AMUSENS

D6.2

Algorithm for the selection of most promising materials by AI

Project number	101130159
Project acronym	AMUSENS
Project title	Adaptable multi-pixel gas sensor platform for a wide range of appliance and consumer markets
Start date of the project	1st June, 2024
Duration	48 months
Call	HORIZON-CL4-2023-RESILIENCE-01-33

Deliverable type	Other
Deliverable reference number	CL4-2023-RESILIENCE-01-33 / D6.2 / v1.0
Work package contributing to the deliverable	WP6
Due date	11 2025 – M18
Actual submission date	18/12/2025

Responsible organisation	IMT
Editor	Luiz MIRANDA
Dissemination level	PU
Revision	v1.0

Abstract	This deliverable describes a software tool developed to extract AI-ready datasets from raw sensor data, apply feature extraction and selection, and identify the optimal combination of sensing materials and their operating parameters. The software is publicly available in the project's code repository.
Keywords	AI algorithm; material selection; project dataset; feature selection.

Funded by the European Union under grant agreement no: 101130159. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Editor
Luiz MIRANDA

Contributors (ordered according to beneficiary numbers)
All partners

Disclaimer

Executive Summary

Deliverable 6.2 presents the outcomes of Task 6.4, which focuses on developing AI-based tools to identify the most promising combinations of sensing materials and operating parameters for the AMUSENS multi-pixel gas sensor platform. The main objective of this work was to develop a set of tools capable of selecting optimal combinations of sensing materials and their operating parameters in order to maximize the ability to discriminate and detect individual target gases. Building on the database produced in Task 6.3, this deliverable introduces a complete methodological and software pipeline capable of transforming long raw time-series measurements into an AI-ready dataset, extracting meaningful features, and applying feature-selection strategies to rank sensing materials according to their contribution to gas discriminability.

The deliverable details the structure and characteristics of the existing dataset, the procedures used to segment raw time-series into individual measurements, and the extraction of relative response features widely adopted in gas-sensor research. It then outlines the feature-selection approaches explored; filter, wrapper, and embedded methods; and presents the baseline filter-based algorithm implemented in the current version of the software. This method evaluates all possible combinations of sensing materials and selects those that maximize class separability using centroid-distance criteria.

The resulting software tool, delivered with a graphical user interface, integrates the entire processing workflow, including file import, data structuring, feature computation, and material-selection optimization, as well as optional inspection and debugging utilities. Additional functionalities, such as automated time-series visualization and manual correction of sensing-material metadata, enhance usability for both experts and non-experts. The final output consists of a ranked list of sensing materials and optimal combinations tailored to either specific target-gas subsets or full-gas scenarios.

Together, these developments provide the first operational version of the AMUSENS AI-based material-selection tool and establish a methodological foundation for future refinements as the project database expands and complementary selection strategies are incorporated.

Table of Content

List of Figures

List of Tables

Chapter 1 Introduction

The AMUSENS project aims to develop a multi-pixel gas sensor, based on metal-oxide gas sensing technology, as an adaptable gas-sensing platform for several applications (personal exposure and breath analysis). The selection of the most promising sensing materials for the multi-pixel platform is part of **WP6** ***"Phase 1: single sensor development – testing"***, where the response of the shortlisted sensing materials and coatings to a selection of target gases (WP4, D4.1) are extracted and characterized in order to comprehend their responsiveness. The results of such characterization, performed during **Task 6.3** ***"Production of database of sensing materials"***, are included in the database that was used during **Task 6.4** ***"AI for the selection of best material combination based on the database"***, which was the entry for the work presented in this report.

1.1 Objective

The main objective was to develop a set of tools to select the optimal combinations of sensing materials and its operating parameters in order to maximize the resulting capabilities of detecting the individual target gases.

1.2 Currently available data

The available dataset from **Task 6.3** includes the response of 6 sensing materials (referred as nanostructures in **D4.1**) in different operating conditions:

- as synthetized;
- combined with three ALD coating materials (two numbers of ALD cycles);
- three operating temperatures (200 °C, 300 °C and 400 °C);
- two different bias voltages (1 V and 10 V);
- two levels of relative humidity each (0 % and 50 %).

