by the data preparation pipeline, RNNs are able to depict complex relationships between input and output also at variable substrate compositions.

Results of the two best performing models plus a selected model are shown in Fig. 8 for each dataset. The OD was kept at 24 h while dataset A is shown at 15 min resolution, Dataset B at 6 h resolution and dataset C at 1 h resolution.

During the prediction of dataset A, until day 13, all the models are able to closely follow and mostly match the methane yield, especially at positive and negative peaks. From day 14 (when the peak height increases) most models start losing precision of predicted methane yield, while a GRU-based RNN is still able to closely depict individual peaks and intermediate periods. This shows that, while error metrics are valuable for an initial model evaluation, the analysis of individual plots are a more meaningful way to understand which models are most suitable for process prediction.

Even when predicting a challenging dataset such as dataset B, individual models are suitable to describe the methane yield in detail. While during the first three prediction days all the three considered models are able to predict the decrease in methane yield, the only model able to predict the renewed increase in yield is SARIMAX. From day 5, SARIMAX still follows closely the measured methane yield, but often predicts the peaks 6 h (one timestep) later than the real ones. While this does not mean that the model fell in a local minimum and uses the previous time step value, since the closest known value is 24 h before, it still does not provide optimal performances for real time applications. The other two models applied, and specifically RF, are able to closely follow measurements even on peaks from day 6 onwards.

While dataset C was trained on process data that involved the feeding of substrates different from the ones fed during the test data process, all selected models are suitable to satisfactorily predict the methane yield 24 h in advance. In general, RF matches better the peaks, even when the peaks are lower it tends to overestimate the methane yield. Moreover, RF is able to properly depict feeding events that result in three subsequent peaks (days 13 and 14 in Fig. 8), while the other two models lack in precision in these cases. BR is instead more prone to closely follow peaks of lower intensity but underestimates higher peaks. GRU RNN is instead more equilibrated, mostly predicting peaks correctly. RF demonstrates to be the most reliable model for model prediction with presence of multiple substrates, while the other two models can also be useful depending on the use case.

3.2. Optimization time performances

Apart from prediction accuracy, the time needed for algorithm convergency is a factor in deciding which algorithms are most suitable for process prediction, since powerful high-performing computing technologies might not always be available, and a low training time could be important for efficient operation procedures. The time performance of the algorithms was tested as percent increase or decrease of a specific algorithm compared to the other algorithms in the same optimization instance. The mean increase or decrease (as well as the standard deviation) along all simulative environments is illustrated in Fig. 9.

The fastest algorithms on average, considering the total time needed for reaching convergency, are k-NN Regressor as the fastest, followed by Multi-Layer ELM and PLS Canonical. The slowest algorithm on average are RNNs (as confirmed by [60]), followed by CNNs, GBR and SARIMAX. While the high convergency time for the first three presented model is typical of those models [61,62], the slowness of SARIMAX is determined by the implementation, which generated and trained a new model at each timestep.

Most of models' speed can be explained by model complexity. The only models whose time performance seems affected by convergence time more than by model complexity are CNN and GBR. The high number of convergency steps required by CNN – higher on average then the steps needed for RNN to converge – is related to the high number of parameters optimized in CNNs, and probably by the sensitivity of most of model hyperparameters such as the kernel length and the pooling length. GBR is probably highly sensitive to input data, since there are only four model hyperparameters to optimize.

Moreover, not all models are consistent with their time performances. The three best performing models are also the most consistent in their time performances, and the three worst-performing models are also the ones which are the least consistent. Simpler models such as LR and BR also suffer from high relative standard deviation, probably due to being too sensitive to the input data.

3.3. Further evaluations of applied models

Effectiveness of applied models might also differ depending on the individual optimization instance (dataset, data resolution and OD). Fig. 10a shows the average performances of the applied models depending on the underlying data resolution. The majority of linear (or pseudo-linear) models, BR, LR and ML-ELM demonstrate an extremely high difference in performances depending on the resolution used, with ML-ELM showing an RMSSE increase of more than 600 % between 6 h resolution and 15 min resolution. In general, models perform better at

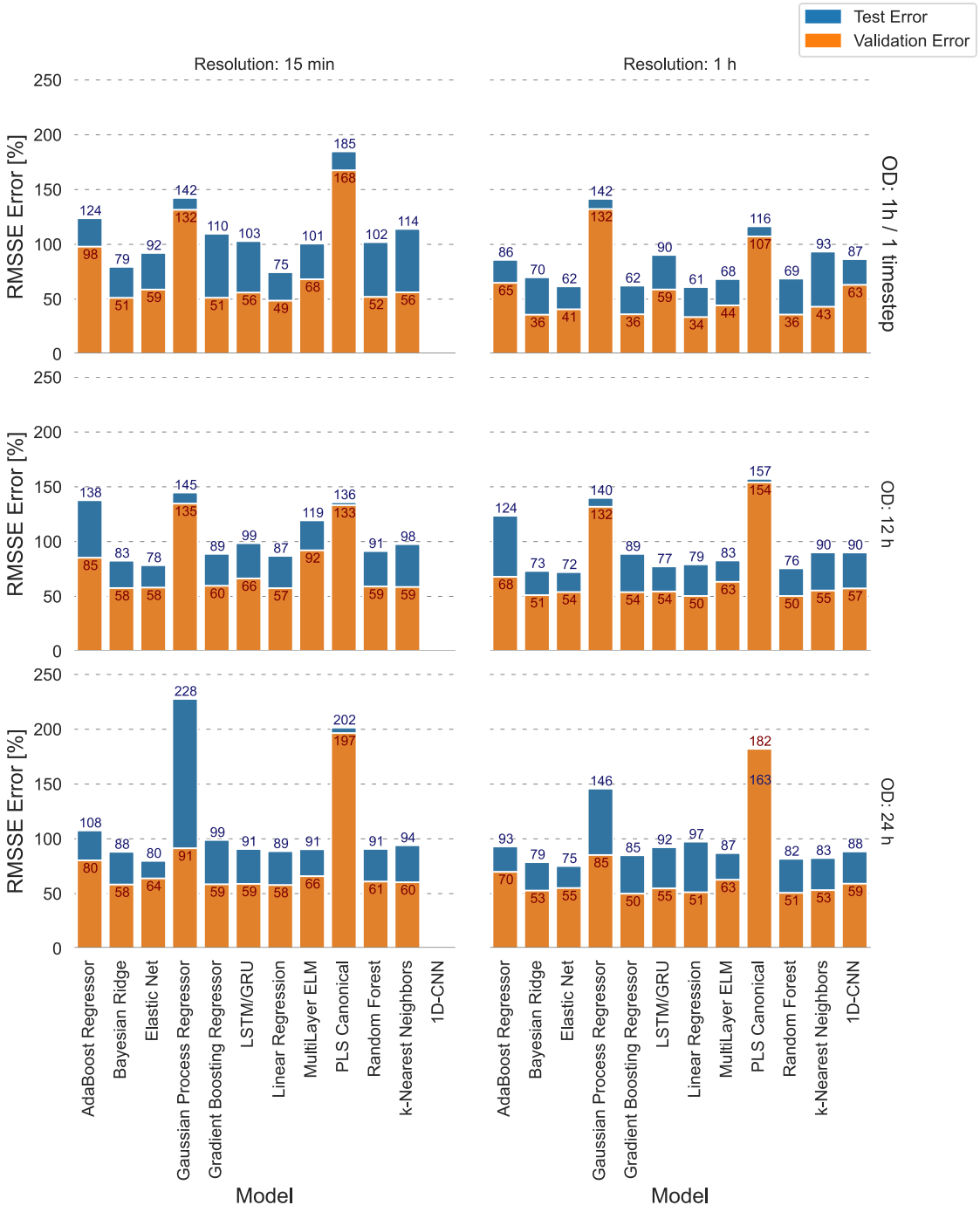


Fig. 5. Best performing model configurations validation and test error for Dataset A. The shown test errors correspond to the best validation result.

higher resolution, but GPR does not follow this pattern, demonstrating worse performances when applied to 6 h resolution cases. Nevertheless, multiple models perform better at 1 h resolution than at 6 h resolution (such as 1D-CNN and BR). This behaviour can be explained for linear models to have more data to be trained on, while for CNN it might be explained with the prediction mechanism based on pattern recognition, which might be hindered at lower resolution. In general, the best performing models are RNN, RF Regressor, k-NN Regressor and ABR. GBR performs better than all the other models at 15 min resolution, confirming its optimal performances also at 1 h resolution, followed closely by RF Regressor, RNN and k-NN Regressor. At 6 h resolution, RNNs perform best, followed by RF Regressor. The good performances of RNNs and RF on 6 h resolution might be attributed to the known effectiveness of these models to model phenomena with a low amount of data [14,12]. Fig. 10b shows model performances depending on the dataset on which the model is applied.

