Information Sciences 566 (2021) 195–214

Contents lists available at ScienceDirect

Information Sciences

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

Echo-state networks for soft sensor design in an SRU process

Luca Patanè *, Maria Gabriella Xibilia

Dipartimento di Ingegneria, Università degli Studi di Messina, Contrada di Dio, 98166 Messina, Italy

ARTICLE INFO

Article history:
Received 25 August 2020
Received in revised form 22 February 2021
Accepted 7 March 2021
Available online 13 March 2021

Keywords:
Soft sensors
Reservoir computing
LSTM
Dynamical models
Refinery plants

ABSTRACT

The implementation of soft sensors for industrial processes is expanding in applications for recent machine learning techniques. In this work, strategies based on reservoir computing are applied to developing dynamical models of target variables in a sulfur recovery unit (SRU) of a refinery plant in Italy. In particular, a specific type of recurrent network, namely an echo-state network (ESN), is adopted to estimate key process variables on the SRU. Two process lines are considered to evaluate the proposed algorithm on different datasets in terms of estimation performance and computational effort of the learning process. The obtained results are evaluated in comparison with other recurrent networks, based on long short-term memory, and with other techniques reported in the literature, demonstrating the feasibility of the proposed approach. Furthermore, the introduction of intrinsic plasticity (IP) is also considered to adapt the reservoir parameters to the provided inputs, achieving a significant improvement in the statistical distribution of the results obtained for the pool of learned networks. The reported results show that ESN-IP represents a suitable solution for identifying dynamical models of the industrial processes, avoiding the time-consuming regressor selection procedure, which is needed when a static network is adopted to design a dynamical model.

1. Introduction

The principles at the basis of Industry 4.0 presuppose the continuous monitoring of processes and working conditions. An efficient measurement system is a fundamental pillar of this strategy. Physical and economic constraints are often in contrast with the tight monitoring and control of process variables. In these scenarios, the opportunity to adopt process models known as soft sensors (SSs), which are devoted to the real-time estimation of relevant variables, represents a significant turning point. SSs can be applied along with hardware measuring devices for the implementation of feedback control strategies. The advantages include the introduction of measurement redundancy (avoiding plant shut down during hardware sensor maintenance), fault detection, and cost reduction. The real-time prediction of the output variables also outperforms hardware devices that can cause considerable delays. The use of SSs in industrial applications and, in particular, the process industry is widely addressed in the literature [12,39,27]. Formulating first-principles mathematical models for industrial processes is challenging because of the nonlinear nature of the physical phenomena involved. An alternative solution, often adopted for industrial SSs, is based on data-driven identification procedures.

In most cases, the design of SSs for industrial processes involves nonlinear methodologies, owing to the process complexity. Nevertheless, linear approaches show significant advantages when the process works close to one or more working

* Corresponding author.
E-mail address: lpatane@unime.it (L. Patanè).

https://doi.org/10.1016/j.ins.2021.03.013
0020-0255/

points and can be, therefore, approximated with linear models around them. Classical linear regression methods like autoregressive (ARX) models or partial least-square regression (PLSR) have been widely discussed in the literature [12]. They are generally used as components of more elaborate modelling procedures, including adaptation strategies. Jiang et al. proposed an optimised sparse partial least square (OSPLS) method to cope with batch-end quality modelling [17]. Linear methods are also beneficial in terms of the interpretability of the designed SS and computational effort. However, to achieve accurate predictions in complex nonlinear processes, a mixture of several linear systems is required. Souza and Araúio proposed a mixture of univariate linear regression models (MULRM) in time-varying scenarios [26]. Nonlinear strategies have been used to improve the model prediction for linear models when complex processes are considered, obtaining a single model covering the whole dynamic of the process [12,27]. Nonlinear machine learning techniques applied in SS design range from classical multilayer perceptrons and support vector machines [12,15] to more recent deep architectures [37], like deep belief networks (DBNs)[24], long short-term memory networks (LSTMs) [43,18], stacked autoencoders (SAEs) [46,38,40,47], Bayesian approaches and adaptive methods [36,22]. Deep networks based on SAEs were adopted by [16] to realise a local-global model distributed framework applied for efficient fault detection in nonlinear plant-wide processes. Yuan et al. proposed a hierarchical structure based on stacked quality-driven autoencoders for learning quality-relevant features [48]. Data augmentation in deep networks was treated in [45], where a simple linear interpolation strategy applied in each layer during the pre-training phase. Wang et al. proposed an extended DBN for feature representation and fault diagnosis in chemical processes, where raw data was used in combination with the hidden feature to mitigate information loss [34]. Other interesting methodologies include the development of time-varying models that further improve the SS performance at the cost of additional complexity in the parameter identification and online implementation [36,22]. Several studies reported in the literature try to solve the identification problem using static models designed with different approaches. In [21], statistical and soft computing models have been proposed for identifying an industrial process, presenting a Gaussian regression as the best performing architecture. However, in most industrial processes, better results have been obtained by developing dynamical models [18,1].

One of the open problems in dynamical SS design, mainly when nonlinear autoregressive models with exogenous inputs (NARX) or nonlinear finite impulse response (NFIR) models are considered, is the input selection phase of both independent input variables and regressors. Despite the many approaches proposed in the literature based on regularisation, correlation analysis or mutual information computation [9], this design phase is still complex and time-consuming but can be alleviated by recurrent structures.

In this work, we evaluate the application of a new class of recurrent networks to develop SSs. Nonlinear dynamical models are implemented, extracting the temporal dependencies among the input data through a projection into a lattice of interconnected neurons, called a reservoir. The presence of recurrent connections allows the formation of internal memory. From this perspective, reservoir computing (RC) is promising for design data-driven models of industrial processes where the nonlinear dynamic behaviour is relevant. Among the different specific architectures of the RC methodology, we considered the commonly used echo-state networks (ESNs), which include handwriting recognition [6] and time series forecasting [35]. Unlike other neural structures, ESNs do not require the explicit identification of the dynamical input-output dependencies in terms of regressors [11]. The increase of dimensionality from input to hidden neurons, which present recurrent connections, allows for a rich combination of dynamics that can be exploited depending on the requested output to be learned. Furthermore, the learning process is limited to the output weights, namely the read-out map, significantly reducing the learning time.

The use of an ESN for time-series prediction is expanding, owing to the simplicity of the learning strategy concerning other recurrent neural structures [20]. The classical ESN architecture can be adjusted to improve the accuracy of prediction. In [13], a singular value decomposition (SVD) method was applied to an ESN (SVD-ESN) to calculate the read-out weights, increasing the accuracy. A particle swarm optimisation algorithm (PSO) was used in [8] to optimise the input connection weights, enhancing the network consistency. In [42], an adaptive ESN (AESN) was proposed as an extension of a Leaky-ESN for modelling discrete-time dynamic nonlinear systems. Applications to a synthetic NARX system and the Mackey-Glass chaotic system were also reported. Additional applications that have been investigated relate to time-series prediction in financial markets [41] and the medical field to classify the actions of Parkinson's patients [19]. However, the use of reservoir-based methods in the industrial field for SS design is still minimal. Related approaches have been recently proposed, demonstrating the potential advantages of ESNs when multiple hidden layers are considered, which have resulted in deep ESNs [5]. Manifold reduction techniques to improve the noise robustness, further applying the concept of neural reuse, were also exploited [2,3].

In this paper, we chose to adopt ESNs with a single hidden layer to improve the process identification with the introduction of an intrinsic plasticity (IP) rule [23,49]. Neural plasticity applied to artificial neural networks can be divided into two main categories: synaptic plasticity that regulates the weight adaptation and IP devoted to shaping the neuron activation function [49,31]. Several models have been proposed to optimise the ESN capabilities to include synaptic plasticity in the internal reservoir. In most cases, global mechanisms based on a self-regulated Hebbian rule were applied. To further improve the learning performance, local strategies were also adopted, specialising parameters or even rules at the level of every single neuron, as proposed in Wang et al. [33], where an evolution strategy with covariance matrix adaptation was reported. In our work, we selected a biologically inspired IP strategy, which is performed during a pre-training phase, to maximise the information transfer from the input data through the hidden layer [31]. This learning process aims at reducing the variance of the distribution related to network performance, improving the statistical significance of the obtained results. This adaptation

mechanism has been successfully applied in a legged robot's central pattern generator to deal with different environmental conditions [10]. Following this research line, we propose an application of ESN to a sulfur recovery unit (SRU) industrial process. The SRU is a crucial processing unit in a refinery plant [12], devoted to removing environmental pollutants from acid gas streams released into the atmosphere. The time evolution of the concentration of two different acid gases is modelled to realise SSs for applications in the feedback control loop. Results are compared with those obtained using an LSTM network, which was recently applied for modelling similar phenomena [18]. The integration of a spatiotemporal attention mechanism in an LSTM was considered in [43] for an industrial hydrocracking process. In [44], a supervised LSTM was proposed to learn quality-relevant hidden dynamics, improving the prediction performance of the developed model.

The proposed work aims to demonstrate the suitability and advantages of the reservoir-based neural structure for the design of SSs for nonlinear dynamical industrial processes. The SS performance is analysed from different perspectives. We consider the quality variable prediction accuracy and other indexes relevant to the process analysis, such as the peak error and the stoichiometric ratio between the output variables. Furthermore, the ESN learning process and IP-rule impact are also investigated in terms of prediction performance and required computational effort during the learning process.

The proposed techniques are subsequently applied to a different dataset related to another processing line of the SRU, which is widely adopted in literature, for an extensive comparison with the state of the art in the field. This choice allowed us to briefly investigate the opportunities related to transfer learning, another relevant topic in SS development. The paper is organised starting with Section 2, which describes the different strategies adopted for SS modelling; in Section 3, the industrial process used as a test-bed for the evaluation of the considered neural architectures is presented. Results obtained on line 2 of the SRU process are outlined in Section 4, whereas a statistical comparison among the considered solutions is investigated in Section 5. Section 6 includes the application of the proposed architecture to the processing line 4 of the SRU, including comparisons with the literature. Finally, conclusions are drawn in Section 7.

2. Recurrent neural architectures

The design of an SS can be performed using static models, which estimate the requested variables based on relevant inputs at the current time without considering their temporal evolution. In industrial plants, this solution is not often applicable, owing to the dynamical nature of the processes involved. To implement dynamical SSs, two different strategies can be pursued. The first is based on using a set of regressors (i.e. current and past values of input and output variables), representing a NARX model, which are implemented with a nonlinear feedforward learning structure [1]. In contrast, the second strategy uses the current independent input values, without regressors, as the input of a recurrent structure [7,15,32,18]. Recurrent neural networks (RNNs) are often applied to model dynamical systems, obtaining better results than standard feedforward networks thanks to the presence of recursive connections. RNNs are capable of storing relevant information on the modelled physical process [25]. The main problem of RNNs is the learning process, which is time-consuming and subject to the vanishing gradient problem. To overcome these limitations, a different methodology, known as reservoir computing, is proposed in this paper [20]. Reservoir-like structures do not require the time-consuming regressor selection procedure, resulting in a simpler learning strategy compared with other RNNs.