Hereafter, a ***Sensor*** (italicized) refers to a specific combination of raw material, coating, operating temperature, and bias voltage, whereas a ***Gas*** (italicized) refers to a target gas at a given relative humidity.

During each set of exposition (according to the experimental protocol proposed in **Task 4.3**) most *Sensors* have two copies being exposed at the same time, providing at least two co-occurring responses, or samples, per *Sensor*. Therefore, according to the set of variables described, the resulting data should provide two independent responses for each of the 6 (raw) x 4 (no-coating + 3 individual coating materials) x 2 (number of ALD cycles) x 3 (operating temperatures) x 2 (bias voltage) = 288 individual *Sensors* to each one of the 12 individual *Gases*, accounting for 288 x 12 = 3456 x 2 (copies) = 6912 unique measuring curves (time-series). Although, the actual dataset was smaller (4120 unique time-series) due to experimental issues.

Table 1. Illustration on how the dataset was organized after processing.

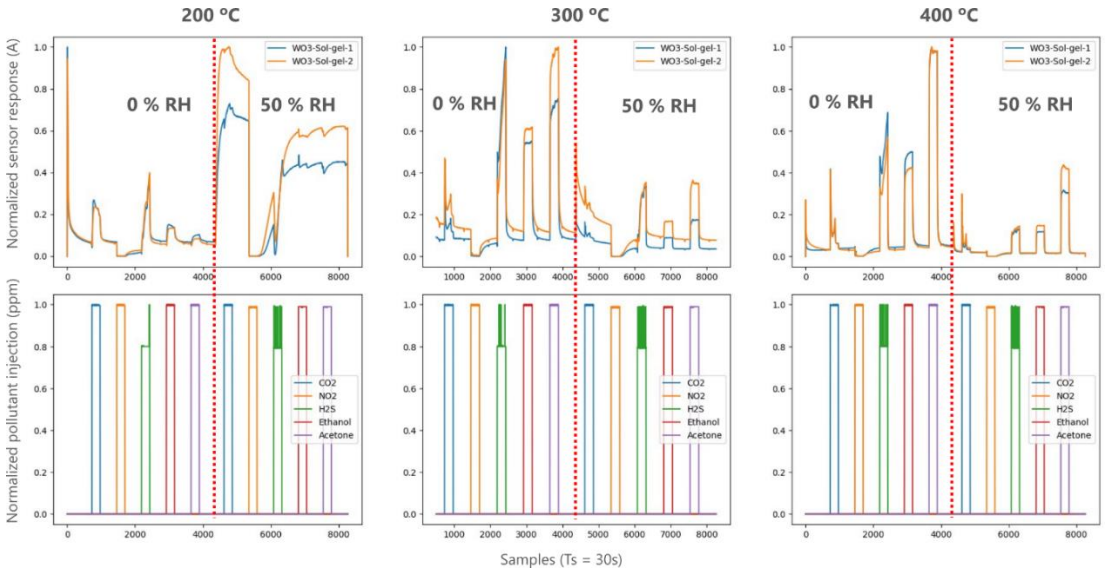
Gas-RH	Co2O3 @ 200 °C (1V)	Co2O3 @ 200 °C (10V)	Co2O3 @ 300 °C (1V)	Co2O3 @ 300 °C (10V)	Co2O3 @ 400 °C (1V)	Co2O3 @ 400 °C (10V)	Co2O3 (Al2O3 35C) @ 200 °C (1V)	…	WO3 (SnO2 50C) @ 400 °C (10V)	WO3 (SnO2 75C) @ 400 °C (1V)	WO3 (SnO2 75C) @ 400 °C (10V)
Acetone (0%)	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies
Acetone (50%)	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies
…											
CO2 (0%)	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies

CO2 (50%	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies
H2S (0%)	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies
H2S (50%)	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies	2 copies		2 copies	2 copies	2 copies

