Soft Sensor Design using Hierarchical Multi-Fidelity Modeling with Bayesian Optimization for Input Variable Selection

Johannes Lips * **Hendrik Lens** * **Paolo Conti** **

* *Department of Power Generation and Automatic Control, IFK, University of Stuttgart, Germany, {johannes.lips; hendrik.lens}@ifk.uni-stuttgart.de*
** *MOX - Department of Mathematics, Politecnico di Milano, Italy, paolo.conti@polimi.it*

Abstract:
Soft sensors, or inferential sensors, are crucial in quality and process control systems because they allow for efficient, online estimation of essential quantities that are otherwise difficult or expensive to measure directly. In many applications, it is common to use cost-effective measurement equipment, offering faster data collection than high-fidelity measurements, albeit at the price of reduced accuracy. These low-fidelity data can provide useful information to enhance the estimation of output quantities of interest, thereby facilitating the design of inferential control systems. In this work, we introduce an innovative approach to soft sensing by employing hierarchical, multi-fidelity surrogate models as soft sensors, integrated with Bayesian optimization for input variable selection. Our method creates a parsimonious model by identifying and organizing relevant inputs into a fidelity hierarchy, which enables a multi-fidelity neural network to sequentially refine estimations by extracting crucial information progressively. First, we showcase the effectiveness of the proposed framework on a numerical benchmark, then we use our method to create a surrogate model as soft sensor for accurately determining the atmospheric particulate matter concentration (PM2.5) using real data collected from low-cost sensors.

Keywords: Soft sensor; Multi-fidelity modeling; Bayesian optimization; Variable selection

1. INTRODUCTION

1.1 *Soft Sensors and Challenges in Soft Sensor Design*

Soft sensors, also known as inferential sensors or virtual sensors, are software-based methods that combine available measurements with dynamic models in order to estimate observable system variables that are not measured directly. By this definition, state observers are a type of soft sensor. Soft sensors are often used in process control to estimate process variables that cannot be automatically measured or can only be measured at high cost, with low sampling frequency or with high delays (Souza et al., 2016). The output of a soft sensor can be used as feedback signal to a controller, which is the basis of inferential control. An industrial example of inferential control using Model-Predictive Control and a soft sensor based on Partial Least Squares is given by Kim et al. (2013).

The soft sensor setup we propose is shown in Fig. 1. The soft sensor takes as input the output measurements $\mathbf{y}_{\mathtt{M}}$ of a system actuated by $\mathbf{u}$ and affected by unknown disturbances and noise. It should be noted that measurements of the inputs $\mathbf{u}$ can be included in $\mathbf{y}_{\mathtt{M}}$. The system outputs are noisy, have limited measurement resolution, and might be affected by sensor dynamics. For these reasons, any subset of $\mathbf{y}_{\mathtt{M}}$ can be seen as a set of Low-Fidelity (LF)

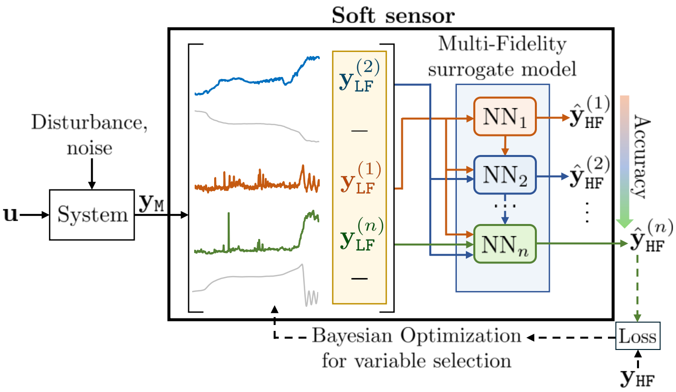

Fig. 1. Proposed soft sensor design using a hierarchical multi-fidelity surrogate model. Bayesian optimization is used for input variable selection during training.

signals, suggesting that these signals are less reliable than the High-Fidelity (HF) variables $\mathbf{y}_{\mathtt{HF}}$, which are the output quantities that we aim to estimate [1]. The development of the soft sensor typically requires a dataset consisting of $(\mathbf{y}_{\mathtt{LF}}, \mathbf{y}_{\mathtt{HF}})$ pairs. If the soft sensor is a dynamic system, the training dataset should be one or multiple time-series with appropriate time resolution.

[1] This terminology is consistent with the one used in the Multi-Fidelity modeling field.

The main challenges in soft sensor development are the treatment of missing data, outlier detection, the selection of input variables, as well as model training, validation and maintenance. Souza et al. (2016) present an overview of the most common methods to address these issues. A review on the use of data-driven soft sensing and control methods in process industry, focused on practical implementation, is provided by Lawrence et al. (2024).

