# CAAP Annual Report

**Date of Report**: *10/15/2024*

**Prepared for**: *U.S. DOT Pipeline and Hazardous Materials Safety Administration*

**Annual Period:** **From** *(September 27, 2023)* **to** *(September 26, 2024)*

**Contract Number**: *693JK32250011CAAP*

**Project Title**: *Determination of Potential Impact Radius for $CO_2$ Pipelines using Machine Learning Approach*

**Prepared by**: *Sam Wang*

**Contact Info.:** *979-845-9803 & gwang@tamu.edu*

# Table of Contents

# Section A: Business and Activities

## (a) Contract Activities

- Contract Modifications:

  Contract was officially extended between PHMSA and Texas A&M University (then internally with Texas A&M Engineering Experiment Station). NO COST amendment to Agreement# 693JK32250011CAAP is to extent the POP through September 25, 2025.

- Educational Activities:

  - Student mentoring: Chi-Yang Li, Jazmine Aiya D. Marquez

  - Student internship: Jazmine Aiya D. Marquez will conduct an internship next summer in 2025

  - Career employed: Pingfan Hu received his PhD degree in May 2023 and now employed by Atlas Copco Power Technique North America

- Others:

  - Dissemination of Project Outcomes: One invited seminar at the University of Arkansas and visit the low-speed wind tunnel for $CO_2$ near field dispersion study in Skylark JIP.

  - Presented in the DOE/DOT Interagency Meeting; Oral presentation is accepted in REX 2025 by PRCI

## (b) Financial Summary

- Federal Cost Activities:

  - PI/Co-PIs/students involvement: PI involvement with 0.75 month of time and efforts; Students with 12 months of time and efforts in total

  - Materials purchased/travel/contractual (consultants/subcontractors): Subcontractor NFPA cost for organizing TAP meeting and taking meeting minutes; no materials purchase and travel cost

- Cost Share Activities:

  - Cost share contribution: ~1 months of PI's time and efforts. He devoted his time to supervise the graduate students, organize TAP meetings, work with NFPA and other TAP members, review all work, technical trouble shooting for CFD, and submit the progress/annual reports.

## (c) Project Schedule Update

- Original Project Schedule:

| Task | Year 1 | | | | Year 2 | | | |
|---|---|---|---|---|---|---|---|---|
| | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| **Establish the CFD models of $CO_2$ release and dispersion from a high-pressure pipeline** | | | | | | | | |
| **Construct the database of $CO_2$ dispersion** | | | | | | | | |
| **Perform QPCR analysis and identify the PIR for $CO_2$ pipelines** | | | | | | | | |
| **Develop a web-based tool to determine the PIR for $CO_2$ pipelines and evacuation time for surrounding public** | | | | | | | | |

- Corrective Actions:

**Task 2** took about 5 quarters (year 1: Q3, Q4, year 2: Q1, Q2, Q3) and **Task 3** took about 3 quarters (year 2: Q3, Q4, year 3: Q1). We are now finishing Task 3 and will work on **Task 4** (year 3: Q2, Q3, Q4).

- Original Project Schedule:

| Task | Year 1 | | | | Year 2 | | | | Year 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| **1** | | | | | | | | | | | | |
| **2** | | | | | | | | | | | | |
| **3** | | | | | | | | | | | | |
| **4** | | | | | | | | | | | | |

## (d) Status Update of the 4th -8th Quarter Technical Activities

- **Task 2**: Construct the database of $CO_2$ dispersion under different scenarios
    - 2.1: Summarize the common $CO_2$ pipeline operating conditions and the dispersion parameters determined for CFD simulations
    - 2.2: Summarize the database for the PIR for $CO_2$ pipelines with different health consequences
- **Task 3**: Perform QPCR analysis and identify the PIR for CO2 pipelines
    - 3.1: Summarize the structure of database by utilizing scatter plots, histograms, and correlation matrices to visualize the continuous variables of the QPCR model
    - 3.2: Integrate the results from selecting suitable descriptors, constructing the QPCR model, and evaluating its performance using $R^2$ and RMSE metrics

## Section B: Detailed Technical Results in the Report Period

### 1. Background and Objectives in the 2nd Annual Report Period

#### 1.1. Objectives in the 2nd Annual Report Period

The primary objective of this project is to create a fast and widely applicable machine-learning based tool, based on simulations from CFD, for evaluating the outcomes of accidental $CO_2$ dispersion and establishing the PIR for $CO_2$ pipelines. Therefore, the proposed project will consist of four stages: (1) Establish the CFD models of $CO_2$ release and dispersion from a high-pressure pipeline; (2) Construct the database of $CO_2$ dispersion under different scenarios; (3) Perform QPCR analysis and identify the PIR for $CO_2$ pipelines; and (4) Develop a web-based tool to determine the PIR for $CO_2$ pipelines and evacuation time for the surrounding public. In this 2nd annual report, we mainly focus on **Stage 2 and part of Stage 3**.

### 2. Studies in the 2nd Annual Report Period

#### 2.1. Calculating the $CO_2$ behavior in the near field

As mentioned in the 1st annual report, 10 times of the distance of Mach disc ($x_m$) from the pipe could be considered as the distance of near field.

$$x_m = 0.6455 \times d_e \times \sqrt{\frac{P_0}{P_a}}$$

Where $d_e$ is the diameter of the nozzle exit, $P_0$ is the stagnation pressure, and $P_a$ is the ambient pressure.

We can then calculate the velocity at the end of the near field based on the assumptions of no ambient fluid entrainment, isentropic flow relationships, and constant pressure at the rupture

point of the pipeline (Birch et al., 1987).

$$V_{CO_2} = V_0 \left\{ C_D + \frac{\left[ 1 - \frac{P_a}{P_0} \times \left( \frac{2}{\gamma + 1} \right)^{\frac{-\gamma}{\gamma - 1}} \right]}{\gamma C_D} \right\}$$

Where $V_{CO_2}$ is the velocity of $CO_2$ in the atmosphere, $V_0$ is the velocity in the pipeline, $C_D$ (1 for the well-rounded nozzle) is the volume discharge coefficient, $\gamma$ (1.30 for $CO_2$) is the ratio of the heat capacities.

In our previous near field simulations (Figure 1-2), it was observed that a large amount of air was entrained, which implied significant pressure drops as $CO_2$ transitioned from the pipeline to the atmosphere. Given the relatively low atmospheric pressure (1 atm), we can use the ideal gas law and the conservation of mass equation to calculate the cross-sectional area of the fluid. However, the cross-section area is the function of $CO_2$ composition, while $CO_2$ composition is also the function of cross-section from calculating the results from simulations (Figure 2). After some trial and error, the $CO_2$ mass fraction was calculated to be 0.2644.

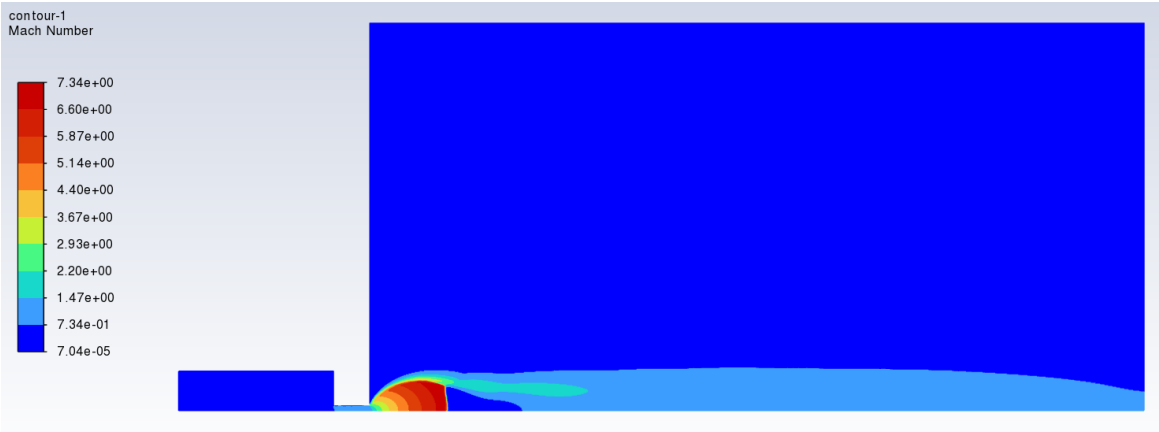


Figure 1. Near field simulations.

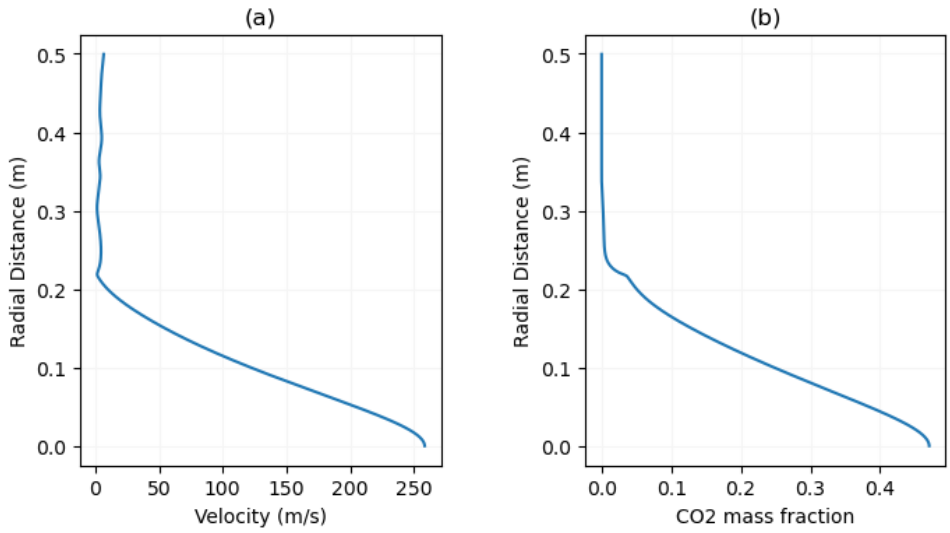


Figure 2. Data from near field simulations: (a) Velocity versus radial distance; (b) $CO_2$ mass fraction versus radial distance.

Because the goal for the simulations is for safety purposes, overestimation is preferred. Therefore, we assumed the fluid composition at the end of the near field to be 30% $CO_2$ and 70% air. Based on this, we can calculate the fluid velocity as follows.

$$V_a = V_{wind} \times 0.7 + V_{CO_2} \times 0.3$$

Where $V_a$ is the velocity of fluid in the atmosphere, and $V_{wind}$ is the velocity of wind.

