Abstract—This article proposes a new data-driven model devised to integrate process knowledge into its structure to increase the human–machine synergy in the process industry. The proposed contextual mixture of experts (cMoE) explicitly uses process knowledge along the model learning stage to mold the historical data to represent operators’ context related to the process through possibility distributions. This model was evaluated in two real case studies for quality prediction, including a sulfur recovery unit and a polymerization process. The cMoEs was employed to represent different contexts in both experiments. The results indicate that integrating process knowledge has increased the predictive performance while improving interpretability by providing insights into the variables affecting the process’s different regimes.

Index Terms—Mixture of experts (MoEs), multimode processes, process knowledge, possibility distribution, soft sensors.

I. INTRODUCTION

There is an increasing demand for industrial digitization toward a more sustainable and greener industrial future. Artificial intelligence (AI) is at the forefront of the fourth industrial revolution by redefining decision-making at the operational and technical levels, allowing faster, data-driven, and automatic decision-making along the value chain. Also, with further growth in industrial data infrastructure, many companies are implementing data-driven predictive models to improve energy efficiencies and industrial sustainability. This can reduce production costs and environmental impact while increasing process efficiency. In parallel, the new upcoming industrial revolution (Industry 5.0) aims to leverage human knowledge and decision-making abilities by strengthening the cooperation between humans and machines [1]. This new revolution will require the data-driven models to be explainable by providing insights into the process to gain the operator’s trust and increase synergy. The human–machine synergy can be further enhanced by incorporating operator domain knowledge and process information into the data-driven models.

Process information can come from various sources, including first-principle equations and process-specific characteristics [2], such as process division structure or multiple operating modes. First-principle models can be combined with data-driven models within the hybrid AI models framework [3] or within the informed machine learning framework [4]. For the process division structure, the multiblock modeling is a common approach for retaining the explainability of multistage processes within a multiblock representation [2]. Multiple operating modes can be caused by a change in feedstock, operation, seasonality (aka. multimode processes), or the sequence of phases comprising a batch cycle production (aka. multiphase processes) [5]. These modes can be represented in a multimodal (ensemble) structure [6]. This work focuses on modeling processes with multiple operating modes, and the proposed method was created with this in mind. However, the proposed method is flexible enough to be applied to other types of processes where process expert knowledge is available.

The modeling of multiple operating modes processes follows from rule-based expert systems [7], [8], [9], [10], clustering [11], mixture models (MM) [12], [13], [14], Gaussian mixture regression (GMR) [15], [16], or mixture of experts (MoE) [6], [17] strategies to identify the groups that represent each operational regime, then combine them according to the process’s regime. Apart from rule-based expert models, none of the abovementioned works discuss using domain knowledge from operators. In practice, such methods do not attempt to represent process characteristics; rather, the goal is to minimize prediction error, and in some cases, domain knowledge is used only to initialize the model structure. As a result, despite being accurate, the model becomes unrepresentative of the process, making it impossible to understand the effects of the variables in the various regimes of the process. In fact, the rich process context available from operators can be valuable to the model. If a model can reflect the operator’s knowledge, it can play an important role in model acceptance in the industry.

This article proposes the contextual mixture of experts (cMoE), a new data-driven model that connects the process’ expert domain knowledge (process context) to the predictive model. The cMoE gives a holistic perspective to the data-driven
model while still adhering to process correlation and representing the operators’ process context. The cMoE is composed of a set of expert and gate models, where each expert/gate is designed to represent a specific context of the process employing possibility distributions. The gates represent operators’ process context in the model by defining the boundaries of each contextual region. These three components, experts and gates, and the operator’s contextual information form the basis of cMoE. The training procedure in cMoE uses a learning approach devised to assure that each expert represents the defined context and, at the same time, generalizes well for unseen data. To allow model interpretability, the gates and expert models are linear models. Also, a $\ell_1$ regularization penalty is applied to the gates and experts learning for a parsimonious representation of each context model.

The application of regularization in MoE, with linear base models is not new, and it has been used for dealing with high-dimensional settings and for feature selection. For example, Khalili [18] investigates the $\ell_1$ penalty and smooth clipped absolute deviation [19] penalties for feature selection in MoE. In addition, Peralta and Soto [20] proposed using the $\ell_1$ penalty for MoE in classification applications. The authors in [21] and [22] investigate the theoretical aspect of MoE with $\ell_1$ penalties. In order to avoid instability in the learning of gating coefficients (typically followed by a softmax function), the authors in [23] proposed the use of a proximal-Newton expectation–maximization (EM). The elastic-net penalty was used by the authors of [6], who employed a regularization penalty for the inverse of Hessian matrix along the gates learning. In the proposed eMoE, the $\ell_1$ penalty is employed to promote sparsity in the gates and experts. The solution for the gates and experts follows from the coordinate gradient descent (CGD) together with the EM framework, as used in [6]. However, there are two significant differences between the method presented here and the work of [6]. The gates and experts models are chosen based on an estimator of the leave-one-out cross-validation (LOOCV) error. The cMoE’s performance is accessed based on an estimated LOOCV, and this is used in the EM algorithm as a stop condition and for model selection. Unlike [6], where regularized Hessian can lead to unstable results, the learning of the gates in the cMoE is based on the Newton update, with a step-size parameter added to control the learning rate and increase model stability.

The proposed method is evaluated in two experiments. The first one is the sulfur recovery unit (SRU) described in [24], where the goal is to predict the H2S at the SRU’s output stream. The operators in that study are more interested in the H2S peaks as they are related to the undesirable behavior of SRU unit. The cMoE is then applied for predicting H2S with separate representations for peaks and nonpeaks components, allowing the identification of causes of the peaks, beyond the prediction of H2S. The second case study investigates the application of cMoE to estimate the acid number in a multiphase batch polymerization process. The process knowledge is presented in an annotated data source for the phase transition. The cMoE is then utilized to provide a contextual model for each phase. In all the experiments, the results indicate that incorporating the operator’s context into the cMoE gives interpretability and insight to the process and significantly increases the model performance.

The main contributions of this work are as follows:
1) the use of possibility distributions to represent the operator’s expert knowledge;
2) the development of a new MM called cMoEs to incorporate the operator’s expert knowledge from the possibility distributions into the model structure;
3) the application of a $\ell_1$ penalty to the gates and experts coefficients to promote sparse solutions;
4) the development of a leave-one-out error (LOOCV) estimator for experts and gates and the cMoE model.

The rest of this article is organized as follows. Section II gives the background for this article. Section III presents the proposed model cMoEs. Section IV presents experimental results. Finally, Section VI concludes this article.

## II. PRELIMINARIES

### A. Notation

In this article, finite random variables are represented by capital letters and their values by the corresponding lowercase letters, e.g., random variable $A$, and corresponding value $a$. Matrices and vectors are represented by boldface capital letters, e.g., $A = [a_{ij}]_{N \times d}$ and boldface lowercase letters, e.g., $a = [a_1, \ldots, a_d]^T$, respectively. The input and output/target variables are defined as $X = \{X_1, \ldots, X_d\}$ and $Y$, respectively. The variables $X_1, \ldots, X_d$ can take $N$ different values as $\{x_{ij} \in X_j : j = 1, \ldots, d\}$ and $\{y_i \in Y : i = 1, \ldots, N\}$, and similarly for $Y$ as $\{y_i \in Y : i = 1, \ldots, N\}$.

