

Group Project Assignment

Screening the polymer flooding using artificial-neural-network

School of Mining and Geosciences

Petroleum Data Analytics

Professor Masoud Riazi

November 11, 2023

2023

**Table of content**

Introduction………………………………………………………….……………....2

Literature review……………………… …………...……...…………………….….2

Workflow…….………………………………………………….…………………...5

Data gathering……………………….…...……………………………………….…4

Data preprocessing……………………………….…………….……………………6

Conclusion…………………………………………………………….……..….......8

References.………………………………………………………………...…….......9

**Introduction**

According to a study of Sun and Ertekin (2020b), the polymer injection is a popular chemical enhanced oil recovery (CEOR) method that is often used after water flooding. A predetermined amount of polymer solution, commonly measured by fraction of pore volume (PV), is injected into the reservoir to decrease aqueous phase mobility and hence increase sweep efficiency. Due to the cheap injection fluid cost and decent incremental recovery, field applications show that application of polymer flooding is more practical in comparison with alternative chemical EOR methods.

By effectively tackling a wide range of reservoir engineering tasks, machine-learning (ML)-based proxy models have demonstrated their computational efficiency and generalization features. One of the most frequently applied ML algorithms in the petroleum sector is the artificial neural network. Structured ANN models have been developed for optimization of field development, defining reservoir characteristics, virtual well testing analysis, infill drilling, and enhanced oil recovery field implementation.

**Literature review**

The research from Sun and Ertekin (2020b), “*Screening and optimization of polymer flooding projects using artificial-neural-network (ANN) based proxies*”, involves creating a synthetic field history using the commercial chemical EOR simulator, CMG-STARS, to build a knowledgebase for training artificial neural networks (ANN). The study utilizes a two-dimensional model with a 67 x 67 grid structure, which represents one-eighth of a 5-spot injection pattern, as illustrated in figure 1. The external boundaries are entirely closed off. The reservoir's properties are consistent and exhibit equal permeability in all directions. The study assumes that the effects of gravitational forces and capillary pressure are minimal. The reservoir fluid model comprises four components: water, polymer, dead oil, and total dissolved solids.

A close-up of a grid

Description automatically generated

Figure 1. Reservoir model with 67 x 67 grid structure.

In the research the ANN systems were built and trained to resolve the following reservoir engineering problems:

* Forward-looking problems: To estimate the project reaction, ANN uses reservoir characteristics and project design for input parameters. In most cases, forward-looking ANN models are used as forecasting proxies to anticipate fluid output and pressure responses.
* Inverse project design problems: ANN forecasts the project design parameters needed to attain a desired result. The project design ANNs use hydrocarbon recovery and reservoir rock and fluid characteristics as input to estimate project design parameters.

As depicted in figure 2 and figure 3, two single-layer feedforward ANN models are trained, with model one having 60 hidden neurons for the forward-looking application and model two having 75 hidden neurons for the inverse design application.

A diagram of a neural network

Description automatically generated

Figure 2. Forward-looking ANN topology.

A diagram of a neural network

Description automatically generated

Figure 3. Inverse design ANN topology.

To build a Forward-looking and Inverse-design models, the reservoir rock properties (such as porosity, permeability, reservoir depth, pressure, etc.), initial conditions (initial oil, water, and saturation), fluid properties (oil and brine density, viscosity, etc.), polymer properties (concentration, viscosity, MW) were used as an input data to get the water and oil production rates, as illustrated in figure 4.

A table with text and words

Description automatically generated

Figure 4. Input and output parameters.

The forward-looking ANN model is evaluated using 200 blind testing, and the results are illustrated in figure 5. Impressively, the majority (89%) of these tests exhibit relative errors below 10%. Among the test cases, the largest relative error is 13.37%, with an average of 5.83%. This statistical analysis demonstrates that the forward-looking ANN models have been effectively trained. The study also presents comparisons between the predictions of project response functions by the ANN models and data from a high-fidelity model. The accuracy of the ANN models is illustrated across various scenarios, with even the worst cases showing satisfactory matches during the water flooding phase.