Furthermore, since the scenario involves $CO_2$ release from both ends of the ruptured pipeline, we double the mass flow rate for the simulations. As fluid velocity is a critical factor for dispersion, we also double the release cross-sectional area, using the previously calculated velocity, to run the simulations. Consequently, the fluid composition, velocity, and area are used to represent the near-field behavior, and Ansys Fluent is applied to simulate the far-field dispersion.

### 2.2. CFD Far-field stage

With calculated near field behavior, we could integrate the velocity, $CO_2$ mass fraction, weather conditions, parameters from Table 1, and five geometries (

Figure 3-Figure 7) to conduct the simulation for far field stage to investigate $CO_2$ concentration versus the distance.

Table 1. The variables for pipeline characteristics and weather conditions.

| | Variable | High | Medium | Low |
|---|---|---|---|---|
| Pipeline characteristics | pressure (MPa) | 20 | 10 | 1 |
| | diameter (inch) | 30 | 16 | 4 |
| | flow rate (MMcfd) | 1300 | 590 | 30 |
| Weather conditions | wind speed (mph) | 25 | 14 | 3 |
| | temperature (°F) | 100 | 60 | 0 |

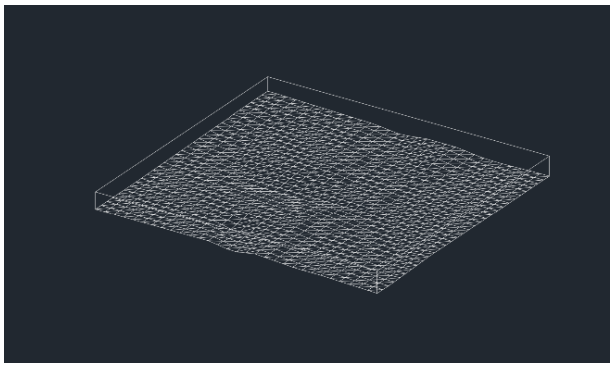

Figure 3. Monticello, Mississippi (Flat)

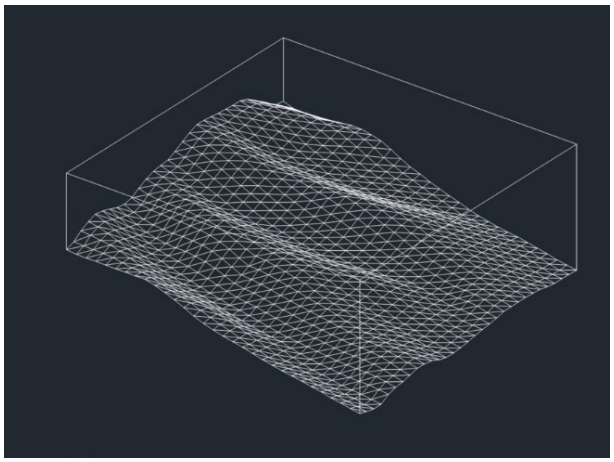

Figure 4. Walsenburg, Colorado (Medium Hill, SH)

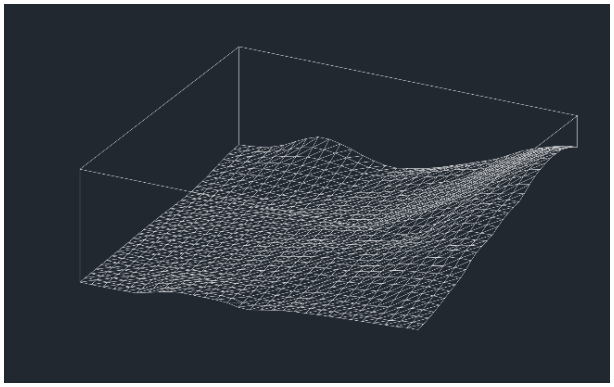

Figure 5. Raton, New Mexico (Big Hill, BH)

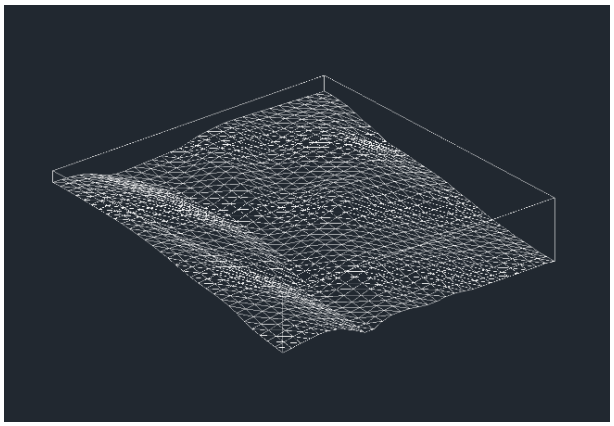

Figure 6. Calistoga, California (Medium Valley, VM)

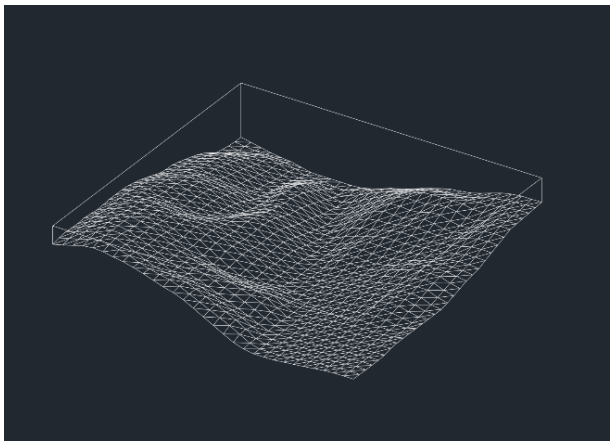

Figure 7. Vernal, Utah (Big Valley, VB)

However, there are over one thousand various scenarios, so it would be extremely difficult to conduct all of them manually. Fortunately, there is a Python package, named *PyFluent* which is provided by Ansys Fluent, which could do us a favor. Thus, once the meshes were created, the subsequent process involved repetitive tasks, such as adjusting parameters, running simulations, and saving results. To streamline this, we used *PyFluent* package to automate the steps and execute the simulations. Therefore, all the tasks could be conducted in more efficient way. The boxplots for each of them are shown in Figure 8-Figure 12, while the histograms for each scenarios are displayed in Figure 13-Figure 17.

From the data, we observed that the sequence of CO2 dispersion distances, from farthest to shortest, follows this order: flat terrain, medium valley, medium hill, big valley, and big hill. These results differ from our initial expectation that the valleys would allow CO2 to disperse the farthest. The reason is that the valleys the CO2 pipeline passes through are located on hills with slopes. Therefore, these valleys are not flat-bottomed, like Santa Elena Canyon in Big Bend National Park. As a result, the slopes hindered CO2 dispersion in the valleys, preventing it from reaching farther than on flat terrain. However, the valley still could have farther dispersion.

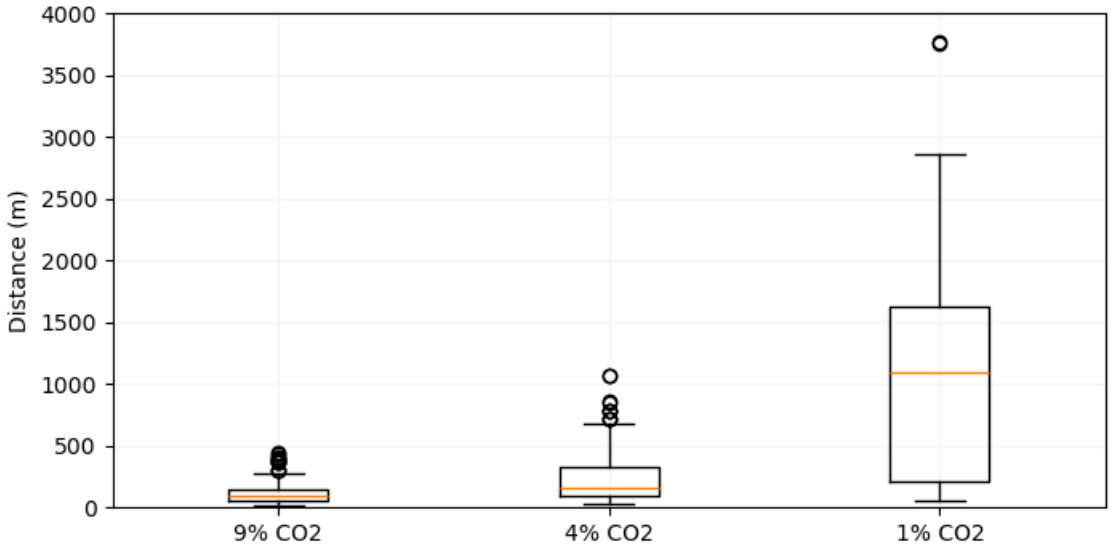


Figure 8. Boxplot of distances for Flat.

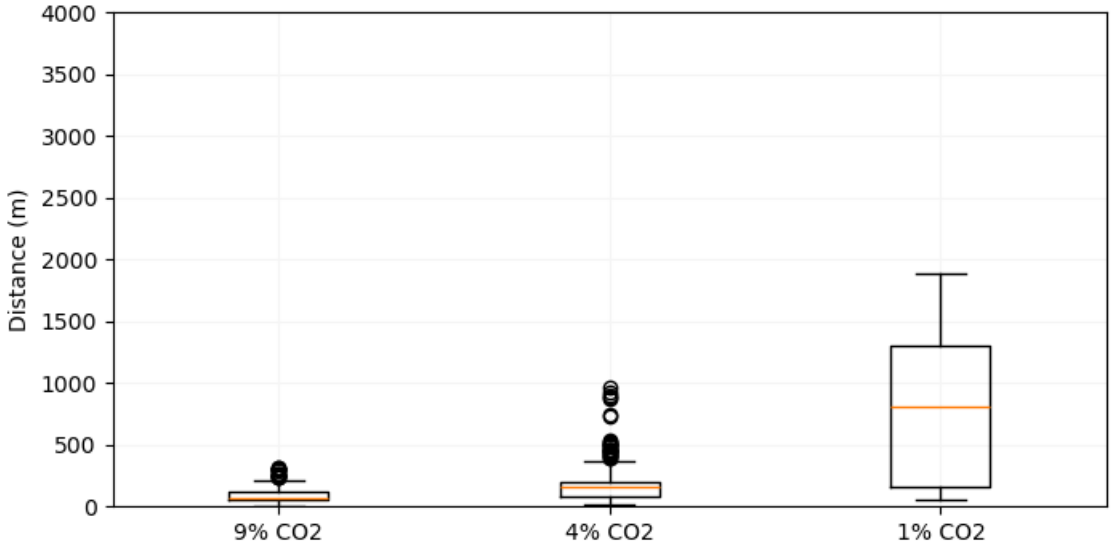


Figure 9. Boxplot of distances for SH.