### B. Mixture of Experts

The MoE is a modeling framework based on the divide and conquer principle. It consists of a set of experts and gates, with the gates defining the boundaries (soft boundaries) and the experts making predictions within the region assigned by the gates. The prediction output of an MoE with $C$ experts is

$$\hat{y}(x_i) = \sum_{c=1}^{C} g_c(x_i) \hat{y}_c(x_i) \tag{1}$$

where $\hat{y}_c(x_i)$ is the expert’s predicted output at region $c$, and $g_c(x_i)$ is the gate function that represents the expert’s boundaries at region $c$.

The probability distribution function (PDF) of the MoE is defined as

$$p(y_i | x_i ; \Omega) = \sum_{c=1}^{C} g_c(x_i) \sigma_c^2 p(y_i | x_i ; \Theta_c). \tag{2}$$

The PDF is the expert $c$ conditional distribution with mean $\hat{y}_c(x_i)$ and $\sigma_c^2$ is the noise variance. The set of expert parameters is defined as $\Theta_c = \{\theta_c, \sigma_c^2\}$. The gates $g_c(x_i; \sigma_c)$ assigns mixture proportions to the experts, with constraints $\sum_{c=1}^{C} g_c(x_i) = 1$ and $0 \leq g_c(x_i) \leq 1$, and for simplicity $g_ci = g_c(x_i)$. The gates
typically follows from the softmax function:
\[
g_{ci} = \exp \left( \mathbf{x}_i^T \mathbf{v}_c \right) / \sum_{k=1}^{C} \exp \left( \mathbf{x}_i^T \mathbf{v}_k \right) \tag{3}
\]
where \( \mathbf{v}_c \) is the parameter that governs the gate \( c \), and \( \mathcal{V} = \{ \mathbf{v}_1, \ldots, \mathbf{v}_C \} \) is the set of all gates parameters.

The collection of all parameters is defined as \( \Theta = \{ \Theta_1, \ldots, \Theta_C, \mathbf{v}_1, \ldots, \mathbf{v}_C \} \).

From the MoE framework, the parameters in \( \Omega \) are found trough the maximization of log-likelihood
\[
\Omega = \arg \max_{\Omega^*} L(\Omega^*) \tag{4}
\]
where the log-likelihood for \( N \) independent identically distributed samples is defined as \( L(\Omega) = \log p(Y | X; \Omega) = \sum_{i=1}^{N} \log p(y_i | x_i; \Omega) \). The solution of (4) follows from the EM algorithm, an iterative procedure that maximizes the log-likelihood from successive steps. In EM, the \( p(Y | X; \Omega) \) is treated as the incomplete data distribution. The missing part, the \( Q \)-function.

\[
p(y_i, z_i | x_i; \Omega) = \prod_{c=1}^{C} g_c(x_i; \mathcal{V})^{z_{ci}} p(y_i | x_i; \Theta_c)^{z_{ci}} \tag{5}
\]
where \( z_{ci} \in \{0, 1\} \) and for each sample \( i \), all variables \( z_{ci} \) are zero, except for a single one. The hidden variable \( z_{ci} \) indicates which expert \( c \) was responsible of generating the sample \( i \). The complete distribution is given by
\[
Q^t(\Omega) = \mathbb{E} \left[ \hat{L}_C(\Omega) \mid \mathcal{Z}, \hat{\Omega}^t \right] = \sum_{i=1}^{N} \sum_{c=1}^{C} \gamma_{ci} \log \left[ g_c(x_i; \mathcal{V}) p(y_i | x_i; \Theta_c) \right] \tag{6}
\]
where \( \gamma_{ci} = \mathbb{E} [z_{ci} | \mathcal{Z}, \hat{\Omega}^t] \) is the posteriori distribution of \( z_{ci} \) after observing the data \( X, Y \) (called responsibilities). The responsibilities accounts for the probability of expert \( c \) has generated the sample \( i \).

In the M-Step, the new parameters are found by maximizing the \( Q \)-function, as the following:
\[
\hat{\Omega}^{t+1} = \arg \max_{\Omega} Q^t(\Omega). \tag{7}
\]

The EM runs until convergence, typically measured by the \( Q \)-function.

### C. Possibility Distribution

Possibility theory is a framework for representing uncertainty, and ambiguous knowledge, from possibility distributions [25], [26]. Let \( \Psi \) represent a finite set of mutually exclusive events, where the true alternative is unknown. This lack of information on the true event resumes the uncertainty in representing the true alternative. A possibility distribution, defined in the set \( \Psi \), maps the set of possible events to the unit interval \([0, 1]\), \( \pi : \Psi \rightarrow [0, 1] \), with at least \( \pi(s) = 1 \) for some \( s \in \Psi \), where the function \( \pi(s) \) represents the state of knowledge by the expert about the state of the data, and \( \pi(s) \) stands as the belief of event \( s \) be the true alternative. The larger \( \pi(s) \), the more plausible, i.e., plausible the event \( s \) is. It can also be interpreted as the “degree of belief”; for example, the “degree of belief” of event \( s \) be true is 0.8. It is assumed to be possibilistic rather than probabilistic, so distinct events may simultaneously have a degree of possibility equal to 1.

Given two possibilities distributions \( \pi_1(s) \) and \( \pi_2(s) \), the possibility distribution \( \pi_1(s) \) is said to be more specific than \( \pi_2(s) \), iff \( \pi_1(s) < \pi_2(s) \) \( \forall s \in A \). Then, \( \pi_1 \) is at least as restrictive and informative as \( \pi_2 \). In the possibilistic framework, extreme forms of partial knowledge can be captured, namely, the following.

1. **Complete knowledge**: for some \( s_0 \), \( \pi(s_0) = 1 \), and \( \pi(s) = 0, \forall s \neq s_0 \) (only \( s_0 \) is possible).
2. **Complete ignorance**: \( \pi(s) = 1, \forall s \in \Psi \) (all events are possible).

The more specific \( \pi \) is, the more informative it is. The minimal specificity principle drives possibility theory [27]. It states that any now-known impossible hypothesis cannot be ruled out. It is a principle of minimal commitment, caution, and information. Essentially, the aim is to maximize possibility degrees while keeping constraints in mind.

### III. Contextual Mixture of Experts

In this section, the cMoEs are presented. Section III-A describes the model structure and its goals. Section III-B introduces the possibilities distributions used in the expert knowledge representation. The learning of the cMoEs is given in Section III-E.

#### A. Model

The structure of the cMoEs is composed by \( C \) experts, and gates models; this is represented in Fig. 1. The context
here refers to any meaningful process data characteristic defined by the analyst or derived from any process information/knowledge. Each context $c$ is encoded by a possibility distribution $\pi_c$, which is used to represent the expert/analyst uncertainty knowledge about the respective context. The analyst inputs each context into the cMoEs by incorporating the context into the model structure and defining each expert model’s expected operating region. Then, each contextual expert model $\hat{y}_i(x_i)$ is trained on the region of context $c$ and makes predictions based on its domain representation. This contextual approach enables the prediction to be divided into different components representing meaningful context specified by the analyst.