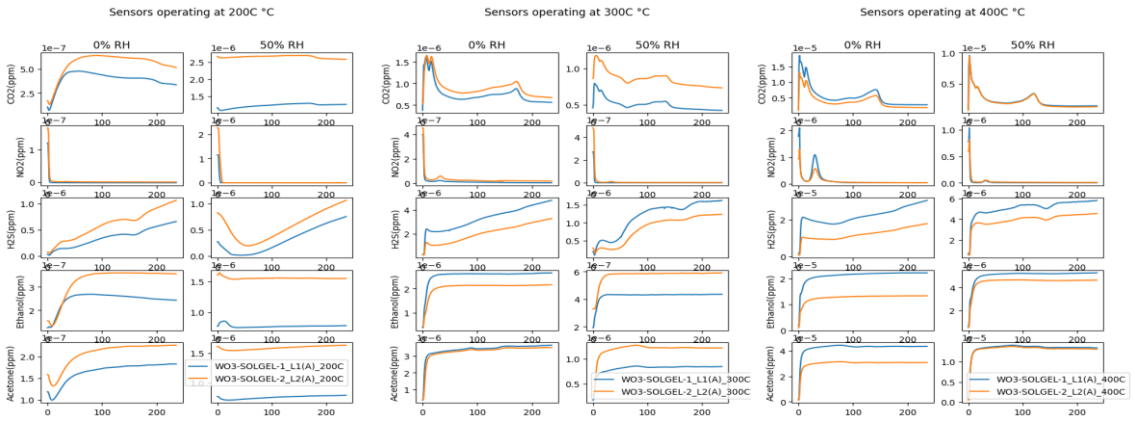
To be able to use this data with an AI algorithm, we organized the data in a machine learning standard format: a table in which the rows correspond to measurements of the same phenomenon (individual *Gases*) and the columns are the features used for the measurements (individual *Sensors*), as illustrated in Table 1. It is from this organization of data that the individual *Sensors* are going to be studied by the AI algorithm, which will select the 4 columns that allows the best segregation.

1.3 Data treatment and feature extraction

Despite the description of the organized dataset, the raw measurements are provided by the partners as a continuous time-series, which are representative of several measurements chained in an individual experiment (i.e. the exposition of the *Sensors* to, for example, Acetone with 50% RH, then H_2S at 50% RH and so on). Following the protocol from **Task 4.3**, the whole experiment is performed in a single run, taking up to 10.5 days of continuous measurements. Additionally, the signal indicating when the individual gases are injected is provided with the continuous measurements. The continuous curves, containing all the measurements, must be segmented into individual curves containing single measurement, as illustrated in Figure 1.



(a)



(b)

Figure 1. Illustration of the segmentation process. (a) Summary of the measurements during three experiments for a *Sensor* at three operating temperatures with indication of moment of injection for each target gas; (b) Result of segmentation during injection of each gas.

In this context, each individual measurement is a time-series containing the response of a *Sensor* to a *Gas*. Their shape (Figure 1(b)) is, in general, similar to the expected response curve (Figure 2).

Although the whole curve can be a useful source of information, it is common practice to extract a feature representative of the whole time-series: the relative response R_s. This is done by using the equation 1.1 when $R_g > R_0$ and equation 1.2 when $R_0 > R_g$ (such as the measurement for NO2 in Figure 1(b)), where R_0 is the *Sensor*'s resistance before the injection of the target gas, and R_g is the maximum absolute response of the *Sensor* during injection, as illustrated in Figure 2.

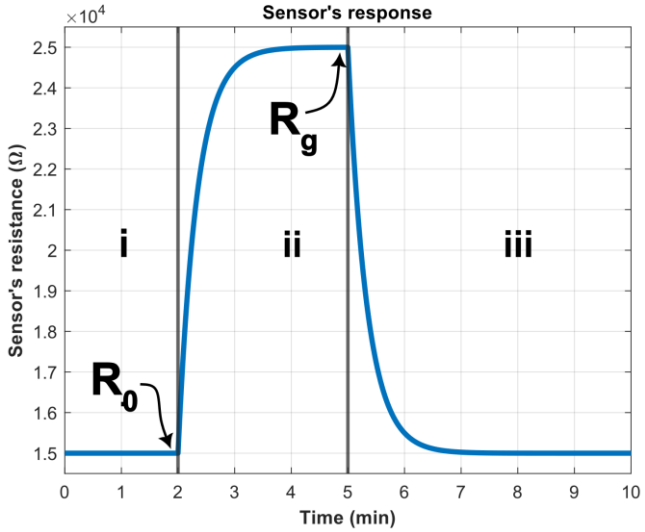


Figure 2. Definitions of R_0 and R_g. R_0 is the *Sensor*'s resistance instants before the injection of the target gas, while R_g is the maximum absolute value of the *Sensor*'s response during injection of the target gas. Regions i, ii and iii correspond to before the injection, during the injection and after the injection of the target gas, respectively.

