



Deliverable D4.4

Deployed Digital Twin on Consortium Servers

Work Package 4

MONITOR: Pilots, Data and Campaigns

Version: Final

Deliverable Overview

This deliverable includes details on the development of physics-based models using data from the pilots. These models create simulated environments that are an adequate representation of the physical world to better understand IAQ. The collected real-world data has been utilised to train Neural Networks (NNs) and be used in Computational Fluid Dynamics (CFD) simulations to underpin underlying processes responsible for indoor air quality. This has been setup as a simulation model, a so-called digital twin on a server for certain spaces, to demonstrate in a visual way to better understand IAQ for the public. This deliverable provides a report on the development of this prototype of the digital twins.

Additional Information

Type: DEM — Demonstrator, pilot, prototype

Dissemination Level: PU – Public

Official Submission Date: 31st of January 2025

Actual Submission Date: 30th of January 2025

Disclaimer

This document contains material, which is the copyright of certain EDIAQI partners, and may not be reproduced or copied without permission. All EDIAQI consortium partners have agreed to the full publication of this document if not declared “Confidential”. The commercial use of any information contained in this document may require a license from the proprietor of that information. The reproduction of this document or of parts of it requires an agreement with the proprietor of that information.

Document Revision History

Version	Date	Description	Partners
V0.1	16 th of January 2025	1 st draft of deliverable for peer review	ASC
V0.2	21 st of January 2025	Final draft of the deliverable	ASC
V0.3	23 rd of January 2025	Internal review of the deliverable	USEV + UMOL
V1.0	24 th of January 2025	Update after the internal review	ASC
Final	27 th of January 2025	Final quality check	LC

Authors and Reviewers

Authors

- Dejan Strbad (ASC)
- Valentino Petrić (ASC)

Reviewers

- Jesica Fernández-Agüera (USEV)
- Pasquale Avino (UMOL)

Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

List of Figures.....	5
List of Tables.....	6
Executive Summary	8
1 Introduction	10
2 Data Collection	11
2.1 Test room setup	11
2.2 Sensors used for data collection	11
2.3 Filtering unit.....	16
3 CFD simulations.....	18
4 Physical Models.....	22
4.1 Mass Balance.....	22
4.2 Decay Method	23
5 Model development.....	25
5.1 Data preparation	25
5.2 Model training.....	26
5.3 Evaluation of models	27
5.4 Prophet model	30
6 Digital Twin	33
6.1 Deployment of Digital Twin	33
6.1.1 Backend Service	33
6.1.2 Frontend Service	34
6.1.3 Architecture Highlights	34
6.1.4 Enhanced Functionality and Monitoring.....	34
6.1.5 Continuous Delivery Pipeline	35
6.2 Functionalities of a Digital Twin.....	35
7 Conclusion.....	38

List of Figures

Figure 1 The designated room for data collection and its location	11
Figure 2 Shows location of sensors in a room	13
Figure 3 Room with all measuring equipment	16
Figure 4 CAD of a room in Simscape.....	18
Figure 5 Comparison of Velocity Magnitude between CFD and Testo 405i Sensor	19
Figure 6 CFD simulations of air velocity.....	20
Figure 7 Particle movement during CFD simulations.	21
Figure 8 Calculation of λ and λd	22
Figure 9 Mass Balance results for one sensor	23
Figure 10 Results of Decay method.....	24
Figure 11 Model evaluation for each target and model was conducted without the filtering unit	28
Figure 12 Metric results for all models and targets without the filtering unit	29
Figure 13 Model evaluation for each target and model used with the filtering unit	29
Figure 14 Metric results for all models and targets with the filtering unit.....	30
Figure 15 Results of the Prophet model on the test set.....	32
Figure 16 GUI with tab 3D model	36
Figure 17 GUI with tab CFD Simulation of Air Fluctuation.....	36
Figure 18 GUI with tab CFD Simulation of Particles in a Room	37

List of Tables

Table 1 Sensor placement in the room.....	12
Table 2 Sensor Accuracy	12
Table 3 List of all measured parameters	14

List of Terms and Abbreviations

Abbreviation	Description
IAQ	Indoor Air Quality
NN	Neural Network
CFD	Computational Fluid Dynamics
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
RF	Random Forest
LR	Linear Regression
R^2	Coefficient of Determination
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
CD	Continuous Delivery
GUI	Graphical user interface

Executive Summary

The aim of this task was to develop physics-based models that simulate real-world environments to better understand indoor air quality (IAQ). The objective was to create simulated environments that represent real physical conditions. Real-world data was used for training Neural Network (NN) models and enhancing Computational Fluid Dynamics (CFD) simulations to capture the underlying processes influencing Indoor Air Quality (IAQ). Data was collected from the real-life settings for approximately one month, with measurements taken every minute.

A key focus of the project was to evaluate the indoor-outdoor relationship of airborne particulate pollutants, including fine particulate matter. A material balance model was employed to describe the rate of change in indoor particle concentration, factoring in ventilation, infiltration, and particle deposition onto surfaces. Ventilation rates were estimated using the decay method of trace gases.

To accurately replicate indoor spaces, a 3D laser-based scanner was used to digitise the room. This digital representation was integrated with a dynamic aerosol box model to evaluate key processes such as coagulation and deposition of particulates.

Machine learning models were also incorporated, including Long Short-Term Memory (LSTM) networks, Multi-Layer Perceptrons (MLP), and other models like Random Forest, Linear Regression, and XGBoost. These models were trained using data from the physics-based models and real-world measurements to enhance IAQ predictions. A user interface was developed for the digital twin, providing interactive and accessible insights into the simulated environments and IAQ dynamics.

The developed predictive models were successfully implemented, demonstrating promising results in forecasting air quality trends. The digital twin, with its robust backend and frontend services, enables real-time monitoring and provides valuable insights into indoor air quality. This work plays a key role in the broader EDIAQI project, particularly in the Filtration Pilot, where it will be used to evaluate filtration unit efficiency. Additionally, the digital twin offers a practical tool for users to optimise air purification strategies within their

indoor spaces. By integrating advanced modelling techniques, this project lays the foundation for future enhancements in intelligent environmental monitoring solutions.

<https://indoor-twin.ediaqi.eu/>

1 Introduction

Indoor Air Quality (IAQ) plays a crucial role in the health, comfort, and productivity of individuals in indoor spaces. Understanding the complex dynamics that control IAQ requires advanced modelling techniques that can accurately replicate the behaviour of airborne pollutants within these environments. This task aims to develop physics-based models by integrating real-world data and cutting-edge simulation technologies to create digital representations of indoor spaces. The goal is to better understand the processes influencing IAQ and predict the impact of various environmental factors on air quality.

Using data collected in this study, this project utilises Computational Fluid Dynamics (CFD) simulations and machine learning models, including Neural Networks (NNs), Random Forest, Linear Regression, XGBoost, and advanced deep learning techniques such as Long Short-Term Memory (LSTM) networks and Multi-Layer Perceptrons (MLPs).

These models are trained on data gathered in this task to capture the underlying processes responsible for changes in indoor air quality. One of the key aspects of this task is the evaluation of the indoor-outdoor relationship of particulate pollutants, particularly fine particulate matter, using a mass balance model to describe the movement and deposition of particles within indoor spaces.

To create highly accurate simulations, a 3D laser-based scanning application was used to digitise the physical environment. This representation is then integrated with dynamic aerosol models to assess the effects of coagulation and deposition on particulate matter. Additionally, the project incorporates a user interface for the digital twin, enabling interactive exploration of the simulated environments and IAQ dynamics. The integration of machine learning and CFD techniques aims to offer predictive capabilities for managing IAQ, providing valuable insights into how different environmental factors interact to influence indoor air quality.

2 Data Collection

In this section, a detailed overview of how the data was collected is presented.

2.1 Test room setup

For this task, a room in a household was used. This bedroom was chosen because of its simple geometry and availability. The room is located in a neighbourhood approximately 20 km from the city of Zagreb, on the first floor of a family home. The room is newly renovated and, as such, is well-suited for this data collection. Figure 1 shows the room before data collection began, as well as the location of the household. The household is situated at the crossroads of the main road and a dead-end street. Near to the household, there are two factories: one producing pharmaceutical and the other engaged in laser cutting of metal sheets. Traffic is much lighter compared to busy city streets.