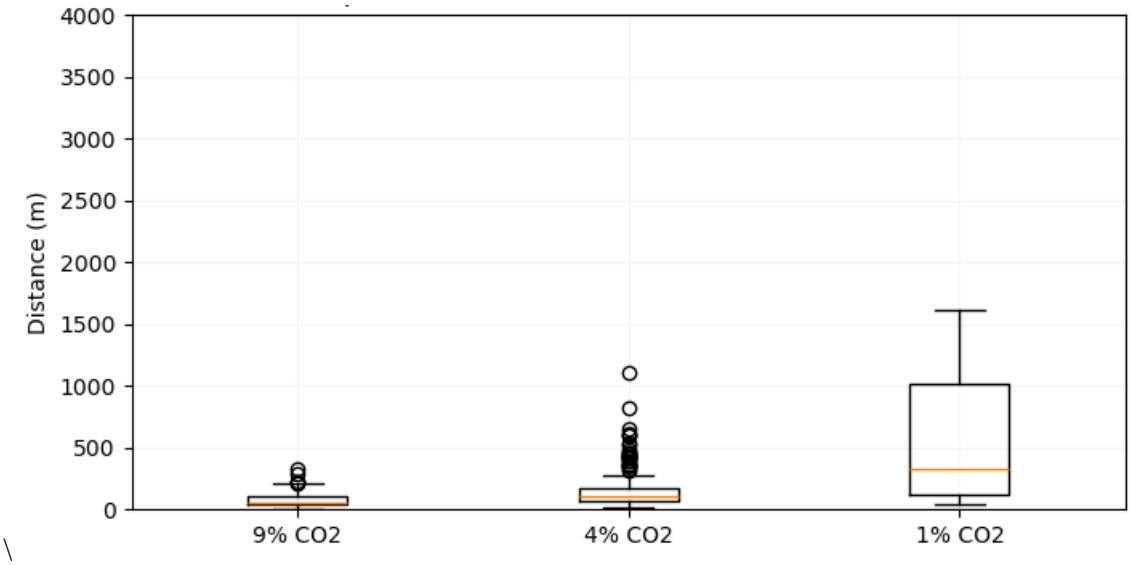


Figure 10. Boxplot of distances for BH.

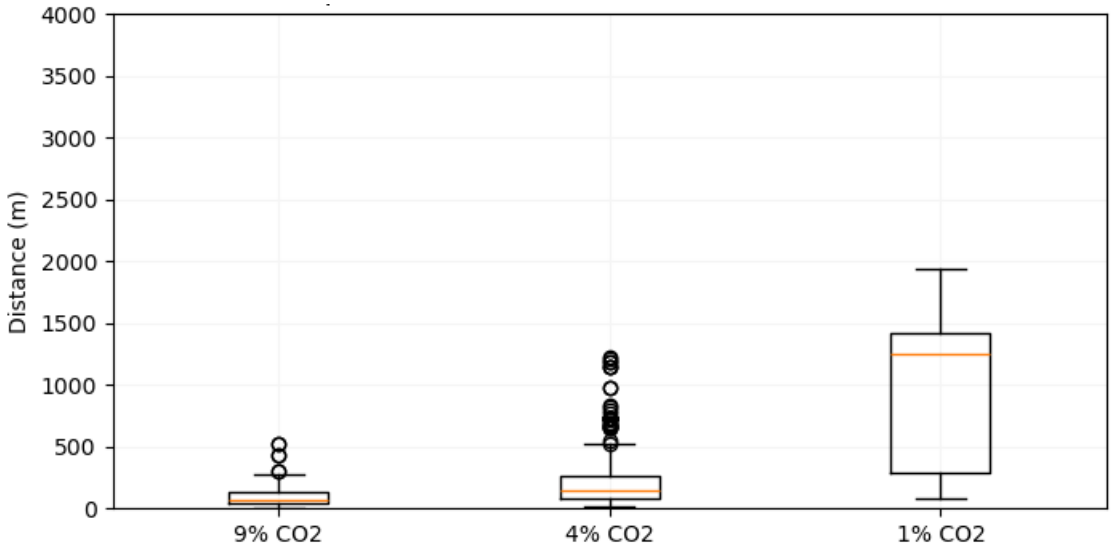


Figure 11. Boxplot of distances for VM.

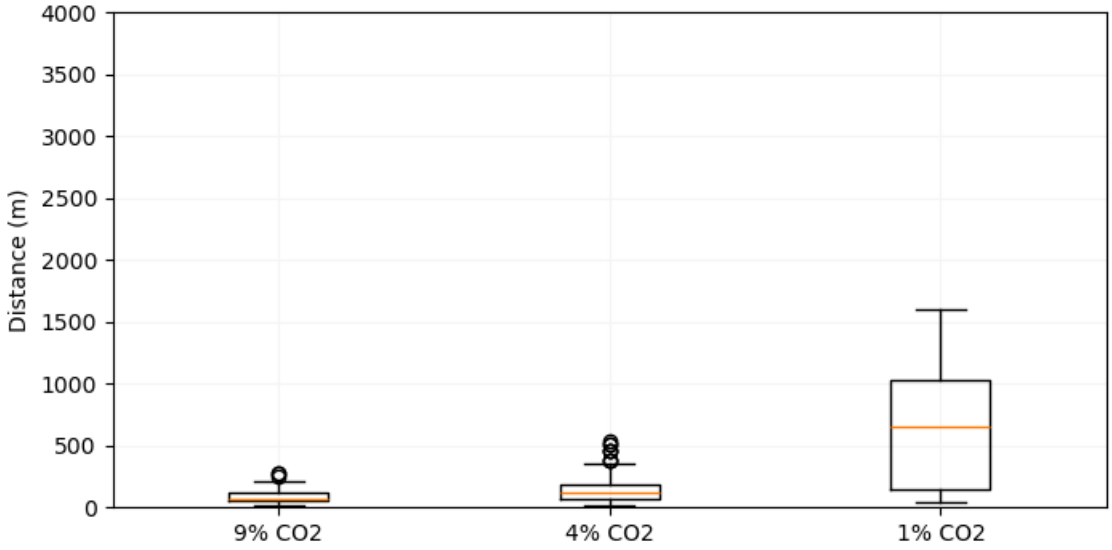


Figure 12. Boxplot of distances for VB.

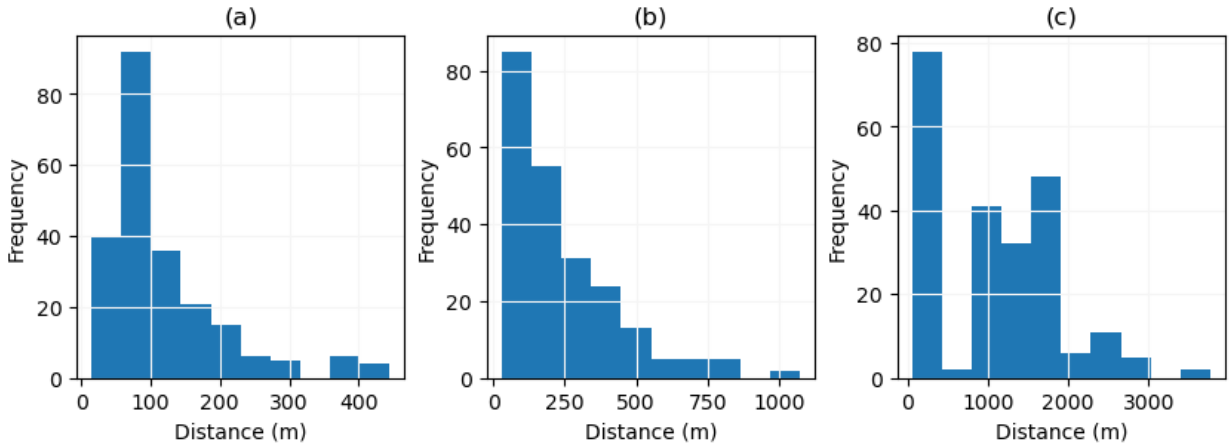


Figure 13. Histograms for Flat: (a) 9% $CO_2$; (b) 4% $CO_2$; (c) 1% $CO_2$.

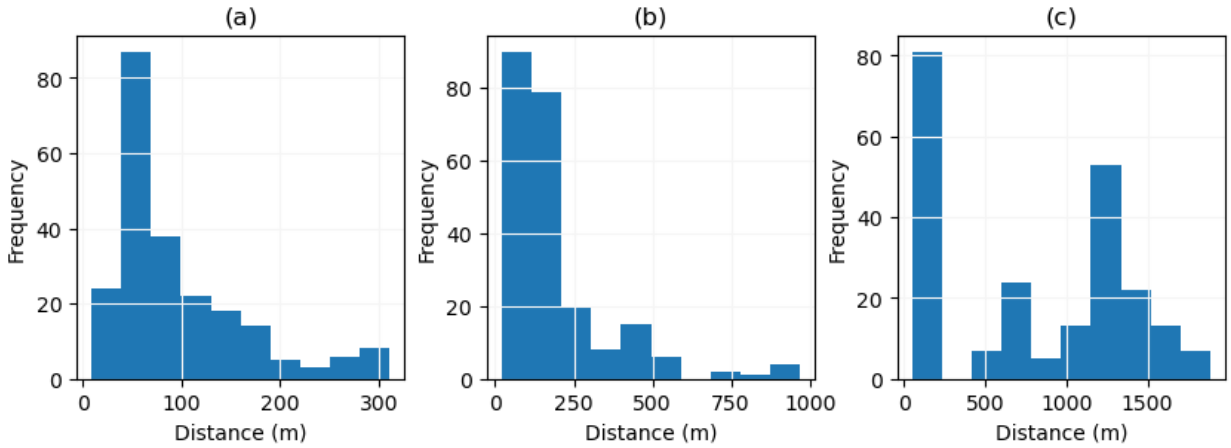


Figure 14. Histograms for SH: (a) 9% $CO_2$; (b) 4% $CO_2$; (c) 1% $CO_2$.