While it is evident from the results that dataset B is the hardest dataset to model, only simpler (linear) models show a big difference (RMSEE $>$ 600 %) between database A, B and C. BR, EN, LR and ML-ELM demonstrate to be inadequate to accurately predict dataset B, and GBR, PLS Canonical and 1D-CNN are also not performing well. The other models perform, in comparison, sufficiently well – specifically GBR, which performs best – whilst none of them scores less than 100 % RMSSE on average. As expected, performances of models on dataset C are lower than on dataset A, due to the more dynamic nature of dataset C, but some models (such as RNN and RF) perform better on the dataset C,

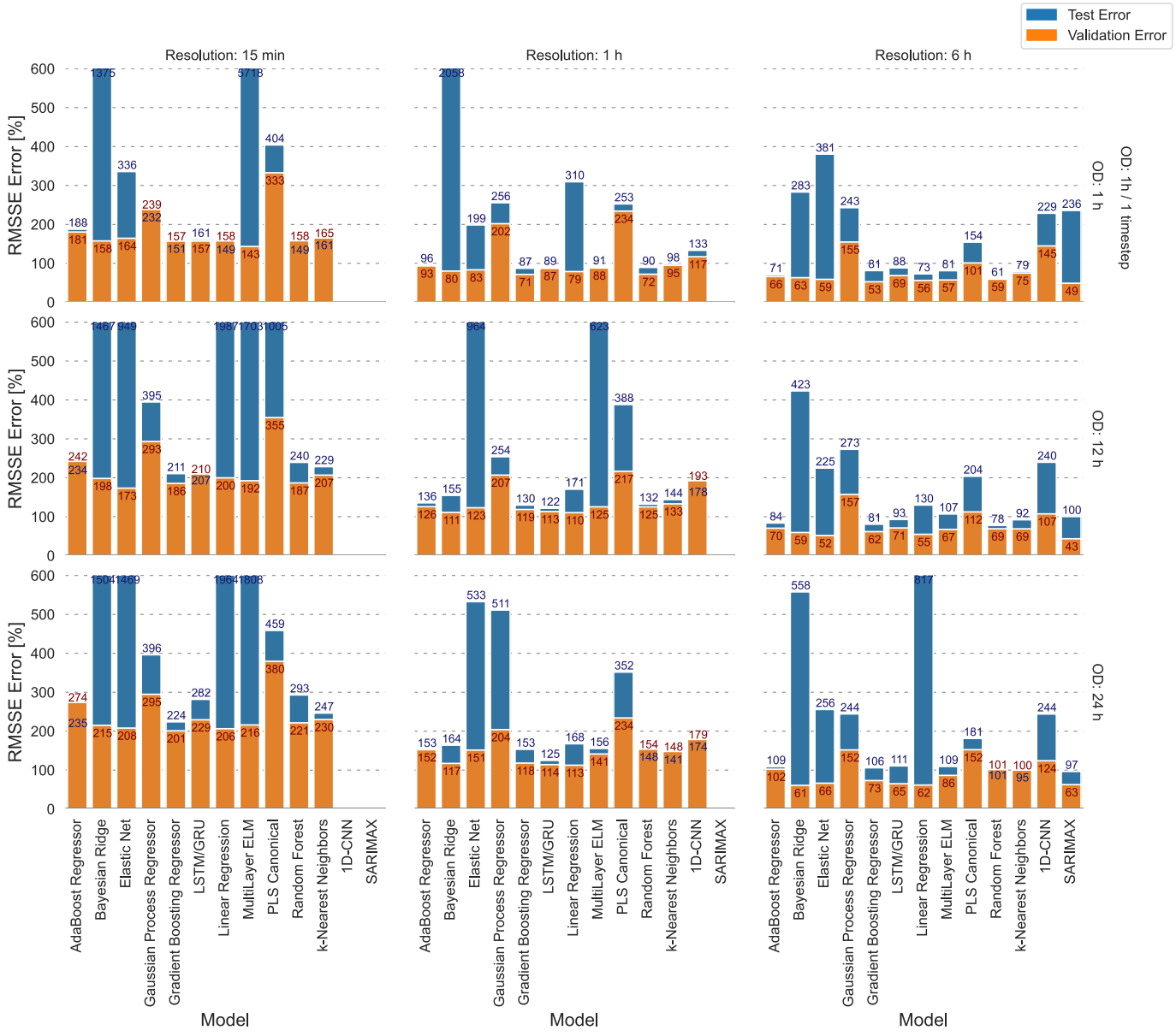


Fig. 6. Best performing model configurations validation and test error for Dataset B.

possibly because of the diversity of the training data, that allowed such models to capture more complex relationships between input and output data. All models, in general, perform well on dataset A and C, except GPR, ABR and PLS Canonical, which overcome on average RMSSE of 100 %.

OD also demonstrates to have a role in model performances. In Fig. 10c average model performances depending on OD are shown. Models that perform well on average, tend to perform better on lower OD (such as RNN and RF, as also demonstrated by [18]). The only exception is SARIMAX, which however was only trained and tested on 6 h resolutions datasets, meaning that the average error is based only on 6 h resolution datasets that are easier to predict than higher resolution datasets. 1D-CNN was trained only on 1 h and 6 h resolutions, meaning that the comparatively good performances (especially on 24 h OD) can be explained by the lack of testing on 15 min resolution datasets. BR, GPR, EN, PLS Canonical and ML-ELM all demonstrate to be inadequate to properly predict biomethane production at any OD. Excluding SARIMAX for the previously mentioned reason, RNNs are on average the best performing model on 24 h OD, followed by GBR. The complexity of those models probably allows them to understand intrinsic process proprieties, and predict longer steps in the future. GBR is the best performing models on 12 h OD, together with RF, both followed by RNN and k-NN. At 1 h OD, RF performs better than any other model, closely followed by GBR.

Moreover, the proneness of models to overfit or include bias is shown in Fig. 10d as a percentual difference between the validation and the test error, in absolute values. The majority of the models that did not perform well in the previous analyses, also reached bad scores in this evaluation. Complex NNs (1D-CNN and RNN) together with k-NN, ABR and RF Regressor perform well, demonstrating on average less than 30 % absolute difference between validation and test data. Considering the results of this analysis, those models can be considered more robust than the other ones. ML-ELM, the least complex NN, surprisingly exhibits a very high variance or bias, demonstrating that it should not be used as a model for the prediction of dynamic methane production rates.

For more precise evaluation of model performances, the MAE and MAPE were calculated for an OD of 24 h. Results are summarized in Table 4.

While not all applied model result in the same ranking independently from the analyzed error metric, models performing well typically