2.1. ESN for soft sensors

ESNs are particular types of RNNs based on the paradigm of reservoir computing. These networks were introduced in [20] and consist of an input layer connected with a massive, randomly generated set of nodes with recurring connections (i.e. reservoir), whose weights remain fixed, and an output layer (i.e. the read-out map) subjected to learning. One of the fundamental characteristics of ESNs is the echo state property, which allows previous states to echo even after they have passed. This property is directly related to the spectral radius (ρ), which is used to scale the eigenvalues of the internal weight matrix. The spectral radius affects the linearity of the system. Having high ρvalues, which usually do not exceed one, will require nonlinear models. Increasing this parameter increases the instability of the system together with the time window for which previous inputs are maintained in memory. The typical update equation of system states is reported in the following:

$$\begin{cases} \mathbf{x}(t+1) = \alpha f\left(\mathbf{W}_{ux}\mathbf{u}(t+1) + \mathbf{W}_{x}\mathbf{x}(t) + \mathbf{W}_{yx}\mathbf{y}(t)\right) + (1-\alpha)\mathbf{x}(t) \\ \mathbf{y}(t) = \mathbf{W}_{xy}\mathbf{z}(t) \end{cases} \tag{1}$$

where $\mathbf{z}(t)$ is the extended state given by the concatenation $[\mathbf{x}(t);\mathbf{u}(t)]$ that represents the state and input vectors, respectively; $\alpha \in]0\ 1]$ is the forgetting factor, adopted when leaky integrator neurons are considered; and $f(\cdot)$ is usually an hyperbolic tangent function. Furthermore, we define $\mathbf{W}_{ux} \in \mathbb{R}^{n_u,n_x}$, $\mathbf{W}_{x} \in \mathbb{R}^{n_x,n_x}$, $\mathbf{W}_{xy} \in \mathbb{R}^{n_x,n_y}$ and $\mathbf{W}_{yx} \in \mathbb{R}^{n_y,n_x}$ as the weight matrices for input-to-reservoir, reservoir-to-reservoir, reservoir-to-output and output-to-reservoir connections, respectively, where n_i represents the dimension of the vector $i \in \{x, y, u\}$. These matrices are randomly initialised, except for $\mathbf{W}_{xy}$, which is subject to learning. A scheme showing the ESN layers and neuron connections is reported in Fig. 1.

The learning strategy adopted to estimate the reservoir-to-output weights, representing the read-out map, is based on a one-step learning phase that considers the whole internal activity obtained processing the complete learning dataset. This

neural activity, which evolves in space (i.e. network neurons) and time, can be expressed as a matrix $\mathbf{Z} \in \mathbb{R}^{n_p, n_x+n_u}$, where n_p is the number of learning patterns.

The read-out map weights are obtained with the following equation:

$$\mathbf{W}_{xy} = \mathbf{Z}^{\dagger}\mathbf{T} \tag{2}$$

where $\mathbf{Z}^{\dagger}$ is the Moore-Penrose matrix of $\mathbf{Z}$, and $\mathbf{T} \in \mathbb{R}^{n_p}$ indicates the actual output signals. The Moore-Penrose matrix evaluation process can consider a regularisation coefficient to avoid hill-posed conditions. In this case, $\mathbf{Z}^{\dagger}$ is evaluated as follows:

$$\mathbf{Z}^{\dagger} \doteq \left[\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + \beta\mathbf{I}\right]^{-1}\mathbf{Z}^{\mathrm{T}} \tag{3}$$

where β is a small constant, and $\mathbf{I}$ is the identity matrix.

To initially activate the network, the first part of the provided data is discarded from the learning and testing process (i.e. washout phase). The main functions of the reservoir are to act as a dynamic expansion and keep a memory of its inputs. As already mentioned, the creation of this large tank is entirely random; therefore, the study of the hyperparameters used to define the reservoir structure and adapt to the process under consideration is necessary. The hyperparameters taken into account during the network optimisation are the spectral radius, the reservoir size, input and teacher scaling, and shift. The feedback connections from the output neurons and the leaking factor α (fixed to $\alpha = 1$) are not included in the optimisation strategy because they negatively affect the overall performance, based on preliminary analyses.

A searching range for the hyperparameters is defined based on the standard values in the literature [32] and based on preliminary simulations. If the best performance is detected on the covered range boundary, the limits are enlarged to determine any margin of improvement. Hyperparameter optimisation is performed following an iterative search; they are ordered based on their expected impact on the results. Starting from the first hyperparameter, ten networks are initialised and learned for each value in the selected range, initialising the others to default values. When the obtained results overperform the previous best solution, in terms of validation error, the current hyperparameters are stored, substituting the last default values, and the next configuration is investigated. This process is iterated until either no other improvements are detected for a complete cycle or a maximum number of iterations is reached. The steps followed for the design of each SS are reported in the Algorithm 1.

Algorithm 1: Algorithm followed for the SS design using an ESN. *HyperParam_identification* and *HyperParam_range_definition* functions are based on the indications provided in [32].

```
Outliers identification and interpolation
Z-score normalisation
Train - Tr, validation - Va and test - Te sets creation
HyperParamList = HyperParam_identification
HyperParamRange = HyperParam_range_definition
BestFound=True; MaxTrials=10; MaxIters=5; BestNMRSE=10; MaxNets=100
while(iter<MaxIters) and (BestFound)do
  forHyperParam ∈ HyperParamListdo
    forHyperParamValue ∈ HyperParamRangedo
    fori = 1 to MaxTrialsdo
      Z=ESNevolution(Tr)
      W_xy = Z†T
      ifNMRSE(Va)<BestNMRSEthen
        BestNMRSE=NMRSE(Va)
        BestFound=True
        DefaultHyperParam=CurrentHyperParam
      end if
    end for
  end for
  end for
  iter++
end while
fori = 1 to MaxNetsdo
  Z=ESNevolution(Tr)
  W_xy = Z†T
  NMRSE(Te); R(Te)
end for
```

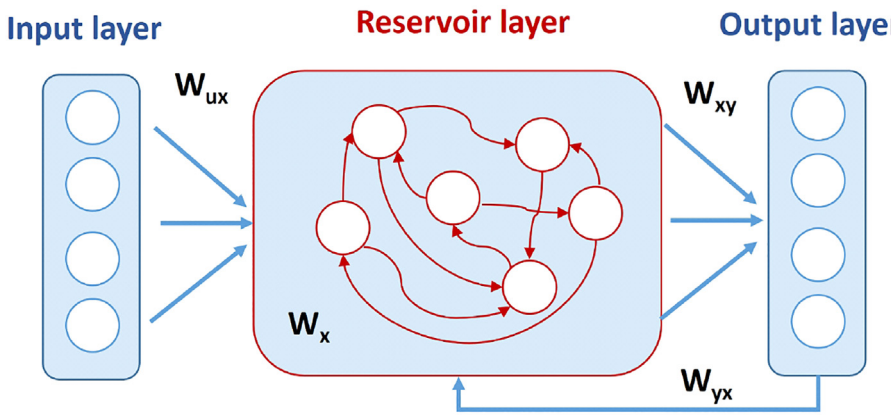

Fig. 1. Scheme of an ESN where the different neural connections are identified.

The requested computational effort is reduced if compared with a complete grid search. Other hyperparameter optimisation methods, based on genetic algorithms and different bio-inspired approaches, have been applied in recent works [29,30] and will be considered for further analyses.

2.2. Intrinsic plasticity

A significant problem related to ESNs is the considerable number of parameters optimised by trial and error. The uncertainty is given by a reservoir, whose weight matrices are generated randomly. The ESN learning process can be improved using an unsupervised rule based on the maximisation of the transferred information, called IP [28].

This rule is derived from a biological mechanism known as homeostatic plasticity: the persistent modification of the intrinsic electrical properties of a neuron based on synaptic activity. The application to an ESN mitigates potential problems due to the generation of a random reservoir, making it more robust and able to adapt its internal dynamics to the system to be modelled. The rule is entirely driven by the input signals and adapts the reservoir in an unsupervised manner, independently from the target signals to be learned with the read-out maps.

The first investigations reported in [31] demonstrated that an exponential output firing rate distribution maximises the information carried. The main objective was to formalise a methodology that maximises the flow of information without overcoming the constraints on the energy consumption of each neuron. IP concretises these considerations through the introduction of three fundamental principles: maximum transportable information, constraints on output distribution and the adaptation of the neuron intrinsic parameters.

Based on these principles, the aim is to impose limits on the average distribution of the neuron output and generate a gradient descent rule based on these limits.

The internal neuron activation function will change: $f_{IP}(x_n) = f(ax_n + b)$, where a and b are the vectors of gain and bias, respectively, given by the IP rule that leads to a maximum entropy distribution [23]; x_n is the neuron input vector; and the adopted transfer function f is the hyperbolic tangent.

To calculate the difference between the real distribution of the output and the desired maximum entropy distribution, which is a Gaussian function for our input domain, the Kullback-Leibler divergence can be adopted [28].

The internal neuron parameters are updated following an online learning rule with stochastic gradient descent:

$$\Delta b = -\eta\left(-\frac{\mu}{\sigma^2} + \frac{y_n}{\sigma^2}(2\sigma^2 + 1 - y_n^2 + \mu y_n)\right) \tag{4}$$

$$\Delta a = \frac{\eta}{a} + \Delta b x_n \tag{5}$$

where η is the learning rate, $y_n = tanh(ax_n + b)$ is the neuron output, μ and σ are the mean and variance of the targeted Gaussian process, respectively, and $\Delta a, \Delta b$ are the incremental updates of the previously defined neuron parameters.

A scheme of the IP procedure applied to the adopted ESN is reported in Fig. 2. The ESN output layer is not involved in the analysis as the IP parameters optimisation is performed during a pre-training phase without needing information on the output variables.

2.3. LSTM architectures

LSTM is a particular type of RNN employed to solve classification, regression and time series prediction. LSTM is usually adopted in the presence of considerable delay between two significant events in the series. An LSTM network consists of several blocks capable of storing a variable for an arbitrary period. Each block is composed of several nodes with cross and recurrent connections.

