

A comparative study of methods for estimating hydrocarbon-water interfacial tension

TRAN Nguyen Thien Tam^{1,*}, HOANG Trong Quang¹

¹ Faculty of Geology & Petroleum Engineering, Ho Chi Minh City University of Technology - Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

* Corresponding email: trantam2512@hcmut.edu.vn

Abstract: *Interfacial tension is the force that acts at the boundary between two immiscible phases. This force keeps the two fluids separated, preventing them from mixing together. Hydrocarbon-water interfacial tension is one of the most critical parameters in petroleum engineering calculations, like enhanced oil recovery, phase behavior in reservoirs, separation processes, and so on. Hydrocarbon-water interfacial tension is often measured in laboratories, but it is costly and time-demanding. Consequently, mathematical methods are developed for estimating hydrocarbon-water interfacial tension, which summary in two main groups are empirical correlations and machine learning algorithms. In this paper, the authors carry out a comparative study of mathematical methods for estimating hydrocarbon-water interfacial tension. Empirical correlations are implemented based on the study by Danesh (1998), Sutton (2006), Sutton (2009) and Meybodi et al. (2016). Machine learning algorithms used include Multi-Layer Perceptron (MLP) and Extra Trees (ET). Research data gathered from open literature incorporates features: critical temperature (T_c), reservoir temperature (T_R), density difference ($\Delta\rho$), reservoir pressure (P), and interfacial tension (IFT). The estimation outcomes are contrasted using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2) to find the optimal model. The results show that the Extra Trees (ET) algorithm is the most accurate model with the lowest MAE and RMSE (0.6391 and 0.9565) and the highest R^2 (0.9871) in the group of empirical correlations and machine learning algorithms. Moreover, the solution indicates two machine learning models having better performance than four empirical correlations due to lower MAE and RMSE and the higher R^2 .*

Keywords: *interfacial tension; critical temperature; reservoir temperature; reservoir pressure; machine learning.*

1. Introduction

Interfacial tension (IFT) is the force that acts at the boundary between two immiscible phases [1]. This force keeps the two fluids separated, preventing them from mixing together [2]. Hydrocarbon-water interfacial tension is one of the most critical parameters in petroleum engineering calculations, like enhanced oil recovery, phase behavior in reservoirs, separation processes, and so on [3], [4].

The most common way to determine the hydrocarbon-water IFT is through laboratory experiments. There are four main group for measurement of interfacial tension are: direct measurement using a microbalance, measurement of capillary pressure, analysis of the balance between capillary and gravity forces, analysis of gravity-distorted drops [5]. However, laboratory experiments are costly and time-demanding [6]. Consequently, mathematical methods are developed for estimating hydrocarbon-water interfacial tension, which summary in two main groups are empirical correlations and machine learning algorithms. The first common empirical correlation for interfacial tension between hydrocarbon and water developed by Danesh (1998) [7]. Later in 2006, Sutton improved the Danesh correlation [8], and in 2009, Sutton modified the 2006 version for two cases: hydrocarbons liquid-water and gas-water [9]. Meybodi et al. (2016) developed a new unique correlation to estimate IFT for both liquid and gaseous hydrocarbons [6].

Despite there being some benefits to estimating IFT, empirical correlation methods have limitations. They frequently not high accuracy and only valid for calculation within the data range of the built correlation model. With the remarkable development of data science in the era of artificial intelligence, there have been many studies applying machine learning algorithms for advancing the estimation of IFT. The machine learning algorithms used are Artificial Neural Network (ANN) [10], Least Square Support Vector Machine (LSSVM) [4], [11], group of methods involve Support Vector Regression (SVR), Decision Tree (DT), Random Forests (RF), Gradient Boosting (GB), Catboosting (CB), and XGBoosting (XGB)

[12], or hybrid methods comprise Radial Basis Function networks optimized by Genetic Algorithm (GA-RBF), Least Square Support Vector Machine optimized by Coupled Simulated Annealing (CSA-LSSVM), and Adaptive Neuro-Fuzzy Inference System optimized by combination of Particle Swarm Optimization (PSO) and hybrid methods (CHPSO-ANFIS) [13].

In this work, the authors execute a comparative study of mathematical methods for estimating hydrocarbon-water interfacial tension based on both empirical correlations and machine learning algorithms. Empirical correlations are used consist of Danesh, Sutton (2006), Sutton (2009), Meybodi et al., while machine learning algorithms involve Multilayer Perceptron (MLP) and Extra Trees (ET). Research data collected from open literature includes parameters: critical temperature (T_C), reservoir temperature (T_R), density difference ($\Delta\rho$), reservoir pressure (P), and interfacial tension (IFT). The estimation results are compared using Mean Absolute Error (MAE), Root Mean Square Error ($RMSE$), and Coefficient of Determination (R^2) to find the optimal model.

2. Methodology

2.1 Empirical correlations

2.1.1 Danesh (1998)

Danesh (1998) [7] developed the first empirical correlation for interfacial tension between hydrocarbon and water as below:

$$IFT = 111(\rho_w - \rho_h)^{1.024} \left(\frac{T_R}{T_C}\right)^{-1.25} \quad (1)$$

where IFT is interfacial tension (mN/m), ρ_w is water density (g/cm³), ρ_h is hydrocarbon density (g/cm³), T_R is reservoir temperature (K), T_C is critical temperature (K)

Define density difference $\Delta\rho = \rho_w - \rho_h$, and $T_r = T_R/T_C$, Eq. (1) becomes

$$IFT = 111(\Delta\rho)^{1.024} (T_r)^{-1.25} \quad (2)$$

2.1.2 Sutton (2006)

Sutton (2006) [8] modified Danesh empirical correlation as follows:

$$IFT = \left(\frac{1.58(\Delta\rho) + 1.76}{(T_r)^{0.3125}}\right)^4 \quad (3)$$

2.1.3 Sutton (2009)

Sutton (2009) [9] advanced Eq. (3) by proposing new empirical correlations for hydrocarbon-water interfacial tension divided into two cases.