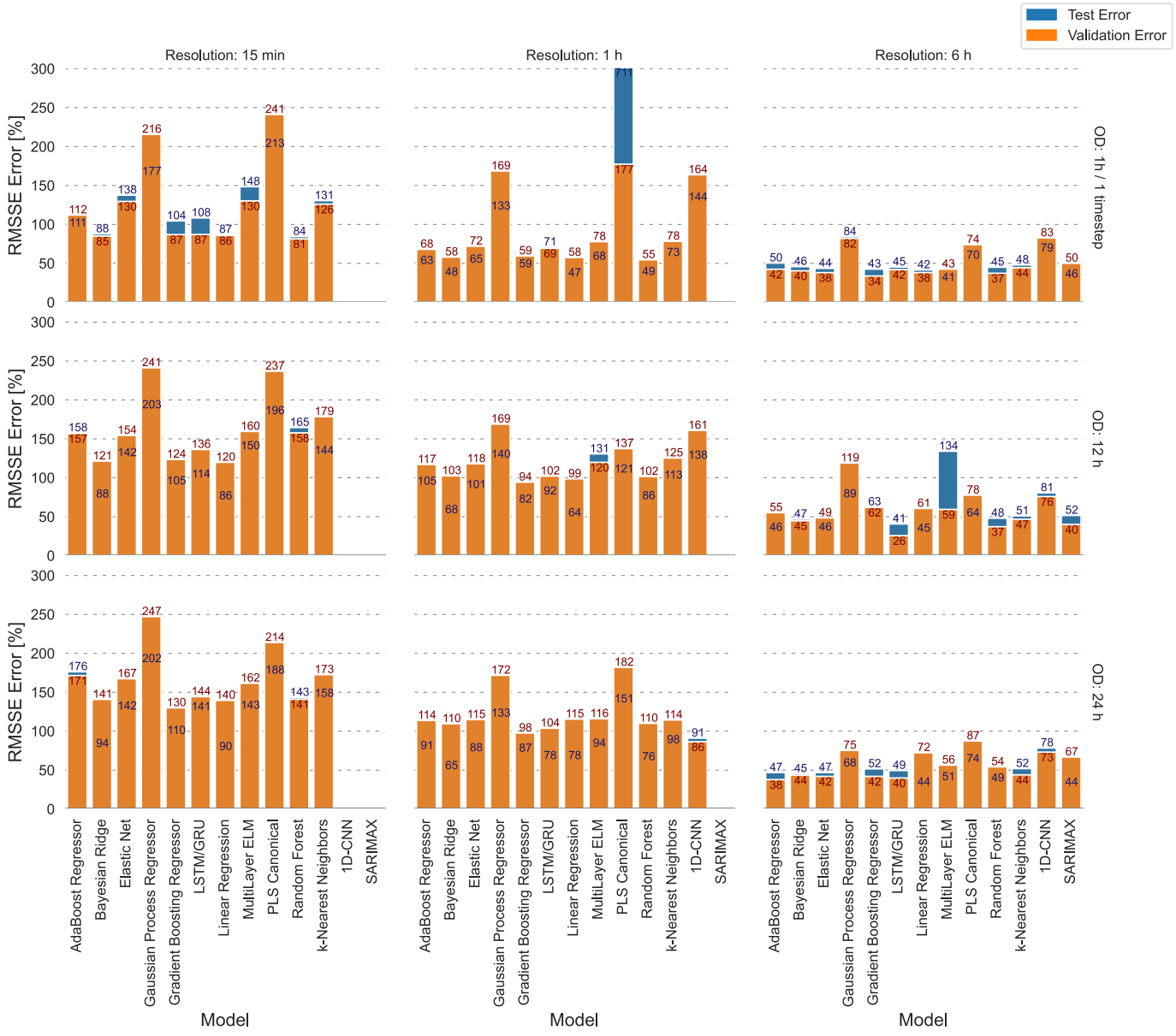


Fig. 7. Best performing model configurations validation and test error for Dataset C.

perform well for all error types. For instance, model predictions resulting in RMSSE between 80 and 94 when predicting dataset A at a resolution of 15 min, are also the top performing models when considering MAE and MAPE. In several cases, specifically for dataset B and C at a resolution of 15 min, and for dataset C at a resolution of 6 h a best model can be defined independently from the analyzed metric.

The differences in model rankings between error metrics can largely be attributed to two factors. First, the increased presence of peaks in certain scenarios tends to increase the RMSSE, whereas these peaks have a less pronounced effect on MAE and MAPE. Second, the similarity of the previous output when predicting the current one plays a role. When the previous value closely resembles the actual one, it provides an advantage to naïve forecasting, which is used as the baseline for RMSSE calculation. This can lower the RMSSE without impacting MAE and MAPE.

Given the importance of peak prediction and the predictive capabilities of naïve forecasts in assessing ML model performance in the context of AD, RMSSE is recommended for use in future experiments.

3.4. Sensitivity analysis

The SA was performed on the three best performing models per dataset, while still selecting 15 min resolution for dataset A, 6 h resolution for dataset B and 1 h for dataset C, all with 24 h OD. The results are shown for datasets A, B and C in Fig. 11.

The surrogate RF models fitted on dataset A, EN and RNN, evince an accuracy of 0.95 and 0.85, respectively. During the training of RNNs, model hyperparameters are more important than data preparation parameters [24]. While hyperparameters are important also when training EN, data preparation parameters are largely more important. This aspect can be explained by the more complex nature of RNNs, and by the need for linear models to have better quality data to deliver optimal results. In general, hyperparameters relative to autoencoders seem to have a high importance for both models, even if not for both models the same parameters are equally important. NaNs handling, excluded input variables and sequence length are among the most important data preparation parameters for both models.

The surrogate RF models fitted on dataset B, SARIMAX and RF, show an accuracy of 0.93 and 0.99, respectively. While both models

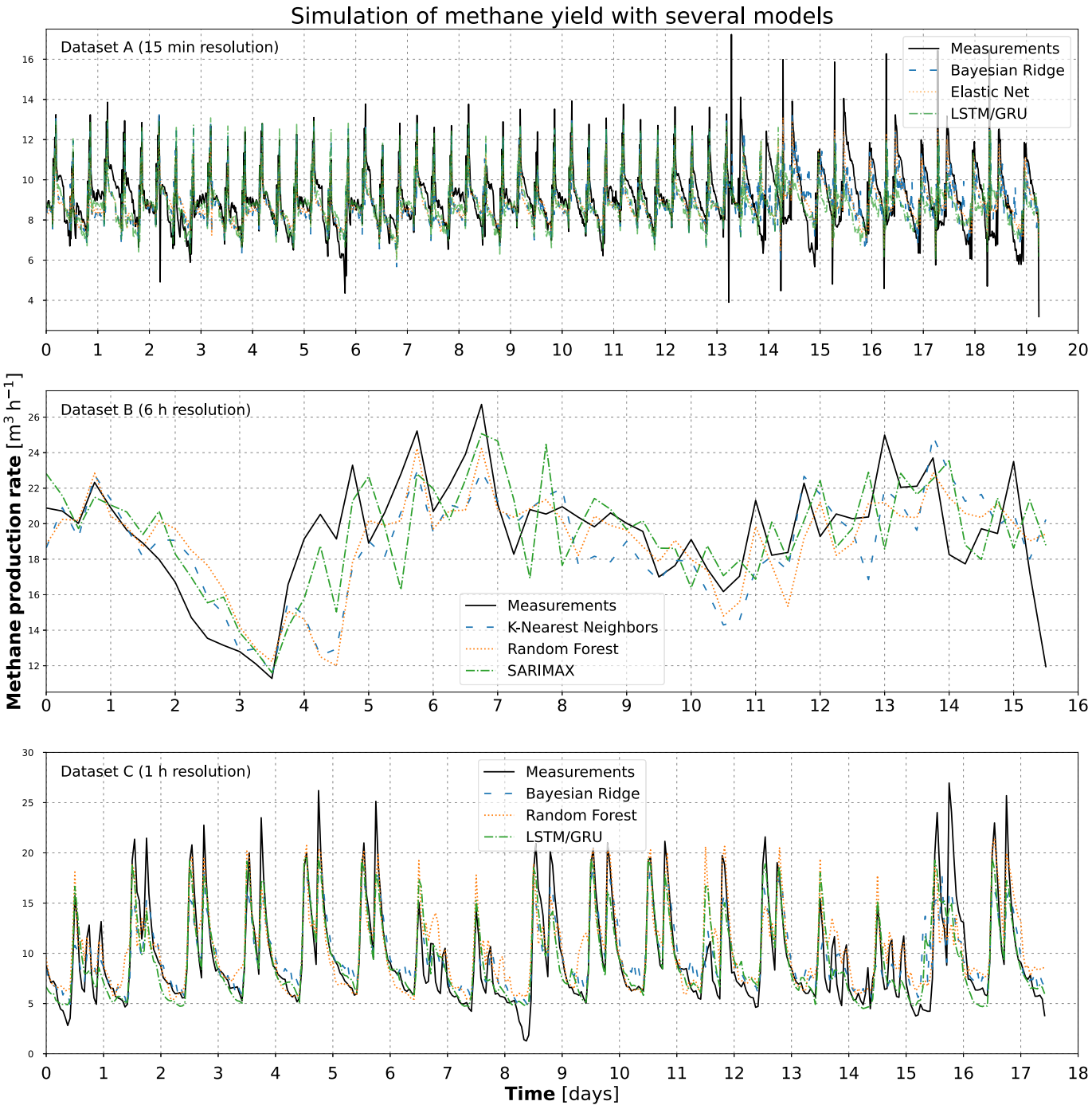


Fig. 8. Simulation results of best models on the test datasets, with an OD of 24 h. Dataset A tested on 15 min resolution, dataset B on 6 h resolution and dataset C 1 h resolution.

demonstrate acceptable performances when predicting dataset B, they work differently and they were therefore expected to be differently influenced by individual model hyperparameters. However, the models are similarly affected by the variation of the data preparation parameters. For both models, data preparation parameters are more influential than hyperparameters (particularly for RF, where the only optimized hyperparameter has a Sobol' index close to zero). In contrast, SARIMAX shows relevance for three of its hyperparameters. Additionally, autoencoder-related features and sequence length parameters (input sequence length for SARIMAX, output sequence length for RF) significantly impact both models. Although to varying extents, both models are also sensitive to the total feed usage and handling of NaNs. This shows that the same data preparation parameters have similar impacts on different models when applied to a non-stationary AD process with constant substrate input and composition.