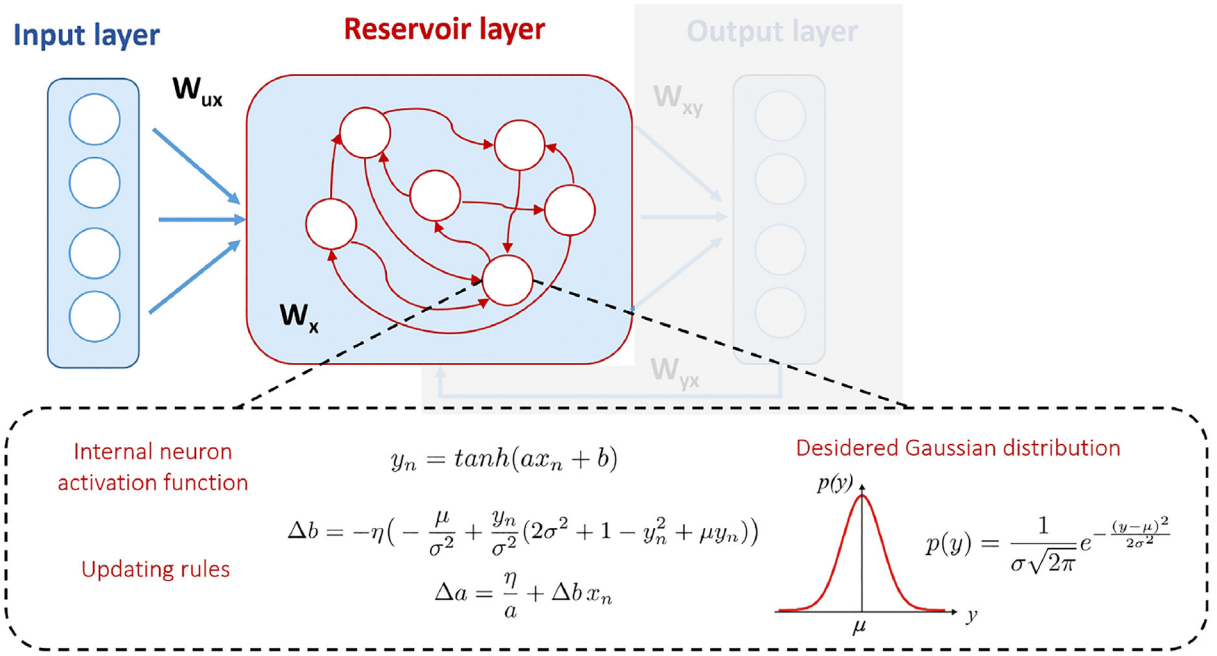

Fig. 2. Scheme describing the application of IP rules to the ESN. The internal neuron parameters (i.e. a and b) are updated using a gradient descent rule to maximise the information transfer. The desired target distribution is a Gaussian function with a given mean and variance (μ, σ). This pre-training procedure is independent of the output variables.

The architecture of the LSTM can be described through the interaction of memory cells with particular structures, called gates, that can modulate the information flow over time. Gate connections are the network parameters used to decide what to store in memory, how long to keep it, and when to read it. This structure improves the information storage as compared to a standard RNN. There are three types of gates: input, forget and output. The LSTM network suffers fewer problems like those described for RNN concerning the vanishing gradient, thanks to gate functions. Further details, including the learning algorithm, are reported in [25].

3. SRU description

The neural structures and learning methods previously introduced are applied to a real case of study. In this work, two SSs for a refinery located in Sicily, Italy, are developed. The proposed SSs estimate the concentration of the acid gases–hydrogen sulfide (H_2S) and sulfur dioxide (SO_2)– in the tail stream of an SRU.

Sulfur recovery is a key processing unit in a refinery plant that removes environmental pollutants from acid gas streams that to be released into the atmosphere. Furthermore, elemental sulfur, a valuable by-product, can be recovered during the process.

The SRU considered in this work consists of four identical subunits (i.e. sulfur lines) working in parallel, each capable of extracting sulfur from acid gases at a rate of 100 tons/day (see Fig. 3).

Each sulfur line receives two kinds of acid gases as input: MEA gas, which comes from the gas washing plants, and SWS gas, rich in H_2S and NH_3, which comes from the Sour Water Stripping plant. The sulfur extraction is performed in reactors, where H_2S is subject to a partial oxidation reaction with air. The remaining gas is fed to the Maxisulfur plant for a different extraction phase. The final gas stream (tail gas) from the SRU contains residual H_2S and SO_2, which needs to be controlled. Air, which supplies oxygen for the reaction, is essential for converting acid gases and responsible for the tail gas composition. The process is difficult to control because excessive airflow increases the concentration of SO_2 with respect to H_2S, while a low airflow rate does the opposite. At present, an online analyser is used to measure the concentration of H_2S and SO_2 in the tail gas of each sulfur line. It also measures the value of $[H_2S] - 2[SO_2]$ (where the brackets indicate concentration) to monitor the performance of the conversion process and control the air-to-feed ratio in the SRU for improved sulfur extraction. The desired value of $[H_2S] - 2[SO_2]$ is zero, which indicates that these pollutants are absent in the tail gas or the reactants are in stoichiometric proportion, which is optimal for the total removal of the sulfur.

Control is improved by a closed-loop algorithm that regulates an additional airflow (AIR_MEA_2) based on the tail gas composition analysis. Acid gases cause damage to sensors, which require frequent maintenance. When the analyser is off-line for maintenance, this control loop cannot work, and the performance of the SRU worsens. Therefore, an SS is needed to replace the off-line analyser. In addition, the SS provides a redundant estimation of the variables of interest for fault detection purposes.

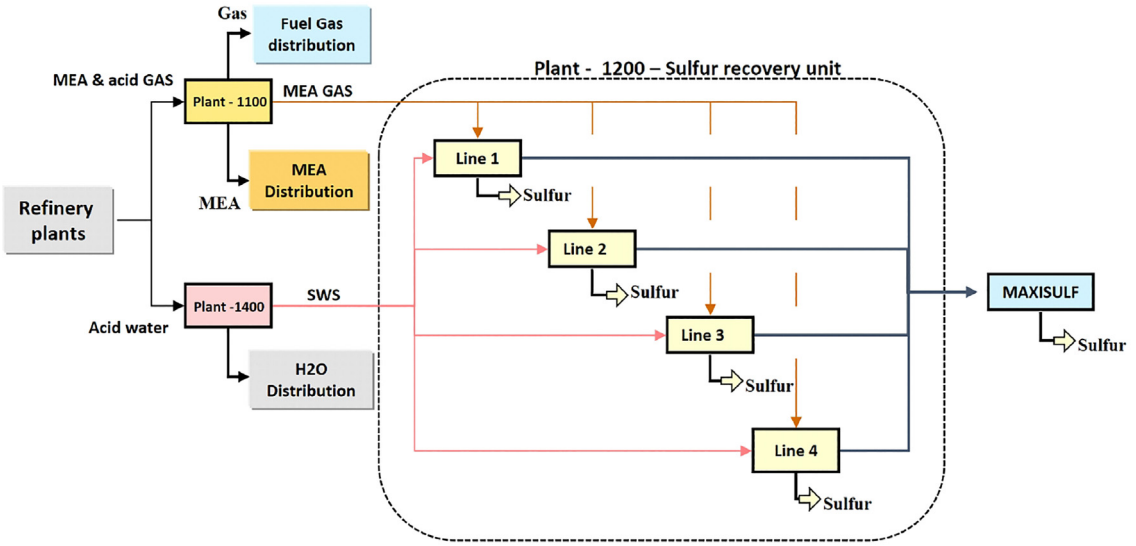

Fig. 3. Block scheme of the sulfur recovery system with four lines, used in a refinery located in Sicily.

The SRU is often used as a benchmark for SS design. Most of the results available in the literature apply to sulfur line 4 [12,5,18,15,4]. In this work, we initially consider the data acquired from line 2. To further analyse the application of the proposed methodology, the data related to sulfur line 4 are also evaluated, comparing the obtained results with some alternative architectures available in the literature. For the reproducibility of the proposed approaches, the normalised input and output data, which were used for sulfur line 2 SS development, are provided in the supplemental material, whereas the data related to line 4 are available in [12].

A simplified scheme related to a single SRU processing line is reported in Fig. 4. The five inputs are the MEA and SWS gases, the corresponding air flows AIR_MEA and AIR_SWS, and the further airflow input AIR_MEA_2, which is determined using the closed-loop control system based on the analysis of the tail gas.

In the considered line, two SSs are developed, one for SO_2 and the other for H_2S. In addition to the prediction accuracy, $[H_2S] - 2[SO_2]$ is evaluated. The presence of peaks in the variables represents a critical condition that, if correctly predicted, allows for process control improvement. Based on suggestions provided by experts in the industry, the critical threshold in the peak analysis is the 30% of the mean value.

4. Simulation results

4.1. Data analysis

As previously introduced, the system consists of four parallel lines that perform the same task independently. Each line receives as input the same type of substances with different gas concentrations and flow rates.

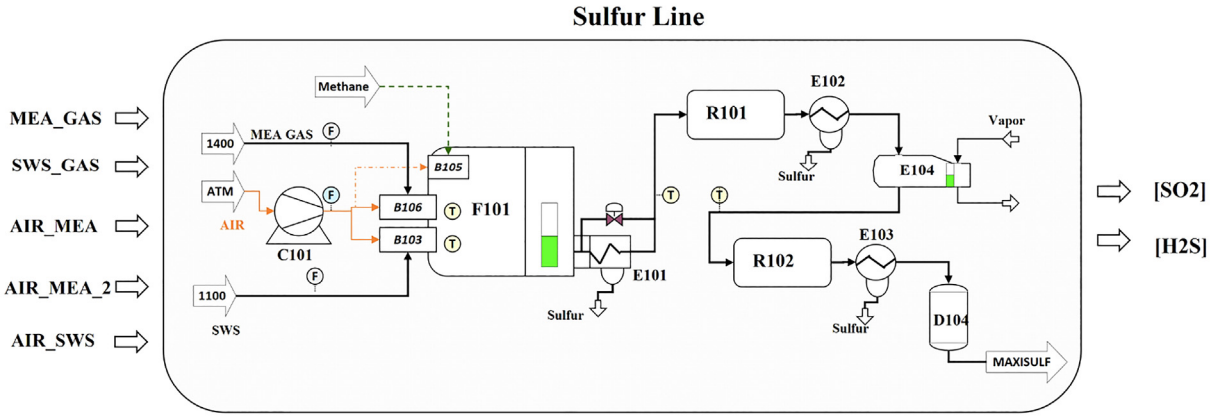

Fig. 4. Schematic description of a sulfur line in which five inputs and two outputs are considered.

The data available for sulfur line 2, chosen by the process experts, consists of 14 400 patterns obtained with a sampling time of one minute. A first analysis was carried out on the data to identify outliers that could invalidate the SS design. A substitution method using interpolation was applied based on the indications from the experts. The obtained dataset is reported in Fig. 5.