For case liquid-water, the IFT empirical correlation is:

$$IFT = \left(\frac{1.53988(\Delta\rho) + 2.08339}{(T_r)^{0.821976 - 0.00183785T_R + 0.00000134016T_R^2}}\right)^{3.667} \quad (4)$$

For case gas-water, which $\Delta\rho > 0.56$, the empirical correlation is:

$$IFT = \left(\frac{1.53988(\Delta\rho) + 2.08339}{(T_C/302.881)^{0.821976 - 0.00183785T_R + 0.00000134016T_R^2}}\right)^{3.667} \quad (5)$$

2.1.4 Meybodi et al. (2016)

Meybodi et al. (2016) [6] developed a IFT empirical correlation for both liquid and gaseous hydrocarbons with formulation:

$$IFT = \left(\frac{a_1 + a_2(\Delta\rho) + a_3(\Delta\rho)^2 + a_4(\Delta\rho)^3}{a_5 + a_6 \frac{(T_R)^{a_7}}{T_C} + a_8 (T_R)^{a_9}}\right)^{a_{10}} \quad (6)$$

where $a_1 = -1.3687340042 \times 10^{-1}$, $a_2 = -3.0391828884 \times 10^{-1}$, $a_3 = 5.6225871072 \times 10^{-1}$, $a_4 = -3.3074367079 \times 10^{-1}$, $a_5 = -3.0050179309 \times 10^0$, $a_6 = 5.8914210205 \times 10^{-5}$, $a_7 = -4.1388901263 \times 10^0$, $a_8 = 3.0084299030 \times 10^0$, $a_9 = -3.8203072876 \times 10^{-3}$, $a_{10} = 3.5000000000 \times 10^0$

2.2 Machine learning algorithms

2.2.1 MLP

Multilayer Perceptron (MLP) is one of the foundational architectures in artificial neural networks (ANN) [14]. The MLP basic structure involve 3 layers: input layer, hidden layers, and an output layer. The basic unit of a MLP is the neuron, or perceptron, and neurons are connected by weights [15]. In association with weights, biases contribute flexibility to neural networks [16]. MLP works by updating weights and biases using two continuous processes: forward propagation and back propagation [17].

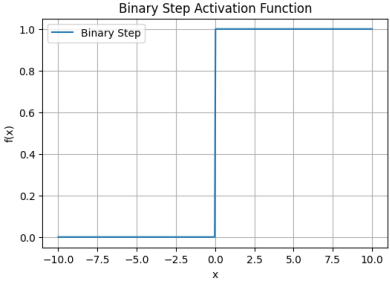
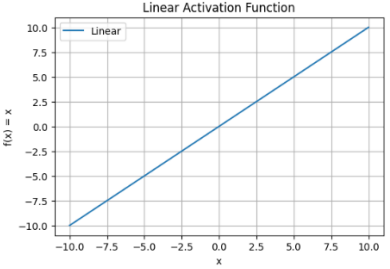
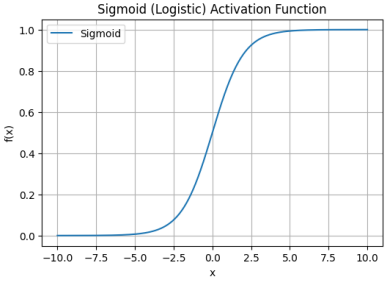
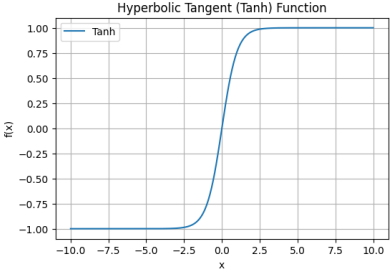
In forward propagation, for a given neuron, output y is a activation of transfer function. Transfer function is the weighted sum of weight multiplier for input value plus a bias. The mathematical form of the output y is as follows

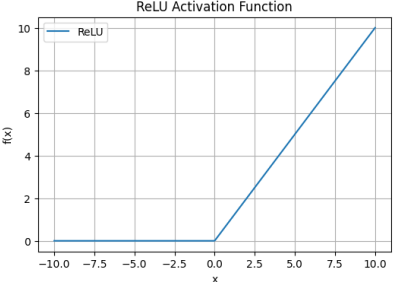
$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \tag{7}$$

where w_i denotes weights, x_i is input values, b indicates bias, and f symbolises transfer function.

In a neural network, activation functions are the way to transmission the information. Some popular activation functions are show in Table 1 [18].

Tab. 1. Popular activation functions

| Activation function | Formula | Graph (Draw by Python code) |
|------------------------------------|--|---|
| Binary step function | $f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$ |  |
| Linear activation function | $f(x) = x$ |  |
| Sigmoid (logistic) function | $f(x) = \frac{1}{1 + e^{-x}}$ |  |
| Hyperbolic tangent (tanh) function | $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ |  |

| Activation function | Formula | Graph (Draw by Python code) |
|------------------------------|---------------------|---|
| Rectified Linear Unit (ReLU) | $f(x) = \max(0, x)$ |  |

In back propagation, the ultimate goal is to minimize a loss function between the output of the forward propagation process and the actual output [19]. A common loss function for the regression problem is Mean Squared Error (MSE), and the classification problem is Cross-Entropy [20].

Here is the formulation of loss function for regression problem:

$$L_R = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{8}$$

where L_R is loss function for regression problem, n refers to the number of data points, y_i is actual value, \hat{y}_i is predicted value.

The formulation of loss function for classification problem as follows:

$$L_C = - \sum_i y_i \log(\hat{y}_i) \tag{9}$$

where L_C is loss function for classification problem, y_i is the true label (one-hot encoded), that is, y_i equals 1 for the correct label and 0 for all other labels, \hat{y}_i is the predicted probability for the i^{th} label.

The loss function is minimized by calculating the gradient of the loss function for each weight in the neural network. After calculating the gradient, the network parameters (weights and biases) are adjusted using an optimization approach like Gradient Descent [21]. Gradient Descent updates the weights and biases by formulations:

$$w_n = w_c - \eta \frac{\partial L}{\partial w_c} \tag{10}$$

$$b_n = b_c - \eta \frac{\partial L}{\partial b_c} \tag{11}$$

where w_n, b_n are the weight and bias associated with the new layer, w_c, b_c are the weight associated with the current layer, η is the learning rate, which determines the step size for updates, and L is loss function.

