On the effectiveness of Kolmogorov–Arnold Networks for enhanced oil recovery prediction in polymer flooding

Samson Dawit Bekele
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
samsondawitb@gmail.com

Yerzhan Kenzhebek
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
kenzhebekyerzhan@gmail.com

Timur Imankulov
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
imankulov.timur@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899678

Abstract— **Enhanced oil recovery techniques like polymer flooding are vital for maximizing hydrocarbon extraction, but predicting recovery factors remains challenging due to complex reservoir dynamics. Traditional artificial neural networks offer predictive power but lack interpretability, limiting their practical utility in engineering applications. This study introduces Kolmogorov–Arnold Networks (KANs) as an interpretable alternative for modeling recovery factors in polymer flooding. Using a synthetically generated dataset of over 160,000 samples with key reservoir parameters, we trained KAN models through a series of experiments, varying network architectures, grid sizes, learning rates, and optimizers (Adam and LBFGS) to evaluate performance and interpretability. The best-performing KAN achieved a Test Mean Squared Error (MSE) of 0.000597 and a Test coefficient of determination (R^2) of 0.902 with only 1,885 parameters, closely matching the performance of a previous deep neural network model that used 43,265 parameters (Test R² of 0.908). While KANs did not surpass the DNN in predictive accuracy, they offered comparable results with significantly reduced complexity and enhanced transparency. The modular structure and learnable activation functions of KANs provide insights into the decision-making process. This work contributes to the EOR field by demonstrating that KANs can effectively model complex reservoir behaviors while being transparent in their decision making.**

Keywords— Enhanced Oil Recovery, Kolmogorov–Arnold Networks, Interpretability, Recovery Factor Prediction, Artificial Neural Networks.

I. INTRODUCTION

The global demand for hydrocarbons continues to necessitate innovative methods to maximize recovery from existing reservoirs. Enhanced oil recovery (EOR) techniques, particularly polymer flooding, have demonstrated significant promise in improving oil extraction efficiency by modifying reservoir properties to enhance sweep efficiency and displacement of oil [1].

Despite its effectiveness, accurately predicting the recovery factor in polymer flooding is a complex task due to the nonlinear dynamics of fluid flow, reservoir heterogeneity, and the interplay of multiple physicochemical factors. Traditional numerical simulation methods, such as finite difference or finite element models, often involve computationally expensive calculations and may struggle to handle the variability of real-world reservoir conditions [2-4].

Recent advancements in machine learning (ML) have opened new avenues for data-driven modeling in oil recovery predictions [5]. Unlike conventional simulation methods, ML models can efficiently learn complex relationships from large datasets, offering faster and often more accurate predictions. Techniques such as neural networks [6], XGBoost [7] and support vector machines have shown significant potential in this domain [8, 9].

Polymer flooding has been extensively studied both in experimental and computational frameworks [10]. Simulation-based approaches, like those based on the Buckley–Leverett equations [11], have provided insights into the mechanics of polymer flooding.

Artificial Neural Networks (ANNs) have been extensively applied to EOR problems due to their ability to model complex nonlinear relationships and handle high-dimensional datasets. For instance, Saberi et al utilized ANNs to predict the performance of polymer flooding by considering parameters such as polymer concentration, salinity, and reservoir properties. The model achieved a high coefficient of determination (R^2) of 0.9990 indicating excellent predictive capability [12]. Similarly, Cheraghi et al. proposed a two-stage screening system employing ANNs to predict suitable EOR methods for candidate reservoirs [13]. Le Van and Chon developed ANN models to evaluate critical performances of CO_2-EOR processes and demonstrated the models' effectiveness in capturing complex reservoir behaviors [14]. Vo Thanh et al. applied ANNs to predict the performance of CO_2-EOR and storage in residual oil zones and they effectively highlighted the models' potential in optimizing CO_2 storage strategies [15]. Mohammadi et al. showcased the applicability of ANNs in simulating thermal recovery processes by employing cascade forward neural networks and the group method of data handling to model crude oil pyrolysis during thermal EOR [16].