The dataset was split into training (70%), validation (15%) and test (15%) sets.

The time dependencies within the process were taken into account, maintaining the provided patterns in the correct sequence.

The normalised root mean square error (NRMSE) was used on the validation set to select the optimal model among those trained for each network. A reasonable model should have an NRMSE between zero and one; the equation is here reported:

$$NMRSE = \sqrt{\frac{1}{\sigma_t^2}\frac{1}{T}\sum_{n=1}^{T}(y(n) - y_t(n))^2} \tag{6}$$

where $y(n)$ is the network output when the $n-th$ input pattern is provided, $y_t(n)$ is the corresponding target value, T is the number of available patterns and σ_t^2 is the target variance. A further check was carried out through the correlation analysis between the estimated and measured output:

$$R = \frac{\sum_{n=1}^{T}(y(n) - \mu_y)(y_t(n) - \mu_{y_t})}{\sqrt{\sum_{n=1}^{T}(y(n) - \mu_y)^2}\sqrt{\sum_{n=1}^{T}(y_t(n) - \mu_{y_t})^2}} \leqslant 1 \tag{7}$$

where μ_i indicates the mean value of the variable $i \in \{y, y_t\}$.

The following simulations were performed in the MATLAB environment using dedicated toolboxes, and the computer configuration was as follows: OS –Windows 10 (64 bit), RAM – 16.0 GB CPU – Core i7-7700HQ (2.8 GHz), and Matlab version – 2020a.

4.2. ESN performance

Following the hyperparameter optimisation strategy described in Section 2.1, each configuration was simulated ten times to determine the best results. The considered hyperparameter searching ranges are shown in Table 1.

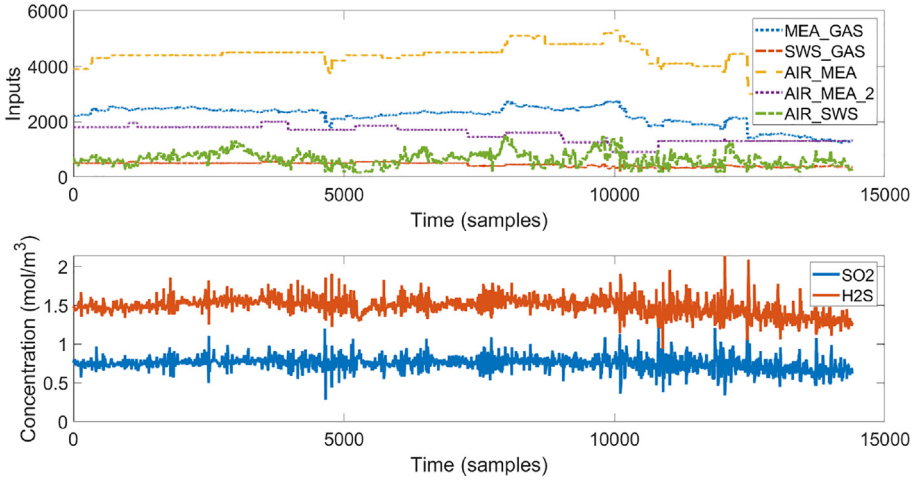

Fig. 5. Inputs and outputs available for sulfur line 2.

Table 1
Searching range for ESN hyperparameters.

Hyperparameter	Range
Spectral radius	[0.5 : 0.05 : 1]
Input scaling	[0.1 : 0.05 : 1.1]
Input shifting	[0 : 0.05 : 1]
Teacher scaling	[0.1 : 0.05 : 1.1]
Teacher shifting	[0 : 0.05 : 1]
Internal neurons	[40 : 5 : 120]

The outcome of the searching strategy corresponds to the optimal ESN configuration, obtained based on the performance on the validation dataset. The first 100 samples in the data sequence were used for the washout phase. The trend of the NMRSE for the optimal configurations during the searching process is reported in Fig. 6 for the H_2S output. The role of each hyperparameter in improving the network performance was different, as shown in Fig. 7, where the performance for the SO_2 output, as a function of the spectral radius, was statistically reported in terms of validation error. The optimal performance was obtained at $\rho = 0.9$. Similarly, the performance obtained during the training and validation phases, as a function of the number of hidden neurons, is reported in Fig. 8 for the H_2S SS. Even if increasing the neuron number improves the network performance during training, the minimum validation error criteria were considered for the hyperparameter selection to avoid overlearning problems.

The analysis of the different models led to the selection of the hyperparameters reported in Table 2.

Table 3 shows the best network performance for SO_2 and H_2S on the training and test data obtained with the optimal hyperparameters previously selected.

As introduced in Section 3, the stoichiometric proportion between the two outputs is a critical aspect to be analysed. The performance obtained, as reported in Table 4, demonstrates that the proposed architecture can obtain a good prediction, particularly when combining the results from the two ESN structures. Finally, based on the requested performance for the SRU, which was constrained by the anti-pollution regulation, the capability of the SSs to estimate the presence of peaks was assessed. A subset of samples that verifies the condition on the 30% over the range was extracted, and the corresponding performance indexes are reported in Table 5. Significant results in terms of estimation error and correlation coefficient were reached, in particular, for the SO_2 output.

4.3. ESN-IP performance

The ESN performance could be improved by adapting the internal weights in the reservoir. In addition to the ESN hyperparameters, the learning rate (η), mean (μ), and variance (σ) of the targeted Gaussian process were included. To find the optimal solution, we adopted the same procedure discussed for the ESNs. A grid-searching procedure on the IP hyperparameters was performed (see Table 6). The pre-training phase was based on the adaptation strategy, as reported in Eqs. 4 and 5.

The optimal values for the ESN-IP hyperparameters are reported in Table 7, whereas Table 8 shows network performances on the training and test data. The results obtained show an improvement for the SO_2 if compared to the ESNs, whereas comparable results were obtained for the H_2S.

The performance obtained for the stoichiometric proportion is reported in Table 9, and the comparison between the network outcome and the target signals are shown in Fig. 9. Finally, the analysis of the peaks is summarised in Table 10.

The results align with the previous cases, with sufficient tail gas composition prediction and accurate peaks prediction of SO_2. An advantage of the ESN-IP strategy is the improvement of the statistical significance of the results. After optimising the hyperparameters, the performances obtained with the ESN-IP approach were more consistent and less related to the initial randomisation, owing to the adaptation of the internal weights, as is demonstrated in Section 5 where the estimation performance and the computational effort for the three considered methods (i.e. ESN, ESN-IP and LSTM) are compared. Therefore, the IP strategy is necessary to limit the number of trained networks needed to find the optimal SS. In fact, even a single trained network, thanks to the internal weights adaptation, can indicate the achievable performance.

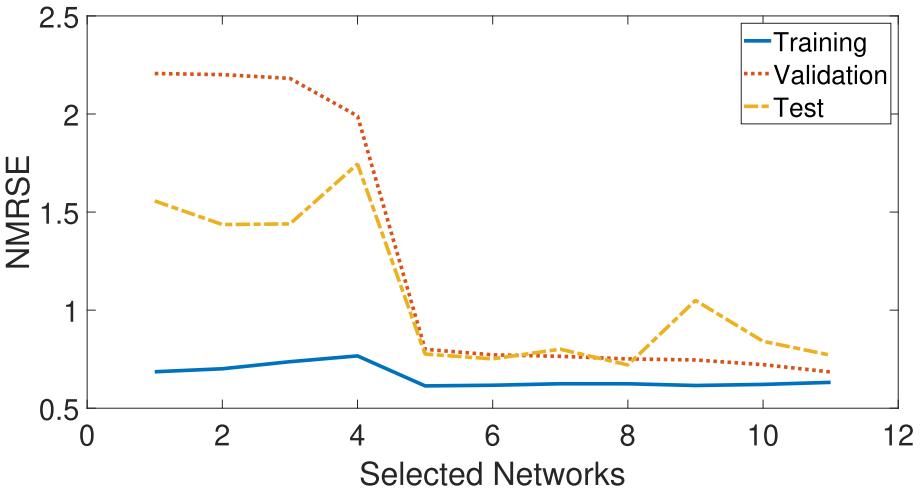

Fig. 6. Trend of the NMRSE error for the training, validation and test patterns. The sequence of networks represents the optimal ESNs, modelling the H_2S dynamics, selected during the hyperparameter optimisation process. The optimal criterium is a minimum validation NMRSE.

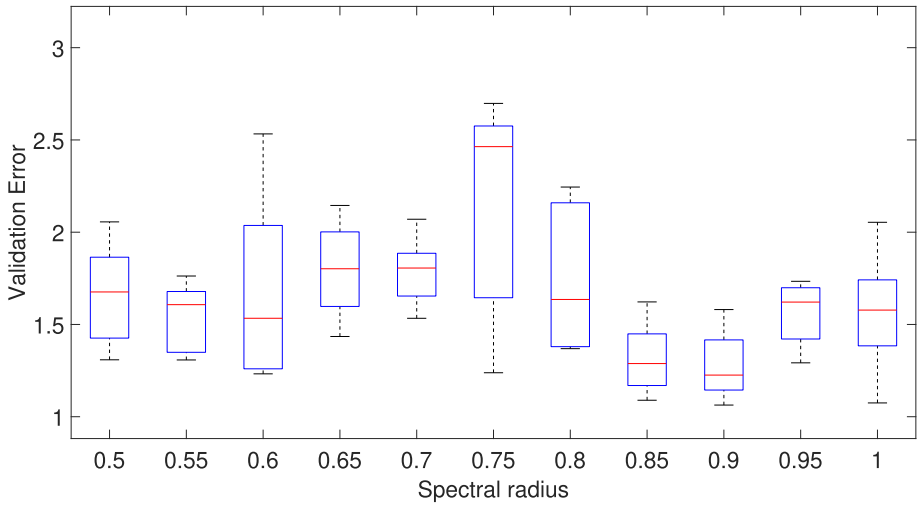

Fig. 7. Performance of the ESN for SO_2 estimation for different values of the spectral radius.