2.2.2 ET

Extremely Randomized Trees, also known as Extra Trees, are a type of ensemble learning technique. In order to improve predictive performance, ensemble learning combines predictions from several models to create a meta-learning model. The essence of the ET algorithm is to generate a large number of decision trees from the training dataset [22].

A decision tree's objective is to iteratively divide the data into subsets while trying to minimize some impurity measure, such as the Mean Squared Error (MSE) for regression and the Gini index or Entropy for classification [23].

Mean Squared Error (MSE) is a regression metric to measure the quality of a prediction model. *MSE* calculates the average squared difference between the measured target values and the predicted target values as follows:

$$MSE = \frac{1}{n} \sum_{i \in S} \left(y_i - \frac{1}{n} \sum_{j \in S} y_j \right)^2 \tag{12}$$

where S is the subset of dataset at a particular node in the tree, n is the number of samples in the subset S , y_i is the measured target values of the i^{th} sample in S , j is an index that refers to one of all data points in the subset S .

The Gini Index (GI), also known as Gini Impurity is a classification metric used in decision tree algorithms to measure the impurity of a dataset. For a dataset with T classes, the Gini Index for a subset S is

$$GI = 1 - \sum_{k=1}^T p_k^2 \tag{13}$$

where T is the number of classes, p_k is the frequency of class k in the subset S .

Entropy (E) is a measure of the uncertainty in a dataset. For a dataset with T classes, the entropy for a subset S is

$$E = - \sum_{k=1}^T p_k \log_2(p_k) \tag{14}$$

where p_k is the probability of class k in the subset S , $\log_2(p_k)$ denotes the uncertainty associated with class k .

In Extra Trees, there are two primary ways that randomness are random feature selection, and random threshold selection. For random feature selection, a random subset of features is chosen at each split. The algorithm advances diversity among the trees, as different trees use different sets of features for splits by using a smaller subset of features. For random threshold selection, instead of finding the best split point, it randomly choosing the split threshold value for each feature [24].

The final step in the Extra Trees algorithm is ensemble aggregation. For classification, each tree outputs a predicted class label, and the final prediction is obtained through majority voting. For regression, each tree outputs a continuous value, and the final prediction is the average of the predictions [25].

2.3 Regression metrics

In order to find the best model in this work, the regression metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2) [26]. Below is their detailed formulation.

2.3.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average absolute difference between predicted and measured values.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{15}$$

where n represents the total number of data points, y_i is measured (or actual) value, and \hat{y}_i is predicted value.

2.3.2 Root Mean Square Error (RMSE)

Mean Squared Error (MSE) calculate the average squared difference between predicted and measured values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{16}$$

Root Mean Squared Error (RMSE) is square root of MSE

$$RMSE = \sqrt{MSE} \tag{17}$$

2.3.3 Coefficient of Determination (R^2)

The Coefficient of Determination, denoted as R^2 , is a measure the percentage of variance in the dependent variable that can be explained by the independent variables in a regression model.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \tag{18}$$

where \bar{y} is the mean of the measured values y_i .

3. Results and Discussions

3.1 Data

In this study, the research data is collected from literature published by Meybodi et al. [6]. The original data has 1105 data points, after processing the non-numerical data, we have 1075 data points. Table 1 shown the descriptive statistics of the processing dataset.

Tab. 2. The processing dataset descriptive statistics

| Metrics | T_C | T_R | $\Delta\rho$ | P | IFT |
|---------|----------|----------|--------------|----------|---------|
| Count | 1075 | 1075 | 1075 | 1075 | 1075 |
| Mean | 505.7926 | 326.3789 | 0.3645 | 25.0403 | 47.1212 |
| Min | 190.5639 | 252.4444 | 0.0939 | 47.4156 | 20.8400 |
| Max | 722.0000 | 550.0000 | 0.9968 | 300.0000 | 75.5000 |

Fig. 1 shows the correlation plot (heatmap) between the features in the processing dataset. The correlation between the features is calculated using the Pearson correlation coefficient. The Pearson correlation coefficient, often denoted as R , is a measure of the linear relationship between two variables, which range from -1 to 1.

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y}_i) (y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \tag{19}$$

where n represents the total number of data points, y_i is measured value, and \hat{y}_i is predicted value, \bar{y} is the mean of the measured values y_i , $\bar{\hat{y}}$ is the mean of the predicted values \hat{y}_i .