The surrogate RF models fitted on dataset C, BR and RNN, had an accuracy of 0.98 and 0.92, respectively. As already described for dataset A, also in dataset C the RNNs are impacted more than the linear model by the model hyperparameters. The sequence length is important for both models, and both models also are impacted by autoencoders-related data preparation parameters, suggesting the importance of autoencoders for both models and more in general for all the cases taken into consideration. NaN handling is important for both models as well, showing a general trend for all applied ML or DL models. RNNs applied

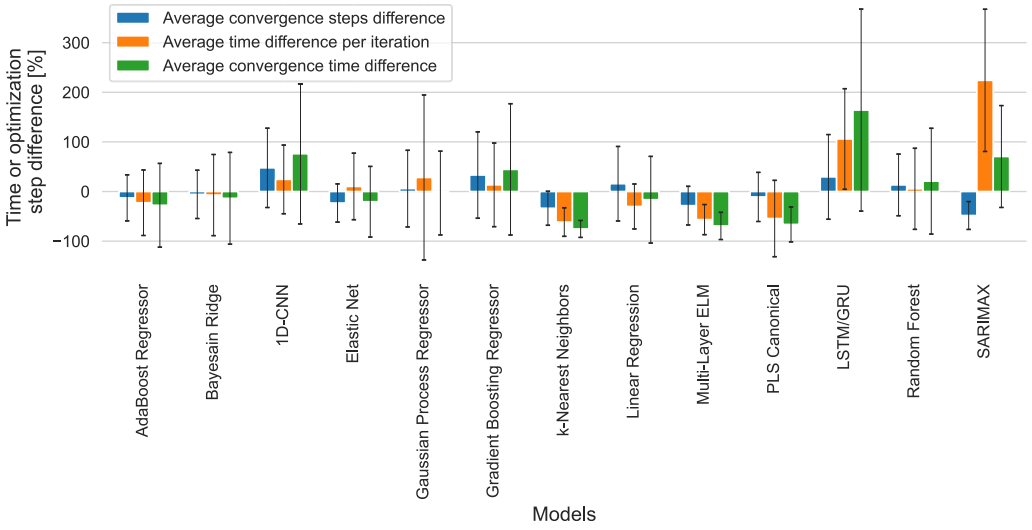


Fig. 9. Increase in time and optimization steps compared to same dataset, resolution and OD scenario per each model. The error bars represent the standard deviation.

to both dataset A and C are heavily impacted by the dropout value in the input cells and less by the removed input feature hyperparameters, suggesting that dropout in the input cells is more effective in limiting the number of input parameters for RNNs compared to the mRMR algorithm.

In general, complex models such as RNNs need a fine calibration of various hyperparameters for accurate process prediction, while simpler models require the tuning of a few data preparation hyperparameters in addition to their most impactful model parameters.

3.5. Feature importance

The FI was calculated for the best performing models in the same optimization instance as within the SA calculation process. Since all models use multi-step data, features were summed after the calculation of the SHAP values. While this process originated relatively color-uniform SHAP distributions for all models using rolling window technique (in this case, EN, RF and BR), RNNs originated less color-uniform distributions, since the contribution of each time step of the cell memory is regulated by the (hidden) state of the cell. Moreover, SARIMAX feature importance could not be analyzed with SHAP due to the different nature of the model. Thus, model coefficients are plotted instead.

Within the prediction of dataset A, the features with the highest importance for the two tested models (EN and RNNs) do not differ in general, as illustrated in Fig. 12.

While some features show a similar quantitative impact on the prediction for both models, such as the recirculated digestate, the HRT, the OLR and the mixing frequency, some other features have a completely different effect, such as the output digestate, that originate both strongly positive and negative effects in the RNNs, while it just originates positive effects in EN. Moreover, the methane production rate has a threefold positive effect in EN compared to the RNN, which explains why RNNs perform better than EN in the presence of higher peaks, as shown in Fig. 8 in Dataset A between days 7 and 9. RNNs seem to be affected by far more parameters than EN, demonstrating the capacity to handle a higher amount of significative input values. This can be deducted from Fig. 12 while observing high importance of the group of other features compared with individual features in both simulation instances. Physicochemical phenomena are mirrored in the FI of the analyzed models. The recirculation of the digestate has in both models a positive impact on the output, probably explained by the increased HRT and the gas exchange between liquid and gaseous phase in the reactor triggered by reactor agitation [63]. RNNs also depict the hydrogen content of produced gas as directly proportional to the methane yield, possibly explained by the hydrogen content of the produced biogas being an AD process indicator [64]. Other phenomena such as the negative impact of mixing frequency on the methane yield can either be present because of inherent process proprieties or because of wrongly fitted model weight [65]. The high importance of the hour of the day (especially for RNNs) is explained by the constant feeding time throughout the experiment. Thus, the model learned that at a certain time the model is fed, originating higher methane production. In general it is demonstrated that none of both models need laboratory analysis further than the VS of the input substrate for accurate process prediction. Instead, in presence of complex data or process inhibition, further laboratory analysis might be beneficial for accurate prediction. Fig. 12b shows the feature importance for SARIMAX and RF on the dataset B.

While an exact comparison of the FI of the two models cannot be extensively performed due to their different nature, both applied models can be singularly analyzed through their own FI applied techniques, and similarities and differences in FI can be found. In general, the HRT, as one of the selected exogenous inputs, has a higher impact than any other SARIMA coefficient when applied to dataset B. Additional factors have a negligible impact on the simulation results. The mixing frequency has also a very low importance because it is a constant term. The mRMR algorithm did not find any better feature in this case. HRT is also a relevant feature when the RF algorithm is applied, being the fourth most important feature. While the methane production rate of previous time steps was not included in the input parameters of SARIMAX since it would have been redundant, the fed corn silage was probably not selected by the mRMR algorithm for SARIMAX because its value is often zero. Moreover, OLR and HRT are mathematically related, and since the GA selected for this model only two exogenous features, OLR was excluded by the mRMR algorithm in SARIMAX. Thus, even though different features are important, both models appear to behave similarly.

The importance of methane production rate, OLR, HRT and fed substrate applies to dataset B as well as in dataset A, while the relative importance of features such as substrate (butyric) acid content and autoencoder-generated features marks a difference between dataset B and the other two tested datasets. This fact can be explained by the increasing process inhibition level, due to increased VFAs content (high importance of features capable of strongly changing the process state). Fig. 12c shows the FI for Bayesian Ridge and RNNs on dataset C.

Both models are heavily impacted by fed sugar beet and corn silage, methane production rate, and moderately impacted by the corn silage

