

Sensor Validation in Carbon Capture and Storage Infrastructures

Amirshayan Haghipour¹ , Gianluca Tabella^{2*} , Jacob Stang², and Pierluigi Salvo Rossi^{1,2,**} 

¹Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway

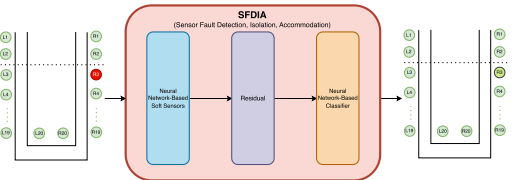
²Department of Gas Technology, SINTEF Energy Research, 4761 Trondheim, Norway

*Member, IEEE

**Senior Member, IEEE

Manuscript received 19 August 2025; accepted 11 September 2025. Date of publication 17 September 2025; date of current version 2 October 2025.

Abstract—Reducing CO₂ emissions in the atmosphere is a critical task, and carbon capture and storage (CCS) plays a crucial role in various actions for mitigating climate change. Safety issues and fiscal metering in CCS systems require reliable sensor measurements. In this work, we propose an architecture for sensor validation, i.e., performing sensor-fault detection, isolation, and accommodation (SFDIA). The architecture is layered and based on soft sensors and a soft classifier. Both the soft sensors and the classifier are built as neural networks. The performances of the SFDIA architecture have been assessed using real-world measurements from a real-scale research facility at SINTEF Energy Research, namely, DeFACTO, and synthetically generated faults. Several experiments have been conducted to explore the transferability of the trained models across time and space.



Index Terms—Sensor signal processing, carbon capture and storage (CCS), machine learning, sensor validation.

I. INTRODUCTION

Combustion of fossil fuels is a primary anthropogenic factor contributing to CO₂ emissions [1]. Using cleaner energy and *carbon capture and storage* (CCS) technologies are among the ways to mitigate the problem of climate change. Replacing fossil fuels with cleaner energy is unlikely in the short and medium terms [2], [3], thus making CCS crucial [4]. CCS comprises technologies that capture, transport, and store carbon in safe geographical places instead of releasing it into the atmosphere, and one critical task is CO₂ injection into an underground formation [5]. Studying CO₂ behavior when injected into the formation is essential to prevent damage formation. We exploit the DeFACTO facility, i.e., a plant built to study CO₂ behavior in thermodynamic conditions often experienced in CCS contexts [6].

Accurate measurement of stored carbon at various stages is essential for safety issues and fiscal metering [7]. However, sensors are prone to faults due to harsh environmental conditions (e.g., pollution, humidity, and corrosion) or internal factors (e.g., overheating) [8], [9]. Thus, it is vital to ensure sensor reliability. The general framework of *sensor-fault detection, isolation, and accommodation* (SFDIA) includes three fundamental tasks: 1) *fault detection*, declaring if the measurements are reliable or not; 2) *fault isolation*, identifying the sensors providing faulty measurements; and 3) *fault accommodation*, replacing the faulty measurements with trustworthy estimates.

Sensor validation can be based on hardware or software/analytical redundancy [10]: the former exploiting dissimilarities between different measurements of the same parameter but leading to increased cost and size; the latter exploiting inconsistencies between actual and expected measurements. Analytical redundancy is obtained by developing soft/virtual sensors [11] via *model-based* and *data-driven* approaches: the former are suitable when the mathematical model of the system is known, the latter are suitable when the model

dynamic is complex, and measurements are available [12]. Several methodologies have been explored to develop soft sensors, including convolutional neural networks [13], attention-based mechanisms [14], autoencoders [15], reinforcement learning [16], and graph neural networks [17]; observer-based methods [18], autoregressive models [19], and Kalman-based architectures [20].

We consider the modular architecture in [21] and present the results from a hybrid validation experiment with real-world measurements and synthetically-generated faults for CCS infrastructures: 1) for the first time, a state-of-the-art data-driven SFDIA architecture is utilized to monitor steady-state CO₂ flow in a real plant; 2) comprehensive performance assessment of detection, isolation, and accommodation is presented;¹ and 3) preliminary results related to model transferability across subsets of sensors are discussed.