$$R_s = \frac{R_g - R_0}{R_0} \quad 1.1$$

$$R_s = \frac{R_0 - R_g}{R_g} \quad 1.2$$

We can get the R_s value for each *Sensor-Gas*, by extracting the corresponding R_g and R_0 from the experimental curves. The resulting dataset table is composed of 288 columns containing all R_s for a single repetition of the experiment. The real number of columns during the development of this work was lower than that due to aforementioned experimental issues, totaling only 103 columns.

Chapter 2 Selection of the optimal combination

The selection phase for identifying the best combination of sensing materials was divided into two steps:

- exploring several feature-selection techniques, and
- evaluating their ability to identify the most relevant inputs for a given model.

Due to limitations of the currently available dataset, specifically the limited number of samples (at most two per class), the investigation and comparison of feature-selection techniques were performed using an open dataset published by Vergara et al. (2013) [1].

Feature-selection methods are commonly grouped into three categories: filter, wrapper, and embedded approaches. Filter approaches select features prior to model training by applying statistical or information-theoretic criteria to identify correlations and redundancies among inputs. Wrapper approaches evaluate multiple subsets of features by training a model and selecting those that yield the best performance depending on the application. Embedded approaches determine feature importance during model training, either through imposed constraints or through the intrinsic structure of the model, such as in tree-based methods [2].

Each approach presents specific advantages and limitations. Filter methods are fast, easy to interpret, and model-agnostic, but they ignore feature interactions and may lead to suboptimal selections. Wrapper methods typically identify highly performant feature combinations and offer greater model flexibility, but they are computationally expensive and more prone to overfitting. Embedded methods are computationally efficient, as feature selection is performed during a single training process; however, they are strongly model-dependent and may require careful parameter tuning, often resulting in slightly lower performance compared to wrapper approaches [2].

Based on these considerations, one method from each category was proposed and implemented for Task 6.4. After selecting sensor combinations using the Vergara dataset [1] and evaluating their performance through gas-classification experiments, classification accuracies of 83 % for the filter approach, 92 % for the wrapper approach, and 91 % for the embedded approach were obtained. These same methods were subsequently applied to the AMUSENS project dataset; however, due to the limited number of available samples, the resulting combinations and performance metrics may not fully reflect real-world behavior and should be interpreted as indicative rather than definitive.

The implemented methods are available in the project repository as callable scripts. Their detailed description is reserved for a follow-up publication, with the exception of the baseline method described in the following section.

2.1 Baseline method

Filter approach is considered as the simplest approach, since it does not require any training step. Accordingly, we considered it as the baseline approach. As the typical feature for the response of gas *Sensor*s is the relative response, and because the objective is to maximize class (individual *Gases*) separability, the proposed method computes the average measurement for each class using data from all available *Sensors*, and stores it as the centroid of each class. The proposed method then: (i) selects a subset of N *Sensors* (for this project N=4) among the available ***S*** *Sensors*, then (ii) computes the average distances between the centroids and stores them, finally, (iii) repeat steps (i) and (ii) for all possible combinations of N *Sensors* in ***S***. The best combination of sensing materials is the one that maximizes the average distance between the centroids. The final rank of *Sensor* importance is obtained by calculating the importance of *Sensor* ***s*** (I_s) through equation 2.1, where ***T*** is the number of times ***s*** appears in the top 10% of sorted combinations, and $\boldsymbol{D_k}$ is the *k-th* average centroid distances in which ***s*** appears.

$$I_s = \sum_{k=1}^{T} D_k \qquad 2.1$$

This method is integrated in the publicly available GUI software, described in Chapter 3.