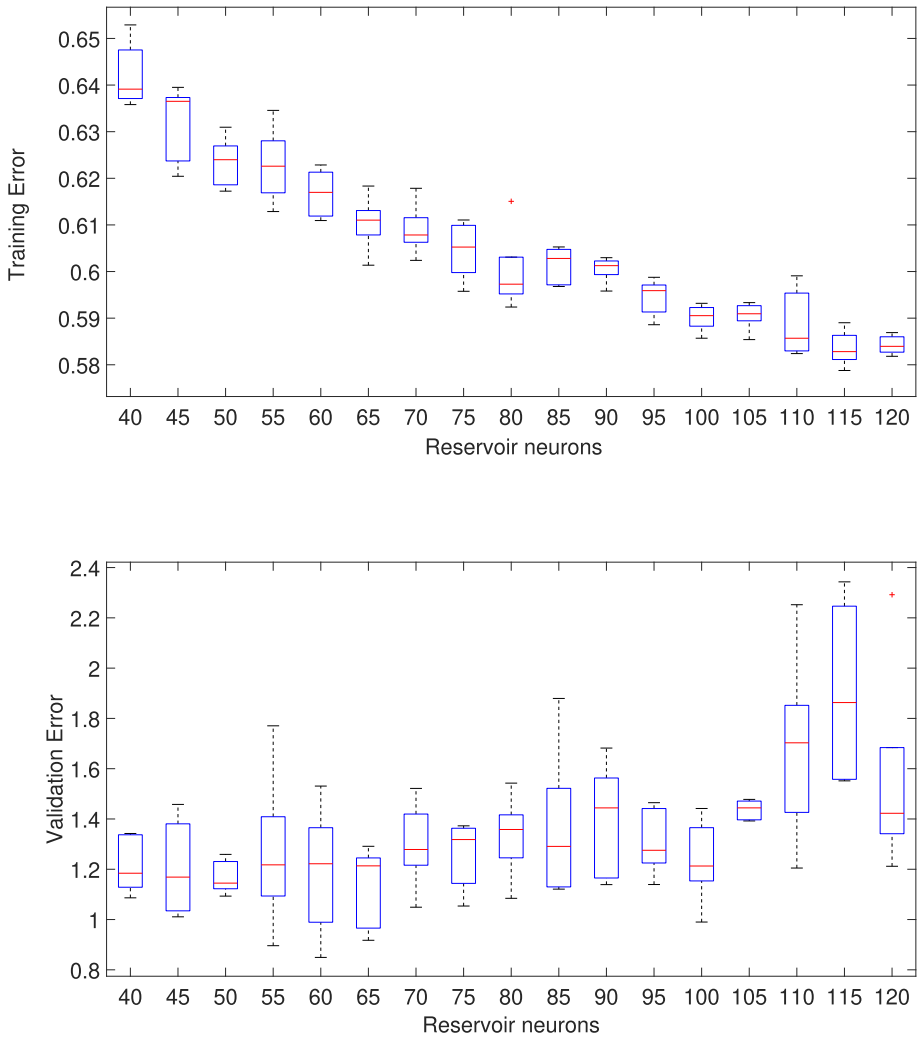

Fig. 8. Trend of the training (top panel) and validation (bottom panel) error when the number of internal neurons of the reservoir is changing in the network estimating H_2S. To avoid overfitting, $n_x = 60$ is selected.

Table 2
ESN network optimal hyperparameters.

	SO_2	H_2S
Structure	5-80-1	5-60-1
Spectral radius	0.90	0.85
Input scaling	0.05	0.05
Input shift	0	0.25
Teacher scaling	0.25	0.8
Teacher shift	0	1

Table 3
Training and test performance for the identified optimal ESN.

	SO_2		H_2S	
	Train	*Test*	*Train*	*Test*
NMRSE	0.57	0.6	0.61	0.62
R	0.81	0.80	0.80	0.79

Table 4
ESN performance on the stoichiometric proportion between the two outputs.

$[H_2S] - 2[SO_2]$	**Train Set**	**Test Set**
NRMSE	0.52	0.51
R	0.85	0.86
Max Err %	12.4	15.8

Table 5
ESN performance on peaks considered when the concentration exceed the 30% of the average value.

Peaks performance		SO_2	H_2S
Train	NRMSE	0.22	0.43
	R	0.98	0.90
Test	NRMSE	0.23	0.51
	R	0.96	0.85

Table 6
ESN-IP additional hyperparameters and corresponding searching range.

Hyperparameter	**Range**
η	$\{10^{-4}, 10^{-5}, 10^{-6}\}$
μ	$[-0.5 : 0.25 : 0.5]$
σ	$[0.1 : 0.1 : 0.5]$

Table 7
Optimal hyperparameters for the ESN-IP network.

	SO_2	H_2S
η	10^{-6}	10^{-6}
μ	-0.25	-0.5
σ	0.4	0.5
Structure	5-40-1	5-70-1
Spectral radius	0.9	0.85
Input scaling	0.05	0.5
Input shift	0.05	0.2
Teacher scaling	0.3	0.3
Teacher shift	0.2	0

Table 8
Training and test performance for the identified optimal ESN-IP.

	SO_2		H_2S	
	Train	*Test*	*Train*	*Test*
NMRSE	0.52	0.57	0.60	0.63
R	0.84	0.82	0.79	0.77

Table 9
ESN-IP performance on the stoichiometric proportion between the two outputs.

$[H_2S] - 2[SO_2]$	**Train Set**	**Test Set**
NRMSE	0.50	0.51
R	0.87	0.86
Max Err %	8.9	12.8

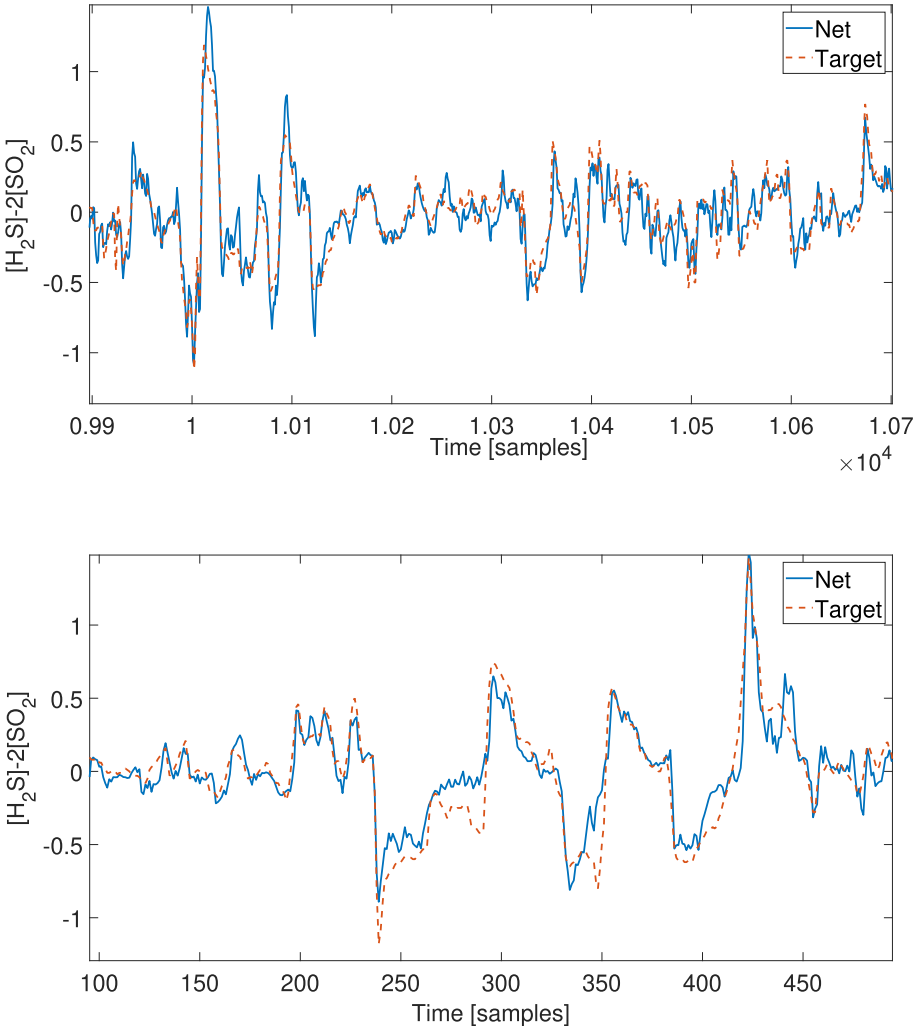

Fig. 9. Comparison between estimated and measured values related to the stoichiometric proportion between the two outputs, obtained with ESN-IP during training (top panel) and test (bottom panel).

Table 10
ESN-IP performance on peaks considered when the concentration exceed the 30% of the average value.

Peak performance		SO_2	H_2S
Train	NRMSE	0.22	0.42
	R	0.97	0.90
Test	NRMSE	0.28	0.54
	R	0.97	0.83

4.4. LSTM performance

The structure adopted for the creation of an LSTM network requires a vector of layers. In all the simulations here reported, in addition to the discussed LSTM levels, a sequence input layer was included with two fully connected layers, one after the input and the other before the output and a dropout layer. Dropout is a technique used to disconnect some neurons during the learning phase. Therefore, each neuron has a probability of missing the training. This mechanism can reduce overfitting, and the adopted dropout probability is 0.5.

Single-layer, two-layer and bidirectional-layer networks were analysed. The bidirectional LSTM layer trains the network both on the input sequence and on an inverted copy of the input sequence, which can provide additional context. The maximum number of learning iterations was set to 250. The same number of neurons was considered for each layer. Ten different networks were simulated for each possible combination of hyperparameters, as reported in Table 11.

Fig. 10 shows how the NMRSE varies on the basis of hidden nodes and layers for the SO_2 output. The best results were obtained with networks composed of a single LSTM layer with a low number of neurons, as reported in Table 12. Fig. 11 shows the variations of the error in relation to the initial scaling of the gate weights. Changing these characteristics did not lead to significant improvements; therefore, they were considered secondary while searching for the best network.

As in the previous cases, the two outputs were studied separately. Table 13 shows network performance for both outputs.

The stoichiometric proportion between the two outputs in the tail gas was computed as expressed before, and results are shown in Fig. 12 and Table 14. The peak analysis is reported in Table 15.

Considerations on the obtained results compared to the ESN performance are summarised after the statistical analysis in the following section.

5. Statistical analysis on the network performance

In addition to exploring the technique and neural architecture to reach the best performance for the selected SS, other relevant characteristics to be evaluated are the statistical significance of the results and the computational effort.

To perform this analysis, we considered, for each architecture, the optimal hyperparameters that were previously identified, simulating 100 different networks. Therefore, we maintained the optimal structure, analysing the effect of the random initialisation of internal weights.

In Table 16, the statistical distribution in terms of average value and the standard deviation is reported for the NMRSE and R coefficient for the testing dataset related to SO_2. Moreover, Fig. 13 shows a comparison between the three techniques in terms of estimation capability and computational effort related to the learning phase of a single network.

