Chemical Engineering Science 306 (2025) 121210

Contents lists available at ScienceDirect

Chemical Engineering Science

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

Automated deep learning model development based on weight sensitivity and model selection statistics

Damla Yalcin [a], Ozgun Deliismail [b], Basak Tuncer [b], Onur Can Boy [b], Ibrahim Bayar [c], Gizem Kayar [c], Muratcan Ozpinar [c], Hasan Sildir [a,*]

[a] Department of Chemical Engineering, Izmir Institute of Technology, Izmir 35430, Turkey
[b] SOCAR Turkey R&D and Innovation Co., Izmir 35800, Turkey
[c] SOCAR Turkey Refinery & Petrochemical Business Unit, Izmir 35800, Turkey

ARTICLE INFO

Keywords:
Mixed-integer programming
Deep learning
Input selection
Weight sensitivity
Bayesian Information Criterion
Akaike Information Criterion

ABSTRACT

Current sustainable production and consumption processes call for technological integration with the realm of computational modeling especially in the form of sophisticated data-driven architectures. Advanced mathematical formulations are essential for deep learning approach to account for revealing patterns under nonlinear and complex interactions to enable better prediction capabilities for subsequent optimization and control tasks. Bayesian Information Criterion and Akaike Information Criterion are introduced as additional constraints to a mixed-integer training problem which employs a parameter sensitivity related objective function, unlike traditional methods which minimize the training error under fixed architecture. The resulting comprehensive optimization formulation is flexible as a simultaneous approach is introduced through algorithmic differentiation to benefit from advanced solvers to handle computational challenges and theoretical issues. Proposed formulation delivers 40% reduction, in architecture with high accuracy. The performance of the approach is compared to fully connected traditional methods on two different case studies from large scale chemical plants.

1. Introduction

Sustainable production is a multifaceted approach to balance the demand, environmental impact, uncertainties associated with the process dynamics and driving forces under economic considerations. While recent advancements in Industrial Internet of Things (IIoT) introduces significant potential for real time analysis and decision making, a comprehensive and plug-and-use formulation is lacked as case dependent tailorization is required in theoretical perspective as nonlinearity, complexity and causality of the data set have a wide spectrum under complex processes (Zhang et al., 2016; Bakshi, 2019; López-Guajardo et al., 2022). Process Intensification (PI) in chemical processes aligns with IIoT objectives, aiming to minimize raw material use, waste products, and energy consumption (Dantas et al., 2021). It optimizes the value chain by transforming products back into raw materials. The convergence of PI and Industry 4.0 presents opportunities for sustainable practices and new technologies. PI uses machine learning for effective information extraction, data pattern recognition, and predictions (López-Guajardo et al., 2022). Thus, automated synthesis of the machine learning (ML) architecture plays a crucial role in this context by introducing more reliable predictions to favor shifts in economic, sustainable, and production frameworks, thus enhancing efficiency in many aspects (López-Guajardo et al., 2022; Dantas et al., 2021; Vinuesa et al., 2020; Jamwal et al., 2022).

Data-driven models have a major practical superiority as those reveal the interactions between process variables with no fundamental knowledge on process driving forces at micro scale, once trained through sophisticated mathematical formulations to account for overfitting problem and input selection tasks in addition to many other numerical issues. In turn, those are useful for sustainable production and consumption processes to handle complex decision making problems involving environmental impacts, energy efficiency, safety, and economic viability. Deep Learning, as a branch of machine learning, is increasingly essential in chemical engineering for managing large-scale, complex datasets and providing insights critical for such sustainable development tasks (López-Guajardo et al., 2022; Dantas et al. , 2021; Vinuesa et al., 2020; Jamwal et al., 2022; Cioffi et al., 2020; Wuest et al., 2016; Wang et al., 2018). In recent years, engineers have shown a strong

* Corresponding author.
E-mail address: hasansildir@iyte.edu.tr (H. Sildir).

https://doi.org/10.1016/j.ces.2025.121210
Received 24 May 2024; Received in revised form 16 July 2024; Accepted 8 January 2025
Available online 13 January 2025

interest for the utilization of those algorithms for modeling of complex process networks (Dobbelaere et al., 2021; Venkatasubramanian, 2019; Schweidtmann, 2021; Wu et al., 2023; Gao et al., 2022; Zhao, 2022; Shang et al., 2014) as those are capable of representing complex behavior under nonlinear and multivariable domain. However, the current challenges with problem formulation and struggle with ill-posed problems with uncertain parameters, complex geometric domains, and stiff partial differential equations in high-dimensional space call for data-driven methods to handle modeling tasks in engineering. Thus, they can overcome the practical limits of mechanistic modeling by representing the complex interactions, although the process is treated as a black box to some extent, and deliver models at a lower cost, in turn being useful for optimization. However, those models are more useful and reliable once the training problem is tightened by additional and case dependent constraints to achieve the best performance based on the statistical quality of the data.

To reveal complex interactions between input and output variables, such as estimating chemical characteristics, creating closure models, evaluating uncertainty, predicting reaction outcomes, and comprehending catalytic processes, ML approaches are implemented previously (Leonard et al., 2021; Trinh et al., 2021). Supervised ML models are frequently used to construct representative relationships and causality among inputs and outputs, for the prediction of the latter with high accuracy. In parallel, deep learning, which employ multiple hidden layers, has been gaining more attention in chemical engineering (Venkatasubramanian, 2019; Schweidtmann, 2021; Zhong et al., 2021). This has resulted in an exponential increase in the number of studies on the subject, since it has a flexible nature of fitting to complex and nonlinear data with high number of interactions among hidden layers. Deep learning and predictive analytics algorithms have the potential to significantly accelerate sustainable chemistry design by deciphering multiple layers of representations from high-throughput experiment and theoretical calculation data without the need for specific feature extractors to be designed and tuned (Leonard et al., 2021; Ning and You, 2019). Deep neural networks (DNNs) are sophisticated, nonlinear, learning-based mathematical formulations which require advanced mathematical tools in the training due to increased complexity, which can be handled by many commercial and open-source optimization solvers. In particular, a feedforward DNN carries information from input layer to the output layer without using feedback connections and propagating the knowledge in a single direction only. Synthesis of those is theoretically challenging once automated approach, which delivers more efficient architectures based on statistical measures and expressions rather than human impact, is required. In order to make an automated approach method to balance the complexity and accuracy, and to decrease the impact of overfitting, Bayesian Information Criterion (BIC) (Schwarz, 1978) and the Akaike Information Criterion (AIC) (Akaike, 1998), are promising and common model selection statistics. AIC is a statistical formulation and manages the trade-off between complexity and fitting performance. Thus, a reduced model with fewer parameters, favored by BIC and AIC during the model development and training, would result a more reliable prediction profile, which is less eposed to the impact of the possible overfitting issues. For such purpose, both BIC and AIC aim to balance the model complexity and fitting performance, in turn delivers similar conclusions intuitively, although the former penalizes the parameters more.