A graph of a number of blue bars

Description automatically generated

Figure 5. Forward-looking ANN blind testing error statistics.

In the study by Al-Ghazal and Ertekin (2018b), “*Assisted Design of Polymer-Gel Floods in Naturally Fractured Reservoirs Using Neuro-Simulation Based Models*”, authors applied same procedure to build the ANN, but for naturally fracture reservoir. The reservoir in the study is described as dual-permeability and three-dimensional. The reservoir is considered to be homogenous in each model, with three layers of identical thickness. The model has two vertical wells: an injector well and a producer well, as illustrated in figure 6. The positioning of these two wells is chosen to represent one-fourth of a five spot pattern.

A colorful cube with black triangles

Description automatically generated with medium confidence

Figure 6. Synthetic reservoir model.

The forward-looking model's ANN structure consists of six hidden layers with a total of 670 neurons. These neurons are distributed across the layers as follows: 90, 95, 110, 150, 110, and 115. Additionally, the input layer comprises 29 neurons, while each set of 68 neurons in the output layer corresponds to oil rate, water cut, and recovery factor.

The Inverse ANN-1 model considers five design parameters, such as injection rate, polymer concentration, etc. The ANN structure includes a single hidden layer with 9 neurons. The input layer consists of 215 neurons, accommodating reservoir properties and production data. The output layer, with 11 neurons, estimates design parameters.

The dataset was organized as follows: 90% for training, 5% for validation, and 5% for testing at random.

The paper titled “*Assisted Design of Polymer-Gel Floods in Naturally Fractured Reservoirs Using Neuro-Simulation Based Models*” was also considered for the purpose of learning how to handle polymer flooding data. This study focuses on providing Enhanced-oil-recovery (EOR) screening criteria addressing the data set quality for polymer-flooding projects, including the issues as outliers, duplicated, missing and inconsistent data. 250 projects out of 481 were filtered out through using boxplots, crossplots, other than that, statistical methods were applied to data eliminating those inconsistencies in data.