The reported statistical analysis shows that ESN and LSTM present a similar distribution for the NMRSE and the correlation coefficient. However, the computational time needed to perform the training phase is significantly different. An average time of roughly 1s is required for ESN training whereas, LSTM needs about 43s for the same task. When IP is included in the ESN process, the computational time is only slightly increased for the ESN. The error distribution presents an average value located on the bottom of the ESN error profile, with a low standard deviation. These considerations indicate the ESN-IP solution is a method that demonstrates high performance and statistical robustness with limited additional computational time. Similar results were obtained for the H_2S output.

Table 11
LSTM hyperparameter searching range.

Hyperparameter	Range
No. Neuron in 1 layer	[20 : 5 : 80]
No. Neuron in 2 layer	[10 : 5 : 40]
No. Neuron in 1 bi-layer	[20 : 5 : 40]
Input gate scaling	[0.1 : 0.1 : 0.8]
Forget gate scaling	[0.1 : 0.1 : 0.6]

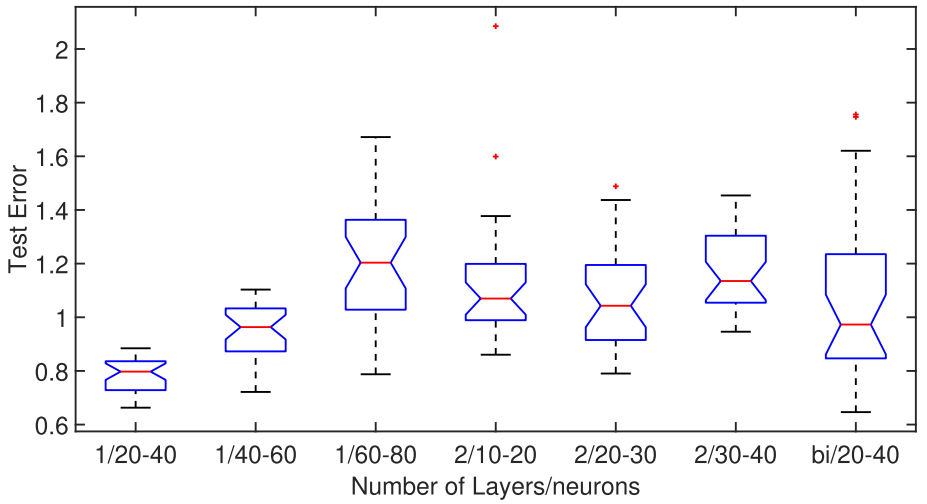

Fig. 10. LSTM test error in function of the number of layers and neurons for the SO_2 output. The last case correspond to a bidirectional layer.

Table 12
LSTM network optimal hyperparameters.

	SO_2	H_2S
Structure	5-40-1	5-35-1
Input scaling	0.5	0.3
Forget scaling	0.6	0.2

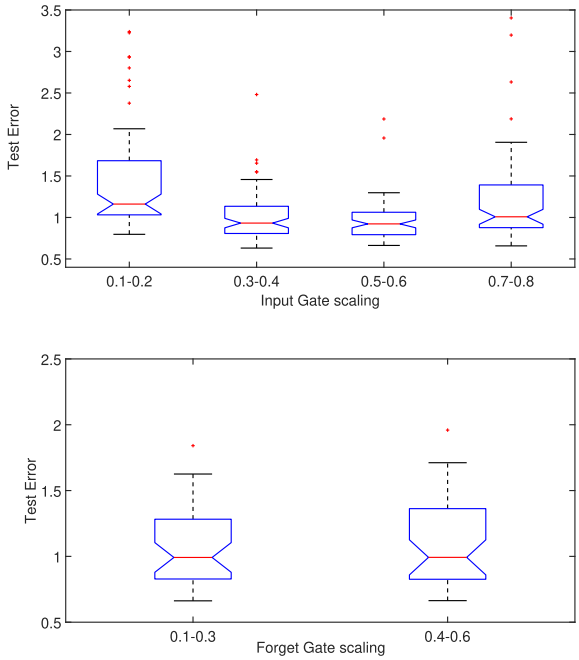

Fig. 11. LSTM test error in function of input (top panel) and forget (bottom panel) weights scaling for the SO_2 output.

Another relevant aspect is the time required for the testing phase, which was considerably reduced. In particular, concerning ESN, we obtained a computational time of $T_{ESN} = 0.285 \pm 0.007$ with almost no difference between the two output variables. For the ESN-IP, the overhead for the internal weights adaptation was present only during the pre-training phase. Therefore, based on the testing, the ESN-IP is computationally equivalent to the standard ESN. For the LSTM, the computa-

Table 13
Training and test performance for the identified optimal LSTM.

	SO_2		H_2S	
	Train	*Test*	*Train*	*Test*
NMRSE	0.55	0.64	0.58	0.65
R	0.83	0.79	0.80	0.76

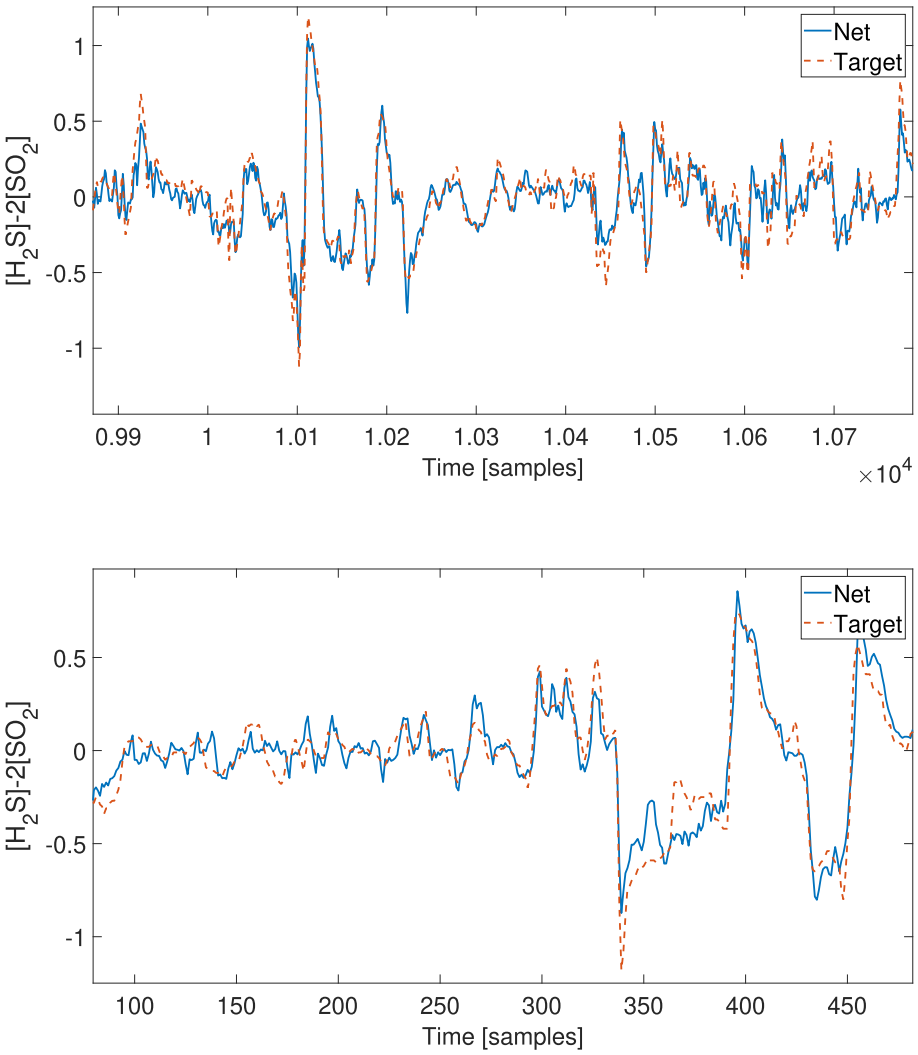

Fig. 12. Comparison between estimated and measured stoichiometric proportion between the two outputs obtained using LSTM: (top panel) training and (bottom panel) test.

Table 14
LSTM performance on the stoichiometric proportion between the two outputs.

$[H_2S] - 2[SO_2]$	**Train set**	**Test set**
NRMSE	0.50	0.61
R	0.86	0.81
Max Err %	8.9	13.9

Table 15
LSTM performance on peaks considered when the concentration exceed the 30% of the avarage value.

Peaks performance		SO_2	H_2S
Train	NRMSE	0.20	0.36
	R	0.98	0.92
Test	NRMSE	0.22	0.58
	R	0.98	0.81

Table 16
Statistical evaluation of NMRSE and R on the testing dataset and computational time, performed over 100 trials, for the different architectures and outputs. The time is related to the learning process of a single network.

	NMRSE		R		Time [s]	
	SO_2	H_2S	SO_2	H_2S	SO_2	H_2S
ESN	1.09 ± 0.26	0.85 ± 0.16	0.54 ± 0.13	0.58 ± 0.04	1.03 ± 0.014	0.97 ± 0.02
ESN-IP	0.66 ± 0.014	0.73 ± 0.02	0.75 ± 0.012	0.66 ± 0.09	1.60 ± 0.02	1.45 ± 0.02
LSTM	1.08 ± 0.18	0.91 ± 0.15	0.56 ± 0.09	0.60 ± 0.08	43.4 ± 0.33	43.89 ± 0.45

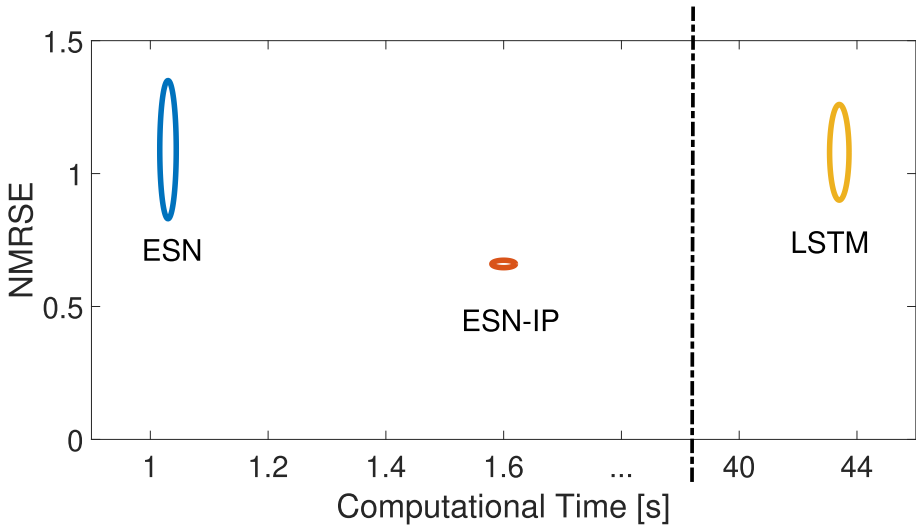

Fig. 13. Comparison between ESN, ESN-IP and LSTM in terms of NMRSE in test and computational time to perform the learning phase of a single network. The centre of each ellipsis represents the average values, whereas the axes indicate the standard deviation. The reported data are related to the SO_2 network as reported in Table 16.

tional time needed for the feedforward phase was $T_{LSTM} = 0.532 \pm 0.011$, with minimal differences between the two output variables. Additionally, the stoichiometric proportion analysis is crucial for the proposed SSs. The reported results show adequate performance for all the proposed architectures. The ESN and ESN-IP achieve slightly better results, confirming the applicability of these methodologies to design data-driven SSs. Moreover, ESN and ESN-IP can obtain similar results in terms of best network performance; however, the optimal structure is more easily determined with the ESN-IP because the internal weight adaptation yields reduced randomness of the initialisation process, concentrating the performance distribution near to the optimal configuration.