The rest of this letter is organized as follows. Section II provides information about the data. Section III explains the system architecture. The experimental setup is described in Sections IV. Section V presents the numerical results. Finally, Section VI concludes this letter.

II. DATA GENERATION

DeFACTO (see Fig. 1) is a research facility in Trondheim (Norway) operated by SINTEF Energy Research and meant for CCS-related experimental activities, working both with the steady state and transient flow of CO₂. DeFACTO allows the circulation of CO₂ in a U-Tube loop that travels underground for ca. 90 m, instrumented with temperature, pressure, and flow rate sensors. We consider steady-state flow mode and select error-free measurements from the temperature sensors inside the well (see Fig. 2). During standstill, the temperature is kept constant by circulating 3.8 m³/h of a mixture with 70% ethanol in water with a constant inlet temperature of 20°C. The temperature sensors are thermocouples of type T, and the sensors are placed inside the process tubes for CO₂ and in the ethanol/water mixture. The sampling frequency is $\frac{2}{3}$ Hz.

¹As performance metrics, we consider *receiver operating characteristic* (ROC), multiclass ROC, *correct classification rate* (CCR), heat map of confusion matrix, and *probability density function* (PDF) of estimation error.

Corresponding author: Amirshayan Haghipour (e-mail: amirshayan.haghipour@ntnu.no).

Associate Editor: Jeronimo Segovia-Fernandez.

Digital Object Identifier 10.1109/LSSENS.2025.3611323

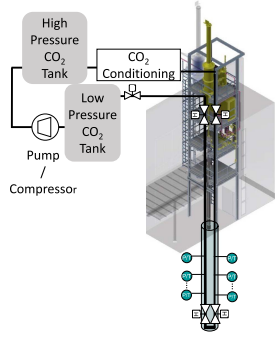


Fig. 1. DeFACTO.

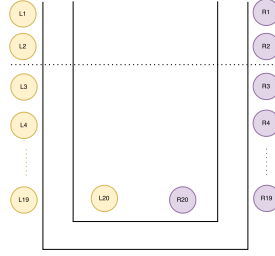


Fig. 2. Selected temperature sensors.

To perform a systematic analysis with significant results from a statistical point of view, synthetic faults are superimposed to the error-free measurements provided via real-world experiments on DeFACTO. We focus on two common types [22], [23], namely, *bias* and *drift*, whose mathematical representations are, respectively

$$b[n] = \begin{cases} b, & n - k \in [0, T) \\ 0, & \text{otherwise} \end{cases} \quad d[n] = \begin{cases} \frac{b(n-k)}{T_d}, & n - k \in [0, T_d) \\ b, & n - k \in [T_d, T_d + T_b) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

b is the level of the fault, $T = T_d + T_b$ is the duration of the fault, T_d is the duration of the drift period, T_b is the duration of the bias period, and k is the starting time of the fault; a signal model $x[n] = s[n] + b[n] + d[n]$ where $s[\cdot]$ and $x[\cdot]$ denote the measured signals in absence and presence of faults, respectively. We assume that side information is available via the following label signal:

$$\ell[n] = \begin{cases} 1, & n - k \in [0, T) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $\ell[n] = 1$ (resp. $\ell[n] = 0$) denotes the presence (resp. absence) of a fault at the n th discrete time.

III. SFDIA ARCHITECTURE

We consider a system equipped with U sensors and denote $x_i[n]$ the (potentially faulty) measurement of the i th sensor at the n th discrete time ($i \in 1, \dots, U$). Also, we denote $\mathbf{x}[n] = (x_1[n], \dots, x_U[n])$ the vector collecting the measurements of all (unreliable) sensors at the n th discrete time, and $\mathbf{x}_{\sim(i)}[n] = (x_1[n], \dots, x_{i-1}[n], x_{i+1}[n], \dots, x_U[n])$ the vector collecting the measurements of all unreliable sensors except the i th one at the n th discrete time. The system consists of three main parts (shown in Fig. 3), which include $U + 1$ networks: 1) a bank of U *soft sensors* (trained on healthy data), 2) a layer for *residual computation*, and 3) one *classifier* (trained on labeled).