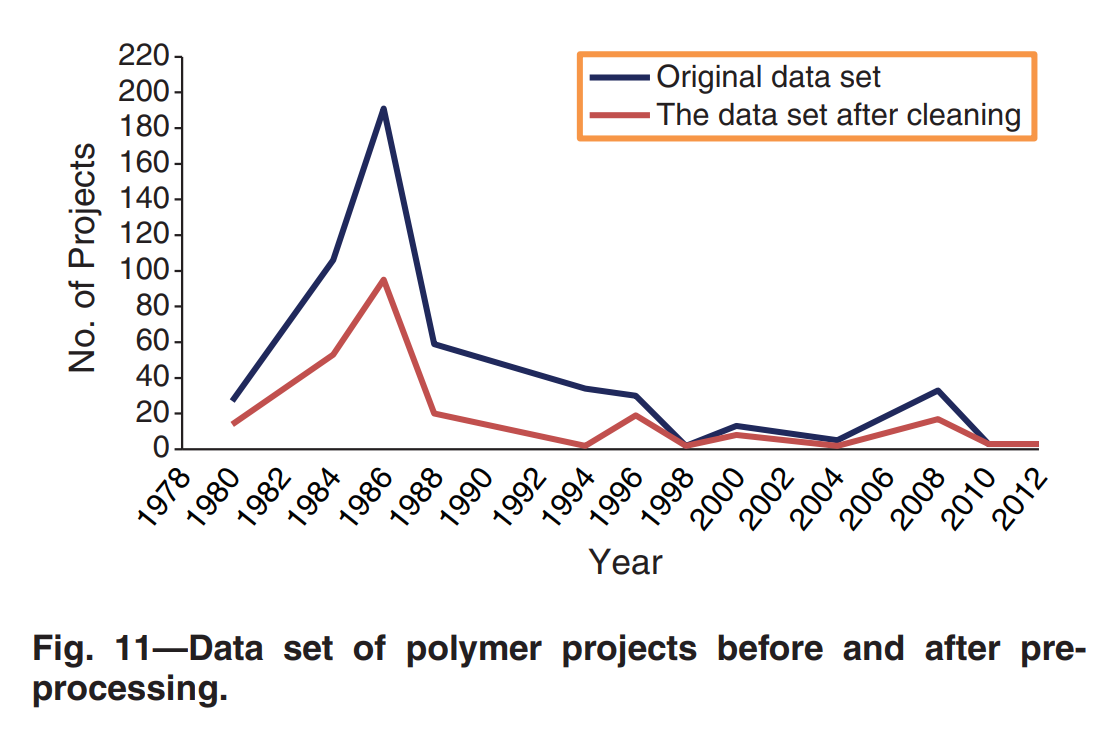


Figure 7. Data set of polymer projects before and after pre-processing

The developed criteria are then formulated showing which properties reservoir should have to successfully apply a polymer-flooding project on is formulated:

* Duplicate data (several countries not updating EOR information, solved by removing data);
* Missing data (several fields missing: permeability, oil saturation (start and end), depth, temperature; solved by ignoring missing values);
* Inconsistent data (impossible values and discrepancies: oil saturation(end) > oil saturation(start), oil saturation (start) < 20%, average reservoir permeability < 1 md, porosity > 40%; detected using boxplots and crossplots).

**Workflow**

The project was planned ahead and all of the respective steps were included in the workflow chart design.

**Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание**  
Figure 8. Workflow chart design

**Data gathering**

Data gathering and preprocessing was implemented according to Saleh et al. (2014c). A data set was created by gathering information about EOR projects. The Worldwide EOR Survey 2008 and Worldwide EOR Survey 2014 published by the Oil and Gas Journal were the source of the data set REFERENCE. Initially data was in PDF format and then transferred to Excel as a table, to be able to preprocess it later.

Data transferring was done in several steps:

1. Open Excel.
2. Go to “Data” option in the menu.
3. Go to “Get Data”, click “From File”, then pick “From PDF”.
4. Import Data.
5. Select tables you need.
6. Load Data.

The gathered data was saved in .xlsx format and contains the following columns:

* Area, acres
* Num. production wells
* Num. injection wells
* Porosity, %
* Permeability, md
* Depth, ft
* Gravity, °API
* Oil viscosity, cp
* Oil temperature, °F
* Satur., % start
* Satur., % end
* Tot.prod., b/d
* Enh.prod., b/d

**Data preprocessing**

Imported data then underwent data preprocessing in Python. For this purpose, Python libraries such as Pandas, Numpy, Matplotlib, Seaborn were used. Data was then studied, and preprocessing began.

The steps stated in Saleh et al. (2014c) were followed. Initially, in contrast to the approach in the referred paper, missing values were not ignored and all data instances with any missing or NaN values were dropped. After this, all duplicates were deleted.

Dataset had many issues with inconsistencies, all of which were addressed, these include the following problems:

* oil saturation(end) > oil saturation(start),
* oil saturation (start) < 20%,
* permeability < 1 md,
* porosity > 40%,
* values given as intervals instead of integer or float values,
* additional text (e.g. “ft” in “Depth” column)

|  |  |
| --- | --- |
| Step | Data size after |
| Initial | (698, 13) |
| Dropping missing values | (675, 13) |
| Dropping duplicates | (651, 13) |
| Cleaning data with extra text, transforming data with interval values | (650, 13) |
| Drop data with oil saturation (start) < 20% | (548, 13) |
| Drop data with oil saturation(end) > oil saturation(start) | (548, 13) |
| Drop data with permeability < 1 md | (522, 13) |
| Drop data with porosity > 40% | (515, 13) |

Figure 9. Data preprocessing steps

To handle these points, Regex and Python methods on strings were used. All changes were conducted step-by-step and dataset size after each change was tracked.