Chapter 3 The resulting software tools

As the main output of this deliverable, a set of software tools was developed and made available at https://git.list.lu/amusens/D6.2.

The repository contains a main script providing a graphical user interface (GUI), which integrates several utilities for selecting optimal combinations of sensing materials, as well as auxiliary scripts implementing additional feature-selection approaches that are not yet fully integrated into the GUI. The codebase is intended to be continuously updated as additional data become available for model training and validation.

The GUI file (GUI_D6.2.py) implements the complete data-processing pipeline starting from raw measurement files. This includes extraction of full measurement time series, segmentation into individual gas-exposure events, and construction of an AI-ready dataset in which rows correspond to individual *Gas* and columns correspond to individual *Sensor* formulations. From this dataset, the relative *Sensor* response (R_s) is computed, and the optimization of sensing-material combinations is performed using the baseline method described in Section 2.1. The optimization can be carried out either using all available gases or a user-defined subset, depending on the intended application.

An overview of the GUI is shown in Figure 3. A typical usage workflow is described in Section 3.1 and further detailed in the repository's README file.

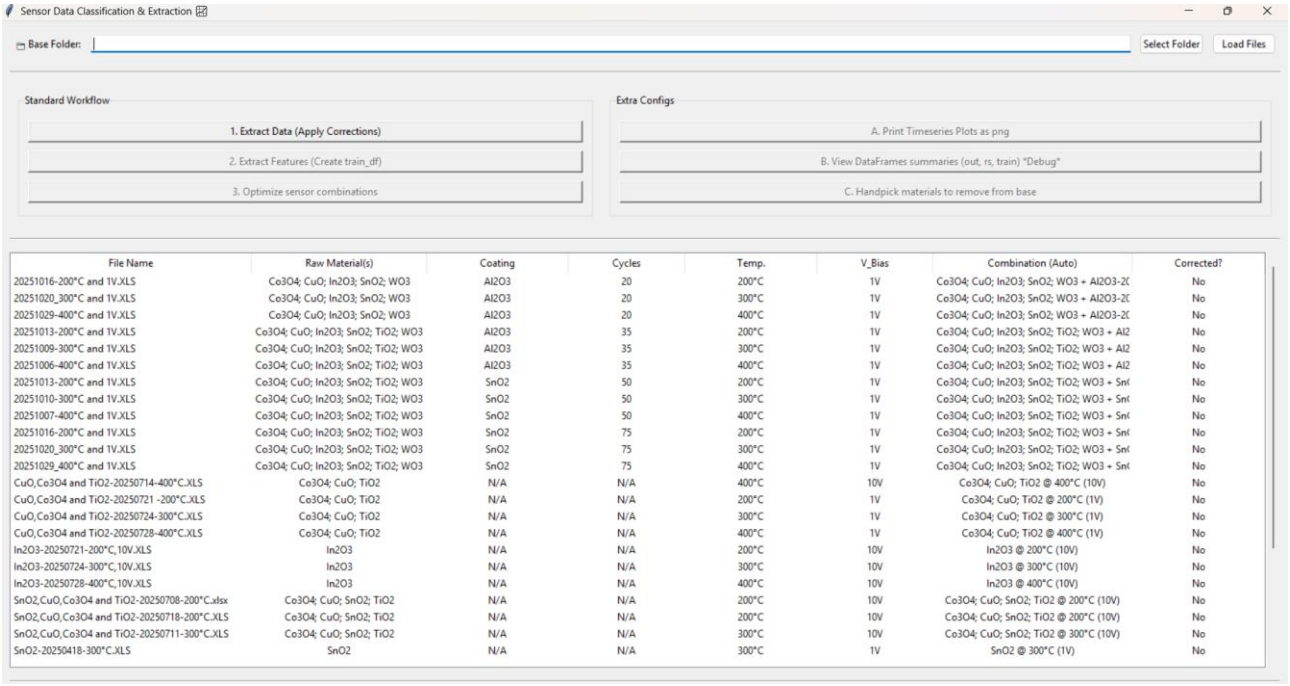


Figure 3. Overview of the GUI

3.1 Standard workflow

To perform sensing-material selection using the GUI, raw measurement files must first be placed in a common directory. This directory is selected using the "Select folder" button in the interface and loaded by pressing the "Load Files" button. Measurements corresponding to raw materials and ALD-coated materials must be stored in separate folders, preferably identified using the suffix _ALD (or -ALD), as illustrated in Figure 4.

