Estimation of solid bitumen content in hydrocarbon reservoirs: fusion of individual machine learning models and petrophysical well-logging data in a committee machine

By:
Ali Gholami Vijouyeh[a], Maha Raoof Hamoudi[b], Dyana Aziz Bayz[c], Ali Kadkhodaie[a]

[a] Earth Sciences Department, Faculty of Natural Science, University of Tabriz, Tabriz, Iran

[b] Department of Natural Resources Engineering and Management, University of Kurdistan Hewler, Erbil, Iraq

c Department of Petroleum Engineering, College of Engineering, Knowledge University, Erbil, Iraq

Content

01 Main

*** correlation_plot.m**

Calculates and shows the correlation between different fields of data using Matlab R2022b.

*** create_data.m**

Creates training and test data from selected features of main data in Matlab R2022b. The results are saved in: ".\01 Data\test_data\\" folder.

*** fuzzy_logic.m**

Implements TS-FIS algorithm in Matlab R2022b. The results are saved in:
".\03 Output files\ 03 Stand-alone algorithms\Fuzzy Logic\\" folder.

*** lightgbm_bitumen.m**

Loads and calls the results of LightGBM algorithm modelling in Matlab R2022b, implemented in Python 3.11. The results are saved in:
".\03 Output files\ 03 Stand-alone algorithms\LightGBM\\" folder.

*** lightgbm_bitumen.py**

Implements LightGBM algorithm in Python 3.11. The results are saved in:
".\03 Output files\ 03 Stand-alone algorithms\LightGBM\\" folder.

*** neural_network.m**

Implements Neural Network algorithm in Matlab R2022b. The script reads the proper network from ".\03 Output files \03 Stand-alone algorithms\Neural Network\Param_OPT \608_net.mat\". The results are saved in:
".\03 Output files \03 Stand-alone algorithms\Neural Network " folder.

*** neuro_fuzzy.m**

Implements Neuro-Fuzzy algorithm in Matlab R2022b. It reads the network from ".\03 Output files \03 Stand-alone algorithms\Neuro Fuzzy\Net \NF_bitumen.mat\". The results are saved in:
".\03 Output files \03 Stand-alone algorithms\Neuro Fuzzy\" folder.

*** optimization.m**

This file implements all optimization committee machine algorithms in Matlab R2022b, which are "GA", "SA", and "ACO_R". The results are saved in:
".\03 Output files\04 Optimization by committee machines\\" folder.

*** RBF.m**

Implements Radial Basis Function algorithm in Matlab R2022b. The results are stored in:
".\03 Output files\ 03 Stand-alone algorithms\RBF\\" folder.

*** xgboost.m**

Loads and calls the results of XGBoost algorithm modelling in Matlab R2022b, implemented in Python 3.11. The results are stored in:
".\03 Output files\ 03 Stand-alone algorithms\XGBoost\\" folder.

*** XGBoost.py**

Implements XGBoost algorithm in Python 3.11. The outcomes are stored in:
".\03 Output files\ 03 Stand-alone algorithms\XGBoost\\" folder.

02 Parameter Tuning

*** lightgbm_bitumen_p.py**

This file has been used to parameter tuning of LightGBM algorithm using Python 3.11. The results are stored in:
".\03 Output files\03 Stand-alone algorithms\LightGBM\Param_OPT\\" folder.

*** neural_network_parameters.m**

This file has been used to parameter tuning for back-propagation neural network algorithm using Matlab R2022b. The results are stored in:

".\03 Output files\03 Stand-alone algorithms\Neural Network\Param_OPT \\" folder.

During this operation, the most optimal model was concluded to be: 608_net.mat (".\03 Output files\03 Stand-alone algorithms\Neural Network\Param_OPT \608_net.mat \").

*** optimization_01_ga.m**

This file has been used to parameter tuning for GA optimization algorithm in Matlab R2022b. The results are stored in:

".\03 Output files\04 Optimization by committee machines\GA\\" folder.

*** optimization_02_sa.m**

This file has been used to parameter tuning for SA optimization algorithm in Matlab R2022b. The results are stored in:

".\03 Output files\04 Optimization by committee machines\SA\\" folder.

*** optimization_03_aco.m**

This script tunes parameter for ACO_R optimization algorithm in Matlab R2022b. The results are stored in:

".\03 Output files\04 Optimization by committee machines\ACO\\" folder.

*** RBF_param.m**

This file has been used to parameter tuning for Radial Basis Function algorithm using Matlab R2022b. The results are stored in:

".\03 Output files\03 Stand-alone algorithms\RBF\Param_OPT\\" folder.