To observe noises and extreme values, boxplots and histograms were plotted. Using both boxplots and histograms, it was depicted that the columns “Area, acres”, “Permeability, md”, “Oil viscosity, cp”, “Tot.prod., b/d”, “Enh.prod., b/d” have the greatest extreme values. To decide whether to remove these outliers from the dataset or not, outlying instances were reviewed and their values for their other columns were checked. It was determined that those values are not totally “noise”, meaning that there were no points that were separated from the dataset.

Изображение выглядит как текст, снимок экрана, число, линия

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание Изображение выглядит как текст, линия, чек, снимок экрана

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, чек, диаграмма

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, число, диаграмма

Автоматически созданное описание

Figure 10. Box plots for data columns with extreme outliers

Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

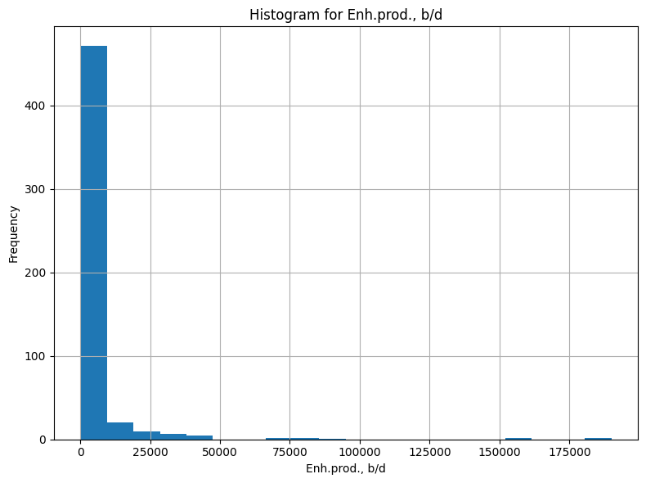


Figure 11. Histograms for data columns with extreme outliers

Then, crossplot was made to observe whether certain model parameters are correlated with each other or not, for the quality of model predictions. There were several dependencies close to linear between certain parameters, which indeed are explained by physical laws. For example, there is a strong linear relationship between Depth, ft and Gravity, API; between Depth, ft and Oil temperature, °F. These can be justified by the physical relationship between these parameters, indeed they are not eliminated due to their important meaning. As the previous papers did not exclude any parameters, it was decided to proceed with this way.