1.2 Hierarchical Multi-Fidelity Regression as Soft Sensor

In this contribution, we explore the possibilities of using a hierarchical Multi-Fidelity (MF) model as a soft sensor, using the setup shown in Fig. 1.

MF modeling was developed to deal with the limitations of computational resources when computing HF simulation outputs, especially in applications that require results in real-time and/or at a high sampling rate. In order to address this issue, MF models incorporate LF data that are readily available, easily accessible or cheaply computable, to improve and speed up HF estimations. This creates an equivalence between MF modeling and soft sensor design, in which LF system outputs are used to estimate HF variables of interest.

A wide range of MF methods have been developed using Gaussian Processes (GP) (Kennedy and O'Hagan, 2000; Alvarez et al., 2012) or neural networks (NNs) (Guo et al., 2022; Meng and Karniadakis, 2020; Motamed, 2020). In particular, MF NNs have demonstrated significant capabilities in handling nonlinear correlation among datasets and in extracting relevant patterns. Especially when dealing with time-series data, NNs based on recurrent or attention mechanism – such as, e.g., long short-term memory networks (LSTM) and transformers, respectively – have enabled the detection of time dependencies in an automatic, data-driven fashion, and improved performance of MF methods in forecasting (Conti et al., 2023, 2024).

Among MF techniques, hierarchical approaches show enhanced flexibility in real-life scenarios, when dynamic integration of newly acquired data is essential, and allow to maintain operational robustness in the case of missing or corrupted inputs, which is advantageous in a soft sensor setting. These models adeptly navigate the balance between accuracy and efficiency, enabling fast estimates by utilizing intermediate results without fully traversing the hierarchy. Furthermore, by breaking down the construction into multiple levels, hierarchical models could potentially enhance feature extraction and pattern recognition, as opposed to processing the data in its entirety at once.

Nevertheless, the applicability of MF techniques faces challenges in scenarios that demand the processing of extensive volumes of LF signals, especially when the input hierarchy is undefined. Such requirements significantly increase model complexity and computational costs. Paired with the limited availability of HF data, this scenario may inadvertently lead to overfitting and diminished model performance.

As shown in Fig. 1, we propose the use of a hierarchical MF model as soft sensor with the integration of Bayesian Optimization (BO) during training to determine the hierarchy of the components of $\mathbf{y}_{\mathtt{M}}$, and discard components which do not improve the model quality, resulting in a subset $\mathbf{y}_{\mathtt{LF}}$, that is used for the MF method. The use of BO reduces the complexity and computational expense of the resulting MF model and avoids overfitting, similar to the approach followed by Lips et al. (2024) for system identification using BO. The variable input selection can be done fully automatically by the BO, or guided by expert knowledge, e.g., in pre-selecting components of $\mathbf{y}_{\mathtt{LF}}$ or pre-assigning components to a certain hierarchy level. Unlike methods based on Principal Component Analysis or Partial Least Squares, the latent space of the proposed method does not rely on linear combinations of components of $\mathbf{y}_{\mathtt{LF}}$ to achieve a dimensionality reduction. This provides transparency on which components of $\mathbf{y}_{\mathtt{LF}}$ are relevant for the soft sensor. It is even possible to use the model when data from the highest hierarchical levels is not available, for example in case a component of $\mathbf{y}_{\mathtt{LF}}$ is corrupted. In this case, the soft sensor can give an intermediate result, which is not affected by the corrupted component, as output.

To the best of our knowledge, both (a) the application of hierarchical MF models in a soft sensor context, and (b) the combination of hierarchical MF models with BO for hierarchy determination and input selection, have not been carried out before. The methodology is explained in Sec. 2. The method is successfully applied to two examples in Sec. 3. The first example is a benchmark problem used in MF modeling. It illustrates that our method successfully determines the correct hierarchy of the LF data, discards noisy signals and can model nonlinear, discontinuous functions. In the second example, a dynamic model is used to create a soft sensor for accurately determining the particulate matter (PM2.5) concentration in ambient air using real data collected from low-cost (LF) sensors. In Sec. 4, conclusions and outlooks (outlines) are provided.

2. METHODOLOGY

2.1 Multi-Fidelity Framework

A hierarchy of n low-fidelity input signals $\{\mathbf{y}_{\mathtt{LF}}^{(i)}\}_{i=1}^{n}$ is considered. For simplicity, we assume all input data share the same dimensionality, i.e., $\mathbf{y}_{\mathtt{LF}}^{(i)} \in \mathbb{R}^{d_{\text{in}}}$ for all $i = 1, \dots, n$. Correspondingly, a hierarchy of neural network models $\{\text{NN}_i\}_{i=1}^{n}$ is sequentially trained to estimate the high-fidelity signals $\mathbf{y}_{\mathtt{HF}} \in \mathbb{R}^{d_{\text{out}}}$, as outlined below.