The output prediction of cMoE is given by a weighted sum of the experts output, given by (1). In cMoE, the gates $g_i(x_i)$ is the probability of sample $x_i$ belonging to the region of context $c$. The expert model $\hat{y}_i(x_i)$ is trained on the region defined by the context $c$, and gives the prediction according with its domain representation. For simplified notation, the contribution of each expert model is defined as $\hat{y}_i = \sum_{c=1}^{C} h_{ci} x_{ci}$, where the input $x_i$ is omitted for clarity, and $h_{ci} = \Delta g_i(x_i)\hat{y}_i(x_i)$, where $\Delta$ stands for defined to be. For example, in a three-phase batch process, the cMoE is set to have three contexts, each one representing a phase. In such case, the simplified representation is given by

$$\hat{y} = h_{\text{phase}_1} + h_{\text{phase}_2} + h_{\text{phase}_3}.$$  

From that, each contextual model can be interpreted separately, or jointly according to the analyst’s needs.

### B. Expert Process Knowledge Representation

Here, the expert’s knowledge for each context is represented by a possibility distribution. For each context $c$, there is an associated possibility distribution $\pi_c(x_i)$ (in short $\pi_c$), where $i \in \Psi$ (where $\Psi$ represents the set of all available samples). The value of $\pi_c$ indicates the degree of belief of the sample $i$ pertains to the region of context $c$. In the case of $\pi_c = 1$, sample $i$ is considered to be fully possible; if $\pi_c = 0$, sample $i$ is considered to be completely impossible to be part of context $c$; any value between these two extremes $\pi_c = p$, $p \in [0,1]$, can be accredited as partial possibility, to a degree certainty $p$, of sample $i$ belonging to the context $c$. Therefore, if $\pi_c = 1$ for all $c = 1, \ldots, C$, the sample $i$ will be defined as being believed to belong to all contextual regions (complete ignorance about sample $i$), whereas if $\pi_c = 1$ for a single $c \in \{1, \ldots, C\}$, the sample $i$ will be defined as being believed to fully certain to belong to context $c$ (complete knowledge about sample $i$), while being impossible to belong the other contexts.

Because expert knowledge’s reliability cannot be fully guaranteed for each context, and because reliability is commonly described with some degree of certainty, for example (80% sure or 70% certain) [28, Ch. 2], the possibility distributions used here are intended to account for this uncertainty in representing expert knowledge of each context. To do so, two possibility distributions are employed in the experiments, the $\alpha$-Certain distribution and the $\beta$-Trapezoidal (fuzzy) distribution, where $\alpha$ and $\beta$ are the degrees of certainty.

### C. $\alpha$-Certain Possibility Distribution

is an imprecise knowledge distribution with a certainty factor $\alpha$. The available knowledge about the true alternative is expressed as a subset $A \subseteq \Psi$ associated with a certain level of trust $\alpha \in [0,1]$, concerning the occurrence of $A$. This can be expressed declaratively as “$A$ is certain to degree $\alpha$.” This type of distribution has been suggested as [29]

$$\pi_{ci} = \begin{cases} 1 & \text{if } i \in A \\ 1 - \alpha & \text{otherwise} \end{cases} \quad (8)$$

If $\alpha = 1$, $\pi_{ci}$ represents the characteristic function of $A$, on the other extreme if $\alpha = 0$, $\pi_{ci}$ represents the total ignorance about $A$. The $\alpha$-Certain possibility distribution is shown in Fig. 2(a).

### D. $\beta$-Trapezoidal Epistemic Possibility Distribution

In the epistemic (fuzzy like) possibility distribution, each event has a degree of belief associated with the possibility of an event. In the epistemic distribution, the available knowledge about the true alternative is given as a constrain defined in terms of “a fuzzy concept” defined in $\Psi$. All standard types defining membership function and representing fuzzy constraints (i.e., triangular, trapezoidal, Gaussian, etc.) can be applied to define epistemic type possibility distributions. In here, the trapezoidal function is defined by a lower limit $a$, an upper limit $d$, a lower support limit $b$, and an upper support limit $c$, where $a < b < c < d$.

$$\pi_{ci} = \max \left( \min \left( \frac{i-a}{b-a}, \frac{d-i}{d-c} \right), \beta \right) \quad (9)$$

To account for the unreliable, a certainty factor $\beta \in [0,1]$ is added, where $\beta = 0$ means a fully unreliable source, and $\beta = 1$ means a fully reliable source. The $\beta$-Trapezoidal distribution is shown in Fig. 2(b). More possibility distributions are also fitted in this framework; for more possibilities distributions, see [28, Ch. 2].

### E. Learning

The goal of cMoE is to integrate the expert knowledge representation (via the possibility distribution) into the model.
structure. To constrain the contextual information, the parameters in $\Omega$ are found trough the maximization of the weighted log-likelihood (WLL) of $\Omega$, defined as

$$
L_C(\Omega) = \sum_{i=1}^{N} \log p(y_i, z_i | x_i; \Omega)^{\pi_i},
$$

(10)

where $\pi_i = [\pi_{i1}, \ldots, \pi_{ic}]^T$ is the contextual weight vector from the expert knowledge and must be specified a priori. The r.h.s power is defined as $p(y_i | x_i; \Omega)^{\pi_i} \triangleq \sum_{c=1}^{C} \pi_{ic} g_c(x_i; \mathcal{V}) p(y_i | x_i; \Theta_c)^{\pi_{ic}}$. The weighted ML estimation of $\Omega$ constrains the contextual information into the model structure, laying down the basis of cMoE framework. The idea is to down-weight samples that have a low degree of belief to belong to regions of expert $c$.

The maximization of the WLL also $L_C(\Omega)$ follows from the EM algorithm. By inserting the contextual weights, the WLL of the complete data distribution becomes

$$
\hat{L}_C(\Omega) = \sum_{i=1}^{N} \log p(y_i, z_i | x_i; \Omega)^{\pi_i},
$$

(11)

where the power at the r.h.s is defined as

$$p(y_i, z_i | x_i; \Omega)^{\pi_i} \triangleq \sum_{c=1}^{C} [g_c(x_i; \mathcal{V}) z_{ci} p(y_i | x_i; \Theta_c)^{z_{ci}}]^{\pi_{ic}}.
$$

Let $\hat{\Omega}^t$ denote the estimated parameters at iteration $t$ of the EM algorithm. The expectation (E-step) and maximization steps (M-step) for WLL are as follows.

a) E-step: From an initial guess $\hat{\Omega}^0$, the expectation of the complete WLL (aka $Q$-function) is computed with respect to the current estimate $\hat{\Omega}^t$

$$Q^t(\Omega) = \sum_{i=1}^{N} \sum_{c=1}^{C} \pi_{ic} \gamma_{ic}^{t} \log [g_c(x_i; \mathcal{V}) p(y_i | x_i; \Theta_c)].
$$

(12)

where

$$
\gamma_{ic}^{t} = \frac{\pi_{ic} g_c(x_i; \mathcal{V}) p(y_i | x_i; \Theta_c)}{\sum_{k=1}^{C} \pi_{ik} g_k(x_i; \mathcal{V}) p(y_i | x_i; \Theta_k)}.
$$