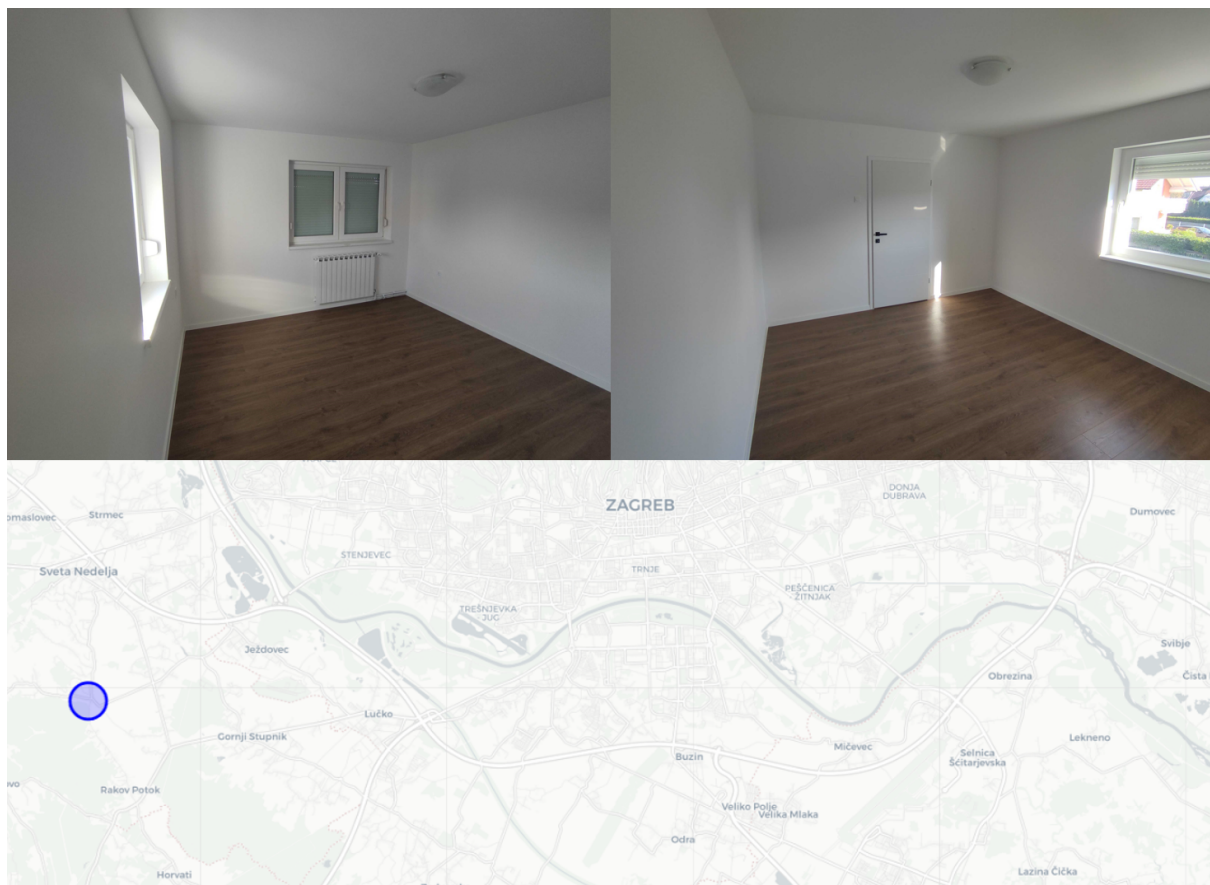


Figure 1 The designated room for data collection and its location

2.2 Sensors used for data collection

This section provides a detailed description of all the sensors used in this study. For the collection of indoor air quality data, six low-cost sensors were used in total, four from Wings

and two from Labservice Analytica. Wings and Labservice Analytica are sensor providers within the EDIAQI project. In addition to these low-cost sensors, Xiaomi Door and Window sensors were used to capture the opening and closing of windows. A Basler camera was used to monitor street traffic outside. All data from the Xiaomi sensors and the Basler camera was captured on a Raspberry Pi, a small single-board computer designed for embedded applications. Using Bluetooth and custom software, the Raspberry Pi collected and processed the data, enabling seamless integration of sensor and video inputs.

To capture data from the outdoor environment, Wings outdoor devices were employed. The six low-cost sensors were placed in two different spots and at three different heights. One set of sensors was positioned near the small window, while the other was placed near the centre of the room. Sensors from both providers recorded measurements every minute from the 23rd of November 2024 to the 19th of December 2024.

The exact measurements and placement of the sensors are detailed in Table 1. All measurements were taken with reference to the corner to the left of the door. Figure 2 shows the locations of the sensors in the room.

Table 1 Sensor placement in the room

Type of Sensor	X	Y	Z
Wings 316	115	137	100
Wings 4	115	137	13
Wings 7	310	225	100
Wings 13	310	225	145
Labservice Analytica 68	310	225	13
Labservice Analytica 69	115	137	145

Table 2 shows the accuracy of sensor number 4 in one Wings device, as well as the Wings outdoor device, for parameters provided by the manufacturer. Unfortunately, Labservice Analytica has not yet developed a method for evaluating each of their sensors.

Table 2 Sensor Accuracy

Type of Sensor	Parameter	Accuracy R ²
Wings Indoor	CO ₂	0,41
	TVOC	0,40

	PM ₁	0,98
	PM _{2,5}	0,98
	PM ₁₀	0,98
	Temperature	0,75
	RH	0,93
	Pressure	1,0
Wings Outdoor	NO ₂	0,66
	O ₃	0,83
	SO ₂	0,80
	CO	0,91
	PM ₁	0,97
	PM _{2,5}	0,98
	PM ₁₀	0,95
	NO	0,92
	Temperature	0,90
	RH	0,96

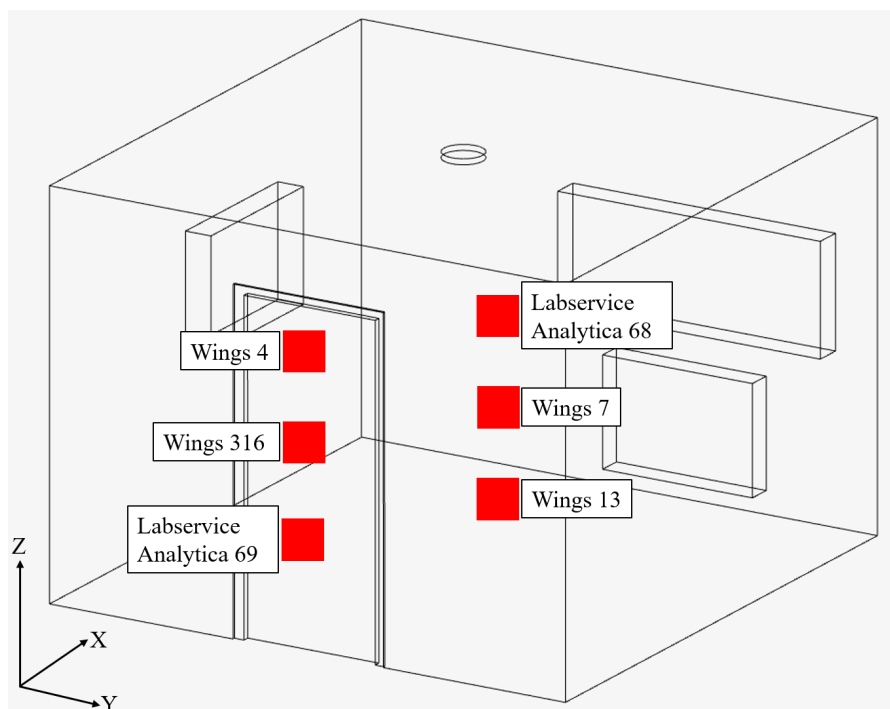


Figure 2 Shows location of sensors in a room

Xiaomi door/window sensors were placed on both windows to monitor opening and closing. However, due to user behaviour, only one window was opened in this study, so data from the other sensor was disregarded. The second window was used for observing traffic, with a camera placed on it and pointed at the street to monitor traffic conditions. An outdoor sensor was placed on the balcony above the window to capture data most representative of the outdoor air entering the room. All the data collected in this study is shown in Table 3, along with the exact measuring units and parameters.

Table 3 List of all measured parameters

Type of Sensor	Measurement	Description
Xiaomi Door/window	str	state of doors/windows, on opened, off closed
Labservice Analytica 68 and Labservice Analytica 69	timestamp	timestamp in UTC
	NetCO2-CO2(ppm)	CO ₂ in ppm
	NetCO2-CO2_ug	CO ₂ in ug
	NetADC_O3-O3_ug	O ₃ in ug
	NetADC_O3-O3(ppb)	O ₃ in ppb
	NetADC_CO-CO_ug	CO in ug
	NetADC_CO-CO(ppb)	CO in ppb
	NetPM-PM10(µg/m ³)	PM ₁₀ in (µg/m ³)
	NetPM-PM25(µg/m ³)	PM _{2,5} in (µg/m ³)
	NetPid_P1-PPB(ppm)	TVOC in ppm
	NetPid_P1-PPB_ug	TVOC in ug
	NetBME280-PERC(%)	Relative humidity in %
	NetBME280-TEMP_EXT(°C)	Temperature in °C
	NetBME280-HPA(hPa)	Atmospheric pressure in hPa
	NetADC_NO2-NO2_ug	NO ₂ in ug
	NetADC_NO2-NO2(ppb)	NO ₂ in ppb
Wings 13, 4, 7 and 316	timestamp	Timestamp in zagreb time
	timezone	Timezone
	CO2	CO ₂ in ppm
	TVOC	TVOC in ppb
	CO	CO in ppm
	PM1	PM ₁ in (µg/m ³)
	PM25	PM _{2,5} in (µg/m ³)
	PM10	PM ₁₀ in (µg/m ³)

	temperature	Temperature in °C
	humidity	Relative humidity in %
	atmospheric_pressure	Atmospheric pressure in hPa
Wings outdoor device	timestamp	Timestamp in zagreb time
	timezone	Timezone
	NO2	NO ₂ in ppb
	O3	O ₃ in ppb
	SO2	SO ₂ in ppb
	CO	CO in ppb
	PM1	PM ₁ in (µg/m ³)
	PM25	PM _{2,5} in (µg/m ³)
	PM10	PM ₁₀ in (µg/m ³)
	NO	NO in ppb
	temperature	Temperature in °C
	humidity	Relative humidity in %
	atmospheric_pressure	Atmospheric pressure in hPa
	noise	Noise in dB
	CO2	CO ₂ in ppm
Traffic	timestamp	Timestamp in zagreb time
	car	Number of cars that passed by in a minute
	truch	Number of trucks that passed by in a minute
	bus	Number of buses that passed by in a minute
	motorcycle	Number of motorcycles that passed by in a minute

Figure 3 shows how the room looked after all the sensors and measuring devices were placed setup.



