Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Data assimilation using Adversarial Neural Networks to help determine COVID infection risks in enclosed spaces

by

Shiqi Yin

Email: shiqi.yin21@imperial.ac.uk
GitHub username: acse-sy121
Repository: https://github.com/ese-msc-2021/irp-sy121

Supervisors:

Prof. Christopher Pain

Dr. Claire Heaney
Dr. Boyang Chen

September 2022

Acknowledgements

Firstly, I am very grateful for the help of my supervisors. Professor Pain often provides constructive advice on the direction of the project, Claire gives me plenty of help with the code and Latex syntax correction and Boyang gives me plenty of help with the dataset and the use of the Paraview software. Secondly, I would like to thank my teammates Jintao (from CP110), Donghu and Jieyi (from CP111) for their help with algorithmic ideas and code. Finally, I would like to thank all my friends from K table of RSM-1.51 for their encouragement and help during a very difficult time.

Thanks to the real data from Prof. Prashant Kumar at Surrey University.

Abstract

Research on the transmission of COVID-19 is vital as the epidemic remains severe in countries worldwide. A common prediction model currently available is the Computational Fluid Dynamics (CFD) model, which accurately predicts the spread of the virus in the air. However, due to the high computational cost of CFD models, the models require a long running time. Therefore, finding a less computationally expensive model for prediction is essential. The objectives of this project are to predict data over time series for different fields such as CO_2 and viruses in enclosed space using an adversarial autoencoder (AAE) and apply data assimilation from the predictions generated by the DA-Pred-AAE model to optimise the results. To achieve the above objectives, the Pred-AAE and DA-Pred-AAE models are proposed, and data assimilation of the predictions generated by the DA-Pred-AAE model is carried out to optimise the results. Further, the model can predict CO_2 concentration levels and viral load data at human locations to determine COVID-19 infection risks. Thus, this project provides an alternative to CFD models.

Keywords: Data Assimilation, Adversarial Autoencoder, POD, Prediction, COVID-19, Machine Learning

Contents

1 Introduction

1.1 Background

COVID-19 is an outbreak that was first detected and broke out in Wuhan, China, in late December 2019 [12]. This epidemic is pneumonia caused by the SARS-CoV-2 virus. Due to multiple mutations of the virus, the pathogenicity of the virus decreased, but its infectivity increased significantly. However, the negative impact of the epidemic on the world is still severe, affecting hundreds of countries and retarding their economic development[8].

In a study of infected people [18], researchers found that SARS-CoV-2, the causative agent of the new infectious disease, has a strong capacity for person-to-person transmission. According to the report from WHO [1], COVID-19 can be transmitted by droplets, by touching the eyes, mouth or nose after contact with objects carrying the virus, or in poorly ventilated rooms[21]. The virus's ability to transmit allows healthy people to become infected in an enclosed space. The study of the transmission of COVID-19 is therefore significant for outbreak prevention and control. One such study is to determine COVID-19 infection risks in enclosed spaces by studying its transmission.

For the transmission of COVID-19 in enclosed spaces, aerosol transmission is a prevalent mode of virus transmission. In Björn Birnir's paper[5], he modelled the airflow in a restaurant and by analysing the changes in virus concentration levels in this enclosed space, and concluded that the transmission of COVID-19 in the restaurant was due to an increase in the concentration level of virus-carrying droplets produced by aerosols carrying the SARS-CoV-2 virus in the air. When people stay in such enclosed spaces for long periods, they can become infected[10]. Therefore, aerosol transmission is one of the primary modes COVID-19 transforms in enclosed spaces, and it can be simulated by modelling aerosol transmission. As aerosol transmission is similar to the transmission of CO_2 from human respiration in enclosed spaces[29], this project uses a model of CO_2 transmission in enclosed spaces to simulate the transmission of COVID-19.

1.2 Literature Review

With the global outbreak of the COVID-19 epidemic, several scientific departments have begun investigating the transmission of the epidemic. One method is Computational Fluid Dynamics (CFD), which is used in various applications, including the modelling of the transmission of the virus. In Khalid M. Saqr's paper[24], he used CFD to model the transmission of COVID-19 in the air and proposed a reproducibility index for the COVID-19 model. In the paper by Peng et al.[23], they used CFD to model the transmission of pathogens in the air by simulating the dilution and diffusion of droplets from the mouth and nose of people after sneezing. In addition to the simulations, they use the model to predict the airborne transmission of pathogens and to find and propose ways to reduce the risk of transmission. Although CFD models can simulate the transmission of COVID-19 in the air with high precision, the high computational cost makes CFD models compute slowly [24, 27].

In terms of prediction, machine learning can also generate good prediction results. Therefore neural networks can be used to predict the transmission of COVID-19 in the air. One of the neural network models that can be used is the autoencoder (AE). Yann LeCun proposed the concept of autoencoder in 1987[14]. It consists of an Encoder and a Decoder. The encoder encodes the input data, and then the encoded data is reconstructed by the decoder after decoding. Makhzani et al. [16] mentioned in their paper that adversarial autoencoder (AAE) is a combination of AE and GAN, and it is easier to train. The main difference between AAE and GAN is the input to D. In AAE, the input to D is the latent space data generated by the encoder in the autoencoder part and the prior[3]. The core part of training AAE is similar to GAN, which is adversarial training between encoder and discriminator. Then, the decoder can be well trained by training the autoencoder part of the AAE model and generating reasonable data[17].

Several existing research studies use machine learning models to perform predictions based on time series. In the paper by J. Chen et al.[7], researchers proposed the EnsemLSTM model based on the LSTM (Long Short Term Memory neural networks) model by combining several optimisation algorithms. The study used wind speed data to train the model and made predictions based on time series to obtain wind speed forecasts for the next 10 minutes to an hour. The conclusion shows that the model achieved a better prediction performance.

In order to improve the performance of prediction models, data assimilation is used to optimise results arising from machine learning in prediction. In the paper published by S. Lee et al.[15], data assimilation was used to improve the performance of air quality prediction in South Korea. In the research, they used 3D-VAR to optimise the prediction of temporal and spatial variation in ground-level PM concentrations to improve the model's performance for predicting PM concentrations in areas lacking observation data.

1.3 Description of Problem and Objectives

In the existing studies, there are few studies on data assimilation for AAE models and few on data assimilation for multiple fields in a dataset by combining observation data in the prediction process. Modelling the transmission of COVID-19 in the air is complex since air movement varies under different indoor conditions. Although Khalid M. Saqr had implemented CFD models to simulate the transmission of COVID-19[24], machine learning is also required to predict the transmission of COVID-19 due to the high computational cost of CFD models. Therefore, the objective of this project is to build an AAE model to predict the transmission of COVID-19 in enclosed spaces and use data assimilation to optimise the prediction results using the observation data of multiple fields.

The following part is what this project implemented. The dataset first needs to be pre-processed, and this is because the dimensionality of the input data is high, and the model cannot accept data with such high dimensionality. After pre-processing the data, AAE models are built and trained using the data mentioned above. Then the prediction algorithm is used to obtain predicted data. Subsequently, data assimilation is applied to the model, the above steps are repeated, and the differences in precision between the predicted values in the two cases are discovered. Three different scenarios are implemented to test the functionality of data assimilation.

Finally, the project implemented the prediction of different data fields at future time levels by the AAE model and improved the prediction results of the model by implementing data assimilation. Furthermore, the AAE model can predict the total viral load and CO_2 concentration level near the heads of the 27 people in the classroom.

2 Methodology

This project replicates the prediction algorithms and data assimilation methods from Silva et al.'s paper[26].

2.1 Dimensionality reduction

A method used for data pre-processing is Reduced-order modelling (ROM). In Marco Fahl and Ekkehard W. Sachs's paper[9], they mentioned that Reduced-order modelling is required to decrease computational difficulties, and one way of implementing reduced-order modelling is proper orthogonal decomposition. In this project, PCA is selected to pre-process the spatial data, which will be used to train the AAE models.

2.2 Prediction using AAE (without sensor data)

In this section, the Pred-AAE model is proposed to perform predictions for the dataset without sensor data in time series after data pre-processing. For a Pred-AAE model trained with m continuous time levels, the output of AAE model $\widetilde{\Psi}^n$ is in the following form:

$$\tilde{\Psi}^n = \begin{pmatrix} \left(\alpha^{n-m+1}\right)^T \\ \left(\alpha^{n-m+2}\right)^T \\ \vdots \\ \left(\alpha^{n-1}\right)^T \\ \left(\alpha^{n}\right)^T \end{pmatrix}, \tag{1}$$

where $\left(\alpha^n\right)^T = \left[\alpha_1^n, \alpha_2^n, \ldots, \alpha_{N_{POD}}^n\right]$, N_{POD} represents the number of POD coefficients and α_i^n represents the i^{th} POD coefficient at time level n.

Autoencoders are used more for generating relevant images or results. Therefore, the input and output of the autoencoder need to be modified due to the difference between the input and output of the autoencoder and the solutions required for this project. The input data for this project when training the adversarial autoencoder is m consecutive time series, so for prediction, the data at the first $m-1$ time levels need to be known to predict the value at time level m. e.g. Suppose that there exist data at time levels [0, 1, ... , m-1], then the AAE model can predict the value at time level m. If it is required to predict the value at time level $m+1$, time level [1, 2, ... , m-1] and the newly predicted value at time level m will be used to predict the value at time level $m+1$. Figure 1a shows how the prediction algorithm works when using Pred-AAE to make predictions.

Suppose the solution between time levels 0 and $m-2$ for POD coefficients are known and it is denoted by $\left\{\tilde{\boldsymbol{\alpha}}^k\right\}_{k=0}^{m-2}$, the AAE model will predict the future results by the following steps:

1. Make initial guess at time level $m-1$ with result at time level $m-2$, which will generate a solution:

$$X^n = \begin{pmatrix} \left(\alpha^{0}\right)^T \\ \left(\alpha^{1}\right)^T \\ \vdots \\ \left(\alpha^{m-3}\right)^T \\ \left(\alpha^{m-2}\right)^T \\ \left(\alpha^{m-2}\right)^T \end{pmatrix}. \tag{2}$$

2. Put X^n into trained AAE model to get the result for $\tilde{X}^n$.
3. Update the solution at the last time level.
4. Repeat step 3 until the solution converges, and output the solution as a prediction at the last time level.
5. Add the last time level solution to the end of the model input data and remove the value of the first time level of the input data.
6. Generate a new X^n, where the initial guess of the last time level is the solution in step 5.
7. Repeat step 2 to step 6 until the final time level value is predicted.

Note that the initial guess is important for generating reasonable predictions, so setting the initial guess at the last time level to the value at the penultimate time level will obtain a more accurate prediction.

2.3 Data Assimilation with AAE (with observation data)

In this section, the DA-Pred-AAE model is proposed based on Pred-AAE mentioned in section 2.2. For a DA-Pred-AAE model trained with m continuous time level, the output $\widetilde{\Psi}^n$ is in the following form:

$$\widetilde{\Psi}^n = \begin{pmatrix} \left(\alpha^{n-m+1}\right)^T, \left(\mu^{n-m+1}\right)^T \\ \left(\alpha^{n-m+2}\right)^T, \left(\mu^{n-m+2}\right)^T \\ \vdots \\ \left(\alpha^{n-1}\right)^T, \left(\mu^{n-1}\right)^T \\ \left(\alpha^{n}\right)^T, \left(\mu^{n}\right)^T \end{pmatrix}, \tag{3}$$

where $(\alpha^n)^T = \left[\alpha_1^n, \alpha_2^n, \ldots, \alpha_{N_{POD}}^n\right]$, $(\mu^n)^T = [\mu_1^n, \mu_2^n, \ldots, \mu N_{\text{sensors}}^n]$, N_{POD} represents the number of POD coefficients, α_i^n represents the i^{th} POD coefficient at time level n, N_{sensor} represents the number of sensors and μ_i^n represents the i^{th} sensor at time level n. For DA-Pred-AAE, the same prediction algorithm is used for $\widetilde{\Psi}^n$.

In order to perform data assimilation, observation data is used in the inputs to the model and predictions are made on the observation data. When performing data assimilation, the following equation is used to update the observation data:

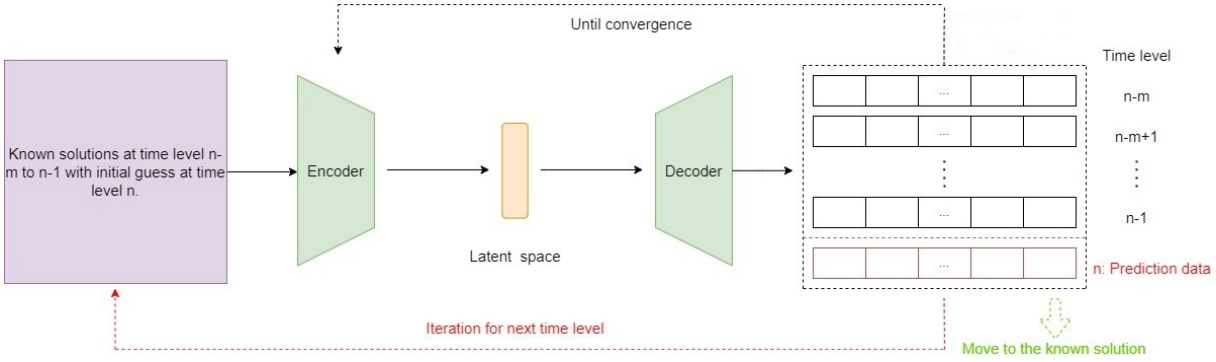
$$\text{Combined obs data} = (1 - weight) \times \text{Predicted obs data} + weight \times \text{Real obs data}, \tag{4}$$

where $weight \in [0, +\infty)$, if $weight = 0$, no real observation data will be considered and if $weight = 1$, the predicted observation data will be replaced by real observation data. If $weight > 1$, this will be over relaxation data assimilation.

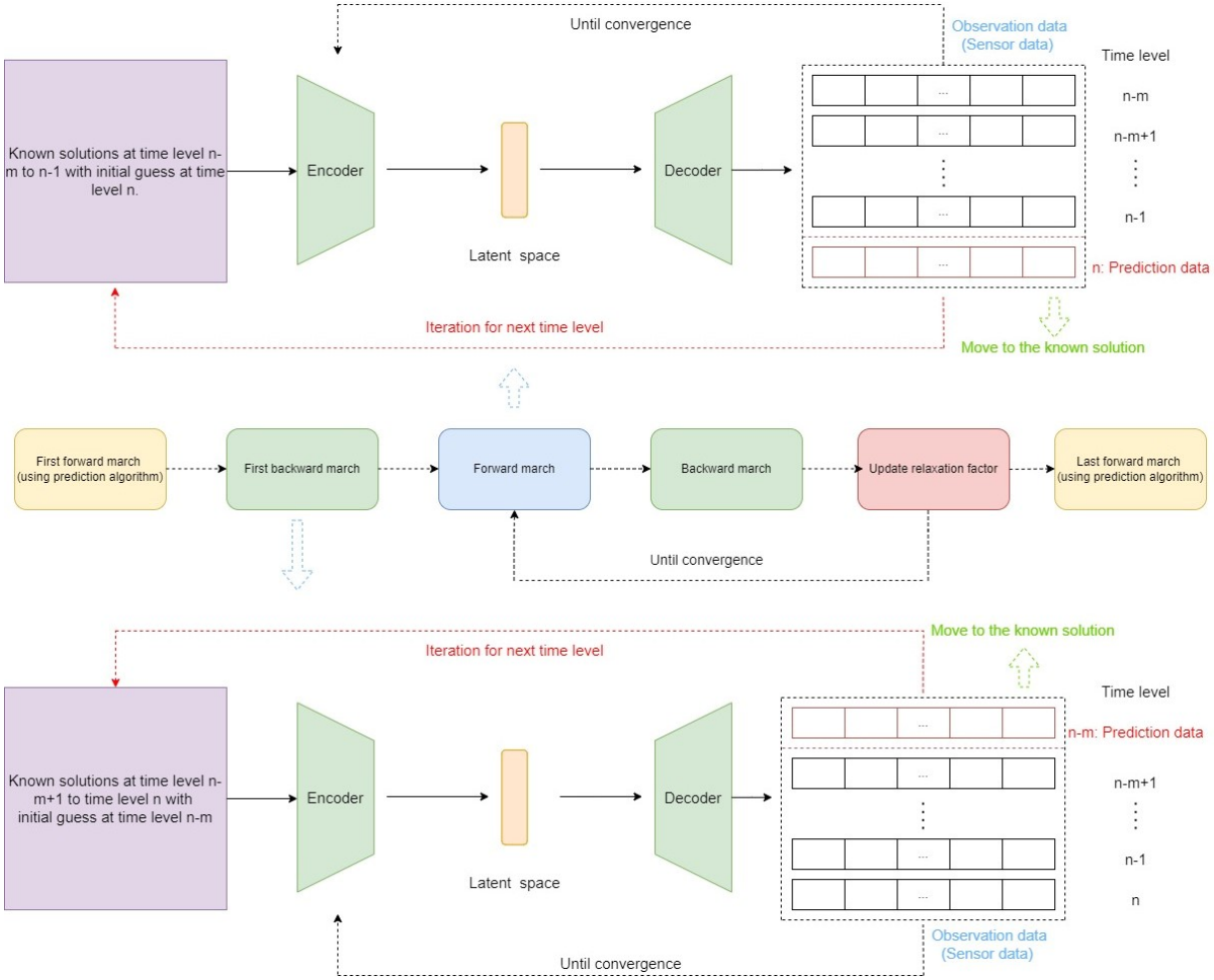
Data assimilation will follow the following steps:

1. Predictions are made using the prediction algorithm mentioned in section 2.2 until the data at the last time level is predicted.
2. Select prediction results of the last $m-1$ time levels obtained in Step 1, perform a backward march to predict the sensor and POD coefficient data at time level $n-m$ and repeat the above prediction until the data for the first time level is predicted. While predicting the observation data for each time level, the predicted observation data for the random time level and the actual observation data are computed with a certain weight to generate new observation data μ and replace the predicted observation data. Eq.(1) is used to compute the newly generated observation data.
3. Select prediction results of the first $m-1$ time levels obtained in step 2, repeat Step 1 and replace the predicted observation data with new observation data calculated using Eq(1).
4. Select prediction results of the last $m-1$ time levels obtained in step 3 and repeat Step 2.
5. Apply relaxation to change the weight for real observation data.
6. Repeat Step 3 to Step 5 until convergence.
7. Select prediction results of the first $m-1$ time levels obtained in step 6, and repeat step 1 to obtain the prediction after data assimilation.

Figure 1b illustrates the basic flow of data assimilation and the algorithm.



(a) AAE Prediction Algorithm



(b) Progress of data assimilation

Figure 1: Description of Prediction and DA algorithm. (a) shows how AAE predict value for one time level and (b) shows how data assimilation is applied to the prediction data and how data assimilation algorithm output its results.

2.4 Description of Dataset

The dataset used for this project is provided by CO-TRACE EPSRC Covid-19 Infection Risk in schools consortium. It is a set of 192060 points in 3D space generated by a CFD model where sensor data

was collected in a classroom with a Dyson fan. The layout of the classroom, the location of the sensors and the position of the 27 individuals are shown in Part one of the Appendices. The collected data are CO_2 concentration levels, temperature and humidity fields. The dataset consists of 720 time levels in one-hour period, each containing seven fields of CO_2 concentration level, velocity in x, y and z axes, temperature, humidity and virus generated by the CFD model. Therefore, PCA is required to reduce the dimension of the dataset. After applying PCA for the 3D data, the dimension for the dataset is reduced to 150. For the observation data used in data assimilation, the data for the seven fields are extracted from the dataset by entering the coordinate values of each of the 18 sensors.

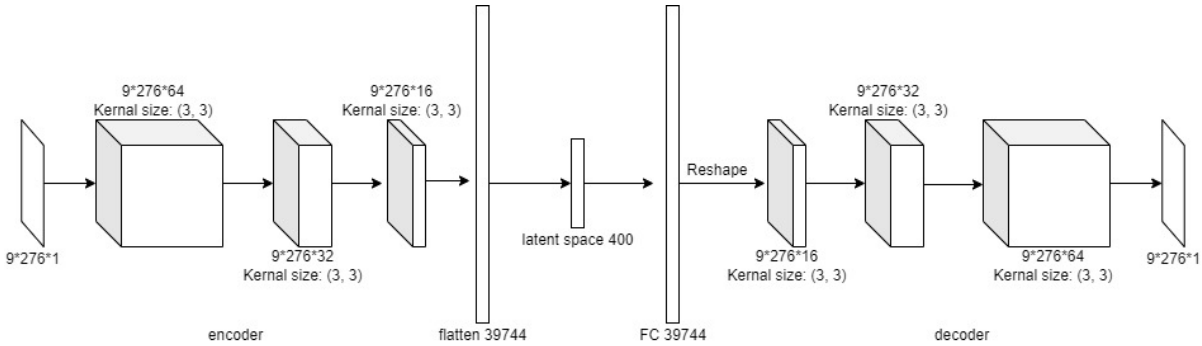
2.5 Description of DA-Pred-AAE model

DA-Pred-AAE uses 150 POD coefficients generated after dimensionality reduction and seven fields of observation data for a total of 18×7=126 parameters. For a continuous time series, the input data to the model is 9 consecutive time levels of data. The first 8 time levels of data are used to predict the data at 9^{th} time level. Thus, the input to the model is the data with a shape of (9, 276, 1). Figure2 shows the AAE model's architecture and the model's training process.