The heavy usage of ANNs in this research field shows the versatility of the algorithm in addressing various EOR scenarios. However, despite their strengths, ANNs often function as "black-box" models – effectively making their predictions difficult to interpret [17]. Even if an ANN makes highly accurate predictions, it often leaves researchers questioning the

This research was funded by the Committee of Science and Higher Education of the Republic of Kazakhstan, grant number AP23489431.

underlying decision-making process of the trained model. ANNs are therefore unreliable in scientific and engineering scenarios where transparency is crucial.

Kolmogorov–Arnold Networks (KANs) offer a promising alternative to traditional ANNs by addressing their "black-box" nature. KAN, as implemented by Liu et al., represents a recently introduced and novel approach in machine learning, garnering growing interest for its strong theoretical foundation and potential to enhance interpretability in complex modeling tasks. Rooted in the Kolmogorov–Arnold representation theorem, KANs provide a mathematically robust framework for approximating any continuous multivariate function. The theorem states that any multivariate function can be represented as a finite sum of univariate functions composed with linear transformations [18]. Unlike standard ANNs, KANs have learnable activation functions. Their modular structure decomposes the complex relationships within a dataset into simpler, interpretable components, thereby making them more transparent in their decision-making process.

Recent studies have highlighted the potential of KANs in tabular data modeling. Gao et al. introduced TabKANet, which integrates KANs with Transformer architectures to unify numerical and categorical feature encoding, achieving superior or comparable performance to Gradient Boosted Decision Trees across various datasets [19]. Similarly, Poeta et al. benchmarked KANs against Multi-Layer Perceptrons (traditional neural networks) and found that KANs excel in accuracy and F1 scores, particularly for larger datasets, albeit at a higher computational cost [20]. These studies underscore the growing interest in KANs for handling complex tabular data.

Building on our previous work, where we applied various machine learning algorithms – including polynomial regression, dense neural networks, and cascade-forward neural networks – to predict oil recovery factors in polymer flooding scenarios [21], we now aim to explore the effectiveness of KANs in this context. Our earlier study demonstrated that polynomial regression achieved an R^2 score of 0.909, while dense and cascade-forward neural networks attained R^2 values of 0.908 and 0.906, respectively. These findings highlighted the potential of ML models in enhancing oil recovery predictions.

In this study, we explore the potential of Kolmogorov–Arnold Networks in predicting recovery factors for polymer flooding. By leveraging the inherent interpretability and flexibility of KANs, we aim to overcome the limitations of traditional ANNs, providing both high predictive accuracy and transparent decision-making insights. This novel application of KANs in the EOR domain represents a step toward integrating interpretable machine learning models in oil recovery research.

In this paper, we investigate whether KANs can offer comparable or superior predictive performance, with the added benefit of improved interpretability, thereby addressing some limitations associated with traditional neural network models. By applying KANs in the EOR domain, we aim to advance the integration of interpretable machine learning models into oil recovery research

II. Methodology

A. Dataset

The dataset used in this study consists of several key variables essential for modeling polymer flooding in EOR. These include absolute permeability (k), pressure (P), porosity (m), oil saturation (So), water saturation (Sw), oil viscosity ($visc$), polymer concentration (Cp), and the oil recovery factor (RF), which serves as the target variable. The dataset represents a diverse range of reservoir and operational conditions. Moreover, gaussian noise was added to the dataset to make the learning process challenging for the models and to promote generalization.

To provide a comprehensive evaluation, the dataset contains more than 160,000 samples generated synthetically, simulating realistic EOR scenarios. For detailed information on the dataset, the data generation process, its distribution and statistical properties, we refer readers to our previous work [21].

Normalization was applied using a Standard Scaler, which transforms each feature to have a mean of zero and a standard deviation of one. This step ensures that all features contribute equally to the learning process. The dataset was then split into training and testing sets with an 80:20 ratio to evaluate model performance.

B. Implementation

In this study, we utilized PyKAN, the official and open-source Python library specifically designed for implementing KANs [22]. PyKAN provides a comprehensive framework for constructing and training KAN models, facilitating their application in various machine learning tasks. The library offers functionalities for defining network architectures, managing training processes, and evaluating model performance.