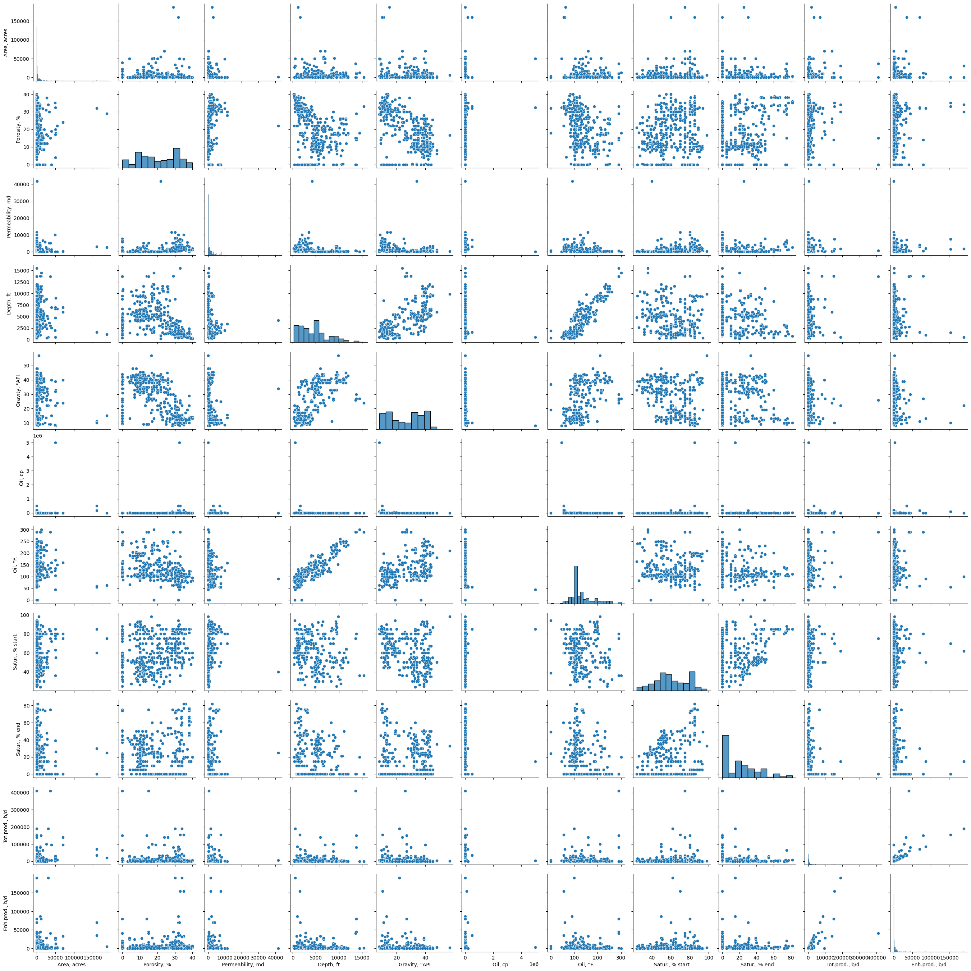


Figure 12. Crossplot for the data

As all the important data visualizations were made, the preprocessed data was extracted and saved.

After initial attempt to train forward-looking ANN model, extreme values of validation loss of 124858464 and validation mean absolute error of 6245.9058 were observed. This issue is caused due to all variables having different scales, therefore min-max normalization was applied on all data columns with the formula of

with being the maximum and being the minimum value in the respective column.

**Model implementation: Forward-looking ANN**

To solve the problem of screening polymer flooding, two models (forward-looking ANN and inverse design ANN) were suggested in Sun and Ertekin (2020b), “*Screening and optimization of polymer flooding projects using artificial-neural-network (ANN) based proxies*”. Project work aimed to replicate these models using our dataset.

Selection of the input and output variables for both forward-looking ANN and inverse design ANN was based on Sun and Ertekin (2020b) as well, selecting the variables that were available in our dataset as input and output.

So, for forward-looking ANN model, the variable selection was as follows:

**Input parameters:**

* Area, acres
* Porosity, %
* Permeability, md
* Depth, ft
* Gravity, °API
* Oil viscosity, cp
* Oil temperature, °F
* Initial oil saturation, %
* End oil saturation, %

**Output parameters**:

* Tot.prod., b/d
* Enh.prod., b/d

The model architecture also followed Sun and Ertekin (2020b) with input, output layers and one hidden layer of size 60. The model was initialized and compiled in Python using the Tensorflow library, then the model was trained using the data. Along with that, plots of training/validation loss, training/validation metric values versus epochs were made. To increase the model performance, the following variations of the original forward-looking ANN model (learning rate = 0.001) are introduced:

* Model with decreased learning rate (0.0005)
* Model with increased learning rate (0.005)
* Model with Dropout layer
* Model with additional Hidden layer.

Based on validation loss and validation metric (MAE) values, the additional model combining variations improving the original model performance was trained as well. The results of the performance evaluation of these forward-looking ANN models were saved in a tabular form.

|  |  |  |
| --- | --- | --- |
| Model | Loss (MSE) | Metric (MAE) |
| Original | 0.0025 | 0.0283 |
| Learning rate = 0.0005 | 0.0020 | 0.0251 |
| Learning rate = 0.005 | 0.0025 | 0.0270 |
| Dropout layer | 0.0018 | 0.0237 |
| Additional Hidden Layer | 0.0023 | 0.0269 |
| Learning rate = 0.0005, Dropout Layer, Additional Hidden Layer | 0.0018 | 0.0201 |