The first part consists of U soft sensors; each implemented as a neural network trained to capture the behavior of one single unreliable sensor. Specifically, the i th soft sensor receives as input the measurements from all the unreliable sensors except the i th unreliable sensor (from current discrete time down to w_f previous discrete time), and it aims at reproducing the measurement of the i th unreliable sensor (at current discrete time). The mathematical description of the i th soft

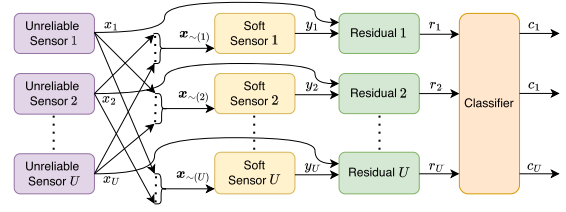


Fig. 3. SFDIA architecture.

Table 1. Depth Level of the Sensors

Sensor	Depth	Sensor	Depth	Sensor	Depth	Sensor	Depth
L1 (R1)	-2 m	L6 (R6)	10 m	L11 (R11)	38 m	L16 (R16)	66 m
L2 (R2)	0 m	L7 (R7)	15 m	L12 (R12)	43 m	L17 (R17)	72 m
L3 (R3)	2 m	L8 (R8)	21 m	L13 (R13)	49 m	L18 (R18)	77 m
L4 (R4)	5 m	L9 (R9)	27 m	L14 (R14)	55 m	L19 (R19)	83 m
L5 (R5)	7 m	L10 (R10)	32 m	L15 (R15)	60 m	L20 (R20)	83 m

sensor is $y_i[n] = f_i^{(N)}(\mathbf{x}_{\sim(i)}[n], \dots, \mathbf{x}_{\sim(i)}[n - w_f + 1])$, where $y_i[n]$ is the estimate by the i th soft sensor at the n th discrete time, $f_i^{(N)}(\cdot)$ denotes a neural network with N hidden layers, and w_f is the size of the sliding window selecting the inputs.

The second part computes the residuals as the difference between the measurements from the unreliable sensors and the estimates from the corresponding virtual sensors, i.e., (in vector form) $\mathbf{r}[n] = \mathbf{x}[n] - \mathbf{y}[n]$, with self-explanatory meaning for the vectors $\mathbf{r}[n]$ and $\mathbf{y}[n]$. Other types of distance between the measurements by the sensors and the estimates by the soft sensors could be considered in this layer.

The third part performs multiclass classification by processing the residual vectors (from current discrete time down to w_g previous discrete time) with a neural network trained to identify inconsistent behavior of the sensors. The mathematical description of the classifier is $c_i[n] = g_i^{(M)}(\mathbf{r}[n], \dots, \mathbf{r}[n - w_g + 1])$, where $g_i^{(M)}(\cdot)$ denotes a neural network with M hidden layers and w_g is the size of the sliding window selecting the inputs. The output $c_i[n] \in [0, 1]$ represents a soft assessment (namely, score) for the i th sensor being healthy or faulty at the n th discrete time.

Although the framework could potentially handle multiple simultaneous faults via thresholding mechanisms in the last layer of the classifier, we only consider the case that at most one single sensor can be faulty at each discrete time. The score vector $\mathbf{c}[n]$ has a central role in the SFDIA mechanism: 1) Detection is performed by comparing the largest entry of the score vector $\mathbf{c}[n]$ with a predefined threshold (θ) and a fault is declared present (respectively, absent) if $\max_{1 \leq i \leq U} c_i[n] > \theta$ (respectively, $\max_{1 \leq i \leq U} c_i[n] \leq \theta$); 2) if a fault is declared present, isolation is performed by selecting the index of the largest entry of the score vector $\mathbf{c}[n]$, i.e., the j th sensor is considered faulty with $j = \arg \max_{1 \leq i \leq U} c_i[n]$; 3) if a fault is declared present and the j th sensor is identified as faulty, then accommodation is performed by replacing the measurement of the faulty sensor ($x_j[n]$) with the estimate ($y_j[n]$) from the corresponding soft sensor.