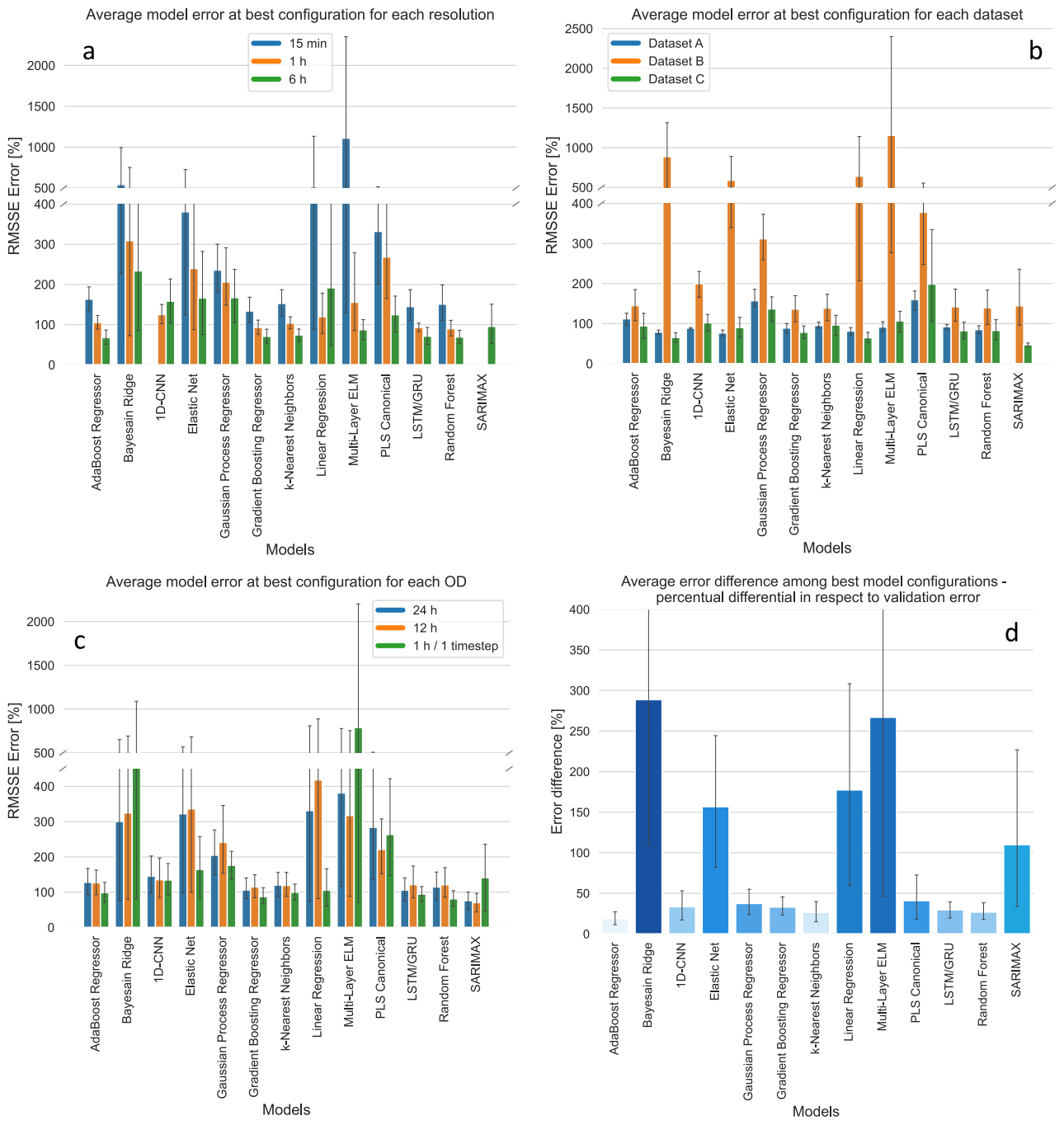


Fig. 10. Impact of data resolution (a), dataset (b), or OD (c) on model performances as well as error difference between validation and test dataset (d).

protein content and by the biogas methane content. While the impact of fed substrate and methane yield of previous time steps was common in all analyzed datasets, the importance of corn silage protein content is high only in dataset C, probably because of more regular changes in the type of fed corn silage. Statistical features such as rolling mean, variance and entropy of biogas yield have importance only in BR when applied to dataset C, demonstrating that RNNs do not need additionally generated statistical features for accurate prediction. As with dataset A, RNNs give high importance to the hour of the day, while BR does not.

In general, both models seem to be affected by similar features (such as fed substrate and methane yield) but they differ in the importance given to further features depending on model characteristics.

Previous studies applying SHAP to AD process (Zhang et al., [23], Schroer and Just [18]) refer, in general, to the amount of substrate added and the previous biogas production as most important features. Zhang et al., [23] also refer to VFAs as an important feature, similarly to the results obtained for Dataset B. In general, results show that previous steps methane production has a high impact on all tested datasets, while HRT, OLR, mixing frequency, hour of the day and dry substrate feed are highly important in two out of the three analyzed datasets. VFAs (in this case, butyric acid) and output digestate are highly important in one out of three datasets. Thus, gas production, methane content in the gas, mixing patterns, TS and VS content of the substrate and amount of dry substrate fed are the necessary measurements for accurate process prediction.

Table 4
Test results of prediction models applied on tested datasets with 24 h OD.[2]

Resolution		15 min			1 h			6 h		
Error metric		RMSSE	MAE	MAPE[1]	RMSSE	MAE	MAPE[1]	RMSSE	MAE	MAPE[1]
Dataset A	Adaboost Regressor	108	0.95	10.7	93	0.95	10.5			
	Bayesian Ridge	88	0.79	8.8	79	0.73	8.1			
	Elastic Net	**80**	0.75	8.5	**75**	0.78	8.6			
	Gaussian Process Regressor	228	1.04	11.7	146	1.00	11.0			
	Gradient Boosting Regressor	99	0.88	9.9	85	0.77	8.5			
	LSTM/GRU	91	**0.74**	**8.4**	92	0.86	9.5			
	Linear Regression	89	0.78	8.7	97	0.88	9.7			
	Multilayer ELM	91	0.86	9.6	87	0.89	9.8			
	PLS Canonical	202	1.64	18.4	163	1.38	15.2			
	Random Forest	91	0.79	8.8	82	**0.64**	**7.0**			
	k-Nearest Neighbors	94	0.82	9.2	83	0.88	9.7			
	1D-CNN				88	0.77	8.5			
Dataset B	Adaboost Regressor	235	2.96	19.2	153	2.98	15.4	109	2.14	10.9
	Bayesian Ridge	1504	4.12	26.8	164	4.37	22.6	558	4.62	23.6
	Elastic Net	1469	4.89	31.8	533	5.77	29.8	256	3.64	18.6
	Gaussian Process Regressor	396	5.92	38.5	511	6.75	34.9	244	5.56	28.5
	Gradient Boosting Regressor	**224**	**2.62**	**17.0**	153	**2.30**	**11.9**	106	2.03	10.4
	LSTM/GRU	282	3.40	22.1	**125**	2.32	12.0	111	2.29	11.7
	Linear Regression	1964	5.59	36.3	168	3.00	15.5	817	8.90	45.6
	Multilayer ELM	1808	3.81	24.8	156	2.79	14.5	109	2.40	12.2
	PLS Canonical	459	6.54	42.5	352	7.16	37.0	181	4.19	21.4
	Random Forest	293	2.93	19.0	148	2.65	13.7	101	**1.90**	**9.7**
	k-Nearest Neighbors	247	2.93	19.0	141	2.49	12.9	**95**	2.01	10.3
	1D-CNN				174	4.59	23.8	244	5.50	28.1
	SARIMAX							97	2.18	11.2
Dataset C	Adaboost Regressor	176	3.65	34.3	91	2.68	27.5	47	2.12	20.5
	Bayesian Ridge	94	1.86	17.5	**65**	1.74	17.9	45	1.85	17.9
	Elastic Net	142	2.74	25.7	88	2.47	25.2	47	1.84	17.8
	Gaussian Process Regressor	202	3.97	37.2	133	4.11	42.0	68	2.91	28.1
	Gradient Boosting Regressor	110	2.32	21.7	87	2.13	21.8	52	1.85	17.9
	LSTM/GRU	141	2.94	27.6	78	1.67	17.1	49	2.00	19.4
	Linear Regression	**90**	**1.78**	**16.7**	78	**1.48**	15.1	**44**	**1.10**	**10.6**
	Multilayer ELM	143	2.55	23.9	94	2.33	23.9	51	2.05	19.9
	PLS Canonical	188	4.78	44.9	151	4.40	45.1	74	3.90	37.8
	Random Forest	143	3.23	30.3	76	1.92	19.7	49	1.76	17.0
	k-Nearest Neighbors	158	3.13	29.4	98	2.52	25.8	52	2.19	21.2
	1D-CNN				86	2.20	22.5	78	3.35	32.4
	SARIMAX							**44**	1.51	14.7

[1] MAPE is approximated as the ratio between MAE and the average target value for the test set.
[2] Bold values indicate the lowest value for each error metric in each data set.