The first neural network model is defined as:

$$\text{NN}_1(\cdot; \mathbf{W}_1) : \mathbf{y}_{\mathtt{LF}}^{(1)} \mapsto \text{NN}_1(\mathbf{y}_{\mathtt{LF}}^{(1)}; \mathbf{W}_1) = \hat{\mathbf{y}}_{\mathtt{HF}}^{(1)} \in \mathbb{R}^{d_{\text{out}}}, \quad (1)$$

where $\mathbf{W}_1$ denotes the network weights. These optimal weights are determined by minimizing the mean squared error (MSE) $||\hat{\mathbf{y}}_{\mathtt{HF}}^{(1)} - \mathbf{y}_{\mathtt{HF}}||_2^2$ through standard gradient-based optimization techniques. For $i = 2, \dots, n$, the subsequent neural networks are iteratively defined as:

$$\text{NN}_i(\cdot; \mathbf{W}_i) : (\mathbf{y}_{\mathtt{LF}}^{(1)}, \dots, \mathbf{y}_{\mathtt{LF}}^{(i)}) \mapsto \text{NN}_i(\mathbf{y}_{\mathtt{LF}}^{(1)}, \dots, \mathbf{y}_{\mathtt{LF}}^{(i)}; \mathbf{W}_i) = \hat{\mathbf{y}}_{\mathtt{HF}}^{(i)} \in \mathbb{R}^{d_{\text{out}}}. \quad (2)$$

Feedback connections between hidden layers of subsequent models facilitate information transfer and incremental refinement of the estimates $\{\mathbf{y}_{\mathtt{HF}}^{(i)}\}_{i=1}^{n}$ (Conti et al., 2025). The hierarchical multi-fidelity model setup is illustrated in Fig. 1. When all input signals in the hierarchy are accessible, the final estimate $\hat{\mathbf{y}}_{\mathtt{HF}}^{(n)}$ is used. Conversely, if an input signal $\mathbf{y}_{\mathtt{LF}}^{(i)}$ is missing – for reasons such as, e.g., computational limitations or signal corruption – continuity in estimation is maintained by utilizing $\hat{\mathbf{y}}_{\mathtt{HF}}^{(i-1)}$.

This flexible framework accommodates various input-output data types by selecting an appropriate neural network architecture. For instance, feed-forward networks are well-suited for parameters and vector-valued inputs, whereas LSTM networks can be used for time-series data. With reference to Fig. 1, the system inputs $\mathbf{u}$ can also be taken as part of the LF input data, entering the hierarchy. This can be achieved by augmenting the system output matrices with feedthrough of the relevant inputs.

2.2 Hierarchy Determination using Bayesian Optimization

The measurement data

$$\mathbf{y}_{\mathtt{M}} = \left[\mathbf{y}_{\mathtt{M}}^{(1)} \mid \mathbf{y}_{\mathtt{M}}^{(2)} \mid \dots \mid \mathbf{y}_{\mathtt{M}}^{(m)}\right] \in \mathbb{R}^{d_{\text{in}} \times m}$$

might contain unnecessary or redundant signals. Using these signals can result in increasing computational costs and in poorer performance of the soft sensor. Because of this, a subset of n signals is to be chosen from $\mathbf{y}_{\mathtt{M}}$. The total number of unique hierarchies drawn from an m-element set shows factorial growth. Hence, extensive grid search should not be used for finding the optimal input variables. We propose to use Bayesian optimization (see, e.g., Garnett (2023)), which – although still subject to the curse of dimensionality – often scales more favourably with problem size. BO is a so-called wrapper method, which is a method that performs variable selection during the soft sensor training and generally perform better than filter methods, which perform variable selection before the training of the soft sensor. Examples of filter methods are the Pearson Correlation Coefficient (CC) and the univariate Mutual Information (MI).

BO is an informed search algorithm that can be used to find an optimized set of hyperparameters for the training of the multi-fidelity surrogate model described in Sec. 2.1. In this context, the hyperparameters considered are the Fidelity Score vector, $\mathbf{FS} \in \Pi(\{1, \dots, m\})$, which is an element of the set of permutations Π of $\{1, \dots, m\}$ indicating a hierarchy of how reliable each signal in $\mathbf{y}_{\mathtt{M}}$ is, and the number n of LF signals to be considered. The signal with fidelity score 1 is the most reliable signal, while the signal with fidelity score m is the least reliable signal. The hyperparameters $\eta = \{\mathbf{FS}, n\}$, selected by BO, unambiguously determine the hierarchy of the LF signals to create the input vector $\mathbf{y}_{\mathtt{LF}} \in \mathbb{R}^{d_{\text{in}} \times n}$ to the multi-fidelity surrogate model. In particular, $\mathbf{y}_{\mathtt{LF}}$ is constructed by stacking the n most reliable signals in reverse order with respect to their fidelity score:

$$\mathbf{y}_{\mathtt{LF}} = \left[\mathbf{y}_{\mathtt{M}}^{(i_n)} \mid \mathbf{y}_{\mathtt{M}}^{(i_n-1)} \mid \cdots \mid \mathbf{y}_{\mathtt{M}}^{(i_1)}\right], \tag{3}$$

where i_j such that $\mathrm{FS}^{(i_j)} = j$. The inverse order accommodates the surrogate modeling requirement of processing inputs from the least to the most reliable, as this reflects the rationale that more reliable data is related to higher computational costs. The advantage of using $\mathbf{FS}$ as hyperparameters and the treatment of permutations that result in the same $\mathbf{y}_{\mathtt{LF}}$ are covered in later sections.

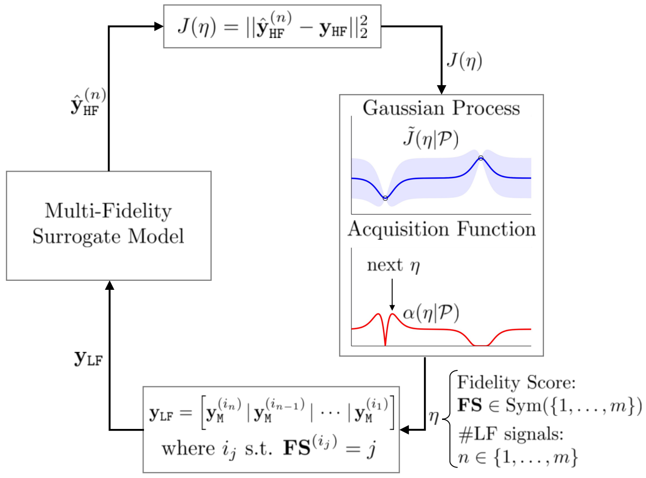

Fig. 2. Bayesian optimization workflow for determining the hierarchy and fidelity of $\mathbf{y}_{\mathtt{M}}$ to build the input, $\mathbf{y}_{\mathtt{LF}}$, to the multi-fidelity surrogate model (see Sec. 2.1).

The BO framework is shown in Fig. 2. In order to optimize the hyperparameters η, BO iteratively performs the MF surrogate model training for a given η, after which it evaluates the loss function $J(\eta)$, which describes the goodness of the final soft sensor output:

$$J(\eta) = \left\|\hat{\mathbf{y}}_{\mathtt{HF}}^{(n)}(\eta) - \mathbf{y}_{\mathtt{HF}}\right\|_2^2. \tag{4}$$

In order to avoid overfitting, this evaluation should be done using a different set of $\mathbf{y}_{\mathtt{HF}}$ than was used in the training of the MF model. Using all collected information $\mathcal{P} = \{(\eta, J(\eta))\}$, BO then proceeds to model $J(\eta)$ as a Gaussian process $\tilde{J}(\eta|\mathcal{P})$. The GP gives the expected value of the loss function for all η, including η that have not been evaluated yet, and a confidence interval, illustrated in dark and light blue, respectively, in Fig. 2. Using the GP, it is possible to choose the next η, for which a trade-off is made between exploring regions of the search space with high uncertainty and exploiting regions with low expected loss. To this end, the acquisition function α is constructed. $\alpha(\eta|\mathcal{P})$ indicates the estimated suitability of η based on the available information. The maximum of α is the next η to be observed. Garnett (2023) describes this procedure in detail.

Traditionally, continuous search spaces are used for BO. Different approaches exist to consider binary or categorical variables (Oh et al., 2019) in a BO framework. In our case, the integer parameters are first relaxed into real-valued parameters for the construction of the GP, and then are mapped back to integers before passing the GP to the acquisition function. For the optimization of α, a gradient-free optimization algorithm is used that can handle discontinuities. One of the assumptions, on which BO is based, is that evaluations of J have a certain correlation in each input dimension (Garnett, 2023). As part of our innovative approach, the particular definition of $\mathbf{FS}$ and n as hyperparameters, justifies this assumption. Indeed, it holds that similar values of $J(\eta)$ can be expected to be found for similar η, e.g., when training the MF surrogate using $\mathbf{y}_{\mathtt{LF}}$ defined by the same $\mathbf{FS}$ but different n, or defined by different $\mathbf{FS}$ in which the signal j has similar fidelity score $\mathbf{FS}^{(j)}$.

Some modifications are used to speed up the BO workflow. Because the training of the MF model is sequential, the MF surrogate training using $\eta = (\mathbf{FS}, n)$ includes the implicit observation of $J(\eta')$ with $\eta' = (\mathbf{FS}, n')$ for $n' \in \{1, ..., n-1\}$. These intermediate results are also included in the set of past observations $\mathcal{P}$. Also, $J(\eta)$ shows functional equality for different η for which the resulting $\mathbf{y}_{\text{LF}}$ is the same. Therefore, before creating the new GP, $\mathcal{P}$ is expanded with all η that were implicitly observed because of this functional equality.