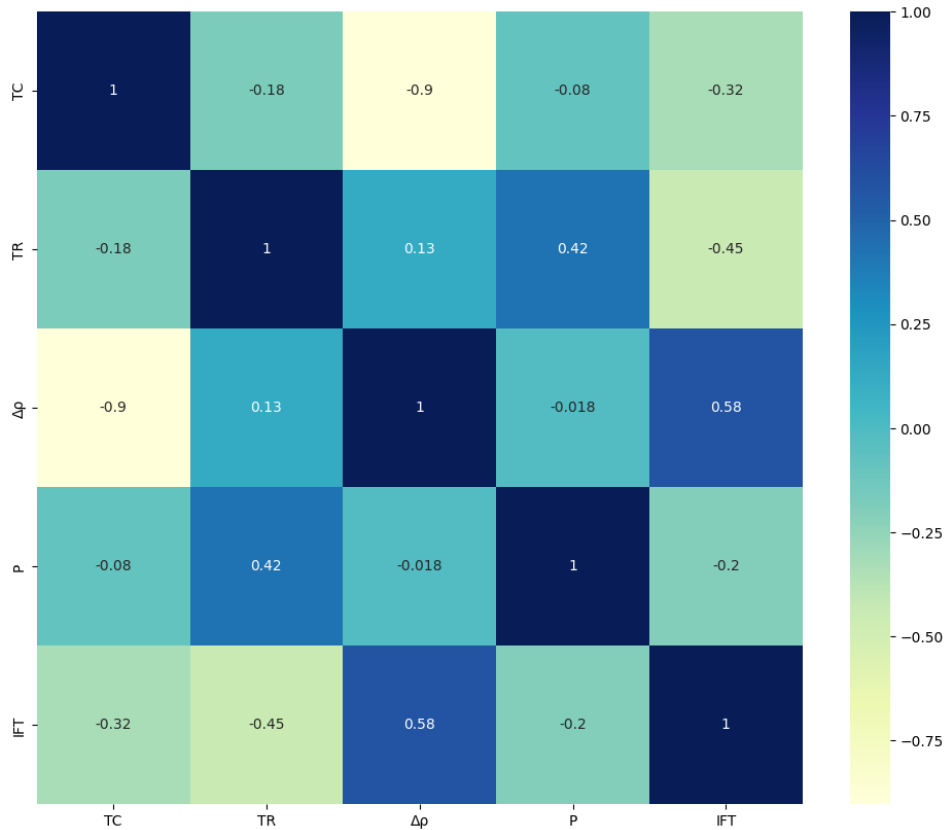


Fig. 1. The processing dataset heatmap correlation plot

Fig. 1 with the coefficients specifies that the input features all have an effect on the target feature IFT. In addition, the number of features is small (5 features, including target variable IFT). Therefore, to avoid missing information, all the features will be used to train the models.

The processing data is divided into two training and testing data sets with the ratio of 80:20 for building all two machine learning models: MLP and ET.

3.2 Results

3.2.1 Empirical correlations

a) Danesh (1998)

Using Eq. (2), we have the IFT correlation plot for the Danesh model, shown in Fig. 2.

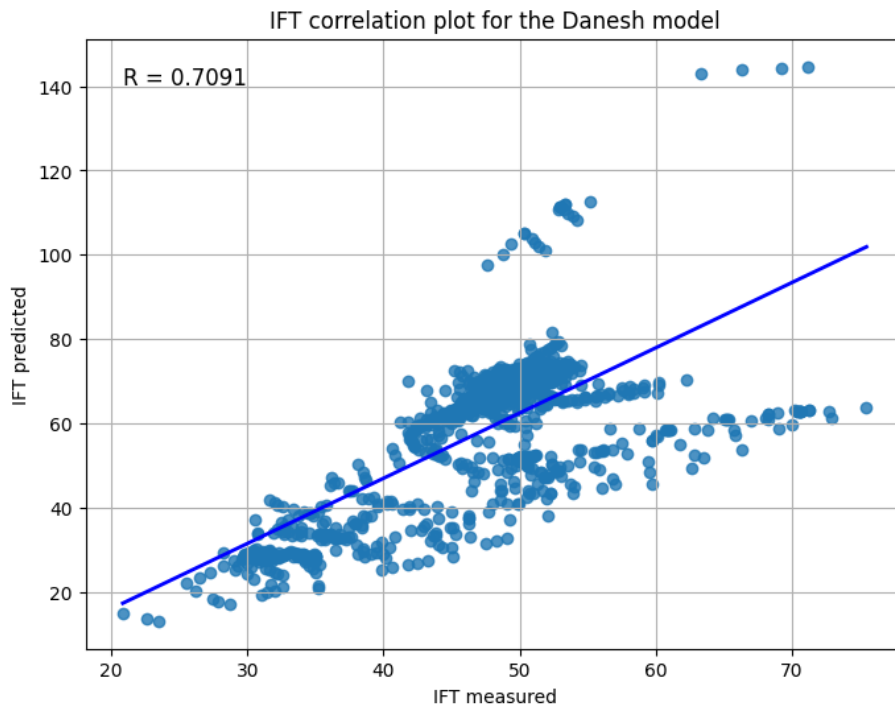


Fig. 2. IFT correlation plot for the Danesh model

b) Sutton (2006)

Using Eq. (3), we have the IFT correlation plot for the Sutton (2006) model, shown in Fig. 3.

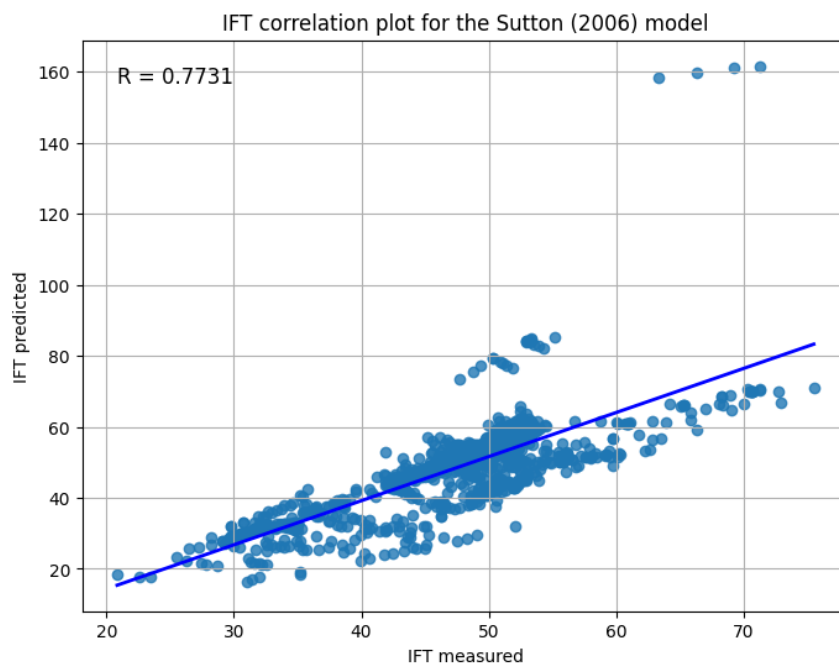


Fig. 3. IFT correlation plot for the Sutton (2006) model

c) Sutton (2009)

Using Eqs. (4) and (5), we have the IFT correlation plot for the Sutton (2009) model, shown in Fig. 4.