$\gamma_{ic}^{t}$ are the responsibilities of expert $c$ generated sample $i$. It should be noted that the responsibilities are also a result of the contextual weights, in the case of $\pi_{ic} = 0$, the responsibility of expert $c$ is $\gamma_{ic} = 0$, indicating that context $c$ has no role in generating the sample $i$.

b) M-step: In the M-Step, the new parameters are found by maximizing the $Q$-function, as the following:

$$
\hat{\Omega}^{t+1} = \arg \max_{\Omega} Q^t(\Omega).
$$

(13)

The $Q$-function is further decomposed to account for the gates and experts contributions separately, as $Q^t(\Omega) = Q^t_g(V) + Q^t_c(\Theta)$, where

$$Q^t_g(V) = \sum_{i=1}^{N} \sum_{c=1}^{C} \pi_{ic} \gamma_{ic}^{t} \log g_c(x_i; \mathcal{V})
$$

and

$$Q^t_c(\Theta) = \sum_{i=1}^{N} \sum_{c=1}^{C} \pi_{ic} \gamma_{ic}^{t} \log p(y_i | x_i; \Theta_c).
$$

The maximization is performed separately for the experts and gates in the updating phase. Here, the experts and gates models are linear, despite more complex models being allowed and easily integrated into this framework.

F. Experts Learning

In the maximization step, the updated experts parameters are found from the maximization of $Q^t_c(\Theta)$. The contribution of the individual experts can be accounted separately

$$Q^t_c(\Theta) = \sum_{c=1}^{C} Q^t_{ec}(\theta_c) = \sum_{c=1}^{C} \sum_{i=1}^{N} \pi_{ic} \gamma_{ic}^{t} \log N(y_i | x_i^T \theta_c, \sigma_c^2)
$$

(14)

where $Q^t_{ec}(\theta_c)$ accounts for the contribution of expert $c$. Hence, the parameters of expert $c$ can be updated apart from the other experts. Equivalently, instead of maximizing $Q^t_{ec}(\theta_c)$, the updated coefficient is found by minimizing the negative of $Q^t_{ec}(\theta_c)$, as

$$\hat{\theta_c}^{t+1} = \arg \min_{\theta_c} \left( \frac{1}{2} \sum_{i=1}^{N} \pi_{ic} \gamma_{ic}^{t} (y_i - x_i^T \theta_c)^2 \right).
$$

(15)

To promote the sparsity of expert $c$ coefficients, a $\ell_1$ penalty is added to (15), leading to

$$\hat{\theta_c}^{t+1} = \arg \min_{\theta_c} \left( \frac{1}{2} \sum_{i=1}^{N} \pi_{ic} \gamma_{ic}^{t} (y_i - x_i^T \theta_c)^2 + \lambda_c^{t} \sum_{j=1}^{d} | \theta_{cj} | \right)
$$

(16)

where $\lambda_c^{t}$ controls the importance of the regularization penalty. This penalty, also known as least absolute shrinkage and selection operator (LASSO), drives irrelevant features coefficients toward zero. This characteristic is suitable for industrial applications where not all variables are relevant to the prediction, providing compact models. Under the cMoE, this penalty will allow the selection of parsimonious models for each expert, reducing the complexity of the overall model structure. Together with the contextual information, the LASSO penalty will provide a relevant set of features for each expert domain, thus allowing a better interpretation of the model representation, as well allowing the learning in scenarios with small number of samples and many features.

The minimization of (16) will follow from the CGD. In CGD, each coefficient is minimized individually at a time. The updated regression coefficient of variable $j$ and expert $c$ is given as

$$\hat{\theta}_{cj}^{t+1} = S \left( \sum_{i=1}^{N} \pi_{ic} \gamma_{ic}^{t} x_{ij} (y_i - \hat{y}_{cj}^{t}) \right) / \sum_{i=1}^{N} \pi_{ic} \gamma_{ic}^{t} x_{ij}^2
$$

(17)

where $\hat{y}_{cj}^{t} = \sum_{d=1}^{d} \pi_{id} x_{id} \theta_{d}^{t}$ is the fitted value of local expert $c$ without the contribution of variable $j$ and the $S(z, \eta)$ is the soft threshold operator, given by $S(z, \eta) = \text{sign}(z)(|z| - \eta)^+$. From (17), the contextual weight adds a weighting factor over the responsibilities. In the case where $\pi_{ic} = 1$ for all experts, the
responsibility will be the primary driving force in determining the contribution for that specific sample.

1) Experts Model Selection: The selection of LASSO regularization can follow from the $k$-fold cross-validation ($k$-CV) error. However, the $k$-CV for the LASSO may present potential bias issues in high-dimensional settings. The reason for bias is that the data change caused by the splitting into folds affects the results. Allowing $k$ to be large enough to reduce the bias is one possible solution. Choosing $k = N$, for example, results in the LOOCV error, an unbiased estimator for the LASSO error. However, the computation of LOOCV is heavy as it requires training the model $N$ times, and one solution is to approximate the LOOCV from the data. Under mild conditions assumptions [30], the prediction LOOCV of linear models with LASSO penalty can be approximated from its active set; the active set is the index of those variables with nonzero coefficients.

Let the active set of the expert $c$ be $\mathcal{E}_c = \{ j \in \{1, \ldots, d \} : \theta_{cj} \neq 0 \}$. Also, let $X_{\mathcal{E}_c}$ for the columns of matrix $X$ in the set $\mathcal{E}_c$.

The LOOCV of expert $c$ can be approximated by

$$\text{CV}_c(\hat{\lambda}_c) = \sum_{i=1}^{N} \pi_{ci} \gamma_{ci}^{-1} \left( y_i - \hat{y}_{ci}^{-(-i)} \right)^2$$  \hspace{1cm} (18)

where $\hat{y}_{ci}^{-(-i)}$ is the estimated output of expert $c$ without sample $i$. The estimation of $\hat{y}_{ci}^{-(-i)}$ from the active set is given by

$$\hat{y}_{ci}^{-(-i)} = \left( \hat{y}_{ci} - \frac{H_{ci}[1]}{1 - H_{ci}[ii]} \right)$$  \hspace{1cm} (19)

where $[H_{ci}]_{ii}$ are the $i$th diagonal element of hat-matrix of expert $c$, which is defined as

$$H_c = X_{\mathcal{E}_c} \left( X_{\mathcal{E}_c}^T \Gamma_c X_{\mathcal{E}_c} \right)^{-1} X_{\mathcal{E}_c}^T \Gamma_c$$

where $\Gamma_c = \text{diag}(\pi_{c1} \gamma_{c1}, \ldots, \pi_{cn} \gamma_{cn})$ is the diagonal matrix of contextual weights and responsibilities, and $X_{\mathcal{E}_c}$ is the active set subset of matrix $X$. In here, the inverse $(X_{\mathcal{E}_c}^T \Gamma_c X_{\mathcal{E}_c})^{-1}$ is computed via the LU decomposition. If this is not possible, e.g., due the matrix becoming too large, one could estimate the validation error from an independent validation set. The value of $\hat{\lambda}_c$ selected is the one that minimizes the value of $\text{CV}_c$. Note that this estimator resembles the predicted residual sum of squares, except that only the active set is used along with the LOOCV estimation for the LASSO.