3.6. Discussion

Results clearly show that model choice must be taken depending on individual operating conditions, resolution and expected OD. In general, PLS and GPR are not recommended for model prediction in any prediction scenarios, since they failed to perform sufficiently well in any of the tested scenarios. When modelling steady-state datasets, simple, linear models such as EN, BR and LR are sufficient for process modelling. While more advanced models, such as RNNs or tree-based methods, still achieve reasonable performance, they do not provide significant improvements over linear models. This suggests that, for steady-state conditions, increasing model complexity does not necessarily enhance predictive accuracy. Linear models also excel when several different substrates are fed into the reactor, but more complex models such as RNNs and RF perform quite similarly, especially at 1 h resolution. However, with increasing data complexity, linear models are not recommended. Signs of overfitting appear as visible from the difference in validation and test error from Fig. 6 and Fig. 10d, and complex models such as RNNs, k-NN and RF outperform simpler models. Since future reactor performances are unknown, more complex models are recommended when used for control purposes, especially due to the high average difference between validation and test datasets of linear models. At higher resolutions and lower OD, RNNs are recommended over RF and k-NN, with GBR and ABR also performing well. While RNNs seem to be the best model to be applied for model prediction in general, the described alternatives must be applied when training times are crucial for model application. Regarding the measurements required for model prediction, steady-state datasets mostly require only feeding amount information, past data of biogas production, methane and hydrogen content as well as reactor and gas temperature. During dynamic operation of the AD process, laboratory measurements of the substrate and digestate are required, especially the VFAs content of the digestate.

The utilization of the applied models for control purposes in industrial scale can be advantageous for feed control in demand-oriented energy production from biogas, as described by Jeong et al. [14]. Gas production data and additional sensor (and eventually laboratory) data might be saved on cloud systems, communicating with the models that could predict future methane production to calculate optimal feed amounts. While the time-consuming data preparation and hyperparameter optimization would be applied before the start of reactor control operations, the re-training of the model on newly obtained process data would take a fraction of the optimization time. One pipeline iteration can be considered as short as 0.05 % of the time required for a complete pipeline optimization (1 iteration instead of 2000 iterations). Moreover, this calculation does not consider that a considerably lower amount of data would be fed to the model at each 1-h timestep. Considering a 96-h optimization time for LSTM NNs simulating the AD process at 1 h resolution, a single re-training of a 5-months dataset would take on average 2.88 min, which would be an acceptable re-training time considering the hypothesized 1 h resolution of the control algorithm. This time is expected to be even lower, since the applied amount of data would be considerably shorter than 5 months.

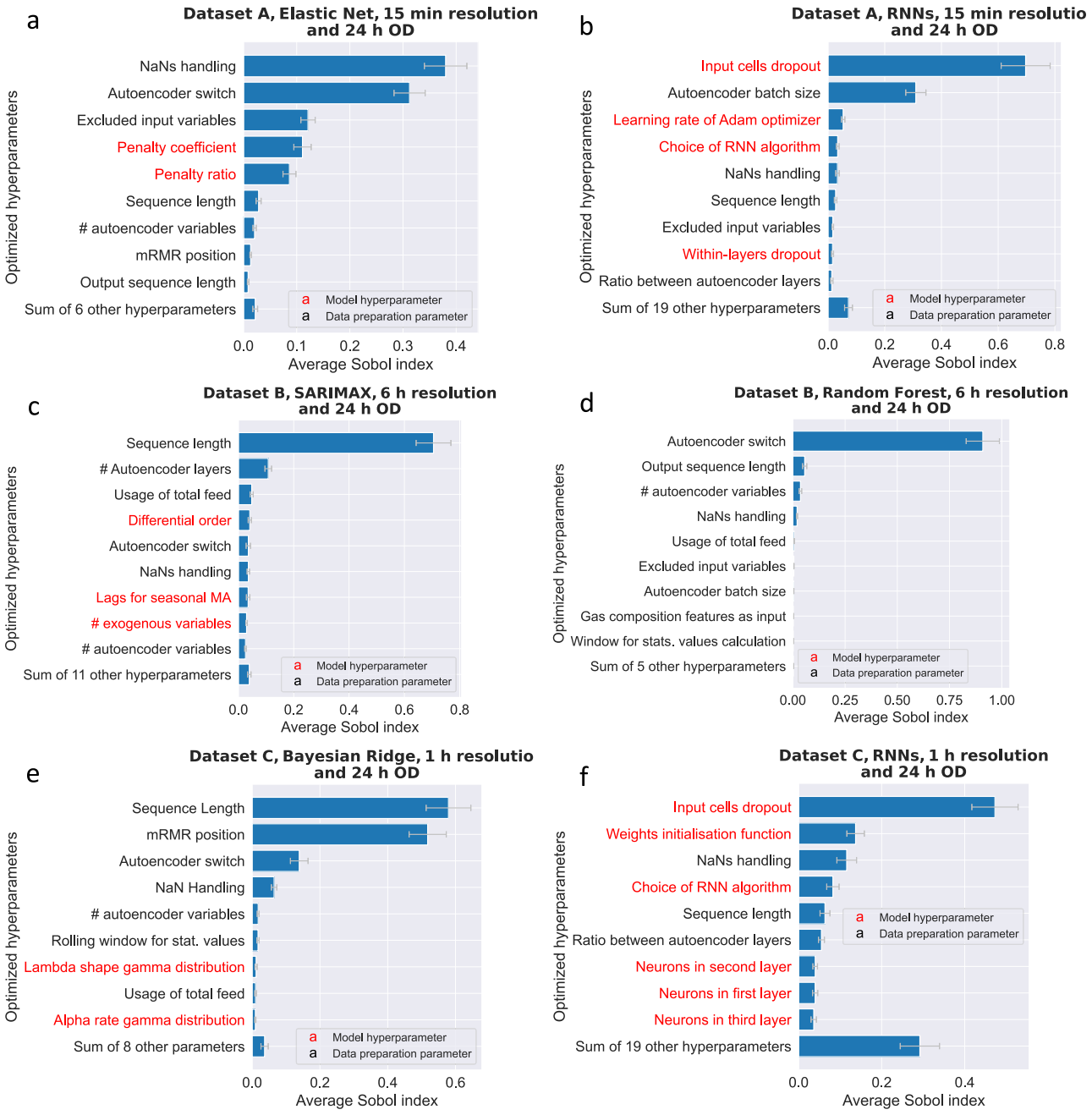


Fig. 11. Sobol' indices of most influential parameters for dataset A (15 min resolution and 24 h OD), dataset B (6 h resolution and 24 h OD) and dataset C (1 h resolution and 24 h OD).

While the optimal results were based on validation accuracy and evaluated on test results for avoiding bias, the training of the models on additional datasets (including re-training) might further increase the understanding of model behaviour. Another aspect that was not analyzed in this study was the impact of measurement frequency on prediction performance. Thus, the effect of daily VFAs analysis can be investigated to determine if datasets with highly dynamic process conditions can benefit from a higher measurement frequency. Additional improvement of model performance might be obtained by filtering the data for noise removal and setting specific error metrics for peak detection. A multi-step prediction of the datasets would also be desired for control purposes.

4. Conclusions

This investigation demonstrates the potential of ML and DL models applied to full-scale AD processes. All models benefitted from an optimization pipeline, including data preparation parameters and hyperparameters, especially linear and ensemble models. Linear models such as BR and LR can predict methane yield during stationary and uninhibited process conditions, resulting in an RMSSE between 61 % and 97 %. The usage of more robust models – such as RF and RNNs – is suggested when the process state is unknown or strongly non-stationary. Most important measurements for such models are the methane yield, fed substrates amounts, as well as calculated features such as OLR and