3. APPLICATION EXAMPLES AND DISCUSSION

Two test cases are presented. The multi-fidelity neural network models for these cases are constructed using TensorFlow's Keras API, with dense layers for the first test case and LSTM layers for the other. Training is performed using the ADAM optimizer with L2 regularization. The BO uses the ARD Matern 5/2 kernel for the GP and the Expected Improvement acquisition function and is implemented in MATLAB (Garnett, 2023).

3.1 Benchmark Example

As a first test case, we consider a standard MF benchmark (Meng and Karniadakis (2020)), which consists in estimating the output of the discontinuous scalar function from a set of low-fidelity inputs. The output f_{HF} simulates the process variable we aim to estimate and is defined as a linear combination of two signals $f_{\text{M}}^{(1)}$ and $f_{\text{M}}^{(2)}$:

$$f_{\text{HF}}(x) = \frac{5}{3} f_{\text{M}}^{(1)}(x) + 2 f_{\text{M}}^{(2)}(x) + \frac{110}{3} + 4H(x - \frac{1}{2}),$$

where $H(x)$ is the Heaviside function, which introduces discontinuities. In order to test the input feature selection capabilities of the algorithm, we included the inputs $f_{\text{M}}^{(3)}, f_{\text{M}}^{(4)}, f_{\text{M}}^{(5)}$, which are disturbed versions of $f_{\text{M}}^{(2)}$, as well as $f_{\text{M}}^{(6)}$, which is pure noise. The set of available low-fidelity inputs (here with subscript M for 'measured') is illustrated in Fig. 3. See Appendix A for mathematical definition of $f_{\text{M}}^{(i)}$, $i = 1, \ldots, 6$. We employ our method to automatically select the relevant input signals and simultaneously construct a hierarchical MF surrogate model as a soft sensor estimating the HF function. To this end, we consider $N_{\text{MF-train}} = N_{\text{BO-train}} = 7$ equidistant locations for MF- and BO-training data sampling for each fidelity level (see Fig. 3). Standard feed-forward networks with dense layers are employed in the MF model to approximate the scalar outputs of the one-dimensional functions ($d_{\text{in}} = d_{\text{out}} = 1$).

Results After 25 iterations, our proposed method successfully detects the signals $f_{\text{M}}^{(2)}$ and $f_{\text{M}}^{(1)}$ to be relevant, which are indeed the components defining f_{HF}. Note that, because of the implicit observations of intermediate models for each iteration, after 25 iterations, up to $6 \cdot 25 = 150$ unique hierarchies have been observed, which is about 7.7 % of the total number of possible hierarchies. The method proves robustness by discarding all the remaining noisy and/or uncorrelated inputs. These results cannot be achieved by using input selection based on the CC and MI methods, as is shown in Appendix A.

The relevant signals, $f_{\text{M}}^{(2)}$ and $f_{\text{M}}^{(1)}$ are used in sequence by our method to construct the hierarchical MF soft sensor

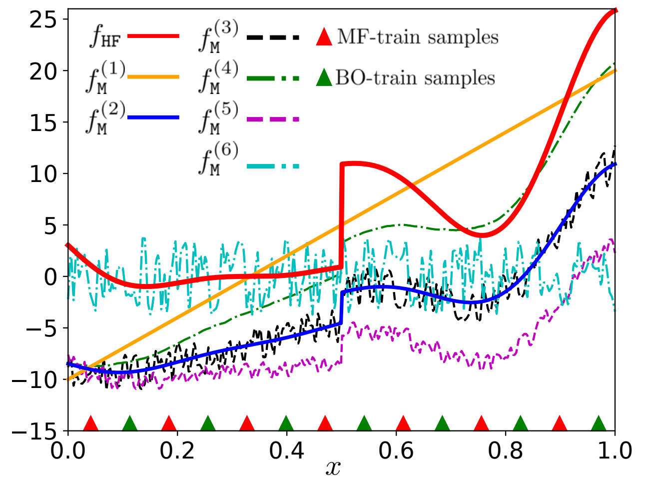

Fig. 3. Low-fidelity input and high-fidelity output functions considered in the benchmark test case. The markers on the x-axis indicate the sampling locations of MF- and BO-training data.