IV. EXPERIMENTAL SETUP

From DeFACTO, we selected 40 *temperature sensors* with symmetric spatial configuration, as shown in Fig. 2, where the dotted line is the ground level and the depth levels of the sensors are in Table 1. To reduce the computational load, the time series measured on DeFACTO have been downsampled with a factor of 4 using a Chebyshev filter of type I and order 8 to avoid aliasing [24]. Each time series is then normalized to span the range $[-1, +1]$.

Multilayer perceptrons are considered for practical implementation of the neural networks $f_i^{(N)}(\cdot)$ and $g_i^{(M)}(\cdot)$. As for the soft sensors, we considered $N = 2$ hidden layers, each with ten nodes. The first layer is fully connected, while the second layer is a dropout layer (to prevent overfitting [25]) with drop probability 0.2. The sliding window selecting the inputs has size $w_f = 20$. As for the classifier, we considered $M = 6$ hidden layers. The first layer is fully connected, the second and fourth layers are dense layers with reduced output size (15), the third and fifth layers are dropout layers with drop probability

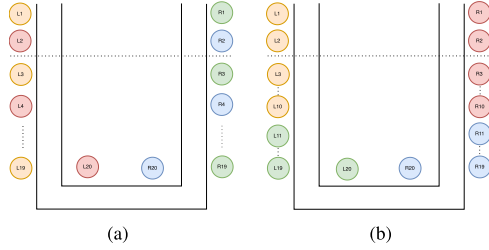


Fig. 4. Experiments for transferability across space: models trained on orange (resp. green) sensors are tested on red (resp. blue) sensors. (a) Sequential scenario. (b) Same-depth scenario.

0.1, and the last layer is a dense layer with output size equal to U . The sliding window selecting the inputs has size $w_g = 20$.

The dataset is divided into groups of size 70%, 15%, and 15% for training, validation, and testing, respectively. We train the neural networks for 40 epochs and incorporate early stopping to prevent overfitting and improve generalization [26]. Soft sensors are trained using the *mean squared error* between the fault-free measurement from each sensor and the estimate provided by the neural network as a loss function. The classifier is trained using the binary cross-entropy between the label signals and the quantized scores provided by the neural network as a loss function. A random combination of different types of faults is superimposed to the fault-free measurements: 250 faults added to the training data and 25 faults added to the test data. The fault duration is generated as a uniform random variable between four and 12 samples, and the absolute level of the faults is generated as a uniform random variable between 0.2 and 0.9 (positive/negative sign selected with equal probability).

Experiments were performed to investigate the model transferability across the temporal dimension. In the experiment *No Split*, the first 85% of the data (from a temporal perspective) is used for training and validation, and the remaining 15% is used for testing. In the experiment *Split S*, the data are divided into S consecutive sections (from a temporal perspective): the first 85% of each section (from a temporal perspective) is used for training and validation, and the remaining 15% of each section is used for testing. We present results related to data splitting into $S \in \{5, 10, 20\}$ sections.

Experiments were performed to investigate the model transferability across the spatial dimension, i.e., understanding if a model trained on a specific group of sensors is useful for monitoring a different group of (similar) sensors. This issue is potentially relevant in industrial applications where a system needs to be deployed without the possibility of a tailored training process. We consider two experiments, as shown in Fig. 4. In *sequential scenario*, the odd-numbered sensors from Group L (respectively, Group R) are used for training and validation, and the even-numbered sensors from Group L (respectively, Group R) are used for testing. In *same-depth scenario* upper (respectively, lower), the first (respectively, last) ten consecutive sensors from Group L are used for training/validation, and the corresponding first (respectively, last) ten consecutive sensors at the same depth from Group R are used for testing. Training/validation and testing are then reversed, and performance is averaged.