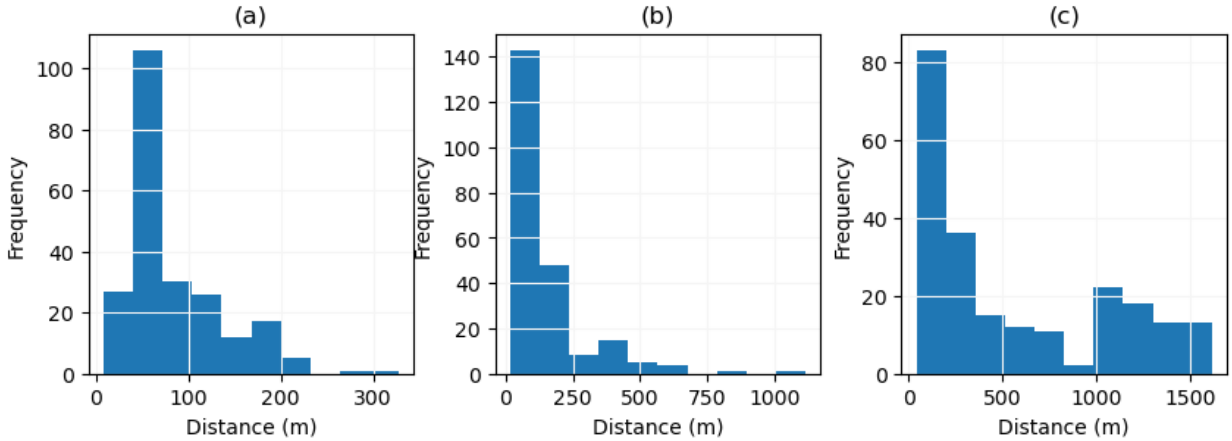


Figure 15. Histograms for BH: (a) 9% $CO_2$; (b) 4% $CO_2$; (c) 1% $CO_2$.

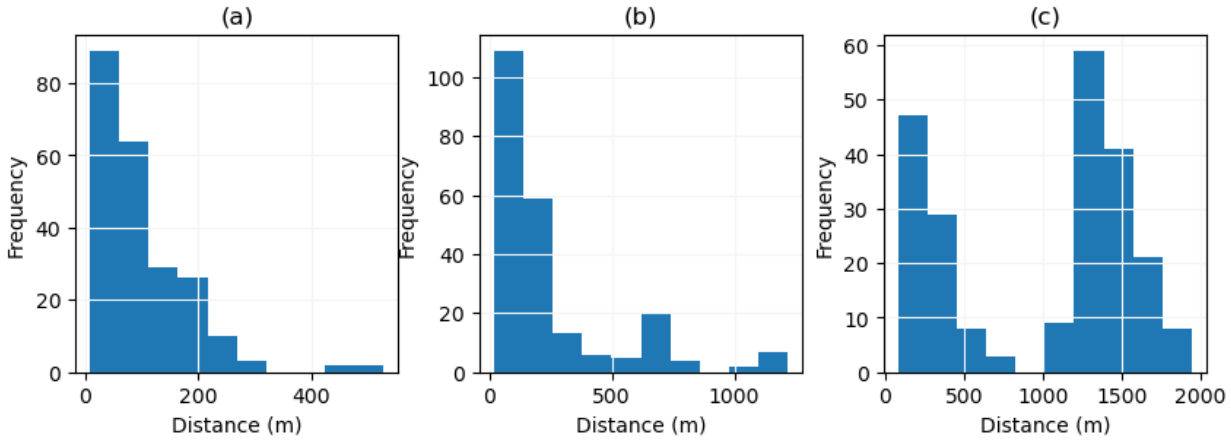


Figure 16. Histograms for VM: (a) 9% $CO_2$; (b) 4% $CO_2$; (c) 1% $CO_2$.

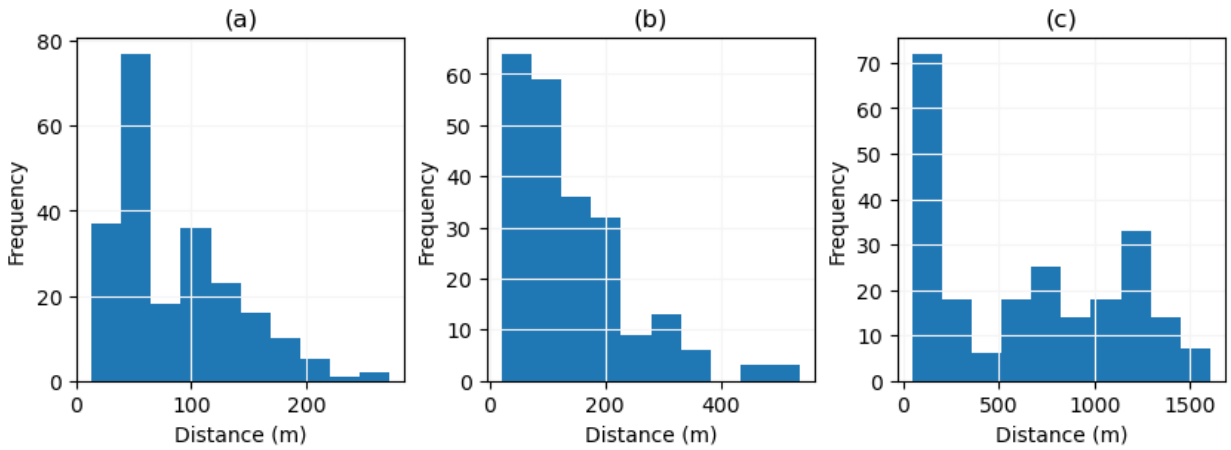


Figure 17. Histograms for VB: (a) 9% $CO_2$; (b) 4% $CO_2$; (c) 1% $CO_2$.

### 2.3. Quantitative Property Consequence Relationship Models (QPCR)

The use of machine learning algorithms to train large consequence databases for comprehensive consequence prediction was first introduced by Sun et al., (2020). This approach was later formalized by (Jiao et al., 2020) who applied it to flammable dispersion in 2020, naming the method quantitative property-consequence relationship (QPCR) analysis. QPCR was inspired by the well-established quantitative structure-property relationship (QSPR) method, which uses structural attributes as descriptors to build mathematical relationships between molecular structures and properties at the quantum chemistry level (Jiao et al., 2019). A procedural diagram of the QSPR model development is shown in Figure 18. While QSPR has been widely used for hazardous property prediction, QPCR differs in that it uses property descriptors as independent variables and quantifies consequence values as dependent variables for predicting outcomes. As illustrated in Figure 18, the steps of descriptor calculation and screening in QSPR are replaced by consequence data generation and collection in QPCR. This method bridges the gap between microscale chemical properties and macroscale consequences, offering a promising approach for developing more reliable and broadly applicable predictions (Jiao et al., 2020).

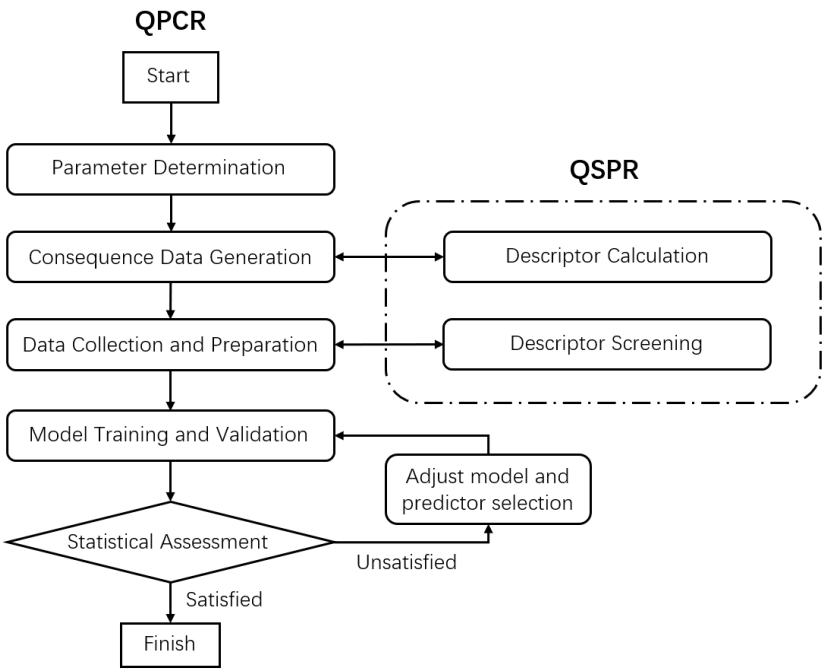


Figure 18. QPCR Development Procedure Diagram

For the project, descriptors are pressure (MPa), diameter (inch), flow rate (MMcfd), wind speed (mph), and temperature (°F). Therefore, we used these features to predict the distances of 9% $CO_2$, 4% $CO_2$, and 1 % $CO_2$. Due to the significant variation in the distance distributions across the three different concentrations, we developed separate models for each concentration. Additionally, because the distributions of distances are skewed (Figure 13-Figure 17), logarithm transformation were applied to the distances, which made the distributions closer to the normal distribution. Furthermore, the correlation matrices for each geometry are shown in Figure 19-Figure 23. From the correlation matrices, it demonstrates that the flow rate has the highest correlation coefficient with distances.

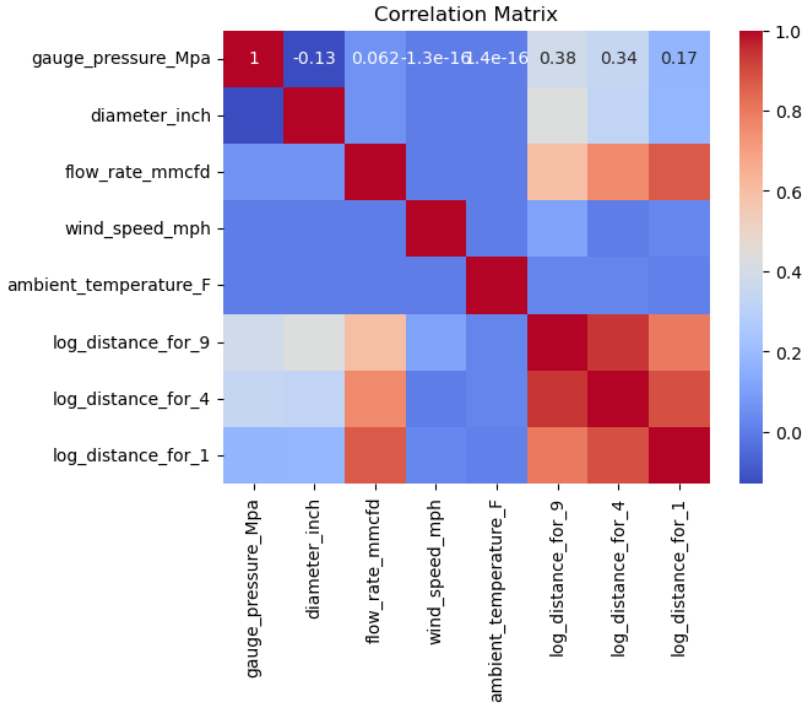


Figure 19. Correlation matric for Flat.