G. Gates Learning

For the gates, the new updated parameters results from the maximization of $Q_g(V)$, or equivalently by minimizing $-Q_g(V)$. By expanding the gates contribution, it becomes

$$Q_g(V) = \sum_{i=1}^{N} \sum_{c=1}^{C} \pi_{ci} \gamma_{ci}^{-1} x_i^T v_c - \phi_1 \log \left( \sum_{k=1}^{C} \exp (x_i^T v_k) \right)$$  \hspace{1cm} (20)

where $\phi_1 = \sum_{c=1}^{C} \pi_{ci} \gamma_{ci}$. The function $Q_g(V)$ is concave in the parameters, and its maximization of $Q_g(V)$ will follow the Newton’s method. Let $\{ \hat{v}_c \}_{c=1}^{C}$ be the current estimates of gates coefficients, the second-order (quadratic) Taylor approximation of (20) around $\{ \hat{v}_c \}_{c=1}^{C}$ is

$$\tilde{Q}_g(V) = \sum_{c=1}^{C} Q_{gc}(v_c) + C \left( \{ \hat{v}_c \}_{c=1}^{C} \right)$$  \hspace{1cm} (21)

$$Q_{gc}(v_c) = -\frac{1}{2} \sum_{i=1}^{N} r_{ci} \left( z_{ci} - x_i^T v_c \right)^2$$  \hspace{1cm} (22)

where $\tilde{Q}_g(V)$ is the second-order Taylor approximation of $Q_g(V)$, $Q_{gc}(v_c)$ accounts for the individual contribution of gate $c$, and $C(\{ v_c \}_{c=1}^{C})$ is a constant term, while $r_{ci}$ and $z_{ci}$ are given by

$$r_{ci} = \phi_1 g_{ci} (1 - g_{ci}) \hspace{1cm} (23)$$

$$z_{ci} = x_i^T v_c + \eta \pi_{ci} \gamma_{ci}^{-1} - \phi_1 g_{ci} (1 - g_{ci})$$  \hspace{1cm} (24)

with the gates $g_{ci}$ computed from (3), and the parameter $\eta$ is a magic parameter added to control the Newton update on the optimization phase.

By adding the LASSO penalty to the gates contribution (22), the new gate coefficients are found as the solution of the following minimization problem:

$$v_c^{t+1} = \arg \min_{v_c} \left[ \frac{1}{2} \sum_{i=1}^{N} r_{ci} \left( z_{ci} - x_i^T v_c \right)^2 + \lambda_2 \sum_{j=1}^{d} |v_{cj}| \right]$$

The gate coefficients are updated from successive local Newton steps. It cycles trough all $C$ gates sequentially, where the values of $g_{ci}$ are calculated from $\{ v_c \}_{c=1}^{C}$, and they must be updated as soon a new $\hat{v}_c$ is computed. The computation of (25) must be repeated until the coefficients converge; usually, few iterations (less than 10) are needed to reach convergence.

The solution of (25) is achieved from the CGD, in which

$$\tilde{v}_c^{t+1} = S \left( \sum_{i=1}^{N} r_{ci} x_{ij} (z_{ci} - z_{cj})^2, \lambda_2^2 \right) / \sum_{i=1}^{N} r_{ci} x_{ij}^2$$  \hspace{1cm} (25)

where $z_{cj} = \sum_{j \neq c} z_{cij} \tilde{v}_c^{t+1}$ is the fitted value of gate $c$, without considering variable $j$.

a) Practicalities in gates update: Along the gates coefficients update, some practical issues must be taken.

1) Care should be taken in the update of (24), to avoid coefficients diverging in order to achieve fitted $g_{ci}$ of 0 or 1. When $g_{ci}$ is within $\xi = 10^{-3}$ of 1, (24) is set to $z_{ci} = x_i^T v_c$, and the weights in (23) $r_{ci}$ are set to $\xi$, 0 is treated similarly.

2) The use of full Newton step $\eta = 1$ in optimization (25) do not guarantee the converge of coefficients [31]. To avoid this issue, along the experiments $\eta$ was fixed to $\eta = 0.1$.

1) Gates Model Selection: Similar to the experts’ procedure, the gates model selection will follow from the estimated LOOCV error. The predicted gate output without the sample $i$ is given by

$$\hat{z}_{ci}^{-(-i)} = \left( \hat{z}_{ci} + \frac{[M_c]_{ii} - 1}{(1 - [M_c]_{ii})} \right)$$  \hspace{1cm} (26)
where $M_c$ is the gate hat matrix at each step of the Newton update, and is computed as

$$M_c = X_{G_c}^T (X_{G_c}^T R_c X_{G_c})^{-1} X_{G_c}^T R_c$$  \hspace{1cm} (27)

where $G_c = \{ j \in \{1, \ldots, N \} \mid v_{cj} \neq 0 \}$ is the active set of gate $c$, and $R_c = \text{diag}(r_{p1}, \ldots, r_{pn})$. The LOOCV is then estimated as

$$CV^g(\hat{g}_c) = \sum_{i=1}^{N} r_{ci} \left( z_i - \hat{z}_{ci}(-i) \right)^2.$$  \hspace{1cm} (28)

The gate regularization parameter is selected to minimize the estimated LOOCV error.

### H. EM Stop Condition

In cMoE, the information on the number of experts must be known a priori, or can be defined by the analyst, so this is not an issue for the design. The EM algorithm’s condition stops must be properly defined to avoid overfitting or a poorly chosen model. Because the implementation is based on a set of linear models, an approximation of LOOCV error is used to assess the model’s quality and set the EM algorithm’s stop criteria. The estimated LOOCV for cMoE is given by

$$CV(\Omega) = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \sum_{c=1}^{C} g_{c}(i)\hat{y}_{ci}(i) \right)^2.$$  \hspace{1cm} (29)

where $\hat{g}_{ci}(i)$ is given by (19) and $g_{c}(i)$ is given by

$$g_{c}(i) = \exp \left( \frac{z_{ci}(i)}{\sum_{k=1}^{C} \exp \left( z_{ki}(i) \right)} \right).$$  \hspace{1cm} (30)

The performance of cMoE is checked at each iteration by computing $CV(\Omega)$; it is expected that the estimated LOOCV $CV(\Omega)$ decreases until a minimum, before beginning to increase continuously. This minimum is found by checking whether for $n_{it}$ iterations, the $CV(\Omega)$ kept only increasing. If so, the cMoE of $n_{it}$ iterations back is considered to have a global minimum error and is selected as optimized model, and the algorithm is terminated. The cMoE algorithm should terminate after $n_{it}$ iterations if the error continually increases, counting from the iteration where $CV(\Omega)$ reaching its minimum. In the experiments, a value of $n_{it} = 6$ was considered.