Each KAN architecture is defined by several key parameters: the width, specifying the number of neurons in each layer as a list (e.g., [2, 3, 1] for 2 input neurons, 3 in the first hidden layer, and 1 output neuron); the grid size, determining the number of intervals for spline-based activation functions; and the spline order (K), representing the order of B-splines used in the activations. Activation functions are initialized to SiLU by default unless otherwise specified. The learning rate (lr) varies depending on the optimizer. Regularization parameters include λ (weight decay, set to 0.001 in all experiments) and $\lambda_{entropy}$ (an entropy-based penalty set to 0.1) to encourage sparsity and enhance model interpretability.

The evaluation metrics used are Mean Squared Error (MSE) and the R^2 score. MSE quantifies the average squared difference between observed and predicted values, providing a measure of the model's prediction accuracy; a lower MSE indicates better performance. R^2, on the other hand, represents the proportion of variance in the dependent variable that is predictable from the independent variables, offering insight into the model's explanatory power; values closer to 1 suggest a stronger fit. While both metrics are valuable, R^2 is often more informative in regression analysis, as it can be expressed as a percentage and is more robust in certain scenarios compared to MSE [23].

C. Experiments

To evaluate the performance of KANs in predicting recovery factors for polymer flooding, two sets of experiments were conducted, each employing a different optimizer to train the models. Specifically, the Adam optimizer and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer were used to compare their effectiveness in minimizing the MSE loss and improving model accuracy. Hyperparameters were chosen via manual tuning and heuristics, balancing model performance and computational efficiency.

All weights were initialized using a normal distribution. Spline parameters were initialized to approximate linear functions for faster convergence. Each experiment involved a fixed number of steps, specified below, to balance computational cost and convergence accuracy. Full-batch optimization was employed for both optimizers.

1) Experiment A: Training with Adam Optimizer

In the first set of experiments, the KAN models were trained using the Adam optimizer, which is well-suited for large datasets and adaptive learning rates. Adam was configured to use full-batch optimization rather than mini-batches, as it allows for consistency with the LBFGS optimizer and better convergence in this context. The experiments ran for 1000 steps because Adam typically requires more iterations to converge compared to LBFGS, which is compensated by its efficiency in handling noisy gradients.

The experimental configurations involved variations in the network width, grid size, K, lr, and training steps. For each configuration, the number of trainable parameters was recorded, and the training time was measured. Model performance was evaluated using both the training and testing MSE as well as the R^2 score. Learning rates were selected heuristically, with values ranging from $10^{-4}\ to\ 2 \times 10^{-3}$ to ensure sufficient updates without overshooting the minimum. The hyperparameter configurations are shown in Table I.

TABLE I. ADAM OPTIMIZER HYPERPARAMETERS – EXPERIMENT A

Case	*Width*	*Grid*	*K*	*lr*
A1	[7, 5, 5, 1]	20	3	0.001
A2	[7, 10, 5, 1]	15	2	0.0005
A3	[7, 5, 5, 5, 1]	10	3	0.0001
A4	[7, 5, 5, 1]	50	4	0.002
A5	[7, 10, 10, 1]	5	3	0.001

2) Experiment B: Training with LBFGS Optimizer

The second set of experiments employed the LBFGS optimizer, a quasi-Newton method that is effective for tasks requiring highly precise convergence, particularly with smooth loss functions. LBFGS was chosen because of its ability to leverage second-order information for rapid convergence in fewer steps, making it ideal for the smooth MSE loss used in this study.

The experiments were limited to 200 steps, as LBFGS generally achieves high convergence efficiency within a smaller number of iterations compared to Adam. Like Adam, LBFGS was also configured for full-batch optimization. This aligns with its inherent design to operate over entire datasets. The same dataset and preprocessing pipeline were used to ensure consistency and comparability between the two optimizers.

Similar to the first experiment, variations in network width, grid size, K, lr, and training steps were explored. Learning rates for LBFGS were also selected heuristically, with values between $10^{-4}\ to\ 2 \times 10^{-3}$, as this range provided stable and consistent convergence during tuning. Training time, MSE, and R^2 were measured and recorded for each configuration. The hyperparameter details can be seen in Table II.