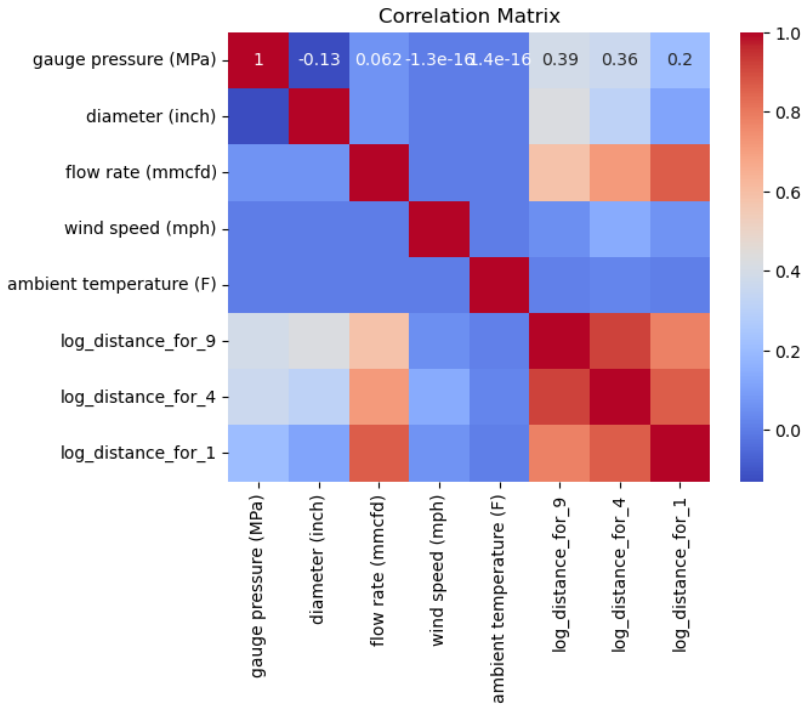


Figure 20. Correlation matric for SH.

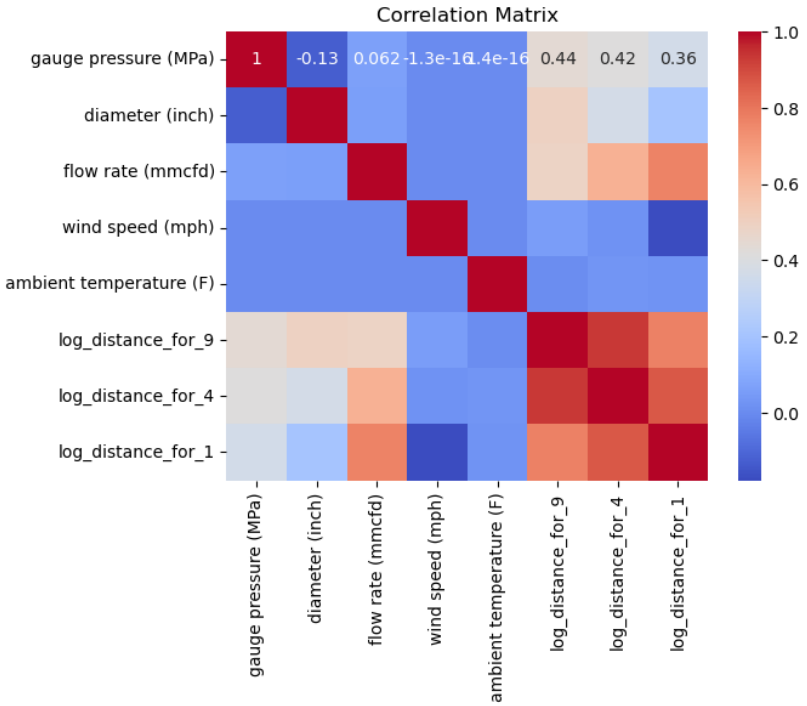


Figure 21. Correlation matric for BH.

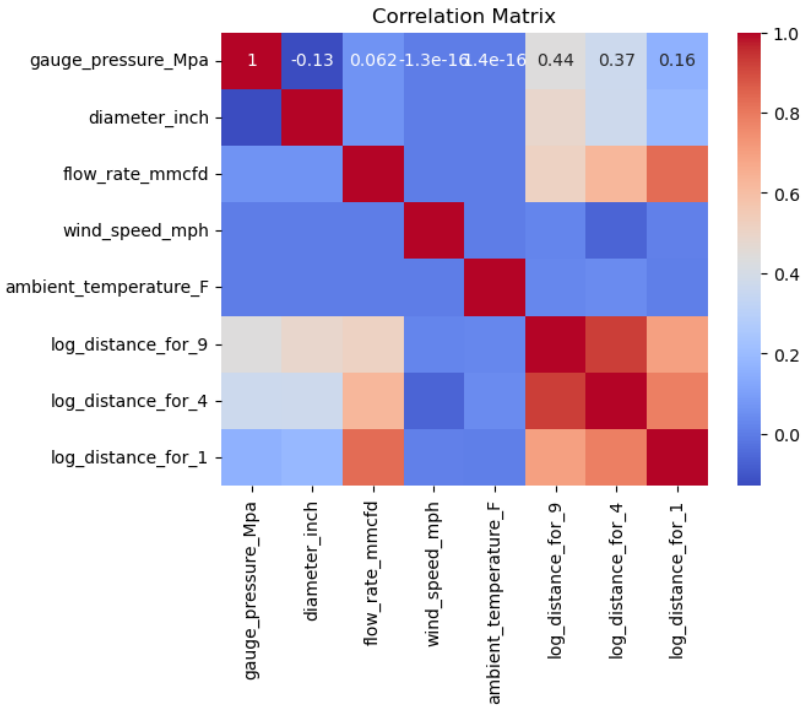


Figure 22. Correlation matric for VM.

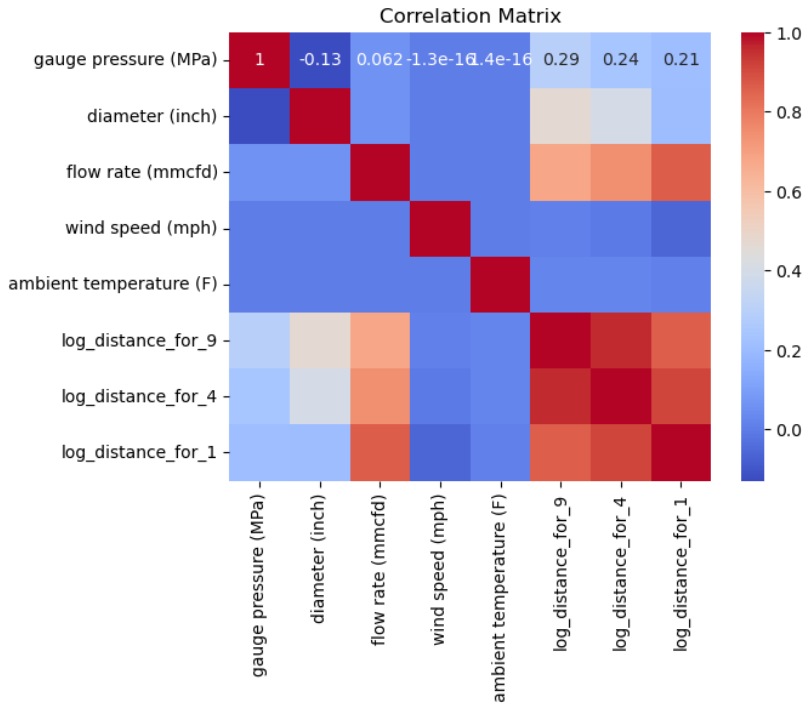


Figure 23. Correlation matric for VB.

On the other hand, the machine learning models used to identify the best-performing model included multiple linear regression (MLR), support vector regression (SVR), k-nearest neighbors (KNN), random forest (RF), extreme gradient boosting regression (XGBoost), gradient boosting regression (GBR), and bootstrap aggregating (Bagging). $R^2$ scores, along with 10-fold cross-validation, were used to select the best model and evaluate performance. In each model, the input features were gauge pressure, pipeline diameter, $CO_2$ flow rate, wind speed, and ambient temperature, while the output (response) was the corresponding distance from the simulation. A random search of hyperparameters, considered a more efficient method for optimizing model performance, was conducted for each model. The best version of each machine learning model is presented in Tables 7, 9, 11, 13, and 15, with the hyperparameters of the best models shown in Tables 8, 10, 12, 14, and 16. The 10-fold cross-validation predictions are illustrated in Figures 24

to 28. All models achieved an $R^2$ score higher than 0.93, indicating high prediction accuracy.

Table 2. Performance for each fine-tuned machine learning model for Flat.

| $CO_2$ concentration (%) | Model | $R^2$ | SD |
|---|---|---|---|
| 9 | Gradient Boosting | 0.9665 | 0.0384 |
| | Bagging | 0.9691 | 0.0301 |
| | Random Forest | 0.9688 | 0.0300 |
| | XGBoost | 0.9782 | 0.0286 |
| | K nearest neighbors | 0.6703 | 0.1188 |
| | Multiple Linear Regression | 0.4806 | 0.1564 |
| | Support Vector Regression | 0.7775 | 0.0939 |
| 4 | Gradient Boosting | 0.9635 | 0.0270 |
| | Bagging | 0.9600 | 0.0355 |
| | Random Forest | 0.9604 | 0.0352 |
| | XGBoost | 0.9690 | 0.0453 |
| | K nearest neighbors | 0.7520 | 0.1533 |
| | Multiple Linear Regression | 0.5468 | 0.1470 |
| | Support Vector Regression | 0.7869 | 0.1187 |
| 1 | Gradient Boosting | 0.9849 | 0.0125 |
| | Bagging | 0.9833 | 0.0119 |
| | Random Forest | 0.9836 | 0.0105 |
| | XGBoost | 0.9886 | 0.0112 |
| | K nearest neighbors | 0.9242 | 0.0393 |
| | Multiple Linear Regression | 0.7943 | 0.0795 |
| | Support Vector Regression | 0.9236 | 0.0333 |

Table 3. Performance for each fine-tuned machine learning model for SH.

| $CO_2$ concentration (%) | Model | $R^2$ | SD |
|---|---|---|---|
| 9 | Gradient Boosting | 0.9830 | 0.0348 |
| | Bagging | 0.9804 | 0.0124 |
| | Random Forest | 0.9806 | 0.0122 |
| | XGBoost | 0.9918 | 0.0093 |
| | K nearest neighbors | 0.6650 | 0.1575 |
| | Multiple Linear Regression | 0.4114 | 0.2990 |
| | Support Vector Regression | 0.7682 | 0.1172 |
| 4 | Gradient Boosting | 0.9672 | 0.0213 |
| | Bagging | 0.9663 | 0.0282 |
| | Random Forest | 0.9665 | 0.0251 |
| | XGBoost | 0.9700 | 0.0346 |
| | K nearest neighbors | 0.7345 | 0.1029 |
| | Multiple Linear Regression | 0.4470 | 0.1690 |
| | Support Vector Regression | 0.7738 | 0.0767 |
| 1 | Gradient Boosting | 0.9940 | 0.0039 |
| | Bagging | 0.9917 | 0.0043 |
| | Random Forest | 0.9918 | 0.0048 |
| | XGBoost | 0.9950 | 0.0026 |
| | K nearest neighbors | 0.9474 | 0.0263 |
| | Multiple Linear Regression | 0.7764 | 0.0841 |
| | Support Vector Regression | 0.9490 | 0.0292 |