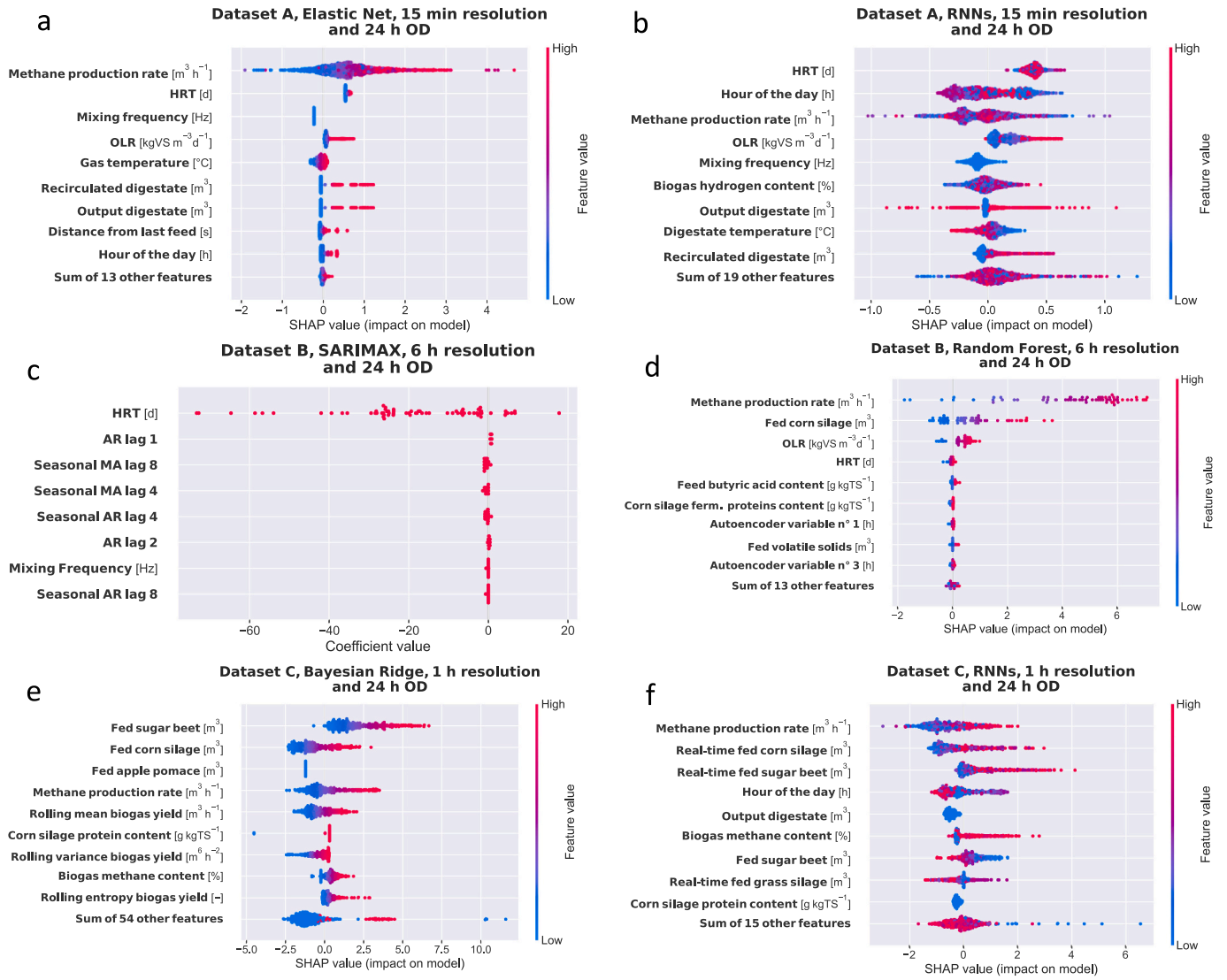


Fig. 12. Feature importance for dataset A (15 min resolution and 24 h OD), B (6 h resolution and 24 h OD) and C (1 h resolution and 24 h OD) on selected models.

HRT. Additional features that can be relevant for prediction models are biogas composition, recirculated and output digestate. However, additional investigation on the suitability of the applied models and input features in case of process disturbance or inhibition are required.

CRediT authorship contribution statement

Alberto Meola: Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Sören Weinrich:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are thankful for funding from the European Regional Development Fund (ERDF) of the research project Integrated control of biogas plants for flexibilization and energetic optimisation (grant 100267056) and of the research project Demand-oriented control of energy from biomass (grant 100143221). The authors are also thankful for funding from the German Federal Ministry of Food and Agriculture of the junior research group on Simulation, monitoring and control of anaerobic digestion plants (grant 2219NR333). Martin Bogdan is acknowledged for supporting the writing of the manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.apenergy.2025.125781.

Data availability

Data will be made available on request.

References

[1] Uddin MM, Wright MM. Anaerobic digestion fundamentals, challenges, and technological advances. Phys Sci Rev 2023;8:2819–37. https://doi.org/10.1515/psr-2021-0068.

[2] Lafratta M, Thorpe RB, Ouki SK, Shana A, Germain E, Willcocks M, et al. Dynamic biogas production from anaerobic digestion of sewage sludge for on-demand electricity generation. Bioresour Technol 2020;310:123415. https://doi.org/10.1016/j.biortech.2020.123415.

[3] Gaida D, Wolf C, Bongards M. Feed control of anaerobic digestion processes for renewable energy production: A review. Renew Sust Energ Rev 2017;68:869–75. https://doi.org/10.1016/j.rser.2016.06.096.

[4] Batstone DJ, Keller J, Angelidaki I, Kalyuzhnyi SV, Pavlostathis SG, Rozzi A, et al. The IWA anaerobic digestion model no 1 (ADM1). Water Sci Technol 2002;45:65–73. https://doi.org/10.2166/wst.2002.0292.

[5] García-Diéguez C, Bernard O, Roca E. Reducing the anaerobic digestion model no. 1 for its application to an industrial wastewater treatment plant treating winery effluent wastewater. Bioresour Technol 2013;132:244–53. https://doi.org/10.1016/j.biortech.2012.12.166.

[6] Mo R, Guo W, Batstone D, Makinia J, Li Y. Modifications to the anaerobic digestion model no. 1 (ADM1) for enhanced understanding and application of the anaerobic treatment processes – A comprehensive review. Water res. 244, 120504. 2023. https://doi.org/10.1016/j.watres.2023.120504.

[7] Weinrich S, Nelles M. Systematic simplification of the anaerobic digestion model no. 1 (ADM1) – model development and stoichiometric analysis. Bioresour. Technol. 2021;333:125124. https://doi.org/10.1016/j.biortech.2021.125124.

[8] Khan M, Chuenchart W, Surendra KC, Kumar Khanal S. Applications of artificial intelligence in anaerobic co-digestion: recent advances and prospects. Bioresour Technol 2023;370:128501. https://doi.org/10.1016/j.biortech.2022.128501.

[9] Ling JYX, Chan YJ, Chen JW, Chong DJS, Tan ALL, Arumugasamy SK, et al. Machine learning methods for the modelling and optimisation of biogas production from anaerobic digestion: a review. Environ Sci Pollut Res 2024. https://doi.org/10.1007/s11356-024-32435-6.

[10] De Clercq D, Wen Z, Fei F, Caicedo L, Yuan K, Shang R. Interpretable machine learning for predicting biomethane production in industrial-scale anaerobic co-digestion. Sci Total Environ 2020;712:134574. https://doi.org/10.1016/j.scitotenv.2019.134574.

[11] Hansen Bolette D, Johansen Rasmus, Tamouk Jamshid, Tidmarsh Christian A, Moeslund Thomas B, Jensen David G. Prediction of the methane production in biogas plants using a combined Gompertz and machine learning model. Presented at the computational science and its applications – ICCSA 2020. 2020.

[12] Wang L, Long F, Liao W, Liu H. Prediction of anaerobic digestion performance and identification of critical operational parameters using machine learning algorithms. Bioresour Technol 2020;298:122495. https://doi.org/10.1016/j.biortech.2019.122495.

[13] Dittmer C, Krümpel J, Lemmer A. Modeling and simulation of biogas production in full scale with time series analysis. Microorganisms 2021;9:324. https://doi.org/10.3390/microorganisms9020324.

[14] Jeong K, Abbas A, Shin J, Son M, Kim YM, Cho KH. Prediction of biogas production in anaerobic co-digestion of organic wastes using deep learning models. Water Res 2021;205:117697. https://doi.org/10.1016/j.watres.2021.117697.

[15] Long F, Wang L, Cai W, Lesnik K, Liu H. Predicting the performance of anaerobic digestion using machine learning algorithms and genomic data. Water Res 2021;199:117182. https://doi.org/10.1016/j.watres.2021.117182.

[16] Wang Y, Huntington T, Scown CD. Tree-based automated machine learning to predict biogas production for anaerobic co-digestion of organic waste. ACS Sustain Chem Eng 2021;9:12990–3000. https://doi.org/10.1021/acssuschemeng.1c04612.

[17] Li C, He P, Peng W, Lü F, Du R, Zhang H. Exploring available input variables for machine learning models to predict biogas production in industrial-scale biogas plants treating food waste. J Clean Prod 2022;380:135074. https://doi.org/10.1016/j.jclepro.2022.135074.

[18] Schroer HW, Just CL. Feature engineering and supervised machine learning to forecast biogas production during municipal anaerobic co-digestion. ACS EST Eng 2023. https://doi.org/10.1021/acsestengg.3c00435.

[19] Sappl J, Harders M, Rauch W. Machine learning for quantile regression of biogas production rates in anaerobic digesters. Sci Total Environ 2023;872:161923. https://doi.org/10.1016/j.scitotenv.2023.161923.