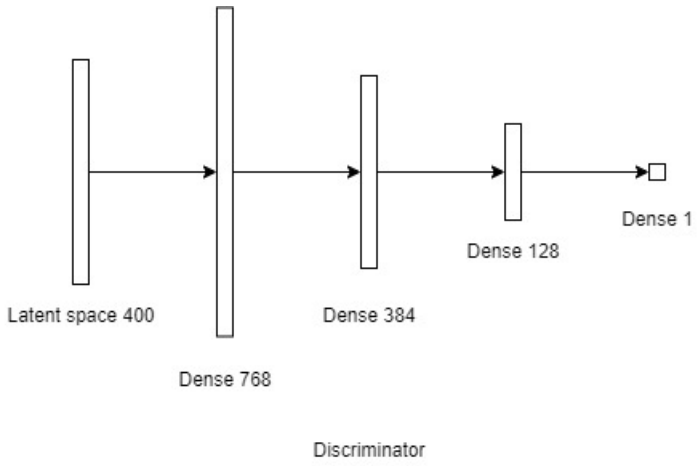
In terms of the model architecture, the model is divided into Autoencoder and Discriminator part. For the Autoencoder, it is shown in Figure 2a that the model requires data with a shape of (9, 276, 1) as input, which is encoded by the Encoder and fed into a fully connected layer of 400 neurons, i.e. the latent space. The data in the latent space is then decoded by the Decoder, and a shape of (9, 276, 1) is output. For Discriminator, as shown in Figure 2b, the model takes the latent space data generated by Encoder as input and outputs 0 or 1 for Fake or Real. Figure 2c shows the training process of DA-Pred-AAE, which first trains the Autoencoder part, followed by Adversarial training of Encoder and Discriminator. Finally, the Decoder is trained until the model converges.

The following part shows the hyperparameters used in this model:

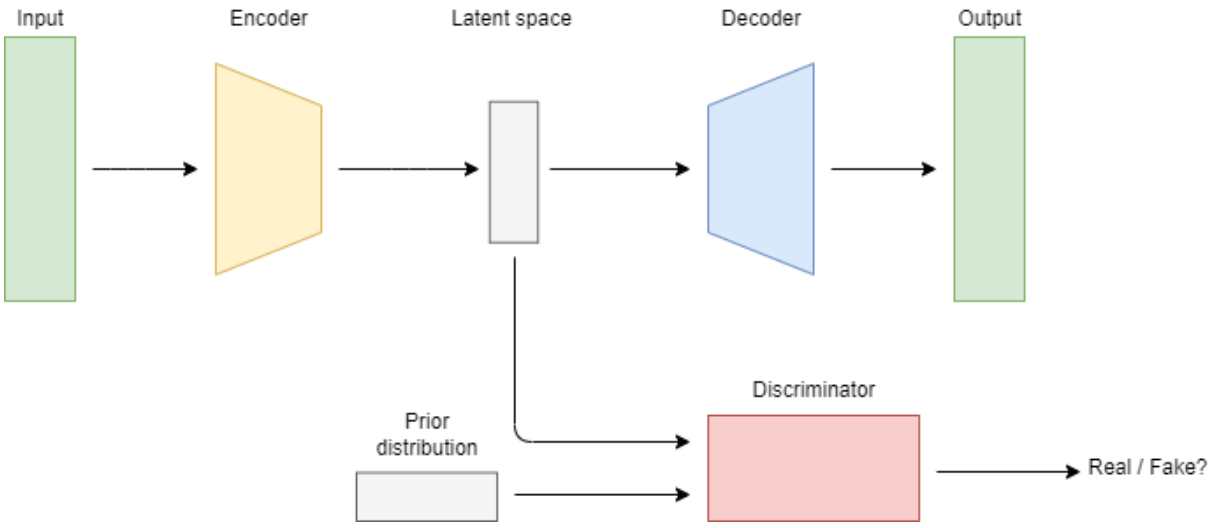
- Epoch: 10000
- Batch size: 128
- Latent space size: 400
- Step: 1
- Input_time levels: 9
- Optimizer for Autoencoder a_optimizer: Adam
- Optimizer for Discriminator d_optimizer: Adam
- Optimizer for Decoder g_optimizer: Adam
- Learning rate for a_optimizer: 1e-4
- Learning rate for g_optimizer: 1e-4
- Learning rate for d_optimizer: 5e-5
- Acitvation function for Discriminator: relu
- Loss function for Autoencoder: mse
- Loss function for Discriminator: cross-entropy
- Loss function for Decoder: cross-entropy



(a) Architecture of DA-Pred-AAE



(b) Architecture of Discriminator



(c) Process when training AAE

Figure 2: Description of AAE and data assimilation. (a) shows the architecture of DA-Pred-AAE model, (b) shows the architecture of discriminator and (c) shows the process when training DA-Pred-AAE model.

3 Code metadata

The project mainly used Google Colab as the compiler, with some of the experimental data being acquired on a local computer. The code used can be accessed at the following link:

https://github.com/ese-msc-2021/irp-sy121

The source folder contains all the code used in the project. The output data and images are stored in Google Drive and can be accessed as described in Readme. The following part is what the source folder contains:

- data_prepocessing: The code in this folder is used to extract the 3D spatial nodes data and observation data from the vtu file, process the data using min max scalers, reduce the dimension of the 3D spatial data using PCA and save the processed data and scalers.
- Experimental_data: The code in this folder is used to extract observation data from the local dataset, pre-process the observation data and save the pre-processed data and scalers. This folder also contains the dataset used to get the experimental data and the output data.
- Pred-AAE: Used to generate a model for prediction data (no sensor data), train the model and use that for prediction.
- DA-Pred-AAE: Generate a model to perform prediction and data assimilation. Use training set to train the model and use the model for predicting data and perform data assimilation using different test cases.
- vtu_files: This folder contains all the files used to get node data in vtu files.
- readme_files: This folder contains all the files used in readme.md.
- Readme: Introduce how the codes for this project can run and how to download codes and files generated by these codes.

The system environment requirements for Colab are shown in the following part:

- Platform: Google Colab Pro +
- Language: Python (Version $\geq$ 3.5)
- Hardware acceletator: GPU (T4 or P100 from NVIDIA)
- RAM required: 51.00GB
- ROM required: 100GB or more
- Required packages:
  - NumPy [11]
  - Scikit-learn [13]
  - Tensorflow [22]
  - keras [19]
  - Matplotlib [4]
  - vtk (pip install required) [25]
  - vtktools.py and tool_io.py (provided)

The system environment information for local machine is shown in the following part:

- Platform: Windows 10 (Version = 19044.1889)
- IDE: VScode
- Language: Python (Version = 3.9)
- Hardware acceletator: CPU (Intel Core i7-11800H)

- RAM: 32 GB
- Required packages:
  - NumPy [11]
  - xlrd (Version = 1.2.0)
- Required software: Paraview (Version = 5.10.1) [2]

Note that Paraview is used to visualise the dataset and the output for predictions and data assimilation results which are stored in vtu files.

4 Results

4.1 Dimensionality Reduction

In this project, the data in the dataset used is 192060 dimensions per field, and the number of fields that will be used is 7, so it is not possible for the Pred-AAE and DA-Pred-AAE models to receive input data of such large dimensions. Therefore, dimensionality reduction of the dataset using POD is necessary. As a result, 90.2% of the explained variances are retained after applying PCA (mentioned in section 2.1) to reduce the dimensionality of the dataset to 150 principal components. Figure 3 below shows the result of PCA.

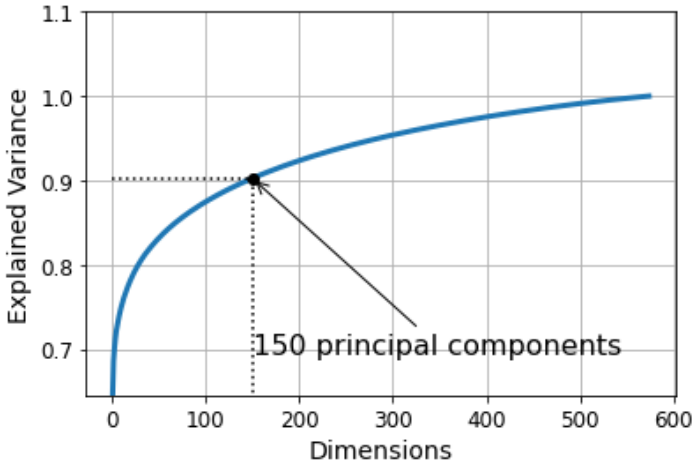


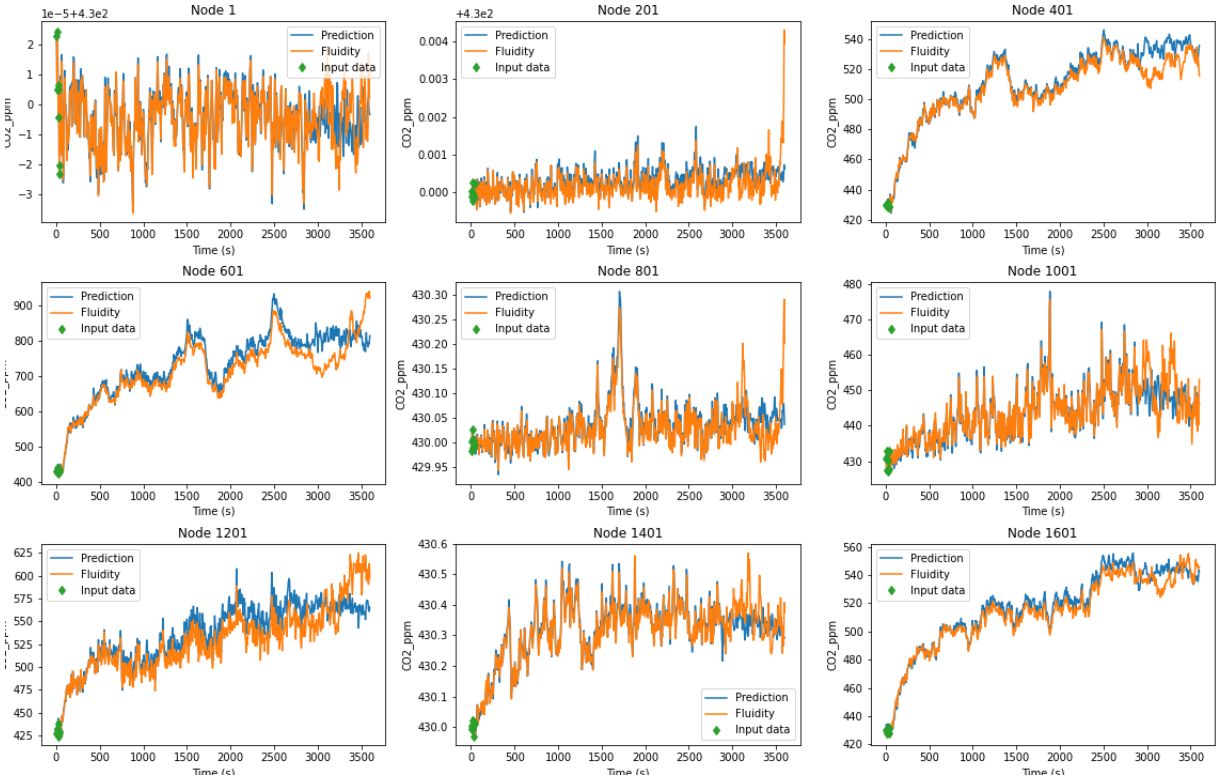
Figure 3: Result after Dimensionality Reduction using PCA. Here the data is reduced to 150 dimensions with 90.2% explained variance is kept.

4.2 Prediction using Pred-AAE model

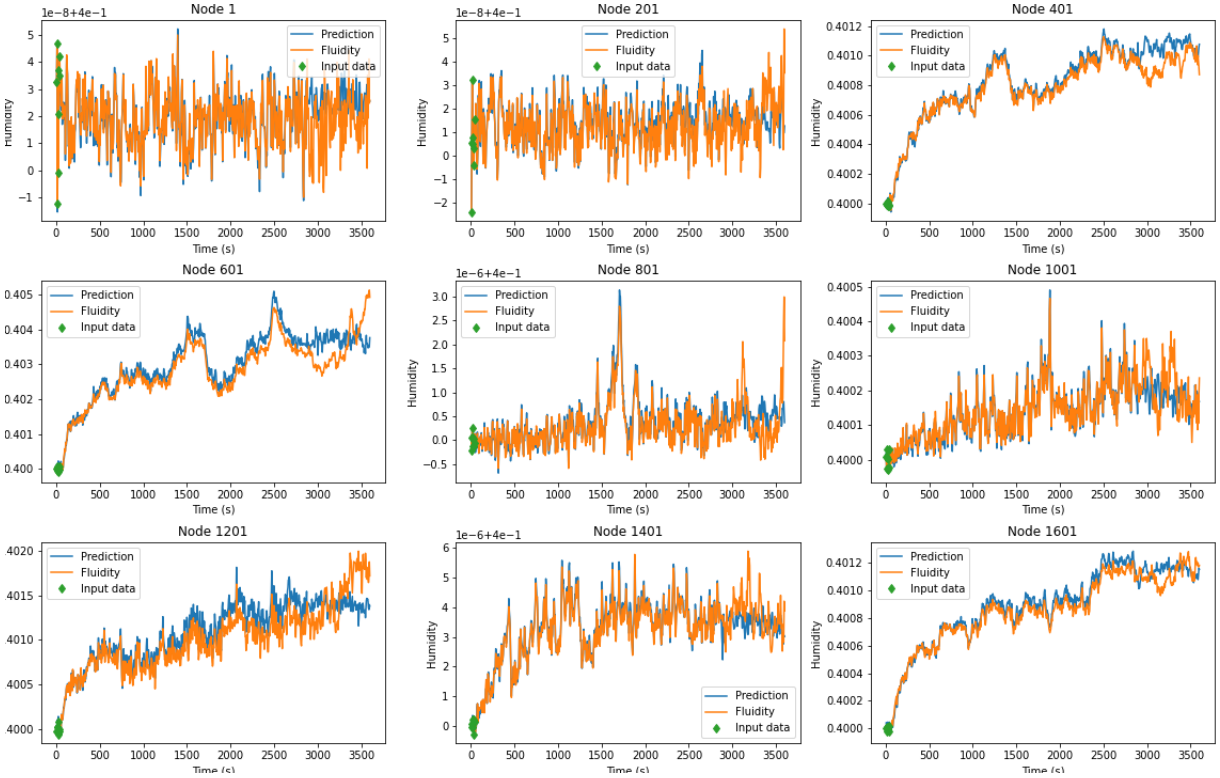
This section uses Pred-AAE to predict CO_2 concentration levels, velocity components along the x, y, and z axes, humidity, temperature, and viral load data. The data used for prediction was generated by the CFD model, normalised by Min Max Scalers (mentioned in section 1.3) [6] and applied PCA [20], which can reduce the dimension to generate data that could be used as input to the model. In order to be able to implement predictions along a time series, the data input to the model needs to be encapsulated as a time series. The Pred-AAE model accepts data with an input shape of (9, 150, 1). Therefore, the model predicts the 9^{th} time level from the first 8 time levels and, by specifying the number of iterations, uses the previous input data and the newly predicted data as known data for the next time level until the data for the last time level is predicted.

After the prediction is completed, the predicted data is transformed into 7 fields of 192060 spatial nodes each by $PCA.inverse\_transform$ and $MinMaxScaler.inverse\_transform$ and compared

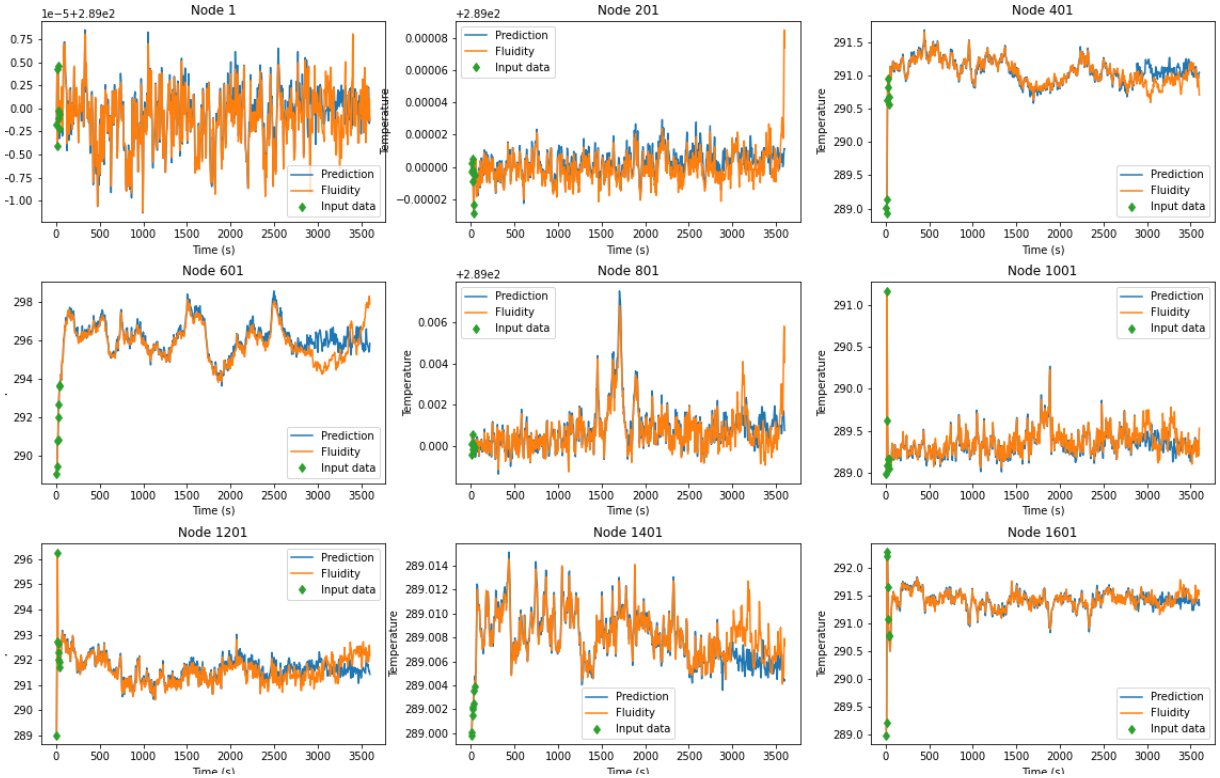
to the data generated by the CFD model. Due to the presence of a large number of nodes, 9 of these nodes are selected for the project to compare their predicted data with the corresponding data generated by the CFD model, which is shown in Figure 4.



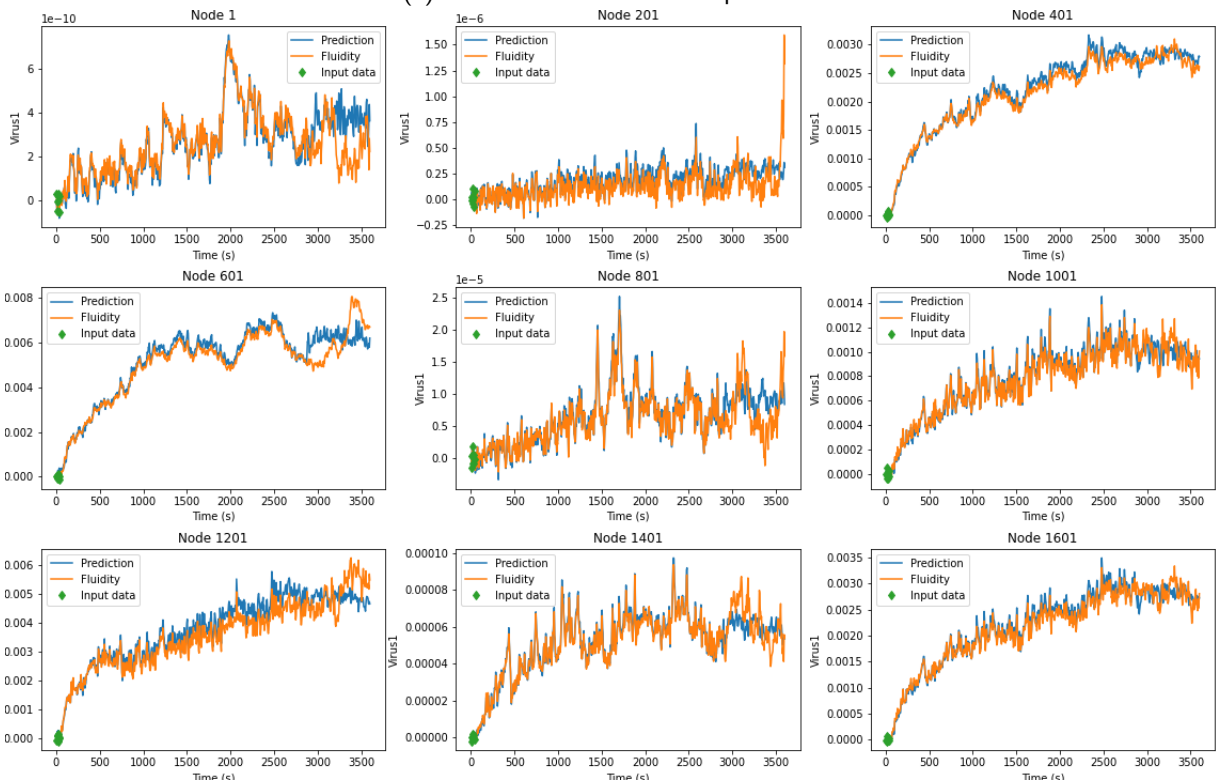
(a) Prediction results of CO_2 concentration level



(b) Prediction results of Humidity



(c) Prediction results of Temperature



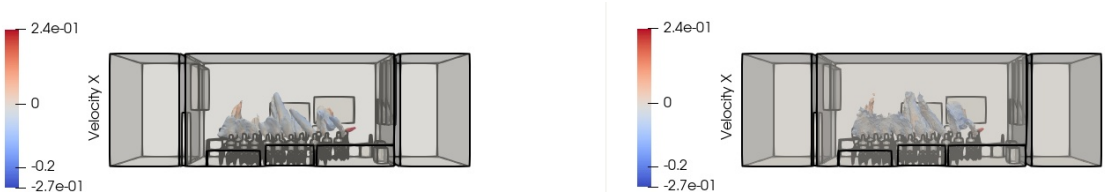
(d) Prediction results of viral load

Figure 4: Prediction results. (a) shows the prediction results of CO_2 concentration level, (b) shows the prediction results of Humidity, (c) shows the prediction results of Temperature and (d) shows the prediction results of viral load.