TABLE II. LBFGS OPTIMIZER HYPERPARAMETERS – EXPERIMENT B

Case	*Width*	*Grid*	*K*	*lr*
B1	[7, 5, 5, 1]	30	3	0.001
B2	[7, 10, 5, 1]	20	2	0.0005
B3	[7, 5, 5, 5, 1]	10	3	0.0001
B4	[7, 5, 5, 1]	50	4	0.002
B5	[7, 10, 10, 1]	5	3	0.001

Both experiments utilized a high-performance NVIDIA RTX 4060Ti GPU to expedite training. For regularization, weight decay (λ) and an entropy-based penalty ($\lambda_{entropy}$) were applied to prevent overfitting and improve model robustness. Seeding was used to ensure reproducibility.

III. RESULTS

A. Experiment A: Adam Optimizer

The performance of KAN models trained using the Adam optimizer is summarized in Table III. Training times varied between 124.328 seconds (Case A2) and 429.850 seconds (Case A4). The variation in training time was directly influenced by the number of trainable parameters, with larger configurations requiring more computation. Test MSE values were consistently low, ranging from 0.000597 (Case A1) to 0.001006 (Case A3), while Test R^2 scores spanned from 0.834 (Case A3) to 0.902 (Case A1).

Case A1 achieved the best Test MSE (0.000597) and the highest Test R^2 (0.902), demonstrating that a balanced configuration with moderate grid size and network complexity can yield superior results. Case A5 also performed well, achieving a similar Test R^2 (0.901) with slightly higher computational efficiency compared to Case A4. Larger grid sizes and increased neuron counts, as seen in Cases A4 and A5, generally led to better R^2 scores. This affirms the importance of model complexity in explaining data variability.

Key insights from this experiment suggest that configurations with a grid size of 20-50 and moderate learning rates (~0.001) provide optimal performance, as they balance accuracy with training time.

TABLE III. EXPERIMENT A RESULTS

Case	*Parameters*	*Training time(s)*	*Test MSE*	*Test R^2*
A1	1885	158.693	0.000597	0.902
A2	2875	124.328	0.000609	0.900
A3	1710	147.418	0.001006	0.834
A4	3900	429.850	0.000624	0.897
A5	2520	137.841	0.000598	0.901

B. Experiment B: LBFGS Optimizer

The LBFGS optimizer demonstrated its strengths in rapid convergence with fewer training steps (200), but training times were longer overall compared to Adam due to the computational demands of the quasi-Newton method. As shown in Table IV, training times ranged from 482.973 seconds (Case B3) to 2029.817 seconds (Case B4). Test MSE values ranged from 0.000618 (Case B1) to 0.000692 (Case B4), while Test R^2 values remained high, mostly above 0.89, except for Case B4, which showed slightly degraded performance.

Case B1 exhibited the best overall performance, achieving a Test MSE of 0.000618 and a Test R^2 of 0.898. Case B5 closely followed with a Test MSE of 0.000619 and an identical Test R^2 of 0.898. Interestingly, smaller grid sizes (e.g., Case B3 with a grid of 10) allowed LBFGS to perform efficiently, achieving reasonable accuracy with reduced training times. Conversely, larger grid sizes, as seen in Case B4, suffered from instability and overfitted, leading to lower accuracy despite extended training times.

The results indicate that LBFGS excels with compact configurations, where its precision is leveraged to achieve fast convergence with minimal overfitting.

TABLE IV. EXPERIMENT B RESULTS

Case	*Parameters*	*Training time(s)*	*Test MSE*	*Test R^2*
B1	2535	1029.793	0.000618	0.898
B2	3500	566.849	0.000628	0.897
B3	1710	482.973	0.000664	0.891
B4	3900	2029.817	0.000692	0.886
B5	2520	633.499	0.000619	0.898

Across both experiments, the Adam optimizer emerged as the preferred choice for its ability to achieve lower training times with comparable accuracy. Case A1 demonstrated the best overall performance, with the lowest Test MSE and the highest Test R^2. Interestingly, the best performing model has one of the fewest parameters and balances grid size as well as K. This suggests that compact models are enough to reach a noteworthy predictive performance for KANs.

For tasks requiring relatively high precision with fewer steps, the LBFGS optimizer showed promise, particularly with compact configurations. Case B1 delivered the best performance for this optimizer.