V. RESULTS

The following results are obtained as average across ten repeated experiments. Alternative methods on the same dataset have shown inferior performance, confirming the comparison discussed in [12].

We consider the *detection rates* and *false-alarm rates* as performance metrics for the detection task. The curve describing the tradeoff between these two metrics by changing the threshold value (θ) is the ROC curve, a common tool for performance assessment in binary detection problems. Fig. 5 shows the ROC performance of the experiments related to the transferability across time. While the performance during training is similar for all considered cases, it is shown that the case *No Split* exhibits no degradation during testing. In contrast, *Split S* cases introduce a degradation increasing with the number of

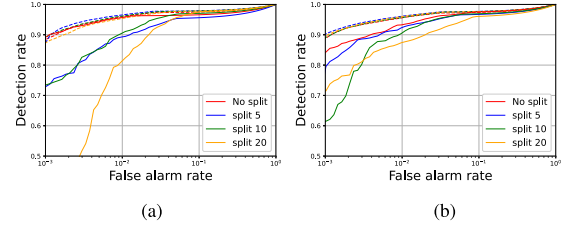


Fig. 5. ROC curves for transferability across time: training (dashed lines) and testing (solid lines). (a) Group L. (b) Group R.

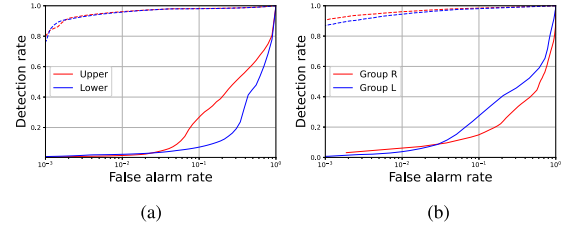


Fig. 6. ROC curves for transferability across space: training (dashed lines) and testing (solid lines). (a) Same altitude. (b) Sequential.

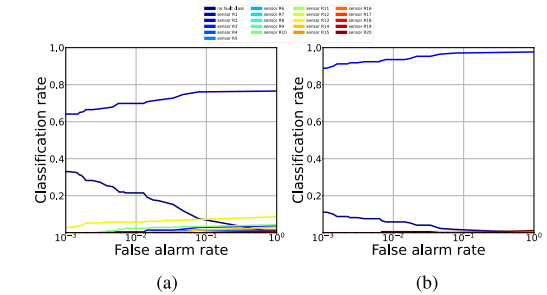


Fig. 7. Isolation performance via multiclass ROC curves (*No Split*). (a) Sensor R1. (b) Sensor R2.

sections S and decreasing with the false-alarm rate. Fig. 6 shows the ROC performance of the experiments related to the transferability across space. While the performance during training is similar for all considered cases, it is shown that transferability across space does not hold for DeFACTO. Thus, data from every individual sensor to be monitored are required for proper model training.

Isolation performances are shown via the *multiclass ROC*, a generalization of the ROC from binary detection to multiple-hypotheses decision [12]. More specifically, the various correct classification rates for each class are plotted against the false-alarm rate according to different selections of the threshold (θ). We report the results related to the case *No Split*, while results for the cases *Sequential Scenario* and *Same-Depth Scenario* are not reported. Fig. 7 shows the multiclass ROC curves for sensors R1 and R2 evaluated during testing. It is apparent how misclassification is significantly negligible, with some exceptions for some sensor pairs exhibiting low but not negligible misclassification rates at very small false-alarm rates, e.g., sensors R1 and R2 in Fig. 7(a). Furthermore, we report in Fig. 8 and Table 2 the confusion matrix via heat map representation and the CCR, respectively, obtained on the test dataset for the threshold (θ) corresponding to the point of operation providing the maximum Youden index² during training. We select the error between the real value of the fault-free signal and the estimation of the corresponding soft sensor in the case of correct fault detection and isolation as a performance metric for the accommodation task. We use kernel density estimation with a Gaussian kernel to compute the PDF during training and testing [28]. Fig. 9 shows the results averaged on all the sets of sensors, where it is apparent how the estimation error is primarily

²It is the difference between the detection and false-alarm rate [27].

