

Supporting Information – Hyperparameters Metrics

A consistent set of thermophysical properties of methane curated with machine learning

Matheus Máximo-Canadas^a, Rubens Caio Souza^b, Julio Cesar Duarte^c, Jakler Nichele^b, Leonardo Santos de Brito Alves^d, Luiz Octavio Vieira Pereira^e, Ligia Gaigher Franco^f, Itamar Borges Jr^{a,*}

^a*Departamento de Química, Instituto Militar de Engenharia (IME), Praça General Tibúrcio, 80, Rio de Janeiro, RJ 22290-270, Brasil*

^b*Departamento de Engenharia de Defesa, Instituto Militar de Engenharia (IME), Praça General Tibúrcio, 80, Rio de Janeiro, RJ 22290-270, Brasil*

^c*Departamento de Computação, Instituto Militar de Engenharia (IME), Praça General Tibúrcio, 80, Rio de Janeiro, RJ 22290-270, Brasil*

^d*Laboratório de Dinâmica dos Fluidos Computacional, Departamento de Engenharia Mecânica, Universidade Federal Fluminense (UFF), Rua Passo da Pátria 156, Bloco E, Sala 216, São Domingos, Niterói, RJ 24210-240, Brasil*

^e*Centro de Pesquisas, Desenvolvimento e Inovação Leopoldo Américo Miguez de Mello - CENPES/PETROBRAS Rua Avenida Horácio Macedo, 950 Cidade Universitária Rio de Janeiro*

^f*Universidade Federal do Espírito Santo (UFES), Centro Tecnológico, Departamento de Engenharia Mecânica. Av. Fernando Ferrari, 514, Goiabeiras, 29075-910 - Vitória, ES – Brasil*

*Email: itamar@ime.eb.br

Summary

Introduction	2
S1 Isobaric heat capacity Cp	2
S2 Isochoric heat capacity Cv	7
S3 Joule-Thomson coefficient	12
S4 Sound Speed	12
S5 Density	16
S6 Volume	20
S7 Enthalpy	26
S8 Viscosity	32

Introduction

This document presents the final hyperparameter configurations obtained on *GridsearchCV* for each Machine Learning (ML) model and thermodynamic phase (liquid, vapor, and supercritical), under two different cross-validation strategies: 10-*fold* and Leave-One-Out (LOO) cross-validation.

S1 Isobaric heat capacity Cp

===== AdaboostLasso - CV = 10-*FOLD* =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostLasso - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostRidge - CV = 10-*FOLD* =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

===== DecisionTree - CV = 10-FOLD =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

Vapor state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5}.

Supercritical state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 100}.

Vapor state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 200}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 200}.

===== KNeighbors - CV = 10-FOLD =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 5, 'p': 1, 'weights': 'distance'}.

Vapor state: {'algorithm': 'brute', 'leaf_size': 20, 'n_neighbors': 9, 'p': 2, 'weights': 'distance'}.

Supercritical state: {'algorithm': 'brute', 'leaf_size': 20, 'n_neighbors': 9, 'p': 1, 'weights': 'distance'}.

===== KNeighbors - CV = LOO =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 7, 'p': 1, 'weights': 'distance'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 9, 'p': 1, 'weights': 'distance'}.

Supercritical state: {'algorithm': 'brute', 'leaf_size': 20, 'n_neighbors': 7, 'p': 1, 'weights': 'distance'}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 128, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.1, 'max_iter': 300, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 32, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate_init': 0.1, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

===== RandomForest - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}.

===== RandomForest - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 300}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 50, 'subsample': 0.8}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

S2 Isochoric heat capacity Cv

===== AdaboostLasso - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostLasso - CV = LOO =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostRidge - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 200}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 200}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 100}.

===== HistGradientBoosting =====

Liquid state for HistGradientBoosting: {'l2_regularization': 0.5, 'learning_rate': 0.05, 'max_bins': 255, 'max_depth': 3, 'max_iter': 100, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

Vapor state for HistGradientBoosting: {'l2_regularization': 0, 'learning_rate': 0.01, 'max_bins': 255, 'max_depth': 3, 'max_iter': 100, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

Supercritical state for HistGradientBoosting: {'l2_regularization': 0.5, 'learning_rate': 0.01, 'max_bins': 255, 'max_depth': 5, 'max_iter': 300, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

===== KNeighbors =====

Liquid state: {'algorithm': 'brute', 'leaf_size': 20, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}.

Supercritical state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 3, 'p': 1, 'weights': 'distance'}.