*** XGBoost_p.py**

This file has been used to parameter tuning of XGBoost algorithm using Python 3.11. The results are stored in:

".\03 Output files\03 Stand-alone algorithms\XGBoost\Param_OPT \\" folder.

03 DT and CGR prediction

CGR prediction

*** create_data.m**

Creates training and test data from selected features of main data in Matlab R2022b to predict CGR well-logging data.

*** neural_network.m**

Implements Neural Network algorithm in Matlab R2022b to estimate CGR well-logging data. The script reads the proper network from ".\03 Output files\02 CGR and DT prediction\CGR Results\Neural Network \Param_OPT \353_net.mat\". The results are saved in:
".\03 Output files\02 CGR and DT prediction\CGR Results\Neural Network\" folder.

*** neural_network_parameters.m**

This file has been used to parameter tuning for back-propagation neural network algorithm using Matlab R2022b to estimate CGR values. The outcomes are stored in:
".\03 Output files\02 CGR and DT prediction\CGR Results\Neural Network \Param_OPT \\" folder.
In this operation, the most optimal model was concluded to be: 353_net.mat (".\03 Output files\02 CGR and DT prediction\CGR Results\Neural Network\Param_OPT \353_net.mat\").

*** script_plot_3d.m**
This script was used to figure the models obtained from parameter tuning of CGR prediction (in Matlab R2022b). The outcomes are stored in:

".\03 Output files\02 CGR and DT prediction\CGR Results\Neural Network\Param_OPT\ann_mse_3d_plot.png & ann_r_3d_plot.png\".

*** tst_prediction.m**

This script estimates and draws the CGR values using the well-logging input data from the model obtained by BP-NN in Matlab R2022b. The outcomes of each well are stored in:

".\03 Output files\02 CGR and DT prediction\CGR Results\Prediction of well b (CGR)\ ".

".\03 Output files\02 CGR and DT prediction\CGR Results\Prediction of well d (CGR)\ ".

DT prediction

*** create_data.m**

Creates training and test data from selected features of main data in Matlab R2022b to predict DT well-logging data.

*** neural_network.m**

Implements Neural Network algorithm in Matlab R2022b to estimate DT well-logging data. The script reads the proper network from ".\03 Output files\02 CGR and DT prediction\ DT Results\Neural Network \Param_OPT \217_net.mat\". The results are saved in:
".\03 Output files\02 CGR and DT prediction\ DT Results\Neural Network\" folder.

*** neural_network_parameters.m**

This file has been used to parameter tuning for back-propagation neural network algorithm using Matlab R2022b to estimate DT values. The outcomes are stored in:
".\03 Output files\02 CGR and DT prediction\DT Results\Neural Network \Param_OPT \\" folder.
In this operation, the most optimal model was concluded to be: 217_net.mat (".\03 Output files\02 CGR and DT prediction\DT Results\Neural Network\Param_OPT \217_net.mat\").

*** script_plot_3d.m**

This script was used to figure the models obtained from parameter tuning of DT prediction (in Matlab R2022b). The outcomes are stored in:

".\03 Output files\02 CGR and DT prediction\DT Results\Neural Network\Param_OPT\ann_mse_3d_plot.png & ann_r_3d_plot.png\".

*** tst_prediction.m**

This script estimates and draws the DT values using the well-logging input data from the model obtained by BP-NN in Matlab R2022b. The outcomes of each well are stored in:

".\03 Output files\02 CGR and DT prediction\DT Results\Prediction of Ahwaz_307 (DT) \ ".

".\03 Output files\02 CGR and DT prediction\ DT Results\Prediction of SD_3 (DT) \ ".

---

04 Multi-variable linear regression

*** three_regression.m**

This file implements multi-variable linear regression method in Matlab R2022b. The results are saved in:

".\03 Output files\05 Multi-variable linear regression\".

---

05 predictions of Bitumen in wells C, D, F

These scripts are designed for prediction of bitumen in other wells using the models obtained from AI systems in this study. To run it:

In first step, run **tst_create_data.m script,** which loads, creates and divides the well-logging input data.

In second step, run **LightGBM_prediction.py,** which loads the model constructed with LightGBM algorithm from Python 3.11 to Matlab R2022b. The results are stored in:

".\03 Output files\03 Stand-alone algorithms\LightGBM\lgbm_y_pred_test.dat\".

In third step, run **XGBoost_prediction.py,** which loads the model constructed with XGBoost algorithm from Python 3.11 to Matlab R2022b. The results are stored in:

".\03 Output files\03 Stand-alone algorithms\XGBoost\xgb_y_pred_test.dat\".

In final step, run **tst_prediction.m,** which load, model and predict the targets using MVLR (as the best algorithm resulted in this study). The results are saved in:
".\03 Output files\07 all wells predictions\".