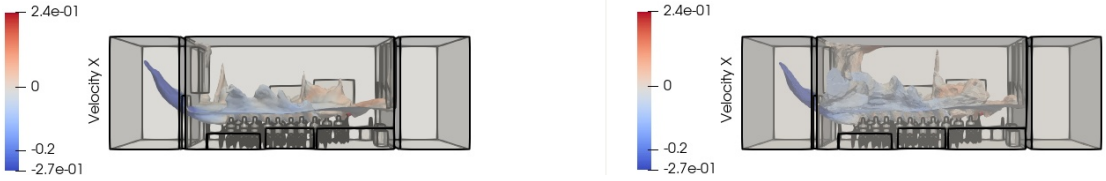
From Figure 4, the predicted data for the different fields are presented in 4 different subplots. (a)

indicates the concentration level of CO_2, (b) represents humidity, (c) represents temperature and (d) represents viral load. Here nodes 1, 201, 401, 601, 801, 1001, 1201, 1401 and 1601 are selected to show how the predicted results differ from the results generated by the CFD model. This result is generated by predicting the data for the last 712 time levels of the entire dataset. The first 80% of this dataset is the training set, and the last 20% is the test set. As can be seen from the figures, the model is able to predict the data for each field in the training set part, but in the test set part, the predictions of the model have a little diversion from those generated by the CFD model.

As the above data does not visualise the effect of the predictions in 3D space, Paraview is used to generate equivalent surfaces for the different fields in space using $contour$. Figure 5 shows the equivalence surfaces or outer surfaces generated by Paraview, where (a) - (c) indicate the concentration level of CO_2 coloured by velocity in x-axis, (d) - (f) represent for humidity coloured by velocity in x-axis, (g) - (i) represents for temperature and (j) and (l) represents for viral load.

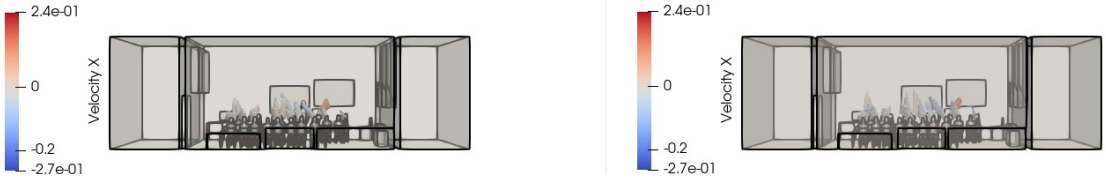


(a) Visualisation results of CO_2 concentration level at time = 215s

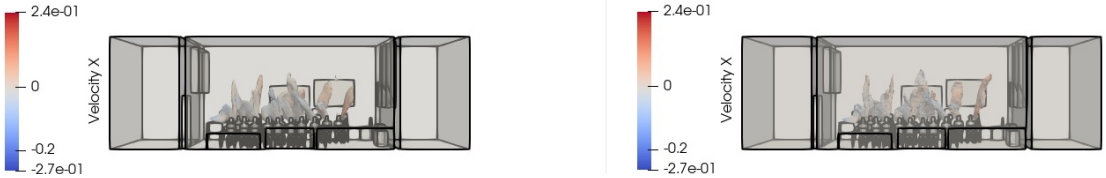


(b) Visualisation results of CO_2 concentration level at time = 1625s

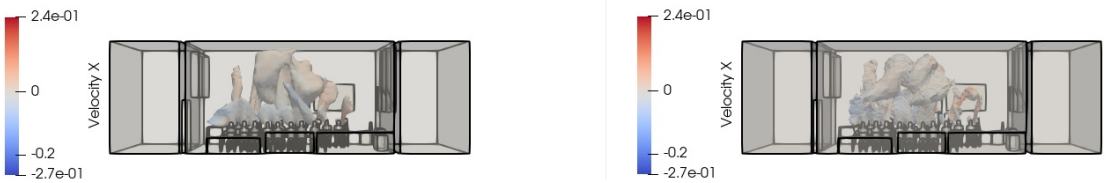
(c) Visualisation results of CO_2 concentration level at time = 3305s



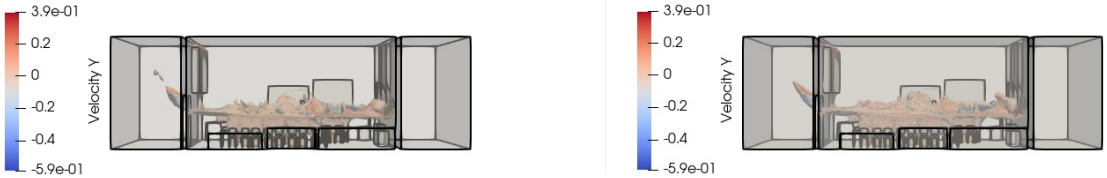
(d) Visualisation results of Humidity at time = 215s



(e) Visualisation results of Humidity at time = 1625s

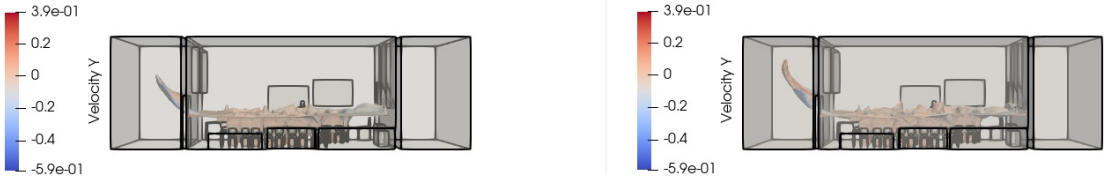


(f) Visualisation results of Humidity at time = 3305s



(g) Visualisation results of Temperature at time = 215s

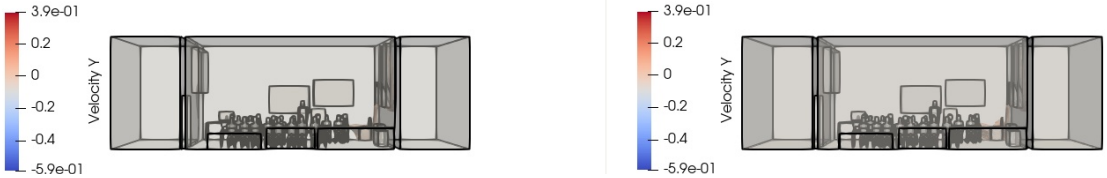
(h) Visualisation results of Temperature at time = 1625s



(i) Visualisation results of Temperature at time = 3305s

(j) Visualisation results of viral load at time = 215s

(k) Visualisation results of viral load at time = 1625s



(l) Visualisation results of Temperature at time = 3305s

Figure 5: Visualisation results at time = 215s, 1625s and 3305s. (a) to (c) shows the visualisation results of CO_2 concentration level (value = 1000 and coloured by Velocity in x-axis), (d) to (f) shows the visualisation results of Humidity (value = 0.41 and coloured by Velocity in x-axis), (g) to (i) shows the visualisation results of Temperature (value = 302 and coloured by Velocity in y-axis), (j) to (l) shows the visualisation results of viral load (value = 0.01 and coloured by Velocity in y-axis). Note: The left figure is for Prediction results, and the right figure is for data from the CFD model.

4.3 Prediction and data assimilation using Pred-AAE model

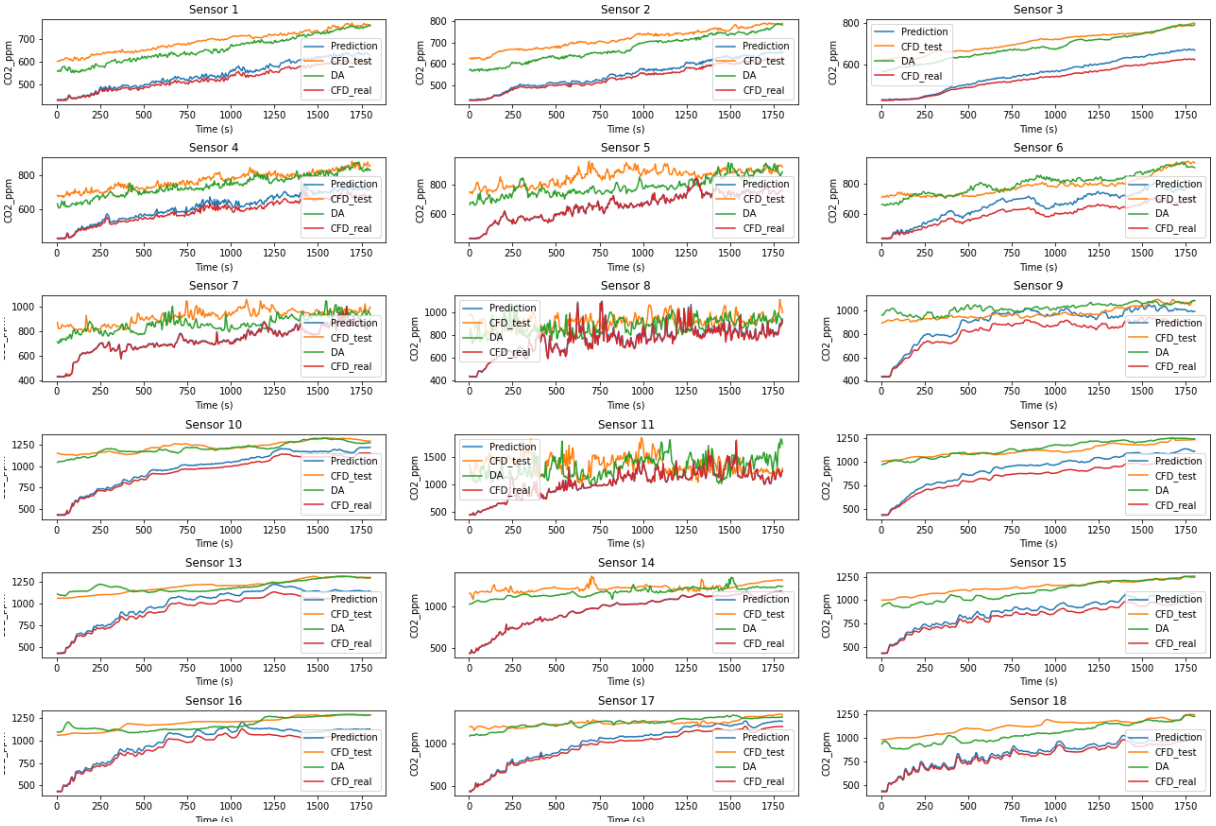
This section uses DA-Pred-AAE (mentioned in section 2.3) is used to complete the basic data assimilation (0s-3600s) and three scenarios. These three scenarios are:

1. Apply data assimilation at the middle (i.e. setting the observation data to start from the middle of the simulation).
2. Apply data assimilation using the observation data for the test set.
3. Apply data assimilation using experimental data (120 time levels in total).

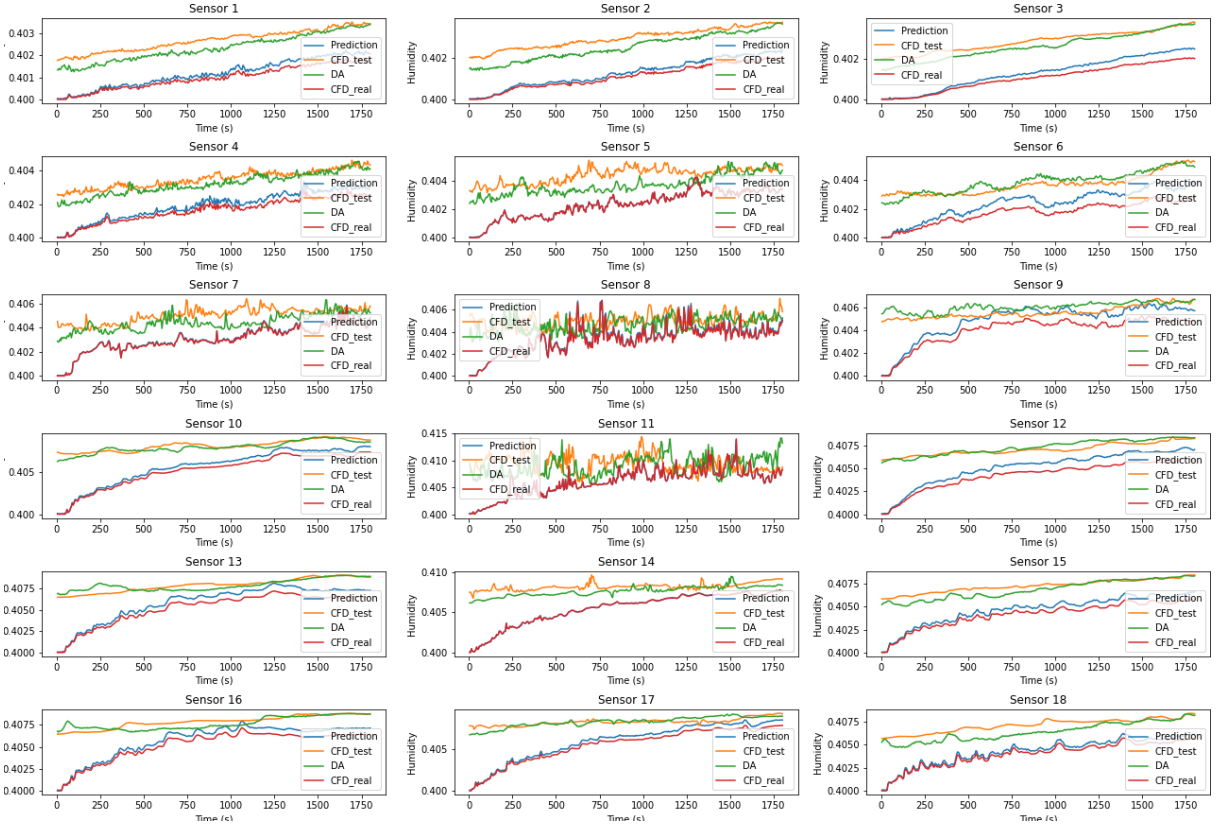
For basic data assimilation, 650 time levels are selected out of 720 time levels for data assimilation. The observation data used for data assimilation are obtained by entering the coordinates of 18 sensors in the dataset. To implement data assimilation, a forward march, a backward march, multiple iterations of the forward and backward march until convergence and a final forward march are performed to obtain the converged data assimilation results. In order to control the weight of observation data in the data assimilation, the project uses weight in the data assimilation process (mentioned in section 2.3). Here the weight of the observation data is initialised to 1 and is dynamically adjusted as the iteration progresses by performing a relaxation. The result figures are shown in appendices (take CO_2 concentration level as an example).

4.3.1 Scenario 1

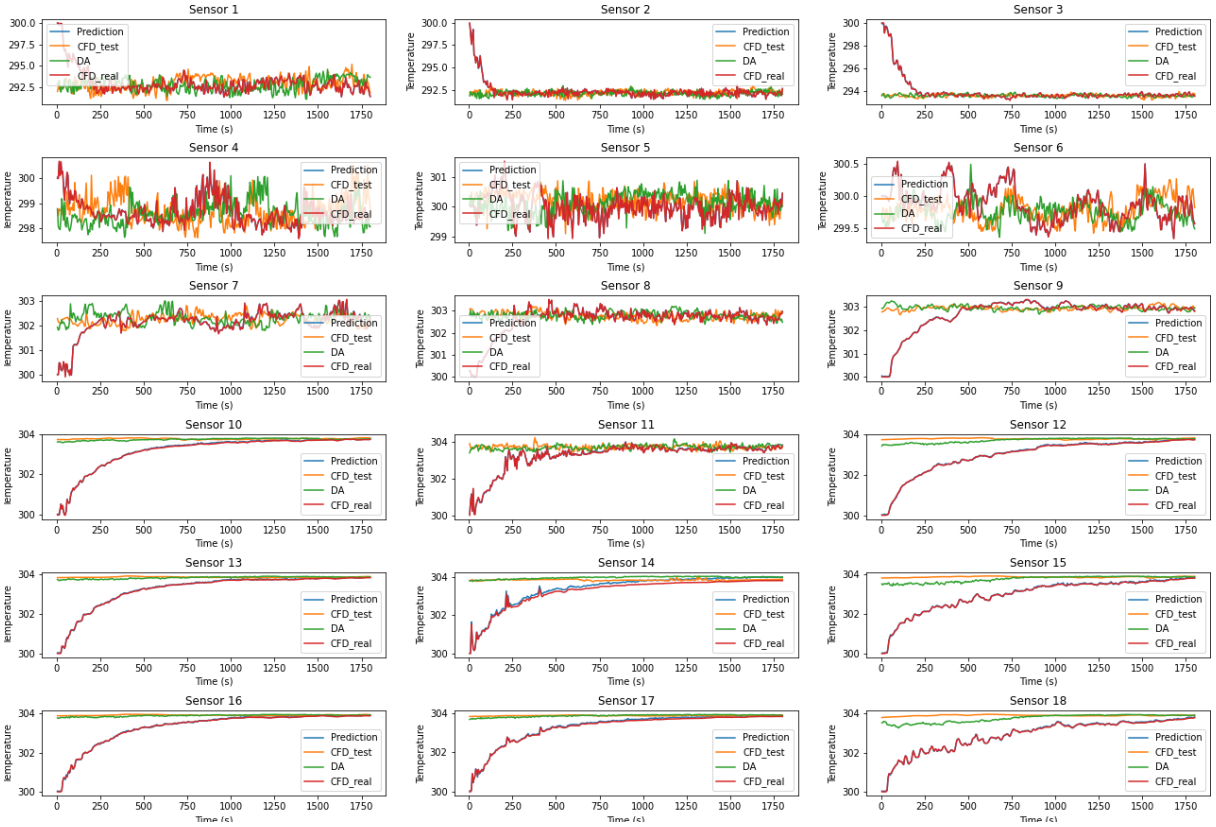
In this section, the data used for data assimilation is the second half of the entire data generated by the CFD model, i.e. time level ≥ 361 (start from 1). This part of the data will be used to perform data assimilation for the predicted data starting at time level $= 1$, and the weight used for the data assimilation is initialised to 70 and the weight will not change in the iterations. Figure 6 shows the results of data assimilation for the seven different fields of sensor and node in scenario 1, where (a) - (g) show the predicted data for the seven fields. (h) - (n) represent the results of the fields in 3D space. The blue line shows the model predictions, the red line shows the observation data at the current time level, the orange line shows the data used for test data assimilation and the green line shows the results after data assimilation using the above test data. It can be observed from the figures that the orange line is close to the green line, indicating that data assimilation has a positive impact on optimising the model prediction results.