Sensitivity analysis (SA) is an important part of model prediction uncertainty and parameter identifiability tasks (Loucks and Van Beek, 2017; Saltelli et al., 2019; Helton et al., 2006; Razavi, 2021; Ricotti and Zio, 1999; Wallace, 2000). Based on the knowledge obtained from SA, some inference based on the prediction robustness and model architecture efficiency can be performed (Helton, 1993; Ginocchi et al., 2021; Saltelli and Sobol, 1995; Zheng and Keller, 2006). It has also found widespread implementations in ML and data science, with current heuristics established to aid in feature and structure selection as well as to solve explainability and interpretability challenges. Further formalization and tailorization of SA approaches and tools may help in addressing the explainability, interpretability, and falsifiability concerns (Saltelli et al., 2021; Razavi, 2021). Also, SA is essential in systems analysis and modeling because it aims to investigate causalities, identify unimportant elements, evaluate data value, and measure the sensitivity of an expected outcome to various choice alternatives, restrictions, assumptions, and uncertainties. It is currently regarded a necessity for modeling to use the sparsity of factors principle (Saltelli et al., 2020; Better regulation: guidelines and toolbox, 2023). SA can be conducted using techniques such as one-at-a-time analysis, partial derivatives, or local sensitivity analysis (Ginocchi et al., 2021). Local Sensitivity Analysis (LSA) is a method to calculate the sensitivity around a nominal point. Despite being criticized for offering a confined overview of the overall space, it is useful particularly when studying parameter relevance in mathematical modeling (Saltelli et al., 2019; Saltelli and Annoni, 2010; Qin et al., 2023). However, LSA accuracy might be limited once uncertainty range and corresponding nonlinear impact are considerable, calling for including higher order terms in the formulations (Li et al., 2023). With a strong relation to uncertainty evaluation (Loucks and Van Beek, 2017; Saltelli, 2019; Helton et al., 2006; Razavi, 2021; Wallace, 2000; Helton, 1993) LSA is used to label parameters with low sensitivity to manage prediction uncertainty. Thus, a model architecture with a more robust prediction profile can be obtained (Loucks and Van Beek, 2017; Razavi, 2021; Bergamini et al., 2019). For DNN tasks, SA would be useful for the input variable selection and model structure formulation with aforementioned theoretical advancements (Razavi, 2021; Saltelli et al., 2021).

In the literature there are significant number of studies on the network structure development; despite very few focus on statistical approaches such as BIC and AIC; or sensitivity analysis based on variables in the training stage. Sildir et al. (Sildir et al., 2020) to solve land cover classification issues using two public hyperspectral datasets and showed the contribution of the reduced network. The authors suggest that the optimal number of hidden layer nodes depends on various factors, as too few can cause significant training errors and overfitting, while too many can lead to compared classification accuracies. The study conducted by Bortolan et al. (Bortolan and Fusaro, 1996) examines at the possibility of building an ANN for ECG diagnostic classification by starting with a huge feature space and then pruning it down. Two different pruning approaches have been investigated in this work; the criteria are based on the error function's sensitivity to the removal of a node (node sensitivity) or a weight (weight sensitivity). Ganesh et al. (Ganesh et al., 2022) suggest a novel DNN pruning algorithm called the slimming neural networks using adaptive connectivity scores that utilizes the use of the adaptive conditional mutual information to measure filter connectivity, a simple set of operating constraints to automatically define the upper pruning percentage limits of layers in a DNN, and a sensitivity criterion to help protect a subset of crucial filters from pruning. With a single train-prune-retrain cycle, SNACS gives a quicker total run-time, increases estimate accuracy, and offers state-of-the-art levels of compression. Agarwal et al. (Agarwal et al., 2006) used the back propagation artificial neural network approach to perform a research on monsoon runoff and sediment output in an Indian catchment. The results were compared with observed data and models for linear transfer functions with one or more inputs. Cross-validation was used to generalize the model parsimony, which was done by network pruning using error sensitivity as a weighting factor. The study found that ANN-based runoff simulation outperformed single-input models in verification and calibration, and the sediment-yield models showed superior results in cross-validation and verification. Ennett et al.'s (Ennett and Frize, 2000) study used weight-elimination on the coronary artery bypass grafting (CABG) database and purposefully increases training sets' death rates to address the problem of skewed a priori probabilities in medical decision aids. The results of the research show that increasing the mortality rate enhances sensitivity at costs of additional indicators of performance, whereas weight-elimination cost

function increases sensitivity without significantly impacting other parameters. Lal et al. (Lal et al., 2004) utilized and contrasted two approaches to predict splice locations in their study. The first approach was utilizing a training dataset that included both genuine and false splice sites to train neural networks. After that, test samples were run through the trained network in order to calculate sensitivity. In the second approach, the prediction of splice locations is done using a pruning maximum likelihood model. This method is aimed at identifying splice sites in sequences regardless of the gene. Lastly, Ari et al. (Ari and Saha, 2009) have developed an optimized Artificial Neural Network structure for classifying heart sound signals automatically. The network was built with a compact output layer and successively optimized, with redundant synaptic weights determined based on local relative sensitivity index, hidden nodes removed, and unneeded input nodes pruned. This method utilizes information that is often accessible during the back propagation procedure, requiring just a little amount of computation. The reduced testing time caused by the smaller ANN increases user comfort in real-time devices. Because of its optimization, the ANN works well on low-cost hardware platforms.

However, most of the literature studies about the network structure reduction were obtained through sequential and heuristic methods with not explicit model selection statistics or sensitivity considerations through simultaneous tasks. In this respect, the main objective of our study is to build a parsimonious model that a balanced between model fit and complexity, achieving this balance by using fewer but more significant parameters in terms of their sensitivities. Such a simultaneous model development task is a comprehensive mixed integer nonlinear programming problem (MINLP) to select the significant inputs and parameters with high sensitivity simultaneously by integrating common model selection statistics, BIC and AIC. Thus, in addition to reduced overfitting to obtain a similar training and test performance, the aim development of a DNN architecture requiring major input variables which are most influential in terms of prediction. Two cases from actual plants, one of which is publicly available, are used to demonstrate the impact and show the contribution.

2. Methodology

2.1. Fully connected deep neural network (f-DNN)

DNNs transform knowledge in input vector, u, to prediction, y, through successive activation functions after some linear operations, operated at different hidden layers. Once the information flow is propagated in a single direction, the architecture is mostly referred feed-forward, as feedback loops are not included in the formulation. A typical formulation to calculate the prediction of a particular layer is given by:

$$y^l = f^l\left(w^l \bullet u^{l-1} + b^l\right) \tag{1}$$

where w^l and b^l the weight matrix and the bias vector of layer l, respectively; u^{l-1} is the input vector to be processed by layer l; y^l is the prediction to be delivered to succeeding layers and is the ultimate prediction once it is the output layer; f^l is the activation function at layer l to introduce nonlinearity, if needed, to the network and has a vast number of options, including sigmoid, hyperbolic tangent, rectified linear unit. In general f-DNNs are trained through some kind of unconstrained nonlinear optimization problems and given by:

$$\mathrm{Min}_{w^1,\cdots,w^L,b^1,\cdots b^L} \quad \frac{1}{D}\sum_{d=1}^{D}\|y_d - y_d^M\|$$

s.t.

$$y_d = f^l\left(w^l \bullet u_d^{l-1} + b^l\right) \quad l = 1,\cdots,L \tag{2}$$

where N is the number of samples; L is the number of layers in the network; u_d is the d^{th} input sample; y_d is the d^{th} sample prediction from f-DNN; y_d^M is the d^{th} sample measurement.

2.2. Pruned deep neural network (p-DNN)

Local sensitivity analysis a useful and common formulation to obtain partial derivatives of variables with respect to model parameters, for the utilization in training in addition to many other theoretical tasks including identifiability issues and uncertainty quantification. Despite locality, they have found significant applications in many fields through various analytical and sampling-based methodologies (Perry et al., 2006). In addition, those methods should be tailored well for the equations under considerations to handle arising complexity and intractable model solvability. In contrast to ODEs, where forward sensitivity expressions have derived, the calculations of sensitivities are a challenging task due to recursive and accumulative nonlinear behavior during the knowledge propagation in the DNN architecture under algebraic formulations. However, once the superstructure of the DNN is provided, algorithmic and automatic differentiation tools are useful to exploit the general formulation once the hyperparameters are specified to calculate:

$$s_d^{y_r,w_{ijl}} = \frac{\partial y_d^r}{\partial w_{ij}^l} \tag{3}$$

where $s_d^{y_r,w_{ijl}}$ is the change of the r^{th} prediction variable in the d^{th} sample to a small change in DNN weight, w_{ij}^l, in l^{th} layer with row and column indices i and j, respectively. Note that the formulation in Eq. (3) is applicable for all weights and biases in a DNN once the activation functions have a continuous nature. Eq. (4) is the sensitivity expression for the weights in the first layer which process the input variables when two hidden layers activated by hyperbolic tangent function exist.