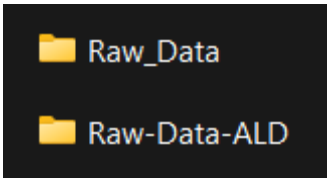


Figure 4. Illustration of the naming of folders for Raw materials and ALD coated

Once the files are loaded and processed, a summary of the imported data is displayed in the window below the control buttons. This summary allows the user to verify whether *Sensor* characteristics, such as base material, coating material, number of ALD cycles, operating temperature, and bias voltage, have been correctly extracted. If any import errors are identified, they can be corrected by double-clicking the corresponding row and entering the correct information in the correction window (Figure 5). Changes are confirmed by pressing the “Save corrections” button.

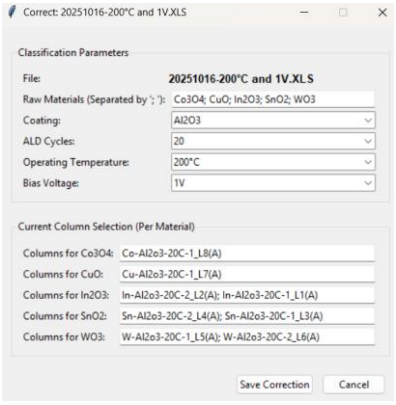


Figure 5. Correction window example.

After validation and correction, the user presses the “Extract data (apply corrections)” button. This step loads the processed data into memory and constructs the AI-ready dataset. Completion of this step is indicated by a pop-up window reporting the number of individual measurement curves extracted.

Feature extraction is then performed by pressing the “Extract Features (Create train_df)” button. This step computes the relative response for each *Sensor* and generates the final dataset used for optimization. The operation is typically fast and concludes with a prompt displaying the dataset dimensions (number of rows and columns).

The final step consists of launching the *Sensor*-selection procedure by pressing the “Optimize sensor combination” button. The user is prompted to select the gases of interest; if all gases are considered target gases, all options must be selected. The code then checks the dataset for potential issues, such as the presence of NaN or infinite values. If such values are detected, the user is prompted to either (i) remove the affected *Sensor* combinations or (ii) replace the problematic values with zeros.

Once the dataset is validated, the software generates a ranking of *Sensor* importance and identifies the optimal combination of *Sensor*s, following the procedure described in Section 2.1. In addition, the tool evaluates alternative combinations by testing the top five combinations formed from the 20 most important *Sensor*s, ranked according to classification performance. This restriction is introduced to avoid evaluating all possible *Sensor* combinations, which can become computationally prohibitive for large *Sensor* pools. The final output is a set of recommended sensing-material combinations tailored to the selected application.

The remaining scripts present in the repository are going to be integrated to the GUI as the project continues and the corresponding research paper is published.

In addition to the standard workflow, some additional tools have been implemented in the GUI, which are described in section 3.2.

3.2 Additional features

3.2.1 Export time-series plots as png

During construction of the AI-ready dataset, individual *Sensor* responses are extracted for all sensing-material combinations. These time-series signals can be exported as a single PNG image to provide a global overview of *Sensor* behavior during experiments. The resulting images are arranged as tables, with rows corresponding to different *Gases* and columns corresponding to different *Sensors*. An example is shown in Figure 6.