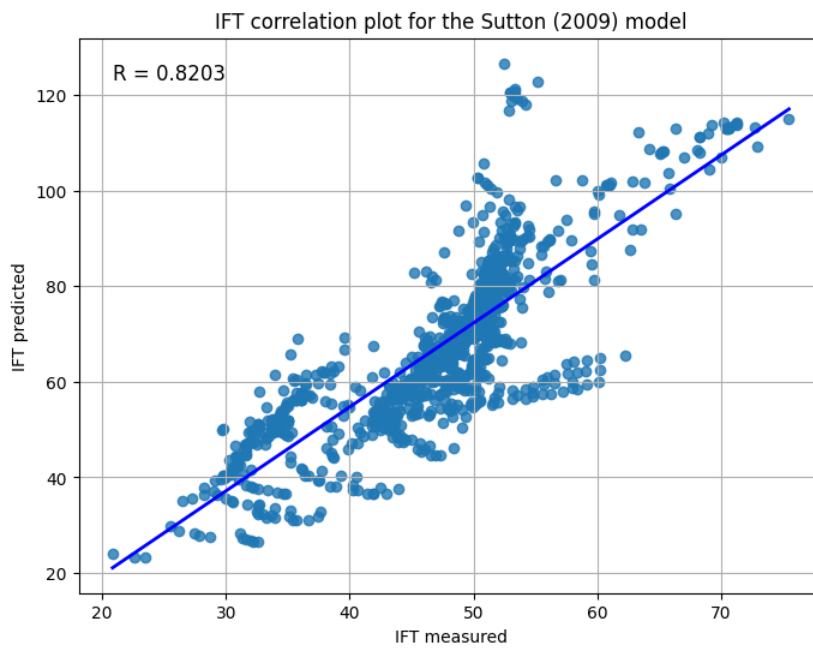


Fig. 4. IFT correlation plot for the Sutton (2009) model

d) Meybodi et al. (2016)

Using Eq. (6), we have the IFT correlation plot for the Meybodi et al. model, shown in Fig. 5.

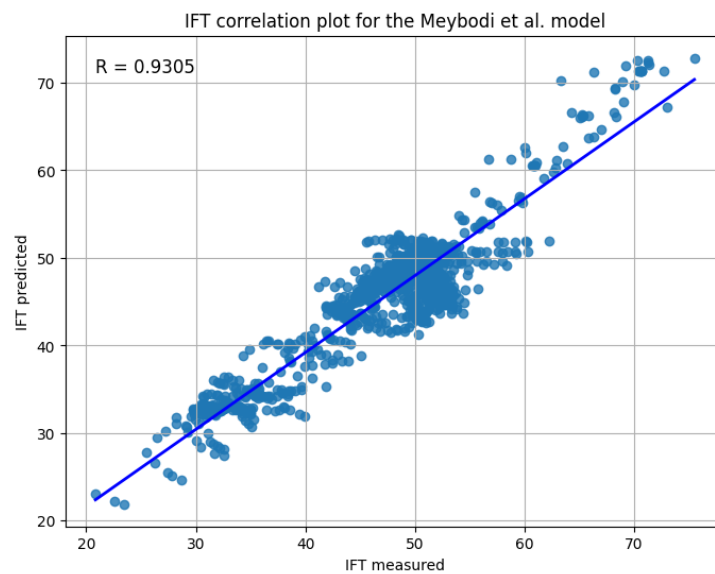


Fig. 5. IFT correlation plot for the Meybodi et al. model

3.2.2 Machine learning algorithms

a) MLP

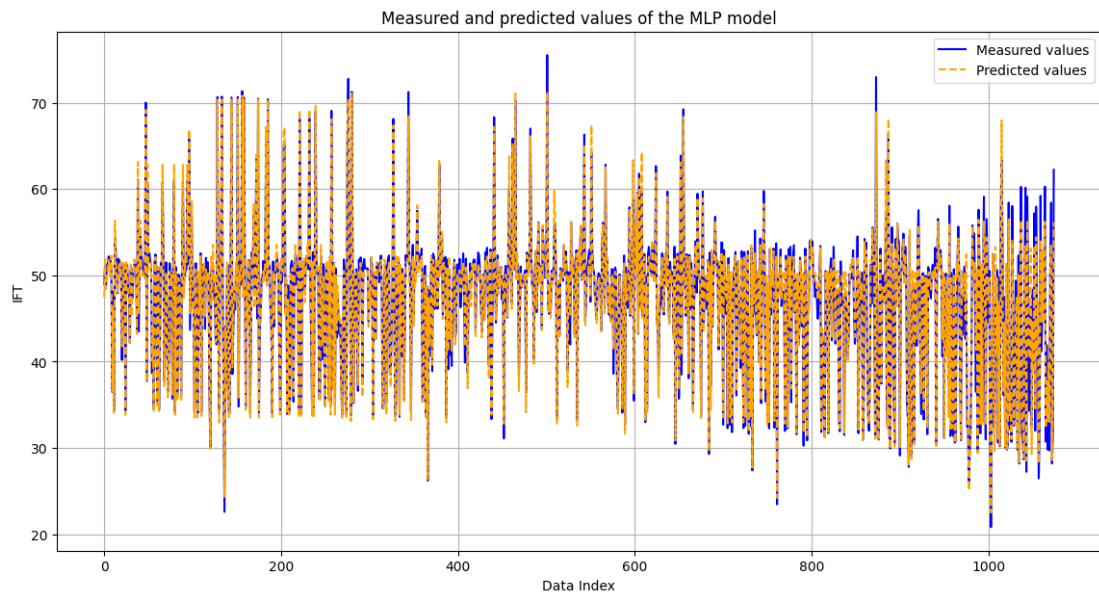


Figure 6. The comparison of measured and predicted values for MLP model

Based on MLPRegressor function in the scikit-learn library [27], the measured and predicted values plot for MLP model, as shown in Fig. 6. The neural network architecture includes 5 hidden layers; 1 layer has 100 neurons, and the transfer function used is the ReLU. Figure 7 show the IFT correlation plot for MLP model.