Table 4. Performance for each fine-tuned machine learning model for BH.

| $CO_2$ concentration (%) | Model | $R^2$ | SD |
|---|---|---|---|
| 9 | Gradient Boosting | 0.9875 | 0.0079 |
| | Bagging | 0.9794 | 0.0109 |
| | Random Forest | 0.9795 | 0.0110 |
| | XGBoost | 0.9878 | 0.0075 |
| | K nearest neighbors | 0.6632 | 0.0999 |
| | Multiple Linear Regression | 0.5376 | 0.1397 |
| | Support Vector Regression | 0.7669 | 0.0921 |

| | | | |
|---|---|---|---|
| 4 | Gradient Boosting | 0.9301 | 0.0409 |
| | Bagging | 0.9272 | 0.0698 |
| | Random Forest | 0.9301 | 0.0629 |
| | XGBoost | 0.9288 | 0.0545 |
| | K nearest neighbors | 0.6416 | 0.1877 |
| | Multiple Linear Regression | 0.2711 | 0.3867 |
| | Support Vector Regression | 0.7174 | 0.1020 |
| 1 | Gradient Boosting | 0.9605 | 0.0237 |
| | Bagging | 0.9566 | 0.0293 |
| | Random Forest | 0.9575 | 0.0271 |
| | XGBoost | 0.9627 | 0.0210 |
| | K nearest neighbors | 0.7759 | 0.1128 |
| | Multiple Linear Regression | 0.5947 | 0.1612 |
| | Support Vector Regression | 0.8121 | 0.0556 |

Table 5. Performance for each fine-tuned machine learning model for VM.

| $CO_2$ concentration (%) | Model | $R^2$ | SD |
|---|---|---|---|
| 9 | Gradient Boosting | 0.9618 | 0.0364 |
| | Bagging | 0.9567 | 0.0298 |
| | Random Forest | 0.9574 | 0.0242 |
| | XGBoost | 0.9725 | 0.0220 |
| | K nearest neighbors | 0.6552 | 0.1310 |
| | Multiple Linear Regression | 0.4830 | 0.1194 |
| | Support Vector Regression | 0.7775 | 0.1117 |
| 4 | Gradient Boosting | 0.9160 | 0.0592 |
| | Bagging | 0.9232 | 0.0656 |
| | Random Forest | 0.9244 | 0.0624 |
| | XGBoost | 0.9330 | 0.0896 |
| | K nearest neighbors | 0.6489 | 0.1178 |
| | Multiple Linear Regression | 0.3963 | 0.0786 |
| | Support Vector Regression | 0.6946 | 0.1359 |
| 1 | Gradient Boosting | 0.9907 | 0.0092 |
| | Bagging | 0.9801 | 0.0091 |

| | | | |
|---|---|---|---|
| | Random Forest | 0.9801 | 0.0086 |
| | XGBoost | 0.9930 | 0.0054 |
| | K nearest neighbors | 0.8853 | 0.0364 |
| | Multiple Linear Regression | 0.7816 | 0.0701 |
| | Support Vector Regression | 0.8801 | 0.0260 |

Table 6. Performance for each fine-tuned machine learning model for VB.

| $CO_2$ concentration (%) | Model | $R^2$ | SD |
|---|---|---|---|
| 9 | Gradient Boosting | 0.9656 | 0.0176 |
| | Bagging | 0.9713 | 0.0225 |
| | Random Forest | 0.9714 | 0.0228 |
| | XGBoost | 0.9762 | 0.0238 |
| | K nearest neighbors | 0.7500 | 0.1345 |
| | Multiple Linear Regression | 0.5813 | 0.1287 |
| | Support Vector Regression | 0.8231 | 0.1026 |
| 4 | Gradient Boosting | 0.9428 | 0.0444 |
| | Bagging | 0.9462 | 0.0345 |
| | Random Forest | 0.9480 | 0.0326 |
| | XGBoost | 0.9626 | 0.0264 |
| | K nearest neighbors | 0.7567 | 0.1161 |
| | Multiple Linear Regression | 0.4461 | 0.1998 |
| | Support Vector Regression | 0.7800 | 0.0954 |
| 1 | Gradient Boosting | 0.9942 | 0.0044 |
| | Bagging | 0.9859 | 0.0047 |
| | Random Forest | 0.9861 | 0.0047 |
| | XGBoost | 0.9952 | 0.0028 |
| | K nearest neighbors | 0.8901 | 0.0305 |
| | Multiple Linear Regression | 0.7897 | 0.0968 |
| | Support Vector Regression | 0.8940 | 0.0269 |

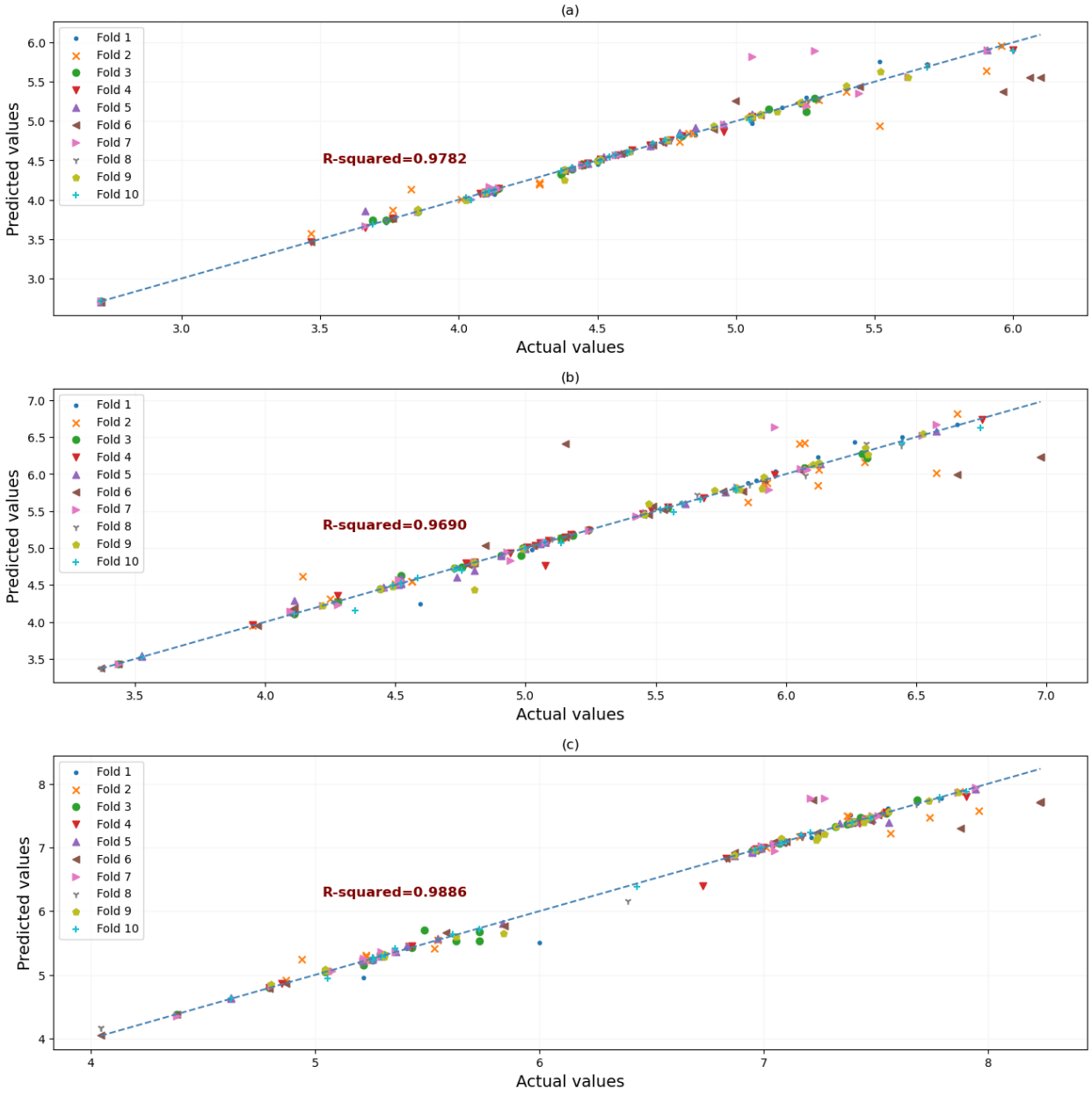


Figure 24. Actual vs. Predicted distances (10-fold cross validation) for Flat: (a) Distance for 9% $CO_2$, (b) Distance for 4% $CO_2$, and (c) Distance for 1% $CO_2$.