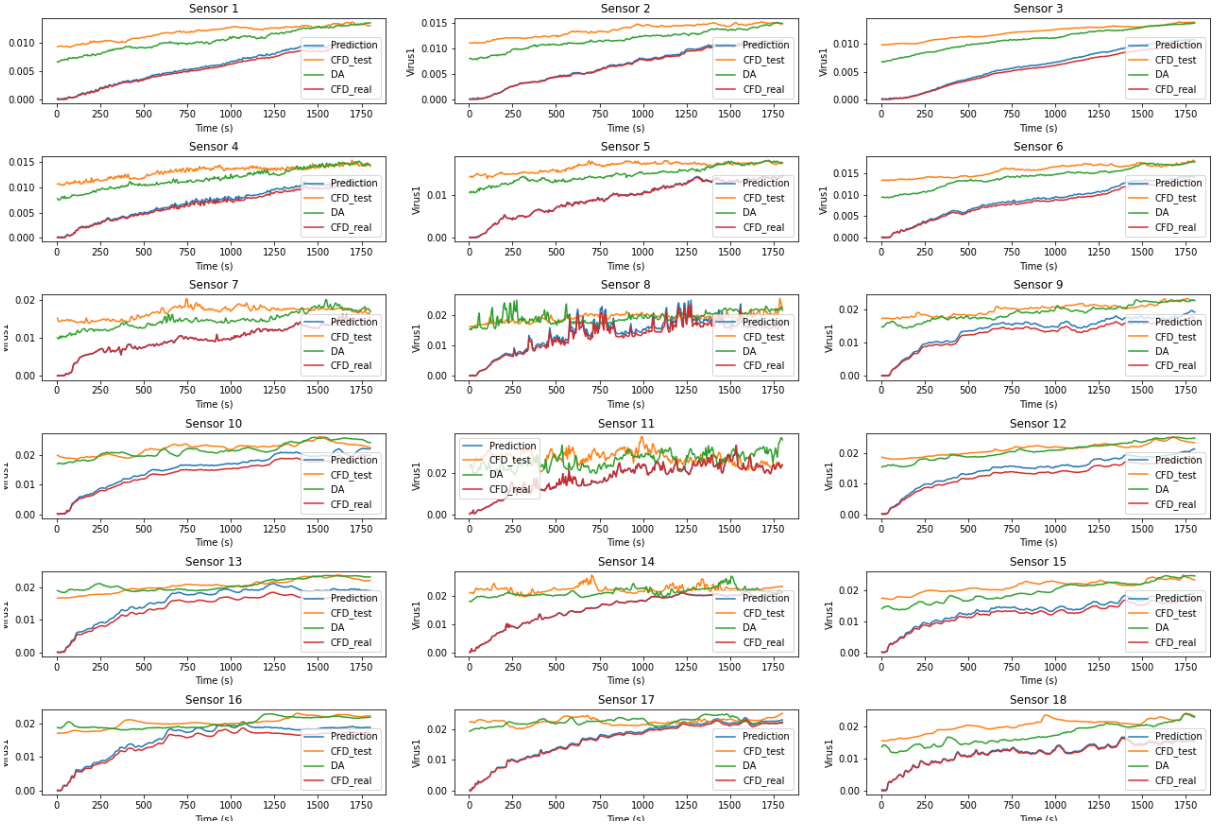
(a) Prediction and DA results of CO_2 concentration level using observation data



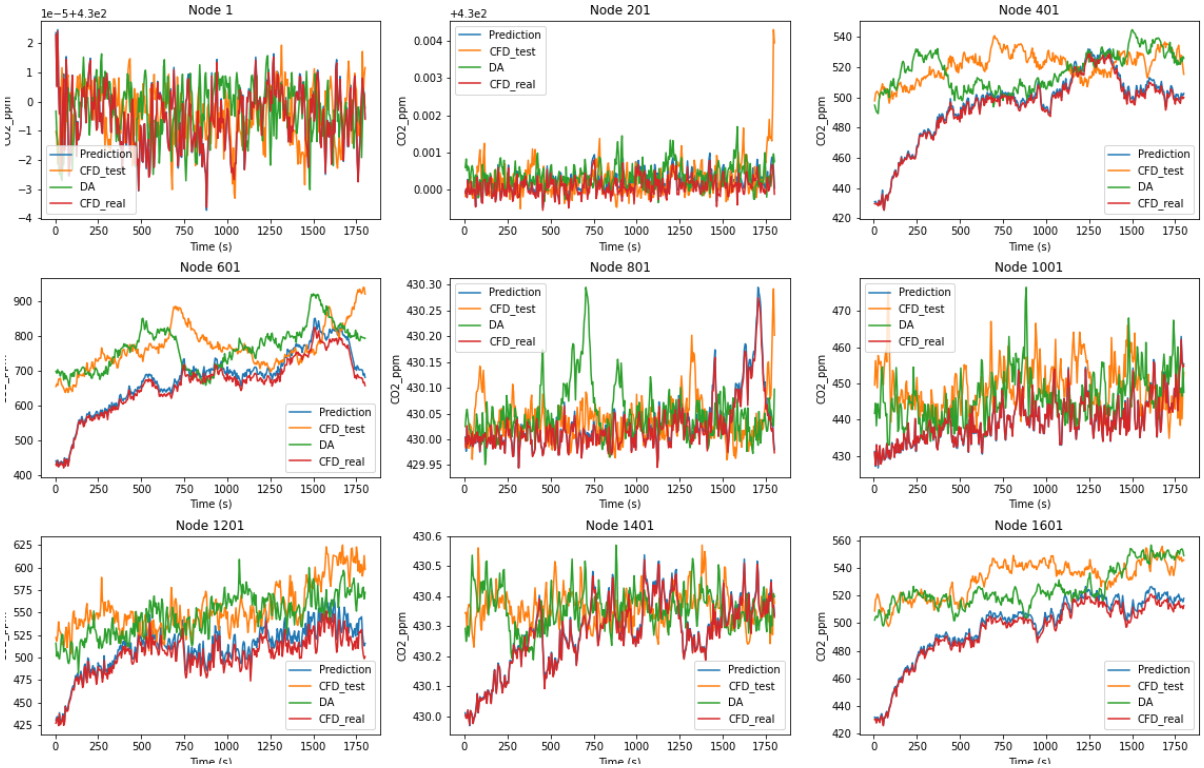
(b) Prediction and DA results of Humidity using observation data



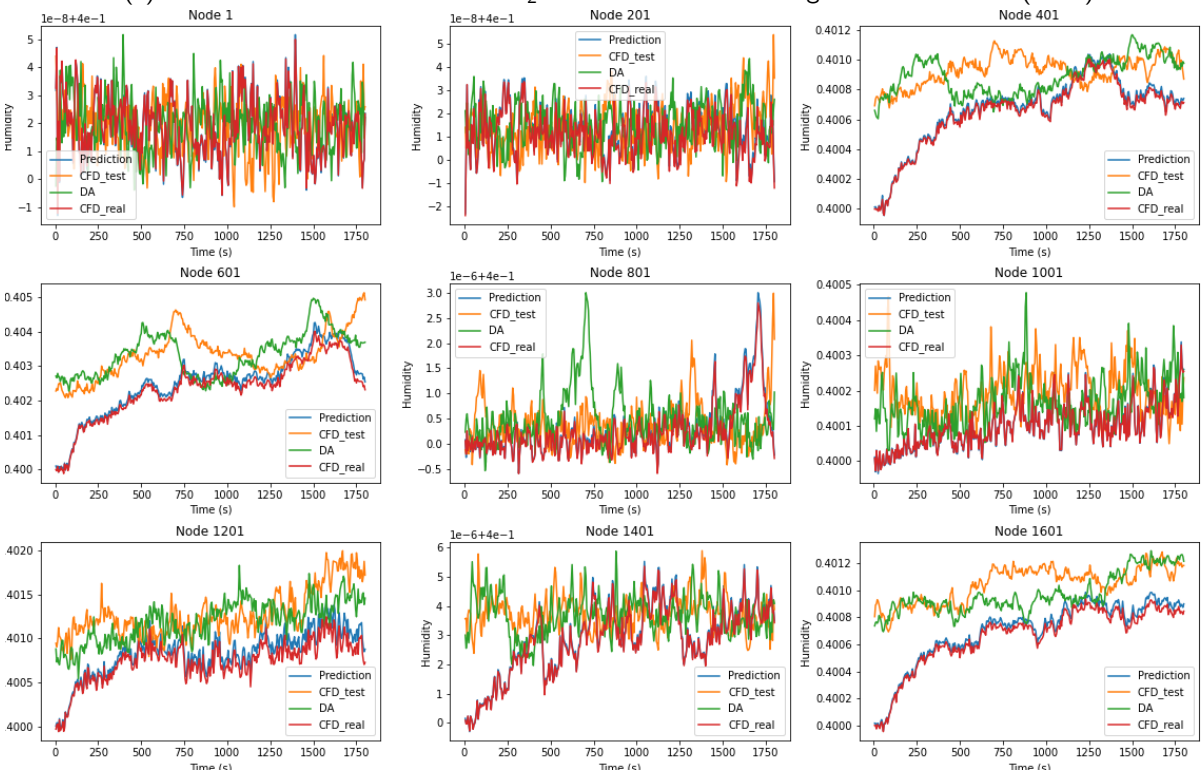
(c) Prediction and DA results of Temperature using observation data



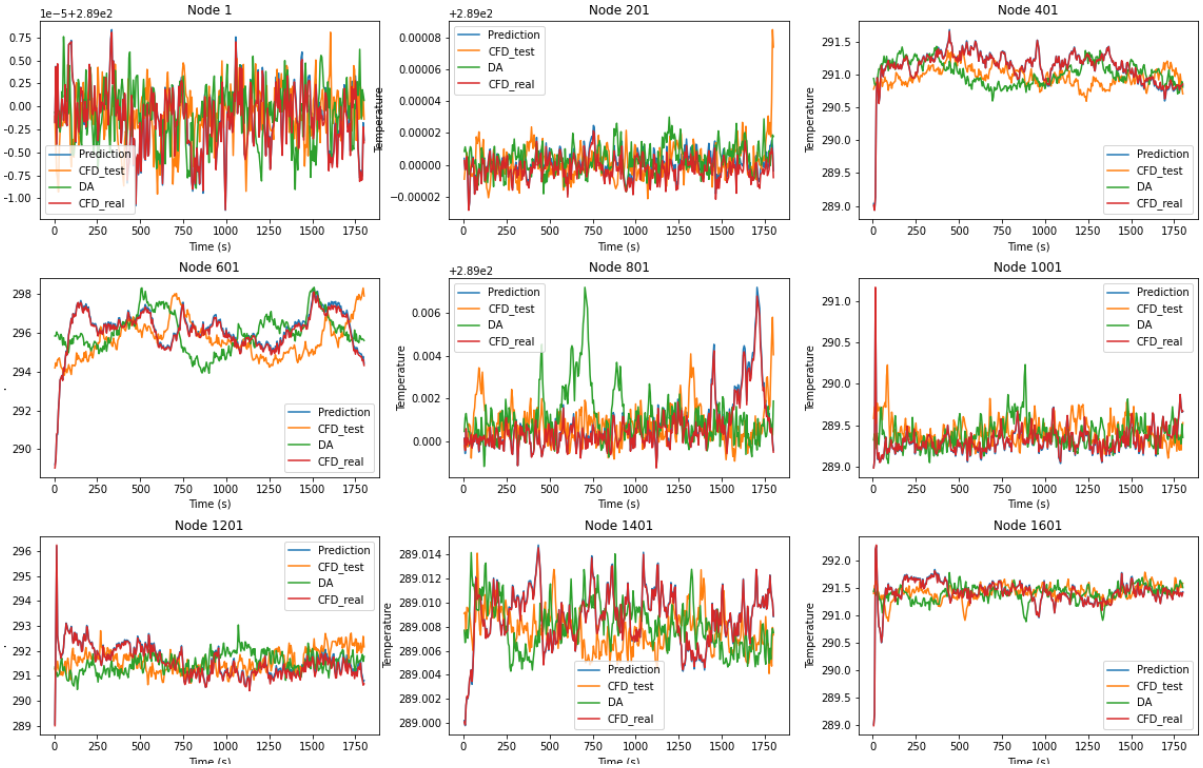
(d) Prediction and DA results of viral load using observation data



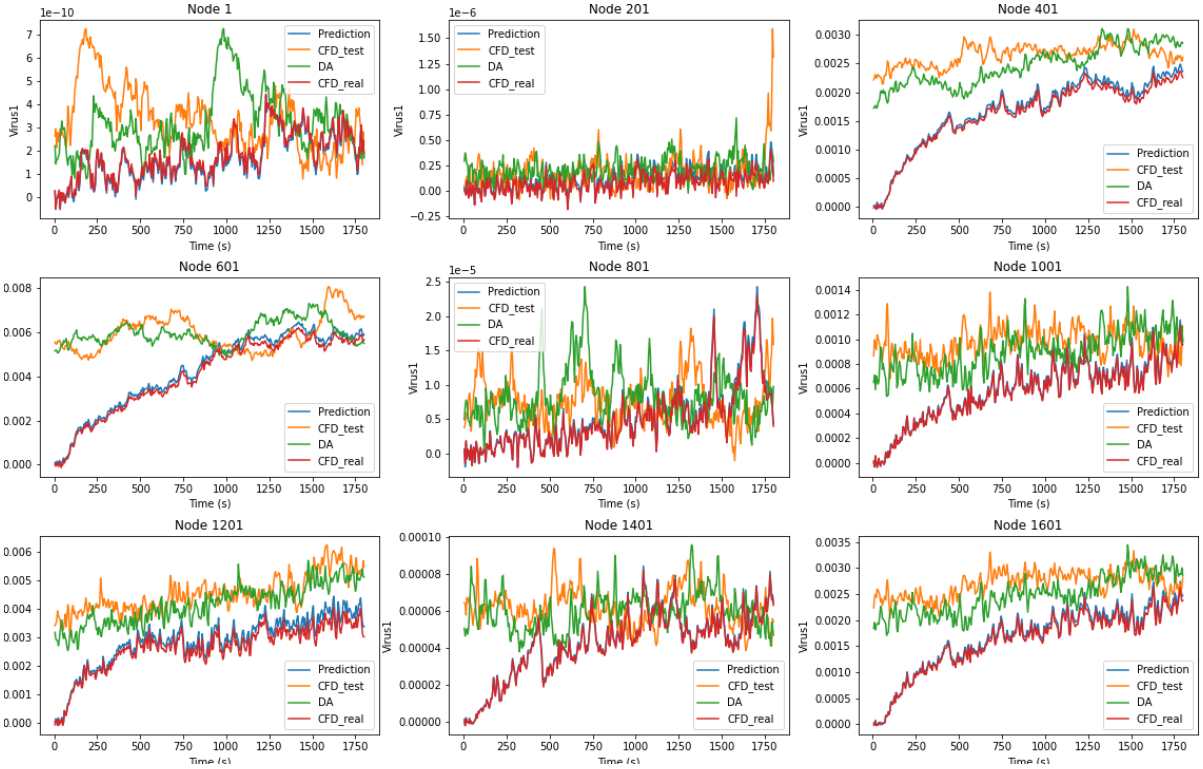
(e) Prediction and DA results of CO_2 concentration level using observation data (node)



(f) Prediction and DA results of Humidity using observation data (node)



(g) Prediction and DA results of Temperature using observation data (node)

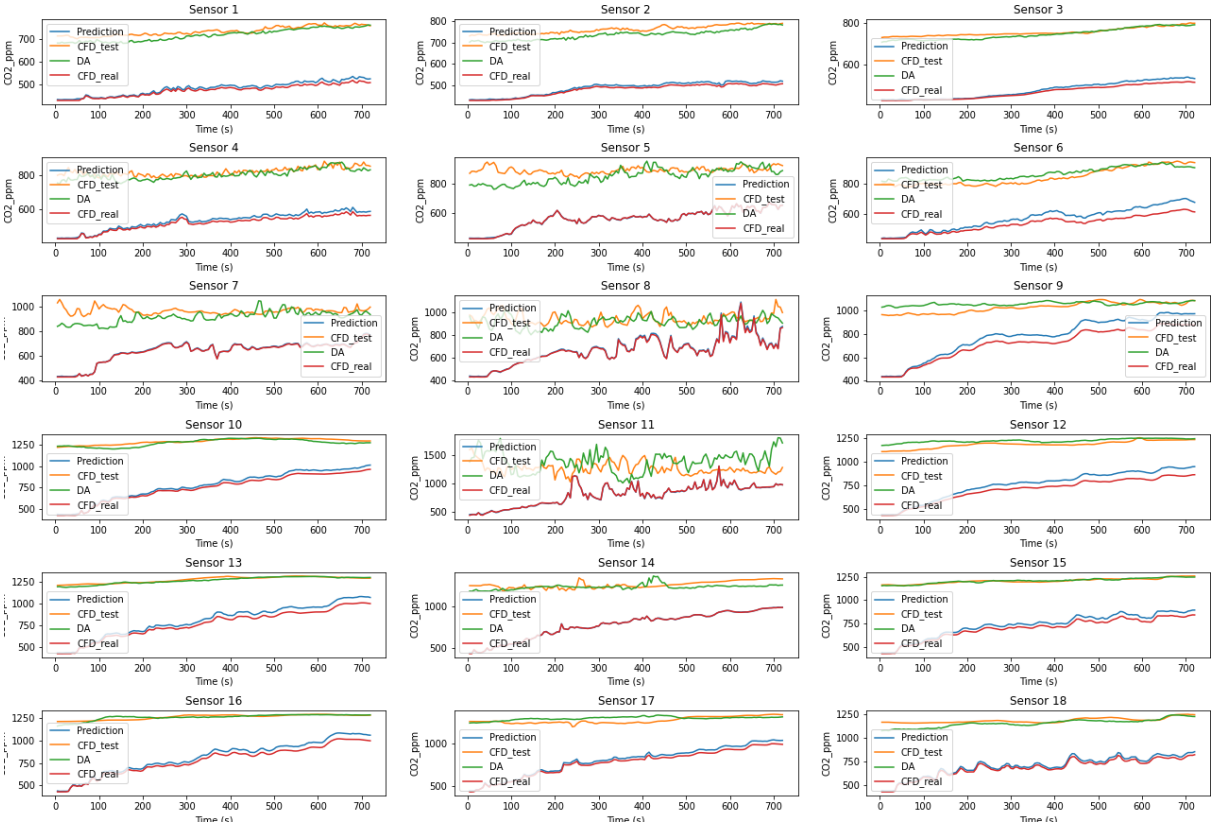


(h) Prediction and DA results of Virus1 using observation data (node)

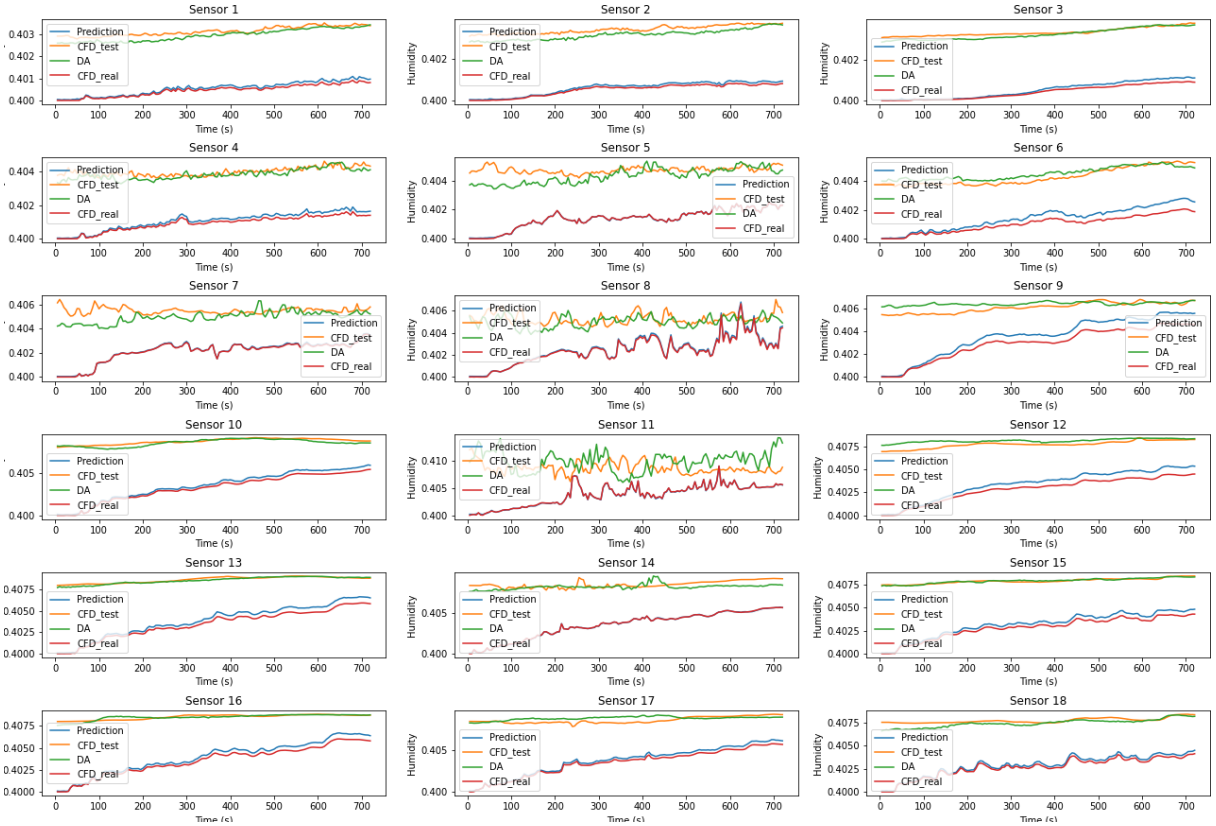
Figure 6: Prediction and DA results. (a) - (d) show the predicted CO_2 concentration levels, humidity, temperature and viral load data and DA results. (e) - (h) represent the results of the above fields in 3D space.