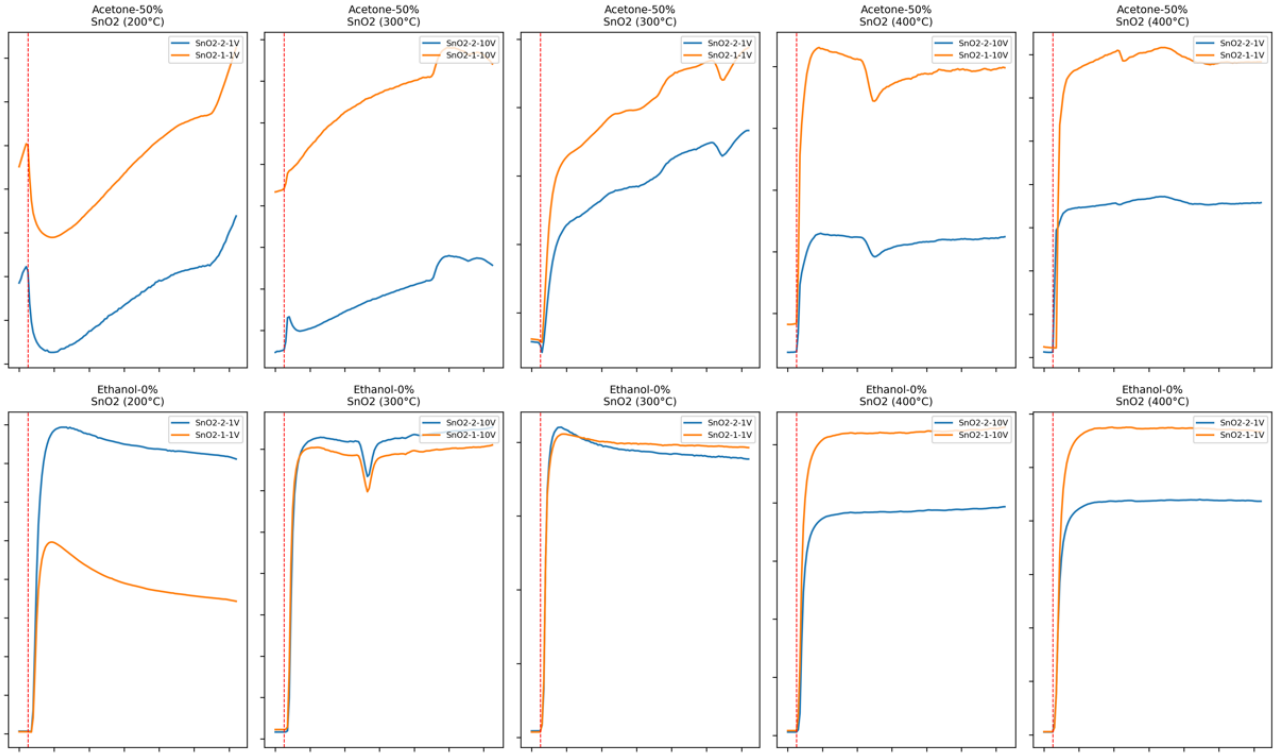


Figure 6. Illustration of the time-series plots

3.2.2 View dataframe summaries

If the users notice any inconsistent behavior on the *Sensor* optimization step and if they also have some knowledge on the use of dataframes in Python, they can try to debug the usage of the GUI by inspecting the summary of the dataframes being computed during the data processing. This can be viewed by pressing the “View dataframe summaries” button.

3.2.3 Handpick materials to be removed

If the correction step does not fully resolve data issues, users may manually remove specific sensing-material combinations using the “Handpick materials to remove from base” option. This allows problematic *Sensor*s to be excluded without reloading the raw files. After removal, feature extraction must be repeated, although this step is significantly faster than re-importing the full dataset.

3.3 Callable scripts for the three feature selection approaches

In addition to the GUI, which implements the baseline feature selection method described in section 2.1, the repository includes three standalone scripts corresponding to the three feature selection methods mentioned in Chapter 2. These scripts can be used independently by importing them into custom Python workflows.

3.3.1 Filter approach script (sensor_selection_filter.py)

This script implements the same method implemented in the GUI. Its input arguments are:

- Train_df (pd.DataFrame): DataFrame with *Sensor* features in columns (should include columns named in sensor_names) and a 'label' column. It is the relative responses table;
- sensor_names (List[str]): List of all available *Sensors* feature columns to consider for combinations.
- num_sensors_to_select (int): Number of *Sensor*s in each combination. Optional, default is 4 *Sensors*;

The script then selects the *Sensors* as described in section 2.1, prints the best combination and returns the ranking of most important *Sensors*.