Figure 3 Room with all measuring equipment

2.3 Filtering unit

Once the measuring period started, only a few activities were conducted inside the room, as we could not perform more due to room constraints. Activities that were carried out inside the room were: opening the windows, performing experiments related to the mass balance model (explained later), and moving the filtering unit. One of the objectives of the digital twin is to evaluate the position of the filtering unit and its impact on air quality inside the room.

For this purpose, we used the filtering unit iTherapy Model 100, which has an intake velocity of 2.8 m/s, an outtake velocity of 1.5 m/s, and an outtake area of 62.1 cm². The iTherapy Model 100 features a 5-layer filter system, starting with pre-filtration to remove PM_{2,5}, dust, and fibres, followed by an antibacterial layer targeting allergens and viruses. Additional layers include nano silver membranes that capture 0.3-micron particles, a cold catalytic process to decompose harmful chemicals, and a vitamin C composite layer that neutralises odours while releasing antioxidants. The filtering unit was placed inside the room and turned on after the first week of measurements to capture the baseline air quality of the room. This approach allowed us to gather data on the room's air quality without the influence of the filter, ensuring that any changes observed after turning it on could be attributed to the filter's impact. It was moved daily to different spots and heights within the room. In total, the filtering unit was placed at 14 different locations, remaining at each position for a minimum of six hours of operating time. The filter was replaced twice during

the measuring cycle as part of an analysis for future tasks. During the analysis, it was confirmed that with a higher filter change rate, the particles became too small in concentration for the equipment to effectively qualify and quantify them, making it challenging to detect the particles on the filters.

3 CFD simulations

To create CFD simulations that demonstrate how air flows through the room and how particles behave, the research team utilised SimScale, developed by SimScale GmbH.

SimScale is a cloud-based simulation platform that enables engineers and researchers to perform computational fluid dynamics (CFD), finite element analysis (FEA), and thermal simulations directly from a web browser. It eliminates the need for expensive hardware and complex software installations, offering powerful tools for modelling fluid flows, structural stress, heat transfer, and more. With its user-friendly interface and collaborative features, SimScale allows teams to work together efficiently on simulation projects. SimScale was used for all aspects of the CFD simulations.

The process began with a 3D scan of the room, which generated an .stl file later used in the digital twin user interface. After making a few minor adjustments to the geometry and shape of the room using the editing tools in SimScale, a CAD model was created. Figure 4 shows how the room appeared in SimScale before any simulations were performed.

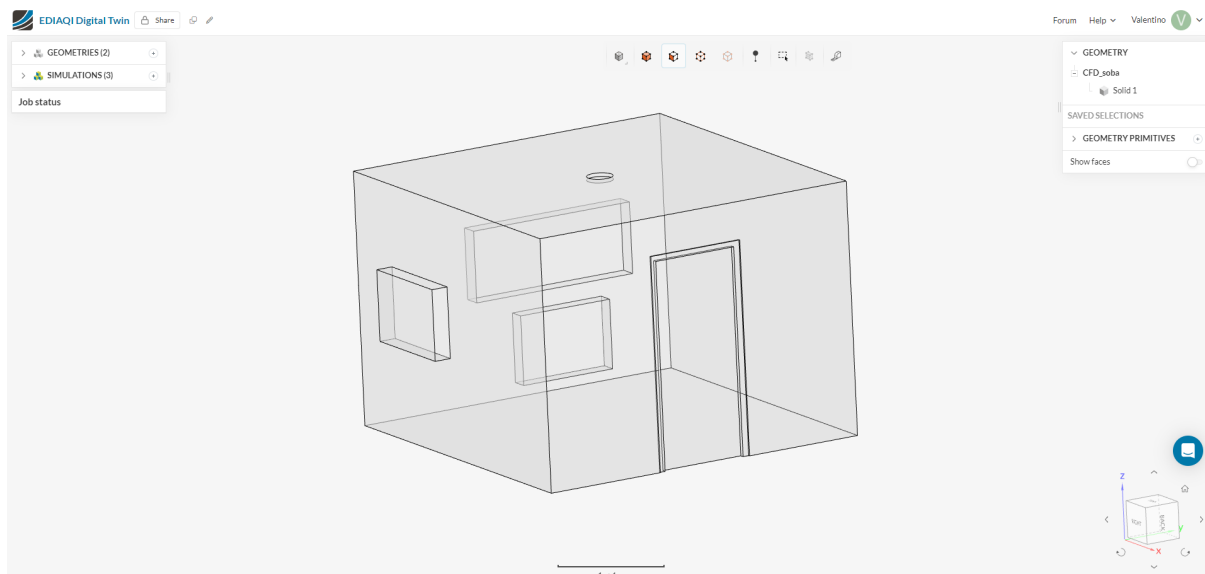


Figure 4 CAD of a room in Simscale

After finalising the geometry, we conducted several simulations to achieve the best results.

To validate our results, we used the Testo 405i sensor to measure airflow. Figure 5 shows the velocity results from the CFD simulation and the Testo 405i sensor. While the data do not overlap perfectly, the trends observed in the CFD simulation closely align with the measured data. This suggests that, based on a thorough comparison of the results, the CFD model is a reliable representation of the system's behaviour and can be considered the most

accurate model for our analysis, given the consistency in the observed trends and the underlying methodology used in its development. This setup was the best for simulation and was used for further analysis. Boundary conditions were set up using sensors to capture the state of the room and the outside environment. The research team used temperature readings from both indoor and outdoor sensors, as well as air flow readings when the window was open. The simulation walls were set up in such a way that the walls with windows were considered external walls, while the walls with doors and the empty wall were treated as internal walls with the same temperature as the room. Pressure outlets were placed at the door frame and in the window that was closed.

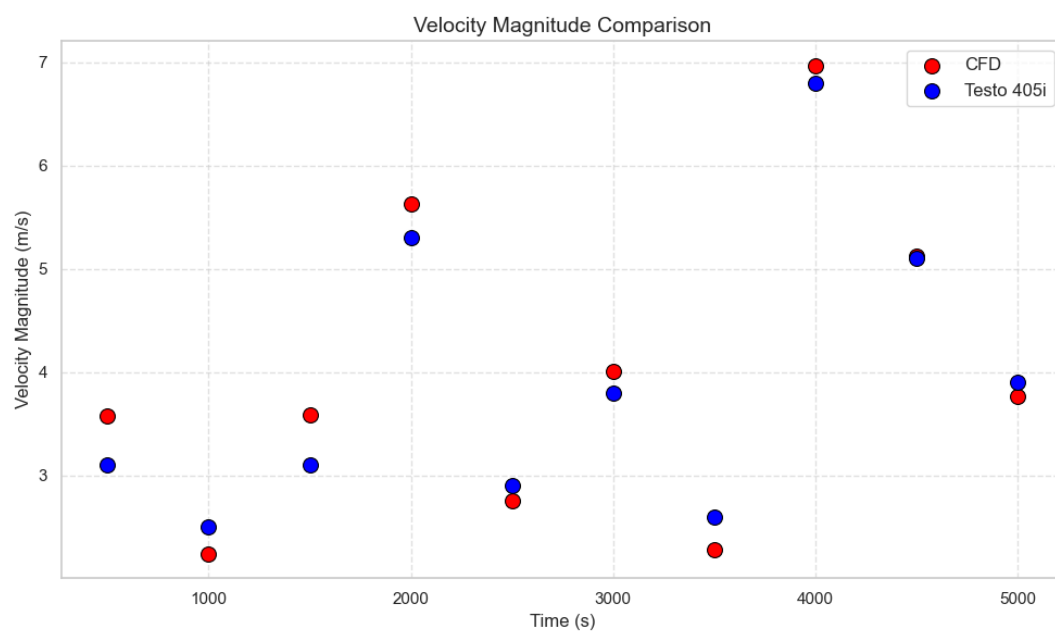


Figure 5 Comparison of Velocity Magnitude between CFD and Testo 405i Sensor

Following an evaluation of the simulations, two were identified as providing the most accurate results, as their trends closely aligned with the measured sensor data. This alignment suggests that, despite minor discrepancies, these simulations reliably represent the system's behaviour and can be considered the most accurate models for our analysis. Figure 6 presents the results of the CFD simulation for air velocity inside the room when the window is open, with an outside temperature of 5°C and an indoor temperature of 20°C. The results are shown for two-time intervals: after 2,500 seconds and 5,000 seconds of simulation. The upper images depict the state after 2,500 seconds, while the lower images represent the state after 5,000 seconds. The images are arranged to display the X, Y, and Z axes in that order.