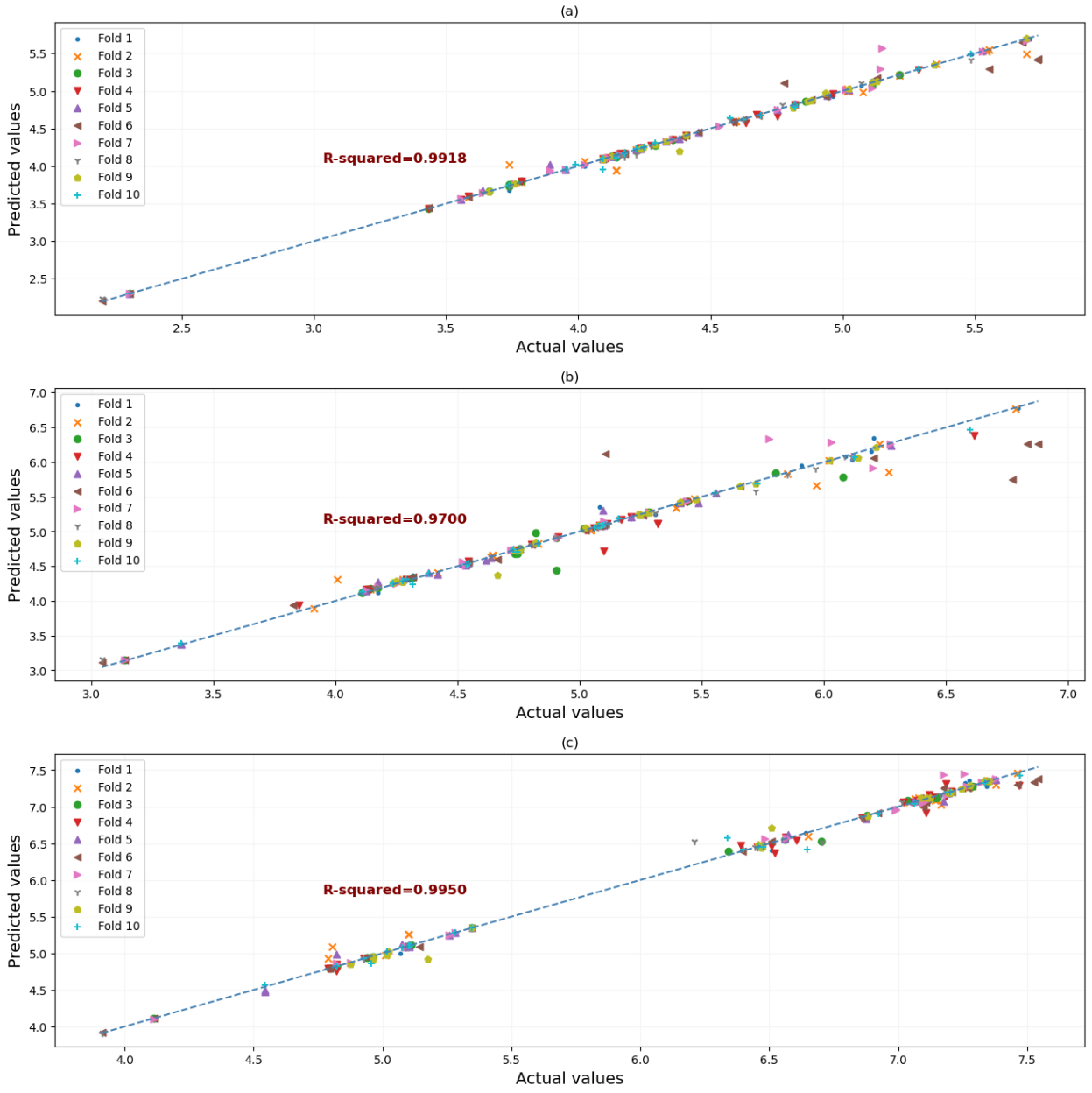


Figure 25. Actual vs. Predicted distances (10-fold cross validation) for SH: (a) Distance for 9% $CO_2$, (b) Distance for 4% $CO_2$, and (c) Distance for 1% $CO_2$.

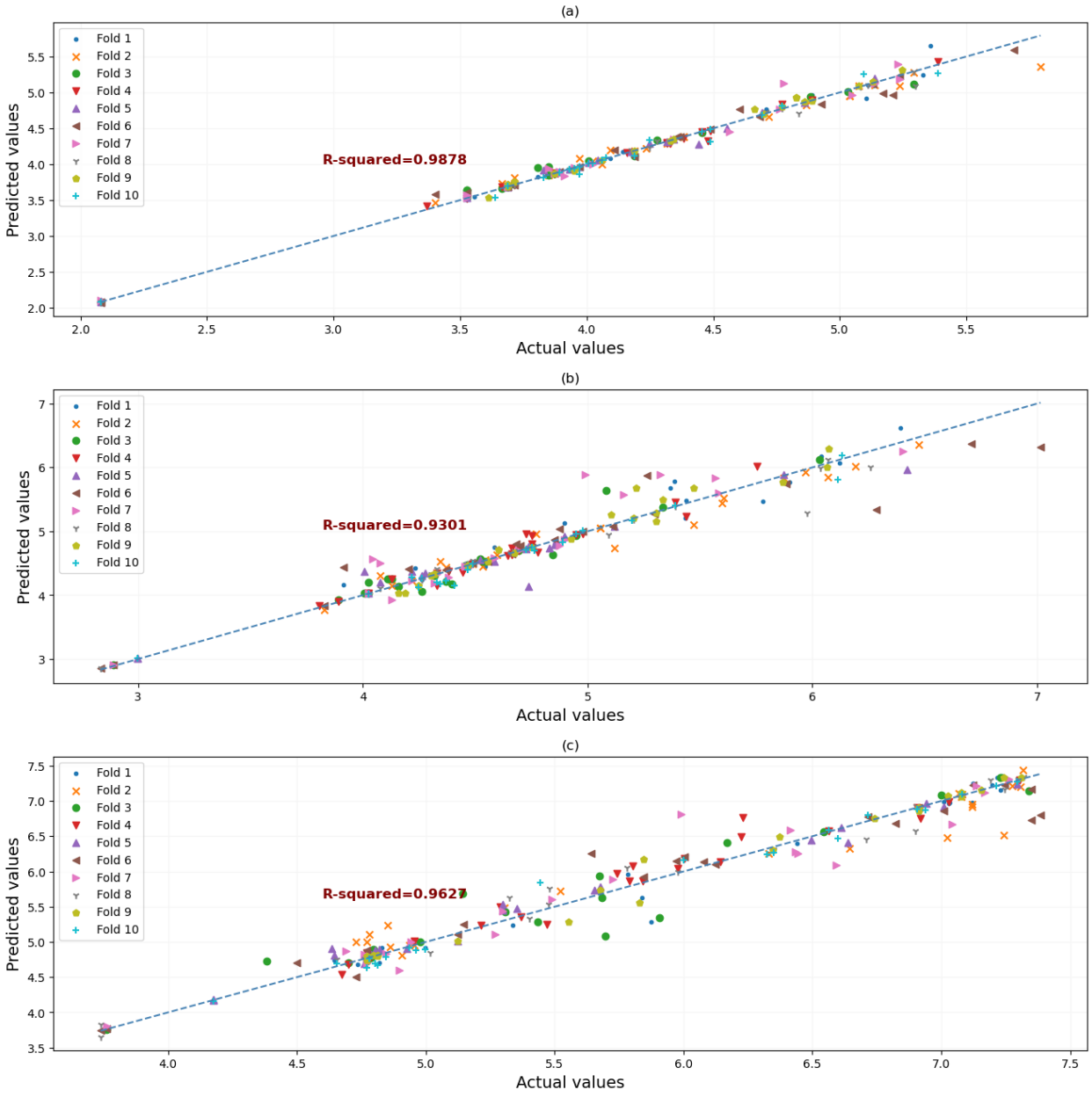


Figure 26. Actual vs. Predicted distances (10-fold cross validation) for BH: (a) Distance for 9% $CO_2$, (b) Distance for 4% $CO_2$, and (c) Distance for 1% $CO_2$.

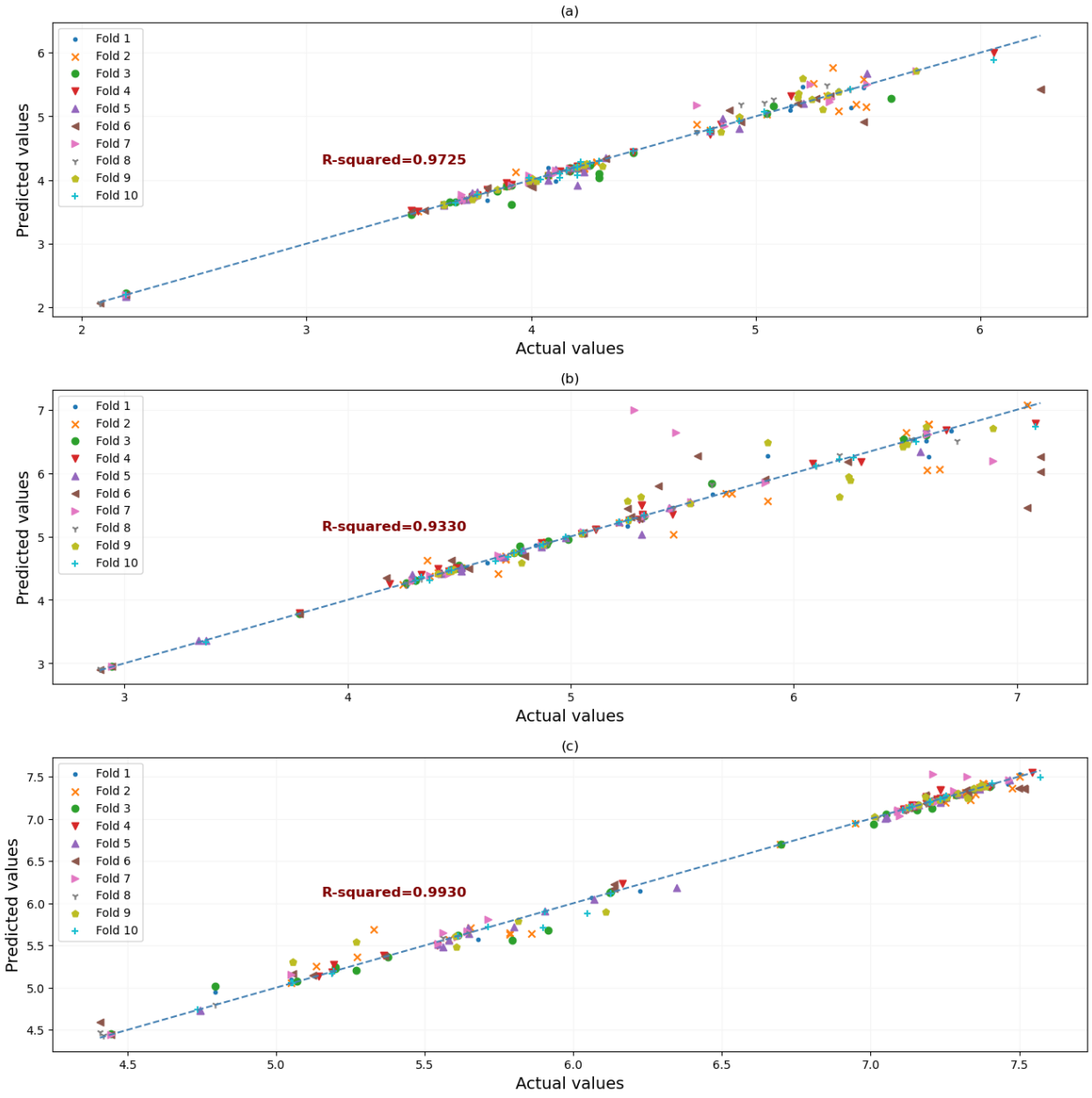


Figure 27. Actual vs. Predicted distances (10-fold cross validation) for VM: (a) Distance for 9% $CO_2$, (b) Distance for 4% $CO_2$, and (c) Distance for 1% $CO_2$.