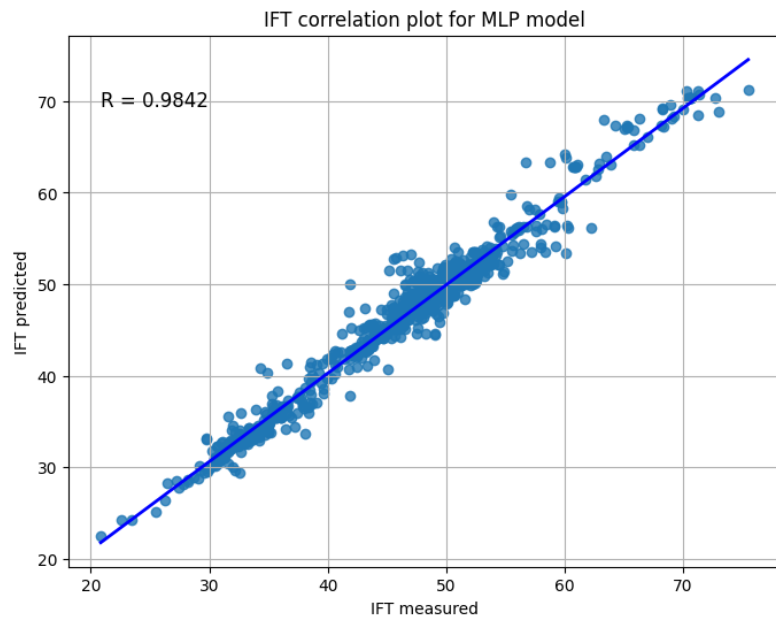


Fig. 7. The IFT correlation plot for MLP model

b) ET

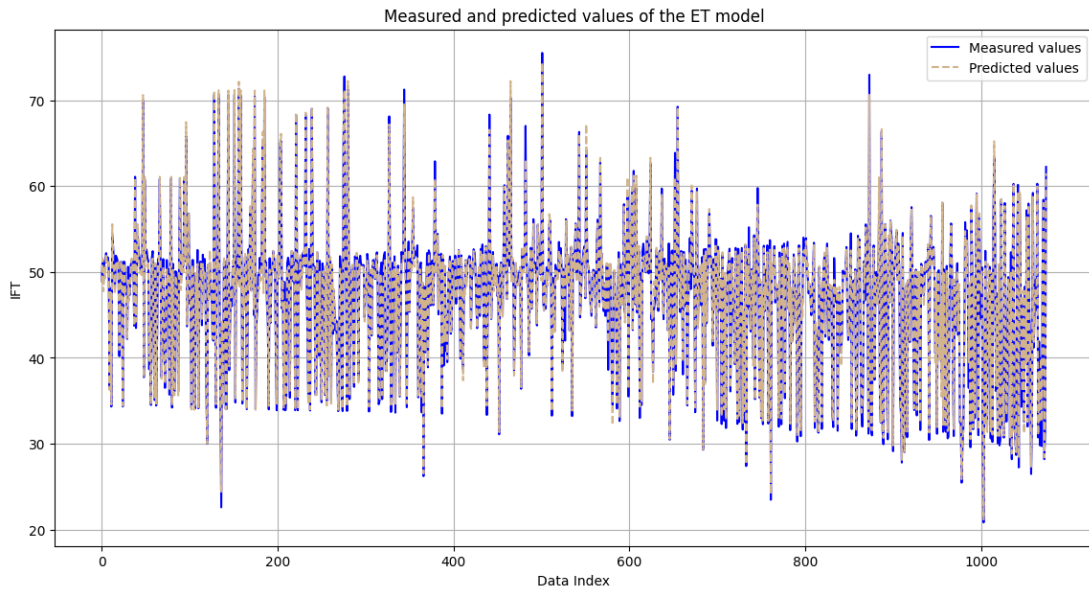


Fig. 8. The comparison of measured and predicted values for ET model

Based on ExtraTreesRegressor function in the scikit-learn library [28], the the measured and predicted values plot for ET model, as shown in Fig. 5. The number of decision trees in the forest is 100. Figure 9 show IFT correlation plot for ET model.

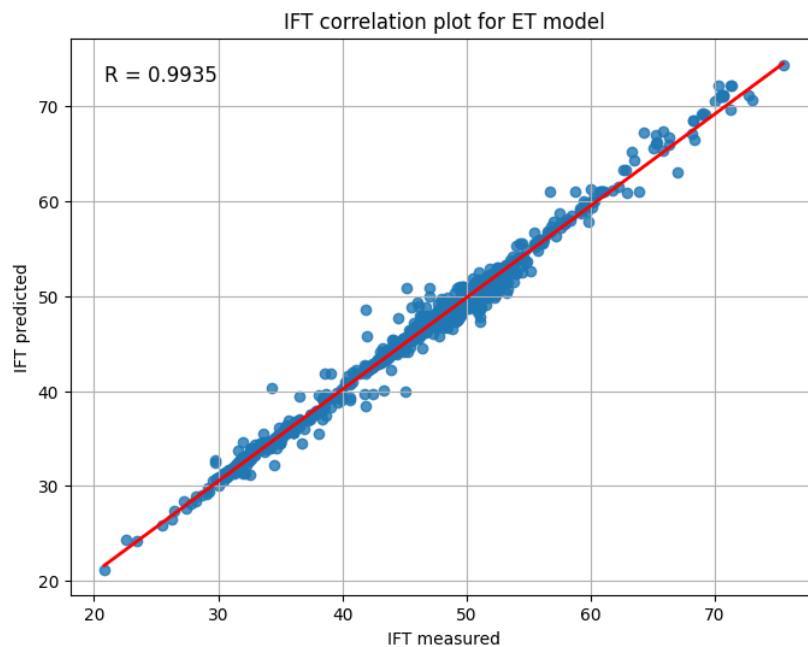


Fig. 9. The IFT correlation plot for ET model

3.3 Discussions

The comparison of all methods for estimating hydrocarbon-water interfacial tension is shown in Table 3.

Tab. 3. The comparison of IFT estimating methods

| Model | MAE | RMSE | R ² |
|----------------|---------|---------|----------------|
| Danesh | 14.3188 | 17.5264 | 0.5028 |
| Sutton (2006) | 5.2701 | 8.8777 | 0.5977 |
| Sutton (2009) | 20.3189 | 23.4720 | 0.6729 |
| Meybodi et al. | 2.6879 | 3.5127 | 0.8659 |
| MLP | 1.0157 | 1.4932 | 0.9687 |

| | | | |
|----|--------|--------|--------|
| ET | 0.6391 | 0.9565 | 0.9871 |
|----|--------|--------|--------|

The outcomes of Table 3 are visualized in Fig. 10. Table 3 and Fig. 10 show that in this study, the ET model has the most accurate prediction results, with the lowest *MAE* and *RMSE* and the highest *R*². In addition, the results also indicate the clear robust of the two machine learning models (MLP and ET) over the empirical correlation models with the lower *MAE* and *RMSE* and the higher *R*².