3.3.2 Wrapper approach (sensor_selection_wrapper.py)

This script implements the proposed wrapper approach. It takes as input arguments:

- X_scaled (np.ndarray): 3D array (*samples, timesteps, features*). *Timesteps* are segments extracted from the time-series curves illustrated in Figure 6, using a sliding window of observation with length shorter than the time-series (in our tests, window length = 6, and step = 2). *Features* are the individual *Sensors* being tested. And *samples* are the individual instances of the shorter segments in *timesteps* measured by the *Sensors* in *features*.
- Y (np.ndarray): 1D array of integer labels containing the labels from which *Gas* a given *sample* came from. Len(Y) = number of *samples* in X_scaled.
- sensor_names (List[str]): List of all available *Sensor* feature names.
- num_sensors_to_select (int): Number of *Sensors* in each combination. Optional, default is 4 S*ensors*.

For each candidate subset of *Sensor*s, the script trains a neural network composed of an LSTM layer followed by a fully connected layer with softmax activation. Classification performance is recorded for each subset, and the subset achieving the highest performance is selected. The script prints the optimal *Sensor* combination and returns a ranking of *Sensor* importance.

3.3.3 Embedded approach (sensor_selection_embedded.py)

This script implements the proposed embedded approach. It takes as input arguments:

- X_scaled: Same as the one from the wrapper approach;
- Y : Same as the one form the wrapper approach;
- sensor_names : idem;
- kfold (int): Number of K-Fold iterations (approximated via repeated splits) to average the importance ranking.

This script selects the most important *Sensor*s from the weights of the input layer of the model after a training round. Training is restricted using the L1/L2 regularization (elastic net) to encourage sparsity and balanced weight distribution, highlighting the influence of each *Sensor* on the model output. The model is the same as the wrapper one (LSTM + full connected softmax) but with different dimensionalities as this approach does not select subsets of *Sensors*, but tests all at the same time. Due to random weight initialization, multiple training rounds are performed, and the final *Sensor* ranking corresponds to the average rank across k-fold iterations. The script prints the best four-*Sensor* combination and returns the full ranking of *Sensor* importance.

Chapter 4 Conclusion

This deliverable presented the first complete implementation of an AI-based framework for selecting the most suitable combinations of sensing materials and operating conditions for the AMUSENS multi-pixel gas sensor platform. Starting from the extensive raw experimental dataset generated in ***Task 6.3***, the work defined a coherent processing pipeline to (i) segment continuous long-term experiments, (ii) extract standardized features, and (iii) assemble a machine-learning-ready dataset suitable for material-selection algorithms. The underlying methodological choices, including the use of relative response as the primary feature and a filter-based centroid-distance criterion as the baseline selection method, were motivated by the constraints of the current dataset and by established practices in gas-sensor signal analysis.

The core output of this deliverable is a functional software tool, equipped with a graphical user interface, that enables partners to explore, preprocess, and analyze sensing-material responses without extensive programming requirements. The tool identifies both the best individual materials and the most discriminative multi-material combinations, facilitating informed decisions for the development of the project's multipixel sensor. Additional utilities, such as visualization features, correction tools, and extensibility hooks for future algorithms, ensure that the software can evolve alongside improvements in data quantity and data quality.

As future datasets become available from subsequent measurement campaigns, the algorithms implemented here will be refined and complemented by more advanced wrapper and embedded methods already prepared in script form. This first release therefore represents not only a practical instrument for the current phase of the project but also a foundational step toward a robust, data-driven material-selection framework that will support the optimization of the AMUSENS platform throughout the remainder of the project.

Chapter 5 List of Abbreviations

Abbreviation	Translation
AI	Artificial intelligence
ALD	Atomic layer deposition
GUI	Graphical user interface
IMT	Institut Mines-Télécom
MOX	Metal oxide (gas sensor)
RH	Relative humidity
WP	Work package
EU	European Union

Chapter 6 Bibliography

[1] Vergara, A., Fonollosa, J., Mahiques, J., Trincavelli, M., Rulkov, N., & Huerta, R. (2013). On the performance of gas sensor arrays in open sampling systems using inhibitory support vector machines. *Sensors and Actuators B: Chemical*, *185*, 462-477.

[2] Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1200-1205). IEEE.