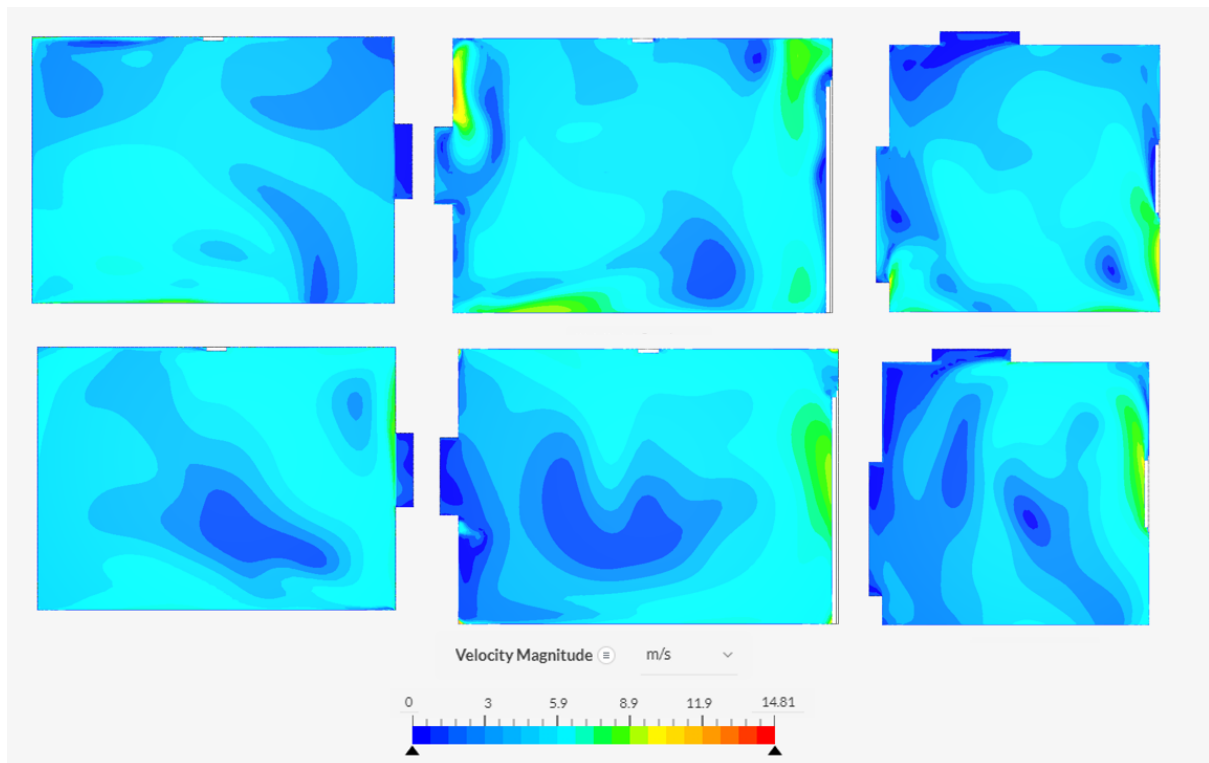


Figure 6 CFD simulations of air velocity

Once the simulation was validated, we used the same setup to observe how particles behave within the room. Particles with a size of $10\ \mu\text{m}$ were used for this analysis. Figure 7 shows the results of particle movement inside the room after the simulation was completed (5,000 seconds). As before, the images are arranged to display the X, Y, and Z axes in that order.

Each path represents the trajectory of particles, along with their velocity over time. Based on these simulations, we can identify areas of turbulence within the room and observe how turbulence affects particle behaviour. Areas with turbulence exhibit lower deposition rates but higher resuspension, leading to elevated particle concentrations. This suggests the presence of some form of ventilation in the system, such as mechanical ventilation or an open window, which can increase particulate matter concentrations while reducing the concentrations of gases, such as CO_2 .

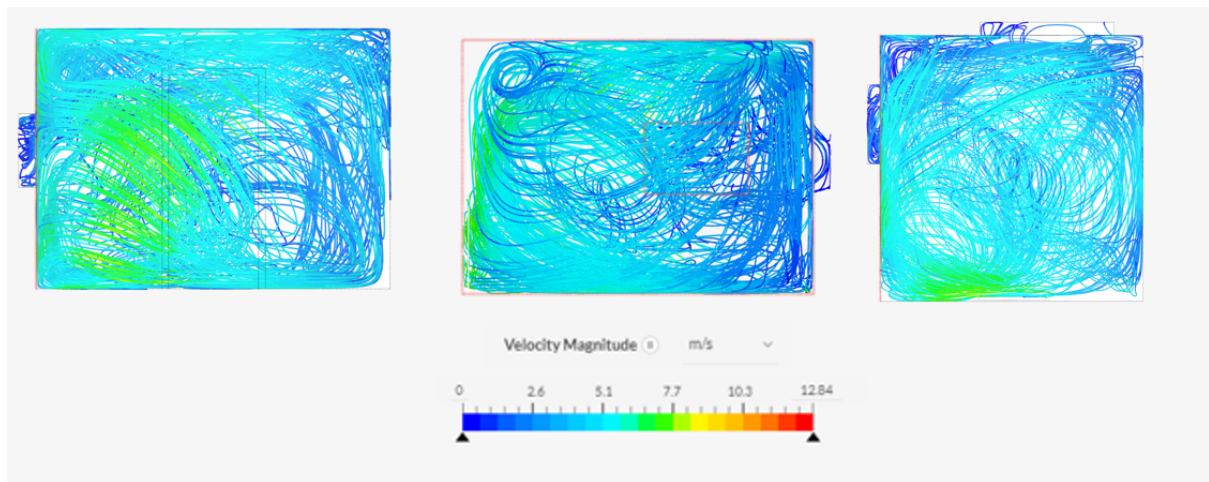


Figure 7 Particle movement during CFD simulations.

4 Physical Models

In this section, two physical models used to create the hybrid model are described, along with the experiments conducted to collect the data necessary for determining the parameters. The two physical models utilised are the mass balance model for particulate matter (PM) and the decay method model for CO₂.

4.1 Mass Balance

The Mass Balance model, or Indoor Aerosol Mass Balance model, is designed to describe changes in indoor aerosol concentration due to various factors. The model is represented by Equation 1, where the change in indoor aerosol concentration (I) over time is based on the outdoor concentration (O), which is multiplied by the penetration factor (P) and the ventilation rate (λ). In addition to outdoor sources, the equation also accounts for the loss of particles from inside to outside, described by λ multiplied by the indoor concentration. Another factor in the model is deposition, which is characterised by the deposition rate (λ_d). Finally, S represents indoor sources, such as humans, cooking, or walking, etc.

$$\frac{dI}{dt} = P\lambda O - \lambda I - \lambda_d I + S \quad (1)$$

To obtain each of these parameters, two experiments were conducted using candles to produce indoor air pollution. The candles were lit, and the room was closed. After 2 hours of the candles burning, they were extinguished, and the window was opened. This experiment was done to measure λ . The second experiment was the same as the first, but instead of opening the window, the room was simply closed for 24 hours. This experiment was done to measure λ_d . A detailed explanation of how this was done is provided in Figure 8.

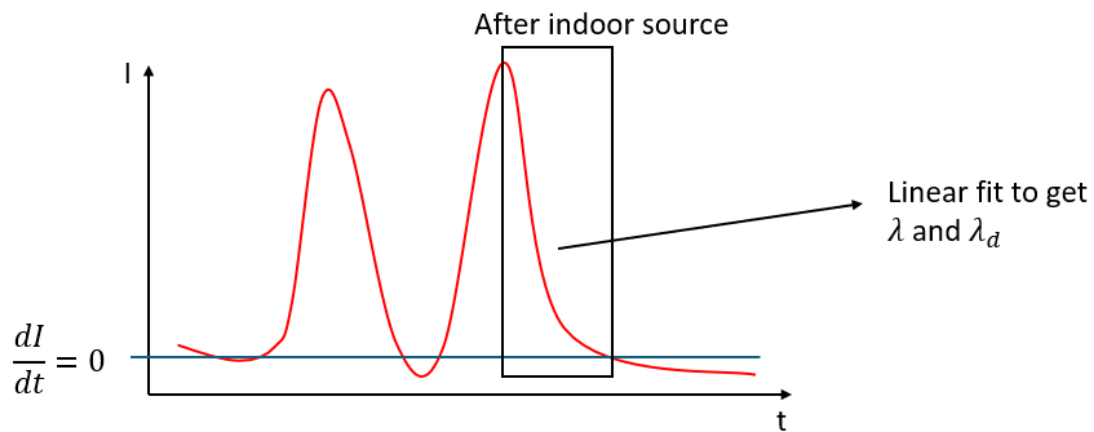


Figure 8 Calculation of λ and λ_d

After the source data collected from low-cost sensors is extracted and linearly fitted to obtain these two parameters, P is calculated using outside sensor data, as described in Equation 2.

$$P = \frac{(\lambda + \lambda_d)I}{\lambda * O} \quad (2)$$

After obtaining all the parameters, the research team used the mass balance model to calculate the air change rate over time and indoor sources. This data was then used for training machine learning models, with more information on how this was done provided in one of the next sections. This was done for PM_{10} and $PM_{2.5}$.

Figure 9 shows the results of the mass balance model, where air exchange represents the change of indoor air over time, and the indoor source indicates whether the change in indoor air is influenced by indoor sources or if windows are open and pollution is coming from outside. The large increase in the indoor source or significant change in dI/dt is influenced by the burning of a candle and the production of PM_{10} from the combustion process.