Comparison of IFT estimating models

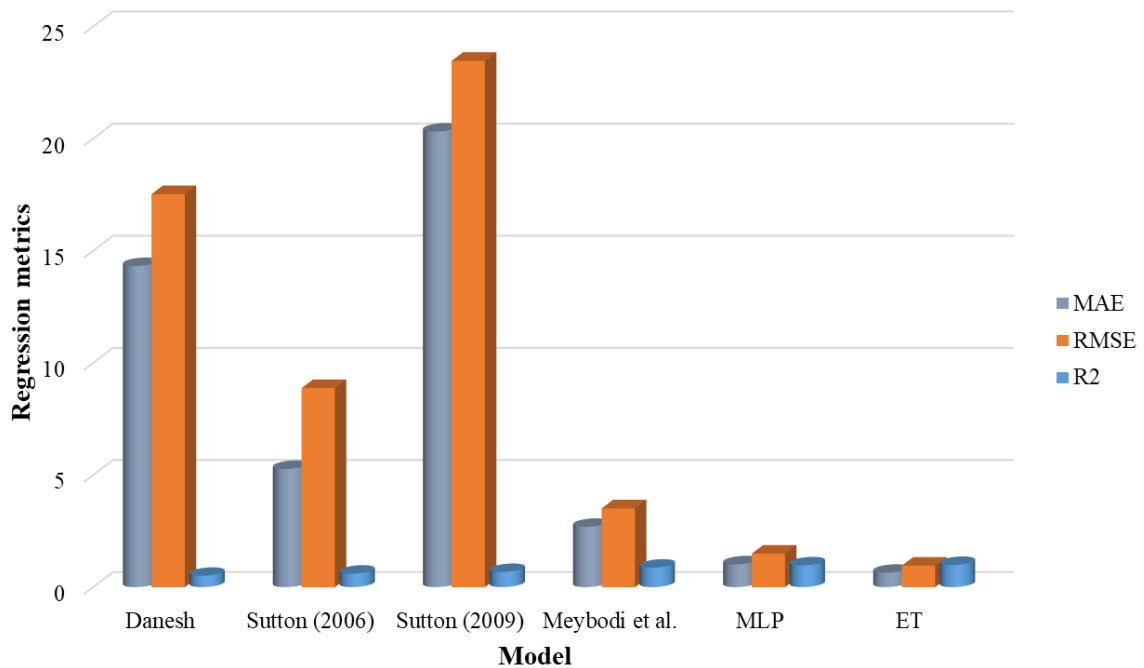


Fig. 10. The comparison of IFT estimating models

In the group of empirical correlation models, one point to note is that Sutton (2009) correlation not only have *R*² value but also *MAE* and *RMSE* values higher than Danesh and Sutton (2006) correlations. It demonstrates that the Sutton (2009) model might better capture non-linear relationships in the data, which would result in a better explanation of variance but also lead to larger errors in comparison with the Danesh and Sutton (2006) models. Furthermore, Meybodi et al. correlation is best model in this group based on lowest *MAE* and *RMSE* and the highest *R*².

Tab. 4. The comparison of IFT machine learning models

| Model | Dataset | <i>MAE</i> | <i>RMSE</i> | <i>R</i> ² |
|-------|---------|------------|-------------|-----------------------|
| MLP | Train | 0.9795 | 1.4563 | 0.9705 |
| | Test | 1.0706 | 1.5588 | 0.9644 |
| | All | 1.0157 | 1.4932 | 0.9687 |
| ET | Train | 0.0407 | 0.1825 | 0.9995 |
| | Test | 0.7772 | 1.2428 | 0.9774 |
| | All | 0.6391 | 0.9565 | 0.9871 |

By the group of machine learning models, Table 4 shows the regression metrics for both train, test, and all datasets. The results of Table 4 demonstrate the similar trend when the *MAE* and *RMSE* values of all MLP and ET models increase throughout the whole of the test, all, and train datasets, respectively, while the *R*² value decreases. This shows that the machine learning models do not have overfitting phenomena and have high confidence in using hydrocarbon-water IFT prediction.

4. Conclusions

This paper make a comparative study of methods for hydrocarbon-water interfacial tension based on empirical correlations and machine learning algorithms. The research data is collected from open literature. The key findings about this work are:

- The original data has 1105 points, including 5 features. After processing the non-numerical data, we have 1075 data points. The processing dataset heatmap correlation plot shows that all 4 input features (T_C , T_R , $\Delta\rho$, P) are related to the target feature (IFT). Therefore, all these features are used to train the machine learning models.

- The IFT estimation models were compared using regression metrics, which showed that ET algorithm is the best model in this study with the highest R^2 value (0.9871) and the lowest MAE and $RMSE$ (0.6391 and 0.9565, respectively).

- The results also point to machine learning models having better estimation than empirical correlations model. This note is drawn because MLP and ET have higher R^2 and lower MAE and $RMSE$ than four empirical correlations.

- In the group of empirical correlations, attention should be paid to the overfitting phenomenon of the Sutton (2009) model. It has both R^2 , MAE , and $RMSE$ values higher than Danesh and Sutton (2006) correlations.

- In the group of machine learning models, increasing trend of MAE and $RMSE$ values as well as decreasing trend of R^2 value for test, all, and train datasets sequentially denote MLP and ET models not be ruled by overfitting phenomena. Therefore, both models in general, but especially the ET model, are highly reliable for estimating hydrocarbon-water IFT.

The findings of this study, according to the authors, can be used with different datasets, and the study ought to be developed for other machine learning algorithms, such as deep learning or ensemble learning, to advance the estimation of hydrocarbon-water IFT.

Acknowledgements

The authors would like to thank Ho Chi Minh City University of Technology (HCMUT), VNU-HCM with the support of time and facilities for this study.