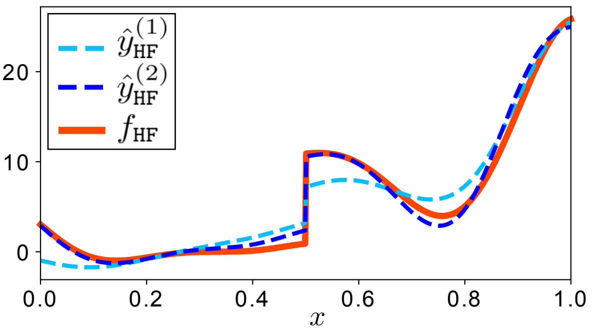

Fig. 4. Comparison of the outputs of the MF surrogate model $\hat{\mathbf{y}}_{\text{HF}}^{(1)}, \hat{\mathbf{y}}_{\text{HF}}^{(2)}$ with respect to the reference f_{HF}.

to estimate f_{HF}. Fig. 4 and Table 1 show the surrogate performance (for both levels in the hierarchy) compared to the reference HF solution, on a test set consisting of $N_{\text{test}} = 100$ equidistant data points.

3.2 Air Quality Monitoring Example

As a second example, we consider an air quality monitoring problem. Because of the negative effects of particulate matter (PM) on human health, PM monitoring is widespread. A distinction is made between PM10, which are coarse particles with a diameter of up to 10 µm, and PM2.5, which are fine particles with a diameter of up to 2.5 µm. The use of low-cost sensors for air quality measurements has gained increased attention in the last decades as they would allow to significantly increase PM monitoring coverage. However, these sensors are associated with lower fidelity than expensive devices. For example, high relative humidity is known to have a negative effect

Table 1. RMSE of the hierarchical soft sensor model for the benchmark test case.

Model	MF-Training	BO-Training	Test
$\hat{y}_{\text{HF}}^{(1)}$	0.058	0.062	0.063
$\hat{y}_{\text{HF}}^{(2)}$	**0.017**	**0.038**	**0.030**

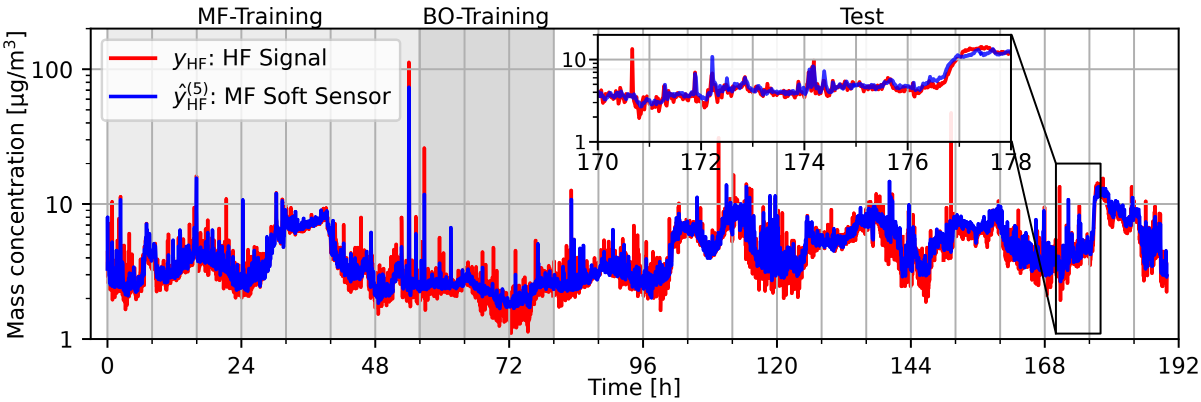

Fig. 5. PM2.5 high-fidelity data and the results from the trained soft sensor.

on the accuracy of PM measurement devices, and is not adequately compensated for by low-cost sensors.

Air quality data from three measurement devices positioned at a single location in Stuttgart, Germany, was kindly provided by Chacón-Mateos et al. (2022). The data is available in 60 s resolution for 190 consecutive hours. Two of the devices, 'LF-Device 1' and 'LF-Device 2', are different low-cost (LF) devices, the third one is a high-cost (HF) reference measurement device, 'HF-Device'. LF-Device 1 is a low-cost sensor of the type *Nova Fitness SDS011*. LF-Device 2 was a custom device, built by the authors, which provided additional LF-measurement data.

For this example, we want to estimate the PM2.5 concentration measured with the HF-Device. A potentially relevant subset of the LF-measurements is pre-selected as available LF inputs for the model training. An overview of considered HF and LF signals is given in Tab. 2.

We employ our method using an LSTM-based NN for the multi-fidelity model to automatically rank and select the relevant input signals and simultaneously construct a dynamic, hierarchical MF model for the soft sensor, estimating the HF signal. The time signal $y^{(1)}_{\mathrm{LF}}$ is pre-selected as input for the first level. We use time-series of 56 h of MF-training data and 24 h of BO-training data.

Results After 25 iterations, the best model found by our proposed method utilizes 5 out of the 8 input signals. The selected input signals and their level hierarchy are given in the last column of Tab. 2. The performance of

Table 2. Low- and High-Fidelity signals for the air quality example, and the input hierarchy of the selected signals for the MF model.