### I. Evaluation of Expert Process Knowledge Integration

In the cMoE model, each gate $g_c$ should reflect the knowledge of each context represented by $\pi_c$, and in the prediction stage, the gates are responsible for automatically determining which context the process is running in and switching to the appropriate expert model. In fact, the $g_c$ is a probability counter-part of the possibility distribution $\pi_c$. The weakest consistency principle [25] leads to a sufficient condition for checking the consistency between the gate’s probability function $g_c$ and the possibility distribution $\pi_c$. It states that a probable occurrence must also be possible to some extent, at least to the same degree. The following inequality can formally express this:

$$g_{ci} < \pi_{ci} \quad \forall i \in \Psi.$$  \hspace{1cm} (31)

The possibility distribution is a upper bound for the probability distribution [29]. Each context possibility distribution $\pi_c$ reflects the knowledge expert’s uncertainty in a quasi-qualitative manner that is less restrictive than probability $g_c$. This is visually represented in Fig. 3, for a hypothetical three phases process. The left picture shows the contextual information provided by the analyst, with a complete ignorance region (common contextual knowledge). The right picture shows the fitted contextual information with well-defined boundaries, represented as the gate’s output. In the specific case where context information follows the complete ignorance possibility distributions ($\pi_{ci} = 1$, for $c = 1, \ldots, C$, and $i = 1, \ldots, N$), the cMoE reduces to MoE.

To access if the assignment of context $c$ is correct, the following consistency index of context $c (C_{1,c})$ is defined

$$C_{1,c} = \frac{1}{N} \sum_{i=1}^{N} I(g_{ci}, \pi_{ci})$$  \hspace{1cm} (32)

where $I(a, b) = 1$ if $a \leq b$, and $0$ otherwise. This consistency index measures the accuracy of the gates’ agreement of context $c$ with respect to the consistency principle. To measure the consistency of all contexts in representing the expert knowledge, the following geometric mean is employed:

$$C_l = \left( \prod_{c=1}^{C} C_{1,c} \right)^{1/C}.$$  \hspace{1cm} (33)

If $C_l = 1$ indicates complete agreement, $C_l = 0$ indicates complete disagreement, indicating an inability of cMoE to incorporate the expert knowledge into its structure. In such cases, the uncertainty in the expert knowledge from the possibility distributions can also be retuned, for example in $\alpha$-Certain distribution, the certainty $\alpha$ parameter can be tuned from automatic methods. In this case, the $\alpha$ should be chosen using the minimal specificity principle, i.e., search for the most informative distribution (lowest $\alpha$), while keeping a desired consistency index. This can be stated as

$$\alpha = \inf \{ \alpha' \in [0, 1] : C_l(\alpha') < \epsilon \}$$  \hspace{1cm} (34)

where $0 \leq \epsilon \leq 1$ is the minimum desired consistency index. The same reasoning can be applied to the $\beta$-Trapezoidal possibility distribution.

### IV. Experimental Results

This section presents the experimental results in two industrial case studies. The first case study deals with the estimation $H_2S$ at...
the output stream of a sulfur recover unit described in [24]. The second case study predicts the acidity number in a multiphase polymerization reactor.

A. State of Art Models

The following models were also implemented along with the experiments for performance comparison purposes. The MoLE with LASSO penalty [6], the LASSO regression model, partial least squares (PLS) regression model, GMR model, decision tree (TREE), and the optimally pruned extreme learning machine model (ELM) [32]. The MoLE and LASSO source code are based on the MoLE Toolbox available at [6]. The PLS is based on the author’s own implementation. The GMR implementation is based on the Netlab Toolbox available at [33]. The TREE is based on the MATLAB implementation of the statistics and machine learning toolbox. The ELM were implemented from the author’s source code, available at [34].

B. Hyperparameters Selection

The selection of MoLE and LASSO regularization parameters follows the predicted LOOCV error described in Section III-F1; the LASSO parameter is denoted as \( \lambda \), while the regularization parameters of MoLE is defined as \( \lambda_p, \lambda_g \), whereas the \( e, g \) superscript denotes the expert and gates parameters, respectively, while \( p \) refers to the expert/gate number. The selection of PLS latent variables \( N_{\text{lat}} \) follows from a 10-fold CV error on the training data. The selection of the hidden neurons \( N_{\text{neu}} \) in the ELM model follows from the optimization procedure described in [32]. The number of components \( N_e \) in the GMR is set equal to the number of contexts. For the tree, the minimum number of leaf node observations \( N_{\text{leaf}} \), was set to be \( N_{\text{leaf}} \) for both experiments.

C. Experimental Settings

The predictive performance is accessed by the root mean square error (RMSE), the coefficient of determination (\( R^2 \)), and the maximum absolute error (MAE). The results of the second case study have been accessed by following a leave-one-batch-out procedure. The models were trained from all batches except one (to be used as a test). This procedure was repeated so that all the batches were used in the testing phase. The performance metrics were averaged and then reported as the final values. Also, a randomization \( t \)-test (from [35]) was used to compare the cMoE performance (the null hypothesis assumes that the RMSE of cMoE and the method to be compared are equal; i.e., equal mean), the \( p \)-value is then reported, if \( p \)-value < 0.05 the null hypothesis is rejected. Along with the training procedure, the data for the training data was autoscaled, and the testing data were rescaled according to the training parameters (mean and variance).

D. SRU Unit

The SRU unit aims to remove pollutants and recover sulfur from acid gas streams. The SRU plant takes two acid gas as input, the first (MEA gas), from gas washing plants, rich in \( \text{H}_2\text{S} \), and the gas from sour water stripping plants (SWS gas), rich in \( \text{H}_2\text{S} \) and \( \text{NH}_3 \) (ammonia). The acid gases are burned into reactors, where \( \text{H}_2\text{S} \) is transformed into sulfur after oxidation reaction with air. Gaseous products from the reaction are cooled down, collected, and further processed, leading to the formation of water and sulfur. The remaining gas nonconverted to sulfur (less than 5%) is further processed for a final conversion phase. The final gas stream contains \( \text{H}_2\text{S} \) and \( \text{SO}_2 \), and online analyzers measure these quantities. The goal is to use the process data to build a soft sensor to replace the online analyzers when under maintenance.

Five main variables are collected, \( X_1 \) is the gas flow in MEA zone, \( X_2 \) is the air flow in MEA zone to the combustion of MEA gas (set manually by the operators), \( X_3 \) is the airflow in MEA zone regulated by an automatic control loop according to the output steam gas composition, \( X_4 \) the airflow in SWS zone (set manually by operators), and \( X_5 \) the gas flow in SWS zone. The target/output is set here to be the \( \text{H}_2\text{S} \) at the end-tail; the \( \text{SO}_2 \) can also be predicted using the same principle presented here. Also, according to [24], the operators are more interested in the models that can predict the \( \text{H}_2\text{S} \) peaks.