6. SS transferability – sulfur line 4 models

To evaluate the capabilities of the proposed architectures in comparison with other techniques discussed in the literature, a different dataset was considered. As mentioned in Section 3, the SRU process has been applied to several studies in the literature regarding the data like that acquired from sulfur line 4 [12]. In addition to a results comparison with the literature, the knowledge obtained from sulfur line 2 can be applied to the line 4 dataset , following a procedure commonly adopted for transfer learning techniques.

Even if the same industrial process as the one examined thus far is considered, the two sulfur lines of the plant behave differently because of the varied working conditions. We initially applied the optimal hyperparameters gained from the dataset related to sulfur line 2. Therefore, following the second part of the Algorithm 1, we performed the statistical analysis; the obtained results are reported in Table 17.

With this new learning process, all networks performed acceptably as compared to the results obtained on sulfur line 2. We then verified the improvement when the hyperparameters were slightly modified, performing a fine-tuning in a reduced searching space.

The obtained optimal hyperparameters are reported in Table 18. The performances of the new models are summarised in Table 19, where the improvement in NMRSE ($\Delta\%$) is also indicated.

The fine-tuning of the hyperparameters results in an improvement by 10% to 20% in almost all the networks.

To further highlight the novelty of our approach, an analysis of the obtained performance and key advantages compared with the techniques available in the literature and applied to the same industrial process was conducted. This analysis is related to the SS estimation performance for the SRU process, as summarised in Table 20. In a recent work, Hikosaka et al. proposed a strategy for selecting appropriate variables input regressors to improve the performance using a PLSR method [14]. The adopted strategy consists of an ensemble learning technique incorporated within a genetic algorithm-based process variables and dynamics selection (EGAVDS). The results obtained for the SRU process, as shown in Table 20, are comparable with the finite impulse response (FIR) approach presented in [12] and the radial basis function-based strategy discussed in [7], showing significant improvements over the standard PLSR method. Our proposed ESN-IP approach outperforms the EGAVDS method both in terms of RMSE and R indexes and further reduces the complexity in the regressor selection. In [18], a deep LSTM was proposed to model the sulfur line 4 data. The structure included two hidden layers and a total of 50 neurons, therefore comparable with the LSTM structures designed in our work. The computational effort required during the learning phase was optimised by taking advantage of GPU computing and resulted in a range between 10 and 60s. These results confirm our statistical analysis, as shown in Fig. 13, where the LSTM training time is around 43 s, which is 40 times longer than the time needed to train an ESN for the same task. The estimation performance obtained in [18] with the deep LSTM is slightly improved over the standard LSTM network, which we developed for comparison. However, the ESN-IP structure still outperformed these results, as reported in Table 20. The application of multi-state–dependent parameter (MSDP) models, as discussed in [4], is particularly effective in minimising the number of variables involved. In fact, the models for each quality variable consist of two output regressors, and the parameters are estimated as functions of the two most relevant input variables. This solution can achieve high performance in one-step-ahead prediction, but the accuracy of the architecture on infinite step prediction is not apparent. On the contrary, the proposed ESN-based architecture realises a dynamic model able to estimate the output variables in time, suitably handling the memory information needed during the process analysis. Another methodology able to obtain high prediction performance is the just-in-time (JIT) learning method [50], an adaptive approach in which the model structure and parameters are updated in time to best match the specific working condition of the plant. SS modelling by JIT techniques is time-consuming and requires extensive memory space to store the different models. Therefore, the improved performance is obtained at the cost of greater complexity in the model structure and a time-consuming optimisation procedure. These aspects are in conflict with the needs of online applications in the industrial field. Finally, the SS models proposed in [5] are based on ESN and, therefore, can be directly compared with our architecture. To improve the flexibility in a traditional ESN, especially for modelling long-term dependent SSs, Bo et al. proposed a deep echo state network (DESN), composed of a series of sub-reservoirs connected in sequence. The inclusion of time delay modules to link adjacent layers (ADESN) was also investigated. The results obtained in the case of DESN and ADESN are almost identical, demonstrating that the long-term memory components are not relevant for the considered application. Comparing the results obtained in [5] with the ESN-IP solution that we are proposing, the prediction results are nearly equivalent in terms of error and correlation coefficient. However, the optimised ESN-IP structure contains fewer reservoir neurons, one-third of the 150 in the ADESN-based solution. This quantity has a direct impact on the computational effort required during the training process. Another relevant advantage acquired with the application of the IP technique is the consistency in obtaining optimal results. This added value facilitates the learning process, reducing the number of trials needed to retrieve the optimal network configuration.

Table 17
Performances obtained using the dataset for sulfur line 4 with the hyperparameters identified for sulfur line 2.

Test performance		LSTM	ESN	ESN IP
	Structure	5-40-1	5-80-1	5-40-1
SO_2	*NRMSE*	0.64	0.45	0.40
	R	0.74	0.88	0.90
	Structure	5-35-1	5-70-1	5-70-1
H_2S	*NRMSE*	0.72	0.61	0.58
	R	0.71	0.84	0.87

Table 18
Hyperparameters optimised for sulfur line 4 (only those different from sulfur line 2 analysis are reported.)

	Structure		Other parameters	
	SO_2	H_2S	SO_2	H_2S
LSTM	5-50-1	5-40-1	input scaling = 0.6	input scaling = 0.7 forget scaling = 0.3
ESN	5-85-1	5-70-1	$\rho = 0.98$ input scaling = 0.01 teacher scaling = 0.26	$\rho = 0.95$ input shift = 0.05 teacher scaling = 0.85
ESN-IP	5-45-1	5-65-1	$\rho = 1$ teacher scaling = 0.25 teacher shift = 0	$\rho = 0.95$ input shift = 0.05 teacher scaling = 0.50 teacher shift = 0.05

Table 19
Performances obtained using the dataset for sulfur line 4 after a fine-tuning of the hyperparameters. $\Delta\%$ represents the percentage of improvement in terms of NMRSE.

Test performance		LSTM	ESN	ESN IP
	Structure	5-50-1	5-85-1	5-45-1
SO_2	$\Delta\%$	19%	18%	17%
	NRMSE	0.52	0.437	0.33
	R	0.85	0.93	0.95
	Structure	5-40-1	5-70-1	5-65-1
H_2S	$\Delta\%$	23%	12%	10%
	NRMSE	0.65	0.54	0.52
	R	0.84	0.89	0.91

Table 20
Review of the SS performance available in the literature for the line 4 SRU process, obtained with different architectures.

		SO_2		H_2S	
		RMSE	R	RMSE	R
	LSTM	0.028	0.85	0.030	0.84
	ESN	0.020	0.93	0.025	0.89
	ESN-IP	0.018	0.95	0.024	0.91
[12]	MLP	0.063	0.9	0.094	0.84
	RBF	0.038	0.76	0.042	0.72
	ANFIS	0.089	0.86	0.034	0.81
[7]	FIR - LS	0.022	0.90	0.031	0.78
	FIR - SKRLS	0.021	0.91	0.032	0.78
[18]	RNN	0.028	-	0.031	-
	Deep LSTM	0.020	-	0.026	-
[14]	PLS	-	-	0.041	0.5
	EGAVDS	-	-	0.031	0.76
[15]	MLR	0.070	-	0.069	-
	SVR	0.068	-	0.067	-
	SVR - LS	0.123	-	0.078	-
	PCR	0.051	-	0.060	-
[47]	SNN	0.034	0.81	-	-
	MNN	0.030	0.86	-	-
	SAE - LS	0.029	0.87	-	-
	SIAE	0.027	0.88	-	-
[4]	SDP	0.014	0.97	0.014	0.97
	MSDP	0.012	0.97	0.012	0.98
[50]	JIT	0.015	0.98	0.015	0.93
	CoJIT	0.017	0.98	0.016	0.93
[5]	DESN	0.017	0.95	0.026	0.91
	ADESN	0.017	0.95	0.025	0.93

7. Conclusions

In this research, the application of ESN for designing SSs was analysed and compared with other neural structures. Reservoir-based architectures for SS dynamical modelling are vital when the selection of the input and output regressors is either difficult or the system order is too large. In fact, the characteristic structure of the ESN allows for the system dynamics to be embedded in the combination of spatiotemporal patterns of activity arising within the reservoir layer. A simple, linear read-out map can be applied to enslave this dynamical activity to estimate a given quality variable. The proposed methodology was applied to an SRU process of a refinery in Italy, showing that SSs can replace the hardware sensors during maintenance, allowing feedback control. The developed neural architectures were adopted to estimate the sulfur concentration in the tail gas in two different sulfur lines, verifying the possibility to transfer the knowledge acquired from one process to the other. The improvement obtained through a fine-tuning of the hyperparameters was also assessed.

The novelty of our work can be summarised as follows. ESNs can be adopted as a reference architecture for SS modelling, having advantages over other deep learning recurrent solutions, like LSTM. The introduction of the IP rule can statistically improve the network performance by adapting the reservoir dynamics to the input data. The results show that the ESN-IP represents a sufficient compromise between computational cost and model performance. Differently from other works related to the SRU, we identified relevant indexes (i.e. peaks and stoichiometric analysis) that are fundamental for the real application of the designed models, obtaining accurate results in terms of estimation error and correlation coefficient. The current limitation of the proposed solution is the variability of the network performance due to the large number of weights and hyperparameters to be initialised and properly tuned. To overcome this issue, the IP technique can be used to guide the internal weight initialisation. Further work can be dedicated to the hyperparameter optimisation procedure, where more sophisticated searching strategies can be assessed. Another aspect that will be investigated is the interpretability of the results. In fact, with a reservoir-based approach, useful information on the behaviour within the model is currently difficult to extract. Therefore, one challenge to be addressed is the combination of the high-performance ESN-based architectures with a better understanding of the process characteristics from the learned structures.

CRediT authorship contribution statement

Luca Patanè: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Maria Gabriella Xibilia:** Conceptualization, Methodology, Software, Data curation, Writing - original draft.