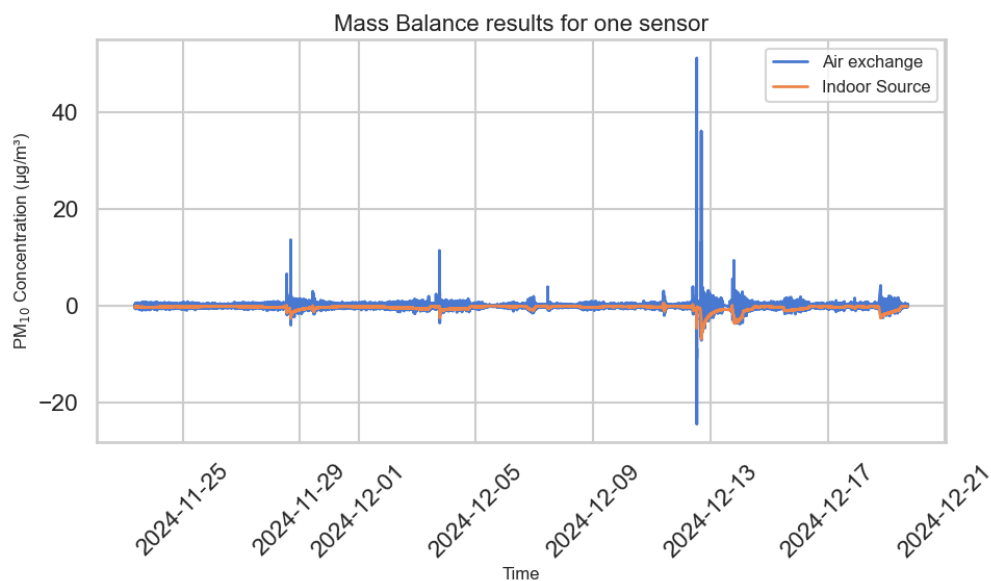


Figure 9 Mass Balance results for one sensor

4.2 Decay Method

Since the previous mass balance model is mainly used for aerosols, decay method is used for CO_2 . Decay method, as applied in this task, is described by Equation 3.

$$C_t = C_{ext} + (C_0 - C_{ext}) * e^{-nt} \quad (3)$$

This equation shows how the concentration of gas changes over time, describing how C_t moves from its initial value C_0 towards a steady-state value C_{ext} . A key factor here is the

exponential decay, represented by Term 4, which decreases over time. The rate of decay is controlled by the constant n . As time t increases, this term decreases, and C_t moves closer to C_{ext} . The constant n determines how fast the decay occurs: if n is larger, the concentration will approach C_{ext} more quickly, and if n is smaller, the concentration will decay more slowly.

$$(C_0 - C_{ext}) * e^{-nt} \quad (4)$$

To obtain the parameter n , we created an experiment similar to the first one conducted to determine the parameters of the mass balance equation. As before, we used candles that were lit and left to burn completely. Afterwards, the window was opened, and we observed how the concentration of CO₂ changed based on the factor n . Using Equation 3, we calculated n since it was the only unknown value. This model is used for gas modelling and is incorporated into the dataset used for training machine learning models.

Figure 10 shows the results of the decay method and real data. In this figure, it can be observed how CO₂ concentrations are affected by ventilation in three different intervals. It can be concluded that a minimum of 5 minutes is needed for CO₂ to reach outside levels, and 10 minutes is required to achieve full ventilation and equilibrium.

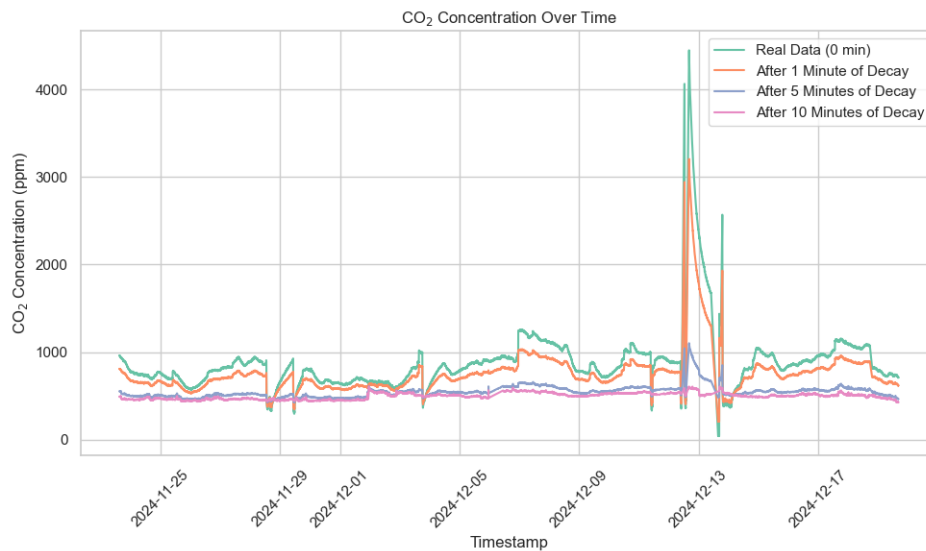


Figure 10 Results of Decay method

5 Model development

This section provides a detailed explanation of how the models were trained and evaluated. It covers the entire process, starting from data pre-processing, including how physical models were incorporated into the dataset, to outlier removal and model selection.

5.1 Data preparation

After all the data was collected, it was first combined into one DataFrame, which was used for the analysis, and all pre-processing methods were applied to it. Afterwards, the research team implemented physical models to the dataset to create a hybrid model. We began by calculating the mass balance model for PM_{10} and $PM_{2.5}$, where the change in indoor concentration over time was calculated, as well as S , based on Equation 1. However, this calculation was only done when the window was open. When the window was closed, the research team removed the part of the equation related to outdoor concentration since there was no air exchange.

Next, the decay method model was applied, as it is only relevant when there is exchange between indoor and outdoor air. The indoor concentration was calculated at three elapsed times: 1 minute, 5 minutes, and 10 minutes. This was done because, after 10 minutes, there was no more change, and the indoor space reached a steady state. This represents the environment, and the current concentration of C will be affected if the window is opened. Lastly, we incorporated the filtering unit's position and whether it was operating. First, the Euclidean distances were calculated based on the locations of the sensors and the filtering unit. This value was then divided by 1, where 1 corresponds to the filtering unit being operational, and 0 corresponds to it being off. When the unit was not working, the distance was not calculated because it did not affect measurements at the sensors.

In addition to all feature engineering, data processing continued with the removal of outliers, which was done using a 3-day window. Measurements below the 25th percentile or above the 75th percentile were dropped. All missing values (NaNs) were removed using the IterativeImputer from scikit-learn. The Iterative Imputer handles missing data by treating the imputation process as a series of predictions. It uses a model trained on the available (observed) data to estimate and fill in the missing values, iteratively refining these predictions for better accuracy.

Furthermore, for every measurement from indoor and outdoor low-cost sensors, lag features were created using previous values to observe how the current state and prior measurements affect the current readings of the devices. After that, to analyse how temporal features like the day of the week and the hour of the day affect measurements, two features, `day_sin` and `hour_sin`, were created. For each of these features, a sinusoidal distribution was used: for `day_sin`, it ranged from the middle of the day to midnight of the next day, and similarly for the week, where Monday marked the start, and Sunday marked the end of the distribution.

The dataset was then divided into two subsets: one where the filtering unit was not running and one where it was. This division was made to obtain two sets of measurements for each pollutant and environmental parameter in the digital twin and to analyse how the location of the filtering unit affects the measurements in the digital twin.

5.2 Model training

In this task, a total of 70 models were trained using Python and its libraries. To be more precise, Python 3.10 was used alongside scikit-learn and PyTorch. For each of the 7 pollutants used in the digital twin (PM_{10} , $PM_{2.5}$, temperature, RH, pressure, TVOC, and CO_2), 2 different datasets were used: one with the filtering unit and one without. Additionally, 5 different models were used: Linear Regression, Random Forest Regression, XGBoost, MLP, and LSTM. Before training, the data was scaled for all models except for RF and XGBoost. The hyperparameters of the models were optimized using grid search, which is incorporated into the scikit-learn library.

The datasets were divided into two subsets: a training set and a test set. A separate validation set was not created because cross-validation was already employed during hyperparameter optimization to ensure robust model evaluation.

The split between the training and test datasets was 80/20, with 80% of the data allocated for training and 20% for testing. This approach ensures that the models have sufficient data for learning while reserving a reasonable portion for independent evaluation of their performance.

For the dataset where the filtering unit was not operational, the training set consisted of 24,733 rows of data, while the test set included 6,184 rows. In contrast, the dataset where

the filtering unit was active contained significantly fewer data points. In this case, the training set included 5,291 rows, and the test set had 1,323 rows.

This disparity in dataset size reflects the operational time and conditions under which the data was collected. When the filtering unit was on, the reduced data size might have been due to limited measurement periods or specific operational constraints. These differences in data volume can have a significant impact on the models' performance, particularly for more complex algorithms that require larger datasets for effective training.

The final input to the model consisted of outdoor sensor readings, all pre-processed features (lag and temporal), and mass balance models. The model's outputs were readings from sensors inside the room, resulting in a total of six targets.

5.3 Evaluation of models

After training, the models were evaluated using three metrics: Coefficient of Determination (R^2), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). R^2 , also known as the goodness-of-fit measure, indicates how well the independent variables in a model explain the variance in the dependent variable. It is calculated as shown in Equation (4), where $\bar{y}_i$ represents the mean of the actual values. The higher the R^2 value, the better the predictions of the model.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (4)$$

Since R^2 alone cannot be the only metric for evaluating model performance, RMSE and MAE were also calculated using Equations (5) and (6), respectively. For both metrics, higher values indicate poorer model performance. Ideally, these metrics should be as close to zero as possible.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

RMSE is particularly useful when penalising large errors heavily, while MAE is simpler and less sensitive to outliers, treating all errors equally. Together, these metrics provide a comprehensive evaluation of the model's predictive performance, with R^2 offering a holistic view of fit quality.