Var.	Description	Hier.
$y^{(1)}_{\mathrm{M}}$	Time [s]	1
$y^{(2)}_{\mathrm{M}}$	Ambient relative humidity (LF-Device 2) [%]	3
$y^{(3)}_{\mathrm{M}}$	Ambient temperature (LF-Device 2) [°C]	-
$y^{(4)}_{\mathrm{M}}$	PM2.5 mass concentration (LF-Device 1) [µg/m³]	4
$y^{(5)}_{\mathrm{M}}$	PM2.5 mass concentration (LF-Device 2) [µg/m³]	2
$y^{(6)}_{\mathrm{M}}$	Internal Relative humidity (LF-Device 2) [%]	-
$y^{(7)}_{\mathrm{M}}$	Internal temperature (LF-Device 2) [°C]	5
$y^{(8)}_{\mathrm{M}}$	Baseline Particle Count (LF-Device 2) [-]	-
y_{HF}	HF PM2.5 mass concentration (HF-Device) [µg/m³]	

Table 3. RMSE [µg/m³] of hierarchical soft sensor models and LF PM2.5 measurements.

Model	MF-Training	BO-Training	Test
$\hat{y}^{(1)}_{\mathrm{HF}}$	2.854	1.990	2.363
$\hat{y}^{(2)}_{\mathrm{HF}}$	1.994	0.762	2.020
$\hat{y}^{(3)}_{\mathrm{HF}}$	1.714	0.700	1.633
$\hat{y}^{(4)}_{\mathrm{HF}}$	1.174	0.541	1.059
$\hat{y}^{(5)}_{\mathrm{HF}}$	**0.831**	**0.508**	**1.028**
$y^{(4)}_{\mathrm{M}}$	1.030	0.642	1.340
$y^{(5)}_{\mathrm{M}}$	1.464	0.623	2.337

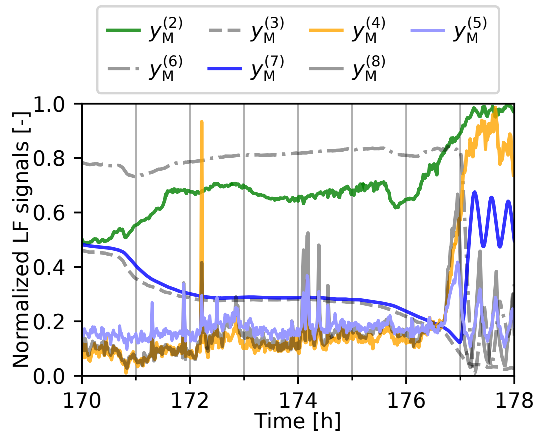

Fig. 6. Normalized inputs for a representative section of the test data. Coloured signals are used by the MF soft sensor $\hat{\mathbf{y}}^{(5)}_{\mathrm{HF}}$.

the final model, $\hat{y}^{(5)}_{\mathrm{HF}}$, as well as the intermediate models, $\hat{y}^{(i)}_{\mathrm{HF}}$ ($i \in \{1, 2, 3, 4\}$), was evaluated on a test time-series of 110 h. The RMSE of the hierarchical models is given in Tab. 3. The model performance systematically increases with increasing hierarchical level. The final model outperforms both LF-sensors on MF-training, BO-training and test set. In case LF-Device 1 (signal $y^{(4)}_{\mathrm{LF}}$) is not available, model $\hat{y}^{(3)}_{\mathrm{HF}}$ can be used, which uses only signals from LF-Device 2 and outperforms the PM2.5 measurement delivered by LF-Device 2 ($y^{(5)}_{\mathrm{LF}}$) for all datasets.

The estimates of the $\hat{y}^{(5)}_{\mathrm{HF}}$ soft sensor are shown together with the HF data in Fig. 5, showing that the MF soft

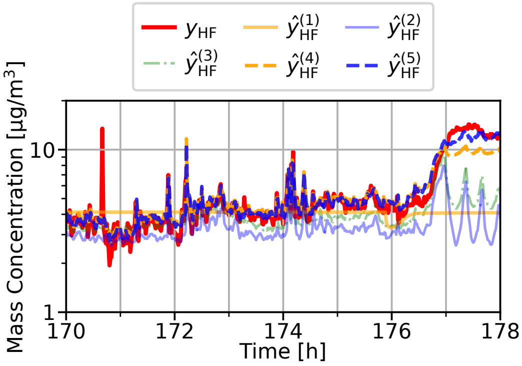

Fig. 7. Comparison of the outputs of the MF soft sensor model at the different hierarchical levels with the reference signal y_{HF} for a section of the test data.