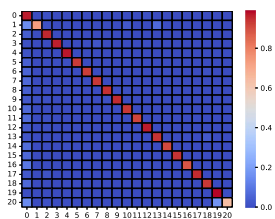


Fig. 8. Confusion matrix with maximum Youden index.

Table 2. Isolation Performance

Sensor	CCR	std	Sensor	CCR	std
No Fault	95.28%	3.41%	R11	93.52%	3.23%
R1	93.30%	4.36%	R12	92.98%	6.80%
R2	93.91%	4.40%	R13	84.34%	7.99%
R3	96.06%	1.89%	R14	87.46%	6.57%
R4	94.60%	2.20%	R15	91.18%	16.68%
R5	94.79%	2.11%	R16	94.49%	2.25%
R6	94.77%	2.66%	R17	95.85%	1.70%
R7	95.20%	2.97%	R18	91.25%	5.30%
R8	94.84%	4.65%	R19	87.62%	5.17%
R9	90.22%	3.95%	R20	49.41%	39.16%
R10	96.13%	1.53%	Average	90.82%	6.14%

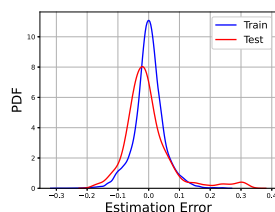


Fig. 9. Estimation-error PDF (accommodation performance).

Table 3. Performance Comparison

Method	CCR	std
MLP	86.51%	6.01%
CNN	89.17%	7.62%
AE	81.15%	14.41%
proposed	90.82%	6.14%

limited in the interval $[-0.2, +0.2]$ both during the training and testing phases. Finally, we compared the proposed layered architecture with alternative solutions, which do not compute explicitly the residuals and implement classification directly from processing the inputs. Table 3 shows the CCR of three alternative solutions based on: multilayer perceptron (MLP) with two hidden layers and 20 nodes per layer; convolutional neural network (CNN) with three layers having 20 filters with 3×3 kernels; autoencoder (AE) with three hidden layers with ten, five, and ten nodes, respectively [29].

VI. CONCLUSION

We presented an SFDIA architecture for monitoring CCS systems. Performance assessment was performed using real measurements from 40 temperature sensors of the DeFACTO facility at SINTEF Energy Research and synthetically generated faults. Experiments have been conducted to explore the transferability of the trained model across time and space: the results were positive in the first case but not in the second, suggesting that measurements are required from each sensor to be monitored.

ACKNOWLEDGMENT

This work was supported by the Research Council of Norway through project SIGNIFY within the IKTPLUSS framework under Grant 311902.