There is a total of 10 000 samples available. The first 5000 were used for training, and the remaining 5000 for test; Fig. 4(a) shows the training data for the SRU dataset, the peaks are clearly visible. Time-lagged features were designed to account for the process dynamics. Then, for the five variables, the time lags of \( X_{i,t−d} \) for variables \( i = 1, \ldots, 5 \) and delays \( d = \{0, 5, 7, 9\} \) were considered, resulting in a total of 20 features.
To predict the H$_2$S, and allow a better understanding of the causes of the peaks, a cMoE model with three contexts was designed. The first, representing the operator context, accounts for the peaks. The second context is designed to represent the nonpeaks. The third context represents the remaining process states that are not accounted for the peaks and nonpeaks. The cMoE model is represented by

$$\hat{y} = h_{\text{peak}} + h_{\text{npeak}} + h_{\text{rem}}$$

where “npeak” is a typo to nonpeaks. Two $\beta$-Trapezoidal possibility distributions were designed for peaks context $\pi_{\text{peaks}}$ and nonpeaks context $\pi_{\text{npeaks}}$, while for the remaining context $\pi_{\text{rem}}$, a complete ignorance distribution is assumed, i.e., $\pi_{\text{rem}} = 1$, $\forall i \in \{1, \ldots, 5000\}$, as there are no information about the context. The peaks were selected manually in the training set, and the peak distribution was designed so that the limits of the trapezoidal function were defined to guarantee that the highest values have $\pi_{\text{peaks}} = 1$ peaks. The possibility distribution for the nonpeaks was designed to be complementary to the context of the peak, with a lower bound defined by the certainty $\beta$. A portion of the expert knowledge feed to cMoE is depicted in Fig. 4(c).

From sample 370–400, a peak in H$_2$S is present (as Ref.). $\pi_{\text{peaks}}$ [see Fig. 4(c) samples 370–400, and 460–490]. The context of nonpeak ($\pi_{\text{npeaks}}$) was designed to be complementary to the peaks context. The remaining context ($\pi_{\text{rem}} = 1$) for all samples. The uncertainty $\beta$ was chosen using the consistency principle described in (34). The consistency index and the LOOCV, for different values of the uncertainty parameter $\alpha$, are shown in Fig. 4(b). The results show that $\beta = 0.3$, the selected uncertainty factor, has the lowest LOOCV, with a consistency index of $C_1 = 0.79$.

Fig. 4(d) shows the gates output prediction for the peaks ($G_{\text{peak}}$), nonpeaks ($G_{\text{npeak}}$), and remaining ($G_{\text{rem}}$) contexts in the test set, for samples 4600–4700. There are two peaks in this portion of the test data, between samples 4620–4640, and 4640–4660. The gates of peaks expert $G_{\text{peak}}$ follow the peak pattern by assigning higher contributions to the peak expert model when the peaks are present. The same behavior is perceived in the nonpeaks gates $G_{\text{npeak}}$, which seems to work complementary to the peaks component. The remaining context gates, $G_{\text{rem}}$, seem somewhat oscillating between the patterns; this seems to be related to a constant operation of the system. The gates coefficients allow identifying the root causes of change between the peaks and nonpeaks. The gates coefficients for the peaks, and nonpeaks is shown in Fig. 4(e). The variable $X_3$ (marked with a rectangle dashed line) has the largest difference between the two gates, indicating this is the main variable that acts on the peaks and nonpeaks model switching. Fig. 4(f) shows the variable $X_3$, together with the H$_2$S peaks. $X_3$ represented the input airflow to control the end tail H$_2$S. Thus, H$_2$S is a consequence of $X_3$. It seems that the control system is unstable, and any oscillation in $X_3$ causes a large oscillation in the H$_2$S. One possible solution to improve the stability of H$_2$S and reduce peaks is to improve the control system. Improvements to the control system can have a positive environmental impact by lowering H$_2$S emissions and/or reducing costs associated with H$_2$S post-treatment. The accuracy of the cMoE was compared with the other state of the art models, and the results are shown in Table I. The results show that the cMoE outperforms all the other models with statistical significance. Results confirm that constraining the model to represent the system’s expected behavior positively impacts the prediction performance. Table II shows the parameters obtained for each model in the H$_2$S experiment.

### E. Polymerization

This case study refers to a polymerization batch process for resin production. The material is loaded into the reactor, which then undergoes the five process phases: heating, pressure, reaction, vacuum, and cooling; most of the phase changes are triggered manually by the operators. The phase change is determined by the quality values of the resin, namely the resin acidity number and the resin viscosity. While a physical sensor measures the viscosity, the acidity number is measured three times, one at the vacuum phase and two at the reaction phase. The objective here is to build a soft sensor to measure the acidity number online and better understand the variables that affect the acidity in that two phases.

For this process, there are data for 33 batches in the specification, with a total of 17 variables measured along the process; they are described in Table III. As there are three acidity measurements for each batch, a total of 99 samples is available. The process variables are synchronized with the acidity number by removing the samples that do not have the corresponding acidity number values.

Then, a cMoE model with two contexts was designed to predict acidity and understand the variables that mostly affect the acidity number. The first context represents the reaction phase, and the second to the vacuum phase. The cMoE model for acidity prediction is represented by

$$\hat{y} = h_{\text{reaction}} + h_{\text{vacuum}}.$$
TABLE II
HYPERPARAMETERS OF THE FITTED H₂S MODELS

| Model | Hyperparameters |
|-------|-----------------|
| cMoE  | \( \lambda_{\text{peak}} = 0.00, \lambda'_{\text{peak}} = 0.00 \) | | MoLE | \( \lambda_{\text{peak}} = 1.00, \lambda'_{\text{peak}} = 1.049 \) |
|       | \( \lambda_{\text{rem}} = 1.00, \lambda'_{\text{rem}} = 1.00 \) | | LASSO | \( \lambda = 1.00 \) |
|       | \( \lambda_{\text{rem}} = 12.61, \lambda'_{\text{rem}} = 1.31 \) | | PLS | \( N_{\text{lat}} = 16 \) |
|       | \( \lambda_{\text{rem}} = 12.83, \lambda'_{\text{rem}} = 0.00 \) | | GMR | \( N_c = 3 \) |
|       | | | TREE | \( N_{\text{leaf}} = 1 \) |
|       | | | ELM | \( N_{\text{nu}} = 160 \) |
continuous process, was broken down into three contexts, with the peaks and nonpeaks contexts being the most relevant to operators. As a result, important information and insights on the status of the control system were obtained by inspecting the main variables causing the transition between the peaks and nonpeaks contexts. In the second case study, a multiphase batch process, the information on phase transitions was the knowledge to be integrated into the model. As a result, more insights into phase transitions and the impact of each variable on each phase were realized.

It is worth noting that the linear models in cMoE are sufficient to integrate the expert knowledge in both case studies; also worth mentioning that the uncertainty parameters in both cases studied were refined from the data, together with an analysis of the consistency index. In cases where the consistency index performs poorly, or one wants to employ a more informative distribution (i.e., lower values of uncertainty), nonlinear models for gates and experts may be employed as an alternative to linear models so that the consistency index’s performance is improved. It would be expected that nonlinear modeling must capture the nonlinear behavior of the data that must be relevant in integrating expert knowledge. Furthermore, the \(\ell_1\) penalty is not required for the linear solution of cMoE; other penalties, such as \(\ell_2\) or \(\ell_{1,2}\), can also be employed. In the case of data collinearity, the solution can be obtained by applying the PLS model to experts and gates, as demonstrated in [17].

Compared to other models that are interpretable by nature, such as the Lasso, PLS, DT, and GMR, they lack mechanisms to integrate expert knowledge. Of course, meaningful relations and rules can be extracted from these models, but this is still driven by the data, and if no context is provided, the extraction of relevant information for more complex relationships, as in the two case studies presented here, is not possible. The contextual framework presented here is flexible enough to allow its implementation in many models, including the GMR model.