$$s_{n_{1s},i_{1s},d}^1 = \sum_{r=1}^{R}\sum_{n_2=1}^{N_2}\left(\sum_{n_1=1}^{N_1} - w_{n_2,n_{1s}} w_{r,n_2} u_{i_{1s},d}\left[\tanh\left\{b_{n_1 s} + \sum_{i=1}^{I} w_{n_1,i}^1 \bullet u_{i,d}\right\}^2 - 1\right]\right) \quad \begin{matrix} n_1 = 1,\cdots,N_1 \\ i = 1,\cdots,I \\ d = 1,\cdots,D \end{matrix} \tag{4}$$

where n_{1s} and i_{1s} are the indices of row and column of the parameter under consideration; d is the index of data sample; R is the number of outputs; N_1 is the number of neurons in the first hidden layer; N_2 is the number of neurons in the second hidden layer; I is the number of inputs; $s_{n_{1s},i_{1s},d}^1$ is the sensitivity coefficient with proper indices. Note that Eq. (4) is derived for a two-hidden layer network which includes hyperbolic tangent function in all layers, except for the input layer, and focuses on the weights between the input layer and the first hidden layer. Advanced algorithmic and automatic differentiation tools are useful and can easily be modified for different tasks easily, if needed, enabling the utilization of rigorous optimization solvers benefiting from sophisticated mathematical reformulations and approximations for handling complex nature of ultimate nonlinearity.

Modified training problem is given by:

$$\mathrm{Min}_{w^1,\cdots,w^L,b^1,\cdots b^L,w^{1B}} \quad \sum_{d=1}^{D}\sum_{n_{1s}=1}^{N_1}\sum_{i_{1s}=1}^{I}\left(w^{1B}_{n_{1s}i_{1s}}s^1_{n_{1s},i_{1s},d}\right)^2 \tag{5.1}$$

s.t.

$$y_d = f^l\left(w^l \bullet u_d^{l-1} + b^l\right) \quad \begin{matrix} l = 1,\cdots,L \\ d = 1,\cdots,D \end{matrix} \tag{5.2}$$

$$\frac{1}{D}\sum_{d=1}^{D}\frac{\|y_d - y_d^M\|}{y_d^M} \le MSE_D \tag{5.3}$$

$$s^1_{n_{1s},i_{1s},d} = \sum_{r=1}^{R}\sum_{n_2=1}^{N_2}\left(\sum_{n_1=1}^{N_1} - w_{n_2,n_{1s}}w_{r,n_2}u_{i_{1s},d}\left[\tanh\left\{b_{n_1s} + \sum_{i=1}^{I}w^1_{n_1,i}\bullet u_{i,d}\right\}^2 - 1\right]\right) \quad \begin{matrix} n_1 = 1,\cdots,N_1 \\ i = 1,\cdots,I \\ d = 1,\cdots,D \end{matrix} \tag{5.4}$$

$$w_{LB}\bullet w^{1B}_{n_{1s}i_{1s}} \le w^1_{n_{1s},i_{1s}} \le w_{UB}\bullet w^{1B}_{n_{1s}i_{1s}} \quad \begin{matrix} n_1 = 1,\cdots,N_1 \\ i = 1,\cdots,I \end{matrix} \tag{5.5}$$

$$w^{1B}_{n_1,i} = w^{1B}_{n_1+1,i} \quad \begin{matrix} n_1 = 1,\cdots,N_1 - 1 \\ i = 1,\cdots,I \end{matrix} \tag{5.6}$$

$$BIC = \frac{-2}{D}\sum_{d=1}^{D}\|y_d - y_d^M\| + log(D)\sum_{n_{1s}=1}^{N_1}\sum_{i_{1s}=1}^{I}w^{1B}_{n_{1s},i_{1s}} \tag{5.7}$$

$$AIC = -\frac{2}{D}\sum_{d=1}^{D}\|y_d - y_d^M\| + \frac{2}{D}\sum_{n_{1s}=1}^{N_1}\sum_{i_{1s}=1}^{I}w^{1B}_{n_{1s},i_{1s}} \tag{5.8}$$

$$BIC_L \le BIC \le BIC_H \tag{5.9}$$

$$AIC_L \le AIC \le AIC_H \tag{5.10}$$

$$w^{1B} \in \{0,1\} \tag{5.11}$$

where a sensitivity related objective function is introduced, unlike traditional training tasks. Note that, the sensitivity information is calculated and introduced for all data in the training region, although not theoretically a necessity, and fewer samples might be considered in the objective function when computational issues become hindering. Normalized training error is introduced in Eq. (5.3) as a constraint where a certain fitting performance which is mostly determined by the measurement accuracy or prediction capability requirement by the plant engineers. Thus, the formulation ensures the selection of maximum sensitivity coefficients whose existence is introduced through the binary variable matrix, $w^{1B}_{n_{1s}i_{1s}}$, which represents the existence of the connections between the input and the first hidden layer. In turn, the sensitivity coefficients of the selected connections are considered in the objective function only, through Eq. (5.5) which constraints the corresponding weights to zero once the connection is eliminated when the binary variable is also zero. The elimination of the connections between input layer and the first layer is further tailored through Eq. (5.6) which converts problem into an input selection formulation to ensure the elimination of all connections from a particular input to the next layer neurons, reducing the network architecture as well, and performing simultaneous input selection. The tradeoff between fitting performance and the model complexity of *p*-DNN has been included in the training problem through introducing *BIC* and *AIC* as additional constraints. Thus, a more convenient network architecture to avoid overfitting is obtained, balancing fitting and complexity with constrained values of *BIC* and *AIC*. With similar conclusions, *BIC* has more penalty on the parameters. Therefore, it is essential to use AIC and BIC in model development with their proven capability in contributions to model selection, overfitting reduction, and balancing model fit and complexity. The two statistical models (*AIC* and *BIC*) are expressed in Eq. (5.7) and Eq. (5.8). Despite significant computational load increase in the training for such superiorities, a fewer number of parameters exist in the ultimate architecture and serves significant superiority in subsequent model update tasks once a new measurement becomes available. In practice, such a complex problem solution is required rarely, once a new input or output is introduced, and in offline mode, making the overall problem solution beneficial in terms of process development.

3. Results

Formulations in Eq. (2) and Eq. (5) are solved through open source IPOPT and BONMIN solver, respectively, through Python/Pyomo interface for comparison purposes based on two industrial datasets, one of which has been publicly available, although the other is not available due to proprietary reasons. Two case study datasets were initially filtered for outliers using the Isolation Forest method, chosen for its scalability and effectiveness in isolating anomalies in large and complex datasets. Specifically, in the Gas hold-up in bubble column case study, data points below 10 % of the target parameter were removed, following a referenced study. Both case studies, including the Gas hold-up in bubble column and Hydrogen production in a reformer plant, used a constant testing data size of 200 randomly selected data points. After filtering, the dataset size for the first case study was reduced from 4042 to 250–300 data points, and for the second case study, from 731 to 250 data points. The filtered data were normalized using MinMaxScaler, scaling variables between −1 and 1 for numerical purposes while preserving data patterns. For both case studies, the filtered and scaled datasets were divided into training datasets (50 or 100 data points) and testing datasets (200 data points). The training dataset was used to discover the connection between dependent and independent variables, while the testing dataset evaluated the trained model's performance. Mean square error (MSE), mean absolute percentage error (MAPE) and R^2 values are calculated for training and test data based on different number of training samples and number of neurons in hidden layers. Computations are performed on an Intel Core i7 processor with 8GBs of RAM.