Literature - References

1. D. Truzzolillo, S. Mora, C. Dupas, and L. Cipelletti, "Nonequilibrium interfacial tension in simple and complex fluids," *Phys. Rev. X*, vol. 6, no. 4, pp. 1–12, 2016, doi: 10.1103/PhysRevX.6.041057.
2. A. Giustiniani, W. Drenckhan, and C. Poulard, "Interfacial tension of reactive, liquid interfaces and its consequences," *Adv. Colloid Interface Sci.*, vol. 247, no. May, pp. 185–197, 2017, doi: 10.1016/j.cis.2017.07.017.
3. C. G. Aranda-Bravo, A. Romero-Martínez, A. Trejo, and J. Águila-Hernández, "Interfacial tension and density of water + branched hydrocarbon binary systems in the range 303-343 K," 2009. doi: 10.1021/ie801101r.
4. M. Lee and A. Bahadori, "Chemical Engineering Research and Design A computational intelligence scheme for prediction of interfacial tension between pure hydrocarbons," *Chem. Eng. Res. Des.*, vol. 95, pp. 79–92, 2015, doi: 10.1016/j.cherd.2015.01.004.
5. J. Drelich, "Measurement of Interfacial Tension in Fluid-Fluid Systems," *Encycl. Surf. Colloid Sci.*, pp. 3152–3166, 2002, [Online]. Available: <http://www.informaworld.com/10.1081/E-ESCS-120000636>
6. M. Kalantari Meybodi, A. Daryasafar, and M. Karimi, "Determination of hydrocarbon-water interfacial tension using a new empirical correlation," *Fluid Phase Equilib.*, vol. 415, pp. 42–50, 2016, doi: 10.1016/j.fluid.2016.01.037.
7. A. Danesh, "Interfacial tension," *Dev. Pet. Sci.*, vol. 47, no. C, pp. 281–299, 1998, doi: 10.1016/S0376-7361(98)80030-7.
8. R. P. Sutton, "Chapter 6 Oil System Correlations," in *Petroleum Engineering Handbook - General Engineering*, 2006.
9. R. P. Sutton, "An improved model for water-hydrocarbon surface tension at reservoir conditions," *Proc. - SPE Annu. Tech. Conf. Exhib.*, vol. 6, no. 3, pp. 3968–3985, 2009, doi: 10.2118/124968-ms.
10. J. Zhang *et al.*, "The use of an artificial neural network to estimate natural gas/water interfacial tension," *Fuel*, vol. 157, pp. 28–36, 2015, doi: 10.1016/j.fuel.2015.04.057.
11. M. H. Emami Baghdadi, H. Darvish, H. Rezaei, and M. Savadinezhad, "Applying LSSVM algorithm as a novel and accurate method for estimation of interfacial tension of brine and hydrocarbons," *Pet.*

- Sci. Technol.*, vol. 36, no. 15, pp. 1170–1174, 2018, doi: 10.1080/10916466.2018.1465963.
12. F. Yousefmarzi, A. Haratian, J. Mahdavi Kalatehno, and M. Keihani Kamal, “Machine learning approaches for estimating interfacial tension between oil/gas and oil/water systems: a performance analysis,” *Sci. Rep.*, vol. 14, no. 1, pp. 1–19, 2024, doi: 10.1038/s41598-024-51597-4.
 13. A. Najafi-Marghmaleki, A. Tatar, A. Barati-Harooni, A. Mohebbi, M. Kalantari-Meybodi, and A. H. Mohammadi, “On the prediction of interfacial tension (IFT) for water-hydrocarbon gas system,” *J. Mol. Liq.*, vol. 224, pp. 976–990, 2016, doi: 10.1016/j.molliq.2016.10.083.
 14. G. F. Marcus, “Multilayer Perceptrons,” *Algebr. Mind*, pp. 7–34, 2019, doi: 10.7551/mitpress/1187.003.0004.
 15. P. A. Jansson, “Neural Networks: An Overview,” *Anal. Chem.*, vol. 63, no. 6, pp. 357–362, 1991, doi: 10.1021/ac00006a739.
 16. E. Domany, “Neural networks: A biased overview,” *J. Stat. Phys.*, vol. 51, no. 5–6, pp. 743–775, 1988, doi: 10.1007/BF01014882.
 17. F. Murtagh, “Multilayer perceptrons for classification and regression,” *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, 1991, doi: 10.1016/0925-2312(91)90023-5.
 18. S. KILIÇARSLAN, K. ADEM, and M. ÇELİK, “An overview of the activation functions used in deep learning algorithms,” *J. New Results Sci.*, vol. 10, no. 3, pp. 75–88, 2021, doi: 10.54187/jnrs.1011739.
 19. J. Henseler, “Back Propagation,” *Lect. Notes Comput. Sci.*, vol. 931, 1995, doi: <https://doi.org/10.1007/BFb0027022>.
 20. X. Yuan, J. Shi, and L. Gu, “A review of deep learning methods for semantic segmentation of remote sensing imagery,” *Expert Syst. Appl.*, vol. 169, no. June 2020, p. 114417, 2021, doi: 10.1016/j.eswa.2020.114417.
 21. S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016, [Online]. Available: <http://arxiv.org/abs/1609.04747>
 22. P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.
 23. L. Rokach and O. Maimon, “Decision Trees,” *Lect. Notes Math.*, vol. 1928, pp. 67–86, 2008, doi: 10.1007/978-3-540-75859-4_5.
 24. Z. tao Chen, Y. kuo Liu, N. Chao, M. M. Ernest, and X. li Guo, “Gamma-rays buildup factor calculation using regression and Extra-Trees,” *Radiat. Phys. Chem.*, vol. 209, no. April, p. 110997, 2023, doi: 10.1016/j.radphyschem.2023.110997.
 25. A. Campagner, D. Ciucci, and F. Cabitza, “Aggregation models in ensemble learning: A large-scale comparison,” *Inf. Fusion*, vol. 90, no. September 2022, pp. 241–252, 2023, doi: 10.1016/j.inffus.2022.09.015.
 26. R. Quan, “iScience II Dung beetle optimization algorithm-based hybrid deep learning model for ultra-short-term PV power prediction,” *ISCIENCE*, vol. 27, no. 11, p. 111126, 2024, doi: 10.1016/j.isci.2024.111126.
 27. S. Developers, “MLPRegressor.” [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.neural_network.MLPRegressor.html
 28. S. Developers, “ExtraTreesRegressor.” [Online]. Available: <https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html>