4.3.2 Scenario 2

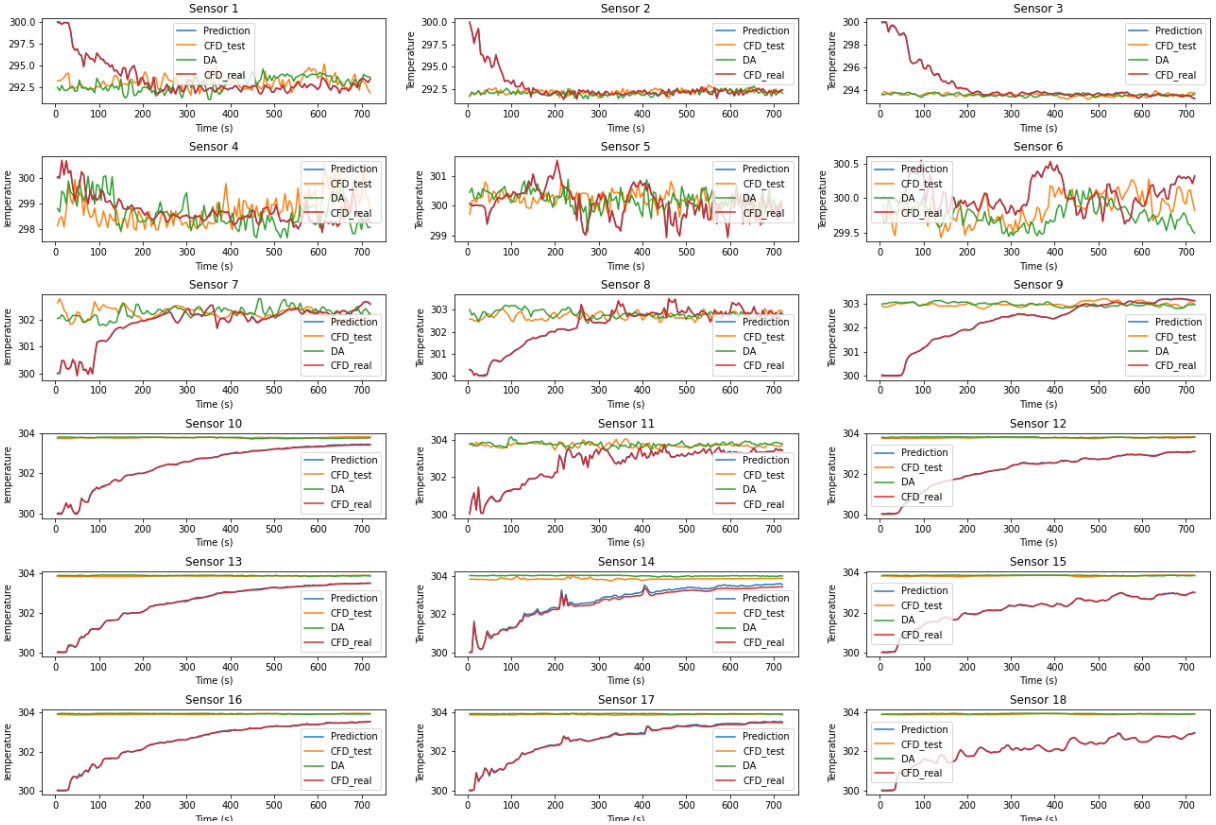
For scenario 2, the data used for data assimilation is the test set part of the data generated by the entire CFD model, i.e. for the model, it is unseen data with time level ≥ 577 (start from 1). This part of the data will be used to perform the same test as in Scenario 1, and the weight of the data assimilation is initialised to 70 and will never change in the iterations. Figure 7 shows the results of this part of the test, with the images in the same order as in scenario 1. It can be observed from the figure that data assimilation using unseen data for this model has a positive optimisation on the prediction results.



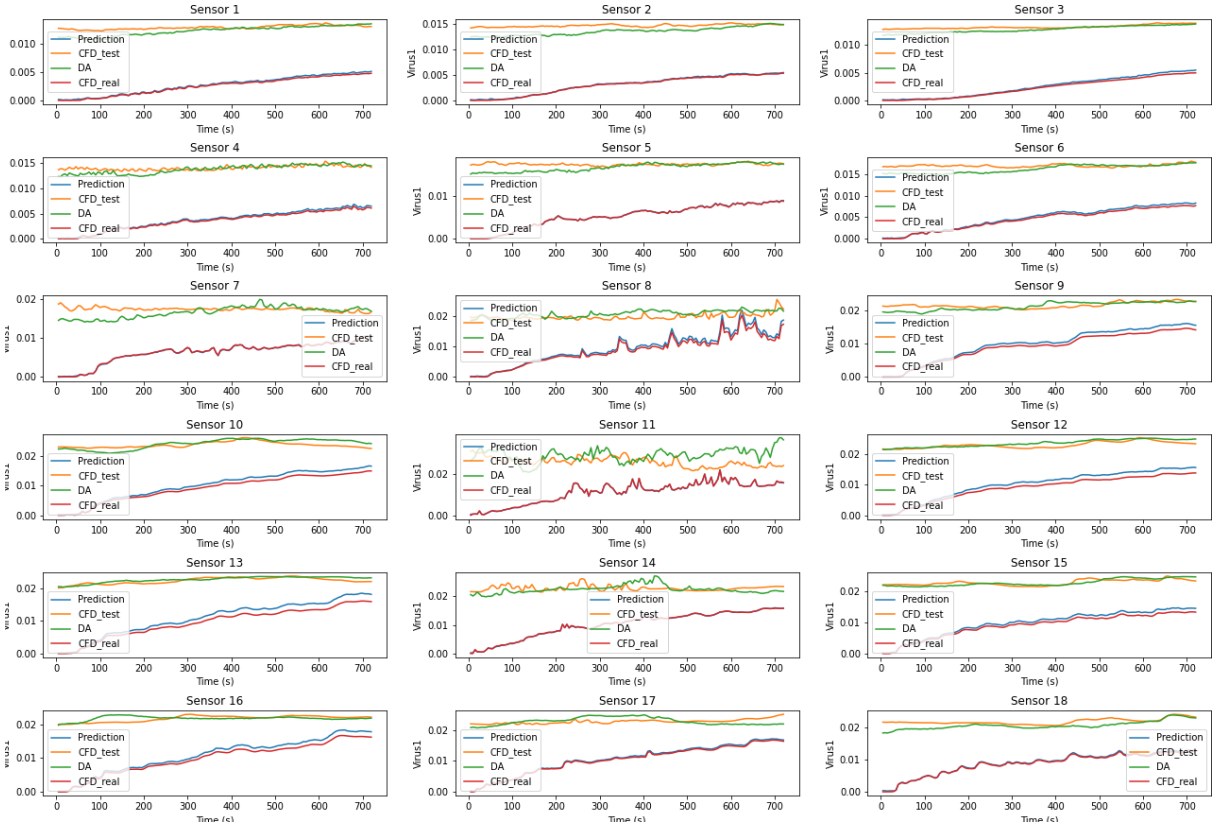
(a) Prediction and DA results of CO_2 concentration level using observation data



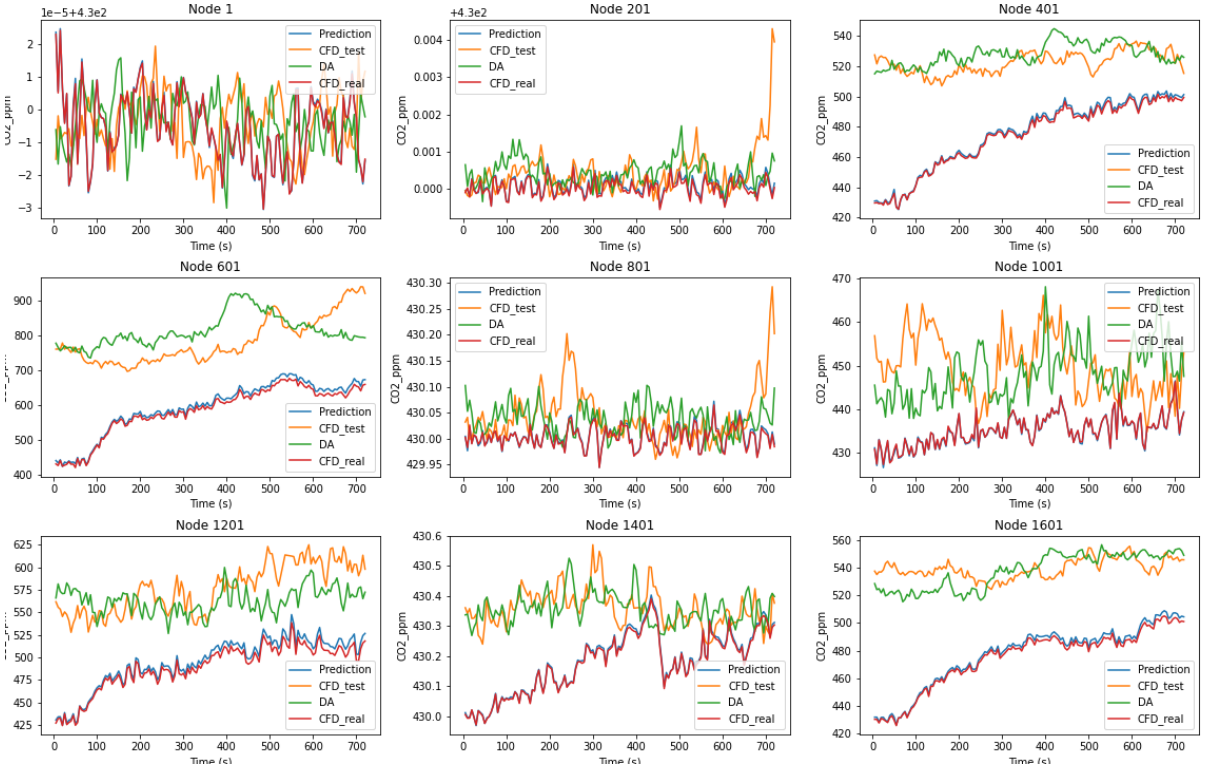
(b) Prediction and DA results of Humidity using observation data



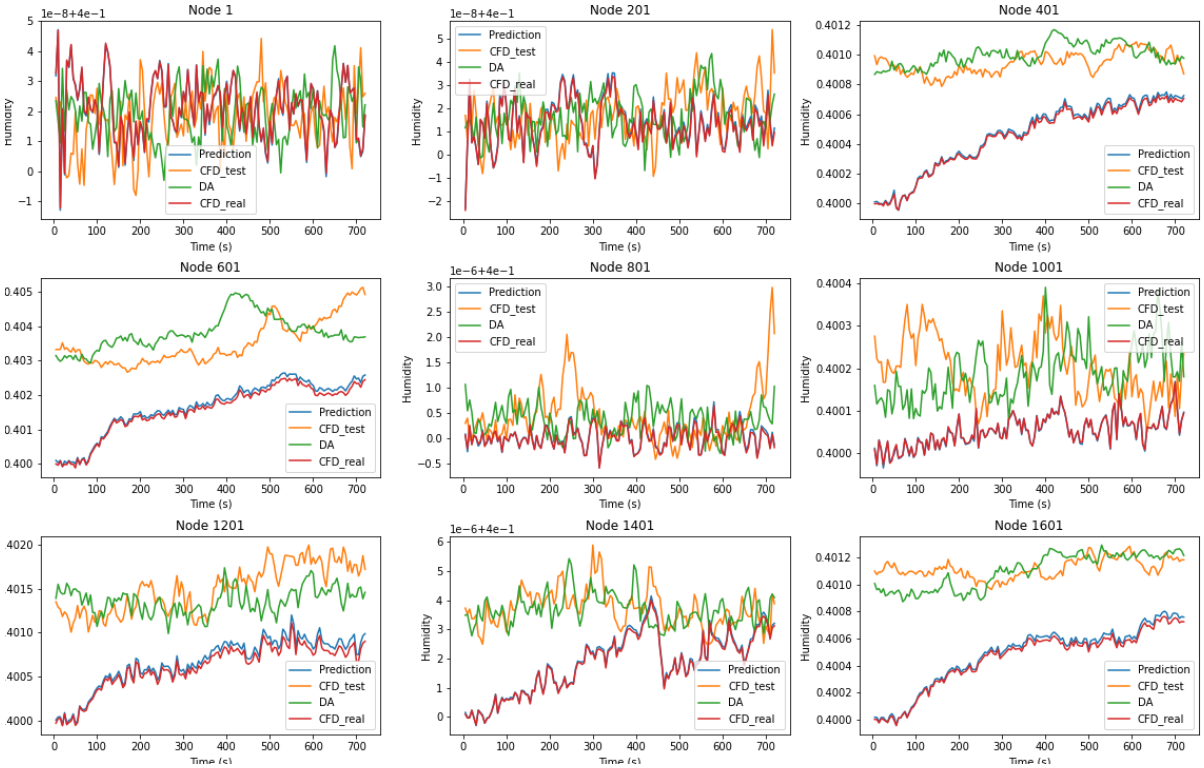
(c) Prediction and DA results of Temperature using observation data



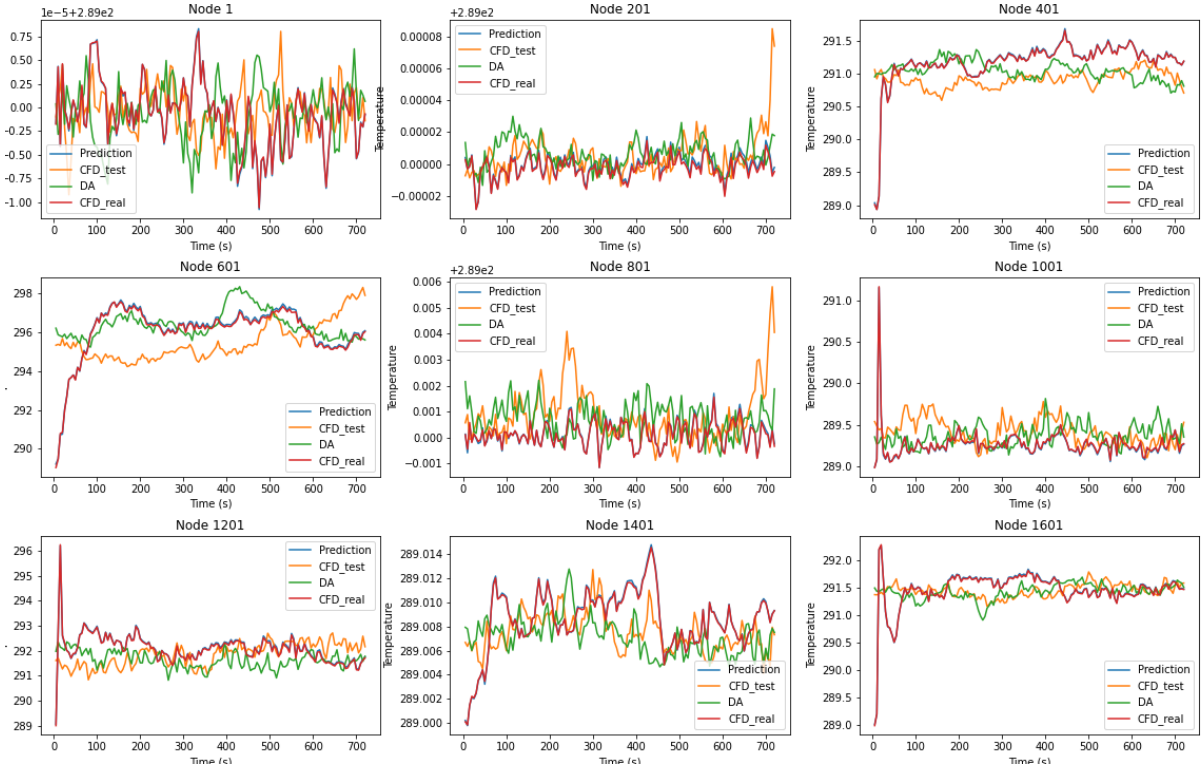
(d) Prediction and DA results of viral load using observation data



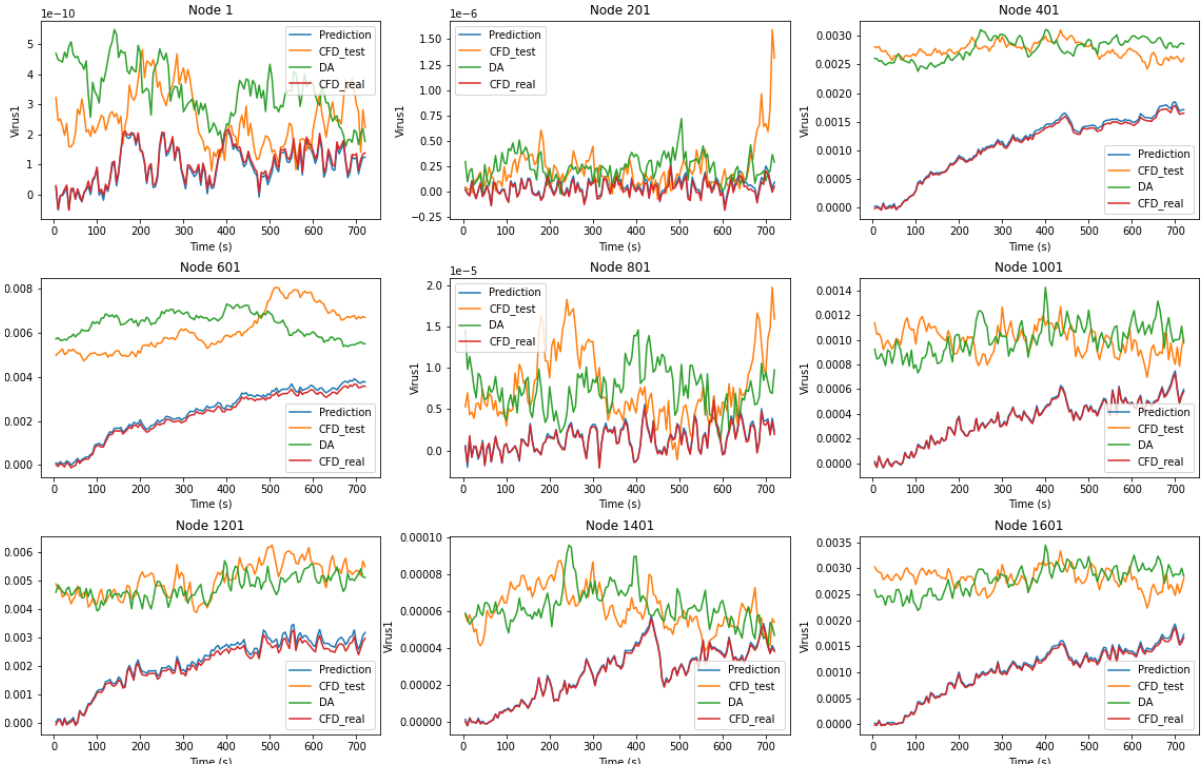
(e) Prediction and DA results of CO_2 concentration level using observation data (node)



(f) Prediction and DA results of Humidity using observation data (node)



(g) Prediction and DA results of Temperature using observation data (node)

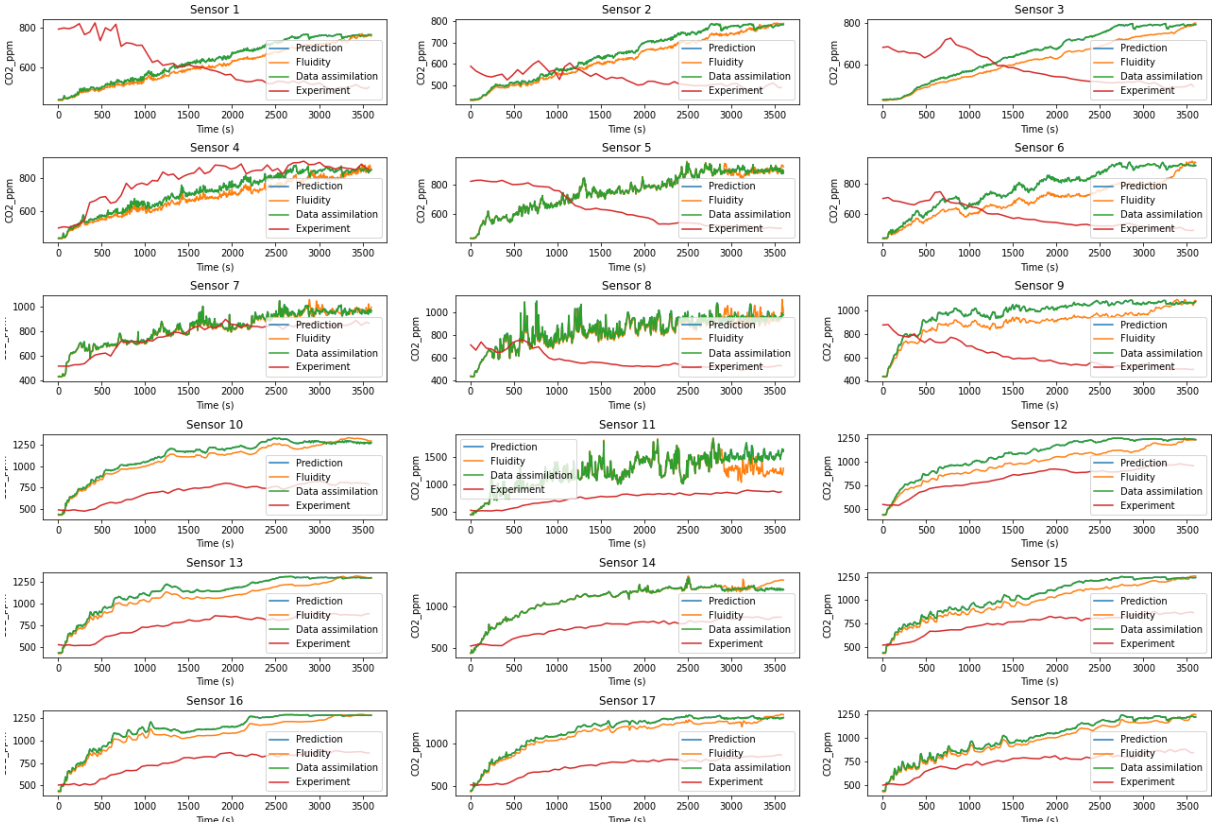


(h) Prediction and DA results of viral load using observation data (node)