Figure 13. Performance evaluation on forward-looking ANN

**Model implementation: Inverse design ANN**

For inverse design ANN model, the variable selection was as follows:

**Input parameters:**

* Area, acres
* Permeability, md
* Depth, ft
* Gravity, °API
* Oil viscosity, cp
* Oil temperature, °F
* Initial oil saturation, %
* End oil saturation, %

**Output parameters**:

* Num. production wells
* Num. injection wells

The model architecture followed Sun and Ertekin (2020b) as in case of forward-looking ANN, with input, output layers and one hidden layer of size 75. Model initialization, compiling and training was done similarly to forward-looking ANN models. To increase the model performance, the following variations of the original inverse design ANN model (learning rate = 0.001) are introduced:

* Model with decreased learning rate (0.0005)
* Model with increased learning rate (0.005)
* Model with Dropout layer
* Model with additional Hidden layer.

As with forward-looking ANN models, judging from the validation loss (MSE) and the validation metric (MAE) values, the new model combining variations improving the original model performance was trained as well. The results of the performance evaluation of these inverse design ANN models were saved in a tabular form.

|  |  |  |
| --- | --- | --- |
| Model | Loss (MSE) | Metric (MAE) |
| Original | 9.2973e-04 | 0.0201 |
| Learning rate = 0.0005 | 8.3254e-04 | 0.0210 |
| Learning rate = 0.005 | 8.1867e-04 | 0.0189 |
| Dropout layer | 7.2635e-04 | 0.0188 |
| Additional Hidden Layer | 8.1966e-04 | 0.0179 |
| Learning rate = 0.005, Dropout Layer, Additional Hidden Layer | 6.1793e-04 | 0.0145 |

Figure 14. Performance evaluation on inverse design ANN

**Discussion**

Based on the validation metric (MAE), and, more importantly, the validation loss (MSE) values, the best model architectures for forward-looking ANN and inverse design ANN were deduced.

|  |  |  |
| --- | --- | --- |
| Model | Loss (MSE) | Metric (MAE) |
| Forward-looking ANN (Learning rate = 0.0005, Dropout Layer, Additional Hidden Layer) | 0.0018 | 0.0201 |
| Inverse design ANN (Learning rate = 0.005, Dropout Layer, Additional Hidden Layer) | 6.1793e-04 | 0.0145 |

Figure 15. Best performing ANN models

The results show that addition of Dropout and additional Hidden layers, tuning the learning rate increases the model performance. The selected models achieve excessively good performance with loss values of 0.0018 and 6.1793e-04, given the simplicity of the models not requiring much time and memory to be implemented, there is no need to look for more complex architectures as these already show more than acceptable performance.

Determining the most important features of the trained models is another goal of this project work. Therefore, for the selected best-performing forward-looking ANN and inverse design ANN models, the feature importance values were computed. In this purpose, weight-based method is used. Feature importance is estimated using the weights in the neural network with the weight values is directly proportional to feature importance. Since both forward-looking ANN and inverse design ANN models are fully connected neural networks, usage of this method is valid. So, the feature importance is computed as the sum of the absolute weights for each feature.

|  |  |  |
| --- | --- | --- |
| Feature | Forward-looking ANN | Inverse design ANN |
| Area, acres | 8.191454 | - |
| Porosity, % | 8.165033 | - |
| Permeability, mD | 9.12878 | 10.403721 |
| Depth, ft | 9.260663 | 10.693703 |
| Gravity, °API | 9.659689 | 9.946156 |
| Oil viscosity, cp | 9.251195 | 11.329239 |
| Oil temperature, °F | 9.238607 | 9.907521 |
| Initial oil saturation, % | 8.05657 | 8.048957 |
| End oil saturation, % | 9.057095 | 10.026518 |

Figure 16. Feature importance values

For both models, all features have approximately similar feature importance values, with 'Gravity, °API' being the most important feature for forward-looking ANN (with score of 9.659689) and 'Oil viscosity, cp' being the most important feature for inverse design ANN (with score of 11.329239). The least important features were observed to be 'Initial oil saturation, %' for both forward-looking ANN and inverse design ANN (with scores of 8.06657 and 8.048957).