Figure 11 shows the results of model training for each of the targets and models used when the filtering unit is not operational. For each of the six sensors' readings inside the room, the measurements closely simulated those from the digital twin. The evaluation of the models is based on the mean of the six predictions. For example, to determine the room temperature in the digital twin, predictions are run for each of the sensors, and then the mean is calculated and presented.



Figure 11 Model evaluation for each target and model was conducted without the filtering unit

For each of the targets or pollutants, linear regression demonstrated the best performance, a result that is further corroborated in Figure 12. The metrics clearly show that linear regression outperformed the other models across all targets. In contrast, both LSTM and MLP struggled with predictions, which can be attributed to the limited size of the training dataset and the simplicity of the models, with only a single hidden layer. Random Forest and XGBoost exhibited similar performance, outperforming the LSTM and MLP models. However, they still did not reach the accuracy of linear regression, which further emphasises how the scale of the dataset influenced model performance. This suggests that more complex models, like LSTM and MLP, might require larger datasets to fully capture the

underlying patterns, while simpler models such as linear regression can achieve better results with limited data.

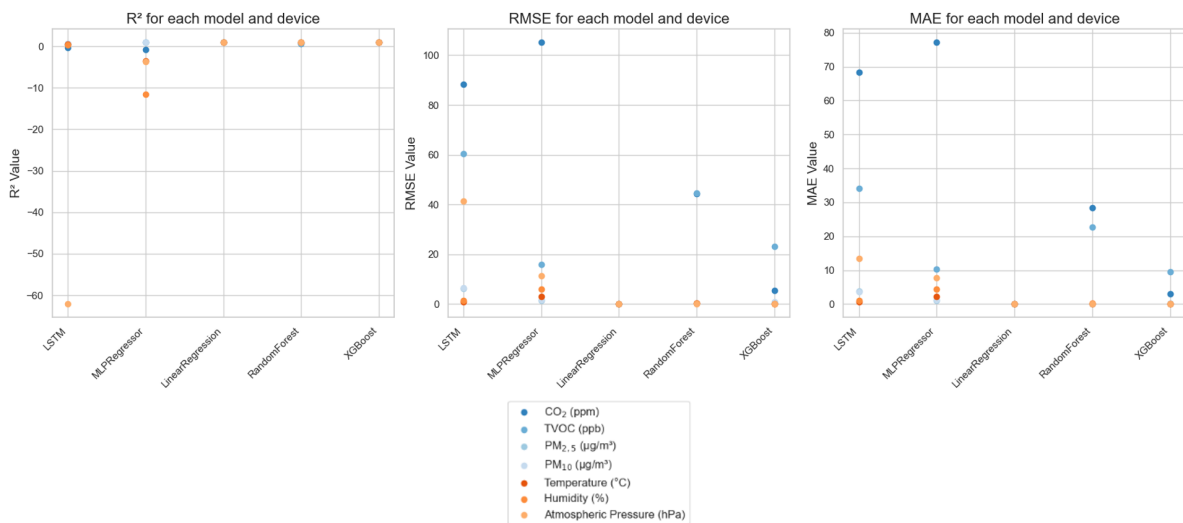


Figure 12 Metric results for all models and targets without the filtering unit

To better understand the performance of the models when only the filtering unit was operational, the dataset was divided into subsets as previously described. Figure 13 presents the results of the model training under this specific condition. The results are consistent with the case where the filtering unit was not operational. Linear regression outperformed other models, delivering the best results for this task.

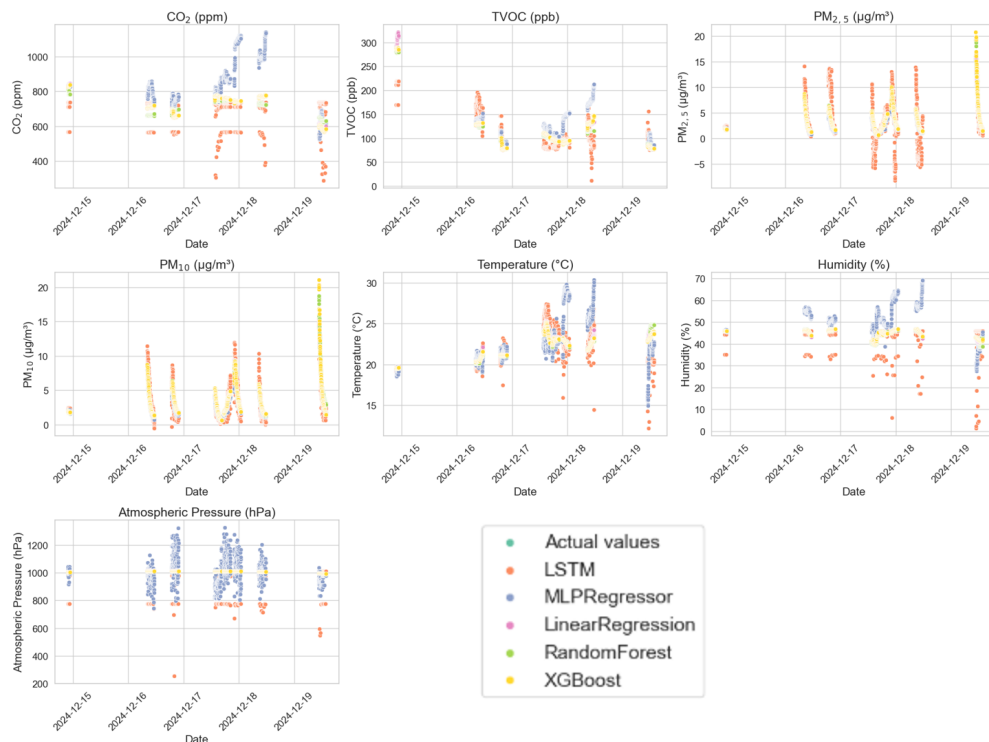


Figure 13 Model evaluation for each target and model used with the filtering unit

This superior performance can be attributed to the small size of the dataset, which is even more limited in this scenario, as it only includes data collected while the filtering unit was active. Linear regression, being a simpler model, performs well on smaller datasets and does not suffer as much from overfitting or complexity-related challenges.

The performance of the other models, including LSTM, MLP, Random Forest, and XGBoost, was largely similar to their performance in the previous task without the filtering unit. This can be observed in Figure 14. Both LSTM and MLP struggled due to the dataset's limited size and their architectures, which are better suited for larger datasets with more temporal complexity. Random Forest and XGBoost showed moderately better performance than LSTM and MLP, but they still fell short of Linear Regression, once again highlighting the impact of dataset scale and the simplicity of the problem on model performance.

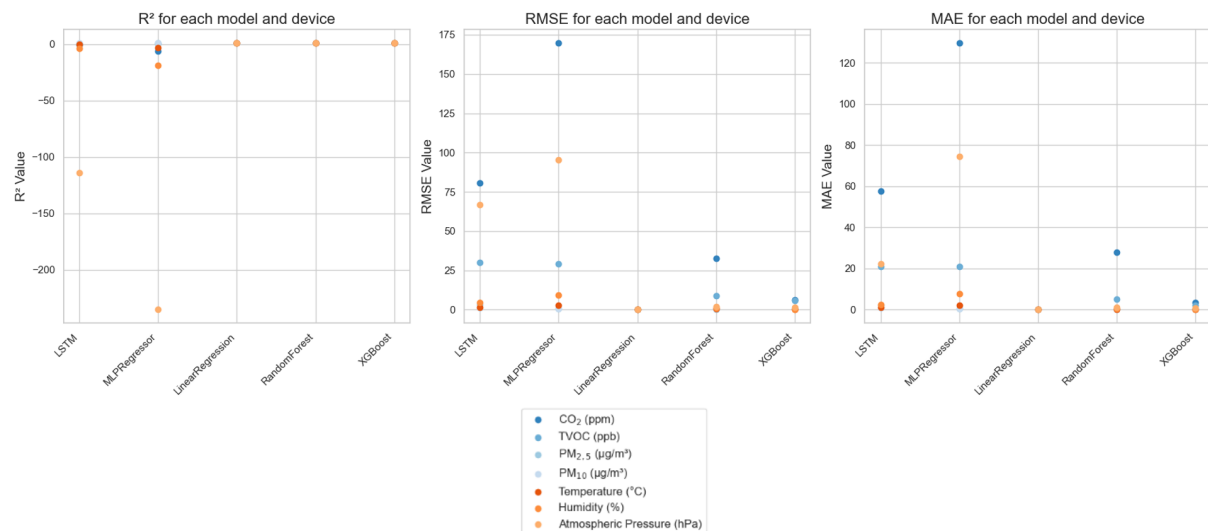


Figure 14 Metric results for all models and targets with the filtering unit

For the reasons stated above, linear regression is implemented in the Digital Twin as a model for predicting indoor air pollutants

5.4 Prophet model

To make future predictions based solely on readings from sensors, a Prophet model was also developed. Prophet is a forecasting tool developed by Facebook, designed to handle time series data with daily, weekly, or yearly seasonality, and it can accommodate holidays or special events. The model works by decomposing the time series into three main components: trend, seasonality, and holiday effects.

- Trend represents the long-term direction in the data (whether it is increasing, decreasing, or staying constant over time).
- Seasonality captures regular patterns that repeat at known intervals, such as weekly, yearly, or daily cycles.
- Holiday effects allow the model to account for special events (e.g., public holidays or specific seasonal events) that may cause irregular spikes or dips.