===== LGBM =====

Liquid state: {'colsample_bytree': 0.8, 'learning_rate': 0.05, 'max_depth': 3, 'min_child_samples': 20, 'n_estimators': 100, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

Vapor state for Adaboost with Decision Ridge: {'colsample_bytree': 0.8, 'learning_rate': 0.05, 'max_depth': 3, 'min_child_samples': 20, 'n_estimators': 50, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.8, 'learning_rate': 0.05, 'max_depth': 3, 'min_child_samples': 20, 'n_estimators': 50, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 50, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 300, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.1, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 100, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 200, 'solver': 'adam'}.

===== RandomForest =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 300}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 300}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 50, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 50, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

S3 Joule-Thomson coefficient

S4 Sound Speed

===== AdaboostLasso - CV = 10-FOLD =====

Vapor state: {'learning_rate': 1.0, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== AdaboostLasso - CV = LOO =====

Vapor state: {'learning_rate': 1.0, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'exponential', 'n_estimators': 200}.

===== AdaboostRidge - CV = 10-FOLD =====

Vapor state: {'learning_rate': 1.0, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Vapor state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = 10-FOLD =====

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = LOO =====

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 100}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

===== BaggingRegressorRidge - CV = LOO =====

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 100}.

===== DecisionTree - CV = 10-FOLD =====

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = LOO =====

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== GradientBoosting - CV = 10-FOLD =====

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== GradientBoosting - CV = LOO =====

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

===== NeuralNetwork - CV = 10-FOLD =====

Vapor state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 32, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 50, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 100, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate_init': 0.001, 'max_iter': 100, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate_init': 0.001, 'max_iter': 100, 'solver': 'adam'}.

===== XGB - CV = 10-FOLD =====

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

===== XGB - CV = LOO =====

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

S5 Density

===== BaggingRegressorLasso - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100}.

===== DecisionTree - CV = 10-FOLD =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2}.

Vapor state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

===== HistGradientBoosting =====

Liquid state for HistGradientBoosting: {'l2_regularization': 0, 'learning_rate': 0.05, 'max_bins': 255, 'max_depth': 3, 'max_iter': 100, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

Vapor state for HistGradientBoosting: {'l2_regularization': 0, 'learning_rate': 0.05, 'max_bins': 255, 'max_depth': 3, 'max_iter': 300, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

Supercritical state for HistGradientBoosting: {'l2_regularization': 0.5, 'learning_rate': 0.05, 'max_bins': 255, 'max_depth': 7, 'max_iter': 300, 'max_leaf_nodes': 31, 'min_samples_leaf': 20}.

===== KNeighbors - CV = 10-FOLD =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 9, 'p': 2, 'weights': 'uniform'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}.

Supercritical state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}.

===== KNeighbors - CV = LOO =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 9, 'p': 2, 'weights': 'uniform'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}.

Supercritical state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 128, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 32, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 128, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'beta_1': 0.9, 'beta_2': 0.999, 'early_stopping': False, 'epsilon': 1e-08, 'hidden_layer_sizes': (512, 256, 128, 64), 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'max_fun': 15000, 'max_iter': 200, 'momentum': 0.9, 'n_iter_no_change': 10, 'nesterovs_momentum': True, 'power_t': 0.5, 'shuffle': True, 'solver': 'adam', 'tol': 0.0001, 'validation_fraction': 0.1, 'verbose': False, 'warm_start': False}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.8}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

S6 Volume

===== AdaboostLasso - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 100}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostLasso - CV = LOO =====

Liquid state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostRidge - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 200}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 100}.

===== BaggingRegressorLasso - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 100}.

===== BaggingRegressorRidge - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 50}.

===== DecisionTree - CV = 10-FOLD =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10}.

Vapor state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Liquid state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10}.

Vapor state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}.

Vapor state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': True, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.1, 'loss': 'quantile', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'learning_rate': 1.0, 'loss': 'quantile', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 0.1, 'loss': 'quantile', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 200}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 100, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 64, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 32, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 200, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

===== NuSVR_ - CV = 10-FOLD =====

Liquid state for NuSVR: {'C': 10, 'degree': 4, 'gamma': 'scale', 'kernel': 'poly', 'nu': 0.5}.

Vapor state for NuSVR: {'C': 10, 'degree': 4, 'gamma': 'scale', 'kernel': 'poly', 'nu': 0.5}.

Supercritical state for NuSVR: {'C': 10, 'degree': 3, 'gamma': 'scale', 'kernel': 'poly', 'nu': 0.5}.