Analyzing the plots of training/validation loss versus epochs and training/validation mean absolute error versus epochs for both models, it can be observed that the trend is similar. For both cases, values fluctuate as the number of epochs eventually converging to certain value. Indeed, forward-looking ANN demonstrates bigger fluctuations with inverse design ANN showing more steady changes. In comparison with other performance plots (given in appendix), the positive effect of combination of tuning learning rate, adding dropout and additional hidden layers can be seen, resulting in smoother plot with maximalized performance on validation set. On top of that, loss and MAE values for validation set are bounded above by the values for training set, indicating the generalization ability of the models, meaning that both models will perform well on new, unseen data.

Изображение выглядит как текст, График, снимок экрана, линия

Автоматически созданное описание

Figure 17. Performance plots for best-performing Forward-looking ANN model

Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание

Figure 18. Performance plots for best-performing Inverse design ANN model

**Conclusion**

In this project, different ANN models for screening the polymer flooding were studied and appropriate neural network designs were selected.

Initially, suitable dataset was found using Worldwide EOR Surveys for 2009 and 2014 years. Then, data was preprocessed, and extreme values were analyzed using Pandas and Matplotlib libraries of Python, respectively. Then, the prepared dataset was used as an input for Forward-looking ANN and Inverse Design ANN models as proposed in Sun and Ertekin (2020b). The models and their variations with different learning rates, additional dropout and hidden layers were trained and their performances were analysed in terms of the validation loss (mean squared error), validation mean absolute error values and performance plots.

It was deduced that Forward-looking ANN with learning rate = 0.0005, dropout layer, additional hidden layer and Inverse design ANN with learning rate = 0.005, dropout Layer, additional hidden layer perform on our dataset the best with validation loss scores of 0.0018 and 6.1793e-04. Other than that, the best models also have good generalization ability proven by performance plots. These models are advised to be used in the screening of polymer flooding due to their simplicity, high accuracy, and high generalization ability. For future work, it is planned to learn more complex variations of the selected architectures and study their performances in comparison to the models obtained in this project.

**References:**

Sun, Q., & Ertekin, T. (2020b). Screening and optimization of polymer flooding projects using artificial-neural-network (ANN) based proxies. Journal of Petroleum Science and Engineering, 185, 106617. <https://doi.org/10.1016/j.petrol.2019.106617>

Al-Ghazal, M., & Ertekin, T. (2018b). Assisted Design of Polymer-Gel Floods in Naturally Fractured Reservoirs Using Neuro-Simulation Based Models. OnePetro. <https://doi.org/10.2118/192602-ms>

Saleh, L., Wei, M., & Bai, B. (2014c). Data analysis and updated screening criteria for polymer flooding based on oilfield data. SPE Reservoir Evaluation & Engineering, 17(01), 15–25. <https://doi.org/10.2118/168220-pa>

**Appendix**

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Figure 19. Performance plots for original Forward-looking ANN model before normalization

Изображение выглядит как текст, График, линия, снимок экрана

Автоматически созданное описание

Figure 20. Performance plots for original Forward-looking ANN model

Изображение выглядит как текст, График, снимок экрана, линия

Автоматически созданное описание

Figure 21. Performance plots for Forward-looking ANN model with decreased learning rate

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Figure 22. Performance plots for Forward-looking ANN model with increase learning rate

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Figure 23. Performance plots for Forward-looking ANN model with Dropout layer

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Figure 24. Performance plots for Forward-looking ANN model with additional hidden layer

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Figure 25. Performance plots for original Inverse design ANN model

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Figure 26. Performance plots for Inverse design ANN model with decreased learning rate

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Figure 27. Performance plots for Inverse design ANN model with increased learning rate

Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание

Figure 28. Performance plots for Inverse design ANN model with Dropout layer

Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Figure 29. Performance plots for Inverse design ANN model with additional hidden layer