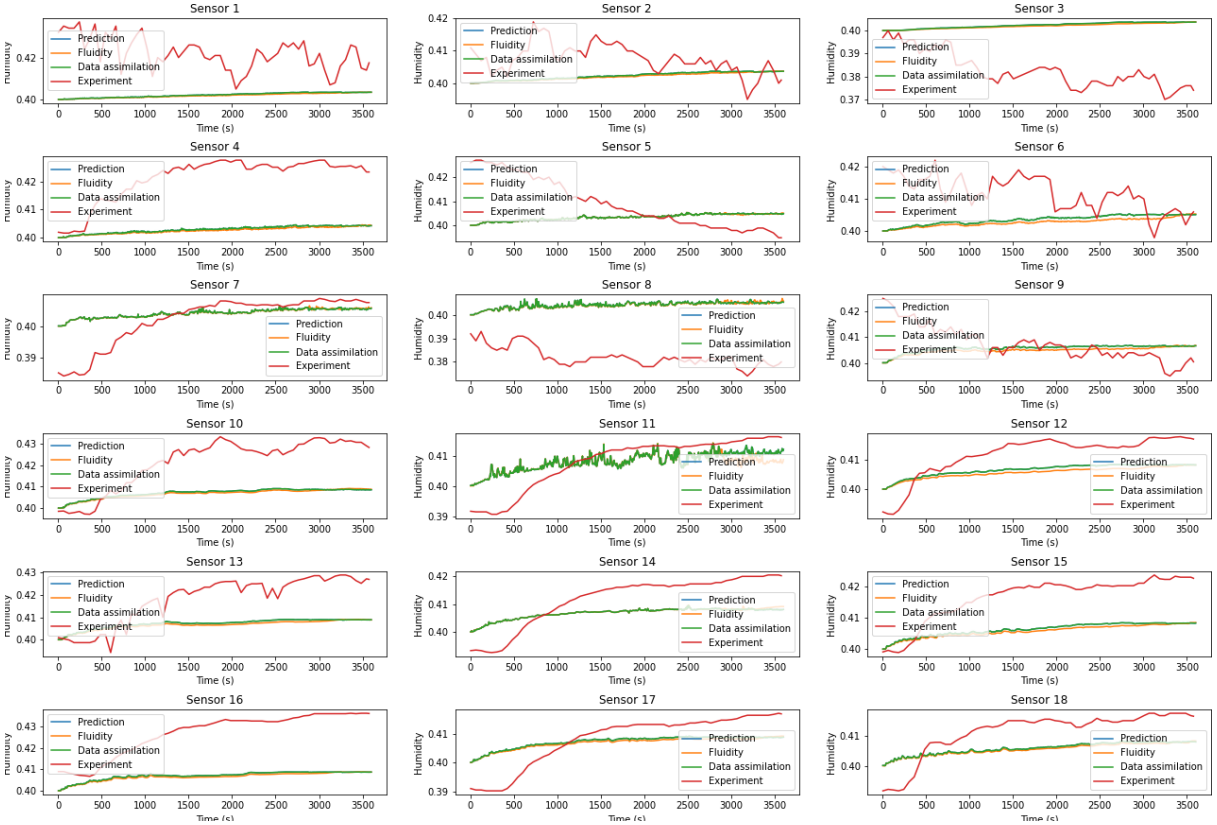
Figure 7: Prediction and DA results. (a) - (d) show the predicted CO_2 concentration levels, humidity, temperature and viral load data and their DA results. (e) - (h) represent the results of the above fields in 3D space.

4.3.3 Scenario 3

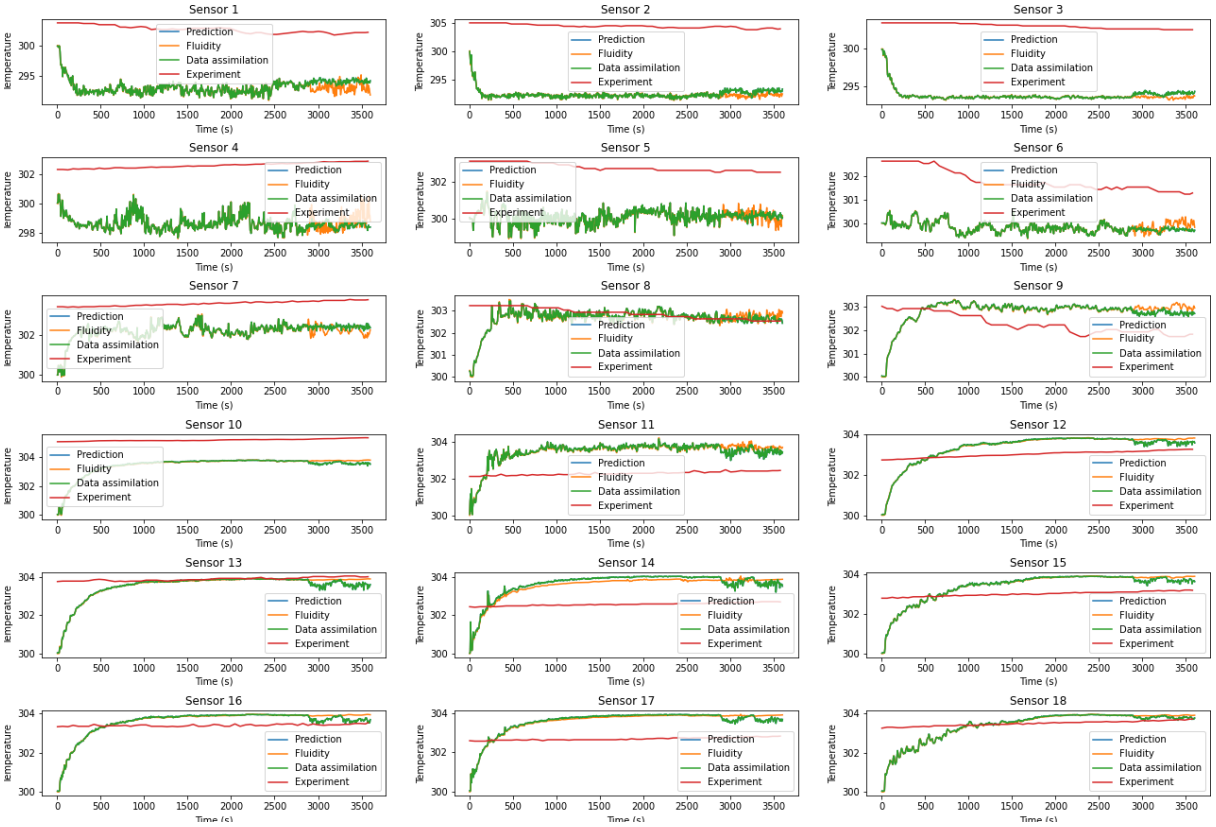
For scenario 3, the data used for data assimilation are the data from the 18 sensors in the classroom. The data here are the CO_2 concentration level, humidity, and temperature fields, selected for the period 1/12/2021, 13:00-14:00, for a total of 120 time levels. This data was obtained by linear interpolation of 60 time levels so that in the results, the experimental data will present in the results of CO_2 concentration levels, humidity and temperature observation data. In this scenario, the DA-Pred-AAE model will predict data for the last 712 time levels, apply data assimilation to the above three fields of the experimental data, and obtain the results with an initial weight of 70 (over relaxation used). Figure 8 illustrates the results of the data assimilation.



(a) Prediction and DA results of CO_2 concentration level using experimental data



(b) Prediction and DA results of Humidity using experimental data



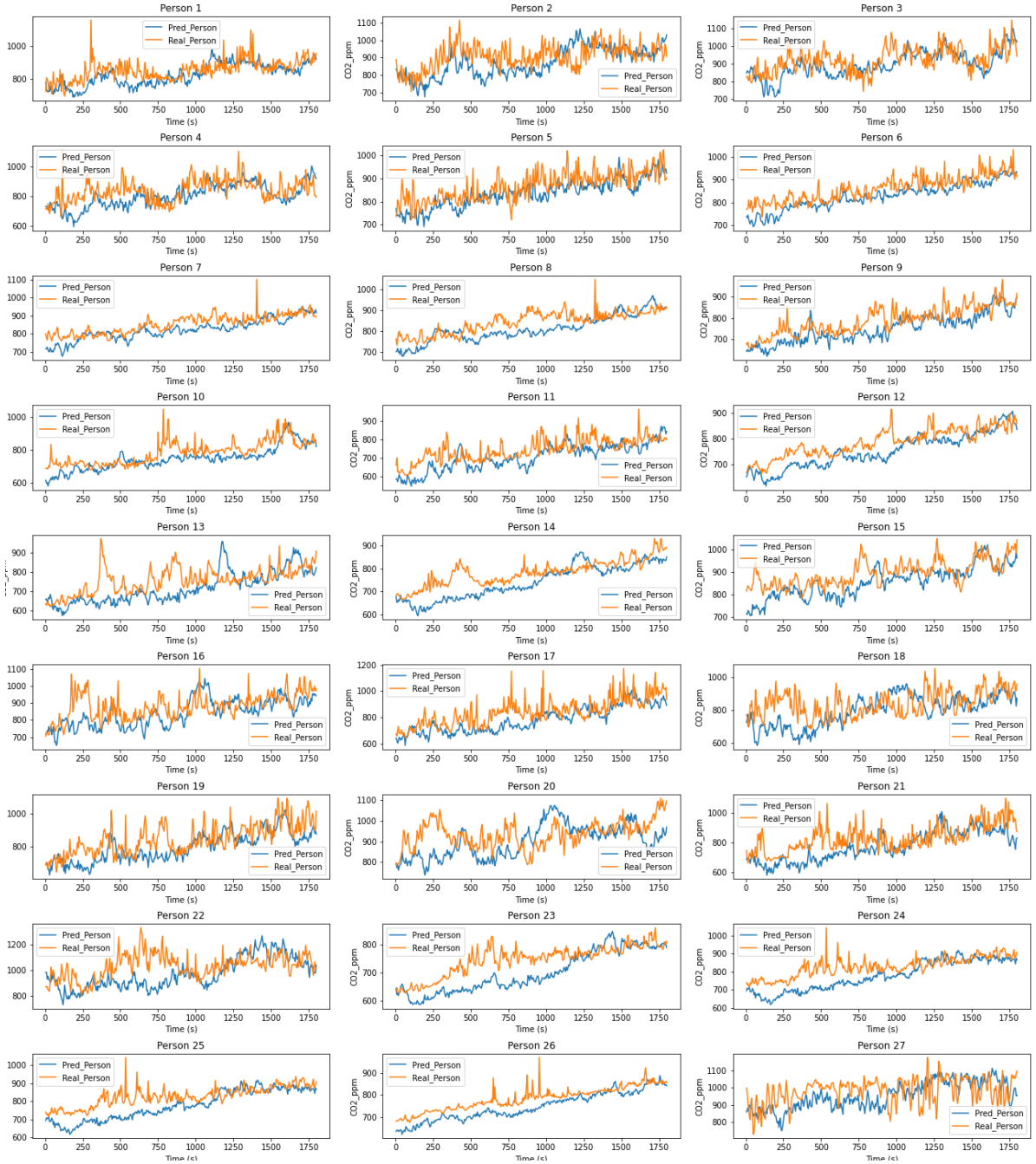
(c) Prediction and DA results of Temperature using experimental data

Figure 8: Prediction and DA results. (a), (b) and (c) shows the results of DA using experimental data.

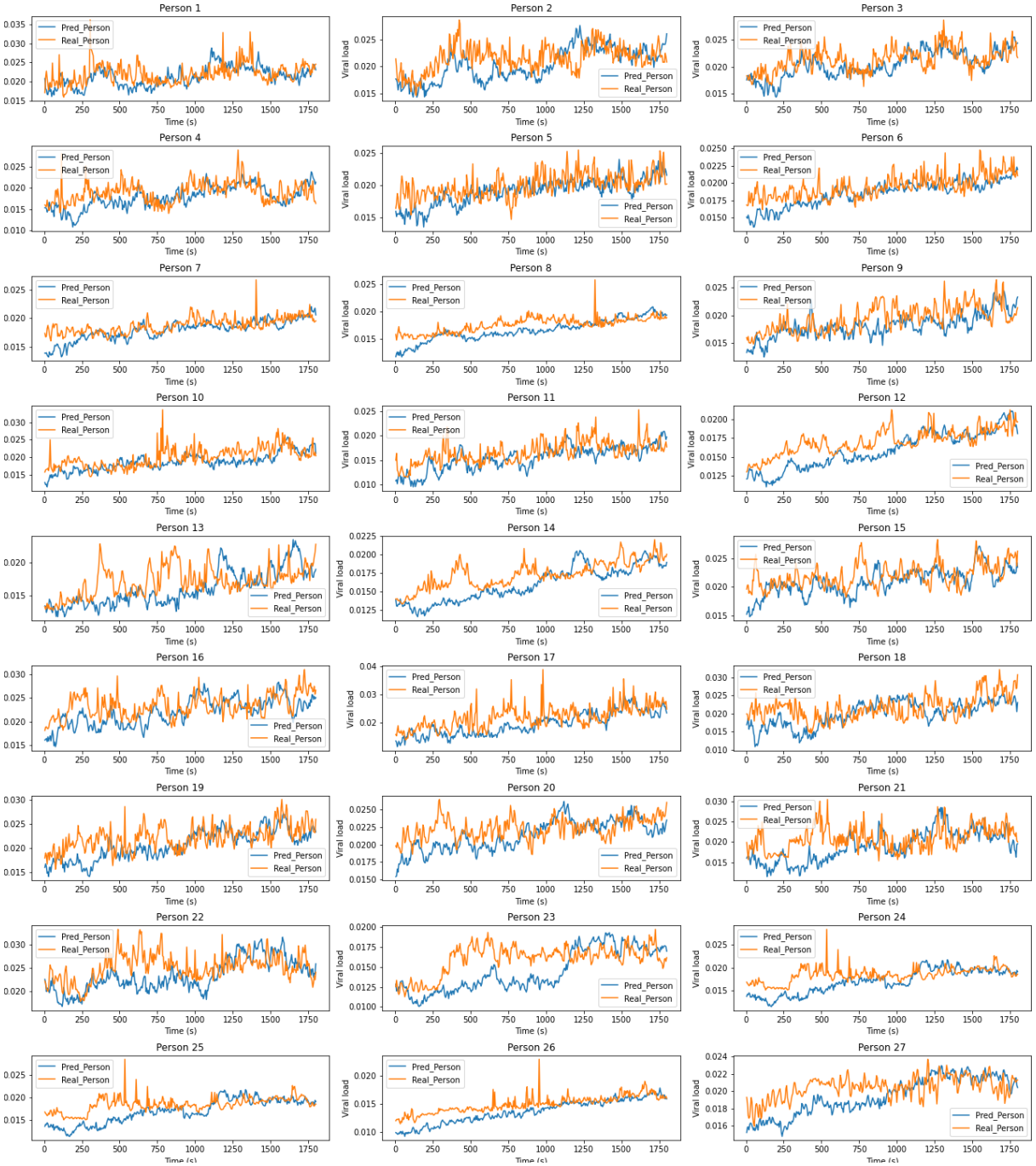
It is known from the above images that the data assimilation results diverge significantly from the experimental data and it is close to the prediction results. This situation may be caused by the insufficient amount of experimental data used for data assimilation. Furthermore, there is no significant improvement in the results after increasing or decreasing the initial weight.

4.3.4 Infection risks after data assimilation

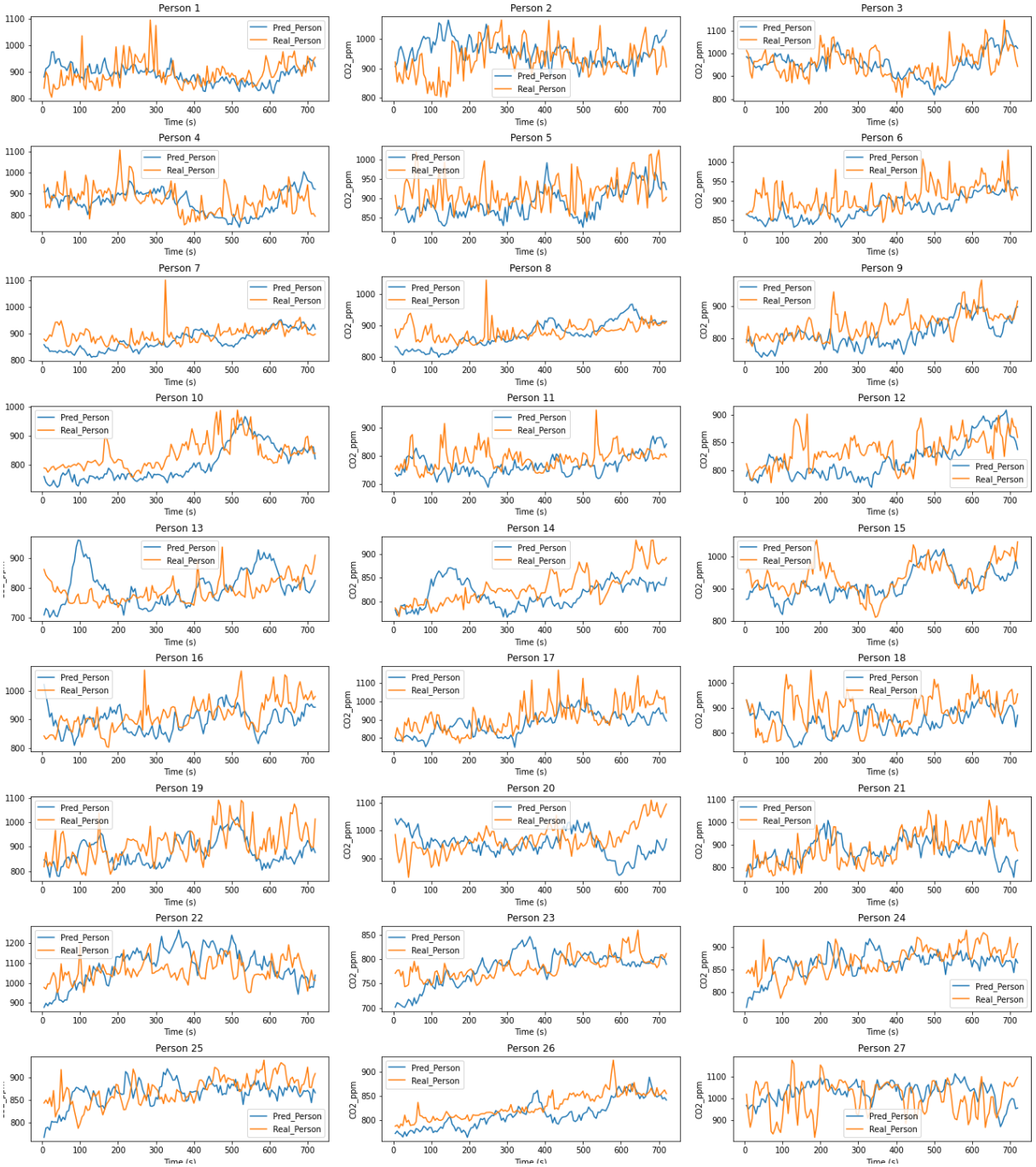
In this section, the predicted values of CO_2 concentration levels and viral load for the 27 individuals in the classroom are optimised by applying data assimilation. Figure 9 shows the output data for CO_2 concentration level and viral load, where (a)-(d) represent the data for case 1 and case 2, respectively. It can be observed from the figure that the results of data assimilation are close to the data generated by the CFD model.