In Fig 1, the predicted versus actual values demonstrate the strong predictive accuracy of the Case A1 model. The plot exhibits a close alignment of the points to the ideal reference line (y = x), which indicates a minimal deviation between the predicted and true recovery factor values. The relatively dense clustering of points along the diagonal in the plot speaks to the potential and reliability of KAN models in predicting recovery factors for polymer flooding scenarios.

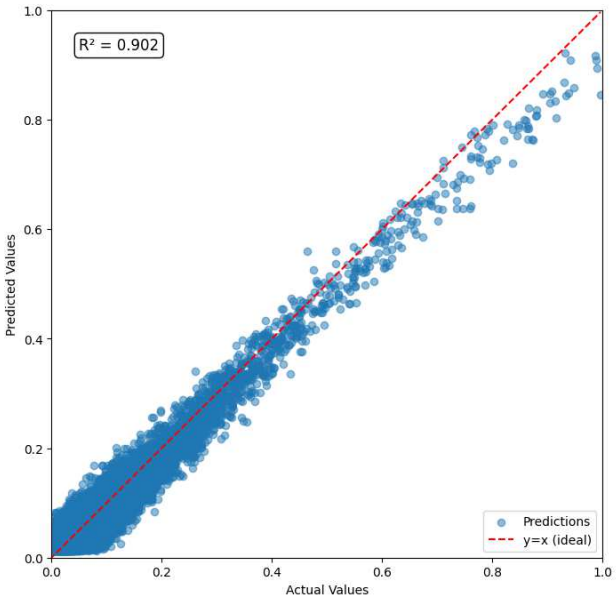

Fig. 1. Predicted vs. Actual Values: Case A1 model

In addition, we can visualize the decision-making process of the best performing KAN model (Case A1), as seen in Fig 2.

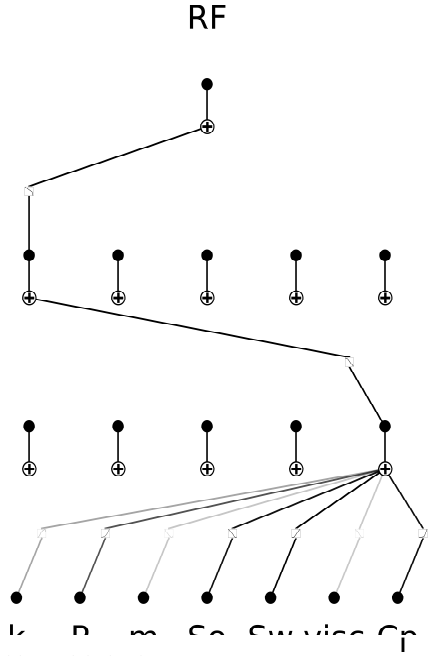

Fig. 2. Interpretable model plot for Case A1

The black lines in Fig 2 indicate active connections, with line thickness corresponding to the significance of the activation. Thinner, lighter lines represent less significant contributions, while the thicker connections highlight the critical pathways for the prediction. This hierarchical and interpretable architecture provides insights into how input features influence the output. This emphasizes the model's ability to integrate both sparse and dense relationships.

Moreover, it is possible to study the activation functions in the model. Fig 3 dives deeper into the individual activation functions of the first layer for three selected connections: (0, 0, 0), (0, 0, 1), and (0, 0, 2). These coordinates indicate the first layer, the first neuron in that layer, and its connections to the first, second, and third input features, respectively. Each subplot demonstrates how the network transforms input values through the activation function of a specific connection. For example, the first activation function exhibits a sinusoidal behavior, while others display distinct nonlinear transformations.

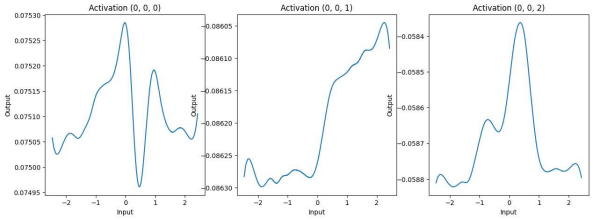

Fig. 3. Visualization of Activation Functions for the First Layer in Case A1 model

The feature attribution analysis, as illustrated in Fig. 4, demonstrates how we can directly extract feature importance from the model itself using the built-in attribution mechanism in KAN. As seen in the visualization, Sw emerges as the most critical feature, with a significantly higher attribution score compared to others. Meanwhile, $Cp, So,$ and P show moderate contributions, while $k, m,$ and $visc$ have minimal impact.