Prophet is designed to work well even with noisy data or missing values and can be easily adjusted by adding seasonality components or holiday effects. However, unfortunately, the results were not satisfactory, as shown in Figure 15. The dataset was too small for Prophet to capture any meaningful trends, which led to poor performance. A separate Prophet model was created for each of the sensors, similar to the machine learning models, and observed that all the sensors produced very similar values. This is likely because the small room selected for data collection caused the readings from all sensors to be quite similar. In this scenario, one Prophet model could influence the entire set of sensors, further limiting the model's ability to detect unique trends or behaviours.

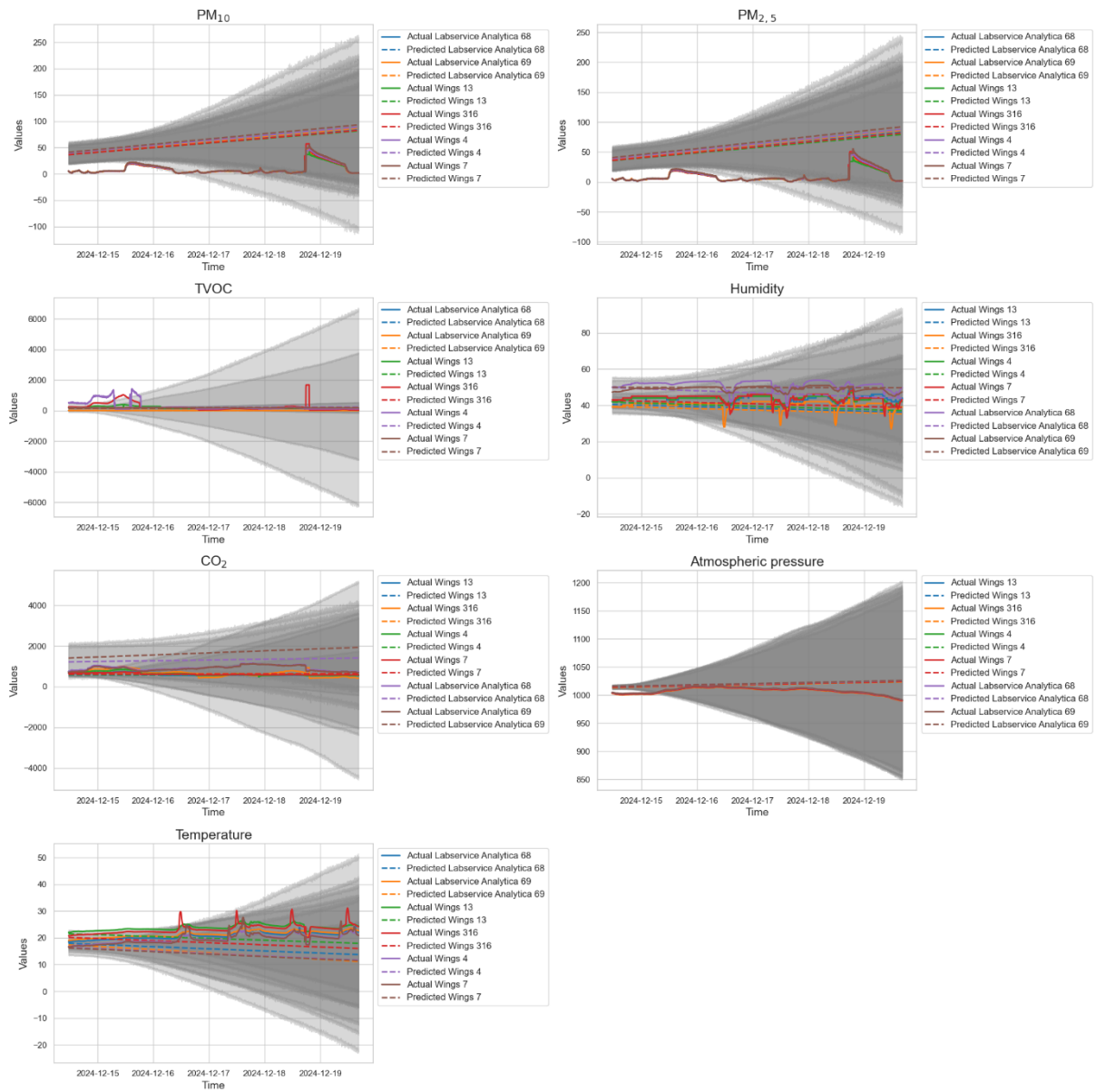


Figure 15 Results of the Prophet model on the test set

6 Digital Twin

In this section, a detailed description of the digital twin is presented, including its main goals, structure, and guidance on how to use it. Additionally, it showcases its features and functionality. The Digital Twin has the same inputs as the ML models. Each pollutant is calculated by inputting lag features, outdoor sensor readings, temporal features, traffic data, and mass balance models. Users can input these settings via the GUI and then view the results for the indoor environment, including how the filtering unit and its placement affect the space. Additionally, the model shows how ventilation through windows impacts indoor CO₂ concentrations.

6.1 Deployment of Digital Twin

The application is deployed within a Kubernetes cluster, leveraging a Continuous Delivery (CD) pipeline to ensure seamless updates and rapid iteration. The deployment architecture is designed around two core services: the Backend Service and the Frontend Service, which work together to deliver the application's functionality and user experience effectively.

6.1.1 Backend Service

The backend service is responsible for the application's core functionality. It exposes a FastAPI-based API (or a similar interface) to handle critical workflows, data processing, and integration with external systems. Key aspects include:

- **Data Processing and Business Logic:** Orchestrates essential workflows, manages data transformations, and communicates with external systems or databases to meet application requirements.
- **Scalable and Resilient Deployment:** The backend is containerised and deployed as pods within the Kubernetes cluster, ensuring high availability and scalability.
- **Dynamic Autoscaling:** Kubernetes autoscaling policies dynamically adjust the number of pods based on metrics like API request load, CPU usage, and memory consumption, ensuring optimal resource utilization.
- **Cluster Communication:** Endpoints are exposed using a Kubernetes Service object, enabling seamless communication within the cluster and external access when required.

6.1.2 Frontend Service

The frontend service provides the user interface, delivering static assets such as HTML, CSS, JavaScript, and other resources. This service is designed to work seamlessly with the backend API to offer a responsive and dynamic user experience. Key aspects include:

- **Static File Hosting:** The service employs a reverse proxy server, such as Nginx or Apache, to efficiently serve static assets and manage client requests.
- **Isolated and Scalable Deployment:** Like the backend, the frontend is containerized and runs in its own pods within the Kubernetes cluster, ensuring high availability.
- **Backend Integration:** The frontend communicates directly with the backend API to fetch data and process user inputs, creating a cohesive flow between the services.

6.1.3 Architecture Highlights

The backend and frontend services are tightly integrated within the Kubernetes environment:

- **Internal Networking:** Both services communicate through Kubernetes-native internal networking and service discovery mechanisms, ensuring efficient and reliable interaction.
- **Traffic Management:** External traffic is routed to the appropriate service using an ingress controller, which directs requests based on paths or domain configurations. This setup simplifies access while maintaining security and performance.

6.1.4 Enhanced Functionality and Monitoring

To support efficient operation and robust application performance, the deployment incorporates:

- **Configuration Management:** Sensitive data and configuration settings are securely managed using Kubernetes ConfigMaps and Secrets.
- **Persistent Data Storage:** Persistent Volumes are utilized where data storage requirements arise, ensuring durability and accessibility.
- **Comprehensive Monitoring:** Tools like Prometheus, Grafana, and the ELK stack provide visibility into system performance and health, enabling proactive issue detection and resolution.

6.1.5 Continuous Delivery Pipeline

The Continuous Delivery (CD) pipeline automates the entire deployment lifecycle to ensure consistent and rapid delivery of updates. Key features include:

- **Automated Build and Deployment:** Application updates are automatically built, tested, and deployed to the Kubernetes cluster.
- **Version Control Integration:** By integrating with version control systems, the pipeline streamlines codebase management, making deployments more predictable and efficient.

6.2 Functionalities of a Digital Twin

The Graphical User Interface (GUI) consists of several sections and windows. When a user opens the GUI, they are greeted with the screen shown in Figure 16. On the left side of the screen, there is a dropdown menu where users can input the data they want to test in the digital twin. This includes indoor and outdoor environment data, ranging from pollutant concentrations to ambient sources of data.

In the "Additional Information" section, users can specify the traffic present on the street in front of the room. Four traffic options are available: cars, trucks, buses, and motorcycles. Since ventilation in this simulation is achieved solely through the window, users can control the window's status by setting it to either open or closed.

Additionally, users can specify the position of the filtering unit. They can place the filtering unit at a precise location within the room by entering distances along the three axes (x, y, z), with the origin (0, 0, 0) located at the corner between the small window and the door.

After entering all the required data, users can press the "Run" button to initiate the backend models. These models evaluate the input data, and the results are displayed on the right side of the screen. Two sets of values are shown: "Filter," which displays the predicted values when the filtering unit is active, and "No Filter," which shows the predicted values when the filtering unit is turned off.

Additionally, there is a chat box above the results section. Once the results are displayed, the data is sent to a chatbot that provides users with best practices for improving indoor air quality.

In the top-right corner, there is an option to choose between two languages: Croatian and English. Users can select their preferred language, and the entire GUI is automatically translated accordingly.

In the middle of the screen, there are three tabs. The first tab contains a 3D scanned model of the room, which users can interact with.