VI. Conclusion

This article proposes the cMoEs model, a data-driven model devised to incorporate operator domain knowledge into its structure. The proposed approach has been shown to increase predictive performance while achieving a direct interpretation of process variable contribution in each regime of the process. This approach was evaluated on two different problems, demonstrating better statistical performance than conventional machine learning models that do not rely on contextual information. The proposed method has strong potential as a stable and explainable framework to include contextual information in data-driven modeling. This is important to help the transition to Industry 5.0 by increasing the human–machine synergy in the process industry. Future research could concentrate on nonlinear functions for experts and gates learning to improve the predictive performance, as well as explainable methods for interpretability.

ACKNOWLEDGMENT

The authors would like to thank all partners within the project “Integrating Sensor Based Process Monitoring and Advanced Process Control (INSPEC),” managed by the Institute for Sustainable Process Technology (ISPT) in Amersfoort, The Netherlands.

REFERENCES

[1] S. Nahavandi, “Industry 5.0—a human–centric solution,” Sustainability, vol. 11, no. 16, pp. 1–13, 2019.
[2] M. S. Reis, G. Gins, and T. J. Rato, “Incorporation of process-specific structure in statistical process monitoring: A review,” J. Qual. Technol., vol. 51, no. 4, pp. 407–421, 2019.
[3] J. Sansana et al., “Recent trends on hybrid modeling for industry 4.0,” Comput. Chem. Eng., vol. 151, 2021, Art. no. 107365.
[4] L. von Rueden et al., “Informed machine learning—A taxonomy and survey of integrating prior knowledge into learning systems,” IEEE Trans. Knowl. Data Eng., to be published, doi: 10.1109/TKDE.2021.3079836.
[5] K. Wang, R. B. Gopalakrishnan, J. Chen, and Z. Song, “Deep learning of complex batch process data and its application on quality prediction,” IEEE Trans. Ind. Informat., vol. 16, no. 12, pp. 7233–7242, Dec. 2020.
[6] F. Souza, J. Mendes, and R. Araújo, “A regularized mixture of linear experts for quality prediction in multimode and multiphase industrial processes,” Appl. Sci., vol. 11, no. 5, pp. 1–15, 2021.
[7] P. Facco, F. Bezzo, and M. Barolo, “Nearest-neighbor method for the automatic maintenance of multivariate statistical soft sensors in batch processing,” Ind. Eng. Chem. Res., vol. 49, no. 5, pp. 2336–2347, 2010.
[8] L. Zhao, C. Zhao, and F. Gao, “Between-mode quality analysis based multimode batch process quality prediction,” Ind. Eng. Chem. Res., vol. 53, no. 40, pp. 15629–15638, 2014.
[9] Y. He, B. Zhu, C. Liu, and J. Zeng, “Quality-related locally weighted non-Gaussian regression based soft sensing for multimode processes,” Ind. Eng. Chem. Res., vol. 57, no. 51, pp. 17452–17461, 2018.
[10] X. Shi, Q. Kang, M. Zhou, A. Abusorrah, and J. An, “Soft sensing of nonlinear and multimode processes based on semi-supervised weighted Gaussian regression,” IEEE Sensors J., vol. 20, no. 21, pp. 12950–12960, Nov. 2020.
[11] L. Luo, S. Bao, J. Mao, D. Tang, and Z. Gao, “Fuzzy phase partition and hybrid modeling based quality prediction and process monitoring methods for multiphase batch processes,” Ind. Eng. Chem. Res., vol. 55, no. 14, pp. 4045–4058, 2016.
[12] J. Yu and S. J. Qin, “Multiway Gaussian mixture model based multiphase batch process monitoring,” Ind. Eng. Chem. Res., vol. 48, no. 18, pp. 8585–8594, 2009.
[13] W. Shao, Z. Ge, Z. Song, and J. Wang, “Semisupervised robust modeling of multimode industrial processes for quality variable prediction based on student’s t mixture model,” IEEE Trans. Ind. Informat., vol. 16, no. 5, pp. 2965–2976, May 2020.
[14] B. Wang, Z. Li, Z. Dai, N. Lawrence, and X. Yan, “Data-driven mode identification and unsupervised fault detection for nonlinear multimode processes,” IEEE Trans. Ind. Informat., vol. 16, no. 6, pp. 3651–3661, Jun. 2020.
[15] W. Shao, Z. Ge, and Z. Song, “Soft-sensor development for processes with multiple operating modes based on semi-supervised Gaussian mixture regression,” IEEE Trans. Control Syst. Technol., vol. 27, no. 5, pp. 2169–2181, Sep. 2019.
[16] J. Wang, W. Shao, and Z. Song, “Student’s-t mixture regression-based robust soft sensor development for multimode industrial processes,” Sensors, vol. 18, no. 11, pp. 1–20, 2018.

[17] F. A. A. Souza and R. Araújo, “Mixture of partial least squares experts and application in prediction settings with multiple operating modes,” Chemometrics Intell. Lab. Syst., vol. 130, pp. 192–202, Jan. 2014.

[18] A. Khalili, “New estimation and feature selection methods in mixture-of-experts models,” Can. J. Statist., vol. 38, no. 4, pp. 519–539, 2010.

[19] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” J. Amer. Statist. Assoc., vol. 96, no. 456, pp. 1348–1360, 2001.

[20] B. Peralta and A. Soto, “Embedded local feature selection within mixture of experts,” Inf. Sci., vol. 269, pp. 176–187, 2014.

[21] F. Chamroukhi and B. T. Huynh, “Regularized maximum-likelihood estimation of mixture-of-experts models for regression and clustering,” in Proc. Int. Joint Conf. Neural Netw., Rio, Brazil, 2018, pp. 1–8.

[22] T. Nguyen, H. D. Nguyen, F. Chamroukhi, and G. J. McLachlan, “An $l_1$-oracle inequality for the lasso in mixture-of-experts regression models,” Jan. 2020, arXiv: 2009.10622.

[23] B. T. Huynh and F. Chamroukhi, “Estimation and feature selection in mixtures of generalized linear experts models,” Jul. 2019, arXiv:1907.06994.

[24] L. Fortuna, S. Graziani, and M. G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes. Berlin, Germany: Springer, 2007.

[25] A. Khalili, Possibility Theory: Qualitative and Quantitative Aspects. Dordrecht, The Netherlands:Springer, 1998, pp. 169–226.

[26] R. R. Yager, “Measuring tranquility and anxiety in decision making: An application of fuzzy sets,” Int. J. Gen. Syst., vol. 8, no. 3, pp. 139–146, 1982.

[27] B. Solaiman and É. Bossé, Fundamental Possibilistic Concepts. Cham, Switzerland:Springer International Publishing, 2019, pp. 13–46.

[28] R. R. Yager, “Variable selection via nonconcave penalized likelihood and its oracle properties,” J. Amer. Statist. Assoc., vol. 96, no. 456, pp. 1348–1360, 2001.

[29] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” J. Amer. Statist. Assoc., vol. 96, no. 456, pp. 1348–1360, 2001.