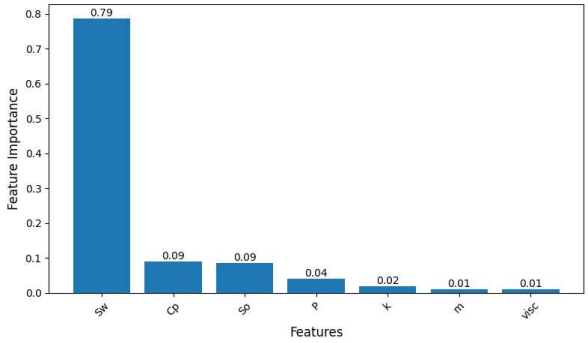

Fig. 4. Feature attribution scores of Case A1 model

It is also possible to extract the specific impact of each feature on individual neurons within its architecture. This is done by quantifying how much each input feature influences the activations of particular neurons in a given layer. For instance, it is possible to examine how features contribute to the activation of specific pathways within the network. This neuron-level feature attribution provides insights into how it interacts with the hierarchical structure of the model. For instance, the feature attributions for neuron 3 in layer 1 are displayed in Fig 5.

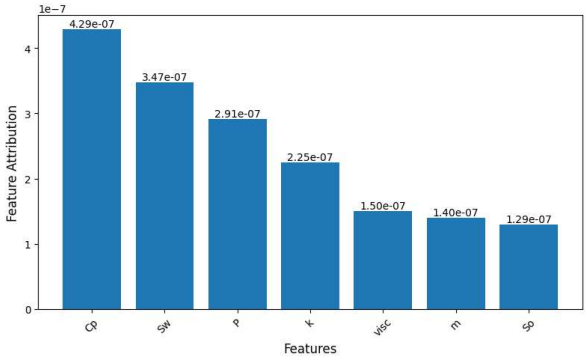

Fig. 5. Feature attribution scores for neuron 3 in layer 1 (Model Case A1)

To visually assess the model's important features and activation functions further, we can prune the model, a process that involves removing less significant input features and connections based on their contribution to the output. This approach simplifies the model and retains only the dominant pathways that are critical to its decision-making process. As shown in Fig. 6, the pruned KAN model focuses on the most influential features, such as Sw, Cp, and So, while eliminating features like m that contribute minimally to the predictions of RF. By removing these weaker connections, pruning enhances the interpretability of the model and makes it easier to understand which features and pathways are most impactful.

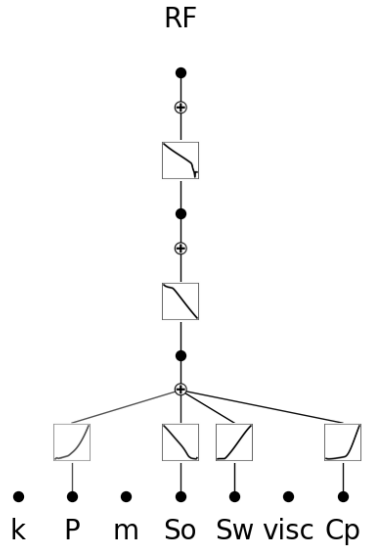

Fig. 6. Pruned model plot for Case A1

IV. Conclusion

In this study, we explored the application of KANs for predicting oil recovery factors in polymer flooding scenarios. Compared to our previous work using a Deep Neural Network (DNN) with 43,265 trainable parameters, the KAN models demonstrated comparable predictive performance with significantly fewer parameters. The best-performing KAN achieved a Test MSE of 0.000597 and a Test R^2 of 0.902, closely matching the DNN's performance (Test R^2 of 0.908) while utilizing only 1,885 parameters.

Although KANs did not surpass the predictive accuracy of the DNN, they offer the distinct advantage of improved interpretability due to their modular structure and learnable activation functions. This interpretability allows for greater insight into the decision-making process of the model, which is particularly valuable in scientific and engineering applications where transparency is crucial. By analyzing activation functions and hierarchical representations, we can better understand how input features influence the output and thereby facilitate more informed decision-making in the EOR field.