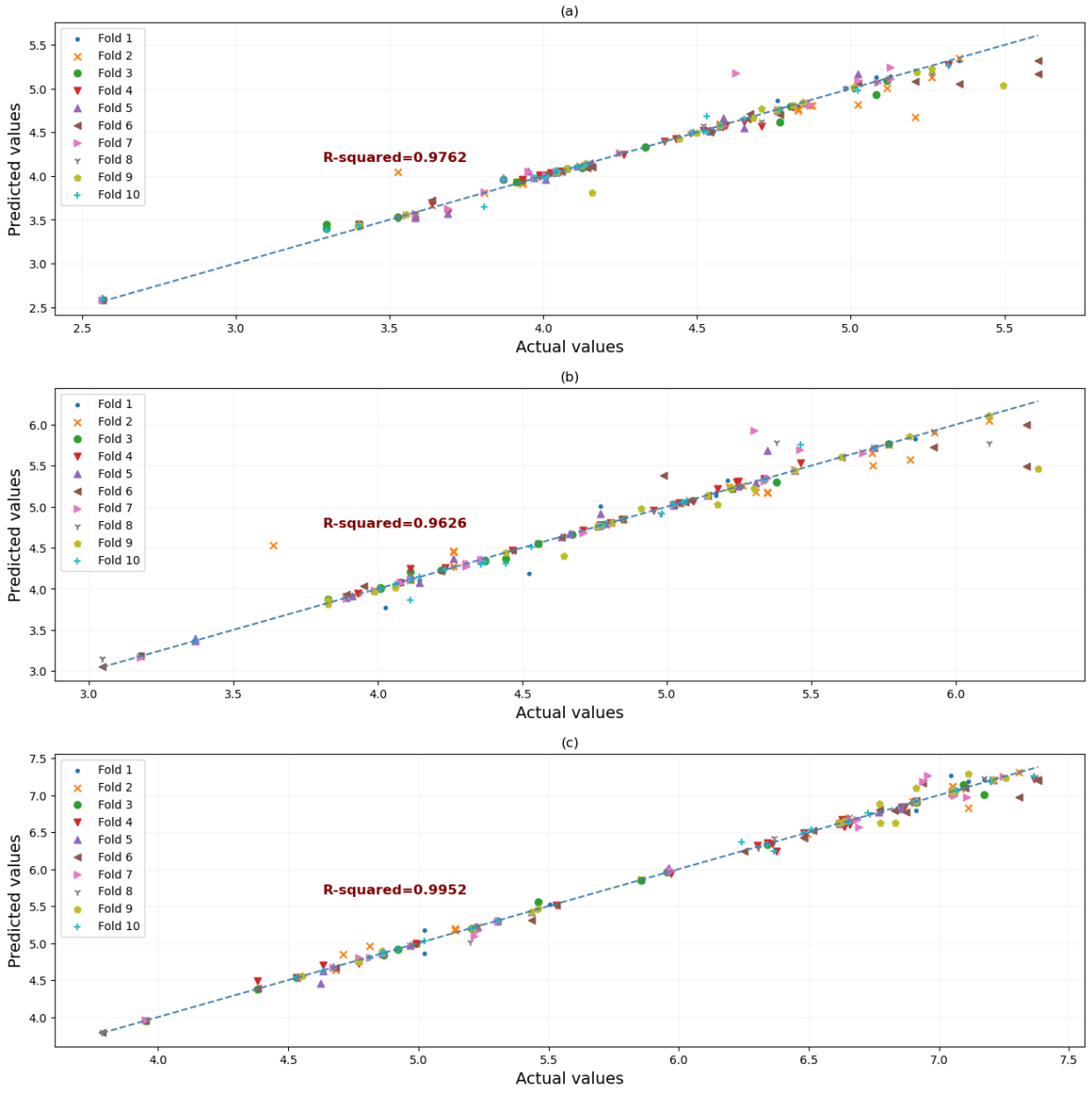


Figure 28. Actual vs. Predicted distances (10-fold cross validation) for VB: (a) Distance for 9% $CO_2$, (b) Distance for 4% $CO_2$, and (c) Distance for 1% $CO_2$.

### 2.4. Time to reach the steady state

All the CFD simulations were conducted under steady state conditions. Based on discussions with $CO_2$ pipeline operators, incidents involving $CO_2$ releases from pipelines typically result in discharges lasting approximately 20 to 30 minutes. This study aims to determine the time required to achieve the steady state.

To assess the time to reach steady state, a transient simulation with a time step of 0.1 seconds was performed. The case with the farthest dispersion was used and the corresponding parameters are enumerated in Table 7. As shown in the simulation results (Figure 29), concentrations of 9%, 4%, and 1% stabilized at approximately 80, 180, and 500 seconds, respectively, all of which are significantly shorter than 20 minutes. Therefore, the use of steady-state simulations is rational.

Table 7. Parameters applied for study.

| Variable | Pressure (MPa) | Diameter (inch) | Flow rate (MMcfd) | Wind speed (mph) | Temperature (°F) |
|---|---|---|---|---|---|
| Value | 10 | 30 | 1300 | 25 | 60 |

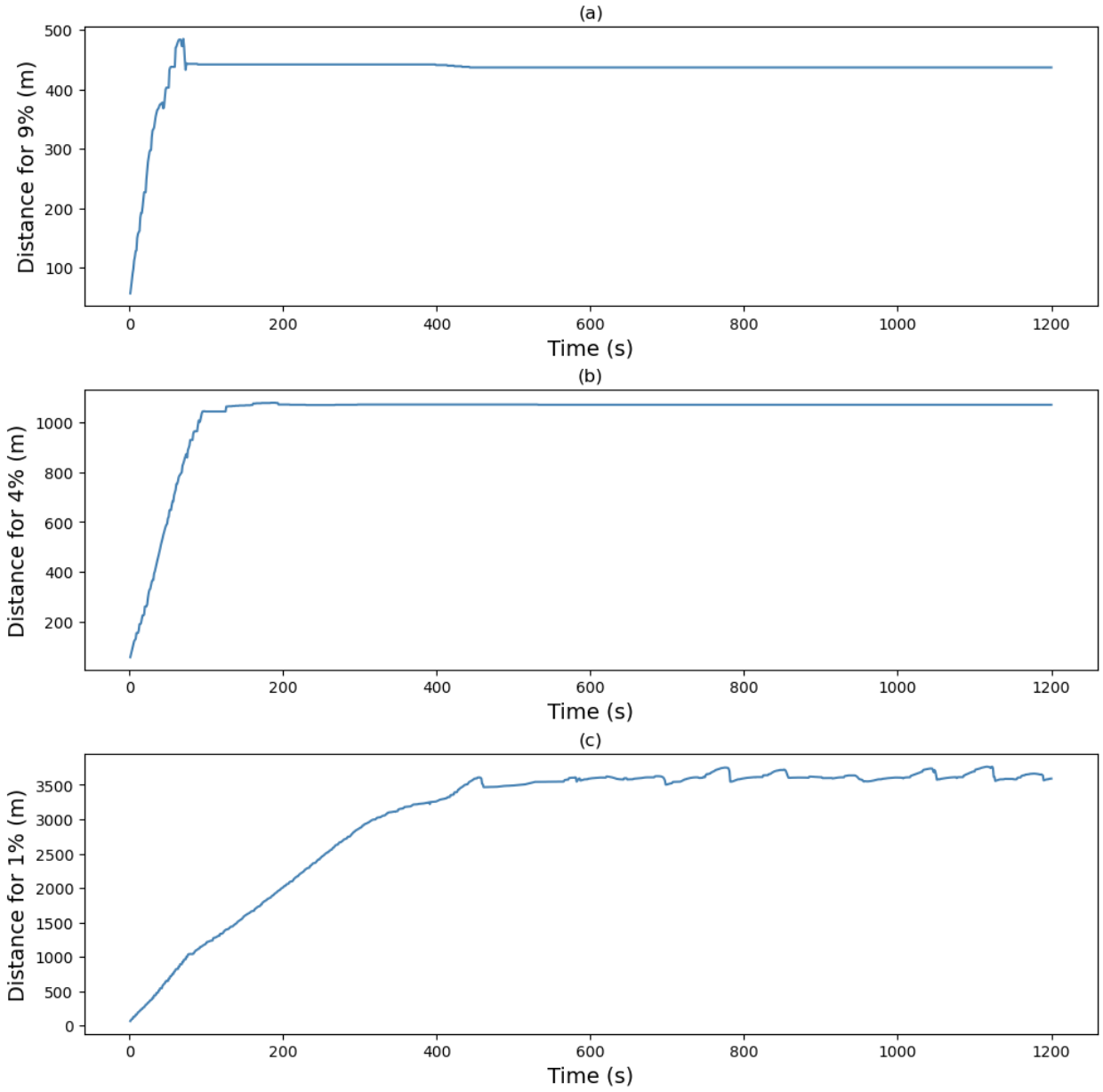


Figure 29. Distances of $CO_2$ concentration versus time: (a) 9%, (b) 4%, and (c) 1%.

## 3. Future Work

- Study the **evacuation time** for the surrounding public. Therefore, the emergency response plan can be organized accordingly to ensure the safety of the communities nearby.

- Conduct **near-field simulations** with the application of UDFs and UDRGMs in Ansys Fluent.

- Develop a **web-based tool** to determine the PIR for $CO_2$ pipelines.

# References


Birch, A. D., Hughes, D. J., & Swaffield, F. (1987). Velocity decay of high pressure jets. *Combustion Science and Technology*, *52*(1–3), 161–171. https://doi.org/10.1080/00102208708952575

Jiao, Z., Escobar-Hernandez, H. U., Parker, T., & Wang, Q. (2019). Review of recent developments of quantitative structure-property relationship models on fire and explosion-related properties. In *Process Safety and Environmental Protection* (Vol. 129, pp. 280–290). Institution of Chemical Engineers. https://doi.org/10.1016/j.psep.2019.06.027

Jiao, Z., Sun, Y., Hong, Y., Parker, T., Hu, P., Mannan, M. S., & Wang, Q. (2020). Development of flammable dispersion quantitative property-consequence relationship models using extreme gradient boosting. *Industrial and Engineering Chemistry Research*, *59*(33), 15109–15118. https://doi.org/10.1021/acs.iecr.0c02822

Sun, Y., Wang, J., Zhu, W., Yuan, S., Hong, Y., Mannan, M. S., & Wilhite, B. (2020). Development of Consequent Models for Three Categories of Fire through Artificial Neural Networks. *Industrial and Engineering Chemistry Research*, *59*(1), 464–474. https://doi.org/10.1021/acs.iecr.9b05032