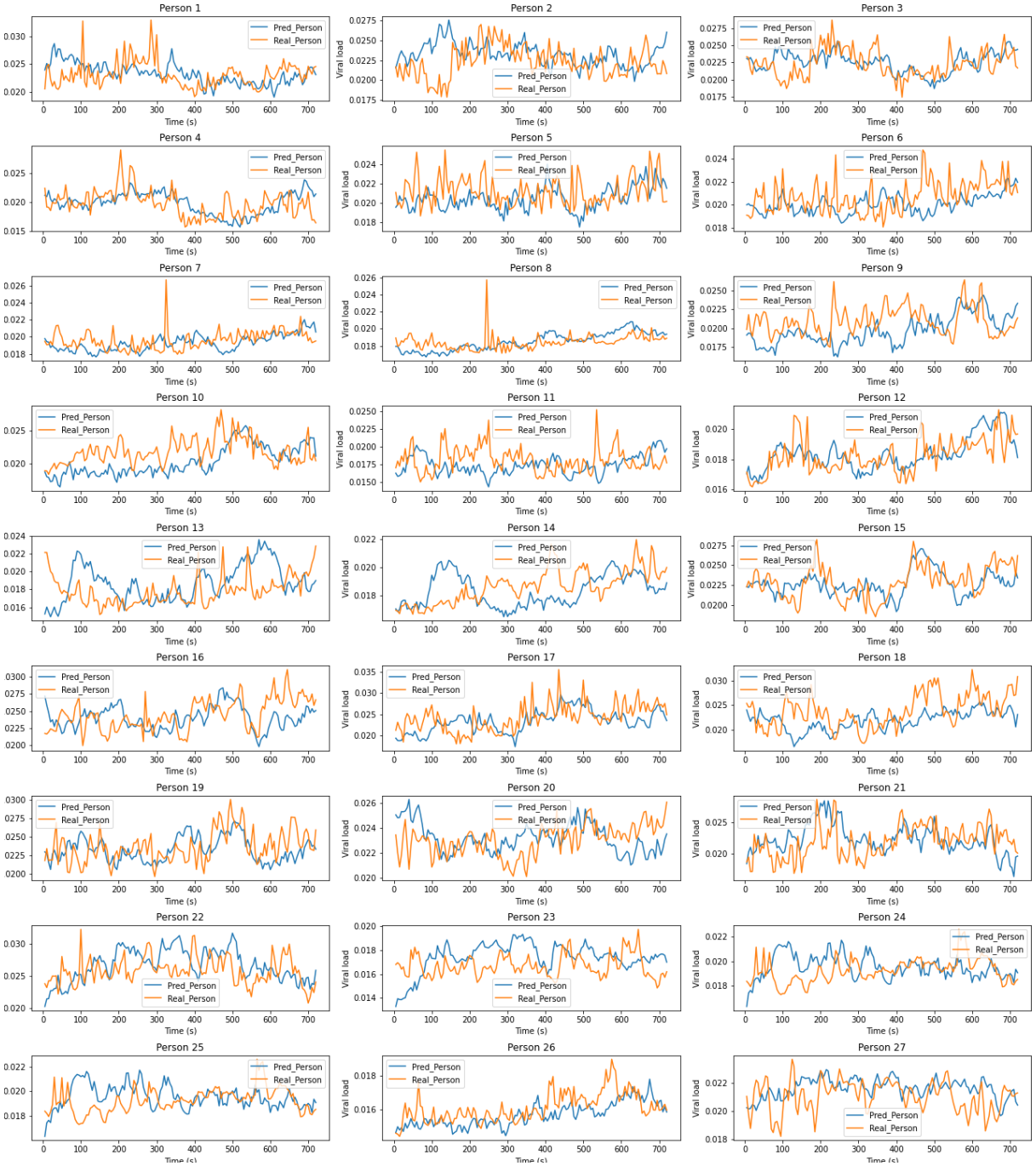
(a) CO_2 concentration level results for 27 people in the classroom (case 1)



(b) Viral load results for 27 people in the classroom (case 1)



(c) CO_2 concentration level results for 27 people in the classroom (case 2)



(d) Viral load results for 27 people in the classroom (case 2)

Figure 9: DA results for 27 people in the classroom. (a) and (b) represent the CO_2 concentration level and viral load data for case 1 and (c) and (d) represent the CO_2 concentration level and viral load data for case 2

5 Discussion and Future work

5.1 Discussion

For the result of the project, the prediction results using Pred-AAE indicate that the model performs well in predicting future data. This is because the predicted data of the model fit well with the output of the CFD model. By using Paraview to visualise the data, the images generated by the predicted values are close to the data generated by the CFD model. For the DA-Pred-AAE model, the data

assimilation results of scenario 1 and scenario 2 show that the application of data assimilation to the model can effectively improve the accuracy of the prediction value. However, in scenario 3, the data assimilation results are significantly different from the experimental data, which may cause by the insufficient data volume. The amount of experimental data used for data assimilation is one-sixth of the number of prediction steps. The amount of data used for data assimilation was substantially less than the prediction steps compared to scenarios 1 and 2. This shows that the limitation of forward and backward march is that the algorithm requires a sufficient number of data. The project can predict the risk of infection in the classroom. In Section 4.3.4, by entering the coordinates of the 27 individuals in the classroom, the CO_2 concentration level and the viral load data at these locations are extracted. By observing the results in figure 9, the model can obtain more accurate results through data assimilation and successfully quantify the infection risk of 27 individuals in the classroom.

In this project, there are two most difficult parts. The first part is to implement data assimilation. Due to the uncertainty of the data assimilation data and the model's prediction performance, the generated data assimilation results may deviate from the observation data. The second part is to build an AAE model with a reasonable architecture and the selection of hyperparameters. Unfortunately, due to the short time period of the project and the limitation of the Google Colab platform on the daily GPU usage, only a small amount of grid search can be applied, so it is difficult to choose the best hyperparameters.

The advantage of this solution is that data assimilation is successfully applied to optimise prediction results, and better results are obtained. Furthermore, the model can predict the CO_2 concentration level and viral load at the position of 27 individuals to predict the risk of COVID-19. However, the limitation of this project is that when applying data assimilation to experimental data with insufficient data, the generated results deviate from the experimental data.

5.2 Future work

The following work is a worthwhile part of future exploration.

1. The data's trend can be considered when generating an initial guess to improve the prediction results.
2. Perform more grid searches to optimise the model.
3. Improve the performance of data assimilation in case of insufficient observation data.
4. Perform dimensionality reduction using SFC-CAE to improve the performance of data dimensionality reduction.
5. Use linear interpolation to get enough experimental data for scenario 3.

6 Conclusion

In this project, the Pred-AAE and DA-Pred-AAE models are implemented, and predictions can be made based on time series. DA-Pred-AAE is based on Pred-AAE with the addition of observation data for data assimilation to modify the model's prediction. Scenarios 1-3 verify the usefulness and weakness of data assimilation by modifying the data with observation data. Furthermore, the above scenarios validate that the model can optimise the prediction results through data assimilation when the observation data are sufficient. Therefore, it is feasible to use a machine learning model as an alternative to a CFD model to reduce computational costs and improve the efficiency of the generated data.

Due to the close relationship between CO_2 and COVID-19 transmissibility[28] and the viral load field provided in the dataset of this project, the model can predict the CO_2 concentration level and viral

load in the room. Furthermore, the coordinates of the human in the room are provided, and the results generated by data assimilation are also close to those generated by the CFD model. Therefore, the model can also predict the CO_2 concentration level and viral load of the human position to determine the risk of infection in the room. In summary, Pred-AAE and DA-Pred-AAE models with data assimilation can be used to predict better future data based on time series and this project fills a gap in research on apply data assimilation for AAE models.

References

[1] Coronavirus disease (COVID-19): How is it transmitted? https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-how-is-it-transmitted. Accessed: 2022-06-21.

[2] James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717(8), 2005.

[3] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. Applications of Generative Adversarial Networks (GANs): An Updated Review. *Archives of Computational Methods in Engineering*, (1), 2019.

[4] Paul Barrett, John Hunter, J Todd Miller, J-C Hsu, and Perry Greenfield. matplotlib–A Portable Python Plotting Package. In *Astronomical data analysis software and systems XIV*, volume 347, page 91, 2005.

[5] Björn Birnir. The build-up of aerosols carrying the SARS-CoV-2 Coronavirus, in poorly ventilated, confined spaces. *medRxiv*, 2020.

[6] Ekaba Bisong. Introduction to scikit-learn. In *Building machine learning and deep learning models on Google cloud platform*, pages 215–229. Springer, 2019.

[7] Jie Chen, Guo-Qiang Zeng, Wuneng Zhou, Wei Du, and Kang-Di Lu. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. *Energy Conversion and Management*, 165:681–695, 2018.

[8] M. Chiah and A. Zhong. Trading from home: The impact of COVID-19 on trading volume around the world. *Finance Research Letters*, 37, 2020.

[9] Marco Fahl and Ekkehard W. Sachs. Reduced Order Modelling Approaches to PDE-Constrained Optimization Based on Proper Orthogonal Decomposition. In Lorenz T. Biegler, Matthias Heinkenschloss, Omar Ghattas, and Bart van Bloemen Waanders, editors, *Large-Scale PDE-Constrained Optimization*, pages 268–280, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[10] Angham G Hadi, Mohammed Kadhom, Nany Hairunisa, Emad Yousif, and Salam A Mohammed. A review on COVID-19: origin, spread, symptoms, treatment, and prevention. *Biointerface Research in Applied Chemistry*, 10(6):7234–7242, 2020.

[11] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[12] D. S. Hui, E. I. Azhar, T. A. Madani, F. Ntoumi, R. Kock, O. Dar, G. Ippolito, T. D. Mchugh, C Zamab, and C Cdab. The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — The latest 2019 novel coronavirus outbreak in Wuhan, China - ScienceDirect. *International Journal of Infectious Diseases*, 91:264–266, 2020.

[13] Oliver Kramer. *Scikit-Learn*, pages 45–53. Springer International Publishing, Cham, 2016.

[14] Yann Lecun. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. Universite P. et M. Curie (Paris 6), June 1987.

[15] Seunghee Lee, Seohui Park, Myong-In Lee, Ganghan Kim, Jungho Im, and Chang-Keun Song. Air Quality Forecasts Improved by Combining Data Assimilation and Machine Learning With Satellite AOD. *Geophysical Research Letters*, 49(1):e2021GL096066, 2022. e2021GL096066 2021GL096066.

[16] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[17] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial Autoencoders, 2015.

[18] G. J. Milne and S. Xie. The Effectiveness of Social Distancing in Mitigating COVID-19 Spread: a modelling analysis. *Cold Spring Harbor Laboratory Press*, 2020.

[19] Jojo Moolayil. *An Introduction to Deep Learning and Keras*, pages 1–16. Apress, Berkeley, CA, 2019.

[20] Praneeth Netrapalli. An Introduction to PCA. 2015.

[21] Zhihong Pang, Pingfan Hu, Xing Lu, Qingsheng Wang, and Zheng O'Neill. A Smart CO2-Based Ventilation Control Framework to Minimize the Infection Risk of COVID-19 In Public Buildings 2. In *Building Simulation 2021 Conference*, 2021.

[22] Santanu Pattanayak. *Introduction to Deep-Learning Concepts and TensorFlow*, pages 89–152. Apress, Berkeley, CA, 2017.

[23] Shanbi Peng, Qikun Chen, and Enbin Liu. The role of computational fluid dynamics tools on investigation of pathogen transmission: Prevention and control. *Science of The Total Environment*, 746:142090, 2020.

[24] Khalid M. Saqr. Amicus Plato, sed magis amica veritas: There is a reproducibility crisis in COVID-19 Computational Fluid Dynamics studies, 2021.

[25] William J Schroeder, Lisa Sobierajski Avila, and William Hoffman. Visualizing with VTK: a tutorial. *IEEE Computer graphics and applications*, 20(5):20–27, 2000.

[26] Vinicius L. S. Silva, Claire E. Heaney, Yaqi Li, and Christopher C. Pain. Data Assimilation Predictive GAN (DA-PredGAN): applied to determine the spread of COVID-19, 2021.

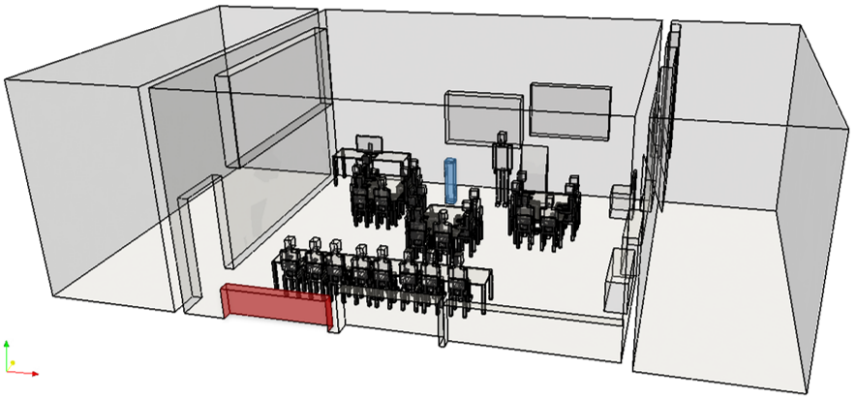
[27] Twan Van Hooff and Bert Blocken. Full-scale measurements of indoor environmental conditions and natural ventilation in a large semi-enclosed stadium: possibilities and limitations for CFD validation. *Journal of Wind Engineering and Industrial Aerodynamics*, 104:330–341, 2012.

[28] Wei Zhang, Xiang Li, and Xu Li. Deep learning-based prognostic approach for lithium-ion batteries with adaptive time-series prediction and on-line validation. *Measurement*, 164:108052, 2020.

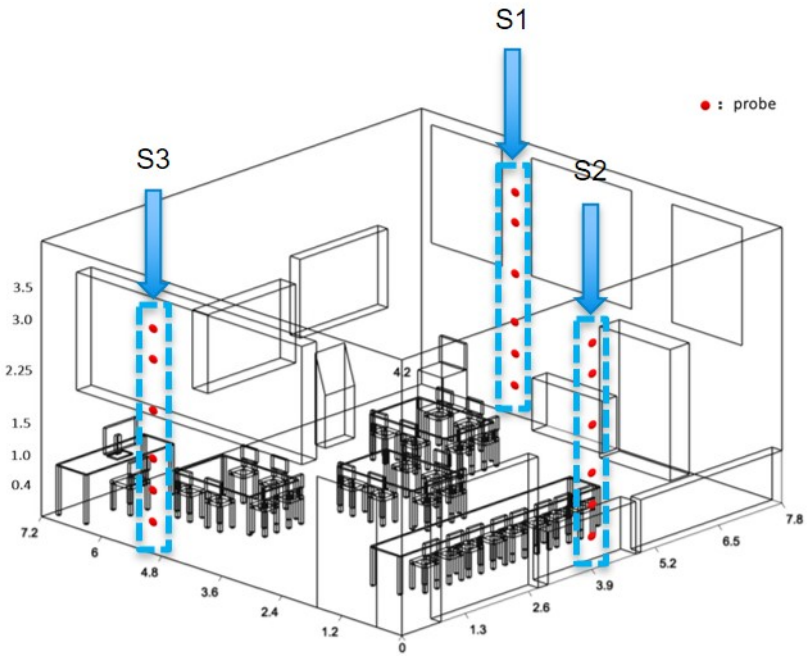
[29] Zhu Zhongming, Lu Linong, Yao Xiaona, Zhang Wangqiang, Liu Wei, et al. CO2 monitoring recommended to manage COVID-19 spread in schools and offices. 2021.

Appendices

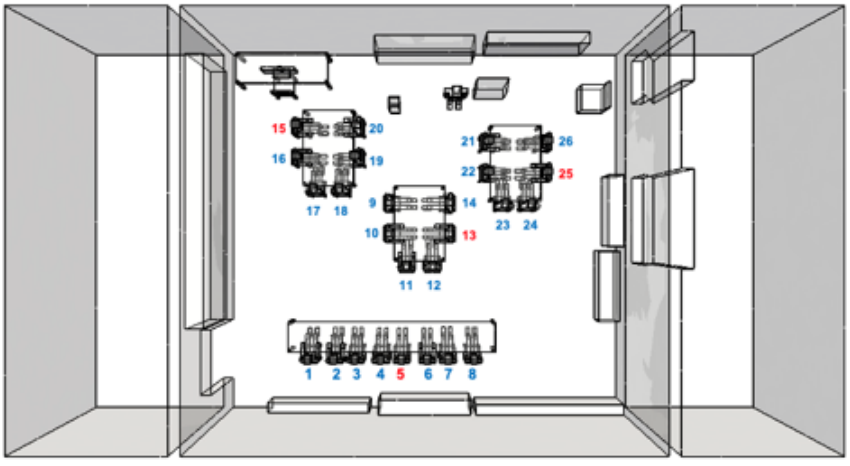
Part 1: Details about the dataset



(a) Structure of the classroom



(b) Layout of the sensors and classroom where data was collected



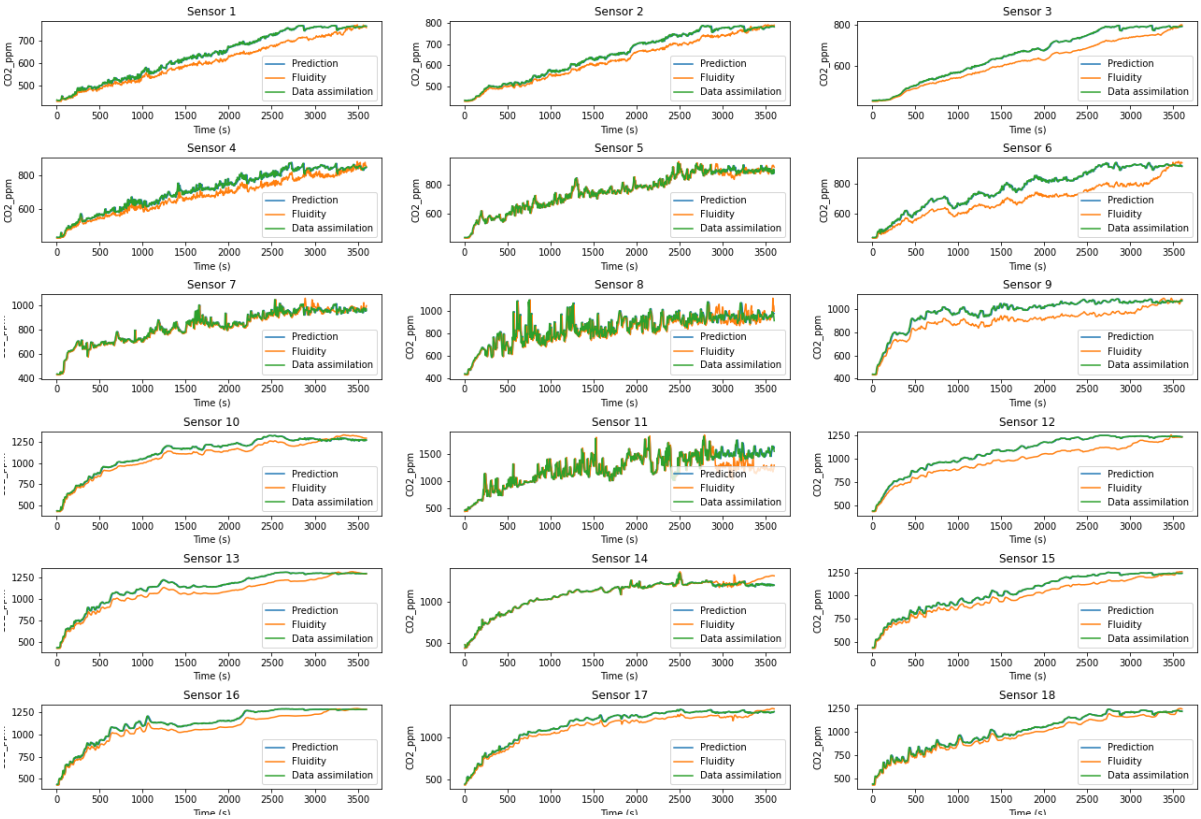
(c) The location of people in the classroom and their numbers

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
X	4.5	5.0	5.4	5.8	6.3	6.8	7.2	7.7	6.3	6.3	6.5	7.0	6.8	6.8	4.5	4.5	4.6	5.0
Y	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	4.0	3.5	3.0	3.0	3.5	4.0	5.6	5.1	4.7	4.7
ID	19	20	21	22	23	24	25	26	27				1-26		27			
X	5.0	5.0	8.2	8.2	8.4	8.8	8.8	8.8	7.2		Z		1.0		1.5			
Y	5.0	5.5	5.4	4.8	4.4	4.4	4.8	5.3	5.6									

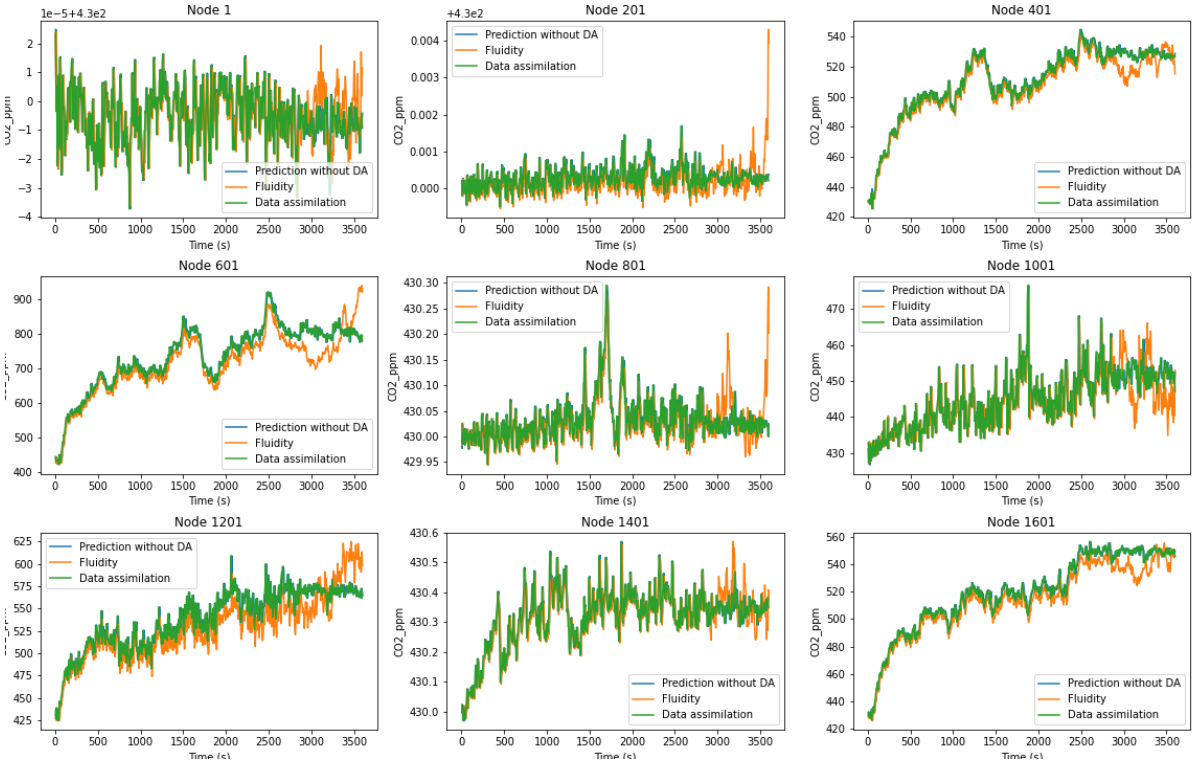
(d) Coordinates of 27 individuals in the classroom space

Figure 10: Layout of the sensors, individuals and classroom.