sensor effectively captures both rapid and gradual signal variations. The performance of the soft sensor remains consistent, even when making forecasts significantly beyond the training time window. Fig. 5 comprises a magnification of a representative section of 8 h of test data, for which the normalized input signals are shown in Fig. 6 and the corresponding results of the different hierarchical models are shown in Fig. 7. The final output $\hat{y}_{\text{HF}}^{(5)}$ is computed using all the selected relevant signals, i.e., $y_{\text{LF}} = (y_{\text{M}}^{(1)}, y_{\text{M}}^{(5)}, y_{\text{M}}^{(2)}, y_{\text{M}}^{(4)}, y_{\text{M}}^{(7)})$. The intermediate output $\hat{\mathbf{y}}_{\text{HF}}^{(i)}$ uses the first i signals from this tuple. The increasing accuracy of the models with increasing number of LF signals used is visible.

4. CONCLUSION AND OUTLOOK

Using hierarchical multi-fidelity models with Bayesian optimization to select and rank relevant features can be effectively used for soft sensor design. The method was introduced and successfully applied on both a nonlinear benchmark and a real-data test case, in which a dynamic model was created using LSTM neural networks.

In future work, we will aim at reducing the computational complexity of the method, e.g., by using different search space design or alternative informed search algorithms. It is possible to apply our method to signals featuring diverse sampling frequencies, which can be handled using a MF approach (Guo et al., 2022). Using the LSTM networks, the method can also be applied for system identification of overactuated systems, as in Lips et al. (2024). Finally, data assimilation techniques could be integrated to monitor and enhance the reliability of the soft sensor over prolonged operations by preventing sensor drift and refining the multi-fidelity model within a digital twin framework, e.g., for predictive maintenance or process optimization.

CODE AND APPENDIX ACCESSIBILITY

The source code of the proposed method and the benchmark example are available as GitHub repository on github.com/ContiPaolo/MultiFidelityBO_SoftSensor, together with Appendix A.

ACKNOWLEDGMENT

PC acknowledges the PRIN Project UQ-FLY (202222PACR).

REFERENCES

Alvarez, M.A., Rosasco, L., Lawrence, N.D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3), 195–266.

Chacón-Mateos, M., Laquai, B., Vogt, U., and Stubenrauch, C. (2022). Evaluation of a low-cost dryer for a low-cost optical particle counter. *Atmospheric Measurement Techniques*, 15(24), 7395–7410.

Conti, P., Guo, M., Frangi, A., and Manzoni, A. (2025). Progressive multi-fidelity learning. URL `www.paolo-conti.com/publication/progressiveMF`. Unpublished.

Conti, P., Guo, M., Manzoni, A., Frangi, A., Brunton, S.L., and Nathan Kutz, J. (2024). Multi-fidelity reduced-order surrogate modelling. *Proceedings of the Royal Society A*, 480(2283), 20230655.

Conti, P., Guo, M., Manzoni, A., and Hesthaven, J.S. (2023). Multi-fidelity surrogate modeling using long short-term memory networks. *Computer Methods in Applied Mechanics and Engineering*, 404, 115811.

Garnett, R. (2023). *Bayesian Optimization*. Cambridge University Press.

Guo, M., Manzoni, A., Amendt, M., Conti, P., and Hesthaven, J.S. (2022). Multi-fidelity regression using artificial neural networks: Efficient approximation of parameter-dependent output quantities. *Computer Methods in Applied Mechanics and Engineering*, 389, 114378.

Kennedy, M.C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1), 1–13.

Kim, S., Kano, M., Hasebe, S., Takinami, A., and Seki, T. (2013). Long-Term Industrial Applications of Inferential Control Based on Just-In-Time Soft-Sensors: Economical Impact and Challenges. *Industrial & Engineering Chemistry Research*, 52(35), 12346–12356.

Lawrence, N.P., Damarla, S.K., Kim, J.W., Tulsyan, A., Amjad, F., Wang, K., Chachuat, B., Lee, J.M., Huang, B., and Bhushan Gopaluni, R. (2024). Machine learning for industrial sensing and control: A survey and practical perspective. *Control Engineering Practice*, 145, 105841.

Lips, J., DeYoung, S., Schoensteiner, M., and Lens, H. (2024). Closed-loop identification of a MSW grate incinerator using bayesian optimization for selecting model inputs and structure. *Control Engineering Practice*, 153(106075).

Meng, X. and Karniadakis, G.E. (2020). A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 401, 109020.

Motamed, M. (2020). A multi-fidelity neural network surrogate sampling method for uncertainty quantification.

Oh, C., Tomczak, J., Gavves, E., and Welling, M. (2019). Combinatorial bayesian optimization using the graph cartesian product. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*.

Souza, F.A.A., Araújo, R., and Mendes, J. (2016). Review of soft sensor methods for regression applications. *Chemometrics and Intelligent Laboratory Systems*, 152, 69–79.