===== NuSVR_ - CV = LOO =====

Liquid state for NuSVR: {'C': 1.0, 'cache_size': 200, 'coef0': 0.0, 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'nu': 0.5, 'shrinking': True, 'tol': 0.001, 'verbose': False}

Vapor state for NuSVR: {'C': 1.0, 'cache_size': 200, 'coef0': 0.0, 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'nu': 0.5, 'shrinking': True, 'tol': 0.001, 'verbose': False}

Supercritical state for NuSVR: {'C': 1.0, 'cache_size': 200, 'coef0': 0.0, 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'nu': 0.5, 'shrinking': True, 'tol': 0.001, 'verbose': False}

===== RandomForest - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}.

===== RandomForest - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.01, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

S7 Enthalpy

===== AdaboostLasso - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 200}.

===== AdaboostLasso - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostRidge - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 1.0, 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 1.0, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 50}.

===== DecisionTree - CV = 10-FOLD =====

Liquid state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Liquid state: {'criterion': 'squared_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 100}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 100}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 200}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 50}.

===== KNeighbors - CV = 10-FOLD =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 3, 'p': 1, 'weights': 'distance'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 20, 'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}.

Supercritical state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}.

===== KNeighbors - CV = LOO =====

Liquid state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 3, 'p': 1, 'weights': 'distance'}.

Vapor state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}.

Supercritical state: {'algorithm': 'auto', 'leaf_size': 40, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 32, 'hidden_layer_sizes': (512, 512, 512), 'learning_rate_init': 0.01, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 200, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 32, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 200, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 200, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 50, 'solver': 'adam'}.

===== RandomForest - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 300}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== RandomForest - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 300}.

Supercritical state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 5, 'n_estimators': 100, 'subsample': 0.6}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 5, 'n_estimators': 50, 'subsample': 0.8}.

S8 Viscosity

===== AdaboostLasso - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 100}.

Vapor state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostLasso - CV = LOO =====

Liquid state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'linear', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== AdaboostRidge - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 100}.

Vapor state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

===== AdaboostRidge - CV = LOO =====

Liquid state: {'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50}.

Vapor state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

Supercritical state: {'learning_rate': 1.0, 'loss': 'square', 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorLasso - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_features': 1.0, 'max_samples': 0.7, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== BaggingRegressorRidge - CV = LOO =====

Liquid state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}.

===== DecisionTree - CV = 10-FOLD =====

Liquid state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}.

===== DecisionTree - CV = LOO =====

Liquid state: {'criterion': 'squared_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

Vapor state: {'criterion': 'absolute_error', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5}.

Supercritical state: {'criterion': 'squared_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2}.

===== ExtraTreesRegressor - CV = 10-FOLD =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

===== ExtraTreesRegressor - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'bootstrap': False, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

===== GradientBoosting - CV = 10-FOLD =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 200}.

===== GradientBoosting - CV = LOO =====

Liquid state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 200}.

Vapor state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 50}.

Supercritical state: {'learning_rate': 0.1, 'loss': 'huber', 'max_depth': 4, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}.

===== LGBM =====

Liquid state: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_samples': 20, 'n_estimators': 100, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

Vapor state for Adaboost with Decision Ridge: {'colsample_bytree': 0.8, 'learning_rate': 0.05, 'max_depth': 3, 'min_child_samples': 20, 'n_estimators': 50, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 5, 'min_child_samples': 20, 'n_estimators': 100, 'num_leaves': 31, 'reg_alpha': 0.1, 'reg_lambda': 0.1, 'subsample': 0.8}.

===== NeuralNetwork - CV = 10-FOLD =====

Liquid state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 128, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 300, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 300, 'solver': 'adam'}.

===== NeuralNetwork - CV = LOO =====

Liquid state: {'activation': 'relu', 'alpha': 0.01, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.001, 'max_iter': 300, 'solver': 'adam'}.

Vapor state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 64, 'hidden_layer_sizes': (512, 256, 128), 'learning_rate_init': 0.1, 'max_iter': 50, 'solver': 'adam'}.

Supercritical state: {'activation': 'relu', 'alpha': 0.0001, 'batch_size': 128, 'hidden_layer_sizes': (1024, 512, 256, 128), 'learning_rate_init': 0.01, 'max_iter': 200, 'solver': 'adam'}.

===== RandomForest - CV = 10-FOLD =====

Liquid state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== RandomForest - CV = LOO =====

Liquid state: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

Vapor state: {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}.

Supercritical state: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}.

===== XGB - CV = 10-FOLD =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.8}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 50, 'subsample': 0.8}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

===== XGB - CV = LOO =====

Liquid state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

Vapor state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 3, 'n_estimators': 50, 'subsample': 0.6}.

Supercritical state: {'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 3, 'n_estimators': 100, 'subsample': 0.6}.