This work contributes to the EOR domain by introducing an interpretable variant of ANNs, demonstrating that KANs can effectively model complex reservoir behaviors with reduced model complexity. The ability to achieve high predictive accuracy with fewer parameters while being interpretable (through feature attribution, learnable and observable activation functions, and pruning) highlights the efficiency of KANs and their potential to bridge the gap between predictive performance and model transparency.

Future research will focus on studying the interpretability of KANs to better understand the decision-making process and underlying patterns in predictions. Additionally, integrating domain-specific physics-based constraints and priors into the learning process could further improve model reliability and accuracy in capturing complex reservoir dynamics. To ensure their robustness and adaptability in diverse scenarios, expanding the application of KANs to other EOR techniques and validating their performance on real-world datasets will also be key areas of exploration.

REFERENCES

[1] M. Ahmadi and Z. Chen, "Challenges and future of chemical assisted heavy oil recovery processes," Advances in Colloid and Interface Science, vol. 275, p. 102081, Jan. 2020, doi: 10.1016/j.cis.2019.102081.

[2] Y. Yuan, A. Cheng, D. Yang, C. Li, and Y. Liu, "Theory and application of numerical simulation method of capillary force enhanced oil production," Appl. Math. Mech.-Engl. Ed., vol. 36, no. 3, pp. 379–400, Mar. 2015, doi: 10.1007/s10483-015-1917-6.

[3] K. G. Salem, A. M. Salem, M. A. Tantawy, A. A. Gawish, S. Gomaa, and A. N. El-hoshoudy, "A Comprehensive Investigation of Nanocomposite Polymer Flooding at Reservoir Conditions: New Insights into Enhanced Oil Recovery," J Polym Environ, vol. 32, no. 11, pp. 5915–5935, Nov. 2024, doi: 10.1007/s10924-024-03336-z.

[4] T. Nassan and M. Amro, "Finite Element Modeling of Immiscible Two-Phase Flow in Oil Reservoirs," Jan. 01, 2022, Social Science Research Network, Rochester, NY: 4898872. doi: 10.2139/ssrn.4898872.

[5] W. Ozowe, A. D. Ogbu, and A. H. Ikevuje, "Data science's pivotal role in enhancing oil recovery methods while minimizing environmental footprints: An insightful review," Computer Science & IT Research Journal, vol. 5, no. 7, Art. no. 7, Jul. 2024, doi: 10.51594/csitrj.v5i7.1348.

[6] W. A. Khan, Z. Rui, T. Hu, Y. Liu, F. Zhang, and Y. Zhao, "Application of Machine Learning and Optimization of Oil Recovery and CO2 Sequestration in the Tight Oil Reservoir," SPE Journal, vol. 29, no. 06, pp. 2772–2792, Jun. 2024, doi: 10.2118/219731-PA.

[7] A. Roustazadeh, B. Ghanbarian, F. Male, M. B. Shadmand, V. Taslimitehrani, and L. W. Lake, "Estimating oil recovery factor using machine learning: Applications of XGBoost classification," Oct. 28, 2022, arXiv: arXiv:2210.16345. doi: 10.48550/arXiv.2210.16345.

[8] D. Bui, A.-M. Koray, E. Appiah Kubi, A. Amosu, and W. Ampomah, "Integrating Machine Learning Workflow into Numerical Simulation for Optimizing Oil Recovery in Sand-Shale Sequences and Highly Heterogeneous Reservoir," Geotechnics, vol. 4, no. 4, Art. no. 4, Dec. 2024, doi: 10.3390/geotechnics4040055.

[9] F. Krasnov, N. Glavnov, and A. Sitnikov, "A Machine Learning Approach to Enhanced Oil Recovery Prediction," in Analysis of Images, Social Networks and Texts, W. M. P. van der Aalst, D. I. Ignatov, M. Khachay, S. O. Kuznetsov, V. Lempitsky, I. A. Lomazova, N. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, A. V. Savchenko, and S. Wasserman, Eds., Cham: Springer International Publishing, 2018, pp. 164–171. doi: 10.1007/978-3-319-73013-4_15.