REFERENCES

- [1] M. F. Hasan, M. S. Zantye, and M.-K. Kazi, "Challenges and opportunities in carbon capture, utilization and storage: A process systems engineering perspective," *Comput. Chem. Eng.*, vol. 166, 2022, Art. no. 107925.
- [2] M. Bui et al., "Carbon capture and storage (CCS): The way forward," *Energy Environ. Sci.*, vol. 11, no. 5, pp. 1062–1176, 2018.
- [3] F. Karimi and R. Khalilpour, "Evolution of carbon capture and storage research: Trends of international collaborations and knowledge maps," *Int. J. Greenhouse Gas Control*, vol. 37, pp. 362–376, 2015.
- [4] S. Budinis et al., "An assessment of CCS costs, barriers and potential," *Energy Strategy Rev.*, vol. 22, pp. 61–81, 2018.
- [5] J. Gibbins and H. Chalmers, "Carbon capture and storage," *Energy Policy*, vol. 36, no. 12, pp. 4317–4322, 2008.
- [6] F. Finotti, "DeFACTO: Underground testing of CO₂ transport and injection," 2020. [Online]. Available: <https://blog.sintef.com/sintefenergy/ccs/defacto-underground-testing-of-co2-transport-and-injection>
- [7] S. W. Løvseth, Y. Arellano, H. Deng, F. Finotti, E. Jukes, and G. Bottino, "Enabling CCS via fiscal metering," in *CO₂ Capture, Transport and Storage. Short Papers From the Int Trondheim CCS Conf.* San Francisco, CA, USA: Academic, 2021.
- [8] D. Li, Y. Wang, B. Wang, C. Wang, and Y. Duan, "Recent advances in sensor fault diagnosis: A review," *Sensors Actuators A: Phys.*, vol. 309, 2020, Art. no. 111990.
- [9] A. Chawla et al., "IoT-based monitoring in carbon capture and storage systems," *IEEE Internet Things Mag.*, vol. 5, no. 4, pp. 106–111, Dec. 2022.
- [10] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, "A review on recent development of spacecraft attitude fault tolerant control system," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3311–3320, May 2016.
- [11] C. G. Mattera, J. Quevedo, T. Escobet, H. R. Shaker, and M. Jradi, "A method for fault detection and diagnostics in ventilation units using virtual sensors," *Sensors*, vol. 18, no. 11, 2018, Art. no. 3931.
- [12] H. Darvishi, D. Ciunzo, E. R. Eide, and P. Salvo Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4827–4838, Feb. 2021.
- [13] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
- [14] L. Feng, C. Zhao, and Y. Sun, "Dual attention-based encoder-decoder: A customized sequence-to-sequence learning for soft sensor development," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3306–3317, Aug. 2021.
- [15] X. Zhang, M. Kano, and M. Tani, "Stacked supervised Poisson autoencoders-based soft-sensor for defects prediction in steelmaking process," *Comput. Chem. Eng.*, vol. 172, 2023, Art. no. 108182.
- [16] J. Xie, O. Dogru, B. Huang, C. Godwaldt, and B. Willms, "Reinforcement learning for soft sensor design through autonomous cross-domain data selection," *Comput. Chem. Eng.*, vol. 173, 2023, Art. no. 108209.
- [17] H. Darvishi, D. Ciunzo, and P. Salvo Rossi, "Deep recurrent graph convolutional architecture for sensor fault detection, isolation and accommodation in digital twins," *IEEE Sensors J.*, vol. 23, no. 23, pp. 29877–29891, Dec. 2023.
- [18] E. Bernardi and E. J. Adam, "Observer-based fault detection and diagnosis strategy for industrial processes," *J. Franklin Inst.*, vol. 357, no. 14, pp. 10054–100 81, 2020.
- [19] N. Jihani, M. Kabbaj, and M. Benbrahim, "Kalman filter based sensor fault detection in wireless sensor network for smart irrigation," *Results Eng.*, vol. 20, 2023, Art. no. 101395.
- [20] K. Shaheen, A. Chawla, F. E. Uilhoorn, and P. Salvo Rossi, "Model-based architecture for multi-sensor fault detection, isolation and accommodation in natural-gas pipelines," *IEEE Sensors J.*, vol. 24, no. 3, pp. 3554–3567, Feb. 2024.
- [21] H. Darvishi, D. Ciunzo, E. R. Eide, and P. Salvo Rossi, "A data-driven architecture for sensor validation based on neural networks," *IEEE Sensors J.*, vol. 24, no. 3, pp. 3554–3567, Feb. 2024.
- [22] H. Liu et al., "Faulty sensor detection, identification and reconstruction of indoor air quality measurement in subway station," in *Proc. 11th Int. Conf. Control, Autom. Syst.*, 2011, pp. 323–328.
- [23] X. J. Luo et al., "Development of clustering-based sensor fault detection and diagnosis strategy for chilled water system," *Energy Buildings*, vol. 186, pp. 17–36, 2019.
- [24] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. London, U.K.: Pearson, 2010.
- [25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [26] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2002, pp. 55–69.
- [27] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.
- [28] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Hoboken, NJ, USA: Wiley, 2015.
- [29] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8462–8471, Sep. 2020.