[20] Wang Z, Peng X, Xia A, Shah AA, Yan H, Huang Y, et al. Comparison of machine learning methods for predicting the methane production from anaerobic digestion of lignocellulosic biomass. Energy 2023;263:125883. https://doi.org/10.1016/j.energy.2022.125883.

[21] Yildirim O, Ozkaya B. Prediction of biogas production of industrial scale anaerobic digestion plant by machine learning algorithms. Chemosphere 2023;335:138976. https://doi.org/10.1016/j.chemosphere.2023.138976.

[22] Sun Y, Dai H, Moayedi H, Nguyen Le B, Muhammad Adnan R. Predicting steady-state biogas production from waste using advanced machine learning-metaheuristic approaches. Fuel 2024;355:129493. https://doi.org/10.1016/j.fuel.2023.129493.

[23] Zhang Y, Zhao Y, Feng Y, Yu Y, Li Y, Li J, et al. Novel intelligent system based on automated machine learning for multiobjective prediction and early warning guidance of biogas performance in industrial-scale garage dry fermentation. ACS EST Eng 2024;4:139–52. https://doi.org/10.1021/acsestengg.3c00079.

[24] Meola A, Winkler M, Weinrich S. Metaheuristic optimization of data preparation and machine learning hyperparameters for prediction of dynamic methane production. Bioresour Technol 2023;372:128604. https://doi.org/10.1016/j.biortech.2023.128604.

[25] Jimenez J, Latrille E, Harmand J, Robles A, Ferrer J, Gaida D, et al. Instrumentation and control of anaerobic digestion processes: a review and some research challenges. Rev Environ Sci Biotechnol 2015;14:615–48. https://doi.org/10.1007/s11157-015-9382-6.

[26] Casolaro A, Capone V, Iannuzzo G, Camastra F. Deep learning for time series forecasting: advances and open problems. Information 2023;14:598. https://doi.org/10.3390/info14110598.

[27] Braun S. LSTM Benchmarks for Deep Learning Frameworks 2018. https://doi.org/10.48550/arXiv.1806.01818.

[28] Xiong R, Yang Y, He D, Zheng K, Zheng S, Xing C, et al. On layer normalization in the transformer architecture, in: Proceedings of the 37th international conference on machine learning. In: Presented at the international conference on machine learning, PMLR; 2020. p. 10524–33.

[29] Zou H, Hastie T. Regularization and variable selection via the elastic net. J. R. Stat Soc Ser B Stat Methodol 2005;67:301–20. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[30] Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning. 2nd ed. New York, USA: Springer; 2009.

[31] Tipping ME. Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 2001;1:211–44.

[32] Kruschke J. Introduction to Bayesian data analysis [WWW document]. Introd Bayesian Data Anal - Media Collect Online 2012. https://media.dlib.indiana.edu/media_objects/vd66w373r [accessed 2.6.24].

[33] Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 2000;42:80–6. https://doi.org/10.2307/1271436.

[34] Härdle W, Simar L. Applied multivariate statistical analysis. 2nd ed. Berlin: Springer; 2007.

[35] Breiman L. Random forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.

[36] Freund Y, Schapire R. Experiments with a new boosting algorithm. Presented at the International Conference on Machine Learning. 1996.

[37] Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Stat 2001;29:1189–232. https://doi.org/10.1214/aos/1013203451.

[38] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. The MIT Press 2005. https://doi.org/10.7551/mitpress/3206.001.0001.

[39] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 1992;46:175–85. https://doi.org/10.2307/2685209.

[40] Ahmed SF, Alam MdSB, Hassan M, Rozbu MR, Ishtiak T, Rafa N, et al. Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artif Intell Rev 2023;56:13521–617. https://doi.org/10.1007/s10462-023-10466-8.

[41] Sarker IH. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci 2021;2:420. https://doi.org/10.1007/s42979-021-00815-1.

[42] Mussumeci E, Codeço Coelho F. Large-scale multivariate forecasting models for dengue - LSTM versus random forest regression. Spat Spatio-Temporal Epidemiol 2020;35:100372. https://doi.org/10.1016/j.sste.2020.100372.

[43] Buber E, Diri B. Performance analysis and CPU vs GPU comparison for deep learning, in: 2018 6th international conference on Control Engineering & Information Technology (CEIT). In: Presented at the 2018 6th international conference on Control Engineering & Information Technology (CEIT); 2018. p. 1–6. https://doi.org/10.1109/CEIT.2018.8751930.

[44] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 IEEE international joint conference on neural networks (IEEE cat. No.04CH37541). Presented at the 2004 IEEE international joint conference on neural networks (IEEE cat. No.04CH37541), pp. 985–990 vol.2. 2004. https://doi.org/10.1109/IJCNN.2004.1380068.

[45] Penrose R. On best approximate solutions of linear matrix equations. Math Proc Camb Philos Soc 1956;52:17–9. https://doi.org/10.1017/S0305004100030929.

[46] Park J-M, Kim J-H. Online recurrent extreme learning machine and its application to time-series prediction. In: Presented at the 2017 international joint conference on neural networks (IJCNN). Anchorage, USA: IEEE; 2017. https://doi.org/10.1109/IJCNN.2017.7966094.

[47] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[48] Van Houdt G, Mosquera C, Nápoles G. A review on the long short-term memory model. Artif Intell Rev 2020;53:5929–55. https://doi.org/10.1007/s10462-020-09838-1.

[49] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Presented at the EMNLP 2014, Association for Computational Linguistics. Qatar: Doha; 2014. p. 1724–34. https://doi.org/10.3115/v1/D14-1179.

[50] Ahmed WS, Karim A, Amir A. The impact of filter size and number of filters on classification accuracy in CNN, in: 2020 international conference on computer science and software engineering (CSASE). In: Presented at the 2020 international

conference on computer science and software engineering (CSASE); 2020. p. 88–93. https://doi.org/10.1109/CSASE48920.2020.9142089.
[51] Durbin J, Koopman SJ. Time series analysis by state space methods. OUP Oxford; 2012.
[52] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27:1226–38. https://doi.org/10.1109/TPAMI.2005.159.
[53] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in neural information processing systems. Curran Associates, Inc; 2017.
[54] Chalkiadakis G, Elkind E, Wooldridge M. Computational aspects of cooperative game theory. Springer Nature; 2022.
[55] Alexandropoulos S-AN, Kotsiantis SB, Vrahatis MN. Data preprocessing in predictive data mining. Knowl Eng Rev 2019;34:e1. https://doi.org/10.1017/S026988891800036X.
[56] Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinforma Comput Biol 2005;3:185–205. https://doi.org/10.1142/s0219720005001004.
[57] Sobol' IM. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simul. The Second IMACS Seminar on Monte Carlo Methods 2001;55:271–80. https://doi.org/10.1016/S0378-4754(00)00270-6.
[58] Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput Phys Commun 2010;181:259–70. https://doi.org/10.1016/j.cpc.2009.09.018.
[59] Iwanaga T, Usher W, Herman J. Toward SALib 2.0: advancing the accessibility and interpretability of global sensitivity analyses. Socio-Environ Syst Model 2022;4:18155. https://doi.org/10.18174/sesmo.18155.
[60] Taye MM. Understanding of machine learning with deep learning: architectures, workflow. Applications and Future Directions Computers 2023;12:91. https://doi.org/10.3390/computers12050091.
[61] Bentéjac C, Csörgő A, Martínez-Muñoz G. A comparative analysis of gradient boosting algorithms. Artif Intell Rev 2021;54:1937–67. https://doi.org/10.1007/s10462-020-09896-5.
[62] Schmitt M. Deep Learning vs. Gradient Boosting: Benchmarking state-of-the-art machine learning algorithms for credit scoring. Papers, Papers; 2022.
[63] Kaparaju P, Buendia I, Ellegaard L, Angelidakia I. Effects of mixing on methane production during thermophilic anaerobic digestion of manure: lab-scale and pilot-scale studies. Bioresour Technol 2008;99:4919–28. https://doi.org/10.1016/j.biortech.2007.09.015.
[64] Wu D, Li L, Peng Y, Yang P, Peng X, Sun Y, et al. State indicators of anaerobic digestion: A critical review on process monitoring and diagnosis. Renew Sust Energ Rev 2021;148:111260. https://doi.org/10.1016/j.rser.2021.111260.
[65] Hoffmann R, Garcia M, Veskivar M, Karim K, Al-Dahhan M, Angenent L. Effect of shear on performance and microbial ecology of continuously stirred anaerobic digesters treating animal manure. Biotechnol Bioeng 2008;100. https://doi.org/10.1002/bit.21730.