Part 2: Results of the basic data assimilation



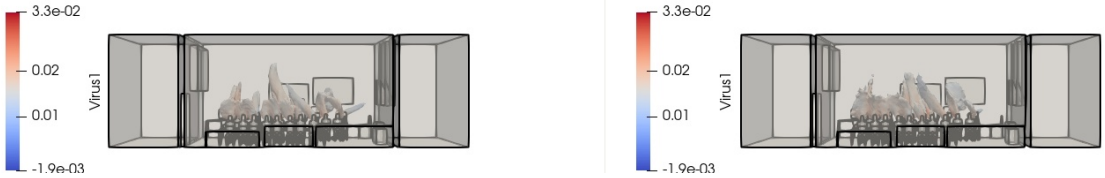
(a) DA results for CO_2 concentration level (sensor).



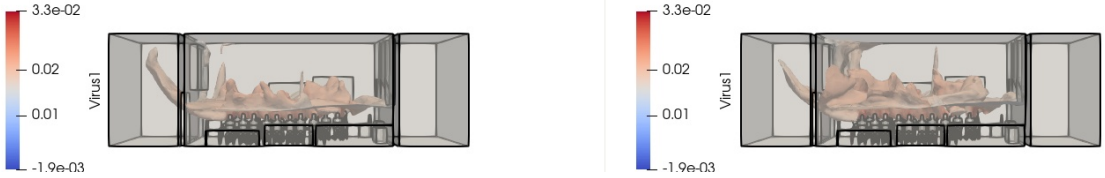
(b) DA results for CO_2 concentration level (node).

Figure 11: Results for Basic DA (CO2). (a) is the result for sensor and (b) is the result for node

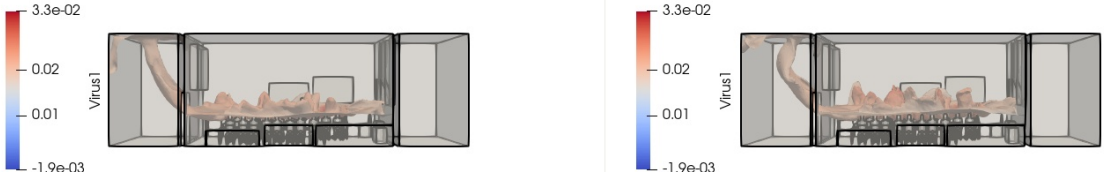
Part 3: Visualisation results of the basic data assimilation



(a) Visualisation (Basic DA results) of CO_2 concentration level at time = 215s



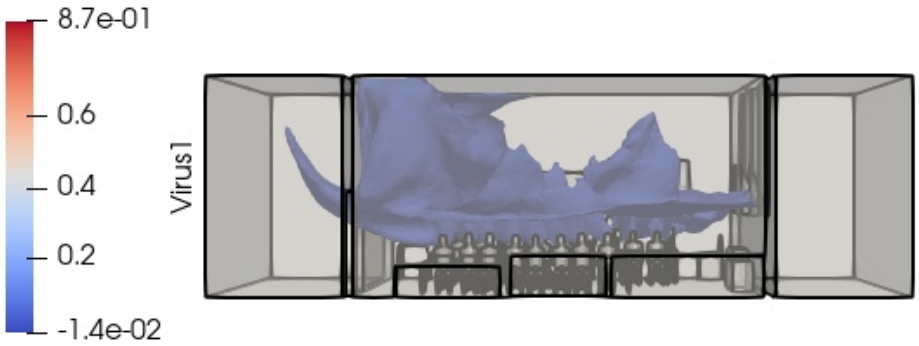
(b) Visualisation (Basic DA results) of CO_2 concentration level at time = 1625s



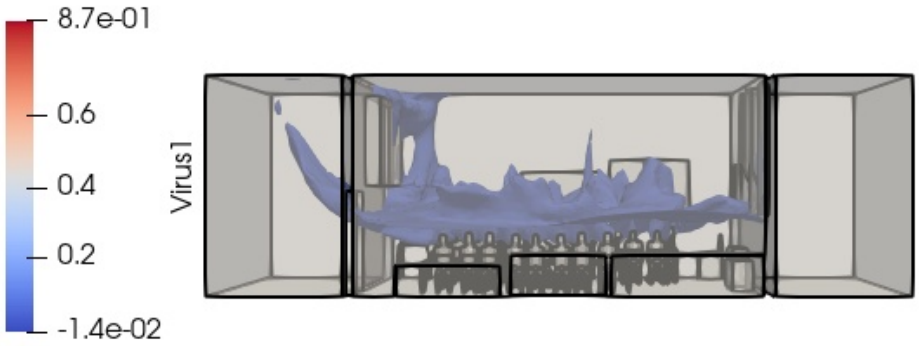
(c) Visualisation (Basic DA results) of CO_2 concentration level at time = 3305s

Figure 12: Visualisation results for Basic DA (CO_2). (a) is the result for sensor and (b) is the result for node, both of them are coloured by viral load.

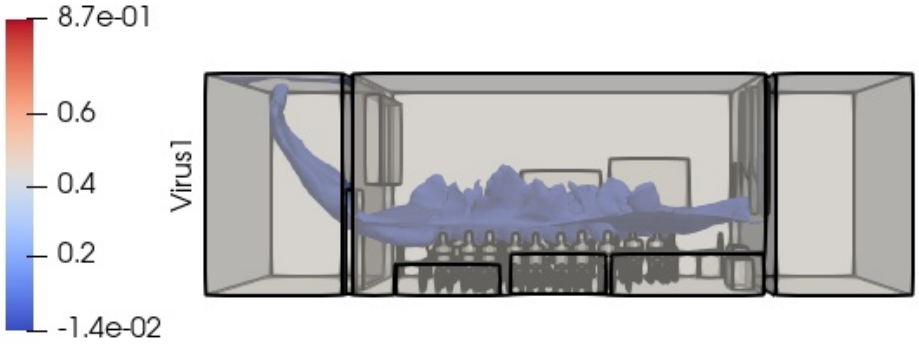
Part 4: Visualisation results of the Case 1 data assimilation



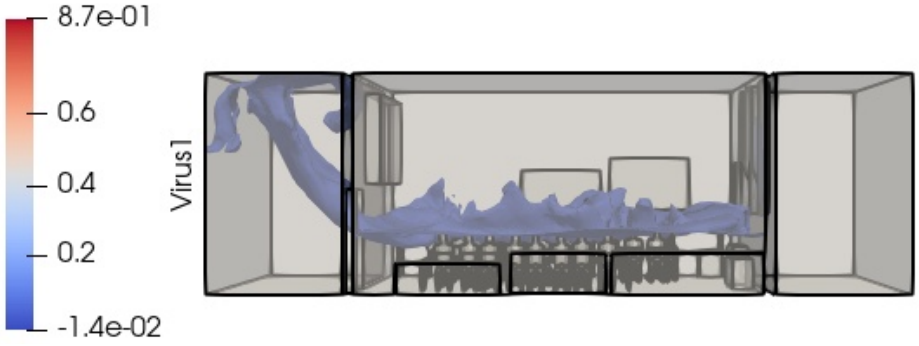
(a) Visualisation (Case 1) of CO_2 concentration level at time = 1800s



(b) Visualisation (Case 1) of CO_2 concentration level at time = 2400s

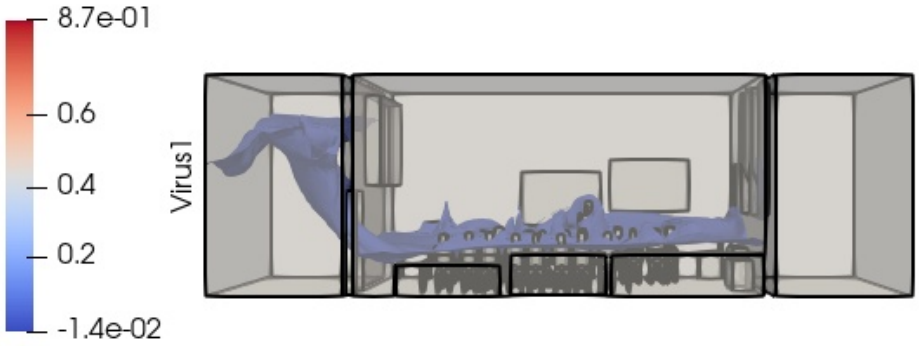


(c) Visualisation (Case 1) of CO_2 concentration level at time = 3200s

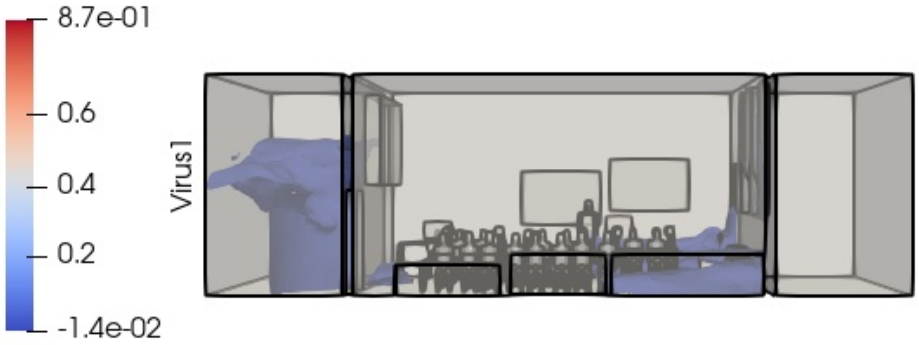


(d) Visualisation (Case 1) of CO_2 concentration level at time = 3600s

Figure 13: Visualisation results for Case 1 DA (CO_2). The values of (a)-(d) are set to 1000 ppm and coloured by viral load.



(a) Visualisation (Case 1) of viral load at time = 1800s



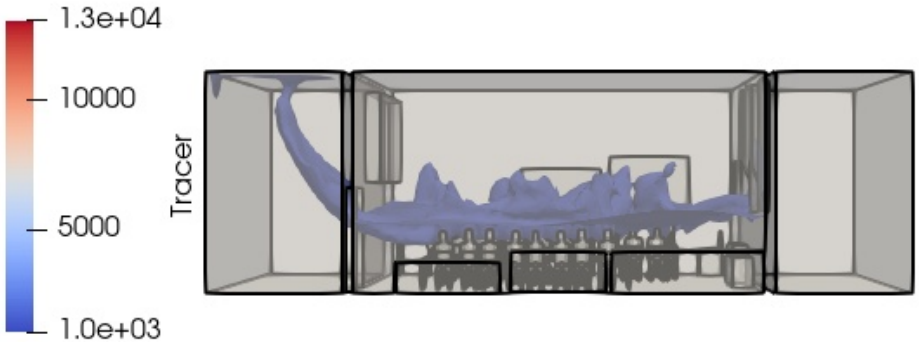
(b) Visualisation (Case 1) of viral load at time = 2400s

(c) Visualisation (Case 1) of viral load at time = 3200s

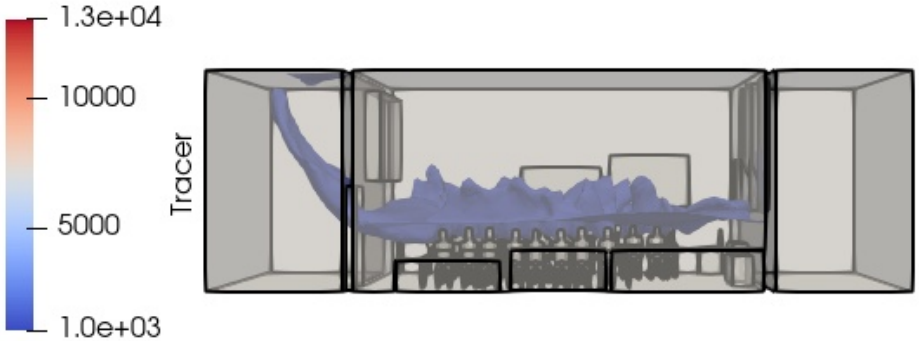
(d) Visualisation (Case 1) of viral load at time = 3600s

Figure 14: Visualisation results for Case 1 DA (viral load). The values of (a)-(d) are set to 0.01.

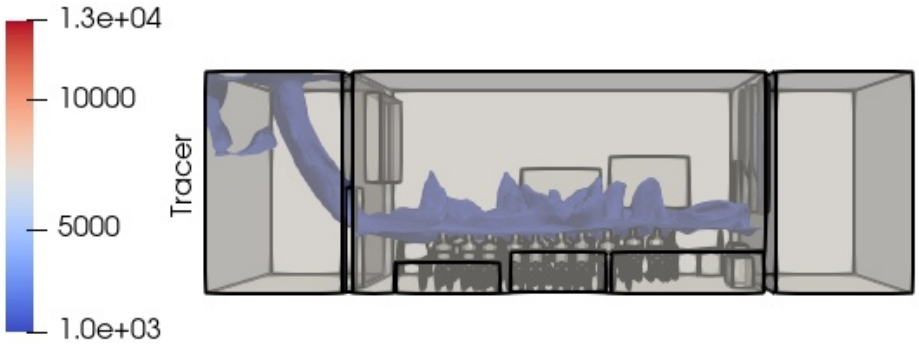
Part 5: Visualisation results of the Case 2 data assimilation



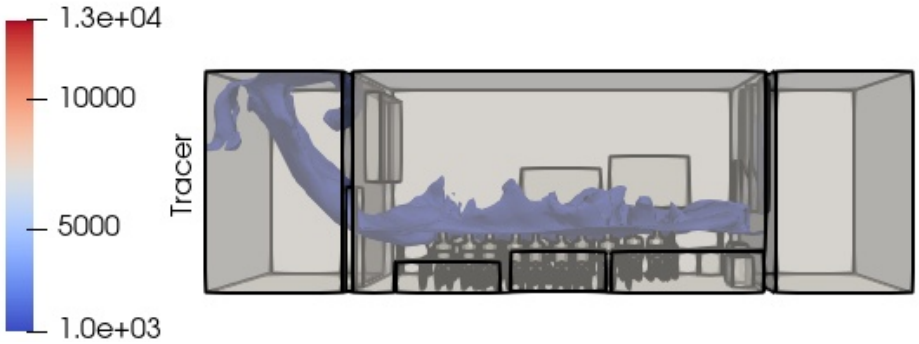
(a) Visualisation (Case 2) of CO_2 concentration level at time = 2880s



(b) Visualisation (Case 2) of CO_2 concentration level at time = 3120s

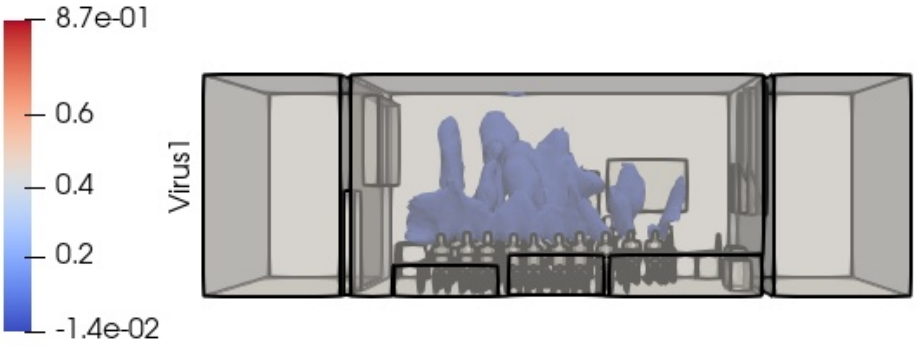


(c) Visualisation (Case 2) of CO_2 concentration level at time = 3360s

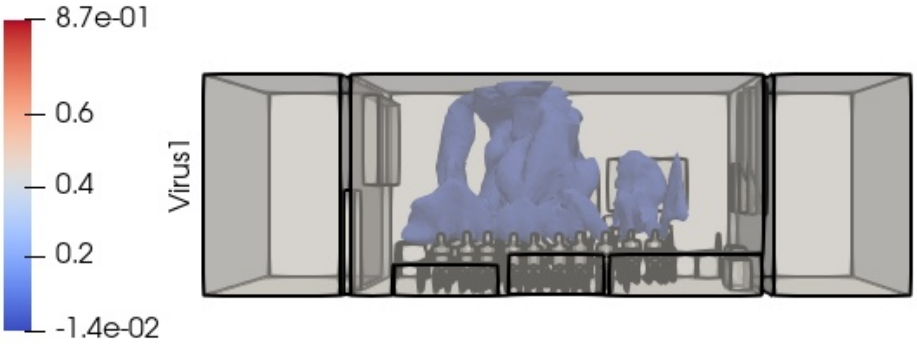


(d) Visualisation (Case 2) of CO_2 concentration level at time = 3600s

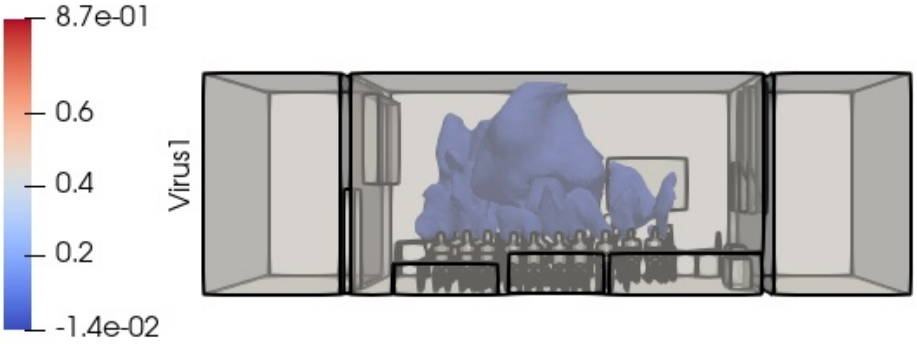
Figure 15: Visualisation results for Case 2 DA (CO_2). The values of (a)-(d) are set to 1000 ppm and coloured by viral load.



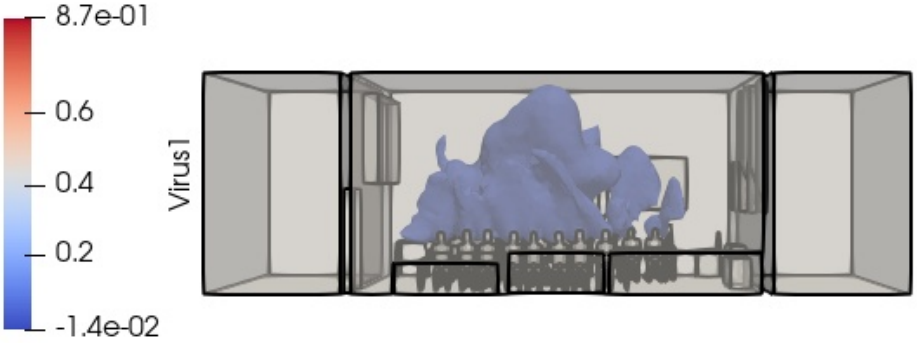
(a) Visualisation (Case 2) of viral load at time = 2880s



(b) Visualisation (Case 2) of viral load at time = 3120s



(c) Visualisation (Case 2) of viral load at time = 3360s



(d) Visualisation (Case 2) of viral load at time = 3600s

Figure 16: Visualisation results for Case 2 DA (viral load). The values of (a)-(d) are set to 0.03.

Part 6: Location of the sensors

$S_1 : [7.8, 4.8, 0.4]$

$S_2 : [3.9, 0, 0.4]$

$S_3 : [0, 4.8, 0.4]$

$S_4 : [7.8, 4.8, 1]$

$S_5 : [3.9, 0, 1]$

$S_6 : [0, 4.8, 1]$

$S_7 : [7.8, 4.8, 1.5]$

$S_8 : [3.9, 0, 1.5]$

$S_9 : [0, 4.8, 1.5]$

$S_{10} : [7.8, 4.8, 2.25]$

$S_{11} : [3.9, 0, 2.25]$

$S_{12} : [0, 4.8, 2.25]$

$S_{13} : [7.8, 4.8, 3]$

$S_{14} : [3.9, 0, 3]$

$S_{15} : [0, 4.8, 3]$

$S_{16} : [7.8, 4.8, 3.5]$

$S_{17} : [3.9, 0, 3.5]$

$S_{18} : [0, 4.8, 3.5]$