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.

References

[1] B. Andò, S. Graziani, M.G. Xibilia, Low-order Nonlinear Finite-Impulse Response Soft Sensors for Ionic Electroactive Actuators Based on Deep Learning, IEEE Trans. Instrum. Meas. 68 (5) (2019) 1637–1646.
[2] P. Arena, L. PatanèA, Spinosa, Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised Liquid State classifiers, Inf. Sci. 478 (2019) 28–39.
[3] P. Arena, L. Patanè, A. Spinosa, Robust modelling of binary decisions in Laplacian Eigenmaps-based Echo State Networks, Engineering Applications of Artificial Intelligence 95 (2020) 103828, ISSN 0952–1976..
[4] B. Bidar, F. Shahraki, J. Sadeghi, M.M. Khalilipour, Soft Sensor Modeling Based on Multi-State-Dependent Parameter Models and Application for Quality Monitoring in Industrial Sulfur Recovery Process, IEEE Sens. J. 18 (11) (2018) 4583–4591.
[5] Y.-C. Bo, P. Wang, X. Zhang, B. Liu, Modeling data-driven sensor with a novel deep echo state network, Chemometrics Intell. Lab. Syst. (2020) 104062.
[6] H. Bunke, T. Varga, Digital Document Processing: Major Directions and Recent Advances, chap. Off-Line Roman Cursive Handwriting Recognition, Springer London, London, 2007, pp. 165–183.
[7] X. Chen, Z. Mao, R. Jia, S. Zhang, Ensemble regularized local finite impulse response models and soft sensor application in nonlinear dynamic industrial processes, Applied Soft Computing 85 (2019) 105806, ISSN 1568–4946..
[8] N. Chouikhi, B. Ammar, N. Rokbani, A.M. Alimi, PSO-based analysis of Echo State Network parameters for time series forecasting, Appl. Soft Comput. 55 (2017) 211–225.
[9] F. Curreri, S. Graziani, M.G. Xibilia, Input selection methods for data-driven Soft sensors design: Application to an industrial process, Inf. Sci. 537 (2020) 1–17, ISSN 0020-0255.
[10] S. Dasgupta, D. Goldschmidt, F. Wörgotter, P. Manoonpong, Distributed recurrent neural forward models with synaptic adaptation and CPG-based control for complex behaviors of walking robots, Front. Neurorobotics 9 (2015) 10.
[11] A. Di Bella, S. Graziani, G. Napoli, M.G. Xibilia, Selection of regressors using correlation analysis to design a Virtual Instrument for an SRU of a refinery, in: 2007 Mediterranean Conference on Control Automation, 1–6, 2007..
[12] L. Fortuna, S. Graziani, A. Rizzo, M. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, first edition., Springer-Verlag, London, 2007.
[13] Y.-L. He, Y. Tian, Y. Xu, Q.-X. Zhu, Novel soft sensor development using echo state network integrated with singular value decomposition: Application to complex chemical processes, Chemometrics Intell. Lab. Syst. (2020) 200, 103981, ISSN 0169-7439.
[14] T. Hikosaka, S. Aoshima, T. Miyao, K. Funatsu, Soft Sensor Modeling for Identifying Significant Process Variables with Time Delays, Ind. Eng. Chem. Res. 59 (26) (2020) 12156–12163.
[15] V. Jain, P. Kishore, R.A. Kumar, A.K. Pani, Inferential Sensing of Output Quality in Petroleum Refinery Using Principal Component Regression and Support Vector Regression, in: 2017 IEEE 7th International Advance Computing Conference (IACC), 2017, pp. 461–465.
[16] Q. Jiang, S. Yan, H. Cheng, X. Yan, Local-Global Modeling and Distributed Computing Framework for Nonlinear Plant-Wide Process Monitoring With Industrial Big Data, IEEE Trans. Neural Networks Learn. Syst. (2020) 1–11.

[17] Q. Jiang, X. Yan, H. Yi, F. Gao, Data-Driven Batch-End Quality Modeling and Monitoring Based on Optimized Sparse Partial Least Squares, IEEE Trans. Industr. Electron. 67 (5) (2020) 4098–4107.
[18] W. Ke, D. Huang, F. Yang, Y. Jiang, Soft sensor development and applications based on LSTM in deep neural networks, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–6.
[19] S.E. Lacy, S.L. Smith, M.A. Lones, Using echo state networks for classification: A case study in Parkinson's disease diagnosis, Artif. Intell. Med. 86 (2018) 53–59.
[20] M. Lukosevicius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Computer Sci. Rev. 3 (3) (2009) 127–149, ISSN 1574–0137..
[21] A. Morey, S. Pradhan, R.A. Kumar, A.K. Pani, V. Vijayan, V. Jain, A. Gupta, Pollutant monitoring in tail gas of sulfur recovery unit with statistical and soft computing models, Chem. Eng. Commun. 206 (1) (2019) 69–85.
[22] R. Parvizi Moghadam, F. Shahraki, J. Sadeghi, Online Monitoring for Industrial Processes Quality Control Using Time Varying Parameter Model, International Journal of Engineering 31 (4) (2018) 524–532, ISSN 1025–2495..
[23] B. Schrauwen, M. Wardermann, D. Verstraeten, J.J. Steil, D. Stroobandt, Improving reservoirs using intrinsic plasticity, Neurocomputing 71 (7) (2008a) 1159–1171, ISSN 0925–2312..
[24] C. Shang, F. Yang, D. Huang, W. Lyu, Data-driven soft sensor development based on deep learning technique, Journal of Process Control 24 (3) (2014) 223–233, ISSN 0959–1524..
[25] A. Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, vol. 404, Elsevier BV, 2020.
[26] F. Souza, R. Araújo, Online Mixture of Univariate Linear Regression Models for Adaptive Soft Sensors, IEEE Trans. Industr. Inf. 10 (2) (2014) 937–945.
[27] F.A. Souza, R. Araújo, J. Mendes, Review of soft sensor methods for regression applications, Chemometrics Intell. Lab. Syst. 152 (2016) 69–79.
[28] J.J. Steil, Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning, Neural Networks 20 (3) (2007) 353–364, ISSN 0893–6080..
[29] Z. Tian, Echo state network based on improved fruit fly optimization algorithm for chaotic time series prediction, Ambient Intell. Human Comput. (2020).
[30] Z. Tian, G. Wang, Y. Ren, Short-term wind speed forecasting based on autoregressive moving average with echo state network compensation, Wind Eng. 44 (2) (2020) 152–167.
[31] J. Triesch, A Gradient Rule for the Plasticity of a Neuron Intrinsic Excitability, Artificial Neural Networks: Biological Inspirations abs/1811.10892 (2005) 65–70..
[32] K. Wang, C. Shang, F. Yang, Y. Jiang, D. Huang, Automatic hyper-parameter tuning for soft sensor modeling based on dynamic deep neural network, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 989–994, 2017..
[33] X. Wang, Y. Jin, K. Hao, Echo state networks regulated by local intrinsic plasticity rules for regression, Neurocomputing 351 (2019a) 111–122, ISSN 0925-2312..
[34] Y. Wang, Z. Pan, X. Yuan, C. Yang, W. Gui, A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network, ISA Transactions 96 (2020) 457–467, ISSN 0019-0578..
[35] Z. Wang, Y.-R. Zeng, S. Wang, L. Wang, Optimizing echo state network with backtracking search optimization algorithm for time series forecasting, Engineering Applications of Artificial Intelligence 81 (2019b) 117–132, ISSN 0952-1976..
[36] Wei Zhang, Yanjun Li, Weili Xiong, Xu. Baoguo, Adaptive soft sensor for online prediction based on enhanced moving window GPR, in: 2015 International Conference on Control, Automation and Information Sciences (ICCAIS), 2015, pp. 291–296.
[37] M. Xibilia, S. Graziani, Deep Learning for Soft Sensor Design, vol. 867, chap. 2, Springer, Cham., 31–59, 2020..
[38] M.G. Xibilia, M. Latino, Z. Marinković, A. Atanasković, N. Donato, Soft Sensors Based on Deep Neural Networks for Applications in Security and Safety, IEEE Trans. Instrum. Meas. 69 (10) (2020) 7869–7876.
[39] X. Yan, J. Wang, Q. Jiang, Deep relevant representation learning for soft sensing, Inf. Sci. 514 (2020) 263–274, ISSN 0020-0255.
[40] L. Yao, X. Jiang, G. Huang, J. Qian, B. Shen, L. Xu, Z. Ge, Virtual Sensing F-CaO Content of Cement Clinker Based on Incremental Deep Dynamic Features Extracting and Transferring Model, IEEE Trans. Instrum. Meas. (2020) 1–10.
[41] W. Yao, Z. Zeng, C. Lian, Generating probabilistic predictions using mean-variance estimation and echo state network, Neurocomputing 219 (2017) 536–547.
[42] X. Yao, Z. Wang, H. Zhang, Prediction and identification of discrete-time dynamic nonlinear systems based on adaptive echo state network, Neural Networks 113 (2019) 11–19.
[43] X. Yuan, L. Li, Y. Shardt, Y. Wang, C. Yang, Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development, IEEE Trans. Industr. Electron. (2020) 1–11.
[44] X. Yuan, L. Li, Y. Wang, Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network, IEEE Trans. Industr. Inf. 16 (5) (2020) 3168–3176.
[45] X. Yuan, C. Ou, Y. Wang, C. Yang, W. Gui, A Layer-Wise Data Augmentation Strategy for Deep Learning Networks and Its Soft Sensor Application in an Industrial Hydrocracking Process, IEEE Trans. Neural Networks Learn. Syst. (2019) 1–10.
[46] X. Yuan, S. Qi, Y. Wang, Stacked Enhanced Auto-encoder for Data-driven Soft Sensing of Quality Variable, IEEE Trans. Instrum. Meas. (2020) 1–10.
[47] X. Yuan, Y. Wang, C. Yang, W. Gui, Stacked isomorphic autoencoder based soft analyzer and its application to sulfur recovery unit, Information Sci. 534 (11) (2020d) 72–84, ISSN 0020-0255..
[48] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, W. Gui, Hierarchical Quality-Relevant Feature Representation for Soft Sensor Modeling: A Novel Deep Learning Strategy, IEEE Trans. Industr. Inf. 16 (6) (2020) 3721–3730.
[49] M.-H. Yusoff, J. Chrol-Cannon, Y. Jin, Modeling neural plasticity in echo state networks for classification and regression, Information Sciences 364–365 (2016) 184–196, ISSN 0020-0255.
[50] J. Zeng, L. Xie, C. Gao, J. Sha, Soft sensor development using non-Gaussian Just-In-Time modeling, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, 5868–5873, 2011..