3.1. Gas hold-up in bubble column

Bubble columns are widely utilized in the chemical industry as gas–liquid contactors and multiphase reactors for various applications like oxidations, hydrogenations, fermentations, and synthetic fuel production due to their high mass transfer rate, interfacial area, and heat transfer coefficient. The design of industrial columns requires understanding characteristics like gas hold-up, heat transfer coefficient, volumetric mass transfer coefficient, and effective interfacial area, which are influenced by physical and chemical qualities, operational circumstances, and geometric parameters (Rollbusch et al., 2015). However, from an industrialization perspective, they have

Table 1
Case 1 variable descriptions.

Type	Description	Tag
Inputs	column diameter	u_1
	liquid height	u_2
	sparger hole diameter	u_3
	percentage free area	u_4
	density of gas	u_5
	viscosity of gas	u_6
	molecular weight of gas	u_7
	density of liquid	u_8
	viscosity of liquid	u_9
	surface tension of liquid	u_{10}
	liquid velocity	u_{11}
	temperature	u_{12}
	pressure	u_{13}
	superficial gas velocity	u_{14}
Output	The gas hold-up	y_1

disadvantages such as difficulty in scaling up (Ren et al., 2006). To handle this difficulty, there are several studies have used ANN to predict bubble diameter, mass transfer coefficient, and gas hold-up. Studies have used ANN to predict k_La, estimate bubble diameter, measure ultrasound, and predict gas hold-up (Alvarez et al., 2001; Wu et al., 2003). The study by Behkish, A. et al. developed and validated a robust ANN for gas holdup prediction in bubble column reactors (BCRs) and slurry bubble column reactors (SBCRs) under various conditions. The model was trained using over 3880 and 1425 data, including gas–liquid-solid properties, operating variables, reactor geometry, and gas sparger type/size through an extensive study on experimental gas holdup and literature, resulting in 90 % prediction accuracy (Behkish et al., 2005). Additionally, in the referenced study, a machine learning-based data-driven methodology is presented for predicting gas hold-up on an industrial scale, using literature data and independent parameters like physical dimensions, sparger design, and physicochemical (Hazare, 2022).

This case is obtained from a dataset of 4042 samples of a gas–liquid bubble column to estimate the gas hold-up based on 14 input variables as shown in Table 1 (Hazare, 2022). In contrast to the referenced work, the sparger type as a design parameter was not used in this investigation.

The performance of *f*-DNN and *p*-DNN are evaluated and compared at different network architectures and number of training samples. The prediction and measurement values are shown in Table 2.

Table 2
f-DNN and *p*-DNN training and test performance with different number of neurons and training samples.

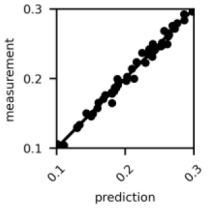
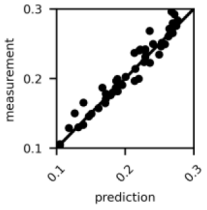
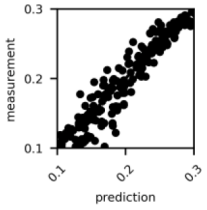
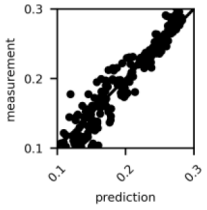
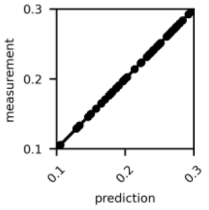
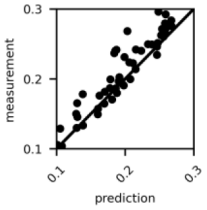
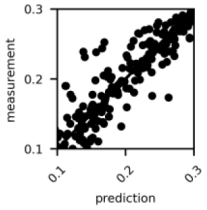
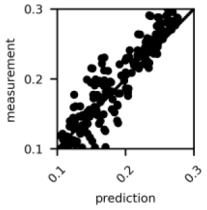
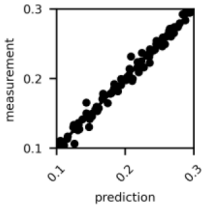
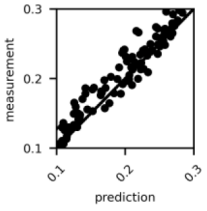
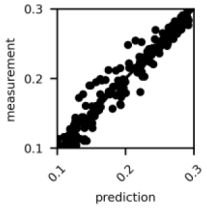
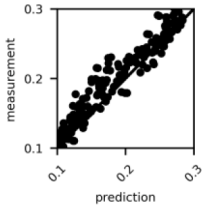
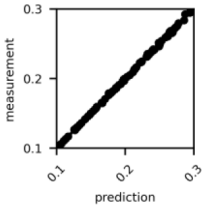
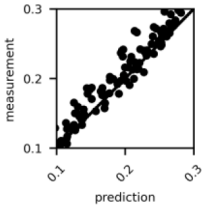
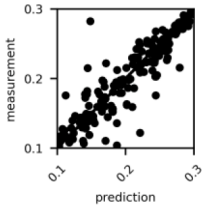
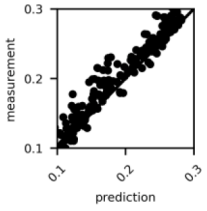
			Training		Test	
	# Neuron					
	1st HL	2nd HL	*f*-DNN	*p*-DNN	*f*-DNN	*p*-DNN
50 training samples	4	3				
	8	6				
100 training samples	4	3				
	8	6				

Table 3
Some common statistics on the performance of *f*-DNN and *p*-DNN.

Method	Training Samples	#N 1st HL	#N 2nd HL	Training			Testing		
				MSE ($\times 10^{-4}$)	MAPE	R^2	MSE ($\times 10^{-4}$)	MAPE	R^2
ANN (Hazare, 2022)		8	6	–	–	–	4.0	7.04	0.87
f-DNN	50	4	3	0.37	2.31	0.99	3.20	7.68	0.91
		8	6	10^{-13}	10^{-7}	0.999	9.53	12.45	0.73
	100	4	3	0.48	2.66	0.99	2.31	6.02	0.93
		8	6	0.015	0.35	0.999	6.46	9.61	0.82
p-DNN	50	4	3	1.56	4.33	0.94	3.23	8.46	0.91
		8	6	5.32	7.80	0.79	5.81	10.05	0.83
	100	4	3	3.71	7.54	0.88	3.96	8.24	0.89
		8	6	3.74	7.21	0.87	3.82	7.61	0.89

Table 4
AIC, BIC, and training time for *p*-DNN development.

Method	Training Samples	#N 1st HL	#N 2nd HL	AIC	BIC	Time (sec)
p-DNN	50	4	3	1.50	136.40	12.74
		8	6	2.73	259.12	67.31
p-DNN	100	4	3	0.73	156.83	445.34
		8	6	1.37	304.18	2877.18

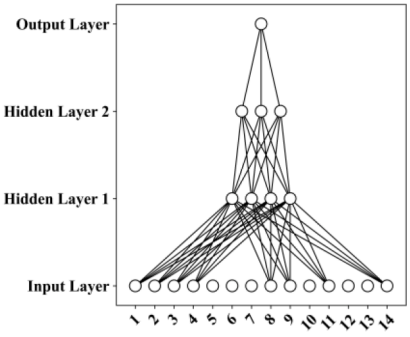
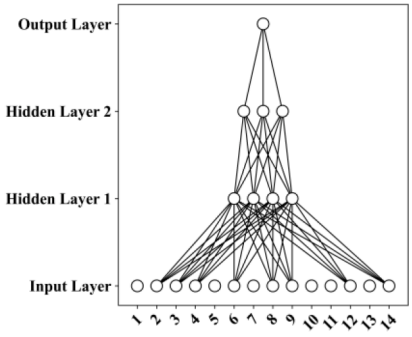
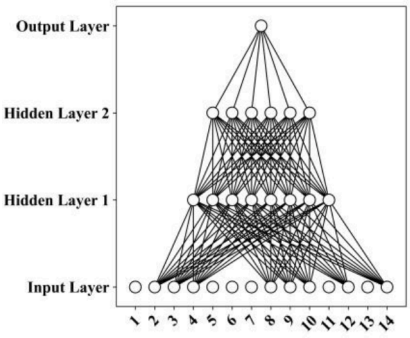
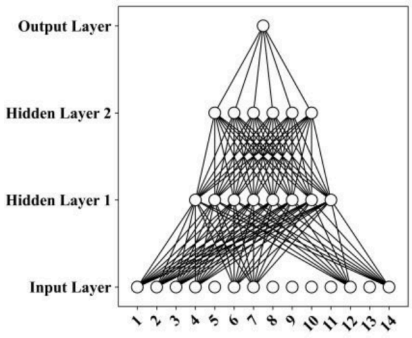
Some commons statistics to quantify the performance comparisons are provided in Table 3.

The MSE, MAPE and R^2 of the *f*-DNN and *p*-DNN are calculated once the predictions are denormalized to actual scale. The test MSEs in the two models increased with model complexity although the impact is smaller for *p*-DNN. In parallel, MSE and MAPE results showed a similar pattern at two different training data number. However, *p*-DNN outperforms *f*-DNN test performance mostly with a more similarity in training and test performance, and compatible with MSEs reported by Hazare et al. (Hazare, 2022). Such a similarity in training and test performance in *p*-DNN is obtained through elimination of some inputs with the formulation in Eq. (5) by constraints in AIC and BIC which are summarized in Table 4:

The connections between layers (input, hidden, and output) in *p*-DNN models are shown in Table 5. Unlike *f*-DNN, processing 14 inputs to represent the superstructure in terms of all available information, a subset of the input variables is connected to succeeding layer. Those connections are formed through automated pruning algorithm presented in Eq. (5) and would deliver different network architectures once the number of hidden layer neurons and data change.

In contrast to *f*-DNN, where 14 inputs are utilized, *p*-DNN eliminates some inputs to achieve constraints by model selection statistics, *BIC* and *AIC*, which primarily take the number of connections and the training error in the formulation. Once the simultaneous optimization formulation with a sensitivity based objective function is employed, the related weights are also significant and contributional in terms of prediction performance. Note that, the test performance of *p*-DNN, with a satisfactory accuracy, is more favorable in terms of theoretical and practical

Table 5
The representation of network structure of *p*-DNN models.

# Neuron		Training Samples	
1st HL	2nd HL	50	100
4	**3**		
8	**6**		

performance with reduced overfitting impact which is observed in the similarity of training and test performance, with a 40 % reduction in input space approximately This means 6 input connections are removed, leaving 8 active connections among the 14 input variables. Consistent results indicate that liquid height (u_2), sparger hole diameter (u_3), percentage free area (u_4), and superficial gas velocity (u_{14}) are essential, while the density of gas (u_5) and pressure (u_{13}) are consistently eliminated. The other input variables possibility of pruning or not pruning was dependent on model design structure based on the number of neurons in the network structure. Such a behavior is more obvious when a higher number of neurons are introduced with few training samples.

Lastly, in the study by Hazare et al. study (Hazare, 2022), column diameter (u_1), liquid height (u_2), sparger hole diameter (u_3), density of gas (u_5), density of liquid (u_8), viscosity of liquid (u_9), surface tension of liquid (u_{10}) and superficial gas velocity (u_{14}) were trained, whereas percentage free area (u_4), viscosity of gas (u_6), molecular weight of gas (u_7), liquid velocity (u_{11}), temperature (u_{12}) and pressure (u_{13}) were not included. This led to a similarity of included and excluded input variables in network structures with *p*-DNN input selection, ranging between 43 % and 71 %.

3.2. Hydrogen production on reformer plant

The hydrogen production plant of SOCAR in Türkiye is established to process natural gas for the production of the required hydrogen throughout the refinery by following conventional steam methane reforming technology. Natural gas is used as the main feed for the production of high-purity hydrogen. In fact, the process mainly consists of a pretreatment section, reformers preceded by a pre-reformer to process different feedstocks at higher temperatures with no fouling from polymerization reactions, a hydrogen purification unit, and some auxiliary combustion and heat recovery systems. The process commences with natural gas purification at which the hydrogen reactor section is used to prevent the reforming catalyst from poisoning due to the substantial impurities in natural gas feed. Therefore, the natural gas feed is initially treated to remove the sulfur-based compounds, chlorides, and some metals to an acceptable level or convert them into non-hazardous components for further catalytic processes. In the pretreatment section, several catalysts, guards, and adsorbents are used for the removal of these compounds from the feed. In general, since the life time of the pre-reformer catalyst is a function of the total sulphur entering the reactor, the pre-reformer life time can be increased by achieving deep desulphurization. Then, treated natural gas is comparatively fed to the reformer train. The pre-reformer is used to adiabatically convert relatively higher hydrocarbons inside treated natural gas into methane, carbon monoxide, carbon dioxide, and hydrogen mixture. The pre-reformer reduces the reformer load and thus its size. Due to the overall exothermic behavior of the complex reactions, the exit temperature of the first reformer increases and then, the effluent is fed to the second reformer after mixing with excess steam to prevent undesired reactions. In the second reformer, steam methane reforming and water–gas shift reactions occur with some side reactions such as Boudard, reduction, and cracking reactions. The overall reaction occurred in the second reformer is endothermic, so heat is externally supplied to the second reformer to sustain the desired conversion. Additionally, the effluent of the second reformer is sent to the MTS reactor at reduced temperature to favor the water–gas shift reaction for the additional hydrogen production. Then, the conventional Pressure Swing Adsorption unit is used to obtain high-purity hydrogen. The plant is highly interacting, and a high number of variables are measured in real-time to ensure the desired hydrogen production rate.

Natural gas was the main feed of the process which produced hydrogen for the refinery. The real plant data are used in the developed model to predict hydrogen production rate from various variables. Among the variables, the feed flow of natural gas was u_{14}. Feed temperature and pressure were given by u_{12} and u_{13}, respectively. An alternative feed was rarely replaced or co-processed with natural gas. The flow of the alternative feed was described in Sm^3/h with u_{17}. The temperature and pressure of the alternative feed are u_{16} and u_{15}, respectively. After hydrogen purification unit, high-purity hydrogen was produced and used as prediction variable and represented by y_1. The flue gas was discharged by the stack after passing the flue gas fans. The flow at the stack was given by u_9 while u_{11} and u_{10} are the temperature and pressure at the stack measured by the Continuous Emission Monitoring System abbreviated as CEMS. Moreover, at the stack, SO_x and NO_x amounts were measured and represented by u_2 and u_3, respectively. Particulate amount at the stack was given by u_4 while carbon monoxide and carbon dioxide were tabulated as volumetric percentages within u_5 and u_6, respectively. Finally, the water (vol. %) and oxygen (vol.%) amounts were described as u_7 and u_8.

Table 6
Case 2 variable descriptions.

Type	Description	Tag
Inputs	capacity reformer	u_1
	SO_x	u_2
	NO_x	u_3
	Particulate	u_4
	CO	u_5
	CO_2	u_6
	H_2O	u_7
	O_2	u_8
	stack flow	u_9
	stack pressure	u_{10}
	stack temperature	u_{11}
	natural gas feed temperature	u_{12}
	natural gas feed pressure	u_{13}
	natural gas flow	u_{14}
	alternative fuel pressure	u_{15}
	alternative fuel temperature	u_{16}
	alternative fuel flow	u_{17}
Output	hydrogen	y_1

In this dataset, there were 17 input as independent variables and an output as dependent variables as shown Table 6. This study aimed to automate the selection of independent variables based on the most influential weight parameters in algorithms using proposed. Table 7 presents the training and testing results for *f*-DNN and *p*-DNN with varying neuron numbers in the hidden layer. The *f*-DNN employs 17 inputs in all calculations in contrast to *p*-DNN which lacks some of the inputs due to elimination based on model selection statistics. The results show no significant change in *f*-DNN and *p*-DNN when increasing the neuron number in the hidden layer with decreasing input variables in the training model.

Table 8 contains several common statistics for performance evaluation. *f*-DNN suffers from significant differences in training and test performances in contrast to *p*-DNN which also delivers satisfactory test performance.

Such a reduction in input space is obtained through *AIC* and *BIC*, which are reported in Table 9, as well as the training times.

Table 10 shows the architectures *p*-DNNs. The selected inputs have 82 % similarity across different simulations employing different DNN architectures. However, *p*-DNN results indicate that u_1, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{14}, and u_{16} should be included in the formulation, in contrast to u_2, u_4, u_{15}, and u_{17}.

4. Conclusions

Sustainable development in chemical engineering is a multi-faceted endeavor, balancing ecological, social, and economic dimensions. The integration IIoT requires the exploitation of complex interactions among process variables to ensure a reliable and robust prediction performance for data driven formulations. In this context, the deployment of advanced machine learning formulations, including deep learning, is a major requirement for sustainable production and consumption in

Table 7
f-DNN and *p*-DNN training and test performance with different neuron numbers and pruning input variables.

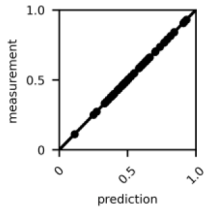
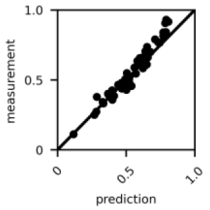
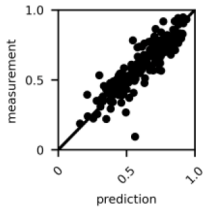
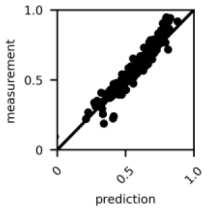
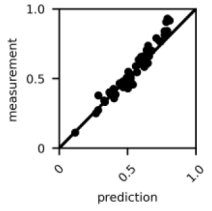
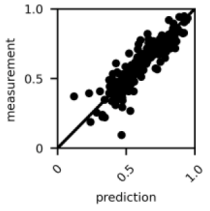
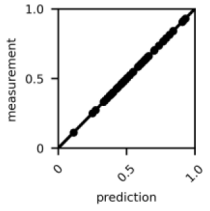
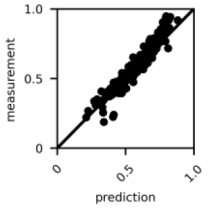
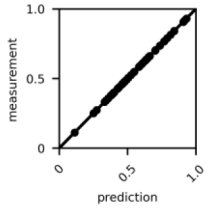
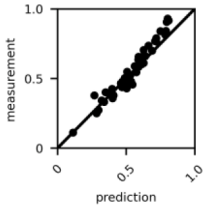
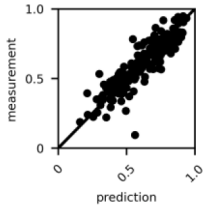
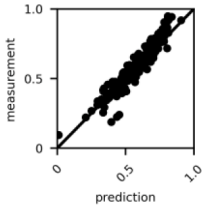
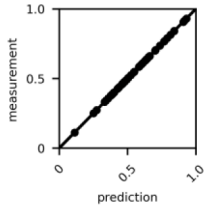
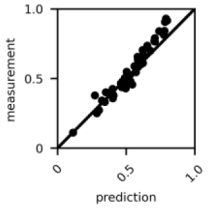
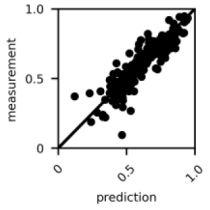
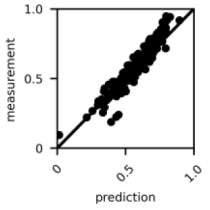
	# Neuron		Training		Test	
			f-DNN	*p*-DNN	*f*-DNN	*p*-DNN
	1st HL	2nd HL				
13 inputs	4	3				
	8	6				
10 inputs	4	3				
	8	6				

Table 8
Some common statistics on the performance of *f*-DNN and *p*-DNN.

				Training			Test		
	# inputs	#N 1st HL	#N 2nd HL	MSE ($x10^{-3}$)	MAPE	R^2	MSE ($x10^{-3}$)	MAPE	R^2
f-DNN	17	4	3	10^{-8}	10^{-4}	0.999	6.24	12.36	0.78
		8	6	10^{-10}	10^{-5}	0.999	5.63	12.17	0.80
p-DNN	13	4	3	2.57	6.99	0.92	2.80	7.97	0.90
		8	6	2.59	7.00	0.92	2.82	8.01	0.90
p-DNN	10	4	3	2.63	7.06	0.92	3.18	8.61	0.89
		8	6	3.03	7.41	0.91	3.46	8.74	0.88

minimizing environmental impacts while ensuring economic viability. Building on this foundation, this study develops an MINLP train and form DNN architecture by selection of high-sensitivity model parameters, which in turn results in input selection, through including BIC and AIC in the training problem to balance the model complexity and fitting performance simultaneously and rigorously. In contrast to traditional unconstrained nonlinear optimization to achieve network weighs under a fixed topology, this formulation is flexible as the training accounts for the existence of particular connections as well as their weights, in turn delivering a robust network with reduced overfitting.

The impact of the model complexity on the predictions has significant outcomes and implications in terms of model robustness and reliability. In theory, challenges increase significantly once the interactions are described by higher number of parameters employed by nonlinear expressions due to increased correlation among the parameters and their impacts on the outputs. Such a theoretical challenge is actually beyond

Table 9
AIC, BIC, and training time for *p*-DNN development.

Method	# inputs	#N 1st HL	#N 2nd HL	AIC	BIC	Time (sec)
p-DNN	13	4	3	2.32	215.35	45.96
		8	6	4.40	418.77	261.17
p-DNN	10	4	3	1.84	168.36	68.73
		8	6	3.43	324.56	871.94

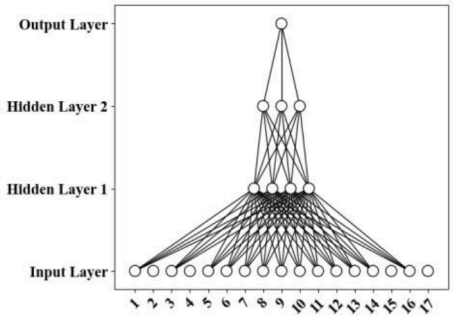
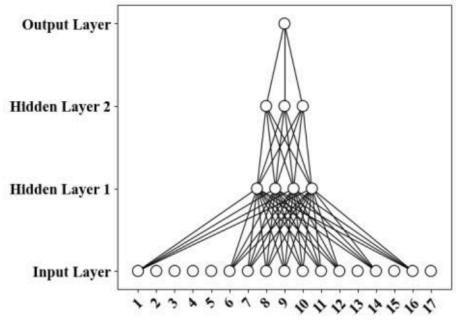
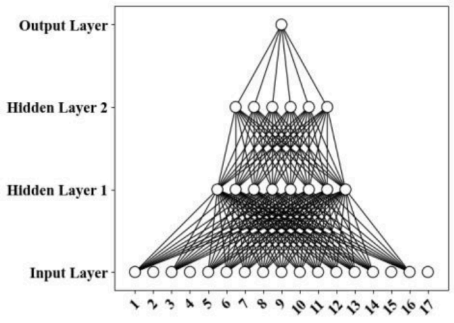
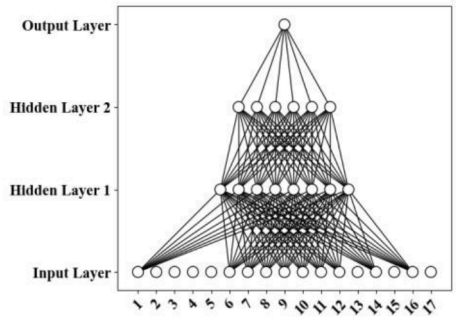
the scope of the neural network architecture development and covers even first principle formulations, resulting in difficult to overcome identifiability issues. In practice, those identifiability issues mostly arise from the lack of spatial and temporal variation of the data, which can not be compensated by collection of large amount of data once, which is a usual case for industrial plants where sensor implementations or frequent laboratory measurements are limited due to safety reasons or mechanical issues. In such cases, a model with representative capability in terms of training and test performance is mostly more reliable once it is constructed upon identifiable parameters. Related unidentifiability issues can statistically and quantitatively be represented and propagated within the model expressions through associated confidence regions, through computation of fisher information matrix and its advanced formulations to calculate the prediction intervals. Moreover, those tasks employ some form of Jacobian formulation which is calculated at a particular point in solution domain, which is computationally challenging once the formulations are highly nonlinear and nonconvex as in the deep learning architecture development. The proposed formulation is built upon statistically proven mathematical issue and do not explicitly address theoretical aspect, since it has already been well studied for various formulations; and left as a future study. On the other hand, our case studies are compatible with the theoretical foundations and show superiority over commonly implemented fully connected deep learning architectures.

The formulation in Eq. (5), in addition to many other considerations, tightens the bounds of deep learning architecture weights through utilization of binary variables. This in turn enables the elimination of the information from corresponding input to the succeeding layers; in other words, the contribution from the input is eliminated. The resulting architecture is insensitive to these input changes and can still perform accurate prediction despite the lacking of the measurement for the corresponding input. Traditional methods, such as usage of large amount of data or regularization formulations to penalize the weight magnitudes through multi objective formulations would not guarantee the elimination of the inputs since the connection still exist after the training. In addition, having a small magnitude weight value for a particular connection between input and the hidden layer does not necessarily bring small sensitivity to the input due to information propagation in the network. Furthermore, the input variable is still necessary to perform predictions, which hinders the real time applications in plants with preserving high amount of sensor requirements. Some sequential pruning algorithms perform retraining after removal of connections which are characterized by small magnitudes, to reduce the computational challenges associated with the simultaneous approach presented in this study.

Proposed formulation is handled through open source BONMIN solver, although more advanced commercial solvers are available (i.e. KNITRO, BARON) to exploit better network architectures with their superiority to treat nonlinear and nonconvex terms through sophisticated reformulations and decompositions, enabling the global optimality. Current approach, although advanced reformulations have not been introduced explicitly, has delivered a DNN architecture with an acceptable accuracy on two industrial datasets. Thus, despite the current solution is local, since the ultimate prediction performance is satisfying on our side, further modifications in the problem solution are not considered and left as a future work.

A major limitation of the proposed rigorous formulation is the complexity of the resulting MINLP which is currently challenging once a high number of data samples are considered. Proposed rigorous approach requires explicit formulation and definition for variables and

Table 10
The representation of network structure of *p*-DNN models.

# Neuron		# inputs	
1st HL	2nd HL	13 inputs	10 inputs
4	**3**		
8	**6**		

constraints, which leads to drastic problem size increase as the number of training samples increase. On the other hand, training size is usually performed offline and rarely; thus, can be performed using high-performance computers and advanced solvers, if needed. In addition, some reformulations and approximations in the nonlinear terms combined with a sophisticated sample selection approach would contribute to the issue.

Despite small number of training data, the current approach delivered similar training and test performance both due to significant reduction in the input space and sensitivity related objective function formulation. On the other hand, the proposed formulation does not need a pre-definition on the number of inputs required for the training and benefits from commonly used model selection statistics to balance the model fitting and its complexity. To favor the computational efficiency, explicit formulations of sensitivity expressions, obtained from algorithmic differentiation, are included in addition to linking constraints to tighten the search space and provide a better convergence rate. Such a sophisticated training problem also makes the DNN more robust to number of hidden layer neurons, which is challenging to determine before solving the optimization problem.

CRediT authorship contribution statement

Damla Yalcin: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation. **Ozgun Deliismail:** Writing – review & editing, Visualization, Validation, Formal analysis, Data curation, Conceptualization. **Basak Tuncer:** Writing – review & editing, Visualization, Validation, Formal analysis, Data curation, Conceptualization. **Onur Can Boy:** Writing – review & editing, Visualization, Validation, Formal analysis, Data curation, Conceptualization. **Ibrahim Bayar:** Writing – review & editing, Validation, Formal analysis, Data curation. **Gizem Kayar:** Writing – review & editing, Validation, Formal analysis, Data curation. **Muratcan Ozpinar:** Writing – review & editing, Validation, Formal analysis, Data curation. **Hasan Sildir:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, 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.

Data availability

The data that has been used is confidential.

References

Agarwal, A., Mishra, S.K., Ram, S., Singh, J.K., 2006. Simulation of runoff and sediment yield using artificial neural networks. *Biosyst. Eng.* 94 (4), 597–613.

Akaike, H. 1998. Information theory and an extension of the maximum likelihood principle. in *Selected papers of hirotugu akaike*, Springer, pp. 199–213.

Alvarez, E., Correa, J.M., Riverol, C., Navaza, J.M., 2000. Model based in neural networks for the prediction of the mass transfer coefficients in bubble columns. Study in Newtonian and non-Newtonianian fluids. *Int. Commun Heat Mass Transf.* 27 (1), 93–98.

Ari, S., Saha, G., 2009. In search of an optimization technique for artificial neural network to classify abnormal heart sounds. *Appl. Soft Comput.* 9 (1), 330–340.

Bakshi, B.R., 2019. Toward sustainable chemical engineering: The role of process systems engineering. *Annu. Rev. Chem. Biomol. Eng.* 10, 265–288.

Behkish, A., Lemoine, R., Sehabiague, L., Oukaci, R., Morsi, B.I., 2005. Prediction of the gas holdup in industrial-scale bubble columns and slurry bubble column reactors using back-propagation neural networks. *Int. J. Chem. Reactor Eng.* 3 (1).

Bergamini, R., Nguyen, T.-V., Elmegaard, B., 2019. Simplification of data acquisition in process integration retrofit studies based on uncertainty and sensitivity analysis. *Front. Energy Res.* 7, 108.

Better regulation: guidelines and toolbox." Accessed: Oct. 16, 2023. [Online]. Available: https://commission.europa.eu/law/law-making-process/planning-and-proposing-law/better-regulation/better-regulation-guidelines-and-toolbox_en.

Bortolan, G., Fusaro, S. 1996. Feature reduction and RBF in classifiers based on ANN. in *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 1996, pp. 925–926.

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., De Felice, F., 2020. Artificial intelligence and machine learning applications in smart production: progress, trends, and directions. *Sustainability*. 12 (2), 492.

Dantas, T.E.T., de-Souza, E.D., Destro, I.R., Hammes, G., Rodriguez, C.M.T., Soares, S.R., 2021. How the combination of circular economy and industry 4.0 can contribute towards achieving the sustainable development goals. *Sustain. Prod. Consum.* 26, 213–227.

Dobbelaere, M.R., Plehiers, P.P., Van de Vijver, R., Stevens, C.V., Van Geem, K.M., 2021. Machine learning in chemical engineering: strengths, weaknesses, opportunities, and threats. *Engineering* 7 (9), 1201–1211.

Ennett, C.M., Frize, M. 2000. Selective sampling to overcome skewed a priori probabilities with neural networks. in *Proceedings of the AMIA Symposium*, American Medical Informatics Association, p. 225.

Ganesh, M.R., Blanchard, D., Corso, J.J., Sekeh, S.Y., 2022. Slimming neural networks using adaptive connectivity scores. *IEEE Trans. Neural Netw. Learn. Syst.*

Gao, H., Zhu, L.-T., Luo, Z.-H., Fraga, M.A., Hsing, I.-M. 2022. Machine learning and data science in Chemical engineering. *Industrial & Engineering Chemistry Research*. 61(24). ACS Publications, pp. 8357–8358, 2022.

Ginocchi, M., Ponci, F., Monti, A., 2021. Sensitivity analysis and power systems: can we bridge the gap? A review and a guide to getting started. *Energies (Basel)* 14 (24), 8274.

Hazare, S.R., et al., 2022. Predictive analysis of gas hold-up in bubble column using machine learning methods. *Chem. Eng. Res. Des.* 184, 724–739.

Helton, J.C., 1993. Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliab. Eng. Syst. Saf.* 42 (2–3), 327–367.

Helton, J.C., Johnson, J.D., Sallaberry, C.J., Storlie, C.B., 2006. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab. Eng. Syst. Saf.* 91 (10–11), 1175–1209.

Jamialahmadi, M., Zehtaban, M.R., Müller-Steinhagen, H., Sarrafi, A., Smith, J.M., 2001. Study of bubble formation under constant flow conditions. *Chem. Eng. Res. Des.* 79 (5), 523–532.

Jamwal, A., Agrawal, R., Sharma, M., 2022. Deep learning for manufacturing sustainability: models, applications in Industry 4.0 and implications. *Int. J. Inf. Manage. Data Insights*. 2 (2), 100107.

Lal, A., Radhakrishnan, S., Srinivas, S.S., Najarian, K., Mays, L.E., 2004. Splice site detection using pruned maximum likelihood model. In: In *the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2836–2839.

Leonard, K.C., Hasan, F., Sneddon, H.F., You, F., 2021. Can artificial intelligence and machine learning be used to accelerate sustainable chemistry and engineering? *ACS Sustain. Chem. Eng.* 9 (18), 6126–6129. ACS Publications.

Li, Y., Zhang, T., Fang, Y., Fan, X., 2023. Method of selecting active parameters using sensitivity analysis and linear programming. *Combust. Sci. Technol.* 1–17.

López-Guajardo, E.A., Delgado-Licona, F., Álvarez, A.J., Nigam, K.D.P., Montesinos-Castellanos, A., Morales-Menendez, R., 2022. Process intensification 4.0: a new approach for attaining new, sustainable and circular processes enabled by machine learning. *Chem. Eng. Process.-Process Intensif.*. 180, 108671.

Loucks, D.P., Van Beek, E., 2017. *Water Resource Systems Planning and Management: An Introduction to Methods, Models, and Applications*. Springer.

Ning, C., You, F., 2019. Optimization under uncertainty in the era of big data and deep learning: when machine learning meets mathematical programming. *Comput. Chem. Eng.* 125, 434–448.

Perry, M.A., Wynn, H.P., Bates, R.A., 2006. Principal components analysis in sensitivity studies of dynamic systems. *Probab. Eng. Mech.* 21 (4), 454–460.

Qin, C., Jin, Y., Tian, M., Ju, P., Zhou, S., 2023. Comparative study of global sensitivity analysis and local sensitivity analysis in power system parameter identification. *Energies (Basel)* 16 (16), 5915.

Razavi, S., et al., 2021. The future of sensitivity analysis: an essential discipline for systems modeling and policy support. *Environ. Model. Softw.* 137, 104954.

Razavi, S., 2021. Deep learning, explained: fundamentals, explainability, and bridgeability to process-based modelling. *Environ. Model. Softw.* 144, 105159.

Ren, F., Wang, J.-F., Li, H.-S., 2006. Direct mass production technique of dimethyl ether from synthesis gas in a circulating slurry bed reactor. *Stud. Surf. Sci. Catal.* 489–492.

Ricotti, M.E., Zio, E., 1999. Neural network approach to sensitivity and uncertainty analysis. *Reliab. Eng. Syst. Saf.* 64 (1), 59–71.

Rollbusch, P., et al., 2015. Bubble columns operated under industrially relevant conditions–current understanding of design parameters. *Chem. Eng. Sci.* 126, 660–678.

Saltelli, A., et al., 2019. Why so many published sensitivity analyses are false: a systematic review of sensitivity analysis practices. *Environ. Model. Softw.* 114, 29–39.

Saltelli, A., et al., 2020. Five ways to ensure that models serve society: a manifesto. Nat. Publ. Group.

Saltelli, A., Annoni, P., 2010. How to avoid a perfunctory sensitivity analysis. *Environ. Model. Softw.* 25 (12), 1508–1517.

Saltelli, A., Jakeman, A., Razavi, S., Wu, Q., 2021. Sensitivity analysis: a discipline coming of age. *Environ. Model. Softw.* 146, 105226.

Saltelli, A., Sobol, I.M., 1995. About the use of rank transformation in sensitivity analysis of model output. *Reliab. Eng. Syst. Saf.* 50 (3), 225–239.

Schwarz, G., 1978. Estimating the dimension of a model. *Ann. Stat.* 461–464.

Schweidtmann, A.M., et al., 2021. Machine learning in chemical engineering: a perspective. *Chem. Ing. Tech.* 93 (12), 2029–2039.

Shang, C., Yang, F., Huang, D., Lyu, W., 2014. Data-driven soft sensor development based on deep learning technique. *J. Process Control.* 24 (3), 223–233.
Sildir, H., Aydin, E., Kavzoglu, T., 2020. Design of feedforward neural networks in the classification of hyperspectral imagery using superstructural optimization. *Remote Sens. (Basel).* 12 (6), 956.
Supardan, M.D., Masuda, Y., Maezawa, A., Uchida, S., 2004. Local gas holdup and mass transfer in a bubble column using an ultrasonic technique and a neural network. *J. Chem. Eng. Jpn.* 37 (8), 927–932.
Trinh, C., Meimaroglou, D., Hoppe, S., 2021. Machine learning in chemical product engineering: the state of the art and a guide for newcomers. *Processes*. 9 (8), 1456.
Utomo, M.B., Sakai, T., Uchida, S., Maezawa, A., 2001. Simultaneous measurement of mean bubble diameter and local gas holdup using ultrasonic method with neural network. *Chem. Eng. Technol.* 24 (5), 493–500.
Venkatasubramanian, V., 2019. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE J.* 65 (2), 466–478.
Vinuesa, R., et al., 2020. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nat. Commun.* 11 (1), 1–10.
Wallace, S.W., 2000. Decision making under uncertainty: Is sensitivity analysis of any use? *Oper. Res.* 48 (1), 20–25.
Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* 48, 144–156.
吴元欣, 罗湘华, 陈启明, 李定或, and 李世荣. Prediction of gas holdup in bubble columns using artificial neural network. 中国化学工程学报: 英文版. 11(2), pp. 162–165, 2003.
Wu, Z., Wang, H., He, C., Zhang, B., Xu, T., Chen, Q., 2023. The application of physics-informed machine learning in multiphysics modeling in chemical engineering. *Ind. Eng. Chem. Res.*
Wuest, T., Weimer, D., Irgens, C., Thoben, K.-D., 2016. Machine learning in manufacturing: advantages, challenges, and applications. *Prod. Manuf. Res.* 4 (1), 23–45.
Zhang, L., Babi, D.K., Gani, R., 2016. New vistas in chemical product and process design. *Annu. Rev. Chem. Biomol. Eng.* 7, 557–582.
Zhao, C., 2022. Perspectives on nonstationary process monitoring in the era of industrial artificial intelligence. *J. Process Control.* 116, 255–272.
Zheng, Y., Keller, A.A., 2006. Understanding parameter sensitivity and its management implications in watershed-scale water quality modeling. *Water Resour Res.* 42 (5).
Zhong, S., et al., 2021. Machine learning: new ideas and tools in environmental science and engineering. *Environ. Sci. Tech.* 55 (19), 12741–12754.