[10] A. Mohsenatabar Firozjaii and H. R. Saghafi, "Review on chemical enhanced oil recovery using polymer flooding: Fundamentals, experimental and numerical simulation," Petroleum, vol. 6, no. 2, pp. 115–122, Jun. 2020, doi: 10.1016/j.petlm.2019.09.003.

[11] S. G. Dardaganian, "The Application of the Buckley-Leverett Frontal Advance Theory to Petroleum Recovery," Journal of Petroleum Technology, vol. 10, no. 04, pp. 49–52, Apr. 1958, doi: 10.2118/835-G.

[12] H. Saberi, E. Esmaeilnezhad, and H. J. Choi, "Artificial Neural Network to Forecast Enhanced Oil Recovery Using Hydrolyzed Polyacrylamide in Sandstone and Carbonate Reservoirs," Polymers, vol. 13, no. 16, Art. no. 16, Jan. 2021, doi: 10.3390/polym13162606.

[13] Y. Cheraghi, S. Kord, and V. Mashayekhizadeh, "A two-stage screening framework for enhanced oil recovery methods, using artificial neural networks," Neural Comput & Applic, vol. 35, no. 23, pp. 17077–17094, Aug. 2023, doi: 10.1007/s00521-023-08557-2.

[14] S. Le Van and B. H. Chon, "Evaluating the critical performances of a CO2–Enhanced oil recovery process using artificial neural network models," Journal of Petroleum Science and Engineering, vol. 157, pp. 207–222, Aug. 2017, doi: 10.1016/j.petrol.2017.07.034.

[15] H. Vo Thanh, Y. Sugai, and K. Sasaki, "Application of artificial neural network for predicting the performance of CO2 enhanced oil recovery and storage in residual oil zones," Sci Rep, vol. 10, no. 1, Art. no. 1, Oct. 2020, doi: 10.1038/s41598-020-73931-2.

[16] M.-R. Mohammadi, A. Hemmati-Sarapardeh, M. Schaffie, M. M. Husein, and M. Ranjbar, "Application of cascade forward neural network and group method of data handling to modeling crude oil pyrolysis during thermal enhanced oil recovery," Journal of Petroleum Science and Engineering, vol. 205, p. 108836, Oct. 2021, doi: 10.1016/j.petrol.2021.108836.

[17] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On Interpretability of Artificial Neural Networks: A Survey," IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 5, no. 6, pp. 741–760, Nov. 2021, doi: 10.1109/TRPMS.2021.3066428.

[18] Z. Liu et al., "KAN: Kolmogorov-Arnold Networks," Jun. 16, 2024, arXiv: arXiv:2404.19756. doi: 10.48550/arXiv.2404.19756.

[19] W. Gao, Z. Gong, Z. Deng, F. Rong, C. Chen, and L. Ma, "TabKANet: Tabular Data Modeling with Kolmogorov-Arnold Network and Transformer," Oct. 02, 2024, arXiv: arXiv:2409.08806. doi: 10.48550/arXiv.2409.08806.

[20] E. Poeta, F. Giobergia, E. Pastor, T. Cerquitelli, and E. Baralis, "A Benchmarking Study of Kolmogorov-Arnold Networks on Tabular Data," Jun. 20, 2024, arXiv: arXiv:2406.14529. doi: 10.48550/arXiv.2406.14529.

[21] T. Imankulov, Y. Kenzhebek, S. D. Bekele, and E. Makhmut, "Enhancing Oil Recovery Predictions by Leveraging Polymer Flooding Simulations and Machine Learning Models on a Large-Scale Synthetic Dataset," Energies, vol. 17, no. 14, Art. no. 14, Jan. 2024, doi: 10.3390/en17143397.

[22] Z. Liu, KindXiaoming/pykan. (Nov. 19, 2024). Jupyter Notebook. Accessed: Nov. 20, 2024. [Online]. Available: https://github.com/KindXiaoming/pykan

[23] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in Artificial Intelligence Application in Networks and Systems, R. Silhavy and P. Silhavy, Eds., Cham: Springer International Publishing, 2023, pp. 15–25. doi: 10.1007/978-3-031-35314-7_2.