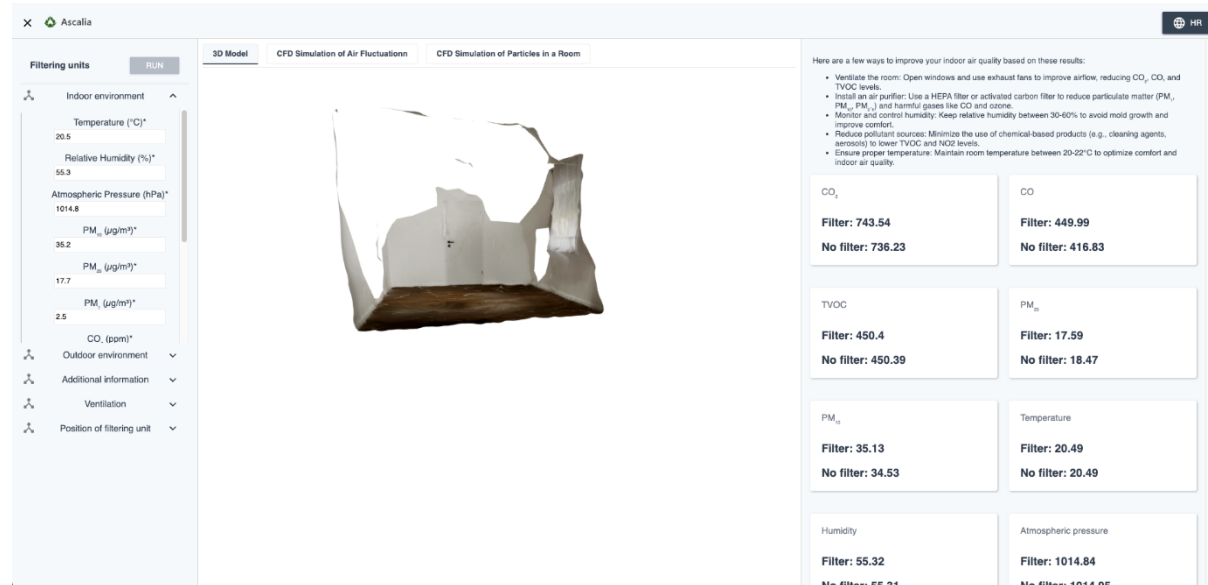


Figure 16 GUI with tab 3D model

Figure 17 shows the GUI and the second tab, which contains a video of the CFD simulation inside the room depicting air fluctuation. The parameters for the simulation are displayed above the video, and the simulation runs in a loop

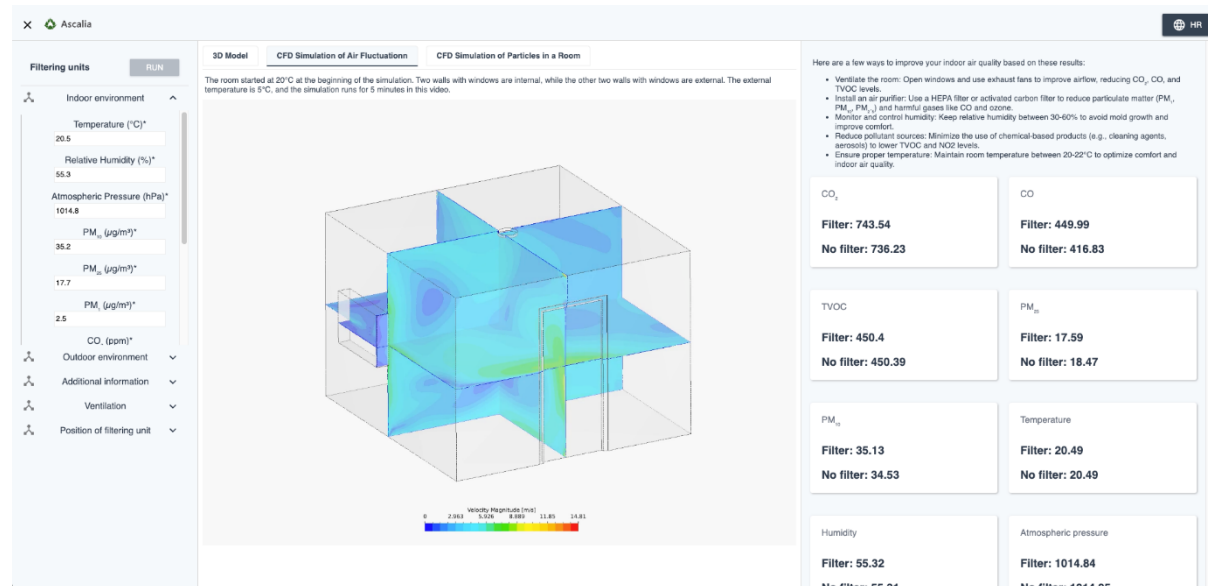


Figure 17 GUI with tab CFD Simulation of Air Fluctuation

Figure 18 shows the GUI and the second tab, which contains a video of the CFD simulation inside the room depicting particles in the air. The parameters for the simulation are displayed above the video, and the simulation runs in a loop.

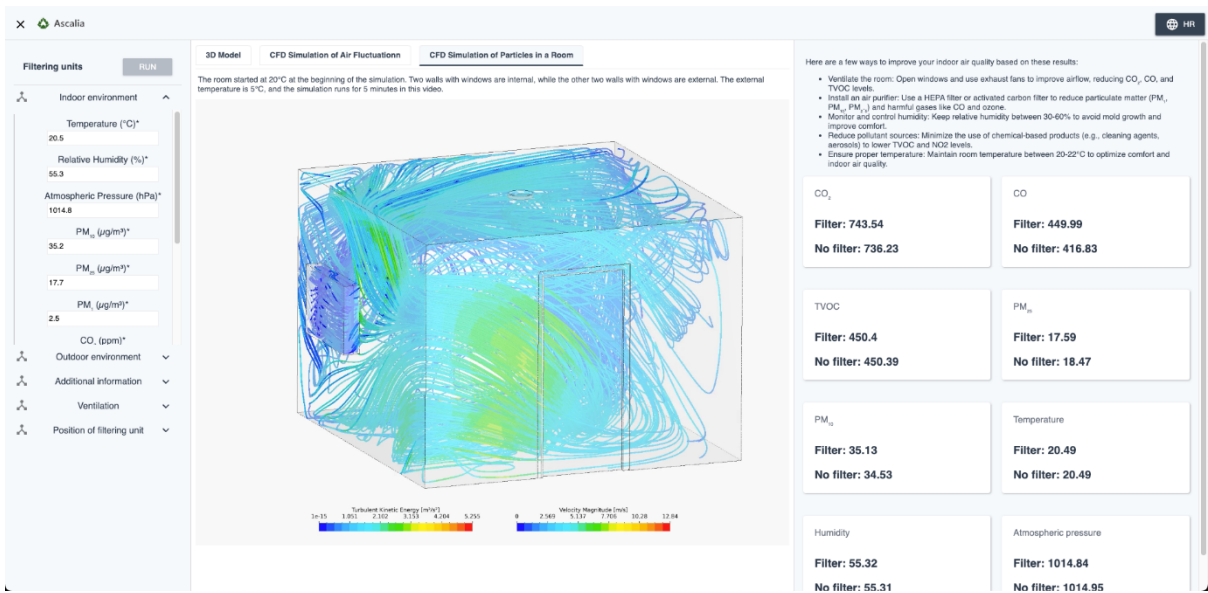


Figure 18 GUI with tab CFD Simulation of Particles in a Room

7 Conclusion

The development of predictive models was critical to the success of this project. After data preparation, model training, and evaluation, the model was implemented and demonstrated promising results in predicting air quality trends. The deployment of the digital twin, with its robust backend and frontend services, showcases the potential of continuous monitoring systems. Through architectural highlights, enhanced functionality, and a continuous delivery pipeline, the digital twin not only provides valuable insights but also enables real-time air quality monitoring. The functionalities embedded within the digital twin ensure its adaptability and reliability in various applications, offering a dependable tool for environmental monitoring. This project has set the foundation for future enhancements, with the possibility of further refining the digital twin to accommodate more complex scenarios and integrate additional sensors for a more holistic environmental analysis. The outcomes of this work contribute to the ongoing advancement of intelligent, real-time monitoring systems for indoor air quality.

This task plays a crucial role in the broader scope of the EDIAQI project, particularly in the Filtration Pilot, where the developed sensors and algorithms will be applied to evaluate the efficiency of filtration unit usage inside office spaces. Additionally, all software components used for capturing people's habits will be leveraged in this context to enhance the accuracy of air quality assessments.

For taxpayers, this task offers a practical tool to test and evaluate the optimal placement of filtration units in their homes. By utilising the digital twin, users can position the filtration unit in various locations within a virtual representation of their room and observe its impact on indoor air quality. This feature provides an intuitive and interactive way for individuals to make informed decisions about air purification strategies within their living spaces.

Furthermore, CFD simulations enable users to gain a better understanding of airflow dynamics, helping them visualize how air circulates throughout the room. This knowledge empowers users to improve their indoor air quality by optimizing the placement of filtration units and ventilation strategies.

In summary, this work is a significant step forward in integrating digital twin technology into real-world applications, supporting both research and practical implementation in indoor air

quality management. The insights gained from this project will inform future developments and contribute to the advancement of intelligent environmental monitoring solutions.



Deliverable D4.4

Deployed digital twin on consortium servers

Work Package 4

MONITOR: Pilots, Data and Campaigns

Version: Final



This project has received funding from the European Union's
Horizon Europe Framework Programme